UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES
ET INFORMATIQUE APPLIQUÉES

PAR
MATHIEU DUGRÉ

CONCEPTION ET RÉALISATION D'UN ENTREPÔT DE DONNÉES : INTÉGRATION À UN SYSTÈME EXISTANT ET ÉTAPE NÉCESSAIRE VERS LE FORAGE DE DONNÉES

MARS 2004

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Sylvain Delisle, directeur de recherche
Département de mathématiques et d'informatique
à l'Université du Québec à Trois-Rivières

M. Ismaïl Biskri, évaluateur
Département de mathématiques et d'informatique
à l'Université du Québec à Trois-Rivières

M. François Meunier, évaluateur
Département de mathématiques et d'informatique
à l'Université du Québec à Trois-Rivières

CONCEPTION ET RÉALISATION D'UN ENTREPÔT DE DONNÉES : INTÉGRATION À UN SYSTÈME EXISTANT ET ÉTAPE NÉCESSAIRE VERS LE FORAGE DE DONNÉES

Mathieu Dugré

SOMMAIRE

Ce projet visait à créer un entrepôt de données pour supporter le forage de données à partir d'une base de données au Laboratoire de recherche sur la performance des entreprises (LaRePE). Le LaRePE est un laboratoire de l'Institut de recherche sur les PME, à l'Université du Québec à Trois-Rivières. Le système PDG comprend une série de questionnaires, plusieurs utilitaires informatiques et une base de données. Le but de ce système est la création de rapports de *benchmarking* pour des PME manufacturières. Ces rapports présentent les résultats d'un diagnostic de la performance d'une PME relativement à un groupe témoin de PME. Les utilitaires développés depuis plusieurs années ne permettent pas d'accomplir toutes les tâches désirées avec les données recueillies. Certaines tâches sont difficiles à réaliser et requièrent beaucoup de temps. Un entrepôt de données permet de régler plusieurs de ces problèmes en agissant comme intermédiaire entre la saisie des données et leur utilisation à des fins de recherche, ou pour créer des rapports de *benchmarking*. Le forage de données peut aider à approfondir la compréhension des données du système. Un entrepôt de données est donc créé, sa conception se base sur une liste de problèmes identifiés par le personnel qui utilise le système. Plusieurs de ces problèmes sont réglés, et de nouveaux logiciels sont ajoutés pour supporter l'analyse de données en ligne (*OLAP*) et le forage de données. Un processus de mise à jour des données est instauré. Les différentes versions des enregistrements des bases de données sont conservées pour permettre de créer un historique des données. De nouveaux utilitaires sont aussi ajoutés pour aider les utilisateurs. Il s'agit d'un dictionnaire de variables et du *Dataset Maker*. Ces utilitaires permettent de localiser et d'extraire des données très rapidement de l'entrepôt, ce qui facilite la recherche et le forage de données. Le logiciel permettant de créer le PDG n'a pas été relié complètement à l'entrepôt au moment d'écrire ce document, mais le sera prochainement. Des vérifications préliminaires indiquent toutefois une préparation des données cinq fois plus rapide comparativement à l'ancien système. Certains utilitaires sont à refaire pour profiter de l'entrepôt, et des projets comme la réingénierie des logiciels de rapport et de questionnaire sont prévus pour profiter des nouvelles possibilités offertes par l'entrepôt. Le forage de données est prêt à servir les chercheurs pour les prochains projets de recherche.

DESIGN AND IMPLEMENTATION OF A DATA WAREHOUSE: INTEGRATION WITHIN AN EXISTING SYSTEM AND A NECESSARY STEP TOWARD DATA MINING

Mathieu Dugré

ABSTRACT

The goal of this project was to create a data warehouse to support data mining using a database that belongs to the Laboratory for research on business performance (LaRePE). LaRePE is a laboratory of the research Institute for SMEs, at the Université du Québec à Trois-Rivières (UQTR). The PDG system consists of several questionnaires, computer utilities and a database. This system creates benchmarking reports for small and medium enterprises (SMEs). These reports offer a diagnostic on the performance of a SME by contrasting the particular SME with an appropriate group of SMEs. The software utilities created for the last several years are not as flexible as the users would want, and the data manipulations take too much time. A data warehouse that allows data mining is an interesting solution, so it becomes the foundation of a new system. It receives all the data and optimizes the creation of benchmarking reports, research projects, data mining, and any other uses for the available data. Its design is based on the needs of the users, who have previously identified several problems with the old system. Several of these problems have been solved. Data mining and On-Line Analytical Process (OLAP) software are now available to the users. An extract, transform and load (ETL) procedure has been implemented to keep all the data in the data warehouse up to date. Historic information is also saved to allow researchers to analyse data on an historical basis. New web tools have been created: a dictionary for variables and the Dataset Maker. These tools use metadata in the data warehouse to create data sets for statistical research, reports and data mining. The software used to create PDG reports has not been linked to the data warehouse at the time of writing this document, but will be shortly. However, preliminary results show that data preparation for reports is over five times faster with the data warehouse. To fully benefit from this data warehouse, all utilities using data from the database will have to be modified. One of these applications is the application used to capture data from paper questionnaires, and of course there is the PDG report. But data mining is ready and can be used by the researchers for their new research projects.

REMERCIEMENTS

Je remercie mes parents pour m'avoir supporté durant toutes mes études universitaires. Leur appui constant m'a permis d'accomplir de grandes choses et de partir bien préparé pour la vie.

Je remercie Jean-François Beaudoin pour m'avoir aidé dans le codage du dictionnaire de variables et du *Dataset Maker*. Je tiens aussi à remercier Julie Croteau, Catherine Therrien et Daniel Pitre pour leur soutien professionnel lors des étapes d'analyse des besoins et pour avoir répondu à mes nombreuses questions. Je remercie tout particulièrement Josée St-Pierre, directrice du LaRePE, pour m'avoir accordé la liberté, les ressources et le temps nécessaires à la réalisation de ce projet de maîtrise.

Je remercie le personnel du Service de l'informatique de l'Université du Québec à Trois-Rivières pour leur disponibilité et leur support technique. Je remercie Louis Brouillette pour ses conseils avec *TOAD* et *Oracle*, de même que Patrick Cossette et Alain Morrissette pour leur aide dans la préparation du serveur Web et des programmes sur *AIX*.

Finalement, je remercie le professeur Sylvain Delisle pour sa direction et son support constant. Sans sa persévérance, sa vivacité d'esprit, son souci du détail et son ambition, ce mémoire n'aurait pas été possible.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

LISTE DES FIGURES

CHAPITRE 1

INTRODUCTION

Il n'est pas rare d'entendre des gestionnaires parler de la difficulté d'obtenir des informations utiles à partir de systèmes informatiques qu'ils ont payés de véritables fortunes. À l'ère où le téra-octet est accessible même aux particuliers, les organisations doivent absolument se doter de systèmes capables de gérer les montagnes de données qu'il est maintenant facile d'accumuler avec toutes les techniques d'acquisition disponibles. Que cette accumulation soit volontaire ou non, sans une infrastructure solide, de bonnes méthodes d'analyse et des rapports judicieusement préparés, la majorité de ces informations seront reléguées aux oubliettes des systèmes d'archivage. Cette notion est encore plus importante pour les organisations dont les activités dépendent des données, comme les laboratoires de recherche.

De nouvelles techniques ont été développées pour exploiter ces quantités astronomiques de données, et les utilisateurs peuvent maintenant compter sur des outils comme l'analyse en ligne des données (*OLAP*) et le forage de données, soit pour obtenir certains renseignements en un coup d'œil, soit pour creuser davantage et faire l'inventaire des informations supportées par les données recueillies. Mais avant de pouvoir en arriver à un accès simplifié aux données, il faut bâtir l'infrastructure. Et dans ce cas précis, l'infrastructure est un entrepôt de données. Cet entrepôt est nécessaire pour réorganiser les données en une forme utilisable par les divers outils qui pourront par la suite accéder à ces données nettoyées et intégrées à partir de nombreuses sources différentes et hétérogènes. Avec une infrastructure comme un entrepôt de données et des techniques adaptées, tels *OLAP* et le forage de données, il est alors possible aux gestionnaires, aux analystes et aux chercheurs de réellement profiter des montagnes de données accumulées dans leurs systèmes informatiques.

1.1 Contexte

Le Laboratoire de recherche sur la performance des entreprises (LaRePE) fait partie de l'Institut de recherche sur les petites et moyennes entreprises (INRPME). L'INRPME est un des instituts de recherche de l'Université du Québec à Trois-Rivières (UQTR). Le projet décrit dans ce mémoire est conçu à l'UQTR dans le cadre du développement d'un nouveau système d'information au LaRePE. Le LaRePE est un laboratoire de recherche qui emploie de nombreux étudiants dans le cadre de stages, de projets de maîtrise ou de doctorat. Il y a aussi quelques professionnels de recherche qui assurent un fonctionnement continu des divers projets, et des chercheurs qui préparent les projets, supervisent les activités et effectuent différentes recherches à l'aide des informations cumulées au Laboratoire. C'est un contexte très dynamique où tout peut changer du jour au lendemain, incluant les priorités, le personnel étudiant et les projets.

Un système informatique comportant un questionnaire électronique, une base de données et un logiciel de création automatique de rapport a été créé au LaRePE. Ce système supporte le rapport PDG qui est un rapport de *benchmarking* (voir l'annexe A). Le rapport peut être utilisé par des entrepreneurs pour comparer leur entreprise à un groupe témoin selon certains critères, comme le secteur d'activité ou la région

administrative de leur entreprise. Le PDG a été développé par une équipe multidisciplinaire composée de chercheurs, d'assistants et de professionnels de recherche en utilisant des outils disponibles sur PC comme *Excel* et *Access* de Microsoft. La première version du logiciel utilisait des formulaires et des tables *Access* pour saisir les données à partir des questionnaires imprimés, mais maintenant c'est une base de données *Oracle* qui fait ce travail. Une fois que les questionnaires sont saisis dans la base de données, le logiciel statistique *SAS*[1] est utilisé pour calculer différentes valeurs récupérées par un classeur *Excel*. Ce classeur utilise des macros et des formules pour afficher les informations sous forme graphique, et il prépare aussi des commentaires automatiquement.

Le système d'information du LaRePE est en constante évolution, et il arrive que certains ajouts majeurs soient apportés comme de nouveaux questionnaires et de nouvelles pages de rapport, ou des modifications touchant l'entretien et la flexibilité des applications. Il est passé par de nombreuses révisions majeures, dont 5 révisions du rapport PDG et 2 de la base de données manufacturière. La conception des éléments du système devrait naturellement aider à supporter ces besoins évolutifs, ce qui n'est pas le cas. D'ailleurs, un manque flagrant de documentation mine considérablement tout effort de modification, et le manque d'encapsulation oblige les programmeurs à se familiariser avec l'ensemble du système avant de pouvoir apporter le moindre changement. Cet état de fait est particulièrement contre-productif lorsqu'un projet arrive à la dernière minute et doit être terminé rapidement, ce qui arrive de temps à autre. Un transfert des données de la base de données *Access* a été fait vers un serveur avec *Oracle 8i*, et les formulaires électroniques de saisie ont été refaits en *PL/SQL*. Bien que ce transfert ait amélioré la sécurité des données du point de vue des copies de sauvegarde (assurées par le Service de l'informatique de l'UQTR) et l'accessibilité (formulaires PL/SQL sur un site Web), certains problèmes n'ont pas été résolus (voir la sous-section 3.1.3). Par exemple, une des raisons principales qui ont mené au transfert précédent vers *Oracle* est le besoin de formulaires de saisie Web, mais les formulaires obtenus en *PL/SQL* sont loin de répondre aux exigences initiales. De plus, la documentation des données est éparpillée et difficilement accessible aux utilisateurs du système.

Une fois ces données conservées, plusieurs outils sont nécessaires pour les exploiter. Le forage de données est vu comme une solution pour une utilisation plus globale des données qui sont accumulées dans les bases de données du Laboratoire. Une approche classique par analyse statistique nécessite la formulation d'hypothèses à l'avance et de leur vérification avec des données. À cause de la grande quantité de données présentes dans les bases de données, il est très difficile pour un analyste de pouvoir saisir à l'avance toutes les nuances possibles pour formuler les bonnes hypothèses. Le forage de données passe plutôt par le chemin inverse pour générer un modèle. Il faut commencer par formuler une question de haut niveau, par exemple : qu'est-ce qui définit une entreprise performante ? Ensuite, un jeu de données est créé et les entreprises performantes sont marquées, sans pour autant savoir pourquoi elles sont performantes. Avec l'analyse statistique, l'analyste doit formuler lui-même ses

[1] http://www.sas.com

hypothèses, puis les valider auprès des données. Par contre, le forage de données donne une série d'outils qui permettent de fabriquer des modèles à partir des données, ce qui revient à demander à l'outil : trouve-moi toutes les hypothèses qui sont supportées par ce jeu de données. Le discernement humain n'est pas remplacé par la machine, l'analyste doit bien comprendre le domaine et les données qu'il utilise. Mais de cette façon, de nombreuses informations peuvent être identifiées, informations qui sont cachées implicitement dans les données et que l'analyste aurait bien pu ignorer parmi les montagnes de données.

Les projets entourant le rapport PDG mobilisent plusieurs ressources, mais il y a de nombreux autres logiciels développés au LaRePE, dont certains qui utilisent aussi la base de données manufacturière. Plusieurs sont des applications de *benchmarking* qui peuvent bénéficier de la présence d'un entrepôt pour réaliser leurs calculs. C'est pourquoi une des motivations internes de la création d'un entrepôt est la possibilité de relier tous ces logiciels entre eux, en utilisant de nouvelles applications de questionnaires et de rapports plus simples à modifier pour créer une plateforme de *benchmarking*. C'est une vision à plus long terme, mais cette possibilité justifie les moyens mis en œuvre pour réviser les processus de manipulation des données et de production des rapports.

1.2 La problématique

Les données du LaRePE provenant de la base de données manufacturière, ainsi que de plusieurs autres projets, sont trop difficiles à exploiter pour des fins de recherches scientifiques, selon les chercheurs qui les utilisent. Il faut trouver une façon de restructurer ces données pour qu'elles soient plus facilement accessibles aux personnes qui en font usage, sans pour autant négliger l'aspect de sécurité entourant les bases de données du Laboratoire. Pour répondre à ces besoins d'accès simplifié aux données, il faut développer un environnement d'exploitation des données du Laboratoire dans un contexte de production continue de statistiques et de rapports. Cet environnement doit être testé et fonctionnel à la fin du travail de maîtrise, et il doit représenter une solution aux problèmes exprimés par les chercheurs du LaRePE.

Les problèmes d'accès aux donnés font l'objet de questionnements depuis plusieurs années, au LaRePE. Ces problèmes ont été formellement identifiés dans le cadre d'un rapport (Dugré et Delisle, 2003) qui explorait différentes pistes de solution, notamment à l'aide de scénarios qui ont été présentés aux chercheurs impliqués au LaRePE. Suite à ce rapport, et après plusieurs rencontres avec les chercheurs du LaRePE, le scénario retenu est celui présentant une réingénierie de toutes les applications (pour la recherche scientifique) et rapports utilisant les données du Laboratoire. Toujours selon ce scénario, c'est un entrepôt de données qui sera au cœur du stockage et de l'exploitation des données au Laboratoire, et le forage de données sera une des nouvelles méthodes utilisées pour faire de la recherche à partir des données.

Ce travail de maîtrise est un ajout à un système existant. il est de nature hautement appliquée : il doit répondre à des problèmes concrets par une solution concrète, dans un contexte où les données servent continuellement et où les ressources sont limitées (temps et argent consacrés au développement de ce projet), afin de permettre de nouveaux développements à partir des solutions proposées et implémentées. À cause de ces contraintes, il n'est pas possible ni souhaitable de se départir des systèmes existants de collecte des données, c'est-à-dire des bases de données qui sont utilisées pour saisir les questionnaires.

C'est pourquoi l'ajout d'un entrepôt de données est la solution retenue : les bases de données existantes font leur travail et contiennent beaucoup de données, et il y a de nombreuses applications utilisées qui produisent continuellement des résultats pour la recherche et qui génèrent des rapports. Le travail nécessaire pour refaire de nouvelles structures de données, ce qui implique aussi de refaire tous les systèmes qui sont actuellement en production, serait beaucoup trop grand et prendrait trop de temps, en considérant les ressources limitées du LaRePE. L'avantage de l'entrepôt de données, c'est qu'il est bâti et alimenté à partir des bases de données existantes, et ce même si les structures de données internes de l'entrepôt sont créées pour être plus adaptées aux nouveaux besoins d'accès aux données. Contrairement à un *système parallèle*[2] qui devrait être alimenté séparément, l'entrepôt de données se met à jour régulièrement à partir des systèmes qui sont appelés à être remplacés. Il permet aussi de documenter les bases de données existantes, et de faire un passage progressif vers les nouveaux systèmes de production de statistiques et de rapports, puisque les anciennes bases de données sont toujours supportées et alimentées en données.

Afin de guider la réflexion sur les améliorations des bases de données du LaRePE, cinq questions ont été posées. Ces questions sont une synthèse des interrogations des chercheurs du LaRePE par rapport à l'amélioration de l'utilisation des données existantes dans les systèmes du Laboratoire. De façon générale, les interrogations des chercheurs tournent autour de quatre points principaux : comment et où stocker les données de façon sécuritaire, comment faire une *base de variables* (ou comment documenter les variables), comment accéder aux données et comment utiliser ces données.

[2] Un système en utilisation parallèle pourrait être vu comme une entité distincte sans lien aux bases de données actuellement utilisées, avec sa propre base de données. Il devrait alors être entretenu séparément, et toutes les données devraient alors être saisies dans les deux systèmes en utilisation, ce qui double la charge de travail à la saisie. De tels systèmes ont d'ailleurs déjà coexistés au LaRePE.

C'est pourquoi les cinq questions suivantes sont présentées dans ce travail, la première servant surtout à introduire les entrepôts de données dans le contexte du Laboratoire :

1. Est-ce qu'il y a lieu de créer un entrepôt de données pour le LaRePE ?
2. Comment consolider différentes sources de données dans un entrepôt de données qui pourra servir à la recherche ?
3. Comment documenter l'entrepôt et maintenir la documentation à jour ?
4. Comment permettre aux chercheurs et aux étudiants d'exploiter le plus simplement possible les données de cet entrepôt, avec les outils de leur choix ?
5. Quels outils d'analyse (statistique, forage de données, etc.) pourraient permettre d'améliorer l'exploitation de cet entrepôt ?

Ces questions sont reprises dans la conclusion et les éléments de réponse sont récupérés à partir des différentes parties de ce mémoire. Les résultats exprimés sont des solutions qui sont apportées par l'entrepôt de données, mais il y a d'autres éléments qui ne peuvent pas être réglés uniquement par l'ajout d'un entrepôt de données, et ils devront faire l'objet de nouveaux projets avant d'être vraiment résolus. Le choix des logiciels, leur intégration à l'entrepôt ainsi qu'une démonstration de leur fonctionnement est faite. D'autres solutions sont décrites, mais laissées complètement à d'autres projets, comme la création d'un nouveau rapport de *benchmarking* utilisant l'entrepôt et supportant une interface Web sécurisée.

1.3 Les entrepôts de données

Les entrepôts de données (*data warehousing*) font partie d'un engouement pour la *business intelligence*. Un entrepôt de données sert à concentrer les données disséminées dans l'entreprise et à les réunir en une série de structures documentées afin de permettre aux analystes et aux décideurs d'y accéder rapidement sans avoir besoin de connaissances techniques de programmation.

Les entrepôts de données permettent de supporter de nouvelles applications analytiques à partir de systèmes déjà existants. Ces systèmes existants sont généralement moins bien adaptés aux besoins des analystes. Ils ont normalement été conçus pour faire fonctionner l'entreprise au quotidien, comme pour les transactions des caisses dans un supermarché. Les entrepôts de données sont aussi vus comme une solution au problème des *legacy systems*[3] parce qu'ils permettent de développer de nouvelles solutions aux problèmes d'aujourd'hui en conservant les systèmes développés avec d'anciennes technologies (aujourd'hui obsolètes) ou des systèmes peu adaptés à l'analyse de données. En restructurant les données de l'entreprise dans un système conçu pour l'analyse, il est alors possible d'enlever le fardeau de ce traitement qui aurait autrement dû être placé sur les systèmes utilisés pour les

[3] Système informatique issu d'une génération précédente de système informatique et qui continue d'être utilisé, après avoir été adapté à un système plus contemporain. (Office de la langue française, 2002)

traitements transactionnels. On peut alors conserver l'ancien système en lui apportant un minimum de modifications pour supporter les nouveaux besoins d'analyse.

Dans le cas du LaRePE, un entrepôt de données constitue une base solide pour le forage de données, et pour le support de toutes les autres opérations analytiques effectuées à partir des données. C'est dans cette optique que le développement de nouveaux logiciels est orienté, et la transition pourra être graduelle, car rien n'oblige à refaire tout le système en même temps. La première phase, l'entrepôt, permettra même de mieux jauger les avantages qui pourront être retirés de la réingénierie des autres applications de *benchmarking*.

1.4 Plan et contenu

Un entrepôt de données a été créé pour remplacer certains éléments du système de production des rapports, et pour compléter d'autres éléments afin de supporter de nouvelles applications et méthodes analytiques. Cet entrepôt de données a été activé en novembre 2003. Il fait maintenant partie d'un système d'information plus vaste permettant la saisie des questionnaires qui sont transmis au Laboratoire, ainsi qu'à la production de statistiques à partir de ces données. Le forage de données est inclus à ce nouveau système et permettra de nouvelles approches pour la recherche. L'entrepôt sera aussi appelé à supporter toutes les étapes de la production des rapports de *benchmarking*, comme le PDG. Mais pour l'instant, il reste encore du travail à faire sur le logiciel de production du rapport, et ces modifications ne font pas partie du projet de conception de l'entrepôt. Le logiciel de production de rapport a toutefois été relié à l'entrepôt pour démontrer la faisabilité de ce lien. Cet exemple donne un avant-goût des avantages provenant de l'utilisation de l'entrepôt comme support principal pour les données analytiques.

Dans le chapitre 2, on affiche une liste des concepts fondamentaux des entrepôts de données. Une description des entrepôts de données est faite, et les principaux constituants d'un bon entrepôt de données sont décrits. Certains points à éviter, comme les schémas en flocons, sont aussi décrits. Quelques exemples d'utilisation des entrepôts de données en entreprise illustrent des cas réels en entreprise. Une courte introduction au forage de données et à la préparation des données est aussi présentée. Le chapitre 3 représente les résultats de l'analyse et de la conception du nouveau système ayant comme base un entrepôt, tout en insistant sur les utilisations classiques des données, comme l'analyse statistique et la production des rapports. Les problèmes de l'ancien système sont présentés, et des solutions sont proposées. Les éléments de conception du nouvel entrepôt sont alors présentés et les solutions reprises afin de résoudre les problèmes énumérés dans l'analyse. Deux nouveaux outils ont été créés spécifiquement pour l'entrepôt. Le dictionnaire de variables documente les variables de l'entrepôt, et le *Dataset Maker* est une sorte d'assistant Web permettant de créer des jeux de données à partir de l'entrepôt. Une sous-section présente les choix technologiques effectués pour l'implémentation de l'entrepôt de données. Quelques exemples d'utilisations classiques des données démontrent le contexte dans lequel l'entrepôt pourra servir en s'en tenant uniquement aux

utilisations des données aux fins de statistiques. Dans le chapitre 4, on mentionne les nouvelles applications possibles avec l'ajout d'un entrepôt de données. Deux techniques sont illustrées plus en détail, il s'agit du processus analytique en ligne (*OLAP*) et du forage de données. Afin de supporter ces techniques, un magasin de données à modèle dimensionnel a été créé. Certains logiciels ont été retenus pour permettre aux utilisateurs de commencer immédiatement leurs essais. D'autres applications sont aussi possibles grâce aux nouvelles structures présentes dans l'entrepôt.

Le chapitre 5 contient quelques résultats de l'utilisation du nouveau système. Certains problèmes ont été réglés, mais d'autres problèmes qui ne dépendaient pas uniquement des domaines touchés par l'entrepôt restent encore à résoudre. Des comparaisons entre l'ancien et le nouveau système de production du PDG démontrent qu'il y aura un gain non négligeable dans les performances de production du rapport. Finalement, des exemples montrent les logiciels retenus pour le forage de données et *OLAP* en utilisant des données provenant de l'entrepôt. Dans le chapitre 6, on offre quelques suggestions pour l'amélioration de l'entrepôt et sur les prochains projets concernant le rapport PDG. Il y a des améliorations possibles aux logiciels et aux éléments de l'entrepôt qui sont déjà en place. Il y a aussi des optimisations à faire lorsque tous les éléments de l'entrepôt pourront être transférés sur une nouvelle version du serveur *Oracle*. Il y a des ajouts qui sont déjà prévus, mais qui, pour diverses raisons, n'ont pas été faits lors de la conception et du déploiement de l'entrepôt. Pour terminer, le chapitre 7 sert de conclusion pour ce mémoire.

CHAPITRE 2

CONCEPTS FONDAMENTAUX

DES ENTREPÔTS DE DONNÉES

Le système d'information développé au LaRePE est basé sur les connaissances techniques des personnes qui ont participé à sa conception et qui n'étaient pas forcément des informaticiens, ainsi qu'aux exigences qui n'ont cessé d'évoluer avec le temps. Lors de la création du système, des outils simples étaient privilégiés (Excel, Access, Visual Basic). Cependant, l'utilisation de ces outils a donné un système qui n'est pas aussi flexible ou extensible que ce qui est aujourd'hui nécessaire. Ce système dont nous avons hérité pourrait être reconstruit pour s'adapter parfaitement à de nouveaux besoins exprimés par les utilisateurs. Certains de ces nouveaux besoins sont, par exemple, la création de rapports sur le Web en peu de temps et la saisie de données par des utilisateurs éloignés (en Europe). Ces contraintes n'existaient pas lors de la création des premiers outils de saisie de questionnaires et de générateurs de rapports. Mais le temps et l'argent requis pour cette tâche sont importants. Cette problématique cadre bien dans la définition des *legacy systems*. La création d'un entrepôt de données est jugée comme étant une bonne alternative pour permettre de continuer à utiliser les anciens systèmes et pour supporter les besoins futurs des utilisateurs (McFadden, Hoffer et Prescott, 1999). Le système de gestion et de saisie des questionnaires qui a été développé il y a plusieurs années pourrait alors être utilisé pour alimenter en données l'entrepôt, et ce dernier servirait alors de support pour des utilisations qui ne sont pas possibles ou faciles à supporter avec l'ancien système, comme le forage de données.

2.1 Définition

Une définition des entrepôts de données a été proposée par Inmon et Hackarton en 1994 :

A data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data used in support of management decision making processes (Inmon et Hackarton, 1994).

Une traduction de cette définition pourrait être :

Les données d'un entrepôt de données sont : intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse (Cauvet et Rosenthal-Sabroux, 2001).

Il existe aussi d'autres définitions d'un entrepôt de données. Selon d'autres auteurs, un entrepôt de données peut être vu comme « un ensemble de vues matérialisées définies par des relations sur des sources de données distantes » (Theodoratos et Bouzeghoub, 2000). Cette définition semble être une simple explication d'une méthode pratique pour réaliser un entrepôt. Comme on va le voir, les vues matérialisées[4] ne permettent pas de résoudre tous les problèmes d'implémentation d'un entrepôt, même si elles peuvent faciliter le chargement des données. Cette définition ne tient pas compte de la nature historique d'un entrepôt, elle ne prévoit

[4] Les vues matérialisées calculent à l'avance des résultats de requêtes SQL dans une base de données et les conservent physiquement pour accélérer les traitements.

pas de méthode pour *historiser* les données qui proviennent des sources de données de l'entrepôt. Des tables supplémentaires sont nécessaires pour créer un historique, car une vue matérialisée effectue une copie des données et supprime la version précédente.

2.1.1 Explication de la définition

Selon la définition de Inmon, un entrepôt de données doit être organisé autour des sujets de l'entreprise (clients, étudiants, produits, etc.). L'entrepôt doit aussi être intégré, c'est-à-dire donner une définition constante de tous les termes et des données qu'il contient. Le vocabulaire utilisé dans l'entrepôt doit être le même, peu importe la personne qui l'utilise. Les données ont une période de validité dans le temps, il est possible de déterminer avec précision quand chaque enregistrement a été inséré dans l'entrepôt. Finalement, il faut que les données soient mises à jour. Il est recommandé de ne pas écraser les anciens enregistrements, ce qui permet de recréer un portrait de l'entreprise dans le temps. L'ensemble de l'entrepôt doit être conçu pour faciliter l'accès aux utilisateurs finaux avec des logiciels d'analyse de données. Ces logiciels sont généralement conçus pour permettre aux décideurs de prendre des décisions plus éclairées en leur donnant accès aux données rapidement et facilement, d'où le terme *business intelligence*.

2.1.2 Différences entre une base de données et un entrepôt

Plus concrètement, la notion même d'entrepôt de données est apparue lorsqu'il a été reconnu qu'il existe des différences profondes entre les systèmes transactionnels (OLTP) et les systèmes informationnels (Devlin et Murphy, 1988). Certaines de ces différences fondamentales sont listées dans le tableau 1. Les utilisateurs, les données, les structures, l'administration, la gestion des systèmes et le rythme quotidien sont tous différents. De nouvelles méthodes et techniques ont vu le jour pour permettre de concevoir et d'implémenter des entrepôts de données. L'une de ces techniques est le modèle dimensionnel (Kimball, 1996). Les éléments de conception d'un entrepôt sont décrits plus en détail dans la sous-section 2.2.

Tableau 1

Différences entre les systèmes opérationnels (transactionnels) et les systèmes informationnels (pour l'analyse, comme un entrepôt de données)

(McFadden, Hoffer et Prescott, 1999)

Caractéristiques	Systèmes opérationnels	Systèmes informationnels
Raison d'être	Supporter les opérations courantes de l'entreprise	Supporter le processus de décision des gestionnaires
Type de données	Représentation courante de l'état de l'entreprise	Historiques ou moment précis dans le temps
Principaux utilisateurs	Commis, vendeurs, administrateurs	Gestionnaires, analystes, clients
Envergure de l'utilisation	Étroit, simples mises à jour et requêtes	Large, requêtes complexes et analyse
But de la conception	Performance	Facilité d'accès et d'utilisation

Le stockage de l'évolution des données constitue une contribution majeure des entrepôts à la prise de décision par rapport aux bases de données de production qui stockent rarement les historiques (Cauvet et Rosenthal-Sabroux, 2001). Un exemple du stockage de l'évolution des données pourrait être de conserver l'information sur le fait qu'une bouteille de l'article X contenait 400 millilitres jusqu'à il y a deux ans, et que depuis les bouteilles contiennent 325 millilitres (même si à l'interne, cette bouteille est toujours considérée comme étant le même article X). Dans une base de données classique, cette donnée pourrait être écrasée et perdue, ce qui peut fausser les résultats d'analyse pour une campagne publicitaire par exemple. Un entrepôt de données n'est pas un simple ensemble de vues matérialisées. Une vue matérialisée permet d'éviter du temps de traitement aux applications clientes en calculant à l'avance certaines valeurs et jointures. Cependant, ces vues sont une représentation directe des relations dans les données sources, et ne prévoient pas implicitement de méthodes pour conserver l'évolution des données. Il faut souvent un mécanisme supplémentaire pour permettre aux vues matérialisées de stocker plusieurs versions des données. Un avantage des entrepôts de données est de pouvoir « naviguer dans le temps » en conservant toutes les valeurs qui sont passées dans les systèmes transactionnels. Ainsi, il est possible d'établir des tendances à long terme, ou d'évaluer l'efficacité de campagnes publicitaires, etc. L'aspect historique des entrepôts est expliqué plus en détail à la sous-section 2.2.8.

Conserver toutes les données historiques peut impliquer l'accès à de vastes quantités de données, mais certaines techniques permettent de réduire cet inconvénient. Par exemple, plusieurs niveaux de granularité peuvent être présents dans l'entrepôt. Les données récentes sont stockées avec un niveau de détail très fin, et les données plus anciennes peuvent être stockées avec un niveau de granularité plus grand (Cauvet et Rosenthal-Sabroux, 2001). On peut ainsi archiver les données plus anciennes et réduire le stress sur les systèmes de stockage des données. Il est cependant préférable de s'assurer de pouvoir retrouver un niveau de granularité plus fin, au besoin.

2.1.3 L'exploitation d'un entrepôt

Puisqu'un entrepôt de données est différent d'une base de données traditionnelle, des logiciels différents sont nécessaires pour l'exploiter. Des méthodes utilisées pour accéder aux informations contenues dans les données d'un entrepôt sont l'analyse statistique, le forage de données et *OLAP* (ou On-Line Analytical Process). L'analyse statistique est utilisée depuis de nombreuses années et elle est très bien documentée. Le forage de données consiste en plusieurs méthodes et outils de plus en plus populaires pour accéder à la vaste quantité de données conservées par les systèmes informatiques (voir la section 2.4). Les logiciels *OLAP* utilisent une structure de données basée sur le modèle dimensionnel. À partir d'une ou plusieurs tables de faits et de plusieurs tables représentant des dimensions, l'utilisateur est capable de combiner les données à différents niveaux d'agrégation pour trouver des informations. La combinaison d'un entrepôt et d'un logiciel *OLAP* permet alors de parcourir une très grande quantité de données beaucoup plus rapidement que ce qui était possible auparavant. De plus, selon les besoins des utilisateurs, il est possible de prévoir des calculs d'agrégation durant le chargement des données dans l'entrepôt, ce qui permet d'avoir des temps de réponse beaucoup plus intéressants avec les différents algorithmes utilisés. La présence de l'entrepôt rend donc possible l'utilisation de techniques comme *OLAP* et le forage de données, tout en supportant les méthodes traditionnelles d'analyse de données, comme l'analyse statistique.

2.2 Conception

Plusieurs éléments doivent être considérés quand on veut créer un entrepôt : l'infrastructure système, les métadonnées, la découverte des données, l'acquisition des données, la distribution des données et les logiciels d'analyse (O'Neil, 1997). Un autre élément à considérer est la structure que l'on veut utiliser pour conserver les données. Ces éléments peuvent prendre beaucoup de temps à mettre en œuvre, la conception d'un entrepôt n'est pas un exercice simple. Un bon entrepôt de données peut prendre plusieurs années et des millions de dollars à concevoir dans une grande entreprise, et nécessite la mise en place d'une bonne équipe de développement.

2.2.1 Infrastructure système

Tout système informatique repose sur une combinaison de ressources matérielles et logicielles. Pour un entrepôt de données, il faut en général prévoir des serveurs puissants avec une grande capacité de stockage de données. Plusieurs logiciels de base de données supportent maintenant les besoins des entrepôts de données, et il y a de nombreux fournisseurs de solutions spécialisées. Un autre aspect de l'infrastructure système est le transfert des données. Les données doivent pouvoir être acheminées en un temps acceptable entre les systèmes de production et l'entrepôt (acquisition des données), mais il faut aussi prévoir une bande passante suffisante pour les distribuer (distribution des données) et permettre un accès en temps réel aux utilisateurs. Tous ces points doivent être considérés, et une faiblesse dans un seul de ces aspects peut nuire grandement au projet d'entrepôt de données.

2.2.2 Les métadonnées

Les métadonnées représentent la forme la plus utile de documentation de l'entrepôt. Ce sont des éléments (sous forme de documents PDF en ligne, pages Web, aide contextuelle, etc.) qui servent à expliquer le fonctionnement des données et l'état de l'entrepôt. L'utilisation de métadonnées permet à la fois aux utilisateurs et à l'équipe d'entretien de l'entrepôt de se retrouver plus facilement. Les métadonnées existent sous deux formes : contrôle et utilisateur (O'Neil, 1997). Les métadonnées de contrôle sont utilisées pour voir au bon fonctionnement de l'entrepôt. Quelques exemples de ces données sont : une liste des données qui ont causé des problèmes durant le chargement, l'information sur la taille des tables, le contenu des tables et des vues, etc. Les métadonnées orientées utilisateur servent à guider l'utilisateur final dans l'entrepôt. C'est une documentation qui définit les termes clairement, sans ambiguïté, et qui décrit les données accessibles à l'utilisateur. Par exemple, ces métadonnées peuvent expliquer comment faire la somme des ventes en fonction des années dans le logiciel mis à la disposition des utilisateurs.

Le rôle des métadonnées dans un entrepôt ne doit pas être sous-estimé. Pour avoir un entrepôt facile à entretenir pour les programmeurs et compréhensible pour l'utilisateur final, il faut disposer d'outils simples et de structures bien définies. L'utilisateur doit avoir accès aux informations nécessaires pour se retrouver dans l'entrepôt et bien interpréter les données. C'est pourquoi les métadonnées doivent être rédigées dans des termes faciles à comprendre pour tous les types d'utilisateurs. Elles devraient lui permettre de répondre aux questions suivantes au sujet de l'entrepôt et des magasins de données (voir la sous-section 2.2.5) (McFadden, Hoffer et Prescott, 1999) :

1. Quels sont les sujets traités dans l'entrepôt et les magasins de données ?
2. Quels faits et dimensions sont inclus dans l'entrepôt et les magasins de données ? Quelle granularité ont les tables de faits?
3. Comment les données du magasin sont-elles dérivées des données de l'entrepôt ? Quelles sont les règles (transformations) appliquées ?
4. Comment les données de l'entrepôt sont-elles dérivées des données sources ? Quelles sont les règles (transformations) appliquées ?
5. Quels rapports et requêtes prédéfinies sont disponibles pour visualiser les données ?
6. Quels outils et techniques d'analyse sont disponibles ?
7. Qui est responsable de la qualité des données, et à qui doit-on faire les demandes de changement ?

Mais il est de plus en plus recommandé d'utiliser les métadonnées pour gérer le chargement des données, ce qui rend l'entrepôt beaucoup plus flexible et facile à modifier. Des outils de chargement et d'entretien d'entrepôts commencent à apparaître (Rifaieh, et Benharkat, 2002). Les métadonnées deviennent alors un moteur pour la création des requêtes dans l'entrepôt, à la fois pour l'entretien et pour l'utilisation des données.

2.2.3 La découverte des données

La phase de découverte des données est généralement celle qui prend le plus de temps à l'équipe de développement (O'Neil, 1997). De nombreux intervenants spécialistes sont nécessaires (par exemple les programmeurs qui ont conçu les systèmes qui serviront de source de données, des analystes, des personnes impliquées dans la gestion de l'entreprise, etc.). Il faut parcourir les différentes sources de données (bases de données dans l'entreprise) pour trouver les données d'intérêt qui seront chargées dans l'entrepôt.

En temps normal, les données de ces systèmes sont incomplètes et doivent subir un nettoyage et des transformations avant d'être utilisables par l'entrepôt. Il y a normalement des incohérences entre les différents systèmes, et pour intégrer correctement les données, il faut réussir à réunir (joindre) les différentes bases de données entre elles. Il peut arriver que des champs soient manquants (données manquantes), que d'autres aient été incorrectement saisies (par exemple, les noms des clients peuvent être saisis différemment ou avoir des erreurs entre les systèmes). C'est pourquoi ce travail est très long, une bonne partie des données peut être traitée automatiquement, mais une certaine quantité doit être manipulée à la main. Les manipulations nécessaires pour rendre l'ensemble des données cohérentes sont parmi les plus importantes pour s'assurer que les résultats obtenus à partir de l'entrepôt seront justes.

2.2.4 L'acquisition des données

Une fois les données identifiées, il reste à remplir l'entrepôt. Un processus d'extraction, de transformation et de chargement des données est alors préparé (processus *ETL, Extract Transform and Load*). L'étape d'extraction des données se fait généralement directement sur les systèmes de production, dans les bases de données en créant un fichier de données. Ce fichier sera par la suite téléchargé sur un deuxième système qui se chargera du reste des manipulations de données. Puisque cette étape se fait sur les systèmes de production, il arrive souvent qu'une fenêtre de temps très limitée soit disponible pour effectuer le travail d'extraction, par exemple durant la nuit de 2 à 3 heures du matin. Parfois, il est même impossible d'arrêter ces systèmes. Une méthode doit alors être trouvée pour s'assurer de l'intégrité des données extraites, tout en dérangeant le moins possibles les logiciels en utilisation sur les serveurs.

Les transformations sont utilisées pour nettoyer les données et pour créer les clés qui serviront dans l'entrepôt. Il peut arriver que des données concernant la même entité (personne, entreprise, etc.) soient présentes dans différents systèmes, mais qu'il n'y ait pas de façon de joindre ces données automatiquement. Par exemple, un des systèmes peut identifier des personnes avec leur numéro d'assurance sociale, et un autre avec un code arbitraire à partir de leur nom et de leur date de naissance. Afin de pouvoir joindre les données et de les insérer dans l'entrepôt, il est nécessaire de créer des clés supplémentaires pour chaque enregistrement (comme un nombre), ce qui permet de les identifier uniquement et de les joindre correctement. Ces clés substituts deviennent alors les clés utilisées pour les jointures dans l'entrepôt.

Il existe des logiciels qui sont spécialisés dans le processus *ETL* d'un entrepôt de données. Il est souvent souhaitable de faire appel à un utilitaire existant sur le marché pour éviter des frais de développement et de test. Certains auteurs disent que peu importe notre situation, elle n'est jamais assez unique pour justifier le développement à l'interne de la majorité des logiciels nécessaires dans un entrepôt de données, y compris les utilitaires *ETL* (O'Neil, 1997). Cependant, une recherche rapide nous montre qu'il y en a des centaines. Aussi, en général ils ne produisent pas un code optimal, ils sont souvent plus coûteux que le temps de développement qu'ils remplacent et leur surabondance rend difficile le processus de sélection (Scalzo, 2003).

2.2.5 La distribution des données

Une fois la phase de chargement des données terminée, il faut les distribuer dans tous les centres d'analyse. Selon les besoins des utilisateurs, il est possible de concevoir un entrepôt de données à l'aide d'un ou de plusieurs magasins de données (*data marts*). Il est possible de réaliser un entrepôt de données où tout est centralisé et où il n'y a pas de magasins de données. Cependant, il est parfois avantageux d'en utiliser, que ce soit pour améliorer le temps de réponse lors de l'exécution de certaines analyses, ou en distribuant physiquement les magasins pour réduire la distance que les données ont à parcourir pour se rendre à l'utilisateur, etc.

Il y a six sortes de magasins de données (O'Neil, 1997) :

- Satellites : récupère toutes ses données de l'entrepôt;
- Alimenteurs (*feeders*) : alimentent l'entrepôt;
- Partition : est un constituant d'un entrepôt virtuel partitionné;
- Mini-entrepôt : comme un entrepôt, sans aller jusqu'au bout d'un projet complet d'entrepôt dans une entreprise (peut être vu comme un projet-pilote);
- Indépendants : c'est un mini-entrepôt avec des programmes de chargement de données qui sont implémentés par des départements au sein d'une entreprise, sans avoir de lien avec un entrepôt de données central;
- Mixés : représente une architecture d'entrepôts de données où plusieurs sortes de magasins sont utilisées.

En fonction du type d'architecture retenue, la phase de distribution des données peut continuer même une fois que l'entrepôt principal est prêt. Tous les magasins doivent éventuellement être mis à jour à partir du contenu de l'entrepôt.

2.2.6 Les logiciels d'analyse

Une fois les données dans l'entrepôt, l'exploitation devient possible avec de nouvelles méthodes. Les requêtes ad hoc sont possibles, mais des outils plus spécialisés comme des logiciels *OLAP* et de forage de données (voir l'annexe B) donnent à l'analyste ou tout autre utilisateur beaucoup plus de puissance et de facilité

pour l'accès à toutes les ressources de l'entrepôt (données et métadonnées). Les logiciels *OLAP* existent sous plusieurs formes. Le modèle dimensionnel est une forme plus naturelle de modélisation des données pour un système d'information, et ces logiciels profitent de cette propriété. Le but de ces logiciels est de permettre à un utilisateur un accès simple aux données sous forme de fenêtres et de graphiques. Il y a des logiciels *MOLAP* (*multi-dimensional OLAP*) qui utilisent une base de données dimensionnelle (parfois propriétaire) et les logiciels *ROLAP* (*relational OLAP*) qui utilisent une base de données relationnelle (Oracle, Access, DB2, etc.). Il y a aussi des logiciels qui importent les données, peu importe leur format (par exemple, un fichier texte). Ces derniers sont qualifiés de *OLAP Client* parce qu'ils fonctionnent comme une application sur un ordinateur personnel. Ces formes de logiciels d'analyse possèdent des avantages et des inconvénients (Dinter, Sapia, Höfling et Blaschka, 1998), mais ceci dépasse les limites de ce travail.

2.2.7 Les modèles et les langages de modélisation

Selon le rôle que l'entrepôt est appelé à jouer dans l'entreprise, plusieurs modèles pour les données peuvent être proposés. Les modèles au cœur de la recherche sur les entrepôts de données sont : le modèle dimensionnel et des extensions du modèle entité-relation standard (Vassiliadis, Simistsis et Skiadopoulos, 2002). Le modèle choisi pour l'entrepôt peut être représenté par le langage UML (Unified Modeling Language).

Le modèle le plus souvent recommandé est le modèle dimensionnel, avec le schéma en étoile (O'Neil, 1997 et McFadden, Hoffer et Prescott, 1999). Ce modèle fonctionne avec une table de faits, c'est le centre du schéma. Chaque enregistrement dans la table de faits constitue un fait, c'est-à-dire l'unité de base. La granularité du schéma permet de déterminer ce qui sera un fait. Par exemple, pour une chaîne de détaillants, un fait pourrait être un article vendu. Un fait pourrait aussi être un ensemble d'articles regroupés par magasin. On pourrait faire le total des articles vendus pour 1 jour, ou pour 1 semaine, etc. Ainsi, plus la granularité est fine, plus on a d'enregistrements dans la table de faits. Ce modèle est recommandé à cause de sa faible complexité, sa facilité de compréhension pour l'utilisateur final et pour les liens directs avec les structures logiques des données (Vassiliadis, Simistsis et Skiadopoulos, 2002).

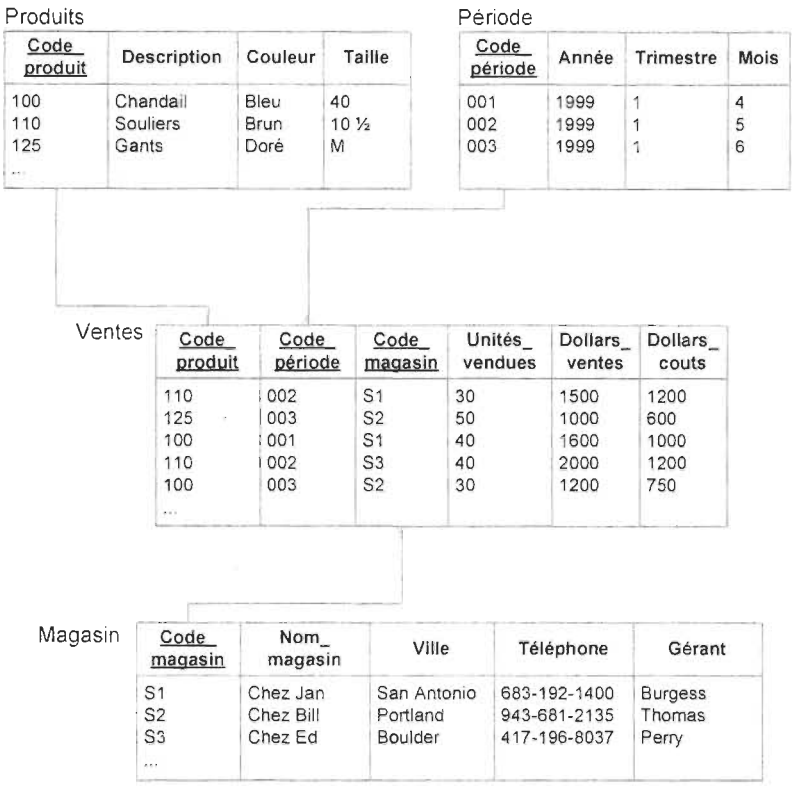


Figure 1 Le modèle en étoile (star schema) (McFadden, Hoffer et Prescott, 1999)

Dans la figure 1, on voit que la table de faits est Ventes, et que les tables de dimensions sont Produits, Période et Magasin. Ces dernières sont toutes liées par une clé à la table Ventes.

Il existe un autre modèle, le modèle dimensionnel utilisant un schéma en flocons (voir figure 2). Ce modèle est une sorte de compromis entre les modèles relationnels et dimensionnels. Le schéma en flocons est supposé diminuer la redondance du schéma en étoile en normalisant certaines des tables de dimensions, surtout lorsqu'elles contiennent beaucoup d'enregistrements. Cependant, l'auteur et concepteur d'entrepôts Ralph Kimball a émis cet avertissement très direct à propos des schémas en flocons : « Ne transformez pas vos dimensions en flocons, même quand elles sont grandes. Si vous le faites, préparez-vous à subir de mauvaises performances de navigation » (Kimball, 1996).

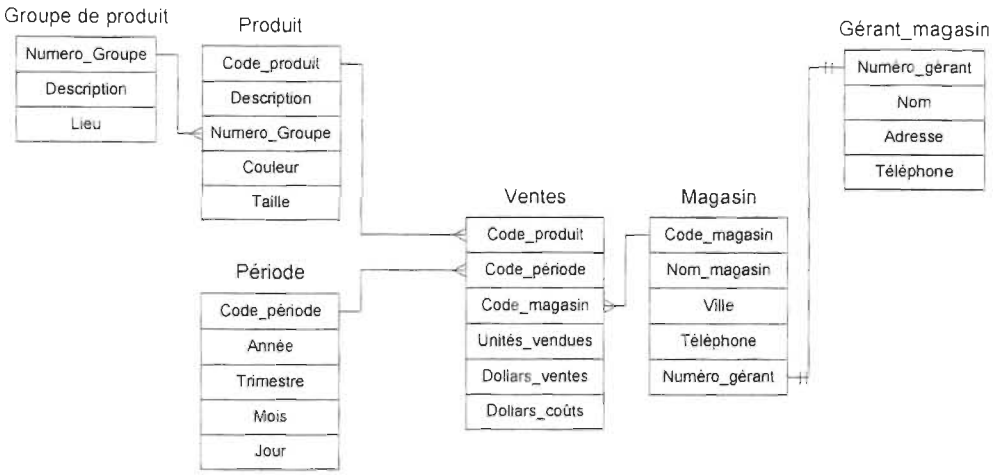


Figure 2 Le modèle en flocon (McFadden, Hoffer et Prescott, 1999)

Certains auteurs mentionnent que la structure relationnelle, avec le modèle entité-relation, reste la plus adaptée et permet une flexibilité accrue lors des changements (SCN, 2001). Le modèle relationnel est utilisé pour toutes sortes de bases de données. Ce modèle utilise une collection de tables pour représenter les données et les relations qu'elles exercent entre elles (Silberschatz, Korth et Sudarshan, 1999). Le langage UML est recommandé à cause de sa popularité et de sa fondation sémantique solide pour la schématisation des bases d'un entrepôt de données (Vassiliadis, Simistsis et Skiadopoulos, 2002). Il existe aussi d'autres façons de structurer les données pour un entrepôt (Golfarelli, Maio, et Rizzi, 1998 et Tsois, Karayannidis et Sellis, 2001), mais il n'y a pas encore de modèle conceptuel unique ni de langage reconnu universellement pour créer un entrepôt.

2.2.8 Les données historiques

Une notion importante dans un entrepôt de données est l'aspect historique des données. Le temps peut avoir plusieurs effets à la fois sur les données et les dimensions. Il est important de pouvoir retracer ces effets puisqu'un entrepôt devrait permettre de faire un portrait des données telles qu'elles étaient à un moment précis (Kimball, 1996). Dans un entrepôt basé sur le modèle dimensionnel, on a généralement une dimension temps qui permet d'associer les données dans les tables à un moment précis. Si on utilise une structure relationnelle, on a aussi un ou plusieurs champs dans les tables qui permettent d'établir une période de validité des données. L'utilisation de ces pointeurs temporels permet de recréer les données telles qu'elles étaient à une date précise. Il est aussi possible de traquer les changements.

2.2.8.1 Les dimensions à changement lent

Les dimensions sont aussi affectées par le temps. Il est généralement plus simple de créer un entrepôt en supposant que les dimensions sont indépendantes entre elles. Ainsi, les dimensions client, produit et temps sont considérées comme étant

indépendantes. Cependant, le statut d'un client ou d'un produit peut changer dans le temps. Par exemple, la cliente Marie peut être célibataire lors de son premier achat, mais se marier après 2 ans. Comment gérer ce changement ? L'auteur Ralph Kimball parle alors de dimensions à changement lent (*slowly changing dimensions*) (Kimball, 1996). Il définit trois façons de traiter ce genre de problème : écraser les valeurs (faire comme si Marie avait toujours été mariée), générer une nouvelle entrée dans la table de dimension (créer un nouveau client Marie), ou créer un champ supplémentaire dans la table de fait pour indiquer à la fois le statut original et statut courant. Ces trois méthodes sont utilisées et permettent de gérer le problème des dimensions à changement lent. Il y a tout de même certains problèmes que ces méthodes ne peuvent régler, et c'est pourquoi la recherche continue. Un article propose des opérateurs pour permettre la mise à jour des dimensions (Hurtado, Mendelzon, et Vaisman, 1999). Une solution qui est proposée au problème de pouvoir assurer un suivi continu dans le temps de toutes les versions des données est la table de faits à plusieurs versions (Body, Miquel, Bédard et Tchounikine, 2002). Dans cette méthode, l'utilisateur choisit le mode de présentation temporel et des facteurs de confiance sont associés aux liaisons entre les données selon les versions. Il y aura probablement de nombreuses autres solutions suggérées pour gérer les dimensions à changement lent.

2.2.8.2 L'expiration des données historiques

Une des raisons pour lesquelles les données historiques ne sont généralement pas conservées dans les bases de données est la grande quantité d'espace qu'elles occupent. Lors de la conception des entrepôts de données, il faut prévoir beaucoup d'espace de stockage en ligne pour cette raison. Mais même avec une très grande capacité de stockage, il peut arriver que la quantité de données soit tout simplement trop grande pour pouvoir tout conserver en ligne. Il y a alors deux solutions généralement utilisées pour régler ce problème : modifier la granularité des données conservées en fonction de leur âge, ou archiver les données les plus âgées (O'Neil, 1997).

La modification de la granularité des données peut se faire à l'aide d'un partitionnement des enregistrements. Plusieurs tables sont créées en fonction de la date ou de l'année courante : une table avec un grain au niveau de chaque transaction peut être créée pour le mois courant, une table avec un grain quotidien pour les 11 mois précédents (c'est-à-dire qu'on fait la somme de chaque groupe de faits pour chaque jour, et c'est cette somme qu'on conserve), et des tables pour chaque année précédente avec un grain hebdomadaire (où on fait la somme de chaque groupe de faits pour une semaine complète) (Cauvet et Rosenthal-Sabroux, 2001). Par exemple, si on était le 31 décembre, on aurait la liste des boîtes de céréales vendues du 1er au 31 décembre, avec le prix exact de chaque boîte. Mais si on voulait avoir la liste des boîtes de céréales vendues en novembre, on aurait le nombre de boîtes vendues à chaque jour, avec la somme des ventes pour chaque jour. Mais on n'aurait plus la liste exacte de chaque boîte de céréale vendue avec son prix, on pourrait seulement accéder au total quotidien.

Une autre façon de réduire la quantité de données stockées dans les tables est d'archiver les données les plus anciennes. Cependant, cette façon de faire va à l'encontre de la définition d'un entrepôt de données, c'est-à-dire permettre de faire un suivi historique de l'entreprise (un entrepôt est une collection de données historisée, voir la section 2.1). Une alternative à l'expiration des données historique est d'ajouter continuellement de l'espace de stockage pour répondre aux besoins. Mais comme ce n'est pas toujours possible, il faut généralement choisir l'une ou l'autre (ou une combinaison) des méthodes présentées pour éviter les dépassements de capacité des systèmes de stockage.

2.3 Application en entreprise

Les entreprises qui se dotent d'un entrepôt de données recherchent des méthodes pour utiliser toutes les données qui sont accumulées par leurs différents points de services (magasins, restaurants, etc.). Il y a de nombreux exemples de grandes chaînes de magasins intéressées par un entrepôt de données pour prendre des décisions d'affaires éclairées (McFadden, Hoffer et Prescott, 1999). Bien souvent, les systèmes qui contiennent les données sources de l'entrepôt sont bâtis autour de vieilles technologies hétérogènes et ne disposent pas de suffisamment de capacité pour permettre l'ajout d'un entrepôt. Aussi, il arrive que ces systèmes soient déjà surchargés et ne puissent pas être mis hors ligne pour permettre l'extraction régulière des données nécessaires à l'entrepôt. L'achat de matériel informatique est cher, et l'allocation de ressources (matérielles et humaines) dans l'entreprise doit être justifiée. De plus, un entrepôt de données est un système qui demande de la maintenance continue. Voici le contexte dans lequel un gestionnaire de projet chargé de la création d'un entrepôt de données se retrouve (O'Neil, 1997).

Malgré ces contraintes, il y a plusieurs exemples de succès d'entrepôts de données dans la littérature. Un de ces exemples est celui d'une grande chaîne de détaillants qui terminait le déploiement d'un entrepôt. Les analystes de cette chaîne ont commencé à chercher des tendances. Un problème est rapidement devenu apparent : la chaîne vendait des jeux vidéo pour une ancienne console, et dans tous ses magasins les jeux pour cette console ne se vendaient plus, sauf dans une région de Floride. Une enquête a rapidement révélé que cette région était habitée par de nombreux grands-parents qui avaient déjà acheté l'ancienne console pour divertir leurs petits-enfants. Ces personnes ne voyaient pas de raison de jeter l'ancienne console, mais ils voyaient l'intérêt d'acheter de nouveaux jeux. Cette découverte a mené à des décisions rapides et décisives dans la chaîne de détaillants, tous les jeux dans les magasins ont été envoyés là où ils se vendaient (O'Neil, 1997).
Un autre exemple est l'entrepôt de données de Statistiques Canada, bâti avec la technologie de Sybase (Sybase, 2003). Dans cet entrepôt, les données de l'organisme gouvernemental sont accessibles à l'aide de différents outils d'analyse, dont *OLAP* pour les analystes. Comme résultats, certaines requêtes qui prenaient plusieurs heures à compléter prennent maintenant 7 secondes, l'espace nécessaire pour le stockage des données a été réduit de 60 %, le coût de l'hébergement sur les serveurs a été réduit, et les temps de chargements des données durant la nuit ont été réduits à aussi peu que 5 minutes.

La décision de se doter d'un entrepôt peut coûter plusieurs millions de dollars à une entreprise, mais s'il est bien conçu et bien déployé, il permet de prendre des décisions plus éclairées plus rapidement et de donner aux décideurs un accès simple et rapide à leurs données.

2.4 La préparation des données

Le forage de données consiste en un ensemble de méthodes et d'outils de modélisation qui permettent de comprendre le comportement des données (voir l'annexe B). Un entrepôt de données est particulièrement adapté à cette classe d'outils. De nombreuses ressources existent pour se familiariser avec les concepts du forage de données, et certains outils sont même disponibles gratuitement pour des essais[5]. Il est important de distinguer le forage de données et la conception d'un entrepôt de données. En effet, l'entrepôt permet de regrouper les données qui seront éventuellement accessibles aux méthodes de forage, mais cet entrepôt ne supporte pas que le forage de données. Il peut aussi supporter d'autres utilisations, comme l'analyse statistique et des applications de production qui nécessitent des données analytiques (moyennes, médianes ...). D'ailleurs, la préparation des données pour un entrepôt doit permettre d'optimiser les utilisations pour améliorer les temps de calculs lors d'analyses. Par contre, la préparation des données pour le forage de données doit être faite en fonction d'un ou de plusieurs buts définis à l'avance. C'est pourquoi on peut dire que la préparation des données du forage de données est très différente de la préparation des données pour un entrepôt de données (Pyle, 1999). Malgré ces différences, les deux se complémentent très bien, et un entrepôt de données constitue une base solide pour supporter les techniques de forage de données.

[5] http://www.kdnuggets.com

CHAPITRE 3

ANALYSE ET CONCEPTION DE L'ENTREPÔT

Un entrepôt de données sert généralement à faire des traitements analytiques sur des bases de données qui n'ont pas été préparées spécifiquement à cet effet. Sans un système en place qui possède déjà des données, un entrepôt n'est pas très utile. Il est aussi préférable que les futurs utilisateurs de l'entrepôt connaissent leurs besoins aux fins d'analyse. Au LaRePE, un système est déjà en place et supporte de nombreux utilitaires qui peuvent bénéficier de l'ajout d'un entrepôt de données. Ces applications peuvent être classées en deux catégories, soit la recherche scientifique et la création de rapports de *benchmarking*. De plus, toute structure créée en lien avec ces utilitaires doit être facile d'entretien et flexible. Le système est appelé à évoluer constamment. Tout au long de la conception de l'entrepôt de données, il faut tenir compte de ces besoins parfois contradictoires.

Ce chapitre présente les détails liés à l'analyse des besoins des utilisateurs, en se basant sur le système déjà en place et sur la conception des divers éléments de l'entrepôt. L'analyse des besoins a fait l'objet d'un document préparé durant l'hiver 2003 (Dugré et Delisle, 2003) et qui avait pour objectif de faire le point sur les différentes applications utilisées au LaRePE, ainsi que d'apporter des solutions aux problèmes identifiés. Plusieurs de ces solutions sont mises en œuvre dans l'entrepôt de données, dont l'ajout du forage de données. De nombreuses améliorations essentielles sont présentées, mais certains changements moins spectaculaires et moins mesurables ne sont pas énumérés. Ces changements sont tout de même importants et font partie du travail visant à améliorer le système d'information du LaRePE.

3.1 Analyse du système

Des données sont accumulées dans la base de données manufacturière depuis plusieurs années déjà. Des applications sont en place, elles servent à alimenter et à utiliser cette base de données. Tout ce travail représente un investissement important en ressources humaines et matérielles. Ces utilitaires ont encore un rôle important à jouer dans les opérations quotidiennes du Laboratoire, mais ils ne sont pas bien adaptés à toutes les utilisations qui sont faites des données. La structure de la base de données et le couplage entre les différents logiciels rendent les nouveaux projets de plus en plus compliqués. De nouvelles sources de données sont régulièrement créées à l'intérieur de la base de données. De nouveaux projets sont aussi ajoutés et leur intégration n'a pas été prévue dans le système de base.

Il est nécessaire de faire un entrepôt parce que le système actuel n'est pas convivial à bien des égards :

- manque de flexibilité ;
- difficulté d'extraction des données pour les rapports et la recherche ;
- grave manque de documentation ;
- très grande dépendance envers quelques personnes clés pour des opérations routinières.

L'entrepôt vient donc combler plusieurs besoins essentiels :

- documenter les données qu'il contient ;
- rendre accessibles les données par une interface simple et conviviale ;
- permettre d'ajouter des sources de données au besoin ;
- capacité à conserver les jeux de données, les rapports, et même la liste des changements apportés aux données dans le temps ;
- identifier et documenter clairement les endroits où les transformations des données pour les statistiques doivent se faire ;
- faire une gestion centralisée de la sécurité pour les accès au système d'information, d'un bout à l'autre ;
- grande flexibilité sur tous les points de vue pour permettre l'évolution constante du système.

Différents éléments sont prévus pour permettre à l'entrepôt de supporter tous ces besoins. Il faut prévoir des algorithmes de chargement pour récupérer et transformer les données provenant des bases de données sources. Il faut aussi prévoir des interfaces pour faire la gestion des accès, pour chercher et modifier le dictionnaire de variables, et une application pour générer des jeux de données. Ces utilitaires doivent être facilement accessibles et bien documentés. Et il ne faut surtout pas négliger la sécurité, car ce système contient des données confidentielles.

3.1.1 Base de données manufacturière

La base de données manufacturière (aussi appelée base de données des questionnaires manufacturiers) a été pensée à la fois pour créer des rapports de *benchmarking* et pour servir de support à la recherche scientifique. Ces deux réalités ont causé des problèmes de structure à la fois dans la conception de la base de données, dans la façon de l'alimenter et dans la façon d'en extraire des données. Le processus de création a été incrémental. De nombreux intervenants ont participé à la conception de sa structure et aux façons de l'utiliser. De plus, avec le temps, de nouveaux logiciels se sont greffés et des modifications ont été apportées à tout le système. Cette évolution dans les besoins fait partie de la structure du Laboratoire, qui est basée sur les projets. Chaque projet apporte des éléments nouveaux qui doivent être intégrés au système.

Pour ces raisons, il n'y a pas de philosophie unique derrière la structure actuelle des données. Il y a de nombreuses tables, un ensemble de logiciels pour l'alimenter en données et de nombreuses applications qui doivent extraire ces données. L'entretien du système est un défi, à la fois à cause de sa complexité interne, mais aussi à cause des nombreux intervenants qui l'utilisent. En effet, c'est une riche source d'information, mais jusqu'à récemment, seul un utilisateur professionnel avait les connaissances suffisantes pour extraire correctement les données brutes. Cette personne devait intervenir dans toutes les activités d'exploitation des données. Puisque la documentation du système est défaillante, il est très difficile pour quiconque de s'aventurer dans les différentes structures en place et de comprendre

comment les utiliser. Il est encore plus compliqué et hasardeux pour quelqu'un de modifier le système. La formation de nouveau personnel pour utiliser ces ressources n'est pas simple.

Plusieurs chercheurs utilisent les données contenues dans la base de données, ainsi que des professionnels de recherche et même des étudiants à la maîtrise. Auparavant, ces utilisateurs sollicitaient tous la même personne pour leurs besoins en jeux de données. Mais comme les besoins étaient grands et qu'il fallait passablement de temps pour préparer les données, il pouvait se passer beaucoup de temps avant que tout le monde soit servi. Les intervenants qui utilisent les données proviennent de domaines divers et n'ont pas nécessairement la même compréhension des données. Ceux qui ont participé à l'élaboration du système ont fini par créer une ontologie commune. Mais plusieurs de ces chercheurs, professionnels et assistants ont quitté le Laboratoire. C'est pourquoi un travail de documentation et de support aux utilisateurs est nécessaire. Le travail effectué pour créer un entrepôt de données demande une bonne compréhension des structures existantes, ce qui permet de former de nouveaux individus sur l'utilisation des données et de documenter le système en cause. Cette expérience peut alors être consignée sous forme de métadonnées (documentation) et mise à la disposition des utilisateurs du système. C'est un travail qui n'a jamais été effectué de façon satisfaisante, et c'est pourquoi la documentation sur la base de données manufacturière et tous les sous-systèmes qui en dépendent est très difficile à trouver. La documentation est essentielle pour permettre aux utilisateurs de comprendre les données.

3.1.1.1 Identification des clés

La base de données manufacturière conserve les données sous la forme de tables comportant des éléments dits *généraux* et *financiers*. Ces deux natures des données sont à la base du système développé pour le rapport PDG, et la compréhension de cette structure est essentielle à tout système qui utilise par la suite les données saisies. Les clés des tables de la base de données identifient toujours une de ces deux natures, et elles peuvent parfois être combinées pour obtenir des données dites *mixtes*.

Les éléments provenant du questionnaire, dits *généraux*, sont des données représentées par un identificateur unique d'entreprise (COD_ESE) et une année de questionnaire (ANNEE). Par exemple, on peut avoir la clé « MA000587, 1999 ». Les éléments provenant des états financiers sont identifiés par l'identificateur d'entreprise (COD_ESE) et l'année financière (PERIODE). Le lien entre les données générales et financières n'est pas fixe. Par exemple, un questionnaire saisi en 1999 peut avoir des données financières allant de 1990 à 1999. Un nouveau questionnaire pour la même entreprise saisi en 2001 permet d'ajouter de nouvelles données financières. Mais les données financières de 1990 à 2001 ne sont pas nécessairement associées à un ou à l'autre des questionnaires, ce sont des données associées à l'entreprise.

Cette réalité permet de faire de nombreuses combinaisons au moment de créer des jeux de données, mais elle permet aussi de nombreux malentendus. Les pires cas surviennent lorsqu'il « manque » des années financières : par exemple, on a les données de 1990 à 1997, puis de 1999 à 2001. Il manque 1998. C'est le genre de problème qui peut survenir avec les données, et les utilisateurs doivent en être informés. Tous les systèmes qui utilisent la base de données manufacturière doivent aussi être conçus en considérant ces possibilités.

3.1.1.2 Structure de la base de données

Le LaRePE fonctionne à partir de projets (recherche, contrats, etc.). Pour cette raison, il arrive que des extensions soient faites à la base de données. Pour rendre la base de données plus flexible, il faut créer les tables générales et financières pour chaque projet (questionnaire complémentaire). Par exemple, pour le rapport PDG lui-même, on aurait les tables PDG_GENERAL et PDG_FINANCIER. Puis pour un projet Acier, on aurait ACIER_GENERAL et ACIER_FINANCIER, etc. Cette méthode permet de créer de nouveaux projets très rapidement en imposant une structure prédéterminée, documentée et codée à tous les projets possibles pouvant être associés à l'entrepôt. Pour qu'un projet puisse être *associé* à l'entrepôt, il est important de conserver les natures général/financier, et les mêmes clés (COD_ESE, ANNEE et COD_ESE, PERIODE), sinon les variables ne pourront que difficilement être comparées entre-elles. L'entrepôt ne vise donc pas à refaire la base de données manufacturière, mais il est conçu pour faciliter l'ajout de nouveaux projets.

3.1.2 Préparation des données

Avec l'ancien système, tous les calculs nécessaires à la préparation des données sont faits avec le logiciel *SAS*. Que l'utilisateur travaille avec *Excel*, *SAS*, *SPSS*, ou tout autre logiciel n'a pas d'importance. Tous doivent absolument utiliser le logiciel *SAS* avec les procédures d'exportation de données de *SAS*. Plusieurs des traitements et manipulations effectués avec *SAS* servent à restructurer les données de manière à ce qu'elles soient plus facilement utilisables par les chercheurs et assistants. Le terme utilisé à l'interne pour désigner ce travail de transformation est *recodification* des variables. Plusieurs variables calculées sont alors créées, il n'y a pas de démarcation claire entre le travail de nettoyage et le travail de création de nouvelles variables.

Avec l'entrepôt de données, c'est une toute nouvelle structure intermédiaire entre les données sources et l'utilisateur final qui apparaît. Le processus de transformation des données peut alors avoir lieu lors de la mise à jour de l'entrepôt, ce qui permet d'éliminer beaucoup de temps de traitement lorsque les utilisateurs ont besoin des données. Par exemple, la préparation des données avec SAS peut prendre jusqu'à 45 minutes au moment où l'utilisateur a besoin d'un jeu de données. Ce travail est effectué durant la nuit dans l'entrepôt. Et, comme il est mentionné dans le chapitre 5, le temps de création d'un jeu de données est maintenant réduit à quelques secondes pour une copie complète de la base de données manufacturière.

3.1.2.1 Calculs effectués dans l'entrepôt

Certains calculs peuvent être faits à l'avance dans l'entrepôt, pour accélérer le temps de réponse lorsque les utilisateurs demandent des données. De cette façon, les applications qui sont reliées à l'entrepôt ont moins de travail à faire pour rendre les données utilisables. Les calculs qui doivent préférablement avoir lieu dans l'entrepôt sont : le calcul du taux de change, l'alignement des données financières sur une même ligne, et le calcul du suivi des variables.

Le taux de change est nécessaire pour toutes les formes d'analyses puisque les dossiers contiennent des données financières de plusieurs pays. Pour l'instant, le taux de change est saisi manuellement et constitue une des sources de données de l'entrepôt. Le calcul est effectué durant le chargement de l'entrepôt (voir l'annexe C), ce qui évite de devoir le calculer chaque fois que les données de l'entrepôt sont utilisées. Il faut plus d'espace de stockage pour conserver les données, mais il est alors possible de réduire de manière substantielle le temps de traitement à l'utilisation. Il y a un nombre non négligeable de champs financiers dans l'entrepôt. Ainsi, les données sont accessibles immédiatement à l'utilisateur. Tout problème lié au taux de change peut être diagnostiqué lors du chargement plutôt qu'au moment où l'utilisateur accède aux données.

La structure des jeux de données qui sont utilisés par les chercheurs du Laboratoire a toujours été basée sur les questionnaires, chaque enregistrement représente un questionnaire. C'est pourquoi les données financières saisies sur plusieurs années doivent être alignées à la suite des variables du questionnaire correspondant. Cependant, les données financières recueillies ne sont pas liées à un questionnaire précis, mais plutôt à l'entreprise. Elles sont indexées en fonction de l'année financière de l'entreprise (le champ PERIODE). Lors de la préparation d'un jeu de données, les informations financières doivent alors être récupérées, puis les champs sont renommés afin de permettre la même variable d'être insérée dans le même enregistrement, tout en représentant plusieurs années. De cette façon, la variable VENTE pourrait devenir VENTE2000, VENTE1999, VENTE1998, etc. Cependant, la méthode utilisée pour renommer les variables fonctionne différemment. Elle utilise deux formes : l'année financière la plus récente pour une entreprise, ou une année de référence arbitraire. Les suffixes NOW, 1AN, 2AN, etc. sont utilisés pour représenter les variables en fonction de l'année financière la plus récente dans l'entrepôt pour chaque entreprise. Les suffixes R1, R2, R3, etc. sont utilisés pour représenter une variable en fonction d'une année de référence arbitraire (par exemple, 2000). On a ainsi le champ VENTENOW qui représente les ventes de l'année 2000 pour une entreprise, puis possiblement les ventes de l'année 2002 pour une autre (en fonction des données les plus récentes dans l'entrepôt). Puis le champ VENTER1 représente les ventes de l'année 2000 (ou toute autre année choisie par l'utilisateur) pour toutes les entreprises. Cette méthode est utilisée depuis longtemps pour l'analyse des données. L'avantage d'utiliser des suffixes prédéterminés plutôt que les années est qu'il est possible de supporter plus facilement certains logiciels qui utilisent les données, sans avoir à les modifier chaque année. On parle ici de rapports automatisés, ou d'utilitaires Web. Il y a aussi des avantages à cette forme de suffixe pour la création de programmes de statistiques pour la recherche.

Une autre forme particulière de calcul est le suivi des variables. Le calcul du suivi est apparu par la nécessité d'utiliser les données *connues* d'une entreprise avec des questionnaires qui ne contiennent pas toutes ces données. Lors de la première participation d'une entreprise au questionnaire manufacturier, un questionnaire complet lui est transmis. Si cette entreprise participe à nouveau à une année subséquente, un questionnaire simplifié, dit de *mise à jour* lui est acheminé. Ce questionnaire réduit le temps de réponse pour les entreprises qui sont déjà inscrites dans la base de données. Le problème provient de la façon dont les données sont utilisées pour créer les jeux de données : un enregistrement a toujours représenté un questionnaire. Puisque les questionnaires de mise à jour ne sont pas aussi complets, certaines informations essentielles (mais connues) sont manquantes. Ce problème occasionne de nombreuses manipulations lors de la préparation de projets de recherche, ces manipulations peuvent même représenter la moitié du temps nécessaire pour préparer un jeu de données à la main. Pour résoudre ce problème, une série de variables ont été identifiées comme étant très peu susceptibles de changer entre la réception d'un questionnaire de *base* et celle d'un questionnaire de *mise à jour*. Par exemple, on a le secteur d'activité et la région administrative. Ces variables sont alors utilisées en conjonction avec les données reçues dans le questionnaire de mise à jour. De nombreux calculs et des transpositions de données sont nécessaires pour permettre d'utiliser les données de cette façon. Anciennement, ce suivi était fait manuellement chaque fois qu'il était nécessaire. Mais maintenant, l'entrepôt dispose d'une liste des variables dont il est jugé correct de faire le suivi de cette façon. Avec cette liste, il est possible d'effectuer le suivi des variables automatiquement durant le chargement des données. Il est important de noter que les deux versions des questionnaires (avec et sans suivi) restent accessibles à l'utilisateur.

3.1.2.2 Calculs qui pourraient être effectués dans l'entrepôt

Certains calculs peuvent être faits dans l'entrepôt, mais ne le sont pas pour diverses raisons. Voici quelques-uns de ces calculs : le calcul de ratios, le calcul des classes et les critères de sélection. Le ratio est une formule (avec une division) qui utilise une ou plusieurs autres variables, et le résultat de cette formule est généralement stocké dans une table SAS ou une feuille Excel. Ces calculs ne sont pas encore disponibles dans l'entrepôt, mais ils sont suffisamment simples pour qu'il soit possible de les réaliser. Il pourrait même être avantageux, du point de vue du temps de préparation des données, que certains de ces ratios soient automatiquement calculés lors du chargement de l'entrepôt. Les classes sont une autre forme de variables calculées : il s'agit généralement de prendre une variable continue (comme l'âge d'une entreprise), et d'en faire des classes. Par exemple, on pourrait avoir 1 pour les entreprises de 0 à 1 an, 2 pour les entreprises de 1 à 3 ans, etc. Ce calcul est aussi relativement simple, et le supporter dans l'entrepôt pourrait simplifier les manipulations de données nécessaires pour un chercheur qui désire commencer un nouveau projet de recherche. Le dernier type de calcul est un peu plus complexe. Les critères de sélections sont des formules qui permettent de sélectionner des entreprises pour établir le groupe de comparaison (ou groupe témoin) lors de la création d'un

rapport de *benchmarking*, comme le PDG. Le calcul des critères de sélection nécessite l'utilisation de plusieurs enregistrements à la fois, ce qui peut devenir compliqué à cause de la nature *historisée* des données. Cependant, il pourrait plus tard s'avérer nécessaire de créer un magasin de données spécifiquement pour les rapports de *benchmarking*. C'est pourquoi ce genre de calcul pourrait devenir intéressant, surtout pour accélérer le temps de traitement des rapports qui devront être effectués sur le Web.

3.1.2.3 Calculs qui ne devraient pas être effectués dans l'entrepôt

Les calculs statistiques, et tout autre calcul qui utilise plus d'un enregistrement à la fois ne devraient pas avoir lieu dans l'entrepôt. Les calculs statistiques sont compliqués par la présence de données historiques. Il serait assez compliqué d'aller faire ce genre de calcul sans savoir pour quelle version de données ils doivent être appliqués. De plus, le fait de toujours exécuter ces calculs pour tous les cas possibles ne serait pas nécessairement utile pour la majorité des cas d'utilisation de l'entrepôt, et l'espace supplémentaire nécessaire pour conserver ces données serait important. Les calculs statistiques sont encore faits dans des logiciels spécialisés en statistique, ou dans des tableurs. Pour l'instant, un support est prévu pour SAS, SPSS et Excel. D'autres logiciels pourront s'ajouter à cette liste, au besoin. Si on parle de magasins de données pour des rapports, il serait possible de faire des préparations de données plus avancées. Mais même dans cette situation, il serait préférable de s'en tenir à un travail de préparation des données à l'intérieur du magasin, et de faire les calculs statistiques ou autres dans le logiciel de rapport.

3.1.3 Problèmes identifiés et solutions proposées

Le système d'information du LaRePE a été développé autour de besoins changeants et différents choix ont été faits pour permettre d'atteindre les objectifs. Ce système fait aussi l'objet d'ajouts et d'entretien depuis plusieurs années. Ces ajouts réguliers ont rendu les structures et le couplage entre les différentes composantes complexes. Malgré ces remarques, le système fonctionne et c'est pourquoi il n'est pas jugé avantageux de tout remplacer en même temps. Une approche incrémentale a plutôt été choisie, et l'entrepôt en est la première phase. L'entrepôt règle les problèmes urgents qui nuisent directement à l'exploitation des données recueillies.

Le modèle incrémental permet de développer une partie du système, pour répondre aux besoins identifiés, puis de revenir auprès des utilisateurs pour déterminer de nouveaux besoins. Ce modèle permet aussi d'éviter de créer un système parallèle qui devrait être entretenu en même temps que le système qu'il tente de remplacer. Cette situation a d'ailleurs déjà été vécue pendant plusieurs années au LaRePE, et c'est le genre de situation qu'il faut éviter. La conception d'un entrepôt demande beaucoup d'efforts. Si les ressources doivent être partagées entre deux systèmes qui font le même travail comme cela a déjà été le cas pour d'autres migrations, le nouveau système ne peut qu'en souffrir.

3.1.3.1 Absence de transparence pour les utilisateurs

Certains aspects des structures rendent l'utilisation statistique des données plus difficile. et l'entrepôt permet de régler plusieurs de ces problèmes de transparence. Certains de ces problèmes proviennent de limitations d'anciens logiciels utilisés, et d'autres sont apparus discrètement, au fil du temps. Les données sont distribuées dans un très grand nombre de tables, ce qui exige une jointure longue et complexe chaque fois que l'on désire récupérer des données. Il y a plus de 50 tables qui sont utilisées pour stocker les enregistrements dans la base de données manufacturière, ce qui exige un travail de recherche de variables chaque fois qu'une modification doit être apportée. Ces tables ne sont pas le résultat d'une normalisation, mais plutôt la solution à d'anciens problèmes. Une de ces raisons est l'utilisation d'Access dans le système. Chaque table Access ne peut contenir plus de 255 champs. Cependant, même avec cette limitation et surtout sans normalisation, 5 tables auraient amplement suffi à la tâche. Avec l'entrepôt, et à l'aide d'Oracle, tous les champs sont conservés dans une table de variables *générales* et une autre de variables *financières*, ce qui simplifie beaucoup la tâche de localisation et d'extraction des données.

Un autre problème présent dans la base de données est le champ PERIODE qui utilise le format « A_1999 » pour représenter l'année de l'état financier (plutôt que « 1999 »). Ce format n'est pas nécessaire. En fait il oblige des transformations supplémentaires lors du traitement statistique. Dans l'entrepôt, le champ est transformé et seule la valeur utile est conservée, c'est-à-dire les 4 chiffres de l'année, ce qui règle effectivement ce problème.

3.1.3.2 Grand nombre de variables dans le questionnaire

La base de données manufacturière utilise plus de 800 champs pour conserver les données des questionnaires. Cette structure demande beaucoup d'espace de stockage et une codification différente des questionnaires pourrait réduire le nombre de champs sans perdre le sens. En fait, au sens mathématique, plusieurs de ces *variables* n'en sont pas : il faudrait au moins se donner une définition claire et précise du concept de variable dans le contexte de l'entrepôt. Pour l'instant, la révision des variables est en cours et le processus pourrait encore durer plusieurs mois. Cependant, aucune forme de révision des codes des questionnaires n'est prévue pour le moment, c'est pourquoi le nombre de variables pourrait ne pas diminuer.

3.1.3.3 Temps de préparation des données très long

Le temps nécessaire pour prendre les données dans la base de données source et d'en faire un jeu de données utile pour les chercheurs prenait souvent plusieurs heures, si ce n'est quelques jours. Bien que tout le processus dépende de plusieurs étapes, dont la plupart sont humaines, il est possible de réduire les manipulations nécessaires pour obtenir un jeu de données de travail (un *data set* directement utilisable par les chercheurs).

Un des problèmes provient du manque de consensus sur *qui fait quoi*, lorsque l'on parle de la préparation des données en relation avec le questionnaire de base et celui de mise à jour. Certaines transformations sont faites directement dans la base de données (*Oracle*). D'autres sont faites dans les outils statistiques (*SAS*). Il faut déterminer à quels endroits les différentes préparations des données doivent être faites (par exemple, dans un entrepôt, ou dans l'outil statistique). La solution à ce problème réside dans la création du *Dataset Maker*, un nouvel outil accessible aux utilisateurs via le Web pour créer leurs propres jeux de données (voir la sous-section 3.2.3). De cette façon, toutes les transformations liées à la manipulation des données sont faites dans l'entrepôt, et l'utilisateur final n'a qu'à choisir la forme que son jeu de données doit prendre.

3.1.3.4 Nécessité de recréer des jeux de données selon la date des données

Lorsqu'un jeu de données est préparé à partir de la base de données à l'aide du logiciel *SAS*, l'état de la base de données devient fixe et seules les données présentes à ce moment sont accessibles. Ces données peuvent être explorées dans le cadre d'un projet de recherche pendant plusieurs semaines. Il peut arriver que des données supplémentaires (champs) soient exigées après plusieurs mois. Cependant, la base de données a évolué et une nouvelle extraction de données avec les mêmes champs ne donnerait pas nécessairement le même jeu de données. Des manipulations sont possibles pour permettre de créer un nouvel ensemble provenant de la concaténation de l'ancien jeu de données et du nouveau, mais ces manipulations prennent un certain temps et seul un utilisateur chevronné peut les accomplir. L'utilisation d'une base de données historique permet de régler tous ces problèmes de manipulation. Quelques requêtes générées automatiquement permettent de recréer les données telles qu'elles étaient à un moment précis.

Pour concevoir une telle base de données historique, une des premières étapes est de conserver une période de validité pour chaque enregistrement. Deux champs sont utilisés, un pour le début et un pour la fin de la période de validité. Pour un enregistrement qui est toujours valide au moment présent, on met une valeur très grande dans le champ de fin de validité, par exemple le 1er janvier 3000. Ceci permet d'éviter de traiter les valeurs vides, valeurs qui sont toujours une source d'erreur dans les requêtes. Le travail est alors beaucoup plus simple.

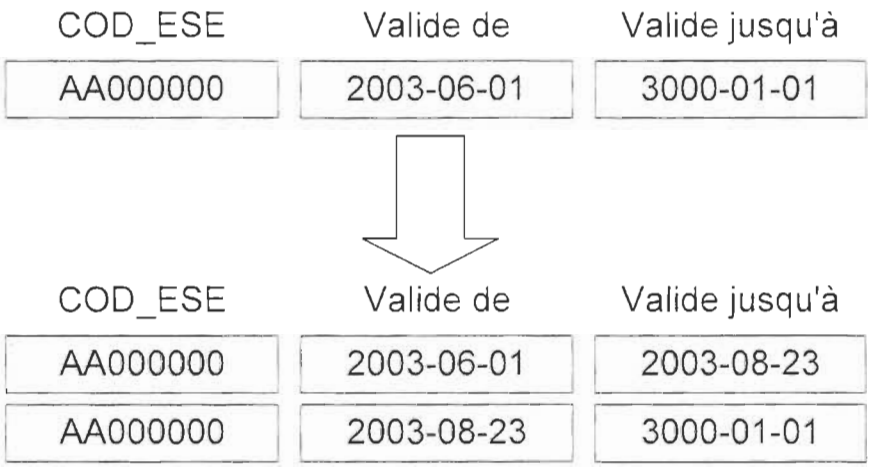


Figure 3 Ajout de champs pour calculer la période de validité

Comme le montre la figure 3, quelques champs ajoutés aux enregistrements permettent de calculer la période de validité. Ces champs sont utilisés lors de l'extraction des données pour déterminer si les enregistrements sont valides pour la période demandée par l'utilisateur. La date « 3000-01-01 » est utilisée pour identifier les enregistrements qui sont encore valides au moment où l'entrepôt est interrogé.

3.2 Conception de l'entrepôt

L'existence de plusieurs systèmes de données au Laboratoire complique l'accès aux informations requises pour la recherche. Le travail de jointure des données est à recommencer pour chaque nouveau projet. Avec beaucoup de travail, il est possible de faire fonctionner les projets du Laboratoire et d'en créer de nouveaux. Mais maintenant, ces projets vont pouvoir profiter pleinement du processus de chargement de l'entrepôt (processus *ETL*). La présence d'un point de vue uniforme sur les données réduit le temps nécessaire pour leur manipulation. Les possibilités d'erreurs provenant de ces manipulations sont aussi réduites puisque les calculs de préparation des données sont faits lors de la préparation de l'entrepôt. Le chargement fait alors l'objet d'une attention particulière. Un autre problème, qui revient à chaque nouveau projet, est l'absence d'une source unifiée de documentation sur les données (métadonnées). La création d'un entrepôt nécessite la création d'une structure pour conserver et rendre accessibles les métadonnées pertinentes. Il est donc nécessaire d'effectuer un travail de consolidation des informations sur les données en une forme utilisable par les utilisateurs. La structure des tables et des *schémas utilisateur* (comptes) employés dans la conception de l'entrepôt est illustrée plus en détail dans l'annexe D.

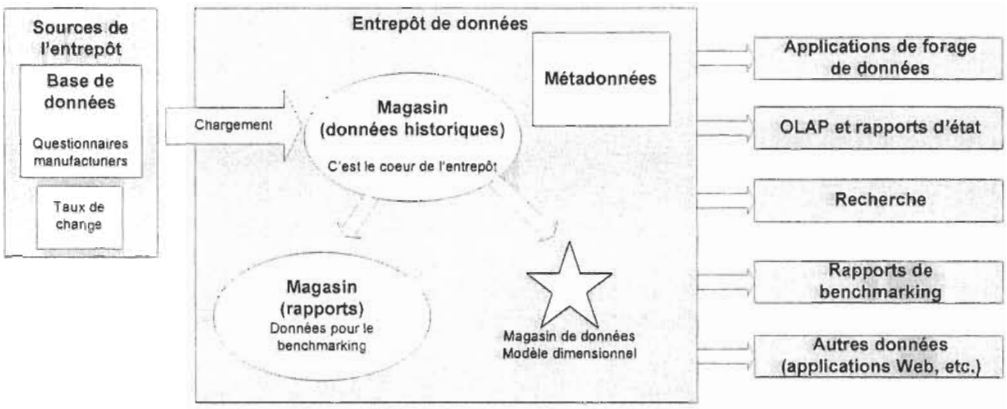


Figure 4 Schéma de l'entrepôt de données du LaRePE

La figure 4 illustre la structure générale de l'entrepôt de données, ainsi que quelques utilisations possibles. Auparavant, la création de jeux de données (*data sets*) était toujours faite manuellement, ce qui demandait toujours beaucoup de temps et de manipulations répétitives. De même, les utilitaires de création de rapports devaient toujours refaire de nombreux calculs assez exigeants du point de vue des ressources système, ce qui prenait beaucoup de temps. Avec l'entrepôt, la majorité des manipulations nécessaires à la préparation des données sont déjà faites à l'avance lors du chargement, puis les données sont stockées et documentées. C'est de cette façon que l'entrepôt de données du LaRePE améliore concrètement l'accès aux données. En plus, il rend l'utilisation de ces données plus facile pour les chercheurs et les étudiants en rendant accessible une documentation complète sur les variables. Le magasin pour les rapports illustré dans la figure 4 n'a pas encore été créé, mais il est prévu de l'ajouter à l'entrepôt pour supporter les futurs rapports Web. Les autres utilisations des données sont expliquées plus en détail au chapitre 4.

3.2.1 Création du magasin historique pour consolider les données

Le système de cueillette des données a été fait pour faciliter la création du rapport dans sa forme et son contenu actuels. La structure de la base de données n'est pas idéale pour l'utilisation de plusieurs algorithmes et techniques de forage de données, d'intelligence artificielle, etc. Une restructuration des données s'est avérée nécessaire. Il a fallu concevoir des étapes d'extraction, de transformation et de chargement (*ETL*) à partir de la base de données des questionnaires manufacturiers pour créer une structure intermédiaire. Cette structure est beaucoup plus apte à répondre aux nouveaux besoins exprimés par le Laboratoire. De plus, cette méthode permet d'ajouter de nombreuses sources d'informations (métadonnées, sources provenant d'autres projets du Laboratoire, bases de données achetées, etc.) et de les combiner à un même endroit. Cette nouvelle structure constitue notre entrepôt de données.

Le magasin historique est le premier élément de l'entrepôt à recevoir des données. C'est ce magasin qui permet d'alimenter tous les autres magasins, et c'est aussi lui qui conserve les données historiques qui permettent de recréer une image des données sources à une date précise. La phase de chargement effectue un nettoyage et certaines transformations sur les données afin de les stocker dans ce magasin. Il faut préparer les données historiques pour identifier correctement les enregistrements dans le temps et consolider toutes les sources de données pour accéder à toute l'information au même endroit. Il faut aussi transformer les enregistrements pour réduire les manipulations subséquentes lors de l'utilisation des données. Et il faut s'assurer que les données présentes dans l'entrepôt soient correctes.

3.2.1.1 Création de l'information historique

La première étape à accomplir est la jointure de toutes les données et la création de la période de validité à l'aide d'une phase *ETL*. Les données sont stockées dans un magasin avec l'ajout d'informations historiques. La base de données permettait déjà une certaine forme de traitement longitudinal des données. Cependant, rien ne permettait de recréer les données telles qu'elles étaient à un moment précis de son existence. C'est pourquoi l'information permettant de créer une période de validité a été ajoutée (voir la sous-section 3.1.3.4).

3.2.1.2 Nettoyage des données

Comme dans toutes les activités de création d'entrepôts de données, les données sources doivent être nettoyées. Cet exercice permet entre autres de combler plusieurs lacunes en documentation et d'explorer les bases de données sources pour détecter les incohérences. Pour permettre la création d'une structure uniforme pour l'entrepôt, la méthode de stockage des données a été révisée pour donner une séparation claire entre les variables générales et financières, et des améliorations ont été apportées pour avoir un accès plus facile aux données. Une des modifications apportées est la réduction du nombre de tables utilisées pour conserver les questionnaires, qui passent de 50 dans la base de données manufacturière à 3 dans l'entrepôt. Le magasin historique a été créé pour permettre d'extraire, de transformer et de charger les données dans l'entrepôt. Ce magasin sert d'intermédiaire entre la base de données des questionnaires et outils qui utilisent les nouvelles structures de données présentes dans l'entrepôt. Pour l'instant, le magasin contient les questionnaires manufacturiers. Mais il a été conçu pour permettre la consolidation de plusieurs nouvelles sources de données provenant d'autres projets du LaRePE. Il serait donc possible de rapidement intégrer des données provenant de plusieurs projets différents. Le processus de consolidation n'est pas *magique* ni *automatique* dans sa configuration, mais l'ajout de nouvelles données est prévu dans l'entrepôt et le magasin historique est l'endroit privilégié pour le faire.

3.2.1.3 Les transformations

Le magasin de données historiques devrait être vu comme un remplacement des procédures de *recodification* et d'intégration des données qui étaient préalablement faites avec *SAS* (Dugré et Delisle, 2003). Ces transformations de données ne requièrent pas nécessairement un logiciel statistique, les transformations ne sont pas des calculs statistiques en soi. De plus, le logiciel statistique n'est pas nécessairement conçu pour gérer un grand volume de données parfois hétérogènes, alors qu'un gestionnaire de bases de données (comme *Oracle*) est beaucoup plus performant pour cette tâche.

Plusieurs transformations sont effectuées lors du chargement, et les résultats sont stockés dans les différentes tables du magasin. Les transformations sont, par exemple :

- le calcul du taux de change (voir la sous-section 3.1.2.1) ;
- le suivi des questionnaires (voir la sous-section 3.1.2.1) ;
- la vérification des données pour s'assurer de leur intégrité.

3.2.1.4 Assurance-qualité

Toutes les transformations et les calculs sont effectués durant le processus quotidien de chargement de l'entrepôt. Si une erreur survient, le chargement est annulé et un message est envoyé aux personnes responsables de l'administration de l'entrepôt. Les données disponibles dans l'entrepôt sont toujours celles provenant du plus récent chargement réussi.

La base de données manufacturière est régulièrement mise à jour, avec l'arrivée et la saisie de chaque nouveau questionnaire. Un processus de validation des données est en place lors de la réception des questionnaires. Chacun est vérifié, corrigé, puis saisi par un assistant. Il est ensuite vérifié par une autre personne afin de s'assurer que les données sont conformes à la fois dans la base de données et sur le questionnaire imprimé. Pour cette raison, chaque questionnaire est mis de côté par le processus de chargement tant que toutes ces étapes de vérification ne sont pas complétées. Cette méthode permet d'éviter aux utilisateurs de l'entrepôt d'avoir à se demander si les données utilisées sont vraiment fiables et vérifiées, c'est l'entrepôt qui s'assure de ne donner accès qu'aux données correctes.

L'assurance-qualité est un processus essentiel dans un entrepôt de données. Les informations sur les chargements sont conservées, et toutes les données du magasin de chargement sont *historisées*. Ainsi, même si les bases de données à la source du magasin ne conservent pas toutes les versions de leurs données (ce qui est rarement le cas dans les bases de données traditionnelles), on peut avoir accès à toutes les versions des données qui ont été conservées par le magasin. Grâce à ces informations, en cas de problème avec le chargement, il est possible de retirer les données fautives et tous les utilisateurs accèdent alors à la dernière version correcte des données de façon transparente, pendant que des correctifs sont apportés.

3.2.2 Les métadonnées de l'entrepôt

Le dictionnaire de variable permet de documenter le contenu de l'entrepôt. Toutes les variables utilisées doivent se trouver dans le dictionnaire. Ce dictionnaire n'a pas seulement un rôle de documentation, il participe activement au fonctionnement de l'entrepôt en permettant d'ajouter des champs à importer lorsque ces champs sont créés dans les bases de données sources. Il permet aussi de créer de nouvelles variables en effectuant des calculs sur les variables existantes (qui proviennent de bases de données, ou qui sont elles aussi des variables calculées dans l'entrepôt).

Lorsqu'on ajoute des variables, le dictionnaire permet de conserver les éléments de documentation suivants (voir l'annexe E et la figure 5) :

- le nom de la variable ;
- le format et l'information sur les codes ;
- des mots clés ;
- des descriptions (sommaires et plus complètes) ;
- la nature de la variable (général, financier, mixte) ;
- la formule (si c'est une variable calculée) ;
- les usages ou applications de la variable, y compris le nom de la personne en ayant demandé la création et à quelles fins ;
- les validations et transformations à faire lors du chargement de la variable à partir des sources de données ;
- si c'est une variable monétaire, quel calcul de devise doit être appliqué ;
- commentaires des utilisateurs.

Le dictionnaire permet évidemment de faire une recherche à partir de différents critères pour trouver les variables (figure 6). En fait, ce dictionnaire pourrait éventuellement devenir une véritable base de connaissances pour des logiciels intelligents, ce qui fait d'ailleurs partie des travaux futurs qui sont déjà envisagés.

Détails de la variable ACHACOAC

Ajouter au panier | Modifier | Retour | Imprimer

INFORMATIONS GÉNÉRALES

Code : ACHACOAC	Nature : Général	Créée par (date) : N/A (2003-05-13)
Label statistique: CONTRÔLE COÛTS - ACHATS		

INFORMATIONS SUPPLÉMENTAIRES (VARIABLE DU QUESTIONNAIRE)

Nom table BD: MAN_ESE_CONT_COUT	Format dans BD: NUMBER(3)
Est-ce une variable monétaire? : Non	Recodage dans SAS:
La variable peut-elle suivre? : Non applicable	

Liste des questionnaires où se retrouve la variable:

Questionnaire	Version	Section	Page	Question
Base	4.1	Production	18	14
Mise à jour	1.1	Production	11	14
Base	5	Production	19	14

Figure 5 Détail de la variable ACHACOAC

Dictionnaire de variables

Créer une nouvelle variable | Gestion des formats

LISTE ET PANIER DE VARIABLES

VARIABLES (1458)

ACAPP
ACCIAMES
ACCIIMOB
ACCIOBJE
ACCIONSP
ACFIB
ACHACNSP
ACHACOAC
ACHCL
ACHCO
ACHCP
ACHDO
ACHFO
ACHME
ACHOP

Afficher

PANIER DE VARIABLES (9)

MAJ panier

ACAPP
ACCIIMOB
ACCIOBJE
ACCIONSP
ACFIB
ACOUT
ACTOT
ANNEE
COD_ESE

Visualiser | Supprimer | Vider

RECHERCHE DE VARIABLES

Nom:		Mots clés:
Logiciel: Tous	Dataset:	Type: Tous les types
Questionnaire: Tous	Version	Page	Question	Recherche	Réinitialiser

Figure 6 Recherche de variables dans le dictionnaire de variables

Un premier inventaire des variables utilisées a été effectué, et devra être complété à l'aide des utilisateurs du système. Ces informations sont accessibles dans le dictionnaire et la documentation de l'entrepôt. Cependant, pour être efficace, il va falloir porter une attention particulière à l'entretien de ce dictionnaire. Si un outil statistique est utilisé à l'extérieur du contexte de l'entrepôt pour créer des variables, le dictionnaire ne peut pas être mis à jour automatiquement. Il permet de documenter ces variables, mais il faut assigner une personne qui va faire des ajouts pour permettre de retrouver les variables qui sont créées dans le cadre des rapports ou de la recherche. Il faudrait même prévoir une méthode pour récupérer les jeux de données qui sont ainsi créés et les conserver. Mais ceci peut faire l'objet d'un autre travail. Pour le moment, l'important est d'avoir un dictionnaire et de déterminer quelqu'un qui est responsable de son entretien.

D'autres formes de métadonnées sont aussi incluses dans l'entrepôt. Il y a l'emplacement dans les questionnaires, ce qui est très utile pour associer les variables aux questions posées aux entrepreneurs. Il y a aussi des pages d'aide en ligne, et une description des données qui sont importées à partir de la base de données manufacturière. Les commentaires des utilisateurs permettront d'ailleurs de compléter cette partie des métadonnées.

3.2.3 Création du *Dataset Maker*

Le fonctionnement du *Dataset Maker*[6] est basé sur la nécessité de donner aux utilisateurs autorisés[7] un moyen simple et rapide pour accéder à des données à jour. Auparavant, le processus pour accéder aux données des questionnaires était long et plusieurs manipulations devaient être faites à la main. Le *Dataset Maker* vient donc automatiser l'ensemble des opérations pour permettre aux utilisateurs autorisés de créer eux-mêmes un jeu de données à partir de l'application Web conçue pour accéder à l'entrepôt. Cette application permet de faire une recherche des variables à partir du dictionnaire et de sélectionner celles qui doivent entrer dans les jeux de données (voir la figure 7). Les jeux de données peuvent être générés selon plusieurs formats (XML, SAS, SPSS, Excel, etc.) et sont conservés dans l'entrepôt pour consultation ultérieure.

---

[6] Le *Dataset Maker* a été réalisé par Jean-François Beaudoin et Mathieu Dugré durant l'été 2003

[7] Pour des raisons de confidentialité et de sécurité, seuls certains utilisateurs ont accès au Dataset Maker pour créer des jeux de données.

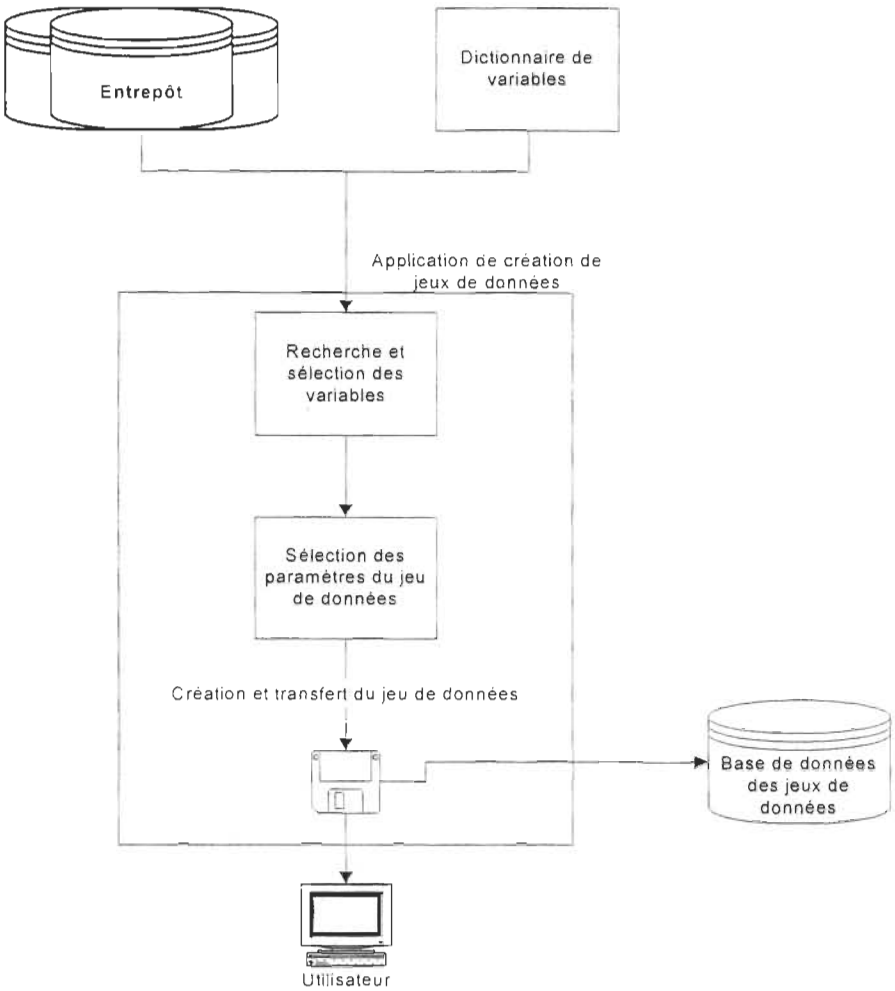


Figure 7 Schéma de création d'un jeu de données

Le processus de création du jeu de données dispose de certains paramètres, dont le nom des variables, le nombre d'années désirées pour les données financières, les types de fichiers à créer, etc. Les jeux de données sont téléchargés sécuritairement sous forme d'un fichier *ZIP*. Un jeu de données peut contenir plusieurs fichiers. Cette archive est aussi conservée directement dans l'entrepôt sous forme de BLOB pour un accès ultérieur rapide. Le *Dataset Maker* comporte beaucoup d'autres fonctionnalités. Pour une liste plus complète, voir l'annexe F.

Le *Dataset Maker* guide l'utilisateur à travers 4 étapes qui lui permettent de choisir les variables à insérer dans son jeu de données, ainsi que la façon de les préparer et le format de fichier à utiliser. Les étapes sont : le choix des variables, le choix d'options et de groupement des variables, la création du jeu de données et la confirmation pour téléchargement. Il est possible de revenir à une étape précédente à tout moment pour modifier les options choisies. Les jeux de données sont conservés dans l'entrepôt et peuvent être consultés à tout moment. Il est aussi possible de les réutiliser afin de créer un nouveau jeu de données avec les mêmes paramètres, mais

avec des données à jour. Cet outil permet d'accélérer grandement le temps de création d'un jeu de données, rendant ainsi beaucoup plus accessible le contenu de l'entrepôt pour les chercheurs et les étudiants qui sont impliqués au LaRePE.

Les étapes de création d'un jeu de données vont maintenant être présentées plus en détail. La première étape (illustrée par les figures 8 et 9) permet de choisir les variables à l'aide du dictionnaire de variables, puis de confirmer le choix de ces variables.

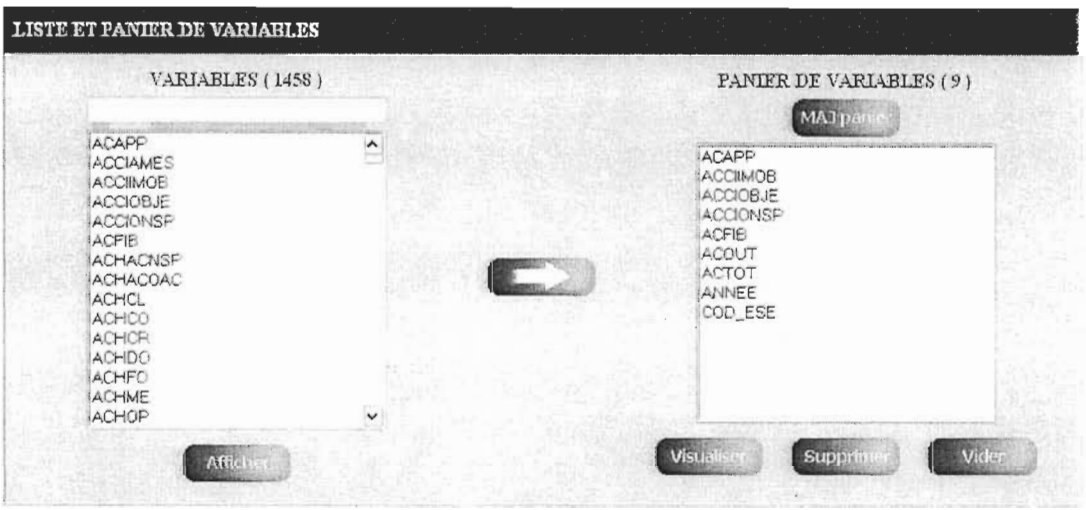


Figure 8 À partir du dictionnaire de variables, on se fabrique un panier qui est ensuite utilisé à l'étape 1 du Dataset Maker

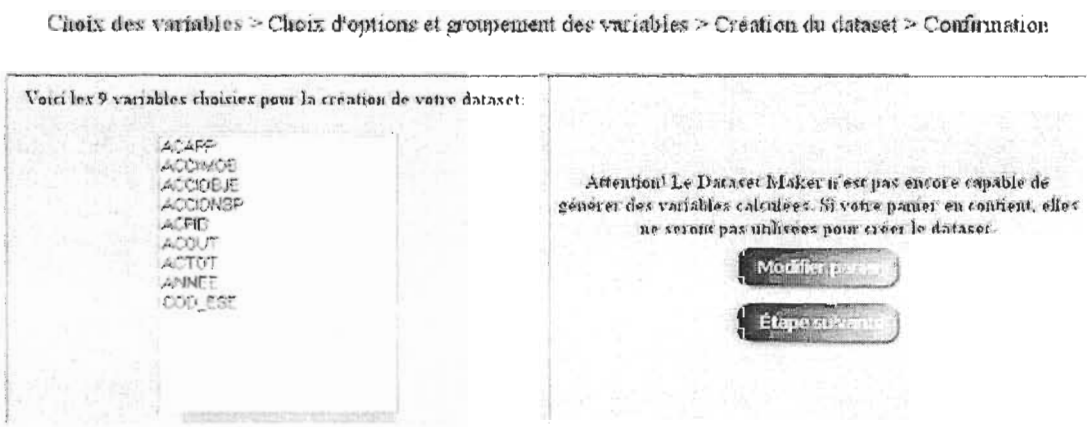


Figure 9 Étape 1 du Dataset Maker, confirmer le choix des variables

À la deuxième étape (voir figure 10), il faut déterminer si on veut utiliser les données du magasin historique avec ou sans le suivi (voir la sous-section 3.1.2.1). Il est aussi possible d'utiliser uniquement les questionnaires les plus complets (ceux de première année) et de ne pas utiliser les questionnaires de mise à jour. Ces deux options nécessitaient auparavant de nombreuses manipulations à la main dans les logiciels statistiques. Les manipulations pouvaient prendre quelques jours, mais maintenant

elles sont disponibles et calculées automatiquement selon les besoins de l'utilisateur. Les variables financières doivent subir une transformation pour être alignées avec les données des questionnaires. Ces transformations peuvent être sélectionnées très facilement par l'utilisateur, il lui suffit de choisir les variables puis d'appuyer sur les flèches à l'écran qui lui permettent de les transférer dans une des cases. La case « R1, R2, R3... » permet de placer les variables financières sur une seule ligne en utilisant l'année de référence choisie (dans l'exemple, 2000) pour le nombre de périodes financières désirées. L'autre case, celle avec « NOW, 1AN, 2AN... » permet de créer des variables sur une seule ligne en partant de la plus récente année financière de chaque entreprise. Ces deux transformations de variables financières étaient très longues à exécuter avec SAS. mais avec la structure interne de l'entrepôt de données, quelques secondes suffisent maintenant à préparer un jeu de données, même très grand.

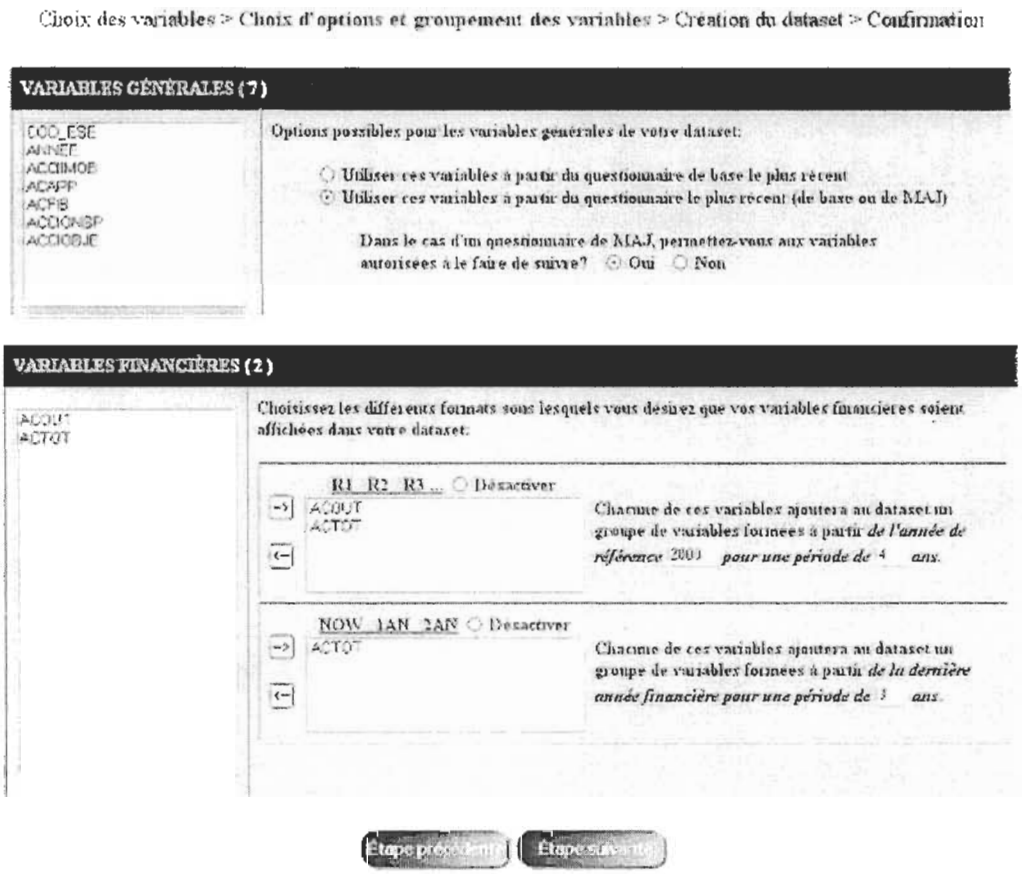


Figure 10 Étape 2 du Dataset Maker, déterminer les paramètres pour accéder aux données dans l'entrepôt en fonction des variables

La troisième étape (figure 11) permet de choisir de nombreuses options qui servent à créer le fichier du jeu de données et à le stocker dans la base de données. Certaines options ont aussi un rôle à jouer dans la sélection des données qui entrent dans la composition du jeu de données. Il faut choisir quel type de fichier sera créé parmi les choix suivants : un fichier de données SAS, un fichier XML qui peut être utilisé par SAS, un fichier Excel et un fichier XML pouvant être affiché dans un navigateur. L'application exige un nom pour le jeu de données. Ce nom est utilisé pour créer les fichiers de données et aussi pour stocker une référence vers les fichiers dans la base de données. Le champ *utilité du jeu de données* est une sorte de description qui sert à décrire le jeu de données dans la liste du *Dataset Maker*. Les autres informations sont des critères de sélection qui déterminent quelles données sont utilisées pour chaque variable. Puisque le *Dataset Maker* fonctionne à partir du magasin historique, il faut toujours choisir une date pour générer les données. Normalement, c'est la date courante qui est utilisée. Il est aussi très important de spécifier la devise dans laquelle les données monétaires doivent être exportées puisque l'entrepôt en contient plusieurs. Il est encore possible de modifier l'année de référence sélectionnée à l'étape 2. Finalement, un champ permet d'ajouter plusieurs critères de sélection à la main, ces critères utilisent la même syntaxe qu'une clause *WHERE* dans une requête *SQL*. Il est possible d'utiliser toutes les variables générales de l'entrepôt, y compris celles qui n'ont pas été sélectionnées pour le jeu de données. Les variables financières ne peuvent pas encore être utilisées, mais elles pourront être supportées dans le futur au besoin.

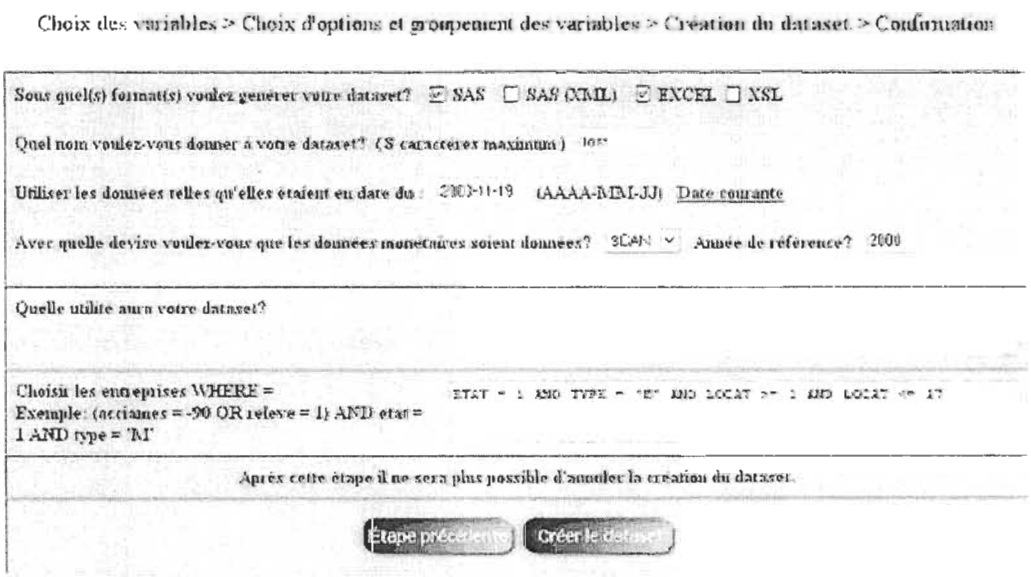


Figure 11 Étape 3 du Dataset Maker, sélectionner les options de création fichier de données à télécharger

La dernière étape de création d'un jeu de données (figure 12) consiste essentiellement en une confirmation de création du fichier, puis au téléchargement de ce dernier. Si une erreur survient, un message est affiché et des correctifs peuvent alors être apportés par l'utilisateur.

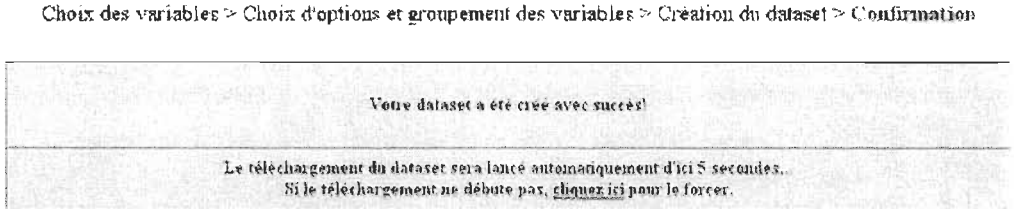


Figure 12 Etape 4 du Dataset Maker, l'application confirme que le jeu de données est prêt et démarre son téléchargement

3.2.4 Chargement des données dans l'entrepôt

La préparation du chargement des données dans l'entrepôt est une phase très importante de la conception de l'entrepôt. C'est un processus qui doit être exécuté régulièrement pour alimenter l'entrepôt en données mises à jour. C'est un processus d'extraction, de transformation et de chargement (en anglais *extract, transform and load* ou *ETL*) qui permet d'aller chercher les données dans les bases de données sources et qui s'assure de l'intégrité de ces données avant de les charger dans l'entrepôt (voir la section assurance-qualité, 3.2.1.4).

Le chargement fonctionne par insertion de données. Si un enregistrement est ajouté ou modifié dans la base de données source, il est inséré dans l'entrepôt avec la date courante. Si cet enregistrement existe déjà dans l'entrepôt (par exemple, le questionnaire est déjà saisi mais il a été modifié), l'ancienne version reçoit une valeur à son champ de fin de validité pour indiquer qu'une nouvelle version de cet enregistrement est disponible. C'est pour cette raison que l'on peut recréer, à tout moment, les valeurs antérieures contenues dans les bases de données sources de l'entrepôt.

Plusieurs étapes ont été proposées pour permettre le chargement des données d'une manière flexible et facilement extensible (voir l'annexe C). Certaines des étapes mentionnées dans ce document n'ont pas été utilisées puisque les données sources sont déjà vérifiées à la main. De plus, à cause d'un manque d'information au niveau des ratios et des autres calculs utilisés par les chercheurs du Laboratoire, les étapes prévoyant la préparation de champs calculés ne sont pas encore présentes dans l'entrepôt. Cependant, l'architecture actuelle permettra de facilement ajouter ces étapes au chargement lorsque cela s'avérera nécessaire. Un chargement par tables successives a été préféré à un chargement de vues matérialisées en cascade parce qu'il est plus facile d'apporter des changements en utilisant des tables temporaires. Mais les vues matérialisées peuvent quand même être utilisées pour optimiser l'entrepôt. Il est possible d'insérer de nouvelles étapes, au besoin, sans avoir à recréer les vues matérialisées pour chaque étape de chargement. La flexibilité est alors accrue.

3.3 Intégration de toutes les composantes

Le système envisagé comporte de nombreux éléments qui doivent interagir pour permettre le fonctionnement de l'entrepôt, et il faut prévoir des méthodes pour relier tous les systèmes impliqués. La flexibilité et l'extensibilité du système sont des critères très importants dans les choix des supports technologiques. Les logiciels choisis sont Oracle et Tomcat, avec le langage de programmation Java. L'interaction entre ces plates-formes technologiques est déjà éprouvée et la documentation est abondante. Le couplage des éléments du système doit être fiable parce qu'il y a de nombreux points où ils doivent interagir lors de la manipulation des données. L'entrepôt est assez complexe, mais il faut surtout prévoir la possibilité de joindre de nouvelles composantes pour vraiment profiter de ce nouveau système. L'approche modulaire est alors très favorisée.

3.3.1 Choix technologiques

L'introduction d'applications Web au LaRePE est relativement récente. Auparavant, tous les logiciels étaient développés en fonction de *SAS*, *Excel* et *Access*. Les professionnels et chercheurs étaient déjà habitués à *SAS*, et les logiciels de Microsoft sont déjà installés sur la majorité des postes de travail. Les utilitaires développés étaient souvent liés à la configuration d'ordinateurs particuliers, et il est encore aujourd'hui difficile de déplacer ces applications (pour la mise à jour d'un ordinateur, par exemple). De plus, il devient maintenant souhaitable de donner accès aux différents logiciels à des utilisateurs qui sont géographiquement distants. L'utilisation d'applications Web sécurisées devient alors très intéressante. Or, il existe de nombreuses technologies qui permettent le développement d'applications Web plus ou moins complexes (par exemple PL/SQL, ASP, PHP, JSP, etc.). La base de données Oracle, le langage de programmation Java et le conteneur Web Tomcat ont été choisis pour la partie Web de l'entrepôt de données.

3.3.1.1 Oracle

Une des raisons qui a mené au choix d'*Oracle* est que l'UQTR possède un serveur Oracle avec du personnel d'entretien. Il n'est pas nécessaire de développer l'expertise pour installer et maintenir un serveur *Oracle* directement au Laboratoire. De plus, *Oracle* dispose de nombreuses technologies optimisant les performances d'un entrepôt de données. Finalement, la compagnie *Oracle* est un des plus grands fournisseurs de solutions de bases de données, il est facile de trouver des logiciels qui peuvent utiliser ce serveur pour exploiter les données d'un entrepôt.

Le serveur *Oracle* de l'UQTR supporte toute la structure des tables de l'entrepôt. Plusieurs fonctionnalités présentes dans *Oracle* sont utilisées pour améliorer le temps de réponse des logiciels, ainsi que l'intégrité des données lors du chargement. Le stockage d'objets binaires (*BLOB*) sert à conserver les jeux de données créés avec le *Dataset Maker* pour un accès ultérieur plus rapide, ainsi que les cubes de données créés par le logiciel *OLAP ContourCube*. Les informations conservées dans les historiques (*logs*) sont utilisées pour permettre d'annuler un chargement qui a

éprouvé des difficultés, assurant ainsi que seulement des données complètes et correctes sont accessibles aux utilisateurs. Pour plus d'informations sur la structure des tables de l'entrepôt et le rôle d'*Oracle*, voir l'annexe D.

3.3.1.2 Langage de programmation

Les langages disponibles lors du choix étaient : *PL/SQL*, *Java/JSP*, *ASP* et *PHP*. Le langage *PL/SQL* d'*Oracle* a été utilisé pour plusieurs applications Web au Laboratoire, mais sa structure trop rigide l'a exclu de notre liste de langages pour développer l'entrepôt. Le langage *JSP* est une forme de script qui est transformé en code *Java* après quelques traitements. Ce langage facilite le développement d'applications Web en permettant de séparer les sections qui servent à l'affichage dans le navigateur des sections de code qui servent aux divers traitements requis. Ce langage est semblable à *PHP* et à *ASP* de Microsoft. Les serveurs que nous voulions utiliser ne supportaient pas le serveur Web *IIS* de Microsoft pour *ASP*, et les solutions *UNIX* pour *ASP* n'étaient pas intéressantes au moment où nous avons fait les choix technologiques. Les outils de base pour programmer en *Java* et en *PHP* sont accessibles gratuitement pour de nombreuses plates-formes, ce qui permet de facilement développer et tester des prototypes sans avoir à trouver de financement important. Du côté de *PHP*, le langage est disponible, populaire, portable et rapide. Pour certains aspects, *PHP* est supérieur à *Java* et *JSP*, surtout du côté de la rapidité d'exécution. Le choix des langages peut parfois être arbitraire, mais le langage retenu a été *Java/JSP*. Une des raisons pour avoir choisi Java est qu'à la base, l'entrepôt n'est pas un site Web, ni même une application Web ; il fait parti d'un système informatique et peut être exécuté de plusieurs façons. La partie Web n'est qu'une composante du système. C'est pourquoi Java a été choisi, afin de supporter le développement d'utilitaires qui ne seront pas nécessairement Web, mais qui pourront alors s'intégrer à la partie Web avec *JSP* très facilement.

3.3.1.3 Tomcat

Le serveur *Tomcat*[8] est un conteneur Web tel qu'il est décrit dans l'architecture *J2EE* de SUN Microsystems[9]. Le conteneur Web a la responsabilité de la distribution de pages Web et de la sécurité d'accès. Le serveur *Tomcat* est distribué gratuitement par le groupe *Apache* et peut être utilisé pour toute application, commerciale ou non. La programmation de logiciels avec *Tomcat* se fait en *Java* et en *JSP*, et le déploiement de *Tomcat* peut se faire sur tout ordinateur qui possède une machine virtuelle et un compilateur *Java*. En effet, *Tomcat* est entièrement programmé en *Java*, ce qui lui permet d'être aussi portable que les applications qui y sont déployées.

[8] http://jakarta.apache.org/tomcat/
[9] http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html

Les autres possibilités étaient *IIS*, le serveur Web de Microsoft, et *Apache* avec le module *PHP*. Le serveur *IIS* a été écarté puisque les serveurs sur lesquels nous développons fonctionnent avec le système d'exploitation *AIX* d'IBM. Ce système d'exploitation ne permet pas d'exécuter *IIS*, et même si d'autres solutions sont disponibles pour supporter *ASP* sur *Unix*, il ne s'est pas avéré avantageux de retenir cette plateforme de développement. Le serveur *Apache* avec le module *PHP* est déjà accessible sur les serveurs que nous utilisons, mais *JSP* a été choisi comme langage pour les raisons énumérées précédemment.

3.3.1.4 Système d'exploitation AIX

L'Université du Québec à Trois-Rivières possède plusieurs serveurs : un serveur contient la base de données *Oracle* et les applications Web dynamiques avec le langage *PL/SQL*, et un autre serveur utilise *Apache* pour télécharger des pages Webs statiques ou dynamiques à l'aide de *PHP*. Puisque les ressources du Laboratoire sont limitées et que l'entretien d'un serveur Web n'était pas souhaitable à l'interne, nous avons fait appel à l'expertise du Service de l'informatique de l'UQTR. Du coup, nous avons dû nous adapter au système d'exploitation *AIX* qui est utilisé sur ces serveurs. Quelques utilitaires d'entretien de l'entrepôt sont utilisés directement à partir du système d'exploitation. Par exemple, l'utilitaire de chargement est invoqué à partir de la ligne de commande pour automatiser son exécution durant la nuit.

3.3.1.5 Documentation du développement

En considérant la rapidité où les changements peuvent survenir au Laboratoire, il est avantageux d'utiliser une méthodologie orientée objet pour concevoir et documenter le système. Les classes deviennent alors plus facilement réutilisables dans les prochains projets, surtout si elles sont bien documentées. L'héritage est utilisé pour simplifier quelques traitements dans toutes les parties de l'entrepôt, et des outils de documentation automatique comme *Javadoc* et les diagrammes *UML* de *JBuilder* viennent faciliter la tâche de création et d'entretien de cette documentation. Cette documentation est surtout technique et vise à faciliter l'entretien du logiciel une fois en service.

3.3.2 Interaction des composantes du système

Les éléments actifs du système sont (voir la figure 13) :

- le serveur *Oracle* pour la base de données ;
- le serveur *Tomcat* pour la distribution des pages Web dynamiques ;
- l'utilitaire *Java* pour charger les données dans l'entrepôt ;
- le serveur *Apache* pour servir le contenu Web avec une connexion *SSL* ;
- le serveur avec le système d'exploitation *AIX* qui supporte tous les logiciels côté serveur.

Le serveur *Tomcat* et l'utilitaire de chargement sont reliés à la base de données pour manipuler les enregistrements des magasins de données. Le serveur *Apache* est une composante optionnelle qui permet d'utiliser le certificat de sécurité de l'Université pour transférer le contenu Web avec cryptage des données. Tous ces programmes sont exécutés sur un serveur *AIX* d'IBM.

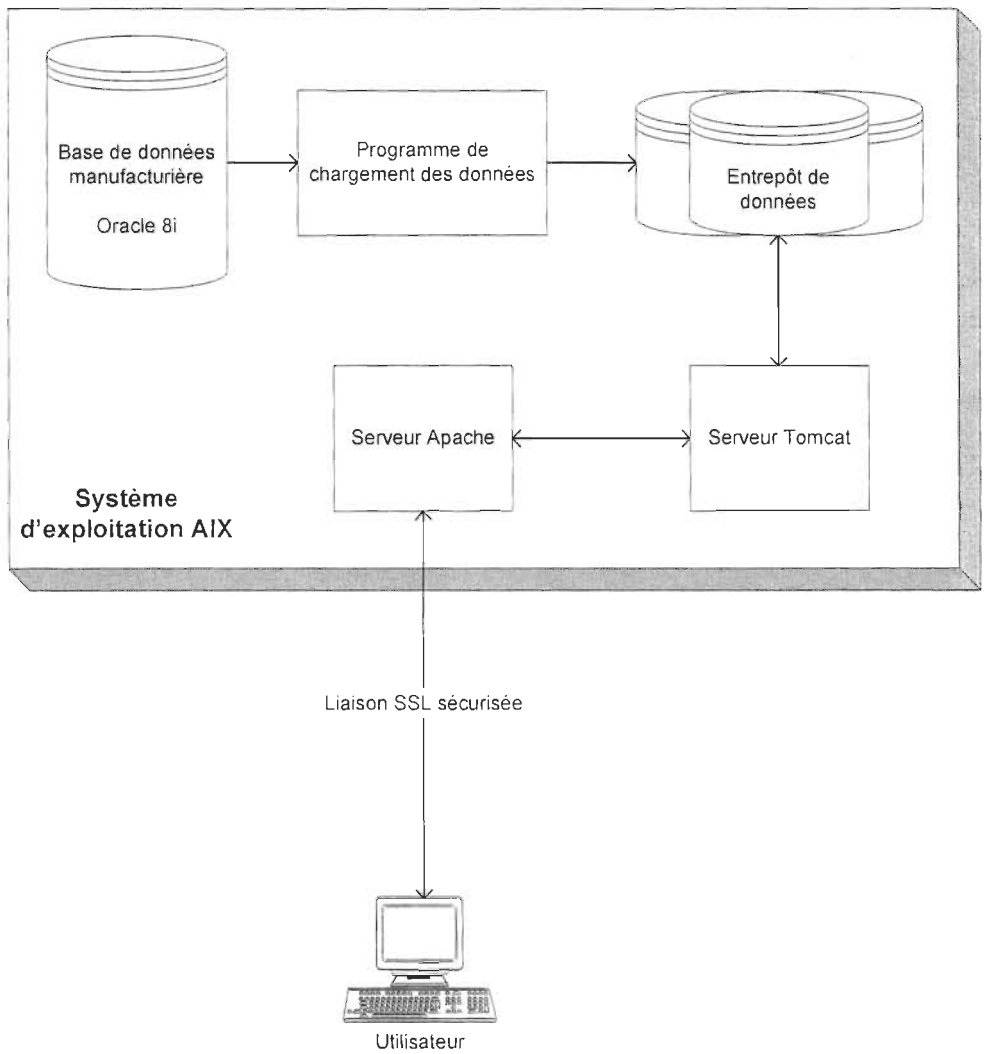


Figure 13 L'interaction entre les différentes composantes liées à l'entrepôt

3.4 Préparation des données pour les autres applications du LaRePE

Comme il a été mentionné précédemment, l'entrepôt doit permettre plusieurs sortes d'utilisations différentes de ses données. Certaines applications doivent pouvoir s'alimenter automatiquement à partir de l'entrepôt pour pouvoir fonctionner (PDG, Balise), et il y a aussi la nécessité de créer des jeux de données pour les utilisateurs lors de projets de recherche. Il peut aussi être nécessaire d'explorer les données pour répondre à des questionnements internes d'ordre administratif au sujet de la base de données. Cette dernière catégorie sera traitée plus en détail dans le chapitre 4. Les deux applications automatiques citées sont décrites plus en détail dans les sous-sections suivantes.

3.4.1 Le rapport PDG

Le rapport PDG (voir l'annexe A) est un logiciel de *benchmarking* qui est développé au LaRePE et en évolution depuis plusieurs années. Le résultat du travail sur ce système est un rapport riche en informations sur plusieurs domaines de la gestion d'une PME manufacturière et compréhensible par les décideurs de telles entreprises. Les données nécessaires à la création du rapport PDG sont tirées d'un questionnaire de 20 pages qui est rempli en entreprise. Ce questionnaire est alors saisi dans un formulaire informatique au LaRePE, après être passé par un organisme intermédiaire pour assurer la confidentialité des entreprises.

Ce rapport utilise depuis toujours la base de données manufacturière de manière relativement brute. Un programme de statistiques est utilisé pour manipuler et transformer les données pour leur donner une forme correcte pour les traitements. Après ce travail, des statistiques sont générées puis récupérées par une feuille de calcul *Excel*. Le désavantage de ce système est que de nombreux intermédiaires logiciels (voir figure 14) et humains sont nécessaires entre la base de données et la feuille finale, et que plusieurs calculs sont redondants d'un questionnaire à l'autre. Un autre désavantage de l'utilisation directe de la base de données est qu'il est maintenant devenu souhaitable de rendre le rapport accessible par le Web, ce qui est difficile et même risqué avec la chaîne de traitement actuelle.

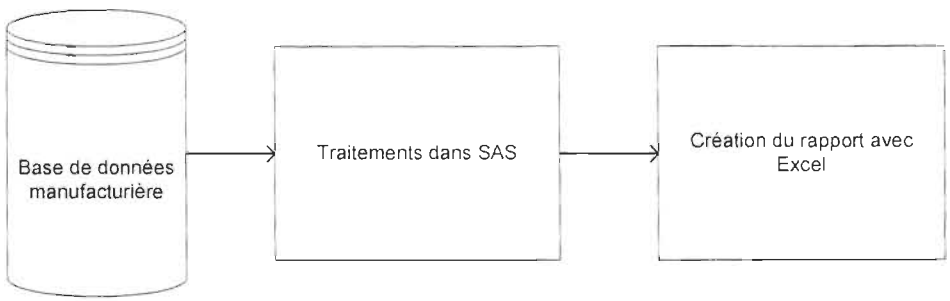


Figure 14 Une version simplifiée de la chaîne de traitement utilisée pour créer un rapport

C'est pourquoi il est prévu de relier le système de création du rapport à l'entrepôt. Le nouveau système de rapport ne sera pas encore opérationnel lors de la publication de ce mémoire (et fera d'ailleurs l'objet d'un projet de maîtrise distinct), mais des résultats de création de rapports avec l'entrepôt sont tout de même présentés dans le chapitre 5.

3.4.2 L'application *Balise*

L'application Web *Balise*[10] est un logiciel de *benchmarking* développé par le LaRePE. Elle utilise temporairement des données de la base de données manufacturière depuis son lancement, afin d'avoir une bonne base de comparaison pour les utilisateurs. De nombreux problèmes de transformations de données ont dû être réglés pour permettre à ce logiciel de fonctionner sans un programme statistique spécialisé. Un entretien continu doit être maintenu afin de s'assurer qu'il n'y a pas de données indésirables qui s'infiltrent dans *Balise*. Un magasin spécialisé pour les rapports permettrait de régler plusieurs des problèmes de transformation des données. La préparation et le calcul des données pourraient se faire dans l'entrepôt, et les applications qui ont besoin de ces données n'auraient qu'à les transférer avec un minimum de transformations. Mais, pour l'instant, il n'est pas prévu de relier l'alimentation en données de *Balise* à l'entrepôt.

Un autre aspect de *Balise* est à considérer : *Balise* collecte des données qui sont comparables à celles récupérées à partir des questionnaires manufacturiers, mais cette collecte est faite sur le Web. Bien que le questionnaire de *Balise* est loin d'être aussi complet que le questionnaire manufacturier, il serait intéressant de transformer la base de données de *Balise* en une source pour l'entrepôt. Les structures de chargement de l'entrepôt ont justement été créées pour répondre à ce genre d'éventualité. Une description plus complète de cette possibilité est faite au chapitre 6.

3.4.3 La recherche

Les projets de recherche utilisent activement toutes les données qui sont saisies dans les systèmes du LaRePE. Il y a les mémoires de maîtrise et thèses de doctorat, des projets d'exploration, des articles et des communications scientifiques, etc. Pour ces projets de recherche, la création de jeux de données permet de vérifier et de valider des hypothèses à l'aide de statistiques. Des exemples de projets de recherche en cours sont le mémoire d'une étudiante de maîtrise, un article sur la croissance des PME et un autre sur les ressources humaines. Dans le cas des mémoires et des articles, les hypothèses sont posées à l'avance, et c'est plus tard que les variables sont identifiées pour permettre une vérification statistique des hypothèses. Ces variables sont utilisées pour créer des jeux de données à partir de la base de données manufacturière. Le processus de création de ces jeux de données prend plusieurs jours, mais la transition au *Dataset Maker* permettra de réduire significativement ce

[10] http://www.balise.ca

temps. Une fois le jeu de données créé, des tests statistiques sont faits. Cependant, pour diverses raisons, il peut arriver qu'après un certain temps, il soit nécessaire de faire un nouveau jeu de données, soit parce que de nouvelles variables doivent être ajoutées, soit pour faire une mise à jour des données pour récupérer les nouveaux questionnaires. À ce moment, il faut retrouver les procédures *SAS* qui ont permis de créer le jeu de données, et de le mettre à jour. Avec le *Dataset Maker*, cette étape pourra être complétée en quelques minutes puisque tous les paramètres utilisés pour créer un jeu de données sont conservés, et il est possible de les modifier à tout moment pour créer un nouveau jeu de données. Mais les méthodes utilisées pour la recherche peuvent changer, et les besoins peuvent évoluer. C'est pourquoi il faut rester vigilant et tenter d'aller au-devant des besoins des chercheurs lorsqu'il est question des projets de recherche utilisant les données du Laboratoire.

CHAPITRE 4

VALEUR AJOUTÉE DE L'ENTREPÔT

Tel qu'il a été présenté au chapitre 3, l'entrepôt de données permet de remplacer l'ancien système d'information du LaRePE lorsqu'il est question d'extraire des informations recueillies à l'aide des questionnaires manufacturiers. Ce nouveau système favorise de nouvelles méthodes pour utiliser les données, des méthodes qui n'étaient pas employées auparavant à cause de structures trop peu flexibles ou tout simplement à cause des difficultés d'accès aux données. Ce système est une base solide pour permettre l'analyse basée sur le forage de données (*data mining*) et *OLAP*. Puisque les données sont réorganisées séparément des sources de données, il est possible de les restructurer pour faciliter l'ajout de ces nouvelles méthodes sans affecter la saisie des données. L'entrepôt est un lieu très propice à l'exploration de différentes sources de données (le questionnaire manufacturier, et d'autres bases de données qui pourront être ajoutées) et permet une approche modulaire pour la création de nouveaux projets.

4.1 Magasin dimensionnel

Un magasin dimensionnel (fonctionnant à partir du modèle dimensionnel avec un schéma en étoile) permet d'utiliser des outils *OLAP* et des logiciels de forage de données. C'est à partir de ce magasin que l'exploration des données se fait sous un tout nouvel angle. Le magasin dimensionnel est une grande valeur ajoutée au travail en cours sur le forage de données et la base de données manufacturière du LaRePE. Les logiciels de forage de données n'ont pas tous besoin d'un magasin utilisant le modèle dimensionnel pour fonctionner, mais l'exercice d'exploration permet tout de même d'utiliser de nouvelles techniques comme *OLAP* pour atteindre une compréhension accrue de certains phénomènes dans la base de données.

Les figures 15 et 16 illustrent le schéma en étoile des points de vues retenus pour les tables de faits, des questionnaires et des entreprises respectivement. Le magasin dimensionnel qui utilise les questionnaires comme fait de base (figure 15) possède les tables de dimension pour la région administrative, les clients du Laboratoire et le secteur d'activité. D'autres dimensions sont pour l'instant implicites dans la table de faits, il s'agit du type de questionnaire, de la date de réception, du sexe du dirigeant, de la version et de l'état d'activité du questionnaire. La table avec le suffixe *_JOIN* permet au logiciel *ContourCube* (voir la sous-section 4.2.1) de préparer les données pour qu'elles soient affichées. Le magasin dimensionnel qui utilise les entreprises comme fait de base (figure 16) possède moins de champs dans la table de faits. Certains champs concernent uniquement les questionnaires et ne sont pas applicables directement aux entreprises. La seule nouvelle dimension est *PARTICIPATION*, qui représente le nombre de questionnaires que cette entreprise a retournés au LaRePE.

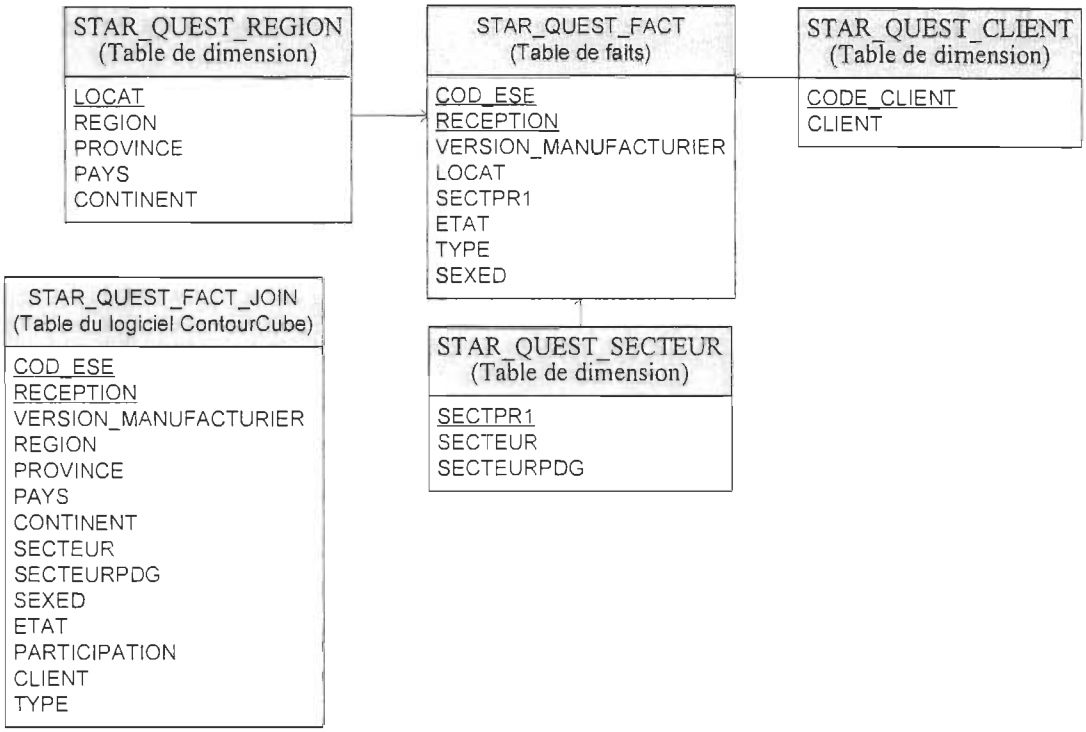


Figure 15 Schéma en étoile pour la table de faits des questionnaires

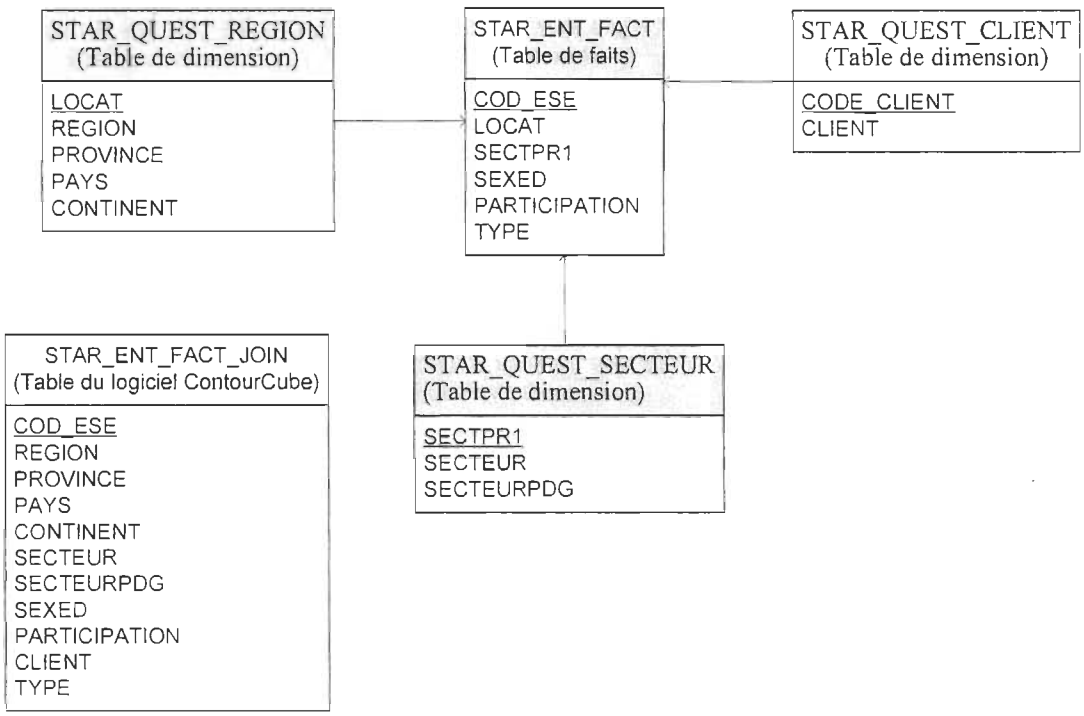


Figure 16 Schéma en étoile pour la table de faits des entreprises

Conceptuellement, on peut voir le magasin dimensionnel comme étant un magasin satellite de l'entrepôt principal. D'ailleurs, les données sont préalablement stockées dans le magasin de données historiques. Un nouveau processus de chargement permet de copier les données nécessaires dans la table de faits. La préparation des devises et le calcul du suivi des valeurs dans le questionnaire de mise à jour sont essentiels pour le fonctionnement du magasin dimensionnel. C'est pourquoi ce magasin est alimenté par le magasin historique. Plusieurs architectures sont possibles pour créer une structure dimensionnelle propre à *OLAP* ou au forage de données. Il y a le schéma en étoile et le schéma en flocon, ainsi que d'autres modèles moins connus (voir la sous-section 2.2.7). Il est peu recommandé d'utiliser le schéma en flocon, c'est pourquoi le schéma en étoile a été choisi pour structurer ce magasin.

4.1.1 Tables de faits

Dans la base de données manufacturière, il est possible d'identifier tous les questionnaires correspondants à une entreprise. Elle utilise un système de codes pour retrouver les entreprises, et l'année de réception pour classer les questionnaires. À partir de ces clés, deux tables des faits ont été préparées pour le magasin dimensionnel. Les informations que l'on peut tirer en se basant sur les questionnaires ne sont pas les mêmes que celles que l'on peut obtenir en se basant uniquement sur les entreprises. Il faut choisir un point de vue des données, les faits ne sont pas les mêmes selon ce point de vue. Par exemple, on peut vouloir savoir combien de questionnaires ont été reçus le mois dernier (point de vue des questionnaires), ou encore combien d'entreprises sont dans le secteur d'activité de l'alimentation (point de vue des entreprises). C'est pourquoi plusieurs tables de faits sont nécessaires. Par contre, chaque table ne comporte pas exactement les mêmes dimensions. Certaines informations propres aux questionnaires sont difficilement applicables aux entreprises. Par exemple, l'information de la date de réception du questionnaire ne peut pas vraiment être transposée à l'entreprise (on aurait une *date de réception* de l'entreprise ?). Certaines autres dimensions sont très utiles dans les deux cas, comme le secteur d'activité ou la région administrative de l'entreprise. Il y a de nombreuses autres possibilités pour la création de dimensions et l'ajout de données utiles aux utilisateurs. La création d'un entrepôt de données est un processus itératif, et nous sommes tout juste à la fin de la première itération. Ce sont les utilisateurs qui guideront les prochains changements dans l'entrepôt.

Pour l'instant, dans les deux tables de faits, la seule opération faite sur les données est le calcul du nombre d'enregistrements. Il n'y a pas encore de données qui permettent de faire des moyennes ou des sommes d'enregistrements. Cependant, l'ajout de telles données pourrait être très utile pour l'exploration de l'entrepôt. Par exemple, l'ajout de la donnée des ventes pour chaque entreprise pourrait servir à faire des moyennes ou des médianes instantanément dans le logiciel *OLAP*. L'âge de l'entreprise est accessible sous forme de dimension, mais il serait possible de l'ajouter dans un nouveau champ numérique en tant que fait pour permettre des calculs statistiques rapides.

Les bases de données qui sont utilisées en ce moment ou qui pourront prochainement servir à l'alimentation de l'entrepôt ne sont pas très volumineuses quant au nombre d'enregistrements qu'elles contiennent. Ce ne sont pas des systèmes transactionnels avec des milliers d'opérations chaque jour. C'est pourquoi il n'a pas été jugé utile d'effectuer des agrégations à l'avance, lors du chargement des données dans les tables de faits. Chaque fait représente alors un questionnaire, ou encore une entreprise, selon la table de faits. D'autres bases de données du Laboratoire pourront être greffées à l'entrepôt, et le même principe pourra s'appliquer tant que l'on continue à avoir comme unité de base le questionnaire ou l'entreprise. Si de nouvelles tables de faits avec des unités de bases différentes doivent être ajoutées à l'entrepôt, il pourra alors devenir avantageux d'effectuer des agrégations lors du chargement. Mais pour l'instant, le temps de traitement durant les calculs et la quantité d'espace disque utilisé ne le justifient pas.

4.1.2 Dimensions

Une des suggestions faite par certains auteurs est de garder les mêmes tables de dimensions pour toutes les tables de faits quand c'est possible (Kimball, 2003), ce qui permet de faire des comparaisons entre les faits plus facilement. C'est l'opération dite *drill accross*. Dans le cas des dimensions utilisées dans l'entrepôt, il y en a plusieurs qui sont réutilisées entre les tables de faits. Par exemple, les dimensions *région* et *secteur d'activité* sont utilisées à la fois pour les questionnaires et pour les entreprises.

De nombreuses dimensions sont possibles, et rendre accessible un aussi vaste choix ne semblait pas pratique ni nécessaire, du moins pour le moment. C'est pourquoi les utilisateurs sont avertis qu'il est possible de demander des ajouts au magasin dimensionnel, selon leurs besoins. Avec le logiciel choisi pour *OLAP*, il est même possible de personnaliser les dimensions accessibles à chaque utilisateur (voir la sous-section 4.2.1). Seules quelques dimensions ont été retenues pour la présentation initiale des outils *OLAP*. Les données utilisées dans l'entrepôt permettent de créer des centaines de dimensions, mais afin de permettre aux utilisateurs de se familiariser avec le système et de le rendre accessible le plus rapidement possible, seules ces dimensions ont été créées :

- région administrative ;
- âge de l'entreprise ;
- nombre d'employés de l'entreprise ;
- secteur d'activité ;
- date de réception du questionnaire ;
- client qui a transmis le questionnaire.

Plusieurs suggestions d'ajouts ont déjà été formulées. Par exemple, il serait utile d'ajouter le taux de croissance des ventes des entreprises sur plusieurs années, le type de production principal de l'entreprise (production unitaire, par lots...), le chiffre d'affaires de l'entreprise, et plusieurs autres. Les utilisateurs sont invités à

suggérer l'ajout de nouveaux éléments (dimensions, données, faits) afin de rendre le système plus adapté à leurs besoins, et plus complet.
Les dimensions dans l'entrepôt peuvent être sujettes à des changements, avec le temps (voir dimensions à changement lent dans la sous-section 2.2.8.1). Cependant, la plupart des changements pourraient être traités par l'ajout d'une nouvelle entrée dans la table des dimensions. Par exemple, il pourrait arriver qu'une nouvelle région administrative soit créée au Québec, alors à partir de ce moment, il serait possible de créer un nouvel enregistrement pour cette dimension et d'insérer de nouveaux enregistrements correspondants aux entreprises qui se retrouveraient alors dans la nouvelle région. De cette façon, les données analysées à partir de ce moment seraient bien classées avec la nouvelle région, et l'analyse des données antérieures serait aussi conforme à l'historique. Comme la majorité des cas envisagés de changements apportés aux dimensions peuvent être traités avec l'ajout de nouveaux enregistrements, aucun autre mécanisme explicite pour traiter ces situations n'a été incorporé à l'entrepôt.

4.1.3 Optimisations

Le logiciel de bases de données utilisé est *Oracle 8i*. Ce gestionnaire de bases de données comporte quelques éléments pour les tables dimensionnelles servant à *OLAP*, dont quelques structures de programmation comme « DIMENSION » qui permet de définir une table comme dimension. Des indices comme « STAR JOIN » peuvent être utilisé dans une requête pour accélérer les jointures entre les dimensions et la table de faits. Des opérateurs comme « CUBE » permettent à un logiciel *OLAP* de faire ses calculs à partir du serveur et de télécharger uniquement les résultats, réduisant ainsi le volume de données qui circule sur le réseau.

Ces optimisations n'ont pas été utilisées dans l'implémentation de l'entrepôt. La raison principale est que le volume de données n'est vraiment pas suffisant pour justifier un travail d'optimisation en profondeur. D'ailleurs, la quantité de données n'est pas sujette à augmenter très rapidement. Les logiciels permettant d'utiliser en profondeur toutes les capacités d'*Oracle* requièrent des investissements en personnel et en argent très importants. Plusieurs logiciels *OLAP* et de forage de données nécessitent au moins la version 9i de la base de données *Oracle*, et cette version n'était pas encore disponible sur les serveurs utilisés par le LaRePE au moment d'activer l'entrepôt. C'est pourquoi un logiciel de moindre envergure, mais offrant tout de même d'excellentes fonctionnalités et performances, a été choisi pour l'entrepôt. Ce logiciel n'est pas suffisamment robuste pour fonctionner dans un entrepôt de données avec des milliards d'enregistrements, mais ce n'est pas du tout la situation de l'entrepôt de données créé au LaRePE. De plus, lorsque *Oracle 9i* (ou *10g*) sera disponible sur les serveurs, il sera alors possible d'effectuer certaines optimisations tout en ayant déjà accumulé une expérience avec l'entrepôt.

4.2 OLAP

Certains rapports sur la gestion des dossiers du Laboratoire existent déjà dans le système actuel. Cependant, les utilitaires employés pour les créer sont peu conviviaux pour les utilisateurs non spécialisés en informatique. Les langages utilisés sont *SQL* et *PL/SQL*. De nombreuses demandes d'information sont faites à l'interne pour avoir des renseignements supplémentaires sur les données. Par exemple, nous avons des rapports qui découpent le nombre de dossiers par régions géographiques, par secteurs d'activités et par mois de réception des dossiers. Il est courant qu'une demande soit faite pour avoir accès au nombre de dossiers reçus durant un mois particulier pour chaque secteur, au Canada. Ce genre de requête (voir figure 17) est facile à effectuer avec un bon logiciel *OLAP*.

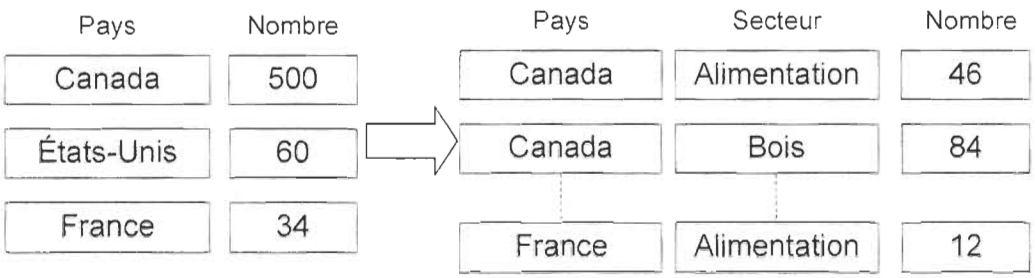


Figure 17 Démonstration d'une opération OLAP drill-down

4.2.1 Logiciel ContourCube

Le logiciel retenu pour *OLAP* au LaRePE est *ContourCube*[11] (voir l'annexe G). C'est une composante *ActiveX* qui peut être intégrée à une application *Visual Basic* ou dans une page Web visualisée par *Internet Explorer* (*ActiveX* est une technologie de Microsoft, Netscape n'est pas supporté). Cette application est capable d'utiliser un lien *ODBC* pour se connecter à une base de données. Par contre, il est aussi possible de générer des fichiers de cubes compressés (des .cube) qui peuvent ensuite être conservés sur un serveur Web et téléchargés au besoin. Cette dernière option a été favorisée puisque, de cette manière, on assure une connexion sécurisée et seuls les utilisateurs authentifiés sont capables d'accéder aux données. Le fichier de cube est généré par un serveur Windows 2000 et téléchargé dans une table du serveur *Oracle* sous forme de *BLOB*. Une page dynamique permet ensuite de télécharger le cube demandé. Avec une connexion *ODBC*, cette logistique est beaucoup plus compliquée, du moins du point de vue de l'utilisateur qui doit s'assurer d'avoir les pilotes pour se connecter à Oracle.

[11] http://www.contourcomponents.com/

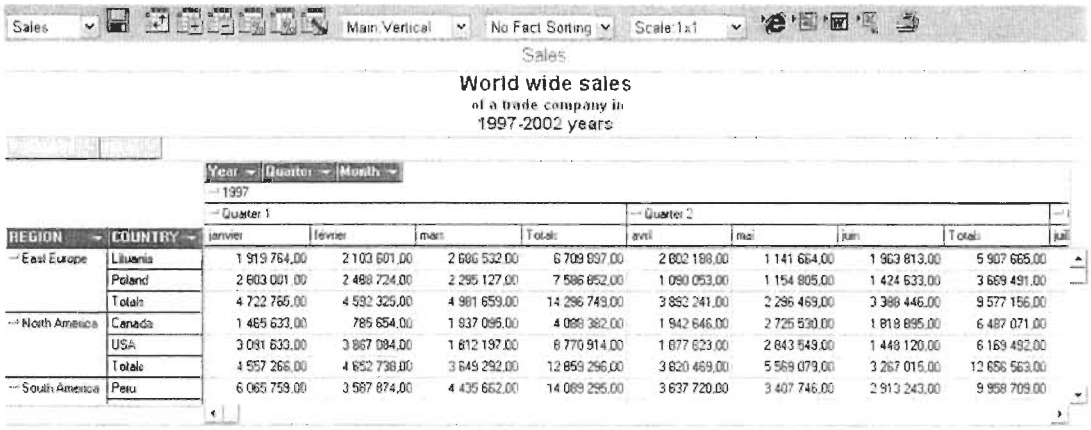


Figure 18 Démonstration de la version Web de ContourCube

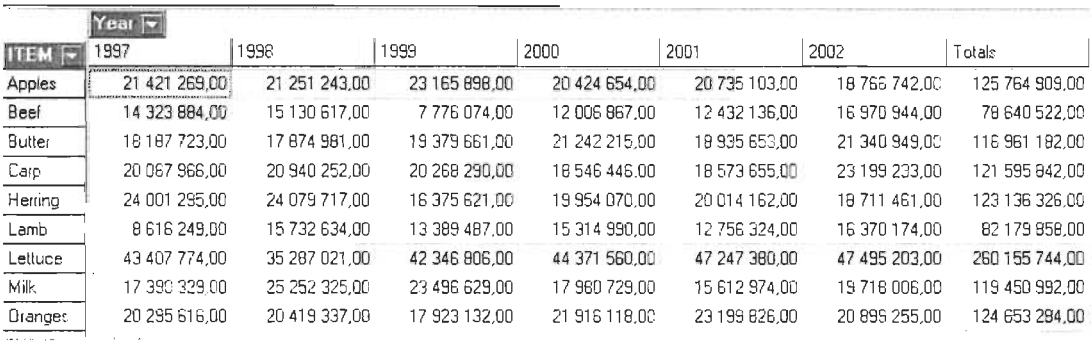


Figure 19 Quelques manipulations simples permettent de changer rapidement les dimensions

Pour des raisons de confidentialité, les vraies valeurs de l'entrepôt ne sont pas montrées dans les figures 18 et 19. L'exemple est fictif, mais il permet d'apprécier la versatilité et facilité d'utilisation de ce logiciel. Une version préparée dans l'entrepôt est fonctionnelle et accessible par les utilisateurs du LaRePE (voir la section 5.5 pour un exemple produit à partir de l'entrepôt).

Le logiciel *ContourCube* utilise une table de faits où la jointure avec les dimensions est déjà faite. Cette table est téléchargée puis peut être conservée dans un fichier pour un accès ultérieur. Des paramètres sont conservés dans ce fichier pour identifier les dimensions et leurs valeurs possibles. Une fois les données prêtes, le logiciel affiche un tableau et une liste de dimensions qui peuvent être facilement manipulées à l'aide de la souris (les boutons *PRODUCT, ITEM, YEAR, QUARTER, MONTH, REGION* et *COUNTRY* dans la figure 18). Il est possible d'afficher une dimension en rangées ou en colonnes en la déplaçant avec la souris. Il est aussi possible d'ajouter plusieurs dimensions, et les colonnes ou les rangées sont recalculées instantanément. Des calculs comme des moyennes ou des médianes sont prévus dans l'interface, bien que ces fonctionnalités ne soient pas encore utilisées dans la version disponible au LaRePE.

4.3 Outils de forage de données

Les méthodes d'exploitation des données permettent différentes techniques pour la création de rapports et la recherche scientifique. Il est possible d'utiliser les logiciels de forage de données pour parcourir l'ensemble des données de l'entrepôt. Puisque la base de données manufacturière est très grande, la recherche se limite souvent à des sous-ensembles. Avec le forage de données, les algorithmes pourront utiliser toutes les données de cette base de données, ainsi que toutes les autres contenues dans l'entrepôt, et identifier les liens entre plusieurs thèmes. Par exemple, avant on pouvait poser la question : est-ce qu'il y a un lien entre l'innovation d'une PME et sa situation financière ? Ou est-ce qu'il est important d'avoir des collaborations pour augmenter son chiffre d'affaires ? Ces hypothèses peuvent être vérifiées avec des techniques statistiques. Maintenant, avec un système de forage de données, il est possible de poser des questions comme : Qu'est-ce qui influence le chiffre d'affaires ? Qu'est-ce qui influence les collaborations ? Qu'est-ce qui influence l'innovation ? Toutes les données de l'entrepôt peuvent être mises à contribution pour répondre à ces questions. Les variables les plus susceptibles d'influencer le thème choisi peuvent être sélectionnées à l'aide du Dataset Maker, puis utilisées dans des logiciels de forage de données.

4.3.1 Forage de données avec SAS Enterprise Miner

Le logiciel retenu au Laboratoire pour le forage de données est *SAS Enterprise Miner*[12]. La raison principale de ce choix est que ce logiciel est déjà inclus dans la licence du logiciel *SAS* utilisé pour les statistiques du Laboratoire. Les méthodes de préparation de jeux de données avec *SAS* sont déjà présentes dans le *Dataset Maker*, le passage à *Enterprise Miner* est donc plus simple. Ce logiciel est très convivial et fonctionne à partir d'une interface graphique pour la majorité de ses opérations, tout en permettant la création de macros et autres procédures *SAS* pour automatiser les processus de traitement des données créés lors du forage de données. Les chercheurs et assistants du LaRePE n'ont pas encore été familiarisés avec l'utilisation du forage de données. Les outils sont tout de même à leur disposition, il ne reste qu'à donner des séances de formation. Comme il a été décrit à la sous-section 2.4, le forage de données permet de nouvelles façons d'explorer les données. L'ajout d'*Enterprise Miner* ne vise pas à remplacer les méthodes d'analyse déjà implantées, mais plutôt à les compléter en permettant de poser les questions différemment.

Certains exemples d'utilisation d'*Entreprise Miner* sont présentés. Pour des raisons de confidentialité, il n'est pas possible de montrer un exemple de forage de données avec les données de l'entrepôt. Mais il est possible de consulter le tutoriel d'*Enterprise Miner* à l'annexe H, et le chapitre 5 présente un exemple de jeu de données importées dans le logiciel. *SAS Enterprise Miner* fonctionne avec une interface graphique qui offre de nombreux éléments pour manipuler les données et appliquer des algorithmes de forage de données. À la base, il faut choisir une ou plusieurs sources de données. Ces sources de données peuvent provenir d'un fichier Excel, d'un jeu de données SAS, d'un entrepôt de données, etc. Le processus de

[12] http://www.sas.com/technologies/analytics/datamining/miner/

nettoyage de données est très important, car même si elles proviennent d'un entrepôt avec des données qui ont déjà été nettoyées, le nettoyage pour faire du forage de données ne fonctionne pas nécessairement sur les mêmes principes. Des données qui sont *propres* pour l'entrepôt ne le sont pas nécessairement pour un projet particulier de forage de données dépendamment des algorithmes particuliers qui seront appliqués (voir la section 2.4). Plusieurs autres opérations de préparation de données sont incluses. Ce logiciel donne accès aux algorithmes d'arbre de décision, de réseaux neuronaux, de régressions logistiques et linéaires, ainsi que plusieurs autres. Un schéma complet de projet de forage de données peut alors comprendre de nombreuses opérations et utiliser plus d'un algorithme (figure 20).

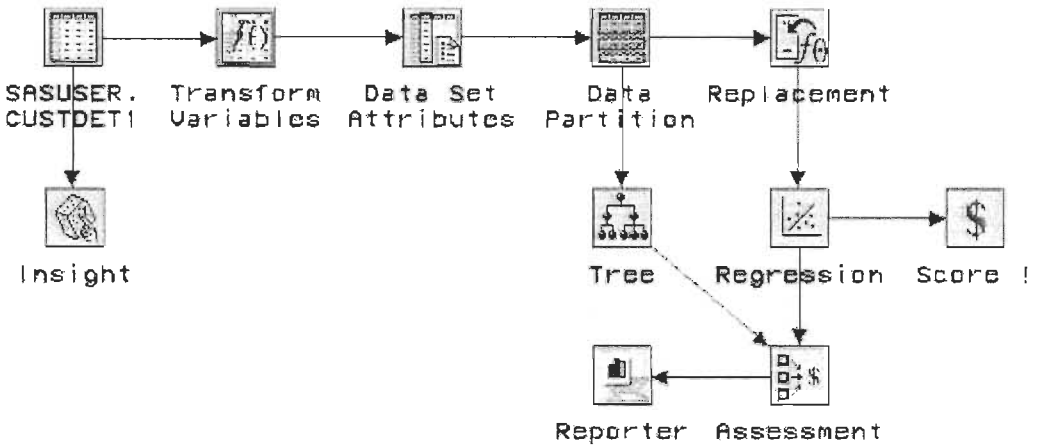


Figure 20 L'exemple d'un projet complet de forage de données dans SAS Enterprise Miner

Le logiciel *SAS Enterprise Miner* permet d'afficher les résultats à l'aide de différentes interfaces, incluant une interface qui combine les résultats de plusieurs algorithmes différents afin de comparer les résultats obtenus (figures 21 et 22). La plupart des éléments dans les fenêtres sont modifiables par l'utilisateur, et il est toujours possible de revenir en arrière pour changer les paramètres du projet. Un avantage de ce logiciel est que chaque étape du projet n'a pas à être recalculée chaque fois qu'un résultat est demandé, sauf si des changements ont été apportés. Ceci permet d'améliorer les performances lors de l'utilisation de nouveaux algorithmes, et incite l'utilisateur à faire de nombreuses expérimentations avec les données ce qui est tout à fait conforme à l'esprit de la démarche de forage de données.

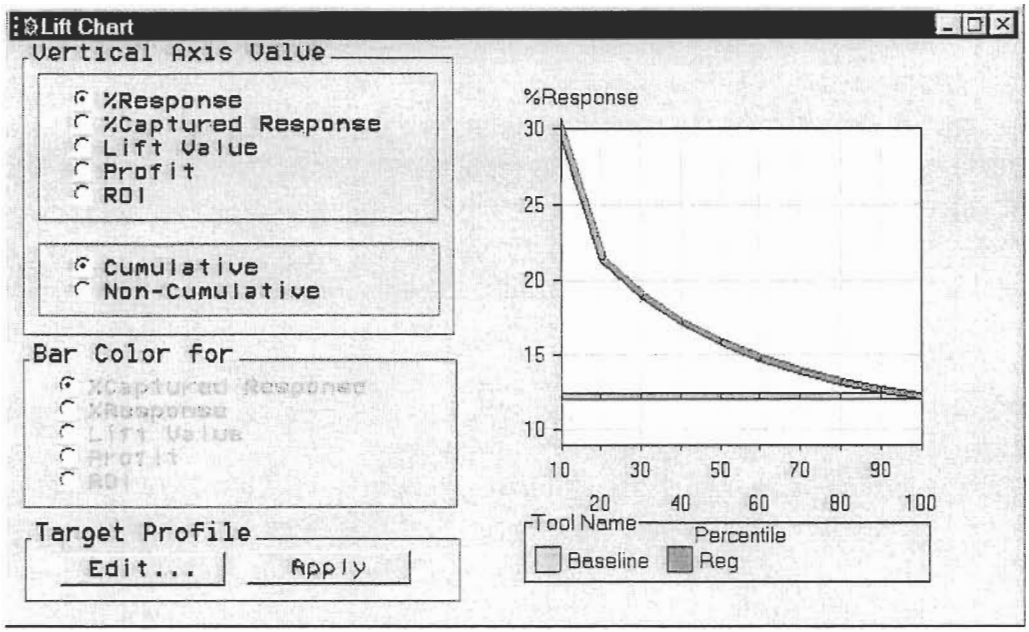


Figure 21 Interface permettant d'explorer certains résultats du forage de données

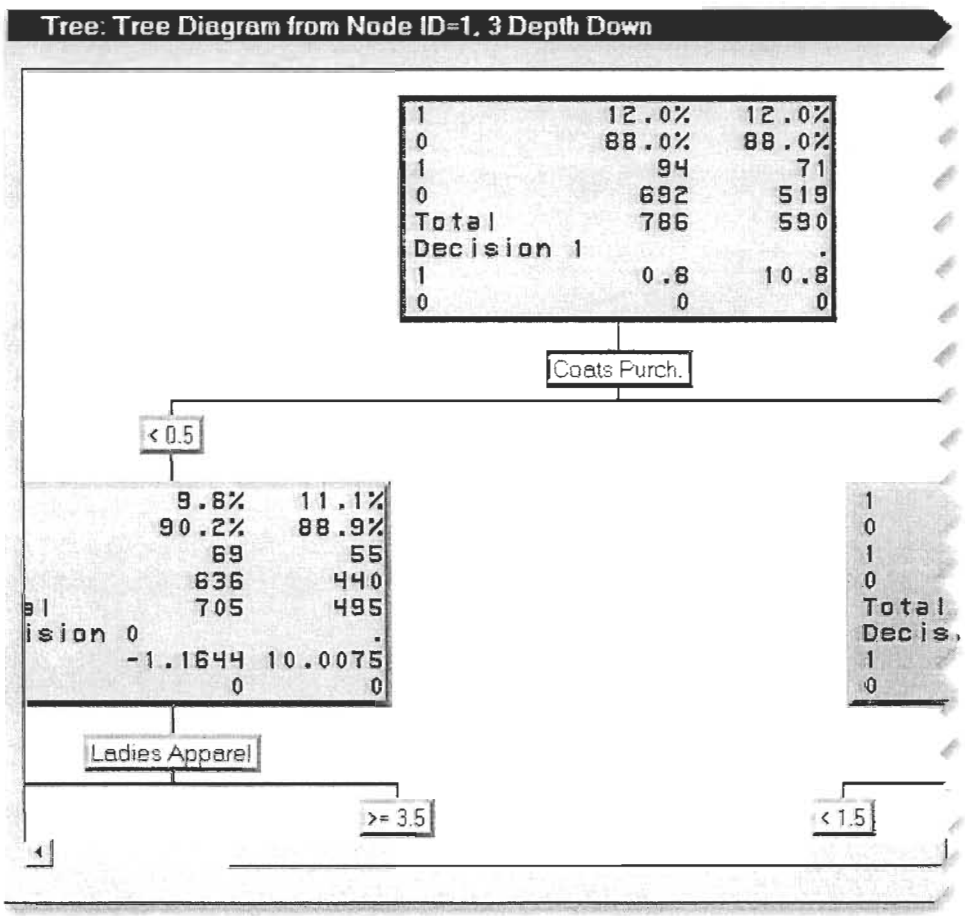


Figure 22 Utilisation des arbres de décision avec SAS Enterprise Miner

Il faut noter que la création de l'entrepôt constitue une base solide pour le forage de données. L'entrepôt permet l'utilisation du forage de données, et les utilisateurs qui désirent en faire l'essai le peuvent déjà. Des formations sont à prévoir pour faciliter l'adoption de ces nouveaux outils par les personnes impliquées au LaRePE.

4.3.2 Autres logiciels

Plusieurs autres logiciels pourront s'ajouter à la liste des outils de forage de données du Laboratoire. Comme il a été mentionné précédemment, le but de ce mémoire était la création d'un entrepôt supportant la recherche et les rapports au LaRePE, ainsi que des nouvelles applications comme le forage de données (*data mining*). C'est pourquoi l'accent a été mis sur les différentes structures de l'entrepôt plutôt que sur le processus de forage de données lui-même. Cependant, certains logiciels ont tout de même été identifiés (voir l'annexe B). Un nouveau logiciel a déjà été proposé par un chercheur. Il s'agit de *NeuroSolutions*[13], un logiciel qui permet de créer des réseaux neuronaux à l'aide d'assistants de création. Ce logiciel peut fonctionner indépendamment ou avec Excel. Il serait même possible de l'alimenter à partir de l'utilitaire *Dataset Maker* créé spécifiquement pour utiliser les données de l'entrepôt. Évidemment, il existe de nombreux autres logiciels dont pourraient se servir les chercheurs, et certaines ressources sur le Web existent pour aider à les trouver. Il y a, par exemple, le site KDNuggets[14] qui se spécialise dans l'identification d'applications pouvant servir à l'analyse de données, comme le forage de données ou le *knowledge discovery*. Certains logiciels décrits sur ce site sont disponibles en version d'essais, pendant 30 ou 60 jours. Ce ne sont pas les ressources qui manquent, et puisque l'entrepôt rend l'utilisation des données beaucoup plus facile qu'auparavant, les chercheurs pourront enfin se concentrer sur de nouvelles méthodes pour les utiliser plutôt que de passer de nombreuses heures à faire des manipulations préparatoires répétitives.

4.4 Objectifs à long terme

Au LaRePE, l'entrepôt est apparu comme une solution à un problème de flexibilité grandissant avec le système d'information créé autour du rapport PDG. Il faut assurer une production continue des rapports, et comme il a déjà été vécu lors d'expériences antérieures, la présence d'un système parallèle en développement n'est pas souhaitable à cause des changements rapides qui surviennent sur ce système. Deux phases de transition entre des systèmes différents ont déjà eu lieu par le passé, et ces phases se sont soldées par de nombreux problèmes (Dugré et Delisle, 2003) provenant surtout du temps de développement nécessaire pour les nouveaux systèmes de remplacement. En créant un entrepôt, le système déjà en fonction continue ses opérations normalement, et une fois l'entrepôt prêt, il est alors possible de faire passer les logiciels de l'ancien au nouveau système avec un minimum de problèmes. La création de l'entrepôt a pris beaucoup moins de temps qu'une révision complète du système en une seule phase.

---

[13] http://www.neurosolutions.com/
[14] http://www.kdnuggets.com/

Une approche modulaire fait maintenant partie de la nouvelle méthode lors de la création de projets. Un modèle de connexion pour les nouvelles sources de données est d'ailleurs proposé dans l'annexe C. Ceci nous amène à une plateforme beaucoup plus intéressante pour développer de nouvelles méthodes et nouveaux outils de diagnostic pour les PME. Et pour améliorer la qualité des évaluations posées par les différentes applications reliées au système, il est même envisagé d'utiliser certaines techniques d'intelligence artificielle.

4.4.1 Approche modulaire

Une des raisons majeures du travail sur l'entrepôt est la nécessité de supporter une approche modulaire pour les nouveaux projets. De nouveaux projets nécessitant une réorganisation des structures de données sont souvent lancés au Laboratoire, et l'entrepôt permet de faire une distinction claire entre la phase de préparation pour la saisie (création d'une base de données et de formulaires pour saisir les données) et de la phase d'exploitation des données (création d'un ou plusieurs programmes de rapports utilisant les nouvelles données). Plutôt que d'avoir à continuellement tenter d'intégrer de plus en plus de données au même questionnaire, il est facilement possible de supporter une structure modulaire de questionnaires en créant de nouvelles sources de données pour l'entrepôt. Les opérations en cours au Laboratoire sont ainsi peu affectées durant le développement de ces modules, et il est possible de vérifier les données indépendamment et d'ensuite les intégrer pour permettre leur exploitation.

4.4.2 Plateforme de diagnostic

La création de l'entrepôt de données fait partie d'une vision à plus long terme au Laboratoire. Le système PDG permet de faire du *benchmarking* pour les PME en utilisant des données provenant de questionnaires conçus pour cet usage. Avec l'entrepôt, les données du questionnaire deviennent plus facilement accessibles pour créer de nouveaux outils, et l'ajout de nouvelles sources de données (questionnaires imprimés ou Web, base de données commerciales, etc.) permet de créer de nouvelles méthodes de diagnostic. Ces méthodes peuvent correspondre à différentes combinaisons de graphiques et de commentaires, et elles ont accès à toutes les données de l'entrepôt, d'où la plateforme de diagnostic. Il est envisagé de créer des logiciels permettant de combiner facilement de nombreuses méthodes de *benchmarking* et de les rendre accessibles sous forme de rapports facilement modifiables par l'utilisateur. Il serait alors très facile de créer des solutions automatisées qui répondent aux besoins changeants des entreprises qui utilisent le *benchmarking*.

4.4.3 Ajout d'intelligence artificielle aux rapports

Pour le système de *benchmarking*, on a besoin de plus qu'un simple système expert ; il faut un système qui peut apprendre à partir des diagnostics posés par les chercheurs, et corriger ses propres suggestions. Les applications qui peuvent bénéficier des techniques d'intelligence artificielle sont nombreuses : la préparation

de commentaires automatiques, le choix des entreprises dans un groupe témoin utilisé dans les statistiques, la sélection de graphiques pertinents pour le dossier en cours, etc. Certaines expérimentations avec des systèmes experts à base de règles ont déjà eu lieu, mais les résultats n'ont pas été à la hauteur des attentes. Les diagnostics posés doivent constamment être révisés et corrigés à la main, et ces corrections ne sont pas introduites dans le système expert actuel puisqu'il n'a pas été conçu pour permettre cette possibilité. Un système qui utilise l'intelligence artificielle et qui est conçu pour apprendre à partir des corrections faites est donc une évolution naturelle du système actuel de *benchmarking*. Plusieurs caractéristiques du système sont à l'étude, et plusieurs projets qui utilisent l'entrepôt pour leurs besoins en données sont envisagés. L'entrepôt a donc été conçu afin de pouvoir supporter ces nouvelles utilisations des données, tout en continuant de supporter les applications qui sont déjà en fonction.

CHAPITRE 5

DISCUSSION ET INTERPRÉTATION DES RÉSULTATS

Le processus de création de l'entrepôt est basé sur des objectifs très pratiques et doit répondre à certains critères qui ont été énoncés lors d'une étude sur les améliorations à apporter au système d'information du PDG (Dugré et Delisle, 2003). Les objectifs principaux ont été atteints (voir la section 1.2), et sont expliqués plus en détail dans ce chapitre. Le forage de données est possible, et l'entrepôt est prêt. Pour l'instant, peu de résultats numériques sont disponibles. Il faudra encore attendre la concrétisation de plusieurs autres projets qui sont déjà prévus pour l'amélioration du système PDG pour avoir des comparaisons précises. Certains résultats concrets sont tout de même présentés.

5.1 Avantage de la structuration sous forme d'entrepôt

L'utilisation d'un entrepôt de données permet de régler plusieurs problèmes qui sont déjà présents au LaRePE. Puisque beaucoup de temps a été investi dans le système d'information du Laboratoire et le rapport de *benchmarking* PDG, il n'était pas jugé intéressant de tout changer immédiatement, du moins pas pour toutes les parties du système en même temps. L'entrepôt est un pont entre les utilisations classiques (statistiques, ancien modèle du rapport de *benchmarking*), et les futures utilisations des données (*OLAP*, forage de données, rapports Web, etc.). D'ailleurs, l'entrepôt a déjà commencé à remplir ces mandats. D'autres avantages sont énoncés dans les sous-sections suivantes, à partir de l'analyse préparée au printemps 2003 (Dugré et Delisle, 2003) afin de fournir des choix éclairés pour les améliorations futures du système de *benchmarking*.

5.2 Comparaison avant-après

L'entrepôt permet de régler plusieurs problèmes (voir la sous-section 3.1.3) qui devenaient de plus en plus nocifs à l'entretien du système d'information du LaRePE. Certains problèmes sont réglés directement par le nouveau système. Par contre, certains problèmes ne découlent pas directement de la structure des données mais plutôt des utilitaires qui utilisent ces données, comme les questionnaires ou les rapports. Ces problèmes devront être réglés durant les phases suivantes du remplacement de l'ancien système. Les concepts utilisés pour le développement de l'ancien système et sa performance laissaient parfois à désirer. L'entrepôt permet déjà d'en améliorer certains éléments, mais il reste encore du chemin à faire dans les prochaines phases. L'entrepôt permet de conserver des informations pour la documentation de plusieurs aspects des données. Il donne aussi accès à de nouveaux outils qui permettent de concilier les différents processus d'utilisation des données. Des gains en performance d'exécution peuvent déjà être remarqués dans quelques cas. Il est aussi maintenant possible de créer des jeux de données à partir de l'entrepôt en utilisant l'état des sources de données *historisées*, ce qui était parfois demandé avec l'ancien système mais difficilement réalisable.

5.2.1 Les problèmes réglés

Le manque d'encapsulation et de modularité dans la programmation est un problème qui provient en partie de l'utilisation de langages de programmation très facile à apprendre, comme Visual Basic, ce qui permet de développer des systèmes très rapidement mais, malheureusement, souvent un peu n'importe comment, surtout si ces personnes ne sont pas des développeurs avertis. Ceci n'est pas un inconvénient pour des petits utilitaires *jetables*, c'est-à-dire à très courte durée de vie. Cependant, pour un logiciel aussi complexe qu'un système d'information relié à divers composants de production, une méthode de conception désorganisée nuit au développement et à l'entretien. Avec l'entrepôt, des modèles de conception utilisent des objets pour communiquer, comme la notion de variable ou de jeu de données. Cette encapsulation permet de développer des applications indépendamment de l'implémentation de chaque concept. L'encapsulation permet d'ailleurs de régler un autre problème, soit l'absence d'une méthode flexible et naturelle pour échanger des données entre les logiciels. Il est alors possible de déléguer le développement de nouvelles composantes sans que les programmeurs aient nécessairement besoin de connaître parfaitement toutes les particularités techniques des systèmes en cause. D'ailleurs, il existe de nouveaux standards d'échange de données pour les applications Web, comme *SOAP*[15], un standard qui fonctionne en *XML* et qui pourrait facilement être ajouté à l'entrepôt avec les applications Web déjà développées.

L'absence de documentation sur l'ancien système est un problème depuis longtemps. De nombreux changements sont survenus, et les personnes qui développent une partie du système d'information restent rarement plus de quelques années étant donné le contexte universitaire dans lequel le LaRePE fonctionne. C'est normal dans le contexte d'un laboratoire de recherche qui engage des étudiants (voir la section 1.1). C'est pourquoi, dans le cadre de la conception de l'entrepôt, une documentation d'envergure a été nécessaire, et est encore en cours. La documentation des liens et des dépendances entre tous les modules qui supportent le rapport de *benchmarking* PDG est nécessaire pour permettre de connecter l'entrepôt aux divers systèmes du LaRePE. Cette documentation a été entreprise avant la conception de l'entrepôt (Dugré et Delisle, 2003), et sera complétée en partie avec le prochain logiciel de production des rapports. Un autre élément essentiel est la création d'une documentation sur les variables. La structure supportant cette documentation est prête, c'est le dictionnaire de variables (voir la sous-section 3.2.2). Le contenu du dictionnaire n'est pas encore complet, mais les principaux intéressés (notamment, les chercheurs) reconnaissent que ce dictionnaire pourrait leur être très utile, et ils devront encore y consacrer des ressources pour le terminer. Un des derniers aspects de la documentation du système à être amélioré est l'ajout de métadonnées pour permettre une compréhension plus rapide du système. Ces métadonnées constituent, avec le dictionnaire de variables et toutes les autres formes de documentation, un ensemble d'informations accessibles en ligne par l'entrepôt. Elles facilitent la compréhension des données dans un contexte dynamique comme celui du LaRePE.

[15] http://www.w3.org/TR/SOAP/

Des améliorations techniques apportent une nouvelle façon de voir les données dans l'entrepôt, et elles facilitent la tâche des personnes qui doivent manipuler les données. La réduction du nombre de tables nécessaires pour stocker les données (voir la sous-section 3.2.1.2) diminue le nombre de jointures qui sont faites lors de l'utilisation des données. Il est plus facile de retrouver une variable ainsi, et la performance du système est accrue. L'identification des endroits où les transformations sur les données doivent avoir lieu permet finalement de régler les conflits qui perdurent depuis des années, à savoir si le nettoyage doit être fait dans la base de données ou dans le logiciel statistique. Avec l'entrepôt, il est clair que toutes les manipulations répétitives et communes à tous les projets de recherche doivent être faites dans le processus de chargement (voir la sous-section 3.2.1.3). Les données sont alors stockées sous une forme directement utilisable pour les statistiques. La réduction du nombre d'interventions humaines nécessaires pour avoir accès aux données amène la possibilité de répondre plus rapidement à la demande. Cette capacité accrue est particulièrement utile durant certaines périodes où les projets se bousculent et où les jeux de données doivent sortir le plus rapidement possible. Le nombre réduit de manipulations diminue aussi les risques d'erreur. Les améliorations du système sont le résultat de la démarche de création d'entrepôt à partir de l'analyse de l'ancien système, et certains utilisateurs ont déjà commencé à en bénéficier.

Même avec tous les changements apportés à l'ancien système pour rendre les données plus accessibles, sans de meilleurs outils destinés aux utilisateurs finaux, tout ce travail n'aurait pas été très utile. Un des outils est un accès simplifié et plus facile par le Web pour consulter la documentation des données et permettre de créer des jeux de données. En utilisant un navigateur standard (*Netscape 7+* ou *Internet Explorer 6+*), les utilisateurs peuvent accéder aux applications mises à leur disposition. Il est alors possible d'utiliser le dictionnaire de variables (voir la sous-section 3.2.2) pour consulter les métadonnées des variables, et il est aussi possible de consulter l'aide en ligne pour avoir accès aux différentes versions des questionnaires imprimés et aussi aux informations sur les données de l'entrepôt en général. Le *Dataset Maker* (voir la section 5.3) est l'application Web qui est accessible aux personnes autorisées pour créer un jeu de données à partir de l'entrepôt. L'ajout d'autres outils accessibles sur ce site, dont *OLAP* (voir la section 5.5), permet aussi d'avoir accès plus rapidement aux données, tout en ayant plus de flexibilité qu'en utilisant l'ancienne méthode des rapports de gestion. Tous ces outils sont déjà disponibles et fonctionnels, et d'autres applications seront sûrement proposées par les utilisateurs et l'entrepôt est prêt à les supporter.

5.2.2 Gains en performance d'exécution

Avec le système de rapport de *benchmarking* basé sur *SAS* et *Excel*, plusieurs optimisations sont nécessaires pour permettre de produire un rapport en moins de 3 minutes, et ces optimisations compliquent l'entretien du système. L'ancien système fonctionne à l'aide de la combinaison de plusieurs logiciels. Il utilise une liste des variables du système pour les insérer dans SAS et les importe dans une feuille *Excel* à l'aide de plusieurs macros. Le programme *SAS* calcule des moyennes et des

médianes qui sont alors importées dans un document *Excel*. De plus, le programme *SAS* doit exécuter de nombreuses manipulations sur les variables puisque les données dans la base de données ne sont pas sous une forme utilisable directement. C'est pourquoi un programme optimisé et spécialement conçu pour le rapport doit être entretenu et modifié à chaque fois qu'il y a un ajout de variables. C'est une réalité qui force les responsables du système à dupliquer les traitements sur les variables. Un programme est optimisé pour le rapport, et un autre plus complet est utilisé pour la recherche.

Dans l'entrepôt, cette duplication et les traitements supplémentaires ne sont plus nécessaires. En effet, comme il est démontré à la sous-section 5.4, il est possible de relier le système de rapport à l'entrepôt pour améliorer les performances, et ce dernier effectue la majorité des étapes de transformation des données nécessaires aux rapports. La seule étape qui n'est pas effectuée est le calcul des statistiques descriptives (moyennes, médianes, etc.), et ces calculs peuvent facilement être ajoutés à un programme de préparation des données pour les rapports. Ce programme est pour l'instant exécuté par *SAS*, mais il serait possible et avantageux d'ajouter un nouveau magasin de données préparant toutes les données pour les rapports, ce qui permettrait de créer un rapport avec un minimum de calculs (voir la sous-section 6.3.2).

Le gain en performance le plus impressionnant est cependant remarqué dans l'application de création de jeux de données, le *Dataset Maker*. C'est en effet cette application Web qui permet aux utilisateurs de créer leur propre jeu de données personnalisé en profitant de tous les calculs effectués à l'avance dans l'entrepôt. Contrairement au programme *SAS* qui doit refaire le calcul de toutes les entreprises lors d'une mise à jour, l'entrepôt maintien ses enregistrements avec des étapes de chargement qui calculent de manière incrémentale la majorité des données. De cette façon, non seulement la phase de mise à jour de l'entrepôt est courte (de l'ordre de 5 minutes, comparativement à 45 minutes pour le programme *SAS*), mais les données sont optimisées pour permettre une récupération rapide d'un jeu de données à jour (environ 30 secondes pour les 800 variables générales et 56 variables financières sur 6 ans). La mise à jour de l'entrepôt est d'ailleurs effectuée la nuit, les utilisateurs n'ont donc jamais à attendre la mise à jour des données lorsqu'ils en ont besoin. Un utilisateur est maintenant capable de se créer un jeu de données en aussi peu que 5 minutes, s'il connaît déjà les variables dont il a besoin. Par rapport au mode précédent de fonctionnement (avec un intervenant, ce qui pouvait prendre plusieurs jours), c'est une amélioration très considérable.

5.2.3 Version historique des données

Il est maintenant possible de conserver différentes versions des données pour recréer ou compléter des jeux de données avec des données telles qu'elles étaient à un moment précis de l'histoire de l'entrepôt. Cette fonctionnalité était depuis longtemps simulée à la main par la personne chargée des statistiques au Laboratoire à l'aide de nombreuses manipulations de données. Elle a été présentée lors d'une réunion aux chercheurs utilisant les données du LaRePE. Un des points qui est ressorti de cette

rencontre est que les chercheurs ne semblaient pas voir l'utilité de recréer un jeu de données de cette façon, alors que cette demande avait justement été faite à plusieurs reprises dans le passé, lors de recherches. Cette fonctionnalité s'est tout de même retrouvée dans la version finale de l'entrepôt, puisque son utilité a déjà été démontrée dans le passé. D'ailleurs, il est ensuite beaucoup plus facile de modifier les données selon les besoins des chercheurs, en mode historique ou non.

5.3 Utilisation pour la recherche

La création de jeux de données pour la recherche est devenue beaucoup plus facile. Une des premières réactions notées au sujet du *Dataset Maker* est un souci de la sécurité. En effet, auparavant il fallait plusieurs jours pour créer des jeux de données complets à partir des spécifications des utilisateurs. Un processus humain faisant participer plusieurs intervenants du Laboratoire était alors nécessaire, incluant une autorisation de la directrice pour faire sortir les données de la base de données, et le concours d'un professionnel de recherche pour créer ce jeu de données. Puisque le nouveau système permet de passer outre plusieurs étapes de ce processus, il est important de s'assurer que seules les personnes autorisées ont accès aux données.

Une des suggestions faites pour améliorer le système est l'instauration d'un processus de *commande* de jeux de données. Selon la suggestion, seules quelques personnes ont un accès complet aux données (la directrice du Laboratoire et un professionnel de recherche), et ce sont ces personnes qui peuvent distribuer les données. Cependant, le système permet quand même de créer un jeu de données à partir des spécifications d'autres utilisateurs. Ces utilisateurs ne peuvent cependant pas prendre possession des données sans passer par le processus humain d'autorisation de sortie des données.

On voit ainsi que la création de ce système d'information a modifié la façon de gérer les jeux de données pour la recherche. Le processus de manipulation des données a été grandement amélioré, au point de devenir presque complètement automatique. Cependant, une intervention humaine a été ajoutée au processus automatique puisqu'il faut gérer avec précaution la circulation des données du Laboratoire. Le système de droits d'accès en place ne pourrait pas avoir le même discernement qu'une personne pour s'assurer que les données ne sont pas utilisées à tort.

5.4 Création d'un rapport relié à l'entrepôt

La création d'un rapport, traditionnellement, passe par plusieurs étapes réparties sur différents logiciels. Les données sont stockées dans une base de données Oracle. Des programmes en *PL/SQL* permettent de choisir une entreprise à partir d'une liste des dossiers. Ensuite, il faut choisir les entreprises qui composeront le groupe témoin, utilisé pour le processus de *benchmarking*. Pour l'instant, ces deux étapes sont simulées dans l'entrepôt puisque le système nécessaire pour les supporter ne faisait pas partie du mandat initial de création de l'entrepôt. L'étape suivante est l'importation des données dans le programme qui calculera les statistiques descriptives. Pour l'instant, tous ces calculs sont effectués dans le logiciel *SAS*, c'est

pourquoi un programme *SAS* est relié à l'entrepôt pour importer les données. La dernière étape consiste à générer le rapport à l'aide d'un fichier *Excel*. Ce dernier récupère ses données à partir de librairies *SAS*. Voici un résumé de ces étapes :

- choisir l'entreprise ;
- créer le groupe témoin ;
- importer les données dans *SAS* (c'est ici que l'exemple est fait) ;
- calculer les résultats avec *SAS* ;
- importer dans le fichier *Excel*.

Pour l'instant, la seule amélioration qu'il est possible de noter est une diminution du temps de préparation des données puisque le calcul des taux de change et la pré-sélection des variables sont déjà effectués dans l'entrepôt. Ceci évite de calculer ces valeurs chaque fois qu'un rapport est demandé. Quelques rapports qui ont déjà été créés avec l'ancien système du PDG ont été pris au hasard, puis recréés en utilisant les données de l'entrepôt. L'exemple illustré utilise uniquement le temps d'importation des données pour les calculs à faire dans *SAS*, avant de produire le rapport à l'aide d'*Excel*. Donc, c'est seulement la partie où les données sont effectivement importées de la base de données ou de l'entrepôt dans *SAS* qui est représentée. Puisque les autres parties touchant la création du rapport n'ont pas encore été modifiées pour profiter de l'entrepôt, leur temps n'a pas été altéré par la procédure, ils ne sont pas retenus dans l'exemple. La vérification de l'amélioration du temps d'importation des données n'est pas très précise, le but est simplement de montrer qu'il y a eu une amélioration dans la majorité des cas.

Pour cet exemple, 13 rapports créés avec le système de production du PDG utilisant la base de données manufacturière sont repris avec une méthode qui permet d'importer les données directement de l'entrepôt. L'hypothèse est que l'entrepôt permet d'importer les données plus rapidement pour les raisons suivantes :

1. Certains calculs (voir la sous-section 3.2.1.3) sont déjà prêts dans l'entrepôt, alors qu'ils doivent être exécutés sur demande à l'aide de vues dans la base de données manufacturière.
2. L'entrepôt utilise une seule table générale, alors que la base de données manufacturière utilise une vue qui fait une jointure de 45 tables pour pouvoir importer les données dans *SAS*.

Des exemples de temps de calcul dans *SAS* sont illustrés dans les figures 23 et 24, et les résultats de la création des rapports sont affichés dans le tableau 2. Les résultats peuvent varier pour plusieurs raisons et ne doivent pas être considérés comme les seuls résultats possibles. Une de ces raisons est que l'entrepôt et la base de données manufacturière fonctionnent dans le même gestionnaire Oracle, ce qui fait que l'antémémoire (*cache*) est partagée. Si un des schémas effectue plus d'appels de lectures de données que l'autre, ce dernier sera pénalisé, car l'antémémoire est limitée et retient seulement les données qui sont lues le plus souvent. Il faut aussi tenir compte de l'utilisation générale du serveur, qui est utilisé par l'ensemble de

l'UQTR pour diverses applications, dont le logiciel Oracle. Un processus de création de rapports est simulé pour extraire les données.

```
113
114  disconnect from oracle;
115
116  quit;
NOTE: PROCEDURE SQL used:
      real time           1:18.19
      cpu time            0.98 seconds
```

Figure 23 Utilisation de la base de données pour créer un rapport PDG

```
5062
5063  disconnect from oracle;
5064
5065  quit;
NOTE: PROCEDURE SQL used:
      real time           6.74 seconds
      cpu time            0.19 seconds
```

Figure 24 Utilisation de l'entrepôt de données pour créer un rapport PDG

Le tableau 2 affiche les temps d'extraction des données à partir de la base de données manufacturière et de l'entrepôt pour quelques rapports. Le numéro de rapport est un numéro arbitraire utilisé pour accéder aux données pour les rapports. Le nombre d'entreprises est déterminé par le groupe témoin sélectionné par des experts du Laboratoire. Le temps en secondes pour chaque jeu de données utilisé pour les rapports est affiché pour la base de données et pour l'entrepôt.

Tableau 2

Les résultats d'une comparaison entre les performances de la base de données manufacturière et l'entrepôt de données du LaRePE pour la préparation de données pour le rapport PDG

Rapport	Nombre d'entreprises dans le rapport	Base de données manufacturière (secondes)	Entrepôt du LaRePE (secondes)
1	14	22,09	3,77
2	15	22,98	3.54
3	42	23,87	4,68
4	42	25,69	6,30
5	15	22,21	3,34
6	16	20,52	3,46
7	42	25,45	3,88
8	42	23,40	4.10
9	32	21,93	4.28
10	32	31,75	3,82
11	21	25,22	8,83
12	17	25,93	3,66

Il est évident que l'entrepôt est plus performant que la base de données, il réussit à extraire ses données plus rapidement en conservant le résultat des calculs à l'avance. La moyenne de temps de création pour la base de données est de 24,25 secondes, tandis que l'entrepôt prend en moyenne 4,47 secondes. L'entrepôt est donc en moyenne 5 fois plus vite que la base de données pour préparer les données du rapport PDG. Il sera d'autant plus avantageux de préparer un magasin de données avec une vue matérialisée pour trier à l'avance les enregistrements, travail actuellement fait par SAS dans tous les cas, et de calculer les ratios utilisés dans les rapports pour permettre de se passer complètement de *SAS* comme intermédiaire de calcul.

L'entrepôt permet maintenant la création de nouveaux programmes qui utiliseront les données déjà disponibles et qui engloberont toutes les étapes de préparation du rapport, incluant le calcul des statistiques. La forme finale du rapport n'a pas été décidée, mais elle devra inclure une version qui peut être générée et affichée sur le Web. C'est pourquoi le temps de calcul devient très critique, et il faut absolument le faire passer de trois minutes avec le système *Excel* à quelques secondes pour cette version Web. Dans les meilleurs cas, les rapports ont pu être créés en moins de 80 secondes en utilisant l'ancien système. On vient de voir que la simple importation des données pouvait bénéficier de l'entrepôt de données. En terminant le travail et en profitant pleinement de l'entrepôt de données, il devrait être possible de créer un rapport pouvant s'exécuter en quelques secondes seulement.

5.5 Différence dans l'accès aux données grâce à OLAP

Un des désavantages des bases de données relationnelles est qu'il est difficile pour un utilisateur non expérimenté en programmation de retrouver les données qui pourraient lui être utiles. Un tel utilisateur se contente généralement des rapports qui sont créés par les programmeurs, sans savoir comment accéder aux données à l'aide de SQL pour faire des requêtes *ad hoc*. Par l'ajout d'un système simple de manipulation des données, on arrive à libérer les programmeurs des requêtes sporadiques en information, et les utilisateurs peuvent aller beaucoup plus loin dans leurs explorations des données. C'est ce qui est visé dans l'utilisation du logiciel *OLAP* inclus dans l'entrepôt du LaRePE. De nombreuses requêtes simples sont demandées, et elles doivent parfois être calculées à la main faute de structures informatiques adaptées. Ces requêtes exigent le concours d'un utilisateur qui a un besoin particulier et d'un programmeur pour extraire les données. Souvent, ces besoins en données ne sont guère plus que des interrogations de fréquences des questionnaires par rapport à une question précise. Ce genre de requête est simple à traiter grâce aux nouveaux outils de l'entrepôt. Pour l'instant, seuls quelques utilisateurs ont fait l'essai du logiciel *ContourCube* mis à leur disposition, mais les commentaires sont positifs et des requêtes d'ajouts de nouvelles données ou dimensions ont déjà été formulées.

Deux exemples d'utilisation de *ContourCube* avec l'entrepôt de données du LaRePE sont maintenant présentés. Avec ce logiciel, les dimensions disponibles sont affichées sous la forme d'une barre au-dessus du tableau où sont affichées les valeurs. Il faut noter que pour préserver la confidentialité des données, tous les nombres ont été retirés du tableau. Les titres des colonnes et des rangées sont toutefois les véritables informations affichées à un utilisateur du système. La figure 25 illustre l'utilisation des dimensions *Année*, *État du questionnaire* et *Région* sur la table de faits des questionnaires. La figure 26 montre les dimensions *Pays* et *Participation* sur la table de faits des entreprises.

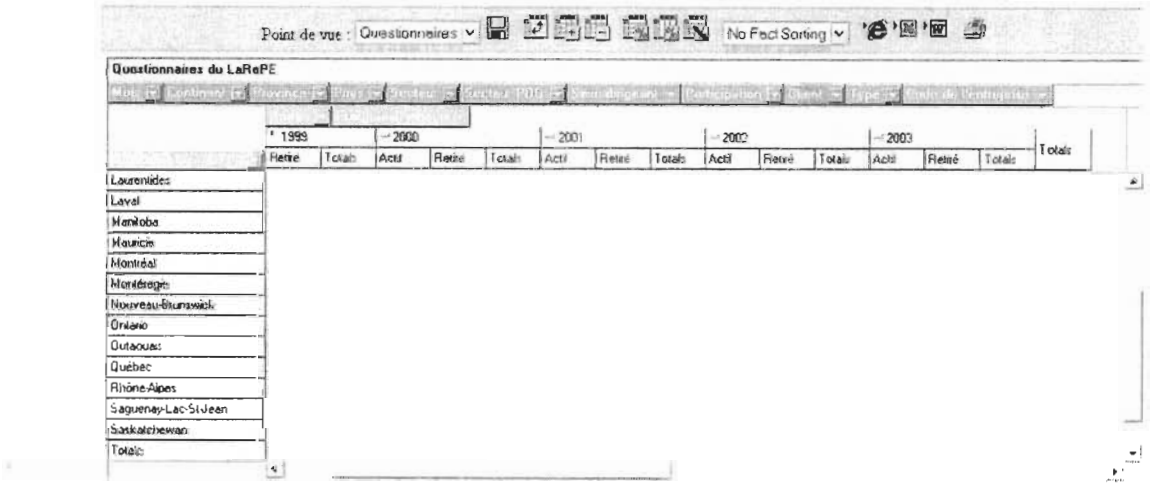


Figure 25 Exemple d'utilisation de ContourCube avec l'entrepôt, table de faits des questionnaires

Figure 26 Utilisation de ContourCube avec la table de faits des entreprises

Bien que les utilisateurs ne se soient pas encore parfaitement familiarisés avec les nouvelles possibilités offertes par *OLAP*, les outils permettent déjà de répondre à de nombreuses questions au sujet des données. Ce logiciel permet même d'exporter les tableaux sous forme de fichier *Excel* pour que ceux qui sont plus à l'aise dans un tableur puissent faire différentes manipulations. La prochaine étape consiste donc à former les divers utilisateurs à ce nouvel outil.

5.6 Utilisation du forage de données

Avec tout le travail qui a été effectué sur l'automatisation des étapes pour accéder aux données, sur la documentation des variables et du système en général, l'entrepôt est maintenant prêt à supporter des logiciels de forage de données. Ce travail d'automatisation des manipulations était nécessaire parce que de nombreux calculs devaient auparavant être effectués dans des logiciels spécialisés, les données ne pouvaient donc pas être acheminées directement à un utilisateur. C'est un processus qui aurait considérablement réduit les avantages du forage de données, qui doit préférablement avoir accès à toutes les données disponibles. Il aurait été très difficile de supporter un processus d'analyse poussé, surtout s'il était souhaitable de permettre une mise à jour régulière des données puisque de nombreux contretemps auraient été introduits dans ce processus. De plus, la documentation des variables était minime, et le forage de données a une grande dépendance envers l'exactitude et la précision des métadonnées (voir la sous-section 2.2.2). La création de l'entrepôt permet alors de supporter directement le forage de données au Laboratoire.

Puisque l'ajout de logiciels de forage de données est très récent, les utilisateurs n'ont pas encore reçu de formation et il n'y a pas encore de résultats de projets de recherche par rapport à l'utilisation du forage de données avec l'entrepôt. Cependant, plusieurs logiciels sont à la disposition des utilisateurs et sont prêts à servir pour accéder aux données. Une démonstration de l'utilisation d'un jeu de données produit avec *SAS Enterprise Miner* est maintenant présentée. Cet exemple sert surtout à montrer qu'il est possible d'importer les données dans *Enterprise Miner*, aucune analyse poussée utilisant des algorithmes de forage de données n'est faite puisque ce travail devrait être laissé à des analystes et chercheurs qui comprennent bien les données. Une utilisation aveugle d'un outil, peu importe sa nature, est une perte de temps et ne prouve rien. C'est pourquoi l'auteur de ce mémoire laisse à de vrais chercheurs en gestion et en finance le rôle d'exploiter pleinement ces nouvelles possibilités. Ce sont d'ailleurs les orientations futures de plusieurs nouveaux projets au LaRePE pour les prochains mois.

Un jeu de données a préalablement été préparé à l'aide du *Dataset Maker* (voir la sous-section 3.2.3). Ce jeu de données a été arbitrairement nommé *COMPAR1*. La figure 27 montre un nouveau projet dans *SAS Enterprise Miner*, nommé *ProjetLaRePE*. Ce projet comprend le diagramme *LaRePE1*, et ce diagramme est affiché dans la section de droite de la figure. Les éléments de programmation représentés graphiquement dans le diagramme de *SAS Enterprise Miner* (comme un jeu de données, ou un algorithme de forage de données) sont appelés des *nodes*. Le jeu de données *COMPAR1* y est affiché (figure 27), ainsi qu'une *node Insight* qui sert à afficher quelques informations de base du jeu de données. Par exemple, on peut voir l'ensemble des champs et des données du jeu de données (figure 28), et d'autres informations comme des statistiques descriptives (fréquences, moyennes, etc.) peuvent aussi être affichées.

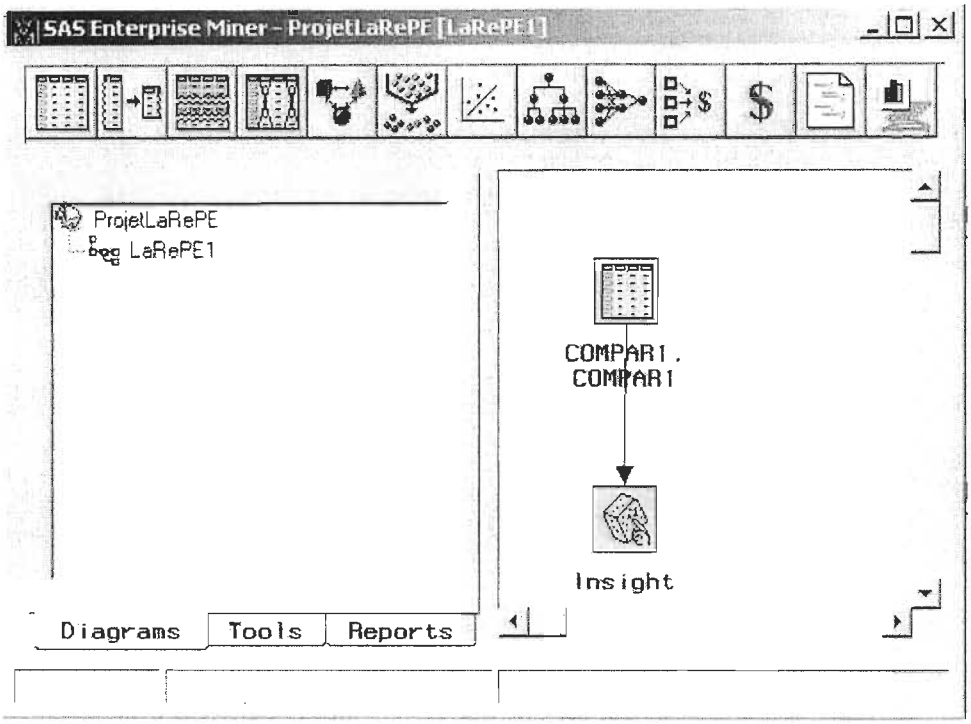


Figure 27 Un projet de forage de données avec des données provenant du Dataset Maker

On peut noter que dans la figure 28, les métadonnées qui proviennent du dictionnaire de variables sont récupérées par *SAS Enterprise Miner*. Ces données sont utilisées pour décrire chaque champ disponible pour le forage de données.

Input Data Source

Data | Variables | Interval Variables | Class Variables | Notes

Name	Type	Format	Informat	Variable Label
COD_ESE	char	$8.	$8.	CODE IDENTIFIANT L'ENTREPR
ANNEE	num	BEST12.	12.	
BUDFO	num	DOLLAR10.	10.	BUDGET FORMATION ANNÉE DER
EMTOT	num	COMMA8.2	8.	NOMBRE TOTAL D'EMPLOYÉ DAN
ETUDE	num	ETUDECLA25.	25.	NIVEAU DE SCOLARITÉ DU DIR
MARGCRED	num	NONOUI25.	25.	ENTREPRISE POSSÈDE MARGE D
RECOM	num	ACCORCLA25.	25.	ENTREPREN RECOMMENDERAIT B
CARECNOW	num	DOLLAR13.	13.	COMPTES À RECEVOIR (DÉBITE
CARECIAN	num	DOLLAR13.	13.	COMPTES À RECEVOIR (DÉBITE
VENTENOW	num	DOLLAR13.	13.	VENTES (CHIFFRE D'AFFAIRES
VENTEIAN	num	DOLLAR13.	13.	VENTES (CHIFFRE D'AFFAIRES

Figure 28 SAS Enterprise Miner récupère les métadonnées transmises par le Dataset Maker

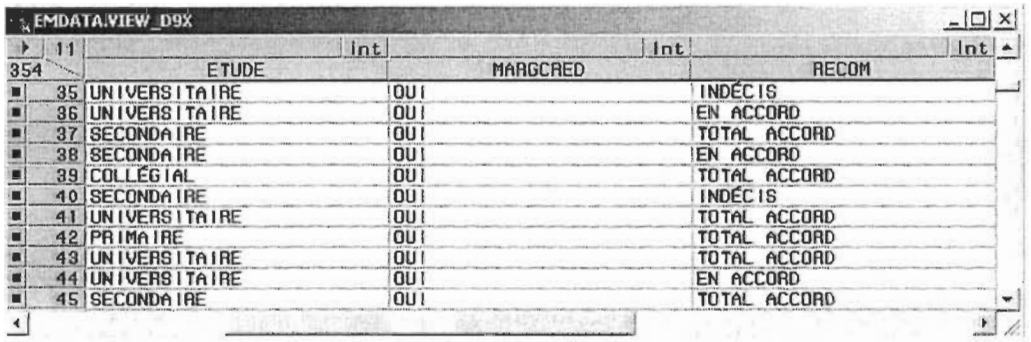


Figure 29 Affichage du jeu de données avec la node Insight de SAS Enterprise Miner

Un projet de forage de données pourrait alors être lancé à partir de cette base (voir figure 29). Il faudrait commencer par nettoyer les données et identifier le but du projet. On pourrait par exemple prendre les variables VENTENOW, VENTE1AN et VENTE2AN pour calculer un taux de croissance. Le nettoyage de ces données pourrait alors consister à déterminer la façon de traiter les entreprises avec des données manquantes. On pourrait choisir de calculer le taux de croissance pour les entreprises avec toutes les données nécessaires, puis d'utiliser la moyenne du taux de ces entreprises pour celles où trop d'années sont manquantes. On pourrait aussi décider de retirer les entreprises qui n'ont pas suffisamment d'informations pour donner un taux correct (selon l'expert). Il peut aussi être nécessaire de gérer les entreprises avec des données qui sont à l'extérieur des valeurs jugées correctes (*outliers*), toujours selon l'expert. Les méthodes utilisées pour le nettoyage sont laissées à la discrétion de l'expert qui effectue l'analyse. Ensuite, toujours selon l'exemple, il est possible d'appliquer des algorithmes de forage de données pour vérifier certaines informations ou tenter d'extraire des informations (ou connaissances) au niveau de la croissance des ventes des entreprises.

CHAPITRE 6

TRAVAUX FUTURS

L'entrepôt préparé dans le cadre de ce mémoire fait partie de la réingénierie d'un système d'information existant pour l'adapter à des besoins qui évoluent constamment. Cet entrepôt représente la première phase de cette transformation, et d'autres phases sont déjà prévues. La création d'un nouveau questionnaire et la préparation d'un système modulaire pour les rapports sont en évaluation. Plusieurs suggestions sont faites dans ce chapitre afin de guider les travaux futurs sur le nouveau système.

6.1 Améliorations souhaitables

Le travail effectué lors de la création de l'entrepôt de données du LaRePE est le résultat de plusieurs années de développement et d'un processus plus récent de réflexion et de questionnements sur le système de rapport PDG. Mais l'entrepôt est au service de ses utilisateurs, et il y a de nombreux points qui devront être améliorés pour mieux répondre aux besoins. Quelques-uns de ces points sont des fonctionnalités qui ont déjà été demandées, mais qui, pour diverses raisons, ne se retrouvent pas encore dans la version fonctionnelle de l'entrepôt. D'autres points seront apportés par les utilisateurs qui commenceront à utiliser de plus en plus les diverses possibilités de l'entrepôt. Une des notions qu'il faut parfois rappeler aux utilisateurs est qu'un système d'information est à leur service. Si ce système est encombrant, peu convivial ou qu'il manque de fonctionnalités, les concepteurs ne sont pas nécessairement ceux qui en souffriront le plus. C'est la responsabilité de l'utilisateur de faire part de ces défauts ou de ces manques, et c'est aux responsables de l'entretien de faire de leur mieux pour répondre aux besoins des utilisateurs.

Cette façon de penser doit absolument être appliquée à l'entrepôt du LaRePE pour qu'il devienne un élément productif et utile à tous les chercheurs et les assistants qui s'en servent. Un outil aussi puissant doit supporter toutes les nouvelles exigences pour concevoir des solutions pour les nouveaux systèmes développés au Laboratoire. C'est pourquoi certaines suggestions sont déjà présentées dans le reste de ce chapitre, et que toute nouvelle idée devrait être notée afin de permettre l'extension de l'entrepôt.

6.1.1 Dataset Maker

La création du *Dataset Maker* s'est faite à partir de l'expérience d'utilisation des données qui était déjà présente au Laboratoire. Cette expérience est basée sur une méthode manuelle qui peut prendre plusieurs jours pour créer un jeu de données. La nouvelle façon d'extraire des données avec le *Dataset Maker* promet de changer les habitudes et de créer de nouveaux besoins. Une fonctionnalité qui devra être ajoutée est la possibilité de distinguer les variables à partir des sources de données d'où elles proviennent (le questionnaire manufacturier, un module supplémentaire comme Acier Plus, etc.) pour les insérer encore plus rapidement dans l'entrepôt. Ainsi, l'ajout de nouveaux modules à l'entrepôt ne nécessitera aucune modification du *Dataset Maker*. Les utilisateurs seront d'ailleurs plus à même d'exploiter les nouvelles sources de données accessibles, et ce, beaucoup plus vite. Chaque source de données et les variables associées doivent être documentées pour permettre aux

utilisateurs de s'y retrouver. Une autre fonctionnalité souhaitable est le support de certaines des variables qui sont calculées dans les logiciels statistiques. Les plus faciles à supporter sont les ratios et les classes obtenues à partir de variables numériques (voir la sous-section 3.1.2). Certaines des valeurs calculées pourraient utiliser des paramètres, comme le nombre de classes à créer pour une variable numérique, ou encore les limites inférieures et supérieures de chaque classe. Mais pour ajouter le calcul des variables, il faut premièrement terminer l'inventaire des variables.

6.1.2 OLAP

L'ajout d'un logiciel *OLAP* est une nouveauté au LaRePE, et peu d'utilisateurs y ont accès pour l'instant, pour des raisons de sécurité. Cependant, il y a déjà certaines demandes qui ont été faites par rapport à la sécurité d'accès aux données. Il a été proposé que les différents utilisateurs n'aient pas nécessairement accès aux mêmes données (dimensions, faits). De cette façon, il serait possible de restreindre la navigation aux seules données utiles pour les recherches en cours. Il est certain que de nombreuses autres dimensions et d'autres faits seront proposés pour les études et les besoins internes de gestion des questionnaires. La plupart de ces utilisations devraient pouvoir être supportées à partir des structures existantes, avec un minimum de modifications.

6.1.3 Chargement des données

La sous-section 3.2.4 explique le principe du chargement des données dans l'entrepôt, et l'annexe C suggère une méthode pour ajouter de nouvelles sources de données. Par contre, si l'ajout de sources de données à partir de nouveaux projets devait se produire régulièrement (2 ou 3 fois par année), il pourrait être utile de créer une procédure automatique. Cette procédure pourrait préparer les tables et les métadonnées de ces nouvelles structures. De cette façon, il serait plus facile de former les personnes pour faire l'entretien de l'entrepôt, et le temps de développement et de jointure entre l'entrepôt et les nouvelles sources serait moins long. La sous-section 6.1.4 énonce d'autres avantages à créer un outil automatique pour entretenir les sources de données, dont la documentation plus facile de nouveaux questionnaires.

6.1.4 Ajout simplifié des différents questionnaires

Une des raisons énoncées pour la création d'un entrepôt est le support de questionnaires et de rapports plus modulaires. Cette modularité permet un temps de déploiement plus rapide durant la période d'élaboration d'un nouveau questionnaire. La préparation pour la saisie dans une base de données est plus rapide, et la création d'un rapport alimenté automatiquement à partir de ces données prend aussi moins de temps. C'est pourquoi l'entrepôt peut supporter de nombreuses sources de données et les intégrer afin d'alimenter des rapports basés sur le même modèle. Cette façon de faire modulaire permet une très grande flexibilité aux chercheurs lors de la création de nouveaux rapports, et diminue grandement le temps de développement nécessaire

pour permettre la saisie et sortir les premiers rapports. Cependant, il n'y a pas d'outils automatiques pour la documentation ou l'administration de nouvelles sources de données. Même si la structure interne de l'entrepôt facilite les ajouts à plusieurs niveaux, il faut quand même tout faire à la main. Il pourrait être très utile de créer de tels outils pour préparer les tables et les étapes de chargement afin d'acheminer les nouvelles données dans l'entrepôt. Il serait alors possible d'assurer que les données ajoutées sont conformes à celles déjà présentes, une documentation semi-automatique des nouvelles variables pourrait suivre, ainsi que la mise en place du processus de mise à jour. De tels outils permettraient d'assurer une bonne cohésion entre la documentation de l'entrepôt et les données qui sont effectivement disponibles, ainsi qu'une disponibilité plus rapide des données pour les rapports.

6.1.5 Normalisation des tables

La normalisation des tables du système PDG (base de données manufacturière, et maintenant l'entrepôt) est une notion qui est revenue à plusieurs reprises dans le passé. Une des raisons qui fait que cette normalisation serait très utile est l'existence des questionnaires de mise à jour des entreprises. Ces questionnaires sont une version réduite du questionnaire de base (celui utilisé la première année). Puisque certaines questions ne sont pas présentes dans chaque version du questionnaire, il serait possible de réduire les problèmes de suivi en s'entendant sur une structure normalisée de tables pour stocker les informations. Mais cette solution ne règle pas tous les problèmes. Puisque les questionnaires changent souvent, une version presque totalement dénormalisée, comme ce qui est utilisé dans le magasin historique, est probablement la meilleure solution pour supporter à long terme les prochains questionnaires. Il faut cependant s'assurer d'avoir des codes prédéterminés pour identifier les champs, à savoir si l'entrepreneur a reçu un questionnaire qui lui permettait de répondre à une question particulière ou non (les versions des questionnaires n'ont pas toutes les mêmes questions). Comme on le voit, les problèmes de structure des tables pour les questionnaires ne sont pas simples.

Une normalisation dans la base de données de saisie serait probablement plus utile qu'une normalisation dans l'entrepôt. D'ailleurs, dans un entrepôt, il est parfois conseillé de dénormaliser afin d'accélérer les traitements des enregistrements. Puisque la structure de la base de données est déjà dénormalisée, il n'y a aucun gain en performance dans l'entrepôt dans ce cas. Cependant, la base de données qui sert de support à la saisie des questionnaires pourrait tirer profit de la normalisation de ses tables. Il pourrait d'ailleurs s'agir d'une normalisation qui se base non seulement sur les codes des entreprises, mais aussi sur les versions des questionnaires. Un travail de révision du questionnaire est déjà prévu, et toutes ces considérations feront l'objet de réflexions futures.

6.1.6 Dictionnaire

Le dictionnaire est un élément nouveau ajouté aux fonctionnalités de l'entrepôt pour répondre aux besoins des utilisateurs. Il fait partie des métadonnées accessibles aux utilisateurs pour les guider dans le contenu de l'entrepôt. Comme le dictionnaire est une nouveauté, les utilisateurs n'ont pas encore eu le temps de faire beaucoup de commentaires. Il faudra attendre un certain temps qu'un accès à plus grande échelle soit possible pour connaître les besoins qu'il reste à combler. De plus, certains aspects du dictionnaire restent encore à terminer, toutes les données n'ont pas encore été insérées puisque la documentation complète des variables n'est pas terminée.

Un des aspects les plus désirables et qui revient le plus souvent est la nécessité d'une classification des variables par thèmes. Cette classification pourrait être accessible directement à partir du moteur de recherche du dictionnaire et permettrait de rechercher les variables en choisissant un domaine d'intérêt. Par exemple les ressources humaines, l'innovation, la collaboration, etc. Cette classification doit cependant être faite par des personnes connaissant bien les variables et chacun des domaines. Le problème de la création d'une classification n'est donc pas vraiment du domaine technique, mais surtout du besoin de ressources qui pourront faire le travail initial de classification. Il faudra assigner des assistants à ce travail fastidieux, mais tout le monde s'entend pour dire qu'une classification des variables serait très utile.

Le dictionnaire pourrait aussi être amélioré en ajoutant des champs de recherche en fonction de besoins des utilisateurs. Seuls quelques critères sont disponibles en ce moment, et il est certain que les utilisateurs en réclameront d'autres, en fonction de leurs besoins. Il faudra faire une liste de ces demandes et les ajouter au moteur de recherche. Finalement, l'ajout de traitements pour les formules des variables calculées pourrait se faire dans le dictionnaire. Un embryon d'analyseur syntaxique est déjà présent dans le dictionnaire, mais il n'a pas été complété puisque les calculs avec des formules ne sont pas encore actifs. Cet analyseur syntaxique pourrait servir à vérifier les formules saisies et à afficher des liens supplémentaires lors de la consultation des variables calculées, dans le dictionnaire. Il sera évidemment utile de pouvoir calculer les valeurs des variables lors de la création des nouveaux rapports, dans les phases subséquentes du développement du nouveau système (voir la section 6.3).

6.2 Optimisation de l'entrepôt

L'entrepôt du LaRePE ne bénéficie pas encore de toutes les optimisations possibles avec *Oracle 8i*. Au départ, c'est le logiciel *Oracle 9i* qui était prévu pour supporter l'entrepôt, et ce logiciel dispose de plusieurs méthodes permettant d'optimiser encore plus l'exécution du code et les requêtes aux entrepôts de données. *Oracle 9i* n'était pas disponible sur les serveurs de l'UQTR au moment de la conception de l'entrepôt, et la faible taille des tables de faits ne justifie pas une optimisation poussée. Ce travail peut être remis à une date ultérieure, lorsque la transition déjà en cours vers *Oracle 9i* ou *10g* sera complétée.

Par contre, de nouveaux projets alimenteront l'entrepôt en données et il pourra devenir avantageux d'effectuer ces optimisations lors de l'entretien des tables de faits actuelles, ou même lors de la création de nouvelles tables. *Oracle 9i* supporte de nombreuses méthodes qui favorisent la réécriture automatique des requêtes à l'aide de vues matérialisées et d'indices adaptés aux entrepôts de données. Il n'est pas encore possible de donner d'exemples de vues matérialisées. Le fonctionnement des applications exploitant directement les données de l'entrepôt, comme le rapport, fait partie des travaux en cours et aucun choix définitif n'est encore fait. Mais ce travail d'optimisation pourrait faire l'objet d'un prochain projet spécifiquement orienté sur l'accélération des requêtes existantes dans l'entrepôt. De plus, lorsqu'il sera appelé à supporter la création régulière de rapports, l'entrepôt devra permettre une bonne vitesse dans les calculs de ces requêtes. Des solutions possibles sont la création de nouveaux magasins de données spécialisés. et l'optimisation du stockage et de l'accès aux données.

6.2.1 Les vues matérialisées

Les vues matérialisées sont un aspect très important dans l'optimisation des requêtes d'un entrepôt. *Oracle 9i* propose de nombreuses améliorations à la préparation et à l'entretien de ces vues. C'est une des premières modifications qui devrait être apportée à l'entrepôt lorsque la nouvelle version d'*Oracle* sera prête. Ces vues permettent de créer rapidement de nouveaux magasins de données en facilitant la distribution des données. De plus, comme il est mentionné ci-haut, *Oracle 9i* dispose de nombreux mécanismes, dont la réécriture des requêtes faite à l'interne par des algorithmes d'optimisation, pour accélérer les requêtes et traitements demandés par les utilisateurs de façon *ad hoc*. L'utilisation de vues matérialisées avec mise à jour rapide (*fast refresh*) pourrait être particulièrement avantageuse.

6.2.2 Utilisation des ressources d'*Oracle 9i* et *10g* pour base de données dimensionnelle

Les nouvelles versions des bases de données *Oracle* ont été optimisées pour gérer de très grands entrepôts de données et faciliter les différentes opérations liées aux principes de *business intelligence*. *Oracle 9i* augmente le nombre d'opérateurs disponibles pour exécuter des opérations liées à *OLAP*, et favorise une plus grande distribution des données sous forme de clusters[16]. De son côté, la version *Oracle 10g* ajoute aux éléments d'*Oracle 9i* des structures spécifiques au forage de données. Les magasins de données sont implémentés directement par le serveur et des algorithmes de forage de données sont disponibles sans avoir à ajouter de modules supplémentaires[17]. De plus, de nouveaux outils ont été ajoutés pour exécuter directement les phases de chargement (*Extract, Transform and Load*) dans cette plus récente version du logiciel *Oracle*, ce qui permet de faciliter la création de nouvelles sources de données. Malheureusement, l'utilisation de ces nouvelles ressources n'a pas été possible au moment de créer l'entrepôt du LaRePE.

---

[16] http://otn.oracle.com/products/oracle9i/index.html

[17] http://otn.oracle.com/products/bi/index.html

6.3 Ajouts prévus

Comme il a été mentionné plusieurs fois dans ce mémoire, la création de l'entrepôt n'est que la première phase d'un projet de réingénierie en profondeur du système d'information du LaRePE (Dugré et Delisle, 2003). Puisque le système qui supporte les rapports PDG est un des systèmes informatiques les plus complets au Laboratoire, c'est le premier à être intégré à ce projet d'entrepôt. Le travail d'intégration des données des questionnaires, de transformation de ces données à des fins de recherche et de production de rapports est presque complété. Il reste cependant beaucoup à faire. Même si le formulaire de saisie des questionnaires fonctionne avec l'entrepôt actuel, il est tout de même prévu d'en refaire un nouveau qui sera accessible par le Web. De nombreux dispositifs de sécurité pour permettre la saisie par des tiers seront ajoutés. Il reste aussi à créer un rapport qui sera plus modulaire et qui pourra pleinement profiter des avancées faites avec la création de l'entrepôt. De nombreux autres projets sont possibles à moyen ou à long terme.

6.3.1 Questionnaire

Le questionnaire manufacturier est un document imprimé qui contient les données utilisées pour produire le rapport PDG. Une version électronique de ce document (voir figure 30) existe pour saisir les données et les transférer dans les tables de la base de données manufacturière. Cette version électronique n'est pas flexible et très difficile d'entretien. Un des défis majeurs de l'entretien est qu'il arrive régulièrement qu'une nouvelle version du questionnaire soit créée. Cette situation oblige la version électronique à supporter de nombreuses versions différentes, parfois avec des variables différentes (Dugré et Delisle, 2003). Il est jugé nécessaire de créer un nouveau questionnaire électronique, plus simple d'entretien et plus modulaire pour recueillir les données des nouveaux questionnaires imprimés ou Web.

Figure 30 Version électronique du questionnaire manufacturier

Ce nouveau questionnaire pourra alors pleinement bénéficier de l'existence de l'entrepôt, dont la présence des métadonnées permettra de lier plus facilement les champs de saisie aux variables. De même, les structures créées pour supporter le questionnaire (par exemple, les pages, les questions, l'emplacement des variables, etc.) pourront être utilisées par l'entrepôt afin d'afficher une documentation plus complète à l'utilisateur lorsqu'il demandera d'afficher les métadonnées sur chaque variable. Une telle intégration entre le questionnaire et l'entrepôt permettrait d'éviter de nombreux problèmes d'actualisation de la documentation sur les variables. Il n'est pas encore possible de donner plus de détails car le développement du questionnaire fait partie des travaux en cours au Laboratoire, et aucun choix définitif n'a été fait. De nombreux outils pourraient être créés afin de faciliter l'entretien des questionnaires et de permettre à des individus sans expérience de programmation d'ajouter ou de modifier les questions accessibles dans les pages des questionnaires. Ainsi, lorsque de nouvelles variables seraient créées, ces outils prépareraient automatiquement la structure de l'entrepôt pour les recevoir et faciliteraient la documentation de ces variables dans le dictionnaire. Comme on le voit, la présence de l'entrepôt servirait alors d'infrastructure pour supporter le questionnaire.

6.3.2 Rapport

Le rapport PDG devra fonctionner sur le Web et générer des évaluations plus rapidement. Pour l'instant, ce rapport fonctionne à base de *macros* Visual Basic, de procédures *SAS* et de formules dans un classeur *Excel*. Afin d'incorporer les nouveaux critères nécessaires à son utilisation sur le Web ainsi que pour ajouter de nouvelles fonctionnalités, un nouveau programme de rapport doit être développé. Puisque plusieurs calculs sont déjà faits à l'avance dans l'entrepôt, les modules utilisés pour créer le rapport pourront en bénéficier afin d'accélérer le traitement des données et la création du rapport sur le Web. Cependant, puisqu'un accès sur le Web implique un temps de réponse très court (au plus une dizaine de secondes), il pourrait s'avérer avantageux de créer un magasin de données pour supporter l'ensemble des calculs du rapport. Par exemple, il serait possible de faire plusieurs calculs par rapport aux secteurs d'activité, de combiner à l'avance des groupes témoins qui sont souvent utilisés, de calculer des moyennes et des médianes qui pourront être utilisées directement dans le rapport, etc. Tous ces calculs pourraient être faits en prévision de futurs rapports. De plus, si *Oracle 9i* ou *10g* est disponible, les vues matérialisées pourraient être utilisées dans le magasin afin d'accélérer de nombreux calculs automatiquement.

Peu importe les méthodes utilisées, le nouveau rapport sera plus modulaire et supportera de nouvelles méthodes pour afficher des informations utiles. L'intelligence artificielle ajoutée à ce rapport fera l'objet d'un autre projet de maîtrise afin de permettre au logiciel d'intégrer plus rapidement les observations faites par les experts. Le rapport devra apprendre à reconnaître de nouvelles situations et poser des diagnostics corrects dans des cas similaires. L'entrepôt servira d'infrastructure à ce nouveau rapport en fournissant les données sources, ainsi que les métadonnées qui seront essentielles aux composantes d'intelligence artificielle.

6.3.3 Utilisations futures

L'entrepôt sert d'infrastructure pour un nouveau système d'information au LaRePE. Il donne accès aux données recueillies par le questionnaire manufacturier, mais plusieurs autres systèmes et questionnaires sont présents au Laboratoire. Le nouveau questionnaire électronique prévu à la sous-section 6.3.1 devrait idéalement être suffisamment flexible pour supporter toutes les autres formes de questionnaires nécessaires, permettant ainsi une saisie directe de tous les autres projets de questionnaires dans l'entrepôt. De la même façon, le nouveau logiciel de rapport devra permettre la création de rapports selon les différentes exigences des chercheurs et des clients du Laboratoire. Il y a déjà plusieurs projets en cours ou en négociation au Laboratoire qui pourraient profiter pleinement de cette nouvelle façon de faire. Il n'est pas encore possible de prévoir toutes les utilisations qui seront faites de ce nouveau système dans le cadre de l'analyse statistique, de la préparation de nouvelles applications de *benchmarking* ou de forage de données. Mais on peut comprendre que la création de l'entrepôt n'est que le début d'une série de projets qui vont révolutionner les anciennes façons de faire et rendre désuets les anciens outils.

CHAPITRE 7

CONCLUSION

L'entrepôt de données du LaRePE est un ajout à un système existant. Le but de cet ajout est d'avoir un accès plus simple et plus rapide aux données, tout en supportant de nouvelles applications, comme le forage de données. Ces besoins auraient été difficiles à combler avec les bases de données existantes, sans avoir à refaire tout le système. Lorsque le nouveau système avec un entrepôt est comparé à l'ancien système, des améliorations notables en ressortent. Certains résultats montrent clairement que le travail de réingénierie est sur la bonne voie, et qu'il offre déjà de nouvelles possibilités qu'il aurait été très difficile de réaliser avec les structures de données antérieures.

7.1 Améliorations de l'ancien système

L'entrepôt rend l'utilisation des données beaucoup plus facile qu'auparavant. Les chercheurs pourront enfin se concentrer sur de nouvelles méthodes pour utiliser ces données plutôt que de passer de nombreuses heures à faire des manipulations préparatoires répétitives. Les outils créés avec l'entrepôt sont une réponse aux nombreux besoins qui ont été évoqués par les chercheurs durant les années où ils ont utilisé la base de données manufacturière. Dans l'introduction, 5 questions ont été posées pour guider le développement de l'entrepôt. Voici à nouveau ces questions, avec les réponses appropriées.

Question 1 : Est-ce qu'il y a lieu de créer un entrepôt de données pour le LaRePE ?

L'analyse du système informatique du LaRePE (voir la section 3.1) permet de constater qu'il y a plusieurs domaines où des problèmes se sont glissés et ils persistent depuis longtemps. Certains de ces problèmes sont bien identifiés (voir la sous-section 3.1.3) et une revue en profondeur du système semble être la meilleure façon de les résoudre. Cependant, vu la nature du système de production des rapports, il n'est pas souhaitable de développer une solution parallèle pendant plusieurs années qui fonctionne en même temps que l'ancien système, ni d'interrompre la production des rapports. C'est pourquoi un entrepôt de données est la solution idéale. Les formulaires de saisie continuent à servir comme auparavant, les données déjà saisies servent directement dans la nouvelle structure, et il est possible de relier les applications utilisées plus rapidement pour permettre de profiter du nouvel entrepôt. Un remplacement graduel des unités de calcul des différents rapports peut se faire sans avoir à fermer prématurément l'ancien système de production.

Question 2 : Comment consolider différentes sources de données dans un entrepôt de données qui pourra servir à la recherche ?

La structure des tables a été revue et les données ont été nettoyées et intégrées à l'entrepôt. Pour ce faire, des étapes d'extraction, de transformation et de chargement (*Extract, Transform and Load*) ont été préparées (voir la sous-section 3.2.4) pour permettre un chargement initial, puis une mise à jour des données de l'entrepôt à partir des différentes bases de données sources. Comme suite à ce travail, les

données sont maintenant insérées dans un magasin historique (voir la sous-section 3.2.1). Il sert à la fois de point d'entrée des données dans l'entrepôt, et de mécanisme pour conserver l'historique des enregistrements dans l'entrepôt. Les données sont alors plus facilement accessibles, car elles sont stockées à un seul endroit, sous une forme qui peut facilement être manipulée pour les besoins des chercheurs et des rapports.

Question 3 : Comment documenter l'entrepôt et maintenir la documentation à jour ?

Le dictionnaire de variables (voir la sous-section 3.2.2) est l'outil tout désigné pour cette tâche. C'est une nécessité pour se retrouver parmi la vaste quantité de variables qui sont conservées à partir du questionnaire manufacturier. Sans ce dictionnaire, il faut parcourir plusieurs documents qui sont parfois incomplets pour connaître la nature de chaque variable et le format qu'elle prend une fois dans la base de données. Il y a aussi d'autres besoins de documentation. Par exemple, il faut connaître les outils accessibles, les dimensions et tables de faits disponibles, il faut aussi avoir des indications sur les transformations posées sur les données lors du chargement, etc. Toutes ces informations sont accessibles dans les pages d'aide de l'entrepôt, et c'est à cet endroit que toutes nouvelles informations, ou métadonnées de l'entrepôt, devraient être affichées.

Question 4 : Comment permettre aux chercheurs et aux étudiants d'exploiter le plus simplement possible les données de cet entrepôt, avec les outils de leur choix ?

Le *Dataset Maker* (voir la sous-section 3.2.3) est une application accessible sur le site Web de l'entrepôt de données du LaRePE et permet un accès direct et rapide à toutes les données. L'ancienne méthode d'accès aux données nécessitait *SAS* et plusieurs jours pour produire un seul jeu de données, à l'aide de nombreux intermédiaires humains. Maintenant, un chercheur qui a une bonne idée du jeu de données qu'il veut obtenir est capable de produire ses données en moins de 5 minutes. Tout le travail est fait en accédant au site Web de l'entrepôt en utilisant le *Dataset Maker*. Des jeux de données peuvent être créés pour *Excel* et *SAS*, et d'autres formats seront supportés, comme *SPSS*.

Question 5 : Quels outils d'analyse (statistique, forage de données, etc.) pourraient permettre d'améliorer l'exploitation de cet entrepôt ?

Plusieurs outils peuvent se connecter à l'entrepôt pour analyser les données. Le logiciel *SAS* est utilisé depuis plusieurs années pour effectuer des études statistiques. Avec l'ajout de l'entrepôt, les logiciels *ContourCube* pour *OLAP* (voir la sous-section 4.2.1) et *SAS Enterprise Miner* pour le forage de données (voir la sous-section 4.3.1) ont été installés et sont maintenant prêts à servir.

7.2 Les résultats de l'ajout d'un entrepôt de données

L'entrepôt fonctionnel a été présenté lors d'une réunion avec les principaux chercheurs du département des sciences de la gestion et les professionnels de recherche du LaRePE. Le produit fini correspond aux attentes des utilisateurs, mais de nombreuses fonctionnalités supplémentaires devront éventuellement y être ajoutées. La réaction des utilisateurs est donc très positive, pour ne pas dire enthousiaste, et le futur est très prometteur.

Le développement du nouveau système de production des rapports n'est pas terminé, il reste encore beaucoup de travail à faire. Il y a cependant quelques résultats concrets. Il est maintenant possible de créer des jeux de données beaucoup plus rapidement qu'avant, avec moins d'intermédiaires humains (voir la sous-section 5.2.2). La facilité et la rapidité avec laquelle un utilisateur autorisé peut maintenant accéder à des données en ont surpris plusieurs. En fait, les gains en performance sont tellement impressionnants qu'une des premières réactions des chercheurs a été un souci au niveau de la sécurité. Sur un autre plan, une version légèrement modifiée du rapport PDG a été testée (voir la section 5.4) à l'aide de l'entrepôt. Les résultats indiquent que même sans une refonte du rapport, des gains de performance significatifs sont notés. La préparation des données pour le rapport PDG est cinq fois plus rapide. Il serait d'autant plus avantageux de poursuivre le travail d'optimisation pour en arriver à une version suffisamment rapide pour être placée sur le Web.

7.3 Nouvelles possibilités

De nouveaux utilitaires ont été ajoutés à la suite de la création de l'entrepôt. Un magasin à modèle dimensionnel (voir la section 4.1) réorganise les données d'une façon différente pour supporter un logiciel *OLAP* (voir la section 4.2). De cette façon, les utilisateurs qui veulent simplement avoir un aperçu des données, ou encore les personnes responsables de la gestion des données interroger les questionnaires dans l'entrepôt. Tout ceci sans avoir besoin d'un informaticien sur place à tout moment puisque le logiciel *ContourCube* est conçu pour être facilement utilisable par n'importe quel utilisateur, avec une formation minimale. Plusieurs tables de faits sont mises à la disposition des utilisateurs (voir la sous-section 4.1.1), ainsi que différentes dimensions (voir la sous-section 4.1.2). Il est possible d'ajouter de nombreuses dimensions, selon les demandes, parce que les questionnaires ainsi que les structures de l'entrepôt comportent de nombreux champs. Il n'a cependant pas été jugé nécessaire de rendre accessible l'ensemble des dimensions possibles parce qu'il y en a trop. Cette longue liste aurait pu intimider les nouveaux utilisateurs. Il est aussi possible de personnaliser les listes de tables de faits et de dimensions, au besoin, ce qui fait de *ContourCube* un outil idéal pour initier les utilisateurs du LaRePE à *OLAP*.

Une autre nouveauté est l'ajout d'un logiciel de forage de données (voir la section 4.3) supporté par le *Dataset Maker* (voir la sous-section 3.2.3). Un utilisateur peut créer un jeu de données provenant de l'entrepôt pour le forage de données à l'aide du *Dataset Maker*, et l'importer dans le logiciel *SAS Enterprise Miner* (voir la section

5.6). Il est alors possible d'utiliser les nombreux algorithmes de *SAS Enterprise Miner* pour faire un projet complet de forage de données (voir la sous-section 4.3.1). L'utilisation du forage de données est nouvelle au LaRePE, et pour l'instant il n'y a pas encore d'exemple concret de projet créé et complété par un chercheur. Mais le logiciel est prêt et mis à la disposition de tous les utilisateurs qui connaissent déjà *SAS*.

7.4 Nombreuses ouvertures

Il y a d'autres projets qui seront concrétisés à plus long terme, mais qui sont maintenant rendus possibles en partie grâce à l'entrepôt de données (voir la section 6.3). Parmi ces projets, il y a la création d'un logiciel plus modulaire pour la création des rapports. L'infrastructure est assurée par l'entrepôt, ce qui garantit une méthode d'accès de base qui sera supportée et entretenue à long terme. Cette approche modulaire permet à son tour de créer des rapports plus flexibles sous la forme d'une plateforme de diagnostic. Ces rapports pourraient être plus intelligents, notamment en utilisant des techniques d'intelligence artificielle pour diminuer le travail nécessaire à la production des rapports de *benchmarking*. Il deviendrait même envisageable de produire des rapports complexes d'une manière complètement automatique. Et ce ne sont encore que quelques exemples des possibilités qu'offre le nouvel entrepôt de données du LaRePE.

BIBLIOGRAPHIE

Body, M., Miquel, M., Bédard, Y., Tchounikine, A. (2002). *A multidimensional and multiversion structure for OLAP applications*. DOLAP 2002, ACM Fifth International Workshop on Data Warehousing and OLAP.

Cauvet, C., Rosenthal-Sabroux, C. (2001). *Ingénierie des systèmes d'information sous la direction de Corine Cauvet, Camille Rosenthal-Sabroux*. Paris Hermès Science Publications, France, 353 p. ill., ISBN 2-7462-0219-0.

Devlin, B. A., Murphy, P. T. (1988). *An Architecture for a Business and Information System*. IBM Systems Journal 27(1): p. 60-80.

Dinter, B., Sapia, C., Höfling, G., Blaschka, M. (1998). *The OLAP market: state of the art and research issues*. DOLAP 1998, ACM First International Workshop on Data Warehousing and OLAP.

Dugré, M., Delisle, S. (2003). *Le système PDG : Évaluation de l'actuel et Éléments de conception du PDG II*. Institut de recherche sur les PME.

Golfarelli, M., Maio, D., Rizzi, S. (1998). *The Dimensional Fact Model: A Conceptual Model for Data Warehouses*. International Journal of Cooperative Information Systems.

Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A. (1999) *Updating OLAP dimensions*. DOLAP 1999, ACM Second International Workshop on Data Warehousing and OLAP.

Inmon, W.H., Hackarton, R.D. (1994). *Using the Data Warehouse*. John Wiley & Sons, 304 p., ISBN: 0471059668.

Kimball, Ralph. *Data Warehouse Fundamentals*. Ralph Kimball Associates. http://www.rkimball.com/html/articlesfolder/DWfundamental.html (Page consultée le 19 novembre 2003)

Kimball, Ralph. (1996). *The data warehouse toolkit : practical techniques for building dimensional data warehouses*. New York : J. Wiley, Etats-Unis, ISBN 0-471-15337-0.

McFadden, F. R., Hoffer, J. A., Prescott, M. B. (1999). *Modern Database Management Fifth Edition*. Addison-Wesley, États-Unis, 622 p., ISBN 0-8053-6054-9.

O'Neil, Bonnie. (1997). *Oracle Data warehousing unleashed Bonnie O'Neil...[et al.]*. Sams Indianapolis, Ind., Etats-Unis, ISBN 0-672-31077-5.

Rifaieh, R., Benharkat, N. A. (2002). *Query-based data warehousing tool*. DOLAP 2002, ACM Fifth International Workshop on Data Warehousing and OLAP.

Scalzo, Bert. (2003). *Oracle DBA Guide to Data Warehousing and Star Schemas*. Prentice Hall, États-Unis, 208 p., ISBN 0-13-032584-8.

SCN Education B.V. (2001). *Data warehousing the ultimate guide to building corporate business intelligence*. Braunschweig/Wiesbaden Vieweg. Grèce, 336 p. ill., ISBN 3-528-05753-X

Silberschatz, A., Korth, H. F., Sudarshan, S. (1999). *Database System Concepts Third Edition*. WCB McGraw-Hill, États-Unis, 819 p., ISBN 0-07-031086-6.

Theodoratos, D., Bouzeghoub, M. (2000). *A general framework for the view selection problem for data warehouse design and evolution*. DOLAP 00, ACM Third International Workshop on Data Warehousing and OLAP.

Tsois, A., Karayannidis, N., Sellis, T. K. (2001). *MAC: Conceptual data modeling for OLAP*. Design and Management of Data Warehouses.

Vassiliadis, P., Simistsis, A., Skiadopoulos, S. (2002). *Conceptual modeling for ETL processes*. National Technical University of Athens, Athens, Greece.

ANNEXE A

Le rapport PDG manufacturier

Ce projet a été conçu et réalisé en partenariat avec:

LaRePE
LAboratoire de REcherche sur la Performance des Entreprises

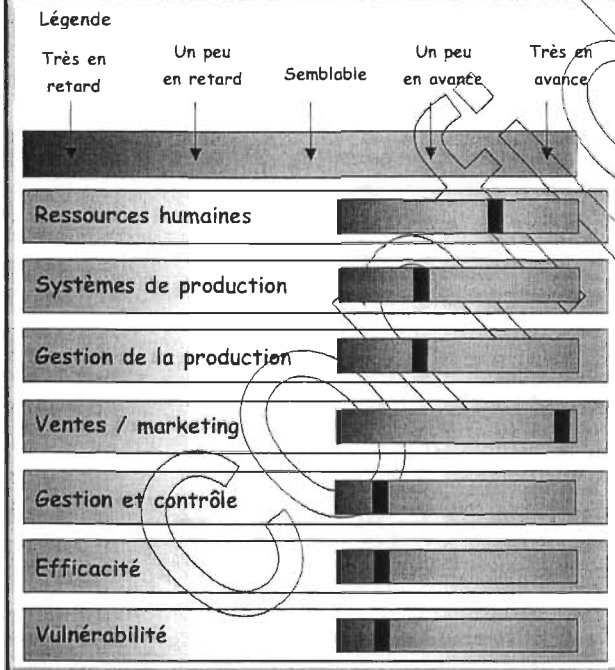
Développement économique Canada | Canada Economic Development
Canada

96
Groupement des chefs d'entreprise

PDG MD manufacturier

sommaire

Les informations générales et financières de votre entreprise ont été mises en parallèle à celles d'un groupe témoin d'entreprises semblables afin de vous situer par rapport à celles-ci. Aucun jugement n'est porté sur votre gestion et sur votre situation générale. Les graphiques et les commentaires sont présentés dans le but de vous aider à faire une réflexion sur la situation de votre entreprise par rapport à vos objectifs personnels. Le tableau qui suit présente sommairement le constat qui se dégage de vos informations qui ont été pondérées pour tenir compte de leur importance relative. La position du curseur pour chacune des fonctions de l'entreprise dépend donc à la fois de ces informations et de l'importance relative qu'elles représentent. Nous avons également identifié deux pistes de réflexions prioritaires que nous vous invitons à étudier à court terme.



Première suggestion:

L'implantation d'un système de calcul de prix de revient permettrait de connaître les coûts réels de production, d'identifier les sources possibles d'inefficacité et d'optimiser la rentabilité de l'entreprise.

Deuxième suggestion:

L'évaluation de la politique de financement et l'analyse de la possibilité de réinjecter de nouveaux capitaux permettraient de réduire le niveau de risque financier et la vulnérabilité de l'entreprise.

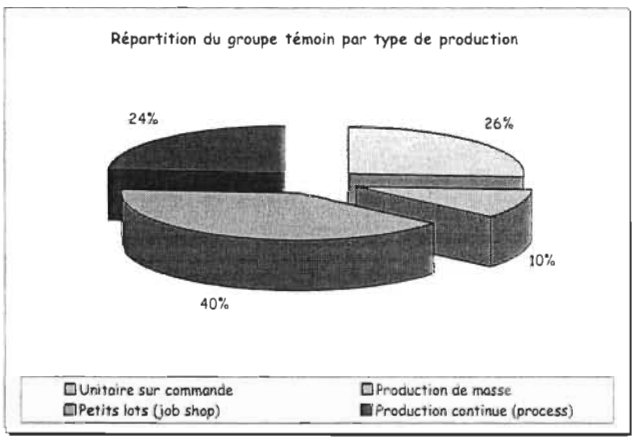
GROUPE TÉMOIN

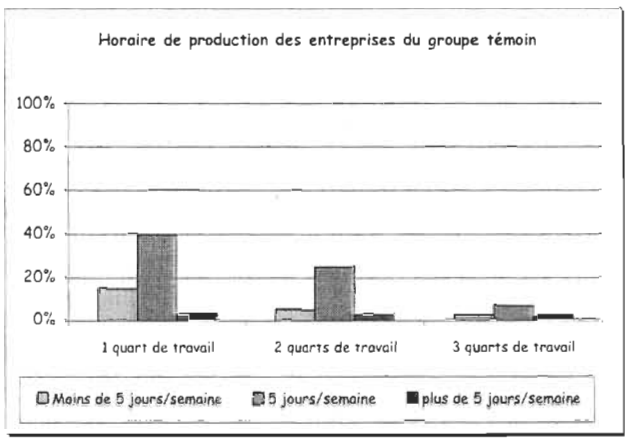
Numéro de l'entreprise	Critère(s) de sélection du groupe témoin
Démonstrateur	Secteur d'activités

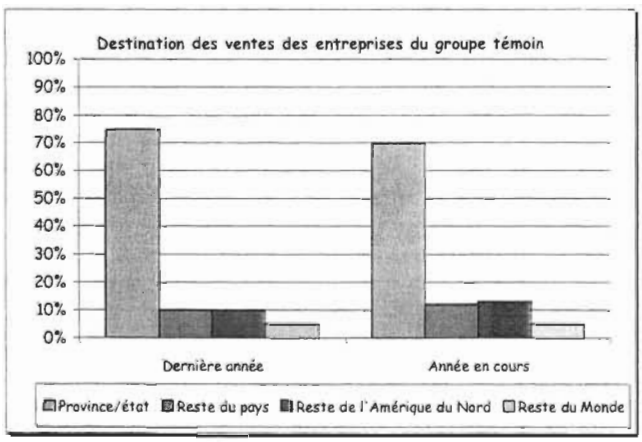
Caractéristiques du groupe témoin

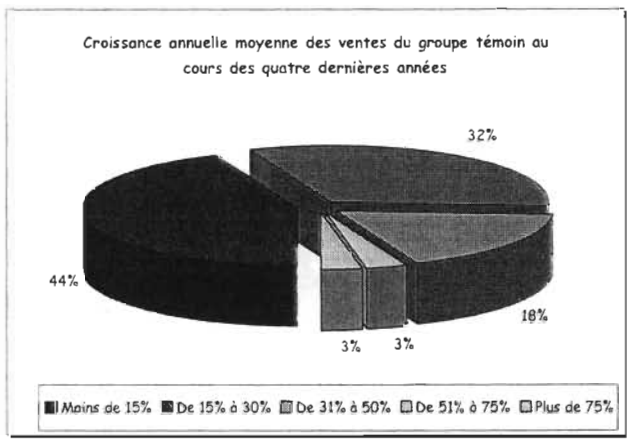
Voici quelques informations sur les entreprises de votre groupe témoin. Ces informations n'ont pas été utilisées pour définir le groupe auquel vous avez été comparé, mais leur présentation vise simplement à vous indiquer comment évoluent des entreprises qui ont des caractéristiques semblables à la vôtre. Ces informations pourraient alors être utiles pour définir certaines stratégies de développement.

Nombre d'entreprises dans le groupe témoin	20 entreprises
Âge médian des entreprises	19 ans
Nombre médian d'employés	35 employés
Pourcentage des entreprises syndiquées	32%
Pourcentage des entreprises qui font de la sous-traitance	63%
Pourcentage des entreprises qui ont des programmes de formation sur mesure	20%
Pourcentage des entreprises qui protègent leurs innovations	30%

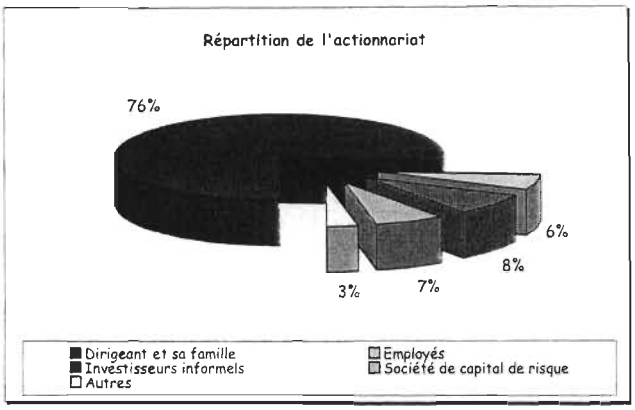


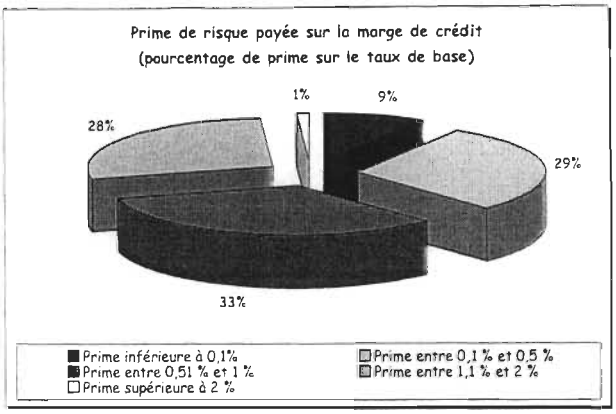


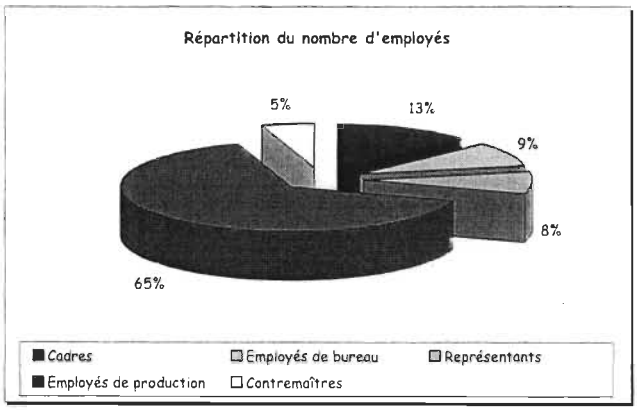


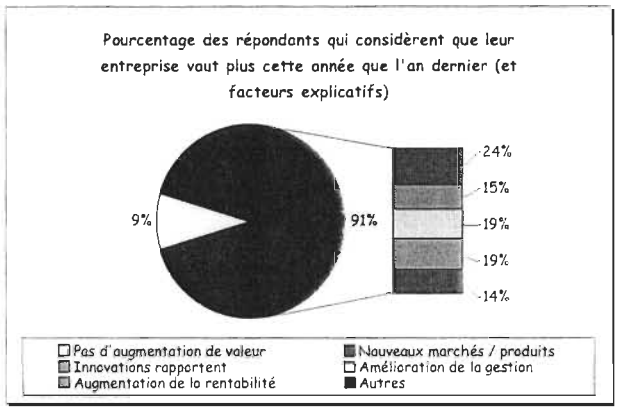


GROUPE TÉMOIN (suite)





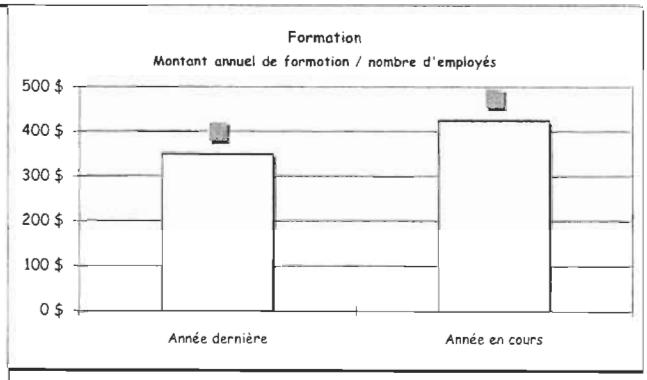


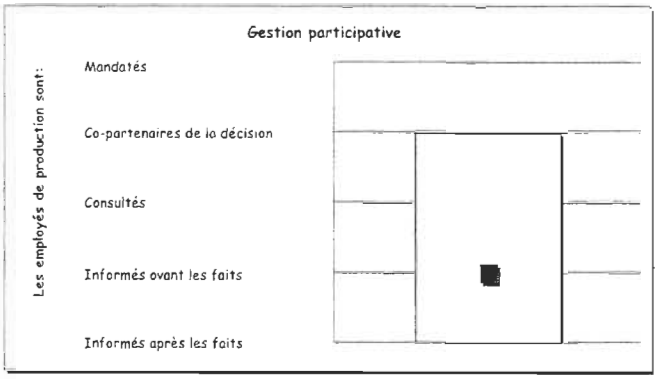


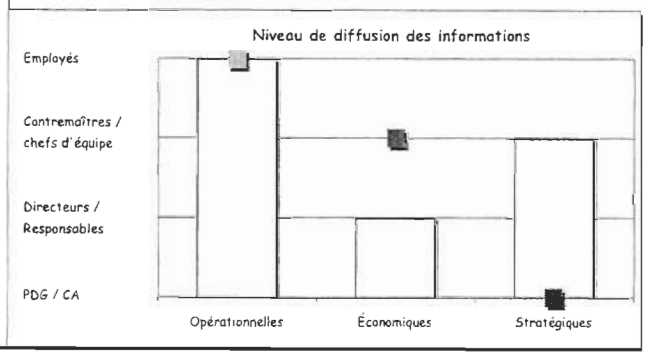
Prévisions des entreprises du groupe témoin concernant certaines informations stratégiques pour la prochaine année:

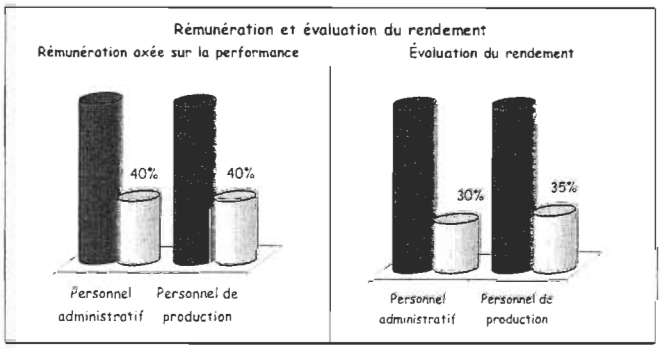
	DIMINUTION	STABILITE	AUGMENTATION
Informations financières			
Croissance annuelle du chiffre d'affaires	5%	9%	86%
Marge bénéficiaire brute	8%	30%	62%
Marge bénéficiaire nette	10%	25%	65%
Endettement total	38%	38%	24%
Taux de rendement des fonds propres	22%	8%	70%
Variations du nombre d'employés			
Cadres	20%	64%	16%
Employés de bureau	21%	53%	26%
Représentants	26%	39%	35%
Employés de production	29%	21%	50%
Contremaîtres (chefs d'équipe)	20%	59%	21%

Pratiques

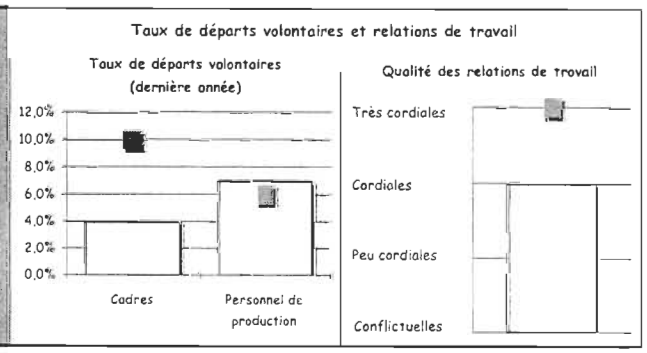


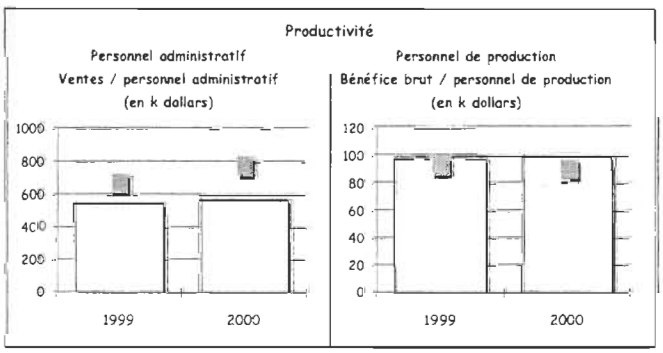






Résultats





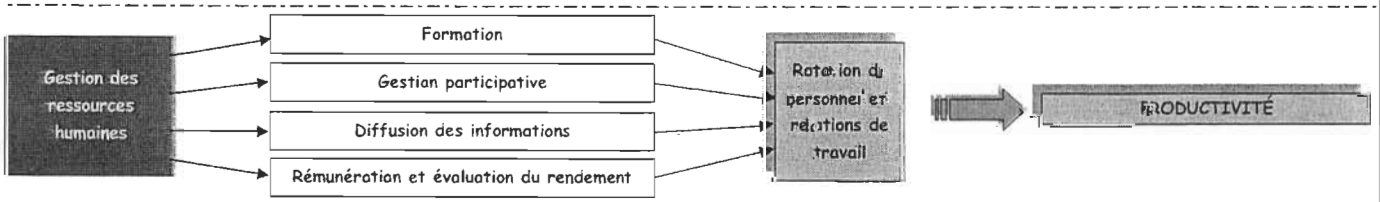
Évaluation des pratiques : Les pratiques de gestion des ressources humaines sont en général plus développées que celles du groupe témoin. Une amélioration de la performance de la gestion des ressources humaines pourrait être envisagée, notamment par l'implantation de la gestion participative pour accroître l'implication des employés dans le développement de l'organisation.

Commentaires sur les résultats : Les résultats concernant l'efficacité de la gestion des ressources humaines sont globalement semblables à ceux du groupe témoin. Il peut être nécessaire de porter une attention particulière aux raisons expliquant le taux élevé de départs volontaires des cadres afin de réduire les coûts de recrutement et de formation.

Bien que l'entreprise soit différente du groupe témoin sur les informations suivantes, celles-ci N'ONT PAS été prises en considération dans l'évaluation :

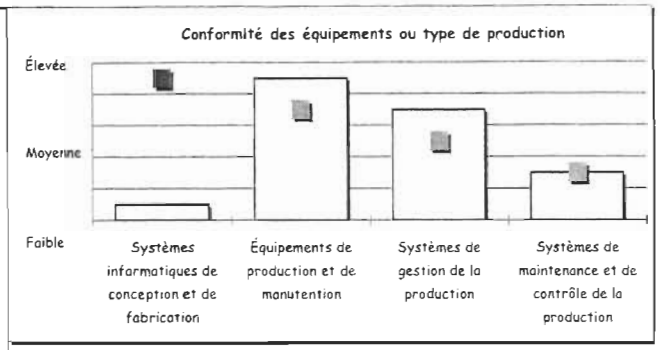
La présence d'un responsable en ressources humaines

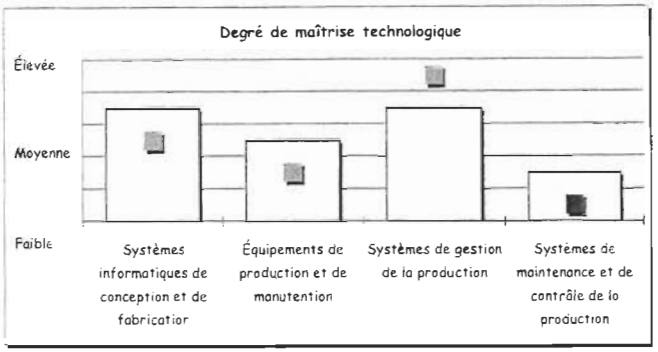
Les ressources humaines constituent de plus en plus l'un des principaux actifs des entreprises performantes. Pour cette raison, il faut constamment mettre à jour leurs connaissances, les tenir au courant du développement de l'entreprise, évaluer leur rendement et les motiver à travers une politique de rémunération incitative afin d'encourager les employés à travailler pour l'atteinte d'objectifs de rendement qui conviennent aux dirigeants.

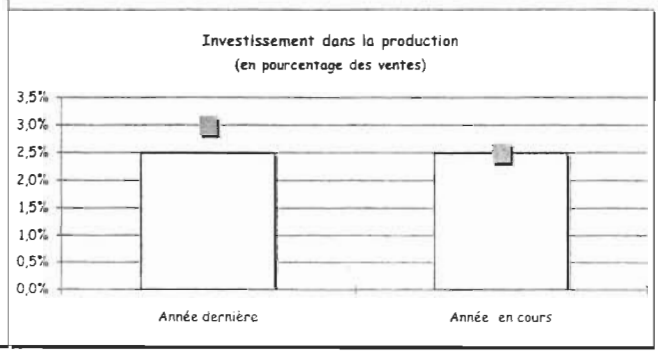


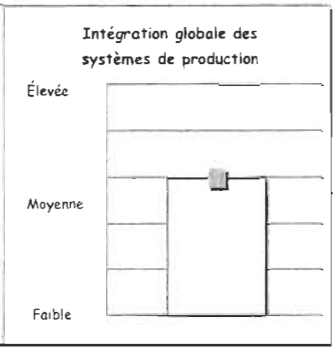
SYSTÈMES DE PRODUCTION

Pratiques

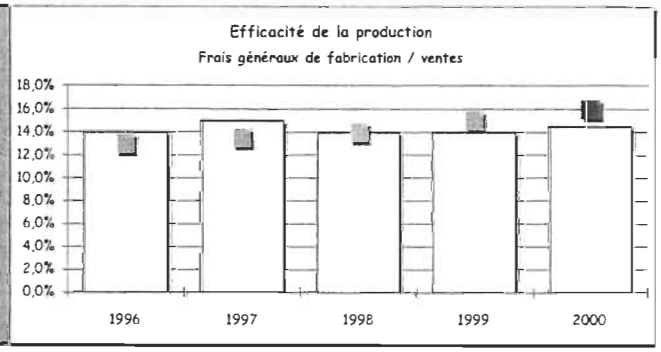


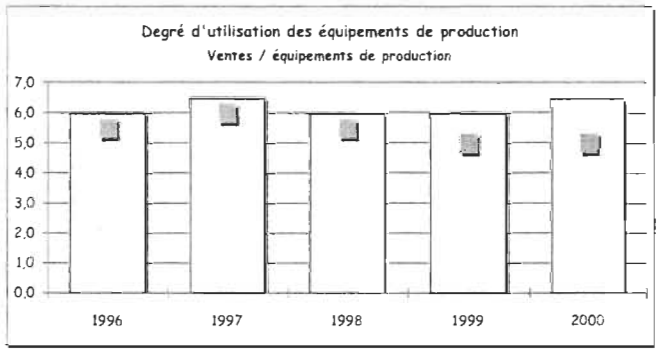






Résultats



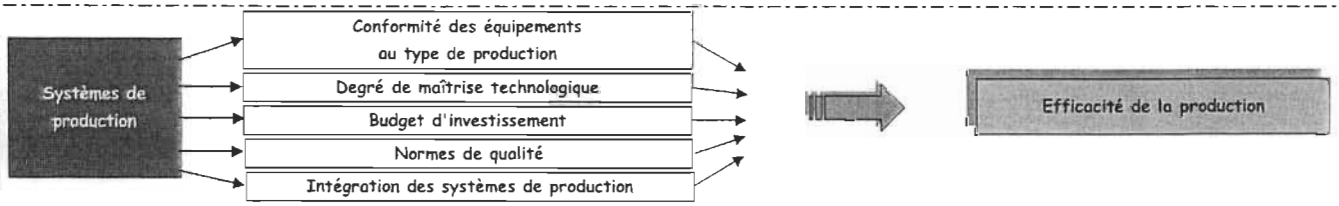


Evaluation des pratiques Les systèmes de production sont en général aussi développés que ceux du groupe témoin. Une modification des pratiques actuelles notamment par l'atteinte d'un plus grand degré de maîtrise technologique des équipements/systèmes en place pourrait contribuer à réduire les temps de production et en accroître l'efficacité.

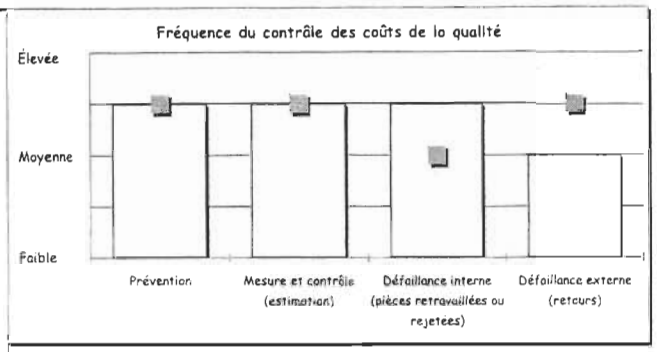
Commentaires sur les résultats : L'examen des frais généraux de l'entreprise montre que l'efficacité de la production s'est détériorée depuis trois ans et, par rapport au groupe témoin, celle-ci est considérée en moyenne inférieure. Au cours des trois dernières années, le degré d'utilisation des équipements a été globalement inférieur au groupe témoin et la situation est demeurée stable. Une attention pourrait être portée aux frais généraux de fabrication dont l'écart par rapport au groupe témoin montre un plus faible degré de compétitivité.

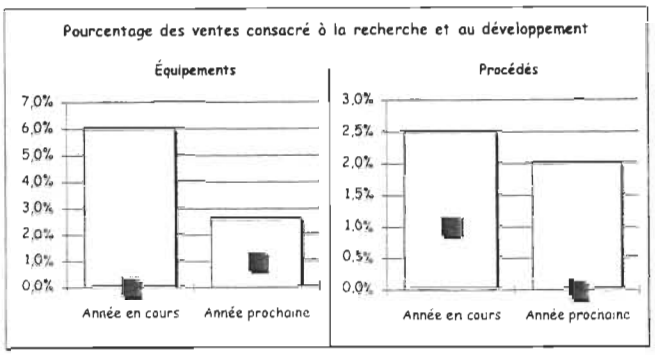
Aucune précision à ajouter sur l'évaluation faite de cette fonction de l'entreprise.

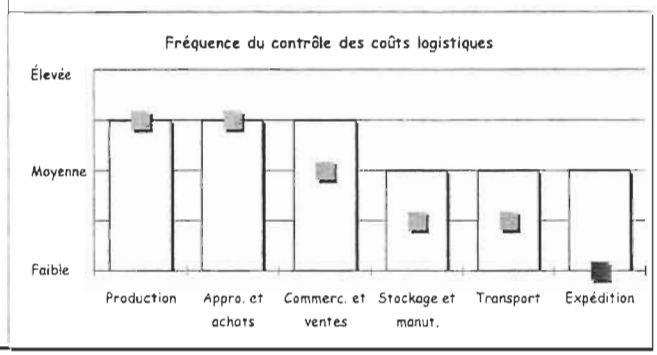
L'efficacité des activités de production est liée au choix des équipements, des aménagements et des technologies, et surtout à leur intégration dans un système efficace qui respecte le type de production de l'entreprise. L'efficacité de la fonction de production permettra de réduire au minimum les coûts de production de l'entreprise et d'améliorer la rentabilité de ses immobilisations.

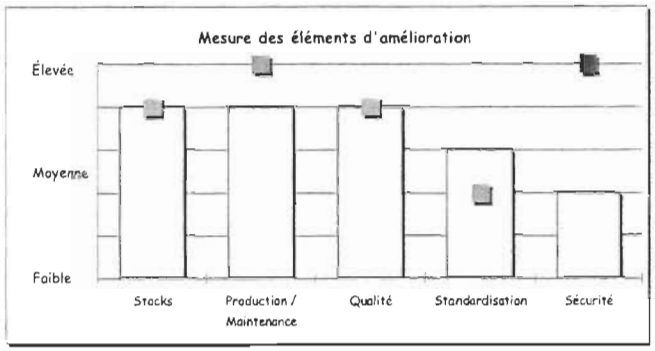


Pratiques

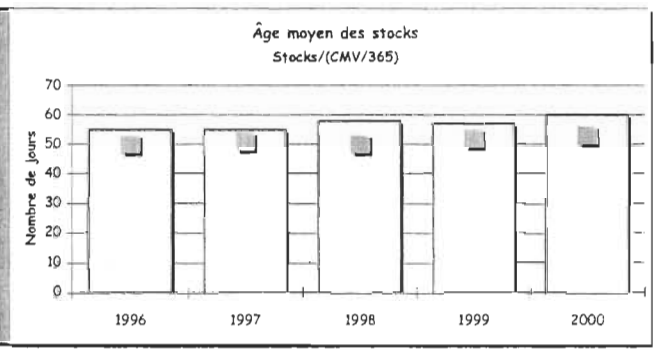


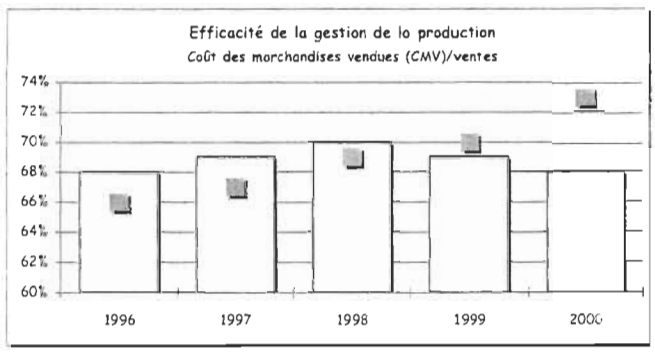






Résultats



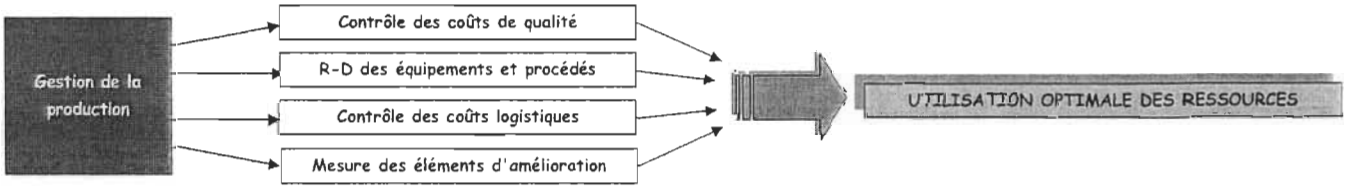


Evaluation des pratiques : Les pratiques de gestion de la production utilisées dans l'entreprise sont en général moins développées que celles du groupe témoin. Une modification de celles-ci pourrait contribuer à la performance de l'entreprise notamment par une évaluation de l'importance d'intensifier les activités de R-D destinées à la production pour accroître l'efficacité des équipements et des systèmes et réduire les coûts associés à cette fonction.

Commentaires sur les résultats : Dans les trois dernières années, le délai de transformation des stocks a été en moyenne inférieur au groupe témoin, et la situation par rapport à celui-ci s'est maintenue. L'évaluation du CMV montre que l'efficacité de la gestion de la production s'est détériorée depuis trois ans, et la position de l'entreprise est relativement défavorable par rapport au groupe témoin. Une attention particulière pourrait être portée au coût des marchandises vendues afin d'augmenter la compétitivité de l'entreprise.

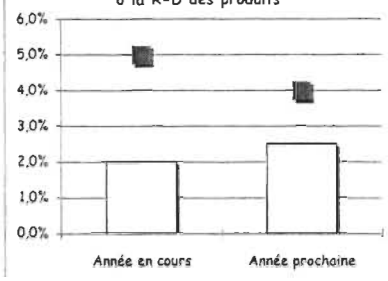
Aucune précision à ajouter sur l'évaluation faite de cette fonction de l'entreprise.

Les activités de contrôle et de gestion sont aussi fondamentales pour l'efficacité de la production que le sont les types d'équipements et d'aménagements. On considère important que les entreprises manufacturières insistent sur leurs activités de planification, sur les pratiques d'amélioration continue, sur le contrôle de la qualité de leurs produits ainsi que sur leurs coûts logistiques.

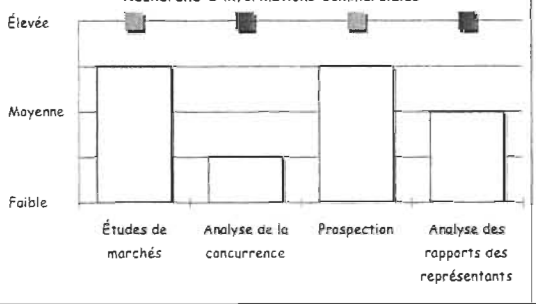


Pratiques

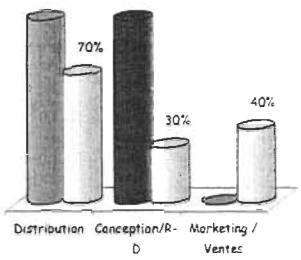
Pourcentage des ventes consacré à la R-D des produits



Recherche d'informations commerciales

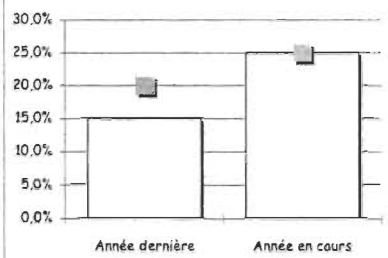


Collaboration d'affaires

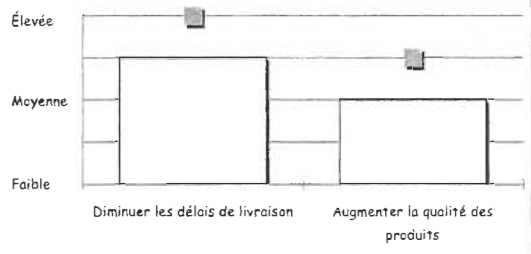


Effort de vente

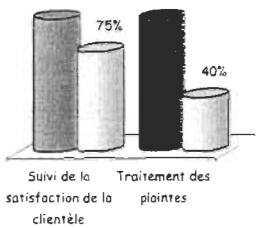
Représentants/nombre total d'employés



Importance des objectifs concernant certaines activités d'amélioration continue

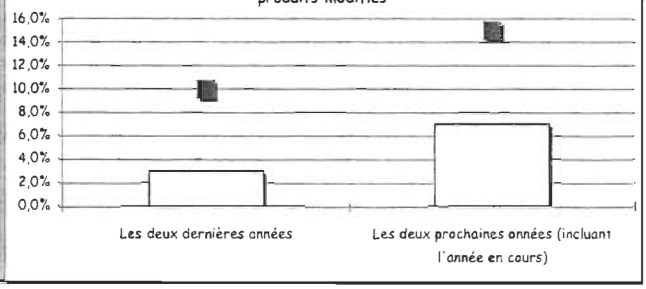


Suivi de satisfaction auprès de la clientèle et traitement des plaintes

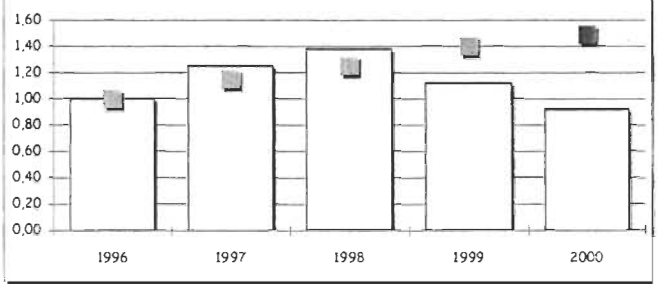


Résultats

Pourcentage des ventes attribuable aux nouveaux produits et aux produits modifiés



Indice de croissance des ventes (nombre de fois les ventes de 1996)



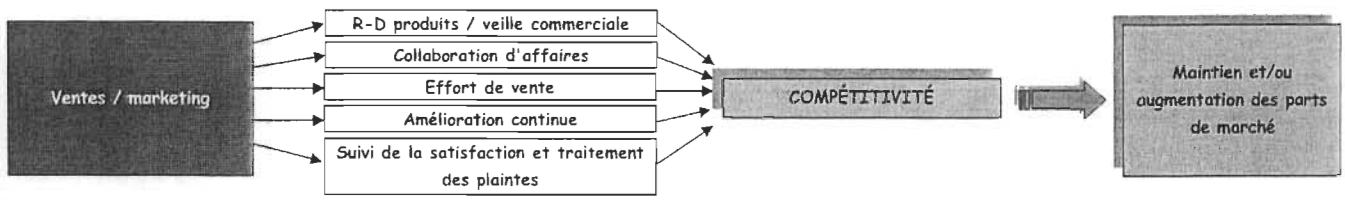
Évaluation des pratiques : Les pratiques de ventes et de marketing utilisées dans l'entreprise sont en général plus développées que celles du groupe témoin. La modification de celles-ci ne semble pas nécessaire à court terme.

Commentaires sur les résultats : Le pourcentage des ventes attribuables aux nouveaux produits est en moyenne supérieur au groupe témoin, et la croissance du chiffre d'affaires est en général plus importante. Les résultats obtenus ne requièrent aucune recommandation particulière.

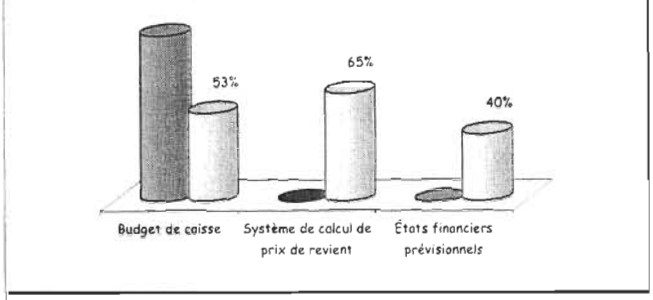
Bien que l'entreprise soit différente du groupe témoin sur les informations suivantes, celles-ci N'ONT PAS été prises en considération dans l'évaluation :

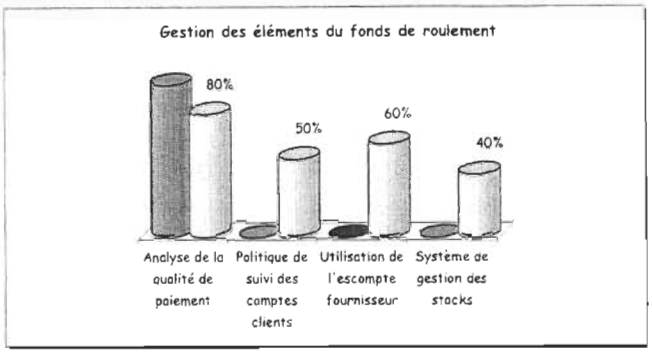
La diversification de la clientèle

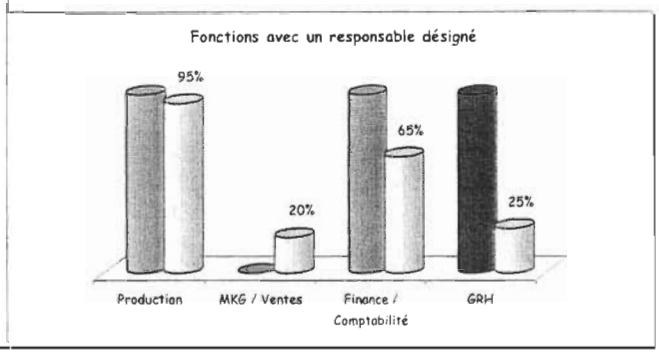
Maintenir sa part de marché est de plus en plus difficile avec la mondialisation des économies qui implique la venue rapide des concurrents et qui accroît continuellement les exigences des clients. Il faut donc évaluer la satisfaction de la clientèle, étudier les marchés actuels et potentiels et analyser toutes les informations commerciales disponibles.

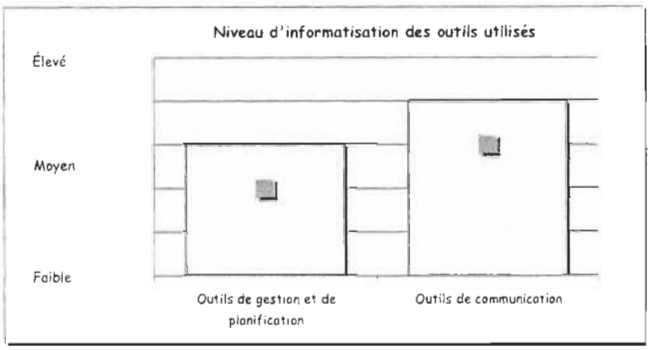


Pratiques

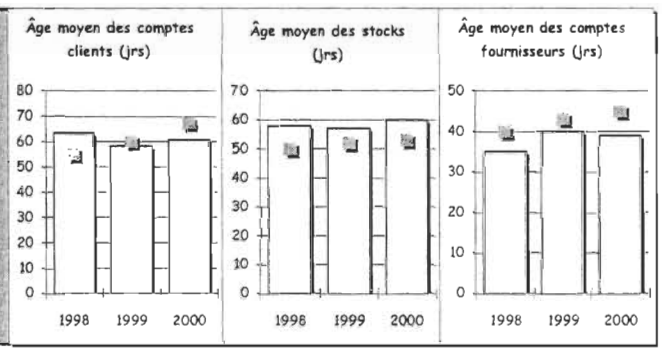


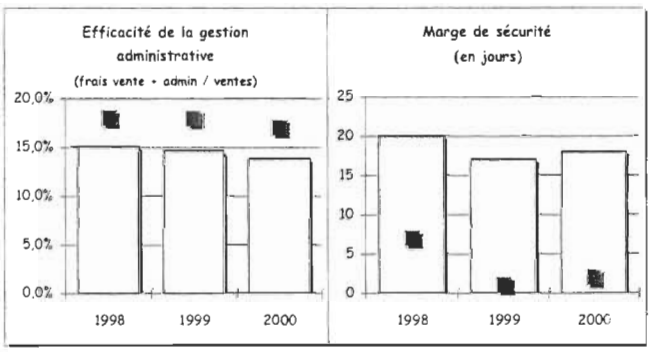






Résultats





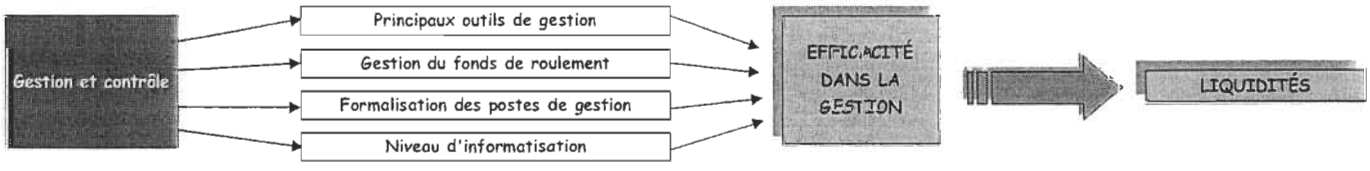
Évaluation des pratiques : Les pratiques de l'entreprise en terme de gestion et de contrôle financier sont en général moins développées que le groupe témoin. Une amélioration de la gestion et du contrôle pourrait être envisagée grâce à l'utilisation d'un système de calcul de prix de revient pour connaître les coûts réels de production de l'entreprise et maximiser sa rentabilité.

Commentaires sur les résultats : D'après les résultats des trois dernières années, les frais de vente et d'administration n'ont montré aucune tendance particulière et ceux-ci ont été en moyenne supérieurs au groupe témoin. Au niveau de la marge de sécurité, l'entreprise présente une situation globalement défavorable par rapport groupe témoin, et celle-ci se détériore. Il serait intéressant de porter une attention particulière aux frais de vente et d'administration pour voir à une utilisation efficace du personnel et des différents outils de gestion et ainsi favoriser le degré de compétitivité de l'entreprise.

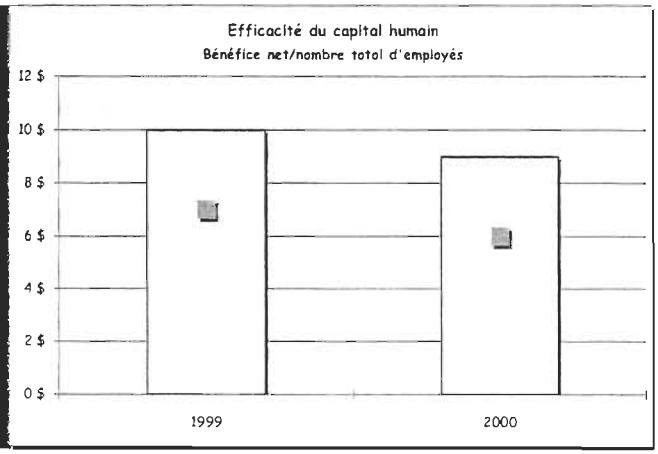
Bien que l'entreprise soit différente du groupe témoin sur les informations suivantes, celles-ci N'ONT PAS été prises en considération dans l'évaluation :

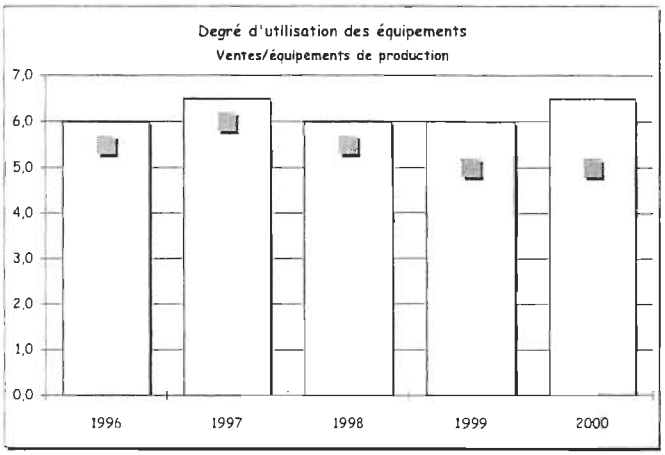
La diversification de la clientèle

La formalisation des activités de gestion est nécessaire à mesure que croît la taille de l'entreprise et qu'augmente son niveau de complexité. Différents outils de gestion permettent ainsi de mieux gérer et planifier les diverses activités qui ont des répercussions à plus ou moins court terme sur les liquidités qui sont essentielles au bon fonctionnement de l'entreprise.



Opérations



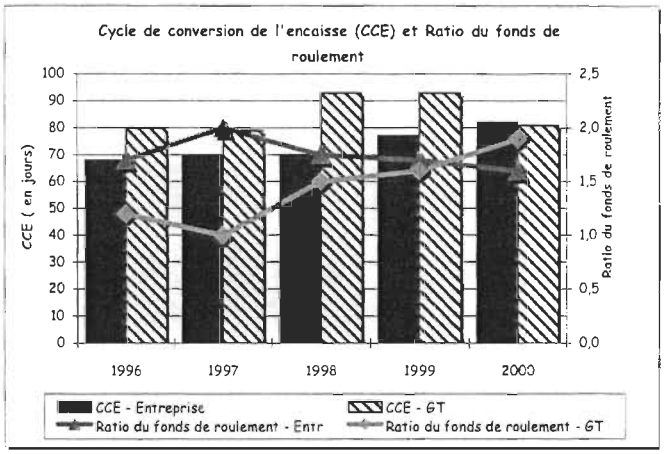


Liquidités

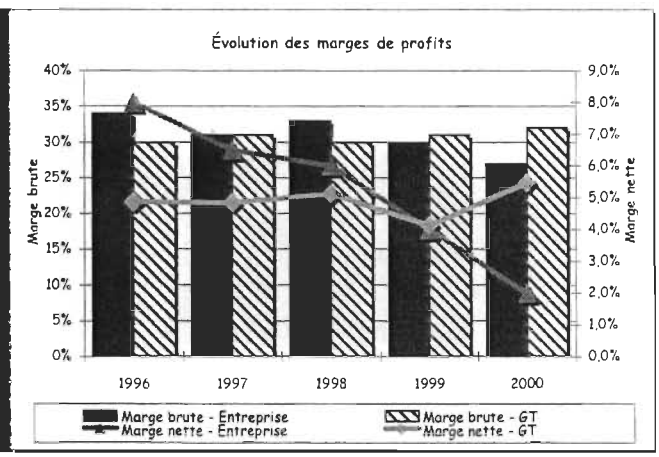
Éléments composant le cycle de conversion de l'encaisse (CCE) :

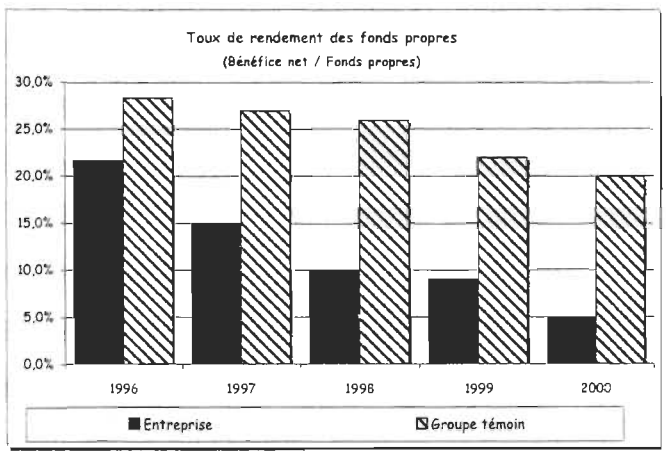
		1999	2000
	Âge moyen des comptes clients	60	68
+	Âge moyen des stocks	52	53
-	Âge moyen des comptes fournisseurs	43	45
=	Cycle de conversion de l'encaisse	69 jours	76 jours

Ces résultats sont illustrés sur la figure ci-contre



Rentabilité

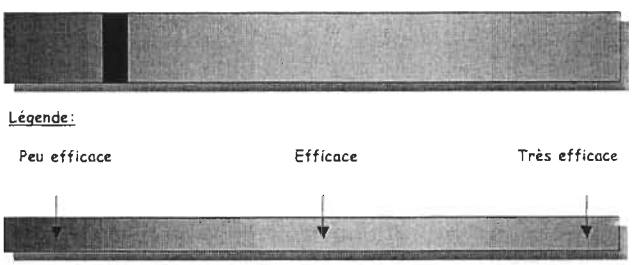




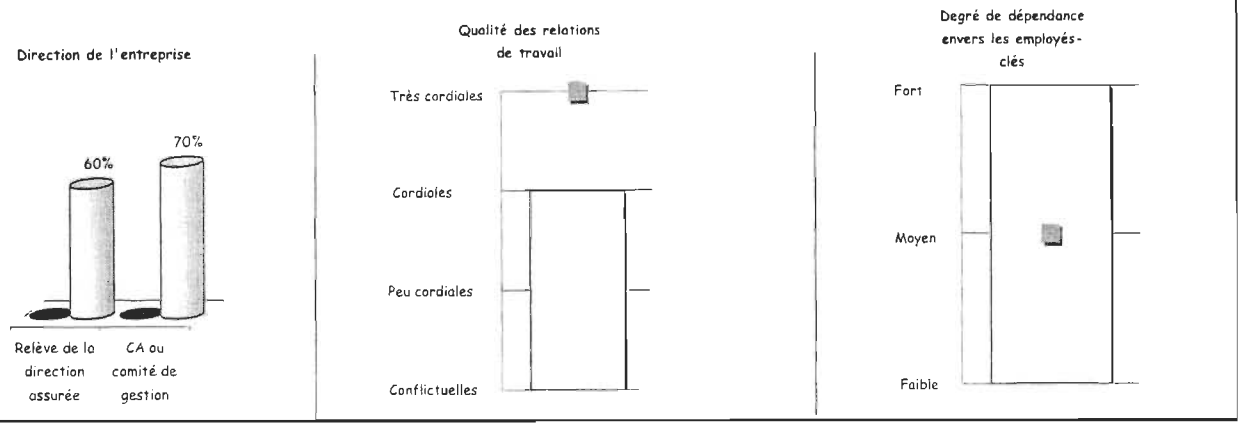
Pour être efficace, l'utilisation des différentes ressources de l'entreprise, qu'elles soient humaines, matérielles ou financières, doit produire l'effet attendu par les dirigeants. C'est l'ensemble de ces informations qui est évalué ici pour donner un indice d'efficacité "global" à l'entreprise.

Évaluation du niveau d'efficacité : Par rapport au groupe témoin, l'efficacité dans l'utilisation des ressources est considérée globalement inférieure. Pour accroître celle-ci, il serait intéressant d'analyser les divers coûts afin d'identifier des sources possibles de gaspillage et s'assurer que l'entreprise en a le plein contrôle.

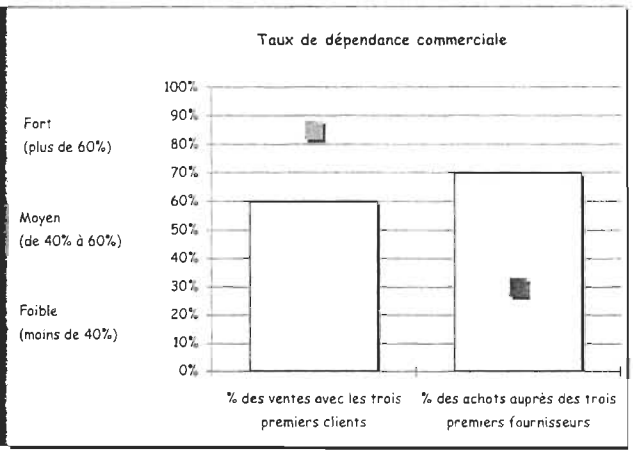
Degré d'efficacité global de l'entreprise

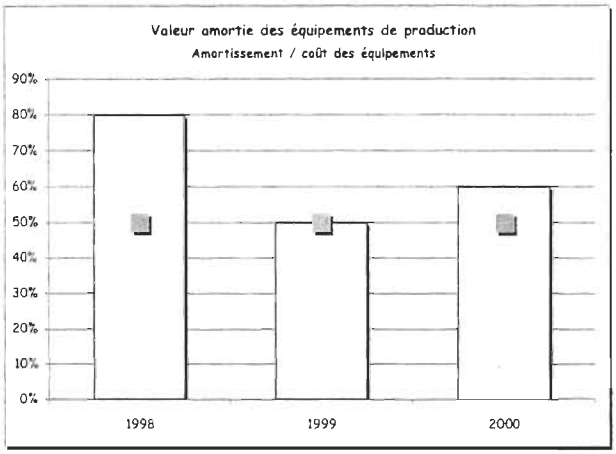


Ressources humaines

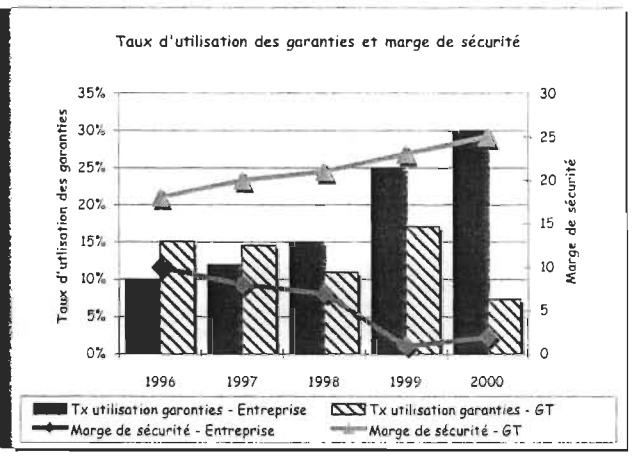


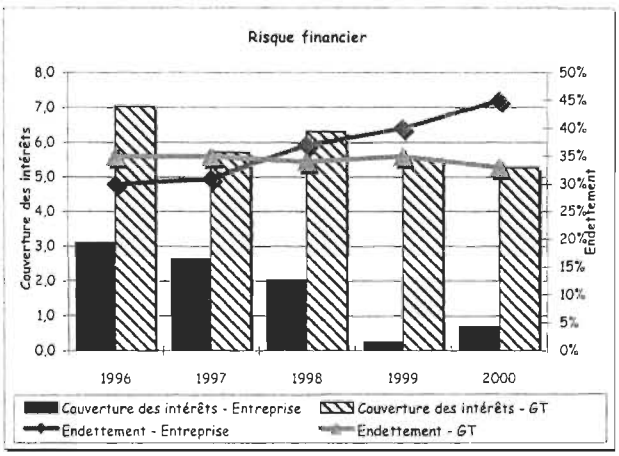
Production





Ressources monétaires

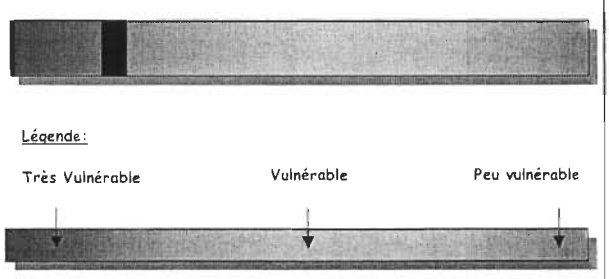




Une entreprise peut être très efficace, mais vulnérable à des perturbations de son environnement, ce qui doit être évalué par la direction. Cette vulnérabilité peut venir de différentes sources, qu'elles soient humaines, financières, commerciales ou de production.

Évaluation du degré de vulnérabilité : Considérant différents aspects de l'entreprise, son degré de vulnérabilité est légèrement supérieur à celui du groupe témoin. Une façon de réduire la vulnérabilité de l'entreprise serait de préparer une relève pour la direction ou de mettre sur pied un conseil d'administration ou un comité de gestion afin d'éviter une perte d'efficacité à l'organisation en l'absence de ses principaux dirigeants.

Degré de vulnérabilité global de l'entreprise



ANNEXE B

Le forage de données

Mathieu Dugré

MAP6007
Consultation scientifique

Le forage de données

Travail remis à
Sylvain Delisle

Université du Québec à Trois-Rivières
Hiver 2002

Table des matières

Introduction

Le forage de données est une application de plus en plus populaire des bases de données. Les algorithmes utilisés dans le processus de forage de données permettent de parcourir toutes les données contenues dans un entrepôt constitué d'une ou plusieurs bases de données et d'identifier, à partir des données, des informations nouvelles et pertinentes. Ces informations peuvent servir à des chercheurs, des gestionnaires pour classifier les clients potentiels à cibler lors de campagnes publicitaires, des assureurs pour trouver les fraudeurs, etc. Ce document vise à identifier des algorithmes et des produits commerciaux qui peuvent servir à la conception et l'implémentation d'une application de forage de données. Cette application fonctionnelle doit pouvoir servir à des chercheurs de différents domaines pour extraire des données d'un entrepôt de données et à appliquer les algorithmes de forages de données, et à faciliter l'interprétation des résultats obtenus.

Les différentes étapes du forage de données doivent être représentées à l'intérieur de l'application qui sera développée. La préparation des données et la visualisation des résultats doivent pouvoir se faire dans cette application. L'application doit aussi permettre la sélection des algorithmes. Il existe plusieurs algorithmes pour effectuer le forage de données. Ces algorithmes n'ont pas tous le même but. Certains servent à faire de la classification d'éléments, d'autres permettent de trouver des règles d'association entre les éléments, et plusieurs permettent de prévoir la classification de nouveaux éléments. Ce document identifie les données à fournir à l'algorithme tout en expliquant le type de résultat qui est produit. La plupart des algorithmes sont déjà implémentés dans des applications commerciales. Ainsi, il est possible de se procurer un ou plusieurs produits et de réutiliser les algorithmes dans l'application de forage de données qui sera produite pour les chercheurs.

Une conclusion est faite à la suite de chaque algorithme et produit commercial. De même, une discussion sur les différents éléments de l'application et d'autres types de considérations au développement d'une application de forage de données est présentée à la fin de ce document.

Étapes du forage de données

Le forage de données est un processus complexe qui implique de nombreuses connaissances et beaucoup de manipulations de données. De plus, des connaissances de base dans le domaine concerné sont essentielles pour pouvoir vérifier et interpréter les résultats du processus. Voici les étapes du forage de données [ADV96] :

1. **Développer une compréhension** du domaine d'application, les connaissances de base nécessaires et les buts des utilisateurs.
2. **Créer un jeu de données cible** : il faut sélectionner le jeu de données, ou se concentrer sur un sous-ensemble de variables ou de données sur lesquelles le processus de découverte doit avoir lieu.
3. **Effectuer le nettoyage des données** et la manipulation des jeux de données : pour permettre d'exploiter les données, il faut utiliser les opérations de base comme la suppression du bruit ou des valeurs extrêmes si nécessaire (selon les algorithmes), faire la collecte des informations nécessaires pour modéliser ou compenser pour le bruit, décider d'une stratégie pour gérer les champs de données manquants (vides) et compenser pour l'information qui change avec le temps ou autres types de changements possibles.
4. **Effectuer la projection des données** : trouver des caractéristiques intéressantes pour représenter les données selon le but ou la tâche à accomplir. Il est possible de réduire le nombre de dimensions ou d'utiliser des méthodes de transformation pour simplifier le nombre de variables à considérer en même temps ou pour trouver les invariants dans les données.
5. **Choisir la tâche de forage de données** : choisir si le but du processus de découverte est la classification, la régression, faire des clusters, des règles associatives, etc.
6. **Choisir les algorithmes de forage de données** : choisir les méthodes à être utilisées pour chercher pour des informations dans les données. Ceci inclus la décision du modèle et des paramètres appropriés et l'association d'une méthode de forage de données avec les critères du processus de découverte.
7. **Exécuter le forage de données** : rechercher l'information d'intérêt dans une forme représentationnelle particulière ou dans un ensemble de telles représentations : règles de classifications ou arborescences, régression, etc.
8. **Interpréter les informations** trouvées, une itération est possible sur les étapes 1 à 7.
9. **Consolider les connaissances découvertes** : incorporer les connaissances dans le système, ou documenter les connaissances et préparer un rapport aux partis impliqués. Il faut aussi s'assurer que les nouvelles connaissances n'entraînent pas de contradiction avec les connaissances déjà extraites ou connues.

Ces étapes sont des recommandations sur la façon dont devrait fonctionner un processus de forage de données complet. Sans entrer dans les détails, les étapes préparent le terrain pour la création d'une application ou de projets de forage de données dans des entreprises. Cependant, il faut noter que les résultats du forage de données dépendent des premières étapes, et que si le jeu de données n'a pas été consciencieusement sélectionné ou que si la personne qui effectue l'opération ne connaît rien au domaine à analyser, les résultats peuvent ne pas être significatifs.

La présentation des étapes du forage de données se voulait une introduction en la matière. Le reste de ce document ne traitera que de quelques algorithmes et la présentation de quelques produits commerciaux. Cependant, le forage de données et l'apprentissage automatique sont des domaines en pleine expansion et les applications qu'on peut en faire sont de plus en plus diversifiées.

Les prochaines sections présenteront les algorithmes de classification et de règles associatives. Quelques applications commerciales sont aussi présentées. Ainsi, c'est surtout la 6^{e} étape du processus de découverte de connaissance (l'exécution du forage de données) qui est traitée. Les autres étapes feront l'objet d'une étude plus approfondie dans un autre travail.

Algorithmes

Il existe plusieurs algorithmes utilisés dans le forage de données. Ces algorithmes peuvent être des classificateurs qui vont permettre de classer des éléments, de prévoir la classe des futurs éléments similaires à ceux déjà classés, et d'afficher le résultat des calculs de différentes façons. Ces algorithmes peuvent aussi analyser des problèmes du style « market-basket » où ils tentent de trouver des règles qui permettent d'expliquer quels types d'éléments se retrouvent le plus souvent en présence les uns des autres (par exemple, dans une épicerie, les clients qui achètent du pain achètent aussi du beurre dans 22% des cas).

Les algorithmes peuvent être de type semi-automatique (ou interactif) au automatique selon qu'ils ont besoin de l'intervention de l'utilisateur durant la phase d'analyse ou non.

Pour extraire l'information des données, diverses méthodes statistiques existent déjà, comme l'analyse de la variance et les régressions linéaires, pour n'en nommer que quelques-unes. Cependant, dans cet ouvrage, l'emphase sera mise sur les algorithmes se spécialisant dans le forage de données. Les méthodes statistiques et probabilistes sont intégrées dans plusieurs des algorithmes de forage de données (comme le réseau bayesien, les clusters). Il existe quelques algorithmes qui utilisent des principes différents, comme les algorithmes d'arbres de décision ID3 ou ID4.5 qui se servent de la théorie de l'information pour classifier les éléments. Et les réseaux neuronaux à base de perceptrons sont un autre type d'algorithmes de classification qui n'utilise pas les statistiques ou les probabilités.

Ces algorithmes ont tous leurs forces et leurs faiblesses : certains peuvent supporter le bruit, d'autres nécessitent que les données soient parfaitement nettoyées et que les irrégularités soient enlevées pour donner des résultats intéressants. Il y a des algorithmes qui peuvent générer des modèles explicatifs très puissants, mais demandent trop de ressources à l'ordinateur pour être utilisables (programmation logique inductive). Et encore d'autres sont capables de trouver des règles et de comprendre des régressions non linéaires, mais il est très difficile de comprendre comment ils y arrivent (réseaux neuronaux).

Les algorithmes décrits sont en fait des classes générales d'algorithmes qui peuvent servir au forage de données. Ces algorithmes ne sont pas spécifiques au forage de données et ont des utilités dans bien d'autres domaines, comme l'analyse statistique ou d'autres domaines l'intelligence artificielle.

Algorithmes de classification

Les algorithmes de classification servent à identifier des classes dans l'ensemble des éléments soumis, ou encore à pouvoir estimer la classe (à l'aide de différentes méthodes) d'un nouvel élément. Les algorithmes de classification utilisent souvent un jeu d'entraînement pour permettre de déterminer les classes de base auxquelles les nouveaux éléments devront être comparés. Un jeu d'entraînement se résume à plusieurs éléments auxquels un expert a associé une classe (par exemple, vrai - faux, petit – moyen – grand, etc.). Il existe de nombreux algorithmes de classifications avec différentes caractéristiques et qui s'appliquent à plusieurs types de problèmes. Certains algorithmes peuvent traiter des attributs continus (nombre de 0 à 10, -100 à 2000), d'autres ne peuvent prendre que des valeurs discrètes (chiffres, mots, classes, ...).

Programmation logique inductive

La programmation logique inductive utilise la logique des prédicats pour trouver des nouveaux concepts. Elle procède par la description de concepts « émergents » qui peuvent être redéfinis, généralisés, spécialisés, etc. Ces concepts sont représentés dans une grammaire particulière qui doit être définie au moment de la programmation. Il est possible d'utiliser des langages existants, comme PROLOG, pour faire des algorithmes en programmation logique inductive. Le gros problème de ce type de programmation est la lourdeur du processus de création de concepts. Il n'est pas possible de stocker toute l'information en mémoire, et l'efficacité des langages et des applications qui intègrent la logique des prédicats n'est pas très grande par rapport au matériel informatique.

Fiche de l'algorithme

Intrants :	Cet algorithme utilise des tuples identifiés comme étant des exemples positifs ou négatifs d'une relation à apprendre. Ces tuples doivent être sous forme de table (enregistrements, attributs). Il faut aussi utiliser un langage de description de concept. Finalement, il est possible d'introduire des connaissances du domaine (background knowledge) avant de démarrer l'algorithme.
Extrants :	Les définitions de nouveaux prédicats (nouveaux concepts) qui peuvent parfois être des généralisations d'un concept sont fournis en sortie. Les résultats peuvent être compréhensibles directement, ou subir une transformation pour donner de vraies règles sur les données qui peuvent être comprises par une personne.

Traitement (ressources, complexité) : La complexité et les ressources utilisées sont très élevées. Tous les éléments en entrée doivent être analysés et comparés aux concepts déjà connus, et aux concepts qu'il est possible d'extraire. Le langage utilisé est généralement peu efficace avec les accès sur le disque (non séquentiels), et la quantité de mémoire nécessaire est très grande.

Désavantages : La programmation logique inductive est très lourde pour l'ordinateur. Le stockage de tous les concepts et le calcul et la généralisation de nouveaux concepts prend aussi beaucoup de temps (complexité algorithmique élevée).

Algorithme :

Exemple de FOIL [MLDM98]:

1. Initialise la clause en définissant la tête qui représente le nom du concept à être appris et laisser le corps vide.
2. Pendant que la clause couvre les exemples négatifs, faire : « trouve un bon littéral à être ajouté au corps de la clause ».
3. Enlever tous les exemples couverts par la clause.
4. Ajouter la clause à la définition du concept émergeant. S'il existe des exemples positifs qui ne sont pas encore couverts, recommencer à l'étape 1.

Exemples :

Ancetre(X, Y) :- parent(X, Y).
Ancetre(X, Y) :- parent(X, Z), parent(Z, Y).
Ancetre(X, Y) :- parent(X, Z), parent(Z, W), parent(W, Y).
(un concept qui devra être généralisé...)

Conclusion : Cet algorithme est intéressant puisqu'il génère des nouvelles lois, ou concepts qui peuvent ensuite être compris, vérifiés et validés par une personne. Il pourrait par contre ne pas être applicable à cause d'exigences élevées en ressources. Il faut absolument utiliser un petit échantillon de données (représentatif si possible).

Arbre de décision

Les arbres de décisions sont de bons algorithmes qui permettent de classifier les éléments dans une structure qui peut servir à la fois de résumé pour la compréhension de l'information et de méthode de prévision pour la classification de nouveaux éléments. Un jeu de données déjà classifiées est nécessaire pour créer un arbre. Il est ensuite facile de récupérer l'arbre et d'afficher l'information que l'algorithme a utilisé pour le créer. Cet algorithme est capable de traiter de très grandes quantités de données, qu'elles soient stockées en mémoire ou dans un fichier (lecture séquentielle).

Fiche de l'algorithme

Intrants : L'algorithme utilise un jeu de données avec des exemples positifs et négatifs (déjà classé). Il est capable de gérer les exemples où les attributs sont manquants (C4.5) en comparant le gain d'information par rapport aux exemples où ces attributs sont présents. Il peut identifier des classes à partir d'attributs continus (C4.5), et il peut aussi utiliser des attributs discrets.

Extrants : L'algorithme fournit une arborescence. C'est un arbre de décision (ou de classification) représentant la situation générale des observations. Cet arbre peut être visualisé directement sans traitement majeur.

Traitement (ressources, complexité) : Un arbre peut généralement tenir en mémoire. Pour s'en assurer, il est possible d'utiliser du « pruning » pour couper des sections qui se répètent. Le calcul de sélection des attributs pour chaque niveau se résume au calcul de l'information fournie par les attributs.

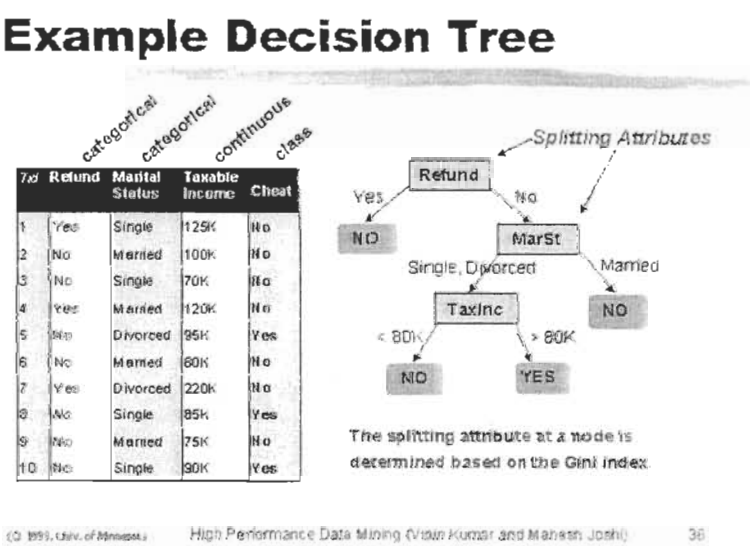
Désavantages : Bien que les arbres de décisions soient relativement faciles à créer et demande peu de ressources, si le nombre de nœuds est trop grand, l'information sera difficile à visualiser et à comprendre par l'utilisateur. Le même problème survient s'il y a un trop grands nombres d'attributs, ou de niveaux. Il faut donc faciliter la visualisation des arbres dans ces situations, ou trouver des alternatives.

Algorithme (ID3) [CLAS94] :

```
function ID3 (R: a set of non-categorical attributes,
              C: the categorical attribute,
              S: a training set) returns a decision tree;
begin
    If S is empty, return a single node with value Failure;
    If S consists of records all with the same value for
      the categorical attribute,
      return a single node with that value;
    If R is empty, then return a single node with as value
      the most frequent of the values of the categorical attribute
      that are found in records of S; [note that then there
      will be errors, that is, records that will be improperly
      classified];
    Let D be the attribute with largest Gain(D,S)
      among attributes in R;
    Let {dj| j=1,2, .., m} be the values of attribute D;
    Let {Sj| j=1,2, .., m} be the subsets of S consisting
      respectively of records with value dj for attribute D;
    Return a tree with root labeled D and arcs labeled
      d1, d2, .., dm going respectively to the trees

        ID3(R-{D}, C, S1), ID3(R-{D}, C, S2), .., ID3(R-{D}, C, Sm);
end ID3;
```

Exemples :



[KJ00]

Conclusion : Les algorithmes d'arbres de décision répondent aux critères requis : traiter des grandes quantités de données et permettre à un utilisateur de comprendre les résultats. Cependant, les arbres générés peuvent parfois être très complexes et incompréhensibles pour une personne. Il faudrait trouver une bonne méthode ou un bon outil pour résumer l'information (ou connaissances) identifiées, autrement dit un outil de visualisation d'arbres de décisions. Ces outils existent (programmes commerciaux ou gratuits) et devront être combinés à l'utilisation des algorithmes sur les arbres de décision.

Algorithmes génétiques

Les algorithmes génétiques présentent une nouvelle façon de poser des problèmes et de chercher des solutions. Une des difficultés dans l'utilisation de ces algorithmes est l'identification d'une méthode de codage pour les valeurs en entrée. Ce codage doit produire des chromosomes qui peuvent être utilisés en entrée pour les mutations et le croisement. Il faut trouver une fonction d'évaluation qui pourrait déterminer les meilleurs individus. Il faut aussi trouver une fonction de répartition qui fait le croisement des individus pour obtenir les nouvelles générations. Les algorithmes génétiques peuvent converger très rapidement en ajustant les fonctions d'évaluation et de répartition.

Fiche de l'algorithme

Intrants : Les algorithmes génétiques utilisent des données qui seront recodées par une fonction de codage. Les gènes ainsi formés seront utilisés pour trouver la solution recherchée, ou faire la classification des éléments. Le codage peut être binaire. Par exemple, la présence de la variable A est codée en un 1 et son absence en un 0, on pourrait avoir le code « 1000 » pour dire que A est présent et que B, C, et D sont absents, ou « 0010 » pour dire que seul C est présent. Une des grandes difficultés des algorithmes génétiques est d'obtenir un codage représentatif des données. Il faut aussi préparer une fonction d'évaluation (quels individus sont des réponses acceptables à la question) et une fonction de répartition (croisement des individus, élitisme, ...).

Extrants : L'algorithme retourne un ensemble de solutions (individus) qui répondent à la question, dont une est la meilleure solution trouvée à cette génération. Ces solutions doivent être décodées pour être utilisées.

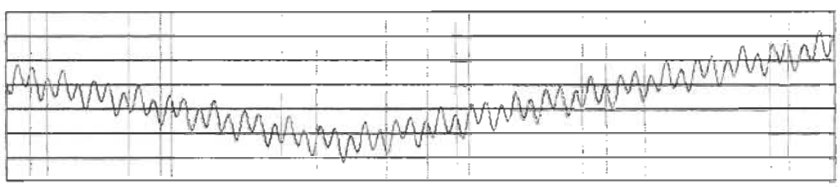
Traitement (ressources, complexité) : La complexité des algorithmes génétiques dépend du nombre d'individus qui sont utilisés, de la taille de ces individus (des gènes), du nombre de générations qui sont nécessaires pour trouver la bonne solution et de la complexité algorithmique des fonctions d'évaluation et de répartition.

Désavantages : Il est difficile de trouver une fonction de répartition appropriée. Cette fonction doit assurer une convergence (éventuelle) vers la réponse. Il faut aussi trouver le moyen de « coder » l'information, ou la solution suggérée sous forme d'individus à croiser et à évaluer.

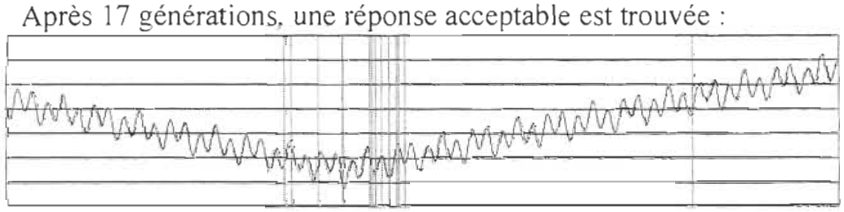
Algorithme :

1. **[Start]** Generate random population of n chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

Exemples : Soit une fonction mathématique quelconque (en bleu). On veut trouver le minimum de cette fonction. Les chromosomes sont des nombres, représentés sous forme binaire. Les barres pales sont des chromosomes qui font partie de la génération courante. La barre foncée est le chromosome qui possède la meilleure réponse par rapport à la fonction recherchée (le minimum). Pendant 17 générations, les chromosomes sont combinés selon une fonction de répartition. L'élitisme est en vigueur : le meilleur individu (barre foncée) survit et est copié dans la prochaine génération.



Après 17 générations, une réponse acceptable est trouvée :



Dans l'exemple, un minimum local a longtemps été la meilleure réponse. http://cs.felk.cvut.cz/~xobitko/ga/gaintro.html

Conclusion : Les algorithmes génétiques permettent de trouver des solutions par rapport à des problèmes complexes. Leur expressivité est seulement limitée à l'imagination de l'utilisateur par rapport à la méthode de codage appliquée pour générer les individus. L'utilisation des algorithmes génétiques dans une application de forage de données serait sûrement très intéressante, à condition de formuler correctement la façon de représenter l'information et de la traiter à l'aide de fonctions d'évaluation, de répartition et de codage des données ou attributs.

Réseaux neuronaux

Les réseaux neuronaux sont des classificateurs utilisés dans de nombreuses applications, et dans des domaines très divers (langue naturelle, traitement d'image, analyse statistique, etc.). Ils sont capables de traiter des règles non linéaires et de gérer élégamment les éléments avec de nombreuses dimensions (attributs). Les réseaux neuronaux se programment à la fois en déterminant le nombre de cellules à utiliser dans les couches d'entrée, de sortie et cachées, et en apprenant par des exemples (algorithme à propagation arrière, ...). Ils sont suffisamment efficaces pour traiter de grandes quantités de données. Il faut s'assurer d'avoir un réseau suffisamment grand (nombre de cellules, couches cachées) pour pouvoir apprendre tout ce qui doit être traité. Deux problèmes peuvent survenir si les réseaux sont trop grands ou trop petits : un réseau trop grand peut mémoriser chaque exemple et avoir de la difficulté à classer des nouveaux exemples; un réseau trop petit peut commencer à régresser dans son efficacité avec un nombre d'exemples croissants. Il faut donc savoir faire des essais et trouver le juste milieu. Un désavantage des réseaux neuronaux est qu'il est difficile d'aller récupérer l'information qu'ils ont « apprise ».

Fiche de l'algorithme

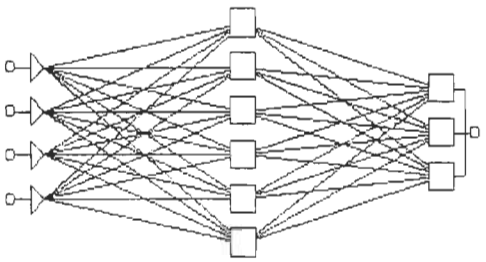
Intrants : La préparation des données est exigeante. L'utilisateur doit avoir une connaissance des méthodes pour préparer les données et comment sélectionner un réseau neuronal approprié (nombre de nœuds à l'entrée, couches cachées, sorties, type de connexions, poids initiaux). Des modèles prédéfinis déjà testés sur différents types d'applications peuvent toujours être proposés. Les attributs peuvent être continus ou discrets, mais les réseaux neuronaux fonctionnent mieux avec les attributs continus.

Extrants : L'utilisateur doit savoir interpréter les résultats. De plus, le réseau neuronal ne produit pas de « règles » ou de résultats en soit, il faut plutôt avoir une technique pour aller voir à l'intérieur du réseau pour en tirer l'information apprise.

Traitement (ressources, complexité) : Les réseaux neuronaux sont plutôt efficaces et rapides. Une de leurs particularités est qu'ils doivent être entraînés à partir de jeux de données soigneusement préparés pour apprendre directement des données. Généralement, le nombre de nœuds n'est pas énorme et le réseau peut tenir en mémoire. Il est possible de faire une seule passe, ou plusieurs passes avec des jeux de données lors de la période d'apprentissage.

Désavantage : Une saturation du réseau peut survenir. Le nombre de jeux en entrée n'apporte plus de nouvelles informations au réseau. Ou pire, il est possible que les capacités de prédiction des réseaux diminuent avec un plus grand nombre d'exemple. L'information est difficile à extraire du réseau, il sert surtout à classer des nouveaux exemples.

Exemples : Réseau feedforward simple



[STATSOFT]

Dans cet exemple, on a un réseau avec 4 neurones à la couche d'entrée, une seule couche cachée de 6 neurones et trois neurones à la couche de sortie.

Conclusion : Vue la possibilité d'utiliser les réseaux neuronaux sur un grand nombre d'attributs et de données, il serait intéressant de les utiliser dans l'application. Ils pourraient même servir sous plusieurs formes (comme support pour d'autres algorithmes, par exemple). Cependant, les réseaux neuronaux peuvent être configurés de plusieurs façons, ne serait-ce que par le nombre de cellules à l'entrée et à la sortie. C'est pourquoi il faudrait s'assurer que l'utilisateur puisse s'y retrouver. Il serait aussi essentiel d'intégrer un algorithme pouvant explorer et retirer l'information contenue dans le réseau neuronal pour la présenter à l'utilisateur. Plusieurs produits commerciaux supportent déjà les réseaux neuronaux.

Classificateurs bayesiens

Les réseaux bayesiens font une classification des éléments à partir de la probabilité d'occurrence des attributs. Toutes les combinaisons des attributs possibles ne sont cependant pas considérées (si la probabilité conditionnelle est de 0, l'information n'est pas conservée en mémoire). Ce type d'algorithme est cependant plutôt difficile à comprendre. Bien que les probabilités conditionnelles ne soient pas en soit une notion compliquée, le nombre de combinaisons qui sont conservées en mémoire peuvent être difficiles à évaluer lorsque vient le temps de comprendre comment l'algorithme fait sa classification. Aussi, une application ou un algorithme qui permet de récupérer et de formuler sous forme d'arborescence (ou autre) les classifications faites par l'algorithme bayesien serait une nécessité pour l'utilisation de cet algorithme.

Fiche de l'algorithme :

Intrants : L'algorithme a besoin d'un jeu d'entraînement (mode assisté), et d'un jeu de test pour vérifier. Il se base sur l'indépendance des attributs (classificateur bayesien naïf), les probabilités sont utilisées.

Extrants : L'algorithme calcule la classification d'un nouvel élément, et retourne cette classe.

Traitement (ressources, complexité) : L'algorithme doit créer une matrice de toutes les probabilités en mémoire. Il peut croître rapidement, mais des méthodes dans les algorithmes propriétaires permettent de limiter la quantité de mémoire requise puisque les matrices sont éparpillées (beaucoup de cellules sont vides).

Désavantages : Il faut assumer l'indépendance entre les divers attributs de l'élément (pour le calcul de la probabilité) pour l'algorithme Naïve Bayes. Si les éléments ne sont pas indépendants, la qualité de la classification est affectée.

Bayesian Classifiers

Each attribute and class label are random variables.

Objective is to classify a given record of attributes (A_1, A_2,...,A_n) to class C s.t. $P(C \mid A_1, A_2, \ldots, A_n)$ is maximal.

Naïve Bayesian Approach:

- Assume independence among attributes A_i
- Estimate $P(A_i \mid C_j)$ for all A_i and C_j
- New point is classified to C_j if $P(C_j) \prod_i P(A_i \mid C_j)$ is maximal.

Generic Approach based on Bayesian Networks:

Represent dependencies using a direct acyclic graph (child conditioned on all its parents). Class variable is a child of all the attributes.

Goal is to get compact and accurate representation of the joint probability distribution of all variables. Learning Bayesian Networks is an active research area.

[KJ00]

Conclusion : Comme les réseaux neuronaux, les réseaux ou classificateurs bayesiens ont besoin d'un jeu de test, donc c'est une méthode assistée (semi-automatique). Cet algorithme peut être utile lorsque l'on cherche à prévoir des nouveaux cas à partir des cas existants. Pour les réseaux neuronaux, des algorithmes existent pour extraire l'information acquise et l'exprimer à l'aide d'arbres de décision ou autres représentation. S'il est possible de trouver des algorithmes similaires pour les classificateurs bayesiens, leur application pourrait être considérée.

K Nearest Neighbour

L'algorithme du K Nearest Neighbor est plutôt simple et efficace à cause de la méthode utilisée pour la classification. Il suffit de prendre une formule qui va donner une valeur (un angle à partir d'un vecteur d'attributs), et trouver à quels éléments en mémoire cette valeur s'approche le plus. Cependant, cet algorithme est assez limité quand vient le temps de traiter des données très bruitées, et ce même s'il peut supporter les données manquantes.

Fiche de l'algorithme

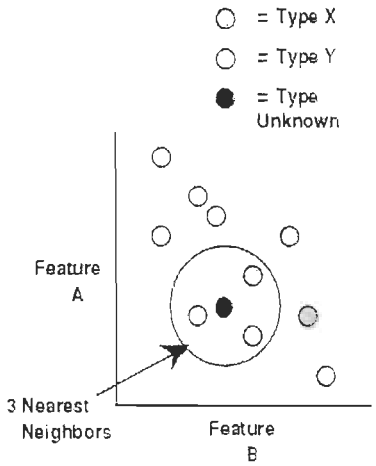
Intrants : L'algorithme utilise les éléments sous forme de vecteurs à classifier. Il faut aussi préciser le nombre de voisins les plus proches à consulter pour la classification (les K Nearest Neighbor). Et comme pour la plupart des algorithmes de classification, il faut un ensemble d'entraînement déjà classifié.

Extrants : L'algorithme retourne une classification pour les nouveaux éléments. Cet algorithme est souvent utilisé pour calculer des clusters, donc les résultats sont calculés directement à l'intérieur de l'algorithme de clusters et la sortie est un ensemble d'éléments classés.

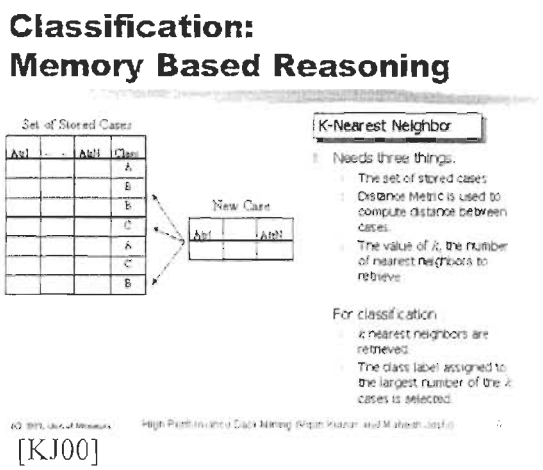
Traitement (ressources, complexité) : L'algorithme mémorise les vecteurs d'entraînement. Le nombre d'exemples doit tenir en mémoire, mais ces exemples sont sous forme de vecteurs et prennent très peu de place. Les attributs devraient être de type continus pour les meilleurs résultats, mais un codage d'attributs discrets peut être employé.

Désavantages : L'algorithme KNN est très sensible aux attributs non significatifs (par exemple, des attributs dépendants d'un autre). L'ensemble d'entraînement doit être soigneusement choisi car il est très important de couvrir l'ensemble des positions dans les classes pour obtenir une classification fiable.

Exemples :



[PUN93]



[KJ00]

Conclusion : L'algorithme KNN permet de prévoir la classification d'un nouvel élément. L'apprentissage se fait à l'aide d'un ensemble d'entraînement, et la classification se fait à l'aide d'une métrique (distance entre les cas de base). Cet algorithme est très simple, et peut être utilisé facilement sur de très gros ensembles de données. Cet algorithme peut être combiné assez facilement avec d'autres algorithmes (exemple : réseau d'entropie [MLDM98], algorithmes génétiques [PUN93]). L'utilisation de l'algorithme brut ne sera peut-être pas utile dans l'application puisque les capacités d'apprentissage non linéaire d'un réseau neuronal ou encore l'expressivité des arbres de classification sont plus adaptés. Cependant, la combinaison d'un algorithme KNN avec un réseau neuronal, un algorithme génétique ou tout autre algorithme est fortement suggérée.

Les clusters

Les méthodes statistiques peuvent être utilisées pour analyser les données. Elles peuvent servir à confirmer ou infirmer des tendances identifiées par les algorithmes de forage de données[18]. Il y a plusieurs méthodes statistiques comme l'analyse de la variance (ANOVA), la régression linéaire, les clusters, etc. Cependant, seuls les clusters ont été retenus puisqu'ils peuvent servir d'algorithme d'analyse automatique (non assistée) de données. En fait, cette technique est souvent utilisée pour se familiariser les données avant même de lancer des algorithmes de forage de données.

Il existe plusieurs algorithmes permettant de faire des regroupements par cluster (i.e. KNN, K-Means, EM... [KNST]). Ces algorithmes comparent les nouvelles données à celles qui sont déjà classées en calculant l'angle entre les vecteurs ou en utilisant des méthodes probabilistes.

Fiche de l'algorithme :

Intrants : Les clusters utilisent un ou plusieurs attributs d'une base de données (des tuples). Il est possible de donner à l'avance le nombre de classes à trouver, ou la distance maximale entre les éléments pouvant appartenir à une classe.

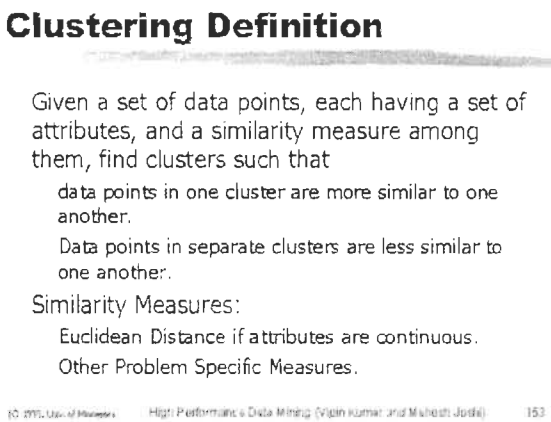
Extrants : L'algorithme produit une liste de classes avec les éléments qui s'y retrouvent.

Traitements (ressources, complexité) : Souvent, les algorithmes de clusters sont implémentés dans des programmes de traitement statistique et sont faits pour pouvoir travailler de manière efficiente sur de très grands ensembles de données (voir SAS, KnowledgeStudio). Les algorithmes de classification à l'intérieur des clusters (K-Means, KNN, ...) sont très rapides et utilisent peu de mémoire.

Désavantage : Il arrive qu'il ne soit pas réellement possible de distinguer les éléments à partir des dimensions évaluées. Dans ce cas, les classifications n'auront pas vraiment de sens. Avec certains algorithmes, il faut même parfois que l'utilisateur connaisse le nombre de classifications qu'il faudrait trouver dans le jeu de données. L'algorithme n'explique pas les raisons de ses classifications, l'utilisateur doit les expliquer lui-même.

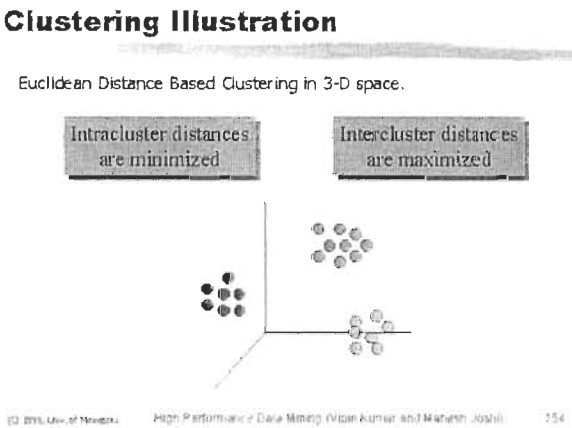
[18] Voir John F. Elder, Daryl Pregibon, A Statistical Perspective on Knowledge Discovery in Databases. [ADV96]

Exemples de clusters :



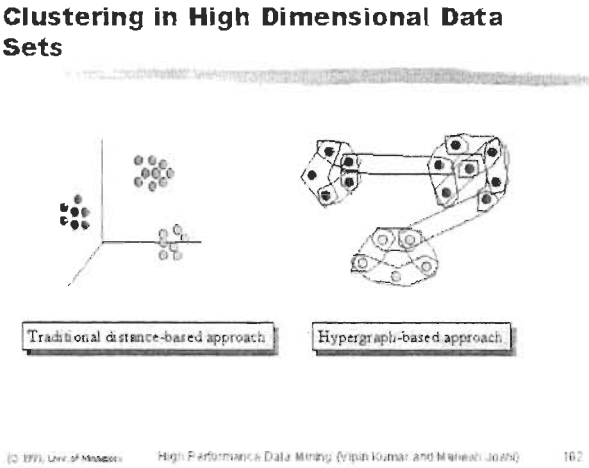
[KJ00]

Exemple de représentation spatiale des points:



[KJ00]

Deux types de clusters



[KJ00]

Conclusion : Les clusters sont de bons algorithmes qui peuvent être couplés à d'autres algorithmes de forage de données. De plus, ils sont relativement faciles à utiliser, comportent peu de paramètres essentiels (voir aucun) et sont capables d'utiliser des données continues ou discrètes (et même parfois les deux en même temps dans un même attribut). L'utilisation des clusters serait donc particulièrement recommandée.

Algorithmes à base de règles associatives

Les algorithmes à base de règles associatives permettent de découvrir des relations plus fines du genre : si X alors Y. Ces algorithmes devraient calculer le support et la confiance qui sont associés aux règles trouvées. Des paramètres permettent d'ajuster le niveau de support et de confiance qui sont requis pour indiquer qu'une règle est intéressante et doit être retenue. Un ensemble avec un support qui dépasse la limite inférieure déterminée par l'utilisateur est dit « fréquent ».

Exemple [QUEST]:
Given a database of transactions, where each transaction consists of a set of items, discover all associations such that the presence of one set of items in a transaction implies the presence of another set of items.

``30% of people who buy diapers also buy beer."

Association Rule Discovery: Support and Confidence

TID	*Items*
1	Bread, Milk
2	Beer, Diaper, Bread, Eggs
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Bread, Diaper, Milk

Association Rule: $X \Rightarrow_{s,\alpha} y$

Support: $s = \frac{\sigma(X \cup y)}{|T|} (s = P(X, y))$

Confidence: $\alpha = \frac{\sigma(X \cup y)}{\sigma(X)|} (\alpha = P(y|X))$

Example:

$\{Diaper, Milk\} \Rightarrow_{s,\alpha} Beer$

$s = \frac{\sigma(Diaper, Milk, Beer)}{\text{Total Number of Transactions}} = \frac{2}{5} = 0.4$

$\alpha = \frac{\sigma(Diaper, Milk, Beer)}{\sigma(Diaper, Milk)|} = 0.66$

 High Performance Data Mining (Vipin Kumar and Mahesh Joshi) 108

A Priori

Cet algorithme prépare des listes de tous les éléments dans une base de données et compte le nombre de fois que ces éléments sont en présence les uns avec les autres. Il calcule des informations du type « *market-basket* ». L'algorithme A Priori permet de répondre à d'autres types de questions que les algorithmes de classification. Cet algorithme identifie les éléments qui se retrouvent souvent associés à d'autres éléments, à partir de conditions déterminées par l'utilisateur. Ce type d'information peut servir pour complémenter des classifications faites par d'autres algorithmes, ou comme indicateurs pour de toutes nouvelles informations. De plus, l'algorithme se prête très bien au traitement de très vastes ensembles de données puisque son utilisation de la mémoire est relativement limitée et qu'il peut utiliser un accès séquentiel à un disque sans accuser de ralentissement exagéré. Il existe aussi un autre algorithme dérivé du nom de AprioriAll proposé par l'équipe Quest d'IBM qui permet de trouver des règles d'association séquentielles [JOS97].

Fiche de l'algorithme :

Intrants : L'algorithme utilise une table avec un attribut discret (mots, classes, etc.) et un attribut d'identification de « transaction » (ou de panier, une entreprise, etc.).

Extrants : Une liste d'éléments sous forme de couples, triplets, etc. est calculée avec une mesure de support et de confiance.

Traitement : Cet algorithme peut travailler à partir des disques (très efficace). Normalement, les résultats peuvent tenir en mémoire (si le support exigé n'est pas trop faible). Des listes des éléments ayant un support et une confiance suffisants sont conservées en mémoire.

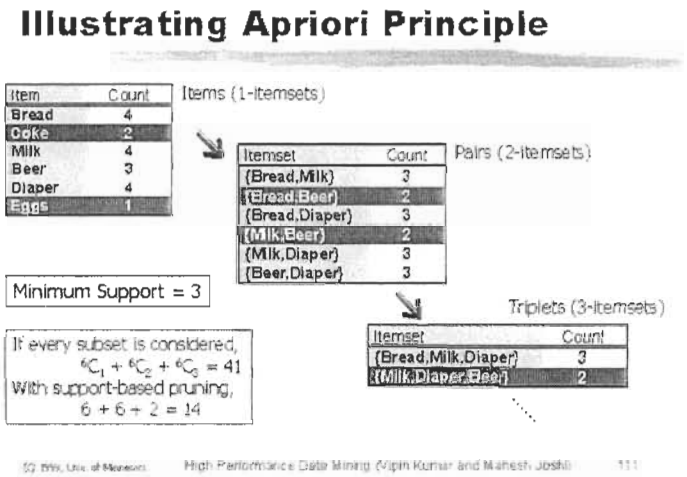
Désavantages : L'algorithme A Priori est conçu pour travailler avec des données de type transactionnelles. Il est possible que de nombreuses manipulations de données soient nécessaires si l'entrepôt de données n'a pas été conçu pour faciliter le traitement de ce type de données. La notion de support est parfois difficile à maîtriser. Aussi, lorsque le support (paramètre) est faible, il peut arriver qu'un nombre très important de règles soit trouvé et que la capacité de la mémoire ne soit plus suffisante. D'ailleurs, la notion de support est contestée par certains chercheurs [CDF00].

Algorithme :

[JOS97]

```
procedure AprioriAlg()
begin
        L1 := {frequent 1-itemsets};
        for ( k := 2; Lk-1 0; k++ ) do {
           Ck= apriori-gen(Lk-1) ; // new candidates
        for all transactions t in the dataset do {
          for all candidates c Ck contained in t do
                c:count++
          }
          Lk = { c Ck | c:count >= min-support}
        }
        Answer := k Lk
End
```

Exemples :



[KJ00]

Apriori Algorithm

F_1 = {frequent 1-item sets};
k = 2;
while(F_{k-1} is not empty) {
    C_k = Apriori_generate(F_{k-1});
    for all transactions t in T {
        Subset(C_k, t);
    }
    F_k = { c in C_k s.t. c.count >= minimum_support};
}
Answer = union of all sets F_k;

[KJ00]

Association Rule Discovery: Apriori_generate

Apriori_generate(F(k-1)) {
    join F_{k-1} with F_{k-1} such that.
        $c_1 = (i_1, i_2, \ldots, i_{k-1})$ and $c_2 = (j_1, j_2, \ldots, j_{k-1})$ join together if
            $i_p = j_p$ for $1 <= p <= k-1$.
    and then new candidate, c, has a form
        $c = (i_1, i_2, .., i_{k-1}, j_{k-1})$.
    c is then added to a *hash-tree* structure.
}

[KJ00]

Conclusion : L'algorithme A Priori permet d'aller chercher des informations qui sont complémentaires au type d'information fournie par les algorithmes de classification. De plus, cet algorithme peut facilement être appliqué à de très vastes quantités de données. Pour ces raisons, il serait très intéressant d'intégrer l'algorithme A Priori (ou une version améliorée) dans l'application de forage de donnée.

Produits commerciaux

Il existe déjà de nombreux produits commerciaux pour faire du forage de données. Les plus grosses compagnies de base de données (Oracle, IBM, Microsoft) ont déjà des produits depuis quelques années qui fonctionnent avec leurs logiciels. Mais il existe aussi d'autres compagnies qui font des suites complètes pour le forage de données qui supportent toutes les étapes comme l'acquisition des données d'un entrepôt, le nettoyage, l'analyse, le forage de données lui même, la visualisation des résultats, etc.

Certains produits supportent même une interface de programmation qui permet de réutiliser les interfaces de visualisation de données ou les algorithmes de forage de données pour développer des applications personnalisées. Ces produits supportent généralement plusieurs types de sources de données et sont disponibles sur plusieurs plateformes.

KnowledgeSTUDIO

Produit : KnowledgeSTUDIO

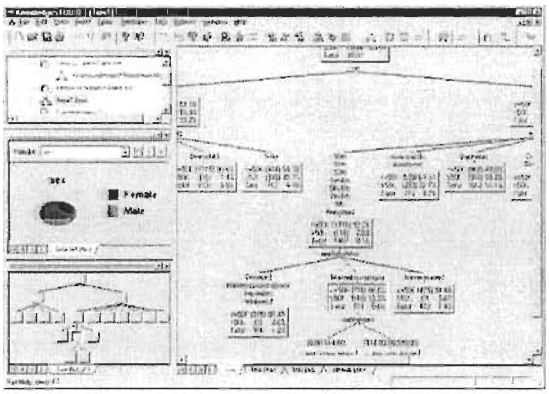
Compagnie : Angoss

Site web : http://www.angoss.com/ProdServ/AnalyticalTools/index.html#seeker

Algorithmes: arbres de decision, réseaux neuronaux (perceptron, probabilistique), clusters (K-Means, Expectation Maximisation).

Plateforme : Microsoft Windows. SDK en contrôles ActiveX. Serveur HP-UX, Solaris.

Description : Suite de data mining complètement modulaire, avec un SDK pour pouvoir utiliser les algorithmes dans d'autres produits (C++ et VB sous Windows, Java dans le futur). Possède plusieurs algorithmes avec quelques améliorations propriétaires aux réseaux neuronaux (apprentissage).



Source de données : ASCII, dBase, Excel, ODBC, SAS, ...

Évaluation : Ce logiciel est une suite complète qui semble très prometteuse. La compagnie Angoss est basée à Toronto et fabrique divers produits de forage de données. Une version serveur est disponible sur différentes plateformes pour les gros projets de forage de données. L'interface facilite toutes les étapes du forage de données, de l'exploration des données à l'interprétation des résultats. Plusieurs algorithmes sont disponibles, de nombreux formats de données peuvent être utilisés à l'importation ou l'exportation des données et une version d'essaie peut être téléchargée du site.

Conclusion : KnowledgeSTUDIO est le genre d'application de développement qui peut aider un effort de forage de données. De nombreux algorithmes de forage de données sont disponibles, et des écrans de visualisation des données (arbres, clusters, réseaux neuronaux, ...) permettent d'analyser et de comprendre les résultats. Ce logiciel est recommandé pour l'application de forage de données à développer.

KWiz

Produit : KWiz

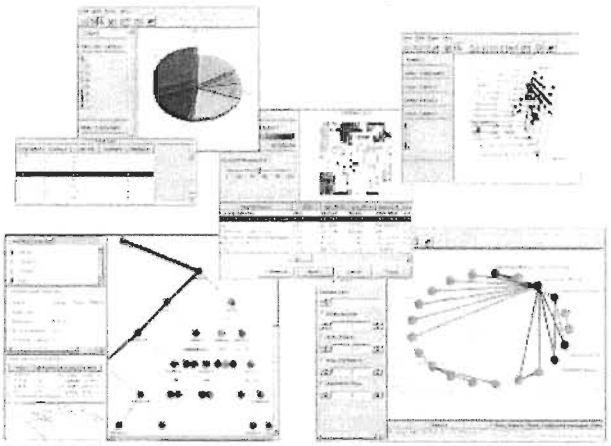
Compagnie : ThinkAnalytics

Site web :
http://www.thinkanalytics.com/products/factsheets/Kwiz_product_brief.htm

Algorithmes: Analyse statistiques, régressions.

Plateforme : Windows, Solaris 2.6/2.7, HP-UX 11.0.

Description: Fournit des modules qui peuvent être utilisés dans une application. Supporte l'architecture client/serveur est les applications web. Possède une partie serveur et une partie client.



Source de données: DB2 (IBM), Oracle, Microsoft SQL Server.

Évaluation : Le logiciel KWiz offre de nombreux outils d'analyse statistique pour faire le traitement à partir d'un logiciel serveur qui est relié à une source de données (base de données). De nombreux composants offrent la possibilité de préparer une application web complète très facilement, avec de nombreux graphiques et rapports.

Conclusion : Ce produit se dit être une application de forage de données. Les informations contenues dans les documents ne semblent pas en accord avec cette affirmation. Le logiciel a plutôt l'apparence d'un logiciel de statistique en version web avec beaucoup d'options de visualisation en ligne. La version d'évaluation pourrait tout de même être téléchargée pour vérifier si ce produit offre vraiment des

possibilités de forage de données. Le produit KWiz ne semble pas être intéressant pour l'application à développer.

Statistica

Produit : Statistica Data Miner

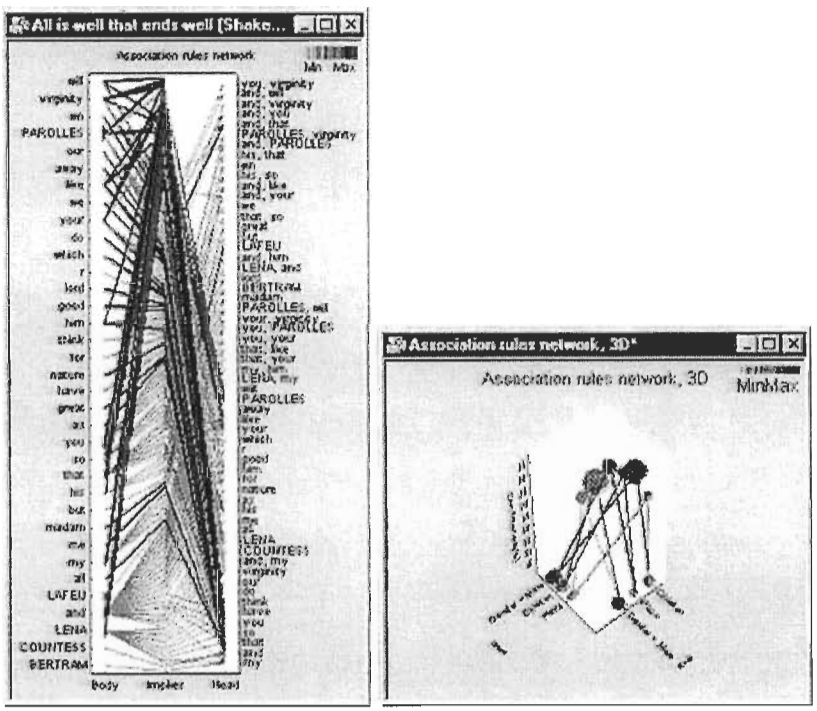
Compagnie : Statsoft

Site web : http://www.statsoft.com/dataminer.html

Algorithmes : réseaux neuronaux, arbres de regression et de classification. modèles multivariés, règles associatives.

Plateforme : Windows

Description : Statistica Data Miner supporte une architecture client/serveur ou application de bureau. Le logiciel possède aussi des composants réutilisables en Visual Basic, C++ ou Java. Les algorithmes sont optimisés pour de très grands ensembles de données. Le produit est conçu pour deux types d'utilisateurs : ceux qui ont besoin d'une solution complète et intégrée, prête à être utilisée, et les utilisateurs qui désirent une application puissante de forage de données pour faire du développement de solutions logicielles personnalisées. De nombreuses interfaces sont disponibles pour préparer et visualiser les données. WebStatistica permet d'exécuter des procedures Statistica par le web (incluant les procédures de forage de données). Permet aussi de faire du « OLAP ».



Source de données : non disponible.

Évaluation : Le logiciel Statistica sert surtout à faire de l'analyse statistique. Le module de forage de données comporte de nombreux algorithmes optimisés pour les très grands jeux de données. Le produit supporte la programmation Visual Basic directement avec son interface, permettant de développer des solutions personnalisées.

Conclusion : Le logiciel semble adapté à l'application de plusieurs types d'algorithmes de forage de données, tout en permettant de développer de nouvelles solutions personnalisées en Visual Basic. Une version de démonstration du module de forage de données n'est cependant pas disponible, alors l'évaluation pratique du produit pourrait être difficile avant l'achat. Ce logiciel est tout de même prometteur et devrait être considéré dans l'application de forage de données à développer.

Oracle Data Warehousing

Produit : Oracle9iDB Data Mining

Compagnie : Oracle

Site web : http://www.oracle.com/ip/index.html?dw_intro.html

Algorithmes: Classificateur bayesien naïf, arbre de décisions, règles associatives, clusters (K-Means, probabilistique).

Plateforme : Windows, Linux, AIX, HP-UX, Solaris.

Description: La composante de forage de données du serveur Oracle est intégrée à l'ordinateur qui sert de serveur. Tout comme le serveur Oracle de base, le module de forage de données peut profiter d'un environnement multi-processeurs. De nombreux types d'algorithmes sont disponibles, et une interface de programmation en Java permet de préparer des applications personnalisées. Des chercheurs de modèles et d'attributs peuvent identifier les variables clés et de déduire le meilleur modèle de forage de données à utiliser.

Source de données: base de données Oracle (9i)

Évaluation : Le module de forage de données pour la base de données Oracle 9i ne peut pas fonctionner avec d'autres sources de données. Les architectures parallèles sont supportées, et les algorithmes sont optimisés pour des calculs sur de très grands ensembles de données. Une option permet d'automatiquement identifier les attributs clés dans les jeux de données. Cependant, aucune interface utilisateur ne semble être disponible. Il faut donc développer une solution personnalisée (application exécutable, web, applet Java, etc.). Mais avec un support pour le standard émergent Java Data Mining, de nombreuses applications pourront être développées et interfacées à la composante de forage de données d'Oracle.

Conclusion : Le module Oracle9DB Data Mining n'offre pas d'interface utilisateur, seulement une interface de programmation avec de nombreux algorithmes. Cette approche correspond à la façon de procéder avec Oracle, qui fournit un logiciel de bases de données avec un tout petit utilitaire pour travailler (SQL-Plus). Puisque le but du projet d'application est de développer un logiciel de forage de données, la possibilité d'utiliser une interface Java avec de nombreux algorithmes optimisés disponibles est un avantage. Le produit Oracle9DB Data Mining est recommandé pour le développement de l'application de forage de données.

SAS

Produit : SAS Analytic Intelligence, SAS Enterprise Miner, SAS Warehouse Administrator

Compagnie : SAS Institute Inc.

Site web : http://www.sas.com

Description : Le logiciel SAS permet de faire des analyses de données avec des statistiques. Le logiciel SAS supporte à la fois la création d'entrepôts de données et des algorithmes pour faire le forage de données. Il supporte aussi la création et l'entretien d'entrepôts de données à partir de diverses bases de données, et l'exploitation de l'entrepôt. Il possède de nombreux outils statistiques en plus des algorithmes et méthodes de forage de données. Des méthodes de visualisation conviviales et interactives des résultats sont disponibles.

Algorithmes supportés : arbres de décision, réseaux neuronaux, raisonnement basé sur la mémoire, clustering, règles associatives.

Plateforme : Windows, AIX.

Sources des données : ODBC (Oracle, Access, etc.), fichiers importés excel et access, fichiers CSV, etc.

Évaluation : Le logiciel SAS est reconnu en statistiques et très utilisés dans l'industrie. Le module de forage de données offre de nombreux algorithmes, dont tous les algorithmes d'analyse statistique disponibles dans la version de base de SAS. Les formules trouvées par le logiciel peuvent être exportées en C et Java pour développement ultérieur. L'outil « Reporter » permet de générer des écrans de sortie en HTML. Le logiciel SAS supporte aussi le standard émergent PMML pour conserver les résultats des algorithmes de forage de données.

Conclusion : SAS s'avère être un outil idéal pour faire l'analyse d'entrepôts de données, et peut de surcroît générer et entretenir cet entrepôt. La plupart des algorithmes reconnus en forage de données sont implémentés dans les modules de forage de données de SAS, et de bonnes méthodes de visualisation sont disponibles. L'utilisation de SAS comme application de forage de données devrait être sérieusement considérée. De plus, la compagnie SAS est très active et fournit régulièrement des mises à jour de ses logiciels, en suivant l'évolution des différents standards (comme PMML).

SPSS

Produit : Clementine

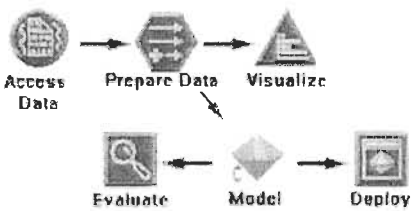
Compagnie : SPSS Inc.

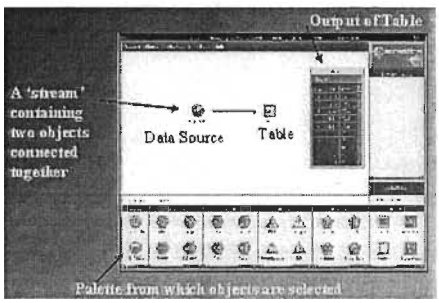
Site web : http://www.spss.com/spssbi/clementine/

Algorithmes supportés : Arbres de classification, réseaux neuronaux, clusters (K-Means), règles associatives, régression linéaire.

Plateforme : Client : Windows.
Serveur : Windows, Solaris, HP-UX, AIX.

Description : L'application Clementine supporte un processus de forage de données interactif, par découvertes et raffinements successifs. Cette application peut utiliser des procédures et des sources de données disponibles en SPSS. Il est possible d'exporter en C le code des modèles générés lors du processus de forage de données. Une interface est disponible avec plusieurs outils de visualisation des données. Le client et le serveur sont nécessaires pour que Clementine fonctionne.





Sources de données : Fichiers SPSS, ODBC, fichiers Access, Excel, texte (séparé par des virgules).

Évaluation : Clementine est une solution pour le forage de données qui s'intègre à un environnement où SPSS est déjà utilisé. Ce logiciel permet d'utiliser des sources de données et des procédures SPSS en affichant tous les résultats dans l'interface utilisateur de Clementine. Les algorithmes sont optimisés pour les grands jeux de données. De nombreuses sources de données peuvent être employées. Le processus de forage de données est complètement interactif et peut se faire à l'aide de quelques clics de souris. Un version d'essai n'est cependant pas disponible.

Conclusion : L'outil Clementine comporte de nombreux algorithmes et peut se connecter à différentes sources de données. Par contre, l'absence de connections pour la programmation à l'extérieur de l'application est une limitation importante. Il est tout de même possible de fabriquer des solutions personnalisées en utilisant la programmation SPSS. L'outil Clementine pourrait être testé, mais à cause des différentes limitations énumérées, il n'est pas recommandé pour le développement de l'application de forage de données.

DBMiner

Produit : DBMiner 2.0

Compagnie : DBMiner Technology inc. (Microsoft)

Site web : http://www.dbminer.com

Algorithmes supportés : Règles associative.

Plateforme : Windows

Sources de données : Microsoft SQL Server 2000, OLEDB (Oracle, IBM, ...)

Description : DBMiner SX 2002 est une application serveur qui offre des capacités de forage de données sur les séquences pour Microsoft SQL Server 2000. DBMiner SX 2002 supporte et étend les standards Microsoft OLE DB pour forage de données. Les algorithmes ont reçus une attention mondiale et sont plus performants que les compétiteurs.

Évaluation : Le logiciel DB Miner supporte uniquement les algorithmes de règles associatives et est optimisé pour la plateforme Microsoft SQL Server 2000. Bien que ce logiciel peut servir à analyser d'autres types de sources de données, il doit se connecter à un serveur SQL de Microsoft pour permettre d'utiliser toutes ses possibilités et la vitesse de traitement tant annoncée. Une version d'évaluation du produit est disponible.

Conclusion : DB Miner est clairement une application qui vise un marché très particulier de clients qui possèdent de très vastes entrepôts de données transactionnelles (par exemple, une chaîne de magasins). Mais dans le cas de l'application à développer, il n'est pas nécessaire d'avoir une application aussi optimisée que semble l'être DB Miner. De plus, un choix dans des algorithmes est préférable, de plus que dans le type de support pour les données (ne pas se limiter à SQL Server 2000 de Microsoft). Se produit ne devrait pas être retenu pour l'application à développer.

DB2 Intelligent Miner for Data

Produit : DB2 Intelligent Miner for Data

Compagnie : IBM

Site web : http://www-3.ibm.com/software/data/iminer/fordata/

Algorithmes: Règles associatives, arbres de classifications, clusters.

Plateforme : AIX, Windows NT, Sun Solaris, OS/390.

Description : DB2 Intelligent Miner for Data est une application qui comporte un ensemble d'outils pour le processus itératif de forage de données. Elle comporte des outils pour faire le traitement des données, des analyses statistiques et la visualisation des résultats. Plusieurs algorithmes de forage de données sont disponibles. L'application peut gérer de très grandes quantités de données grâce à son architecture parallèle. Plusieurs serveurs peuvent facilement être combinés. Une interface de programmation est disponible pour programmer des solutions personnalisées.

Source de données : DB2 2.1.1 (IBM).

Évaluation : L'application DB2 Intelligent Miner for Data semble restreinte à l'utilisation de la base de données DB2 d'IBM. Les algorithmes sont très optimisés pour les calculs très exigeants et le travail en parallèle. Les algorithmes disponibles peuvent être combinés pour trouver des informations. Le système semble être complètement intégré, c'est une solution complète pour les compagnies qui cherchent à faire du forage de données avec peu de développement.

Conclusion : DB2 Intelligent Miner for Data est un logiciel qui a été développé par IBM pour ses clients qui possèdent déjà des serveurs IBM, mais surtout la base de données DB2 d'IBM. L'application n'est pas disponible sur d'autres types de serveurs de données. De plus, une version d'essai n'est pas disponible (seulement un document de démonstration). À moins de disposer d'un serveur IBM, cette application ne peut pas être retenue pour l'application de forage de données à développer.

Conclusion

Ce travail de consultation est une étape préparatoire au développement d'une application de forage de données qui devra pouvoir fonctionner à partir d'un entrepôt de données conçu sur Oracle et qui sera utile à des chercheurs de différents domaines. L'objectif du développement de cette application est d'avoir un produit fonctionnel qui peut servir à des spécialistes d'un domaine (autre que l'informatique ou les mathématiques). Le produit doit donc fournir suffisamment d'automatismes pour rendre l'interface conviviale et permettre l'apprentissage des différentes fonctionnalités et algorithmes au rythme du chercheur. Pour développer une telle application, il faut trouver des outils qui peuvent accéder aux informations sur Oracle et qui permettent de développer des solutions personnalisées. Les algorithmes de forage de données utilisés doivent pouvoir à la fois traiter de grandes quantités de données et produire des résultats qui sont compréhensibles et réutilisables par les chercheurs. De plus, l'application qui sera développée autour de ces algorithmes doit tenir compte de leurs exigences, voir faire un certain nettoyage des données avant de les acheminer aux algorithmes. Finalement, l'application doit être conviviale et permettre aux chercheurs de facilement accéder aux données et de produire des résultats utiles à leurs recherches.

Certains produits commerciaux sont déjà disponibles et correspondent aux exigences de l'application à développer. Cependant, il reste encore à faire une sélection dans les produits suggérés en utilisant les versions d'évaluation pour faire un choix et déterminer lequel correspond le mieux à ce qui est recherché pour compléter l'application. Il est cependant évident que la plupart des algorithmes de forage de données sont déjà disponibles sous la forme de fonctions qui peuvent être appelées à partir d'autres produits. Le travail de développement ne devrait donc pas nécessiter l'implémentation des algorithmes les plus couramment utilisés dans le domaine du forage de données, comme les arbres de classification ou de décision, les réseaux neuronaux, les classificateurs bayesiens, les clusters avec K-Means et KNN, ou les algorithmes à règles associatives. Par contre, les algorithmes génétiques ne sont pas utilisés souvent dans les applications de forage de données (vue l'implémentation très personnalisée qu'il faut refaire à chaque fois). Alors, à l'exception des algorithmes génétiques, les autres algorithmes devraient être réutilisés à partir d'une application commerciale. Il serait aussi possible de tenter une combinaison de différents algorithmes.

Discussion

Bien que les algorithmes présentés permettent de faire le traitement à partir d'un entrepôt de données et que les produits commerciaux puissent aider à la visualisation et la compréhension des informations trouvées, il y a d'autres aspects au forage de données. Avant d'utiliser les algorithmes, il faut avoir préparé le jeu de données, il faut connaître le domaine d'application des données pour pouvoir interpréter les résultats. Il y a aussi l'aspect de l'accès aux données, du droit d'utiliser ces données. Toutes ces questions, et bien d'autres, font parties du domaines des entrepôts de données et du forage de données. Quelques-unes de ces questions sont présentées un peu plus en détail pour compléter la discussion.

Choix des algorithmes

Les différents algorithmes présentés ont chacun leurs forces et leurs faiblesses. Il serait intéressant de les comparer sur différents types de jeux de données de façon à permettre une sélection à l'utilisateur selon le type de données à analyser.

Protection de la vie privée

[ADV96] p.25

Il ne faut pas oublier que la constitution d'un entrepôt de données peut parfois être controversée, surtout lorsque vient de temps de faire des croisements de bases de données. Il suffit de se rappeler les débats sur la place publique qui sont survenus lorsque le gouvernement québécois a voulu exploiter ce qu'il appelait le « méga-fichier » sur la population québécoise. C'est pourquoi il faut s'assurer d'avoir l'accord des partis impliqués avant de procéder à la préparation d'un entrepôt de données.

Nettoyage des données

Il y a plusieurs aspects à considérer lors de la préparation des jeux de données. Par exemple, quoi faire avec les données manquantes? Quels types de données les algorithmes peuvent-ils utiliser? Faut-il générer des classes à partir des données continues? Il arrive aussi que des données complètement erronées réussissent à s'introduire dans les entrepôts. Comment faut-il réagir aux valeurs extrêmes? De surcroît, il est possible que certaines valeurs extrêmes identifiées apportent vraiment des informations importantes au sujet d'un phénomène qui n'avait pas encore été considéré. Le travail de nettoyage de données n'est pas trivial, et un expert du domaine est souvent nécessaire pour permettre de mener à bien cette étape du forage de données.

Méthodes automatiques ou semi-automatiques

Il existe des algorithmes qui doivent absolument être interactifs pour produire les résultats attendus, c'est-à-dire qu'un expert doit fournir des indications sur la façon de regrouper les éléments, choisir les paramètres ou fournir des jeux d'entraînement. Il existe aussi des algorithmes qui peuvent fonctionner indépendamment d'un utilisateur, en parcourant les données sans avoir besoin d'interagir avec un utilisateur (comme les clusters). Ces deux types d'algorithmes peuvent être utiles dans l'application de forage de données, les algorithmes automatiques servant souvent à familiariser l'utilisateur avec les données avant de commencer le véritable processus de forage de données.

Efficience sur de grands jeux de données

Il ne faut pas oublier que le forage de données a pour but de trouver des informations à partir des données qui sont récoltées à d'autres fins, et qu'il y a souvent beaucoup de données récoltées. Ces algorithmes doivent être capables de traiter des quantités très importantes de données dans un laps de temps acceptable. Certains algorithmes ne sont pas encore au point (par exemple, la programmation logique inductive). Mais avec la recherche, il sera sûrement possible de trouver de nouveaux algorithmes encore plus performant et plus expressifs que ceux qui ont été recommandés dans ce travail.

Plusieurs types d'algorithmes ont été présentés. Certains sont moins souvent utilisés, comme l'algorithme de programmation logique. Ces algorithmes sont parfois trop gourmands en ressources, ou d'autres fois simplement trop difficiles à généraliser (comme les algorithmes génétiques).

Bibliographie

[ADV96] Usama M. Fayyad. Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAi Press/The MIT Press. 1996. 609 p. ISBN 0-262-56097-6

[CDF00] Edith Cohen, Mayur Datar, Shinji Fujiwara, Aristides Gionis, Piotr Indyk, Rajeev Motwani, Jeffrey D. Ullman, Cheng Yang. *Finding Interesting Associations without Support Pruning*. 2000.

[CLAS94] *Building Classification Models: ID3 and C4.5*. http://yoda.cis.temple.edu:8080/UGAIWWW/lectures/C45/

[JOS97] Karuna Pande Joshi. *Analysis of Data Mining Algorithms*. 1997. http://userpages.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm

[KJ00] *Tutorial on High Performance Data Mining*. http://www-users.cs.umn.edu/~mjoshi/hpdmtut/

[KNST] Angoss, *KnowledgeSTUDIO*. http://www.angoss.com/ProdServ/AnalyticalTools/kstudio/whitepaper.htm

[MLDM98] Ryszard S. Michalski. Ivan Bratko, Miroslav Kubat. *Machine Learning and Data Mining*. John Wiley & Sons Ltd. 1998. 456 p. ISBN 0-471-97199-5

[PUN93] W.F. Punch, E.D. Goodman, Min Pei, Lai Chia-Shun, P. Howland, R. Enbody. Further Research on Feature Selection and Classification Using Genetic Algorithms. Apparu dans ICGA93, p. 557-564, Champaign Ill. 1993. 8 p.

[QUEST] *Quest : The Data Mining Group* (IBM). http://www.almaden.ibm.com/cs/quest/

[STATSOFT] *Statsoft products*. http://www.statsoft.com/

ANNEXE C

Le chargement des données dans l'entrepôt

1. Introduction

L'entrepôt supporte de nombreuses applications et les données qu'il contient doivent être mise à jour régulièrement. Un processus de chargement des données extensible existe afin de faciliter la maintenance et l'ajout de sources de données. La base de réflexion qui a mené à ce processus est a été déterminé dans un document durant l'hiver 2003 (Dugré et Delisle, 2003). Le principe de base du chargement est expliqué à la section 2. Les étapes du chargement sont énumérées à la section 3. Finalement, des recommandations sont faites pour l'ajout de nouvelles sources de données à la section 4.

2. Principe de base

Le chargement doit rester flexible et facile d'entretien. Des tables ordinaires sont utilisées lors du chargement, et les algorithmes des différentes étapes de chargement se chargent de transférer les données entre les tables lorsque c'est nécessaire.

2.1 Pourquoi ne pas utiliser de vues matérialisées

Les vues matérialisées d'Oracle 8i n'ont pas été utilisées. Ces vues ne permettent pas d'obtenir toutes les optimisations souhaitables (par exemple, les vues matérialisées supportent un *fast refresh* uniquement si aucune jointure entre tables n'est faite), et il aurait été plus difficile de supporter l'ajout automatique de champs. De plus, en utilisant un transfert manuel des données, il est possible d'utiliser le *rollback* durant les transactions, ce qui facilite grandement la récupération lorsque des problèmes surviennent lors de chargement. Puisque la taille individuelle de chaque table n'est pas trop grande pour le *rollback segment* du serveur utilisé, c'est un avantage indéniable lors de la préparation du chargement de pouvoir s'assurer que tous les changements erronés seront inversés. De cette façon, il est facile de s'assurer que seule la plus récente version correcte des données est accessible. Aussi, puisque des tables sont utilisées plutôt que des vues matérialisées, il est facile d'interrompre le processus de chargement pour voir où les erreurs sont survenues.

2.2 Fonctionnement du gestionnaire

Chaque source de données utilise une classe « pilote » qui charge toutes les étapes de chargement en mémoire. Ces étapes héritent de l'interface « EtapeChargement » et peuvent alors être exécutées par la classe principale « GestionnaireChargement ». Ce gestionnaire appelle les classes pilote une à une, et ces classes retournent une liste des étapes à exécuter, en fonction de l'étape actuellement exécutée. Par exemple, si le gestionnaire indique qu'il est rendu à l'étape 3, les pilotes lui retournent une liste d'étapes à exécuter qui correspondent à l'étape 3 pour chaque source de données. De cette façon, si un problème survient, toutes les sources de données sont rendues au même point et il est plus facile de trouver les erreurs. Pendant ce temps, l'étape 8 est

appelée en mode récupération pour faire le nettoyage de l'entrepôt (principalement à l'aide d'un rollback) pour s'assurer que les utilisateurs auront accès à une version correcte des données.

C'est le gestionnaire qui se charge d'exécuter les étapes dans l'ordre, et c'est aussi lui qui décide s'il faut invoquer l'étape de nettoyage (l'étape 8) en mode de récupération après erreur ou si le déroulement du chargement s'est bien déroulé. En cas d'erreur, le gestionnaire transmet un courriel aux administrateurs de l'entrepôt pour leur indiquer qu'un problème s'est produit.

Note : Pour l'instant, tous ces comportements, incluant le chargement des pilotes et la liste des courriels des administrateurs sont codés directement dans le gestionnaire. Pour les modifier, il faut disposer du code source et d'un compilateur Java. Cette situation **devrait** être modifiée et le mode d'entretien du gestionnaire **décrit dans ce document**.

2.3 Historique des données

Les données qui doivent conserver des données historiques utilisent le champ « ROWDATE » pour indiquer la date effective du chargement de l'enregistrement dans l'entrepôt et « FINVALIDE » lorsque l'étape 2 indique que l'enregistrement source à été modifié, toujours à partir de la date actuelle du chargement. Une ancienne version de la base de données source est toujours conservée dans le schéma de chargement et identifié par l'étape 2, c'est cette copie qui permet de détecter les changements. Lorsque l'étape 8 est franchie en mode normal, l'ancienne copie de la base de données est supprimée et c'est la nouvelle qui prend sa place.

3. Liste des étapes

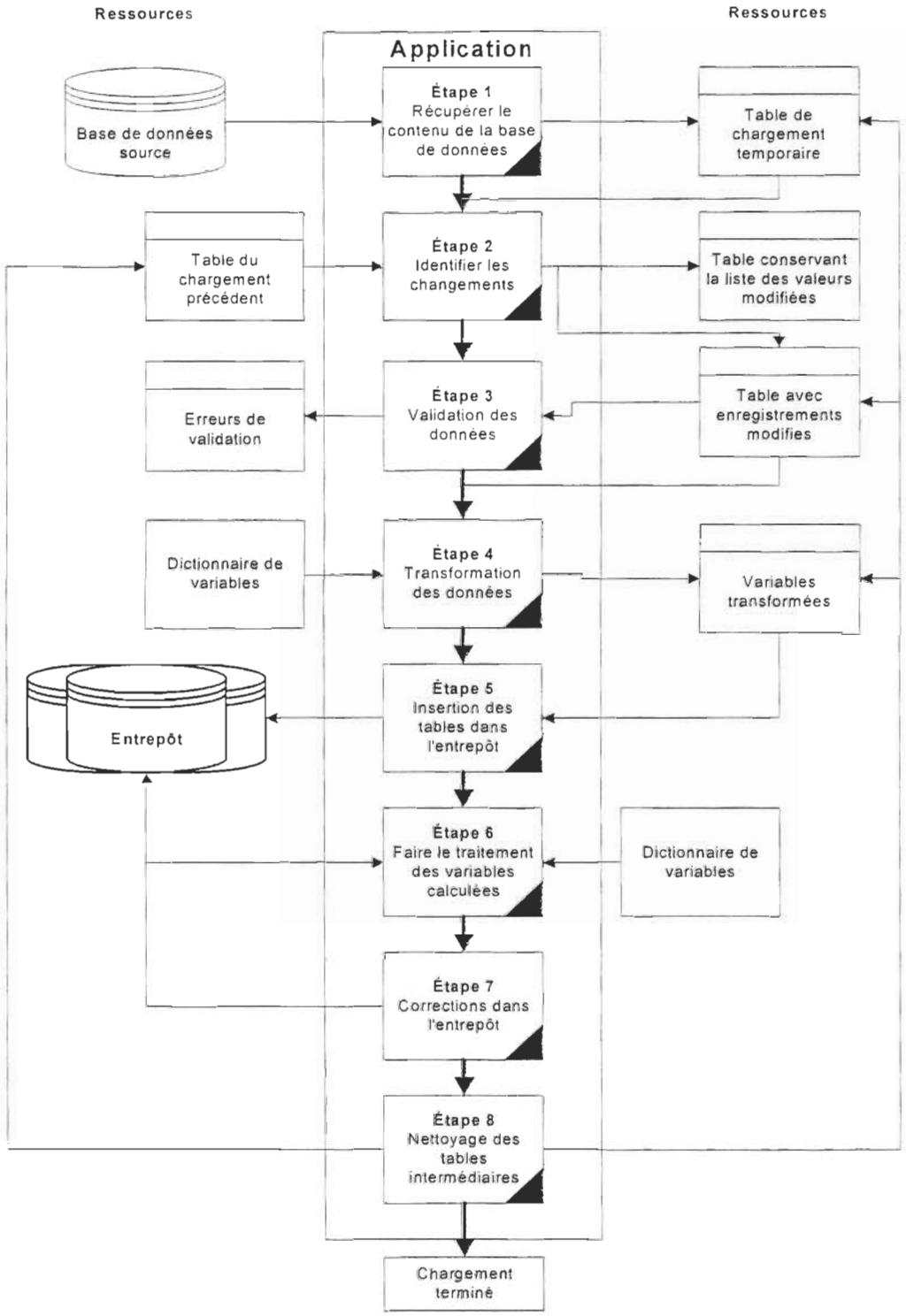


Figure 1 Étapes de chargement (Dugré et Delisle, 2003)

3.1 Étape 1 : Importer les données

L'étape 1 consiste à copier le contenu des différentes sources de données dans une table intermédiaire de l'entrepôt, pour faire des traitements (et transformations) localement. On ne modifie pas les données dans la base de données.

En cas de besoin, si des champs nouveaux sont identifiés dans la table source, ils sont créés dans la table destination (la table temporaire de chargement). Cette façon de faire permet un entretien simplifié de l'entrepôt, puisque certains traitements de base sont faits automatiquement. Il faudra tout de même documenter cette nouvelle variable dans le dictionnaire à la main.

3.2 Étape 2 : Identifier les changements

Une table permet de comparer les données déjà chargées dans l'entrepôt aux nouvelles données qui ont été importées. Les différences (insertion, modification, suppressions) sont retenues et leur traitement se poursuivra dans les prochaines étapes, éliminant ainsi beaucoup de traitement redondant. C'est aussi de cette façon qu'on peut faire un suivi dans le temps de la base de données, puisqu'à chaque chargement de l'entrepôt, on génère une liste des changements et on conserve les nouvelles données (tables *LOG*).

C'est une série de requêtes SQL générées à l'aide de la liste des champs qui est utilisée pour comparer les données. Une requête détecte les ajouts, une détecte les suppressions, et une troisième détecte les modifications. Voici à quoi ces requêtes ressemblent :

Détection d'ajouts :

```
SELECT clés
FROM nouvelle
MINUS
SELECT clés
FROM ancienne
```

Suppression :

```
SELECT clés
FROM ancienne
MINUS
SELECT clés
FROM nouvelle
```

Modification :

```
SELECT clés
FROM nouvelle, ancienne
WHERE nouvelle.cle = ancienne.cle
```

```
AND (nouvelle.champ1 != ancienne.champ1 OR
nouvelle.champ2 != ancienne.champ2 ...)
```

De cette façon, c'est le serveur Oracle qui effectue le plus gros de la tâche, et il s'en acquitte très rapidement. On évite d'avoir à faire des calculs dans des algorithmes créés à cet effet, et d'avoir à les optimiser.

3.3 Étape 3 : Valider et nettoyer les données

On reprend la validation des données qui est faite dans les questionnaires pour s'assurer qu'il n'y a pas eu de corruption (peu importe la raison). On peut aussi être plus exigeant. Par exemple, on peut décider d'importer des données qui sont « bizarres », mais en émettant un avertissement et en désactivant l'entreprise dans l'entrepôt.

Note : Cette étape n'est pas encore utilisée dans l'entrepôt actif, mais avec l'ajout de nouvelles sources de données qui ne sont pas méticuleusement vérifiées à la main (comme Balise), elle deviendra nécessaire.

3.4 Étape 4 : Transformer les données

Cette étape sert à faire toutes les transformations qui sont nécessaires aux traitements statistiques des données. C'est aussi dans cette étape qu'on va calculer le taux de change, ce qui va dupliquer les données par autant de devises qui sont en utilisation dans l'entrepôt. Il est préférable de calculer le taux de change immédiatement plutôt que sur demande, ce calcul est relativement exigeant et appliqué à un très grand nombre de données.

Les données sources pour le taux de change constituent une source de données pour l'entrepôt. Pour l'instant, les données sont saisies à la main dans la base de données manufacturières et importées dans l'entrepôt. Cependant, si des changements sont détectés dans le taux de change (une correction sur un taux antérieur), il faut s'assurer de recalculer les champs affectés comme s'ils venaient d'être importés à nouveau. Ce traitement est fait dans le chargement du taux de change, et il est considéré dans l'étape 2 de toutes les sources de données.

Le calcul fonctionne à partir de la classe « CalculTauxChange » qui permet de déterminer les différentes conversions à effectuer. Cette classe utilise une fonction PL/SQL pour calculer les valeurs (dans PKG_TAUX_CHANGE du schéma de chargement). Elle génère une requête SQL qui utilise cette fonction et conserve tous les enregistrements dans une nouvelle table, en ajoutant le champ DEVISE qui permet de savoir dans quelle devise l'enregistrement est calculé.

3.5 Étape 5 : Transférer dans l'entrepôt

On récupère les nouvelles données avec toutes les transformations, et on les insère dans les bonnes tables de l'entrepôt. Les champs sont insérés dans les tables, et les anciens sont conservés. Il existe cependant une option qui permet de fonctionner par écrasement d'anciennes valeurs, même si cette option n'est pas activée. Il est **très important** de noter que c'est seulement à partir de l'étape 5 que les données de l'entrepôt sont modifiées par le chargement. En effet, les étapes 1 à 4 fonctionnent dans un schéma séparé qui n'a aucune incidence sur le contenu de l'entrepôt. C'est pourquoi une transaction est démarrée dès le début de l'étape 5, et cette transaction prend fin uniquement lorsque le nettoyage de l'étape 8 est terminé. Ainsi, si un problème survient durant les étapes 5 à 8, un *rollback* est utilisé pour éliminer les changements à l'entrepôt, et les tables utilisées dans les étapes 1 à 4 sont tout simplement effacées (sauf l'historique de l'étape 2, bien sûr).

3.6 Étape 6 : Préparer les variables calculées

On peut enfin faire les calculs pour les variables qui sont issues d'autres variables. Par exemple, on peut créer une variable VARCAL1 qui est issue du calcul (VARIABLE1 + VARIABLE2 / 20). On ne prévoit pas, pour le moment, permettre des calculs statistiques poussés dans l'entrepôt (analyse de la variance, régressions linéaires, etc.). C'est à cette étape qu'on effectue un calcul particulier, le suivi des variables dans les questionnaires.

Note : Les variables calculées ne sont pas encore implémentées dans l'entrepôt actif. Cependant, elles devront être supportées et calculées dans cette étape à l'avenir.

3.7 Étape 7 : Mettre à jour les index

Il peut arriver que certaines situations requièrent des corrections dans les données qui sont déjà dans l'entrepôt. Ces changements doivent être clairement identifiés, mais ils entraînent néanmoins une correction des données qu'il faut prévoir. Mais en général, l'étape 7 sert à calculer les champs spéciaux qui servent d'index pour un accès rapide aux données dans l'entrepôt. Par exemple, il y a le champ PLUSRECENT qui permet d'aller chercher les enregistrements les plus récents pour toutes les tables de l'entrepôt, sans avoir à faire de calcul avec les dates (ROWDATE et FINVALIDE).

Note : techniquement, le calcul de ce champ fonctionne avec FINVALIDE puisque la date 3000-01-01 représente un enregistrement qui est valide au moment présent. Cependant, puisque PLUSRECENT utilise un index binaire, utiliser le critère PLUSRECENT = 1 retourne les enregistrements plus rapidement que d'utiliser le critère FINVALIDE = 3000-01-01.

3.8 Étape 8 : Préparation pour le prochain chargement

Les différentes tables qui ont servi au chargement de l'entrepôt peuvent maintenant être vidées et préparées pour le prochain chargement. Le chargement est une fonctionnalité qui doit permettre un travail automatique ou assisté, selon les besoins. Il va arriver que des données soient incorrectes. Si jamais le chargement doit être interrompu, il doit être possible de corriger les problèmes et de redémarrer à l'étape où l'erreur est survenue.

Il existe un mode spécial appelé « récupération » qui peut être invoqué par le gestionnaire de chargement. Normalement, le nettoyage supprime le contenu des tables utilisées lors du chargement (étapes 1 à 4) avec l'ancienne copie de la base de données, puis copie la base de données utilisée à l'étape 1 pour en faire la nouvelle version de comparaison et un commit est fait. Lorsque le mode récupération est utilisé, l'ancienne copie de la base de données n'est pas effacée, les tables des étapes 1 à 4 sont vidées et un rollback de la transaction démarrée à l'étape 5 est fait. Le mode de récupération est donc la méthode utilisée par le gestionnaire pour s'assurer qu'en toute circonstance, une version correcte des données est accessible à l'utilisateur. Ces données ne sont peut-être pas toujours les plus récentes, mais au moins elles sont sûres, et c'est le but du mode récupération.

3.9 Autres étapes

Il existe des traitements qui sont spécifiques à certaines sources de données et qui ne sont pas spécifiquement des étapes numérotées. Par exemple, il y a le suivi pour le questionnaire. Pour être exécutés, ces traitements doivent cependant être inclus au pilote de chaque source de données et retourné au gestionnaire durant une étape spécifique. Par exemple, l'étape du suivi est ajoutée à l'étape 5 du chargement pour le questionnaire manufacturier. Il est donc possible d'étendre le comportement du chargement très facilement, en intercalant les traitements nécessaires entre les étapes, et ce à partir du pilote de chargement de cette source de données.

4. Ajout de sources de données

L'entrepôt est maintenant prêt à fonctionner, mais il est loin d'être complet! De nombreuses informations pourraient y être ajoutées, et c'est d'ailleurs dans cette optique que le processus de chargement a été conçu. C'est pourquoi une méthode est proposée afin de faciliter l'ajout de nouvelles sources de données à l'entrepôt.

4.1 Identifier les étapes nécessaires

Il peut arriver que toutes les étapes disponibles ne soient pas nécessaires lors du chargement d'une source de données particulière. Par exemple, si une source de taux de change mise à jour par le Web est ajoutée, il n'est pas nécessaire de faire les étapes 3, 4, 6 ni 7 puisque ces étapes représentent des phases de calcul et qu'on peut assumer que la source choisie sera fiable (sinon, qu'est-ce qu'on pourrait y faire de toute façon?). D'un autre côté, si on désire ajouter une source comme la base de

données de Balise, la plupart des étapes seront nécessaires. Le choix des étapes est important puisqu'il faut ensuite créer un espace temporaire pour faire les traitements, ainsi que de nouvelles tables dans l'entrepôt.

4.2 Identifier les tables à importer

Une fois les étapes choisies, il faut soigneusement sélectionner les tables qui seront utiles dans l'entrepôt. Par exemple, une table qui contient la liste des projets d'un utilisateur peut s'avérer utile, mais la table contenant les questions (et non les réponses) ne serait d'aucune utilité à l'entrepôt, ne serait-ce que parce que le contenu est relativement statique. Ainsi, les tables sont choisies et il peut parfois s'avérer utile de faire des jointures ou autres manipulations jugées utiles à l'avance, pour faciliter le travail de chargement.

4.3 Créer les tables dans le schéma de chargement

Pour chaque table identifiée à la source, il faudra préparer une ou des tables dans l'espace de chargement de l'entrepôt (selon le nombre d'étapes retenues). Il faudra nécessairement une table pour importer les données, et une autre pour conserver une ancienne version de la table pour fins de détection des changements. Si des calculs de validation ou de transformations sont nécessaires, il faudra encore d'autres tables. Toutes ces tables doivent contenir les mêmes champs avec les mêmes types de données que les tables sources.

4.4 Créer les tables dans l'entrepôt

Une fois le travail de préparation terminé, il faut prévoir une ou des tables à l'intérieur de l'entrepôt pour conserver les données. Cette table devrait d'ailleurs avoir un champ ROWDATE et un champ FINVALIDE pour permettre de conserver l'historique des enregistrements, un champ PLUSRECENT pour trouver les enregistrements les plus récents à l'aide d'un index binaire. La clé devrait aussi permettre de retracer l'enregistrement correspondant dans la source de données. Les autres champs peuvent être transformés ou calculés durant le processus de chargement.

4.5 Créer le gestionnaire et les pilotes de chargement

Une fois que toutes les tables sont prêtes, il faut créer un pilote qui permettra de faire le chargement initial ainsi que tenir à jour les données dans l'entrepôt. Ce pilote doit hériter de la classe « GestionnaireChargement » afin de pouvoir s'exécuter dans l'application de chargement (c'est la classe chargementEntrepot). De plus, les étapes configurées dans le pilote peuvent être les mêmes que celles qui existent déjà, avec des paramètres de configuration différents (tous décrits dans la javadoc), ou encore de toutes nouvelles étapes créées spécifiquement pour cette source de données. Dans ce dernier cas, il faudra absolument que l'étape hérite de l'interface « EtapeChargement ».

Les étapes de chargement sont une suite de requêtes SQL et de traitements qui permettent de traiter les données pour les importer dans l'entrepôt. Il n'y a pas de convention particulière à respecter lors de leur création, à l'exception du fait que chaque étape individuelle ne devrait exécuter que ce qui est décrit dans ce document à la section 3.

5. L'application de chargement

La classe chargementEntrepot (voir figure 2) se charge de faire le chargement des données de l'entrepôt à partir des différents gestionnaires créés. Il faut inclure chaque gestionnaire dans cette classe pour permettre un chargement correct. Plusieurs vérifications sont faites pour détecter les erreurs, et un message est envoyé aux administrateurs en cas de problème. Cette classe pourrait cependant bénéficier d'un remodelage, notamment en rendant le processus de configuration plus flexible.

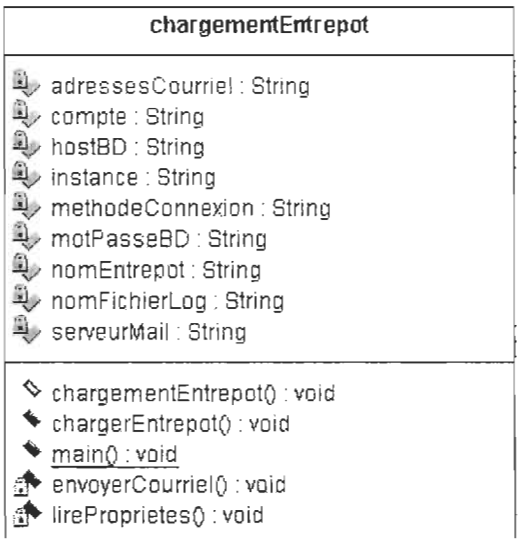


Figure 2 Schéma de la classe chargementEntrepot

Chaque gestionnaire (voir figure 3) est appelé tour à tour, et certains de leurs comportements peuvent être configuré pour leur exécution. Les gestionnaires contiennent leurs propres pilotes de chargement des données et ils doivent être programmés pour pouvoir retourner une exception si jamais une erreur de chargement survient. C'est la méthode *charger* qui démarre le chargement.

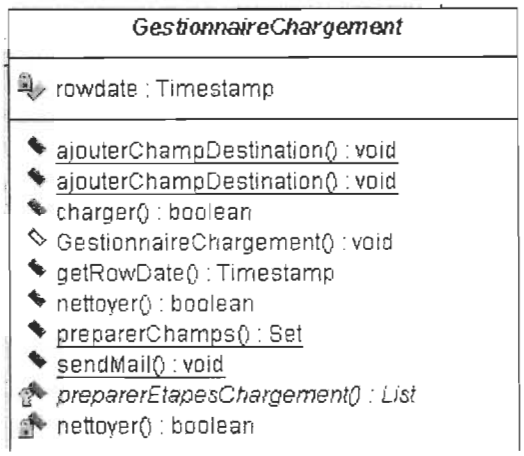


Figure 3 Schéma de la classe GestionnaireChargement

Le gestionnaire de chargement possède la méthode *preparerEtapesChargement* qui lui permet de générer une liste des étapes à parcourir pour charger les données. Cette liste d'étapes dépend de la configuration du chargement pour chaque base de données source. Chaque étape implémente l'interface *EtapeChargement* (voir figure 4), ce qui permet au gestionnaire de les exécuter plus simplement. Un exemple est la base de données manufacturières, qui possède les étapes suivantes :

- Transfert des données (étape 1)
- Vérification des changements (étape 2)
- Calcul du taux de change et du suivi (étape 4)
- Transfert des données vers l'entrepôt (étape 5)
- Corrections des index et champs temporels (étape 7)
- Nettoyage des tables de chargement (étape 8)

L'étape 3 n'est pas encore utilisée, c'est une étape réservée pour les validations des données à l'entrée de l'entrepôt. Les validations qui seront nécessaires restent encore à déterminer (travail en cours). L'étape 6 sera bientôt utilisée pour générer des variables *calculées*, qui proviennent de ratios, de moyennes, etc.

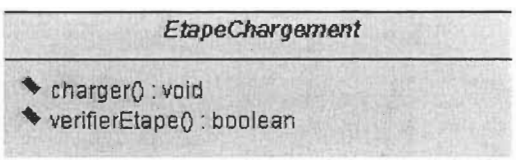


Figure 4 Schéma de l'interface EtapeChargement

La méthode *charger* sert à exécuter les algorithmes de préparation des données pour l'étape de chargement. La méthode *verifierEtape* permet de vérifier si les conditions initiales de l'exécution de cette étape sont respectées.

ANNEXE D

Schémas et tables de l'entrepôt de données

SCHÉMAS ET TABLES DE L'ENTREPÔT

1. Introduction

L'entrepôt est conçu à partir de plusieurs tables dans le serveur de bases de données Oracle 8i. La version 9i ou 10g n'était pas disponible sur les serveurs utilisés lors de l'activation de l'entrepôt. Afin de séparer les éléments sensibles, plusieurs schémas sont utilisés pour conserver les tables, ce qui permet de séparer sémantiquement les différentes parties de l'entrepôt. De cette façon, même si quelqu'un réussissait à accéder à des schémas (voir compte) utilisés sur Oracle, les dommages qu'il pourrait causer seraient réduits. La raison d'être de chaque schéma de support est décrite dans la section 2, puis les diagrammes entité-relation sont présentés à la section 3. L'entrepôt contient un magasin historique qui conserve les données sous la forme de questionnaires (section 4) et un magasin multidimensionnel avec 2 points de vues différents sur les données (section 5). Afin d'assurer le chargement et l'entretien des données, certaines parties du code ont été faites en PL/SQL (section 6) et des vues ont été utilisées (section 7). La section 8 présente les aspects de sécurité intégrés directement à la base de données. Finalement, la section 9 fait l'inventaire du code SQL et PL/SQL utilisé pour créer tous les schémas de support et les magasins.

2. Description des schémas Oracle

Plusieurs schémas Oracle sont utilisés. Ce sont des comptes utilisateurs qui avec des privilèges leur permettant de créer des tables, des vues, du code PL/SQL, etc. Ces schémas permettent de diviser la structure de l'entrepôt en unités logiques qui effectuent des tâches distinctes, comme le chargement ou le stockage des métadonnées. Voici une liste des schémas utilisés avec une courte description.

- ENTLAR_ETL : C'est le schéma qui est utilisé par le logiciel de chargement.
- ENTLAR_WEB : Ce schéma contient des tables temporaires ainsi que les comptes des utilisateurs qui peuvent accéder à l'entrepôt par le Web.
- ENTLAR_METADATA : C'est ici que toutes les informations et les tables se trouvent pour supporter le dictionnaire de variables et toutes les autres métadonnées de l'entrepôt.
- ENTLAR_DATAMARTS : Contient les magasins de l'entrepôt, soit les tables pour le magasin historique et pour le magasin dimensionnel.

3. Modèle entité-relation des tables de support

Les tables de support sont celles qui ne contiennent pas directement les données de l'entrepôt. Elles peuvent être temporaires, comme pour le chargement ou le support de certains calculs complexes, ou encore permanentes, comme celles qui contiennent la liste des utilisateurs qui peuvent accéder à l'entrepôt par le Web. Les tables contenant les métadonnées sont aussi considérées comme des tables de support, plusieurs de ces tables servent en effet à la navigation pour utiliser les données de l'entrepôt.

3.1 Schéma de chargement ENTLAR_ETL

Le schéma de chargement sert uniquement à l'application de chargement des données. Cette application est normalement la seule qui dispose du mot de passe permettant l'accès externe à ce compte. Ce schéma possède des droits en écriture dans les schémas ENTLAR_METADATA et ENTLAR_DATAMARTS, il ne faut donc pas donner un accès à une application tierce sans bonne raison. La plupart des tables de ce schéma permettent de parcourir les étapes de chargement dans un ordre linéaire, ce qui facilite d'ailleurs l'identification de problèmes lorsqu'ils surviennent.

Le schéma de chargement doit aussi disposer d'un accès en lecture dans toutes les bases de données qui sont utilisées comme source de l'entrepôt. Pour l'instant, les accès sont vers le schéma MANUFACTURIER qui contient la base de données manufacturière.

À cause de la méthode utilisée pour le chargement, il devrait normalement y avoir une série de tables pour chaque pilote créé dans l'application de chargement des données. Normalement, le nom de chaque table devrait commencer par son identificateur de série, par exemple QM pour Questionnaire Manufacturier. La deuxième partie du nom représente l'étape de la table, par exemple LOAD01, LOAD02, etc. La dernière partie du nom peut être omise, mais sert généralement à identifier le pilote qui se charge de cette table si le gestionnaire de chargement dispose de plusieurs pilotes pour le même projet de chargement. Par exemple, on peut avoir les tables suivantes : QM_LOAD01_GENERAL, QM_LOAD04_FINANCIER_TC (où FINANCIER_TC représente un identificateur de pilote différent). De cette façon, on peut créer un gestionnaire de chargement des données pour le questionnaire manufacturier, et ce gestionnaire dispose de 2 pilotes différents, un pour la partie générale (le questionnaire lui-même), et un pour la partie financière.

Voici une liste des tables du schéma de chargement :

- ENTLAR_TAUX_CHANGE_MODIFIE : Utilisée pour conserver les changements apportés au taux de change dans la base de données manufacturière. Permet de recalculer les questionnaires au besoin.
- QM_LOAD01_FINANCIER_01 : Copie de la table MAN_FINANCIER du schéma MANUFACTURIER.
- QM_LOAD01_GENERAL_01 : Copie des tables générales de la base de données manufacturière.
- QM_LOAD02_FINANCIER : Conserve les enregistrements qui ont été modifiés depuis le dernier chargement, tels qu'identifiés à l'étape 2 du chargement.
- QM_LOAD02_FINANCIER_COMP : Conserve une ancienne copie de la table QM_LOAD01_FINANCIER qui permet d'identifier les changements dans les enregistrements.

- QM_LOAD02_GENERAL : Conserve les enregistrements qui ont été modifiés depuis le dernier chargement, tels qu'identifiés à l'étape 2 du chargement.
- QM_LOAD02_GENERAL_COMP : Conserve une ancienne copie de la table QM_LOAD01_FINANCIER qui permet d'identifier les changements dans les enregistrements.
- QM_LOAD04_FINANCIER_TC et QM_LOAD04_GENERAL_TC : Ces tables ajoutent le champ DEVISE aux tables de chargement des étapes précédentes, ce qui permet d'appliquer le taux de change à tous les enregistrements. Cette table contient donc *n* fois le nombre d'enregistrements de la table précédente (LOAD02), où *n* est le nombre de devises supportées.
- QM_LOADLOG_FINANCIER : Conserve la liste des enregistrements modifiés à partir des clés COD_ESE et PERIODE.
- QM_LOADLOG_FINANCIER_CHANGE : Conserve la liste des modifications apportées aux champs, à partir des clés COD_ESE, PERIODE et NOMCHAMP. Conserve aussi la valeur.
- QM_LOADLOG_GENERAL : Conserve la liste des enregistrements modifiés à partir des clés COD_ESE et ANNEE.
- QM_LOADLOG_GENERAL_CHANGE : Conserve la liste des modifications apportées aux champs, à partir des clés COD_ESE, ANNEE et NOMCHAMP. Conserve aussi la valeur. Note : les champs sont limités à 100 caractères, si jamais une variable de description avec plus de 100 caractères est modifiée, elle ne peut être conservée.

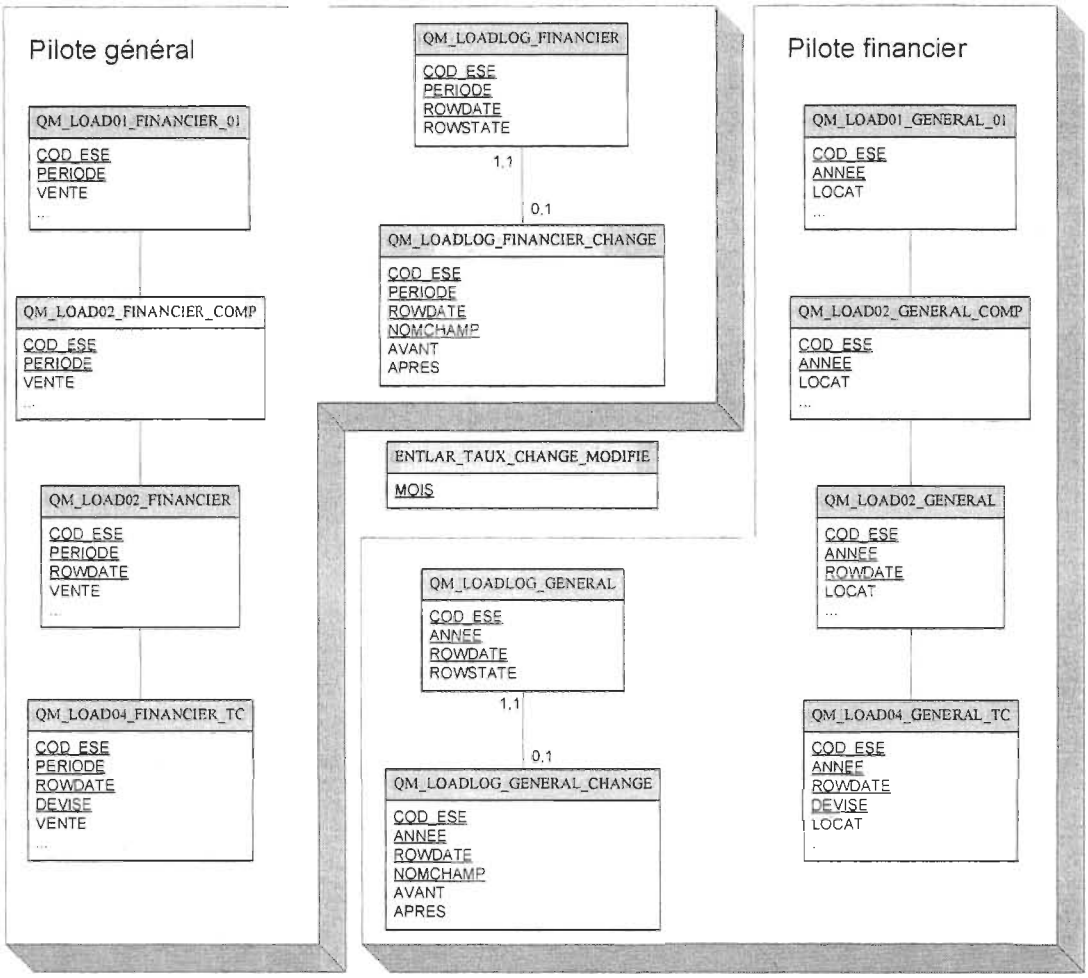


Figure 2 Diagramme entité-relation du schéma de chargement

Il faut noter que les tables utilisées pour le chargement, telles que présentées dans le diagramme entité-relation, ne montrent pas vraiment de relations entre-elles. C'est que de nombreux traitements sont faits entre les tables, et il arrive souvent qu'à un certain moment, une relation qui pourrait être faite ne soit pas respectée. C'est pourquoi aucune relation n'a été représentée pour les tables de chargement. Les tables de *log* ont une autre raison d'être, elles permettent de suivre la trace des chargements d'une manière historique. Ces tables sont plutôt un support pour l'application de chargement, et non pour les données.

3.2 Schéma Web ENTLAR_WEB

Le schéma Web sert surtout de compte d'accès à la base de données pour l'application Web sur Tomcat. Puisque le serveur Web doit avoir accès à un schéma sur Oracle pour opérer, il est préférable de lui en créer un spécifiquement à son usage, uniquement avec les accès dont il a besoin pour fonctionner. De cette façon, on évite des catastrophes potentielles. Le schéma Web a des accès en lecture seule

avec les magasins. Il a cependant des accès en lecture et écriture dans le schéma des métadonnées puisqu'il faut permettre aux utilisateurs d'accéder et de modifier le dictionnaire de variables, ainsi que les autres métadonnées. Il a aussi un accès en écriture dans la table des jeux de données (DATASETS) pour conserver les jeux de données créés par les utilisateurs. Finalement, le schéma Web n'a aucun accès au schéma de chargement.

Ce schéma possède seulement quelques tables temporaires et deux tables qui permettent à Tomcat d'authentifier les utilisateurs qui désirent accéder aux applications, ainsi que les rôles (droits) de chacun. Ces tables sont :

- DICT_TEMP_RECHERCHE : C'est une table temporaire (qui se vide automatiquement après chaque transaction) qui sert à afficher la liste des variables dans le moteur de recherche des variables.
- QM_DATASET_SELECT_FINANCIER : Cette table temporaire fait la liste des entreprises et des années financières à utiliser dans un jeu de données demandé par l'utilisateur.
- QM_DATASET_SELECT_GENERAL : Cette table temporaire fait la liste des entreprises à utiliser dans un jeu de données demandé par l'utilisateur.
- UTIL_ROLES_UTILISATEURS : Contient les droits accordés à chaque utilisateur sous forme de rôles définis dans l'application Web (fichier web.xml).
- UTIL_UTILISATEURS : C'est la liste des utilisateurs qui peuvent accéder à l'application Web. Les administrateurs de l'application disposent d'une interface Web pour accéder et modifier cette table.

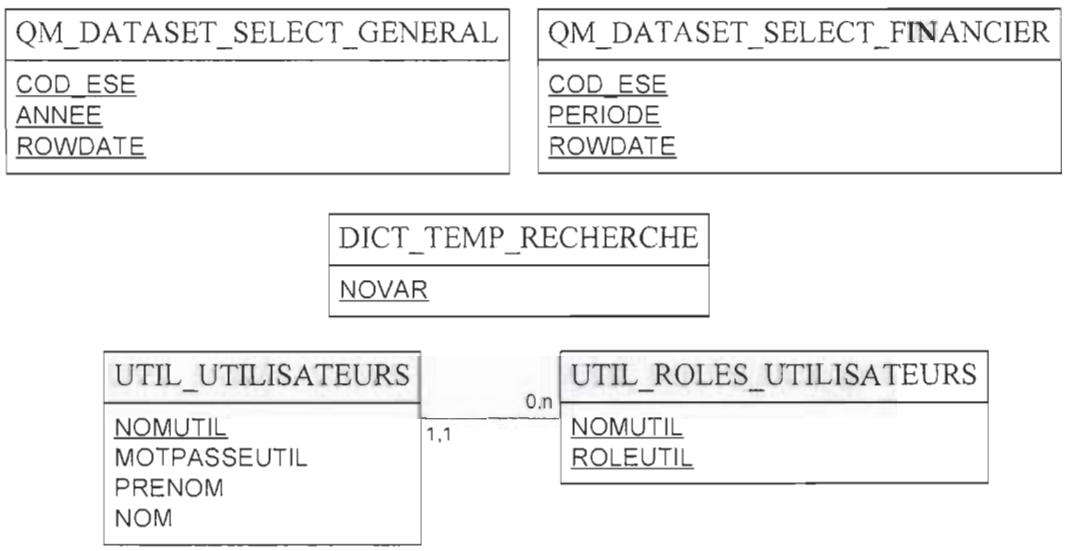


Figure 3 Diagramme entité-relation du schéma Web

Encore une fois, on remarque qu'il n'y a pas tellement de relations entre les tables. Elles sont majoritairement indépendantes, il n'y a que les tables utilisées pour

l'authentification avec Tomcat qui sont vraiment des tables d'application qui ont toujours des données. Les autres tables sont temporaires et normalement vides.

3.3 Schéma des métadonnées

Le schéma des métadonnées supporte toutes les tables utilisées par le dictionnaire de variable. C'est aussi dans ce schéma que les autres tables de métadonnées sont conservées. Normalement, il ne devrait pas être nécessaire d'utiliser le compte associé au schéma ENTLAR_METADATA (avec son mot de passe) pour fonctionner, à l'exception de l'entretien. Ce compte n'a pas d'accès vers d'autres schémas, mais tous les autres schémas l'utilisent.

Il y a des tables spécifiquement pour le dictionnaire, elles sont identifiées par le préfixe DICT_. Pour l'instant, une seule autre table est présente, elle fait la liste des devises supportées par l'entrepôt. Voici la liste des tables :

- DICT_ALIAS : Cette table permet de conserver une liste d'alias associés à chaque variable. Il pourrait être nécessaire, dans le futur, de renommer des variables, et afin de supporter les anciennes applications, il serait alors possible d'utiliser un alias. Cette table est utilisée dans la recherche des noms de variables.
- DICT_COMMENTAIRES : Conserve les commentaires faits par des utilisateurs qui modifient des variables.
- DICT_ETIQUETTES_LISTES : C'est une liste de champs textes qui sont utilisés à différents endroits dans l'application du dictionnaire, pour afficher de l'information (pas nécessairement associé à une variable).
- DICT_MOTSCLES : Fait la liste des mots clés qui peuvent être utilisés pour retrouver chaque variable, à l'aide du moteur de recherche.
- DICT_QUESTORIGINE : Conserve le questionnaire, la section, la page et la question où on peut retrouver chaque variable qui provient d'un questionnaire. Une variable peut se retrouver dans plusieurs questionnaires.
- DICT_VARIABLES : C'est la liste de base du dictionnaire de variables. Toutes les variables doivent s'y retrouver.
- DICT_VARIABLES_CALC : Conserve les variables dites « calculées » qui sont dérivées des variables provenant du questionnaire.
- DICT_VARIABLES_CALC_DEP : Normalement, chaque variable calculée devrait utiliser au moins une autre variable pour son calcul, et ces variables sont listées dans cette table.
- DICT_VARIABLES_CODES : Conserve la signification des codes pour les variables (-90 = vide, 1 = oui, 2 = non, etc.)
- DICT_VARIABLES_FORMATS : Permet de lier un code avec une variable. Le nom des formats est celui utilisé dans le programme de *recherche* de SAS.
- DICT_VARIABLES_FORMATSTATS : Fait la liste des logiciels où sont utilisés les variables, avec leur format dans ce logiciel.

- DICT_VARIABLES_QUEST : C'est la liste des variables qui proviennent du questionnaire, avec quelques informations supplémentaires (comme la table où se trouve la variable dans la base de données, son format, etc.)
- DICT_VARUTILISATION : Conserve les utilisations faites de la variable par les chercheurs, assistants, etc.
- ENTLAR_LISTE_DEVISES : C'est une table qui est utilisée lors du chargement pour savoir quelles devises sont utilisées dans l'entrepôt.

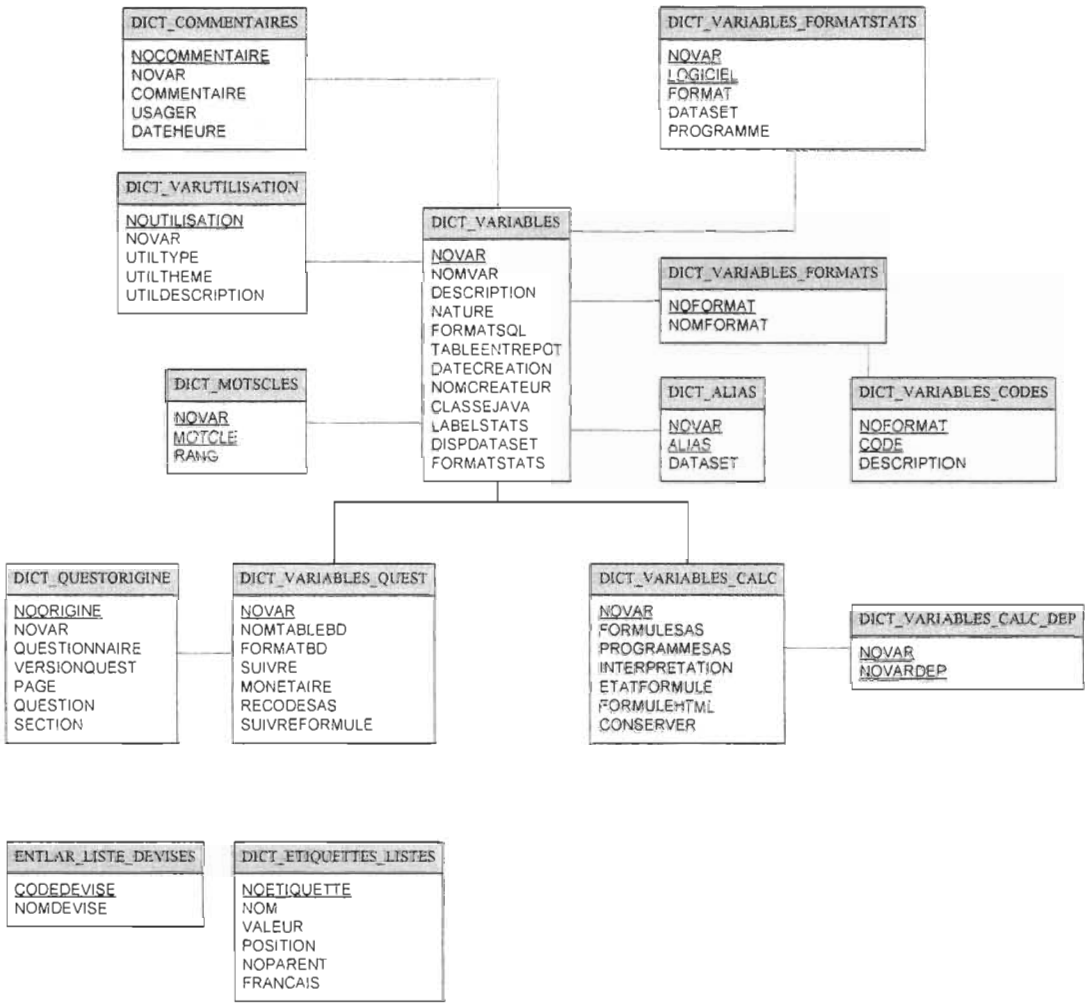


Figure 4 Diagramme entité-relation du schéma des métadonnées

On peut remarqué qu'il y a une relation de spécialisation au niveau des variables qui proviennent du questionnaire ou les variables calculées. En effet, une variable ne peut pas à la fois provenir du questionnaire et d'un calcul. Les tables ENTLAR_LISTE_DEVISES et DICT_ETIQUETTES_LISTES sont indépendantes des autres tables et sont utilisées par des applications sans avoir de lien particulier aux variables du dictionnaire.

4. Le schéma des données de l'entrepôt ENTLAR_DATAMARTS

Pour l'instant, le schéma ENTLAR_DATAMARTS contient tous les magasins de l'entrepôt de données du LaRePE. Ce schéma doit en tout temps contenir des données correctes, il faut donc s'assurer d'avoir des copies de sauvegarde faites régulièrement (normalement, c'est le Service de l'informatique de l'université qui s'en charge).

4.1 Magasin historique

Le magasin historique est un des magasins du schéma ENTLAR_DATAMARTS, et c'est ce magasin qui permet d'extraire des jeux de données à l'aide du Dataset Maker. Le magasin historique conserve toutes les versions des questionnaires qui ont été saisis dans la base de données manufacturière, avec leur date d'insertion dans l'entrepôt. C'est pour cette raison qu'il est possible de recréer un jeu de données en fonction d'une date. Il est alimenté par le logiciel de chargement des données.

Bien que plusieurs autres tables soient présentes dans le schéma ENTLAR_DATAMARTS, seules celles qui sont utilisées pour le magasin historique sont listées :

- QM_FINANCIER_01 : Contient toutes les variables financières.
- QM_GENERAL_01 : Contient tous les questionnaires et les variables générales.
- QM_GENERAL_01_SUIVI : C'est la table générale, à laquelle on a appliqué le suivi pour les variables concernées.

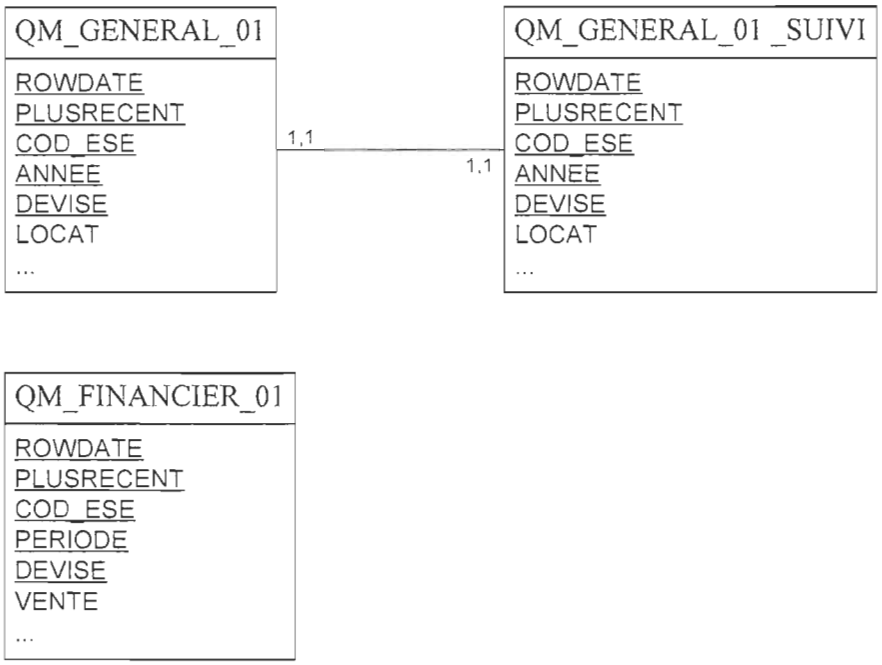


Figure 5 Diagramme entité-relation du magasin historique

Comme on le voit dans la figure 4, la table de suivi contient essentiellement les mêmes enregistrements que la table générale, avec les mêmes champs. La seule différence est dans le contenu de quelques-uns de ces champs, qui ont été calculés pour que les données non requestionnées soient recopiées dans un questionnaire ultérieur.

4.2 Magasin dimensionnel

L'entrepôt de données supporte une structure dimensionnelle qui permet à des outils d'OLAP et de forage de données un accès simplifié. Pour l'instant, le seul outil qui utilise pleinement les structures dimensionnelles est le logiciel ContourCube, auquel on peut accéder à partir du site Web de l'entrepôt.

Plusieurs tables de faits et de dimensions sont présentes, et elles sont toutes alimentées par l'application de chargement des données à partir du magasin historique. Le magasin dimensionnel tire toutes ses données du magasin historique, en utilisant les questionnaires les plus récents. Cependant, il serait possible, au besoin, de recréer le magasin dimensionnel à partir d'anciennes versions des données. Il serait aussi possible de créer une table de faits qui utiliserait toutes les versions des données, mais pour l'instant seules quelques dimensions ont été choisies pour fins de démonstration.

Les tables suivantes sont utilisées pour les dimensions :

- STAR_GT_AGE : Dimension de l'âge de l'entreprise.
- STAR_GT_CHIFFRE_AFFAIRES : Dimension du chiffre d'affaires de l'entreprise.
- STAR_GT_EMPLOYE : Dimension du nombre d'employés.
- STAR_GT_TAUX_CROISSANCE : Dimension du taux de croissance de l'entreprise.
- STAR_GT_TYPE_PRODUCTION : Dimension du type de production de l'entreprise.
- STAR_QUEST_CLIENT : Dimension qui permet d'identifier le client (intermédiaire) du Laboratoire qui a transmis le questionnaire.
- STAR_QUEST_REGION : Dimension de la région de l'entreprise.
- STAR_QUEST_SECTEUR : Dimension du secteur de l'entreprise.

La table STAR_CUBES est utilisée pour stocker les fichiers du logiciel ContourCube dans un champ BLOB. Le logiciel peut alors les télécharger en toute sécurité via le Web. Cette façon de faire permet aussi d'archiver les anciennes versions des cubes.

4.2.1 Point de vue des questionnaires

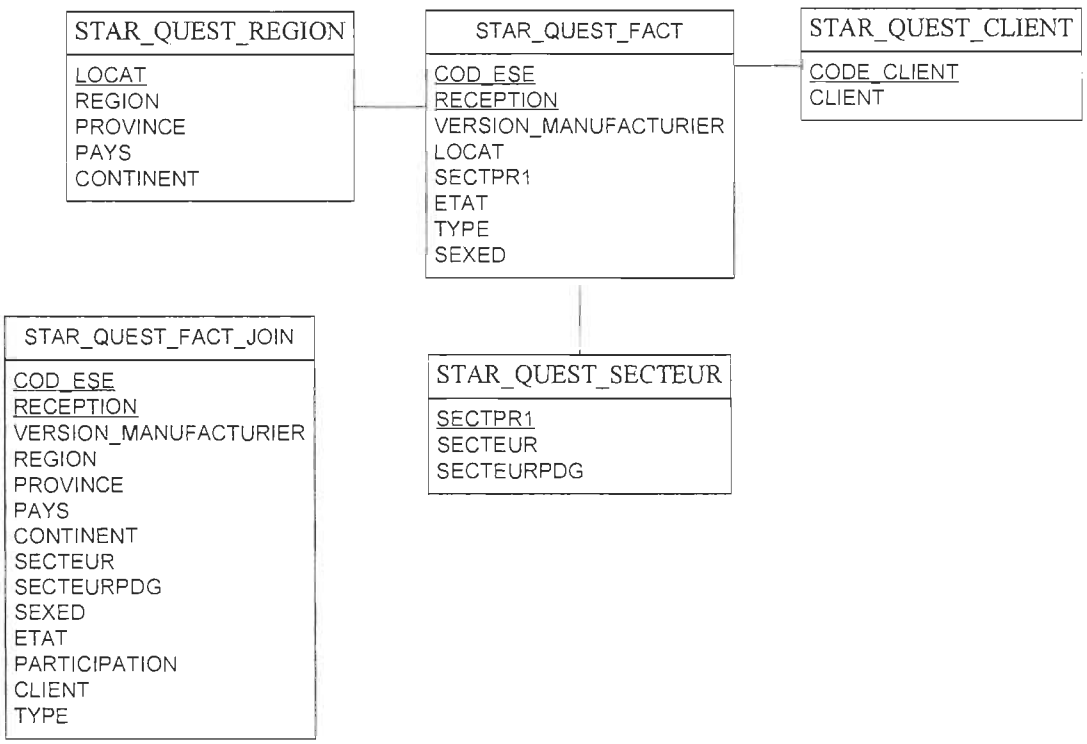


Figure 6 Schéma en étoile pour la table de faits des questionnaires

Comme on peut le voir, pour l'instant la liste des dimensions est très restreinte. En temps normal, il faudrait au moins avoir une table de dimension supplémentaire pour le temps, et une autre pour chaque champ qui est présent dans la table de faits. Cependant, le magasin dimensionnel n'est qu'un prototype qui devra évoluer puisque les utilisateurs n'avaient pas vraiment d'idée sur ce qui était réellement possible. C'est pourquoi ce schéma, ainsi que les autres magasins dimensionnels actuellement en fonction, sera appelé à changer beaucoup lorsque les utilisateurs commenceront à donner leur opinion.

Pour l'instant, le programme de chargement des données s'occupe de peupler la table STAR_QUEST_FACT_JOIN en y ajoutant quelques éléments qui ne se trouvent pas encore dans les tables de dimensions. C'est cette table qui est utilisée par ContourCube pour afficher ses informations.

4.2.2 Point de vue des entreprises

Cette table de faits utilise seulement le plus récent questionnaire qui a été reçu de la part d'une même entreprise. De cette façon, on a accès à des informations d'une autre nature, comme par exemple le nombre d'entreprises qui proviennent d'une région spécifique (plutôt que le nombre de questionnaires qui proviennent de cette même région). Cette table de faits ressemble à celle des questionnaires, avec quelques dimensions en moins, pour l'instant. En effet, la date de réception d'un questionnaire ne veut pas dire grand-chose quand on parle d'une entreprises. Certains autres champs prennent aussi une nouvelle signification.

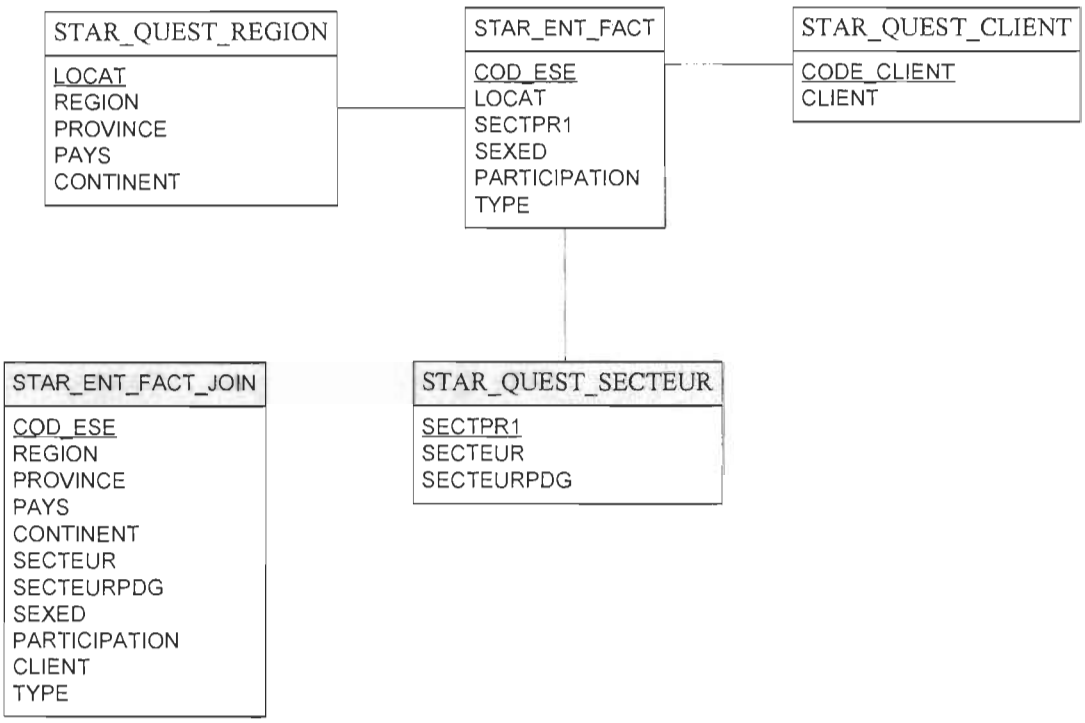


Figure 7 Schéma en étoile pour la table de faits des entreprises

Comme dans le cas précédent, la table STAR_ENT_FACT_JOIN est utilisée par ContourCube pour générer les données à afficher.

4.2.3 Point de vue du groupe témoin

La table de faits utilisée pour le groupe témoin, ainsi que les dimensions, font l'objet d'une démonstration qui n'a pas été complétée. Il faudra attendre d'avoir des demandes de la part des utilisateurs pour voir s'il est intéressant de poursuivre dans cette voie.

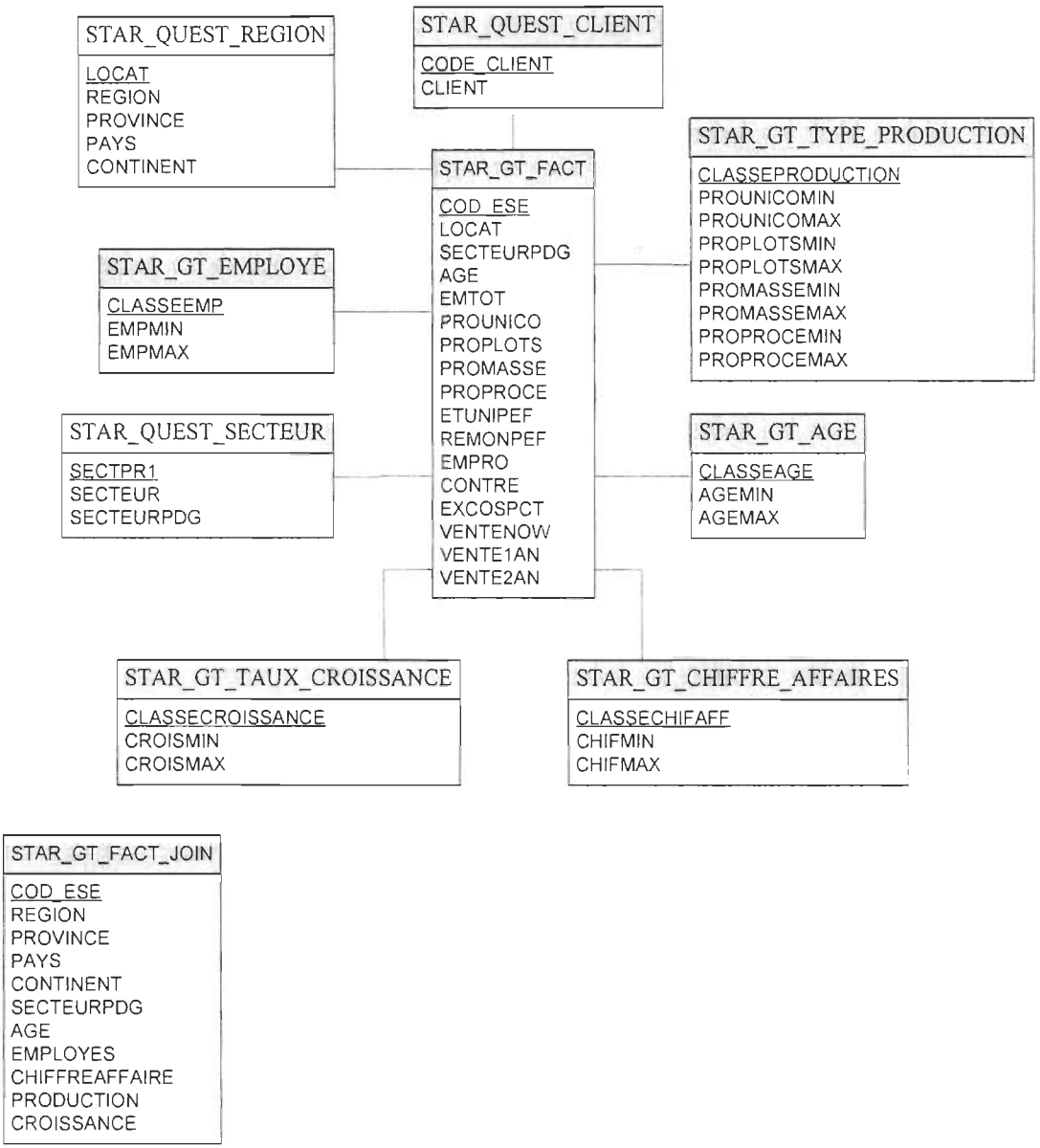


Figure 8 Schéma en étoile de la table de faits du groupe témoin

Il faut noter que certains éléments de cette table de faits ne sont pas encore calculés, comme le type de production et le taux de croissance. Les calculs sont assez complexes et ont traditionnellement été faits dans un logiciel statistique. Il serait évidemment possible de les faire pour les insérer dans la table STAR_GT_FACT_JOIN, mais il faudrait définir correctement la formule à utiliser (c'est plutôt nébuleux dans les logiciels de statistiques).

4.3 Les jeux de données

Le Dataset Maker est un logiciel qui a été développé spécialement pour faciliter l'utilisation des données dans l'entrepôt. Il accède au magasin historique et permet de sélectionner les variables et la façon de les présenter pour les utiliser dans un logiciel tiers pour l'analyse statistique.

Tous les jeux de données créés avec le Dataset Maker sont conservés, ainsi qu'une référence pour les variables demandées. Ces informations sont stockées dans les tables suivantes :

- QM_DATASET : Conserve l'information ainsi que l'archive ZIP du jeu de données créé par l'utilisateur.
- QM_DATASET_VARIABLES : Conserve la liste des variables demandées par l'utilisateur pour son jeu de données.

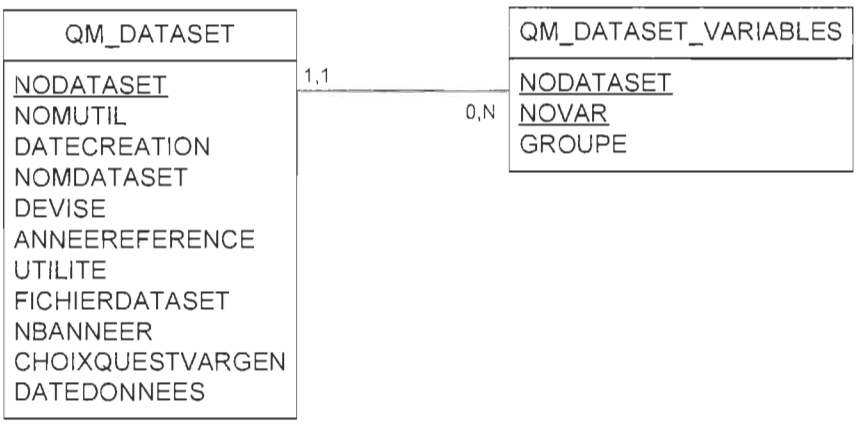


Figure 9 Diagramme entité-relation des tables de jeux de données

4.4 Le taux de change

La présence de plusieurs devises dans l'entrepôt complique la compréhension des données, mais elle est néanmoins essentielle à cause de la nature internationale des sources de données. C'est pourquoi il faut s'assurer que toutes les données monétaires sont accessibles dans les devises supportées par l'entrepôt. C'est le logiciel de chargement de l'entrepôt qui se charge de calculer les divers taux de change et de placer les données aux bons endroits. Il est ensuite possible de référer à une devise particulière à l'aide du champ DEVISE qui est contenu dans toutes les tables où figurent des informations monétaires.

Une table permet de conserver l'ensemble des taux permettant la conversion entre les devises, c'est la table ENTLAR_TAUX CHANGE.

Table ENTLAR_TAUX_CHANGE

Cette table conserve tous les taux de change utiliser pour convertir les données monétaires de l'entrepôt. Normalement ce calcul est effectué lors du chargement des données.

Champs :

- MOIS : C'est un champ date qui représente le dernier jour du mois en cause. Le taux de change est toujours celui du dernier jour ouvrable de ce même mois, mais le champ dans la table utiliser la fonction LAST_DAY pour que les algorithmes utilisant la date puissent se retrouver plus facilement.
- DEVISESOURCE : C'est la devise source du taux, ou si l'on préfère celle qui équivaut à 1 quand on dit : pour 1$US, on obtient x$CAN
- DEVISEDEST : C'est la devise de destination, ou si l'on préfère celle qui équivaut à x quand on dit : pour 1$US, on obtient x$CAN.
- STOCKS : Le taux stock, saisi à la main.
- FLUX : Le taux flux, calculé à partir de la moyenne des 12 derniers taux STOCKS (incluant le mois courant).

5. Les package, fonctions, procédures PL/SQL et les vues

Dans le schéma ENTLAR_METADATA, il y a des séquences qui permettent de créer des clés primaires uniques pour plusieurs tables. Ces séquences sont : SEQ_COMMENTAIRE, SEQ_NOFORMAT, SEQ_NOVAR, SEQ_ORIGINE et SEQ_UTILISATION qui sont respectivement associées aux tables DICT_COMMENTAIRES, DICT_FORMAT, DICT_VARIABLES, DICT_QUESTORIGINE et DICT_VARUTILISATION.

Dans le schéma ENTLAR_ETL, il y a un PACKAGE PL/SQL qui permet de calculer les taux de conversion pour le taux de change lors du chargement. C'est le package PKG_TAUX_CHANGE. La fonction FCT_CV_DEVISE utilise plusieurs paramètres pour retourner la valeur monétaire dans la devise désirée.

Les champs obligatoires sont :

- montant : C'est la donnée monétaire à convertir.
- taux_change_origine : Le taux de change de la devise d'origine à la date correspondant au montant.
- taux_change_destination : Le taux de change de la devise destination à la date correspondant au montant.
- devise_origine : Le sigle de la devise d'origine, utilisé pour savoir si on doit utiliser la devise de référence (US).
- devise_destination : Le sigle de la devise de destination, utilisé pour savoir si on doit utiliser la devise de référence (US).

Dans ce même schéma, il existe aussi deux vues. La première est QM_VUE_GENERAL_MANUFACTURIER qui fait une jointure sur toutes les

tables du schéma MANUFACTURIER pour faciliter le chargement des données. Il est important de mettre cette vue à jour lorsqu'on ajoute ou retire des champs dans la base de données manufacturière. L'autre vue sert à calculer les taux de change, c'est la vue VUE_ENTLAR_CHANGE_REFERENCE. Cette vue permet de combiner toutes les possibilités de taux de change, incluant la devise de référence à la fois comme source et comme destination. Elle facilite beaucoup le travail de calcul des taux de change.

Dans le schéma ENTLAR_DATAMARTS, il existe une fonction ELIMINER_CODES. Cette fonction permet de retirer les codes spéciaux -99, -97, -90 et -88, et retourne null à la place.

6. La sécurité avec les schémas

Comme il a été mentionné précédemment à plusieurs reprises, une des raisons de l'utilisation de plusieurs schémas Oracle est la sécurité de l'application. Le schéma ENTLAR_WEB est celui où la sécurité est la plus importante. En restreignant le serveur Web à ce schéma, on oblige chaque autre élément de l'entrepôt à être activement mis à la disponibilité du schéma Web, diminuant le risque d'exposer des données qui ne devraient pas être accessibles à l'extérieur. De plus, le schéma Web dispose de peu de droits d'accès en écriture, ce qui limite les dommages qu'un éventuel incident de sécurité pourrait engendrer.

Le schéma ENTLAR_METADATA n'a accès à aucunes ressources à l'extérieur des ressources de son propre schéma. Le schéma ENTLAR_DATAMARTS n'utilise aucune ressource à l'extérieur de son schéma. Il est d'ailleurs fortement recommandé de ne jamais donner un accès à ce schéma en utilisant directement le compte ENTLAR_DATAMARTS, ce qui donnerait un accès complet sans restrictions à toutes les structures et données du schéma.

Le schéma ENTLAR_ETL accède à de nombreuses ressources. Il doit disposer d'un accès en lecture pour toutes les sources de données à importer dans l'entrepôt. Il doit aussi pouvoir lire les tables du schéma ENTLAR_METADATA pour quelques calculs, comme le suivi et le taux de change. Et il doit évidemment avoir des accès en lecture et en écriture dans les tables de données du schéma ENTLAR_DATAMARTS pour mettre à jour les données. Le schéma de chargement devrait donc faire l'objet d'une sécurité importante puisqu'il est possible de faire beaucoup de ravage si on met n'importe qui au volant... Finalement, le schéma ENTLAR_WEB n'a pas accès pas au schéma ENTLAR_ETL. Mais il a besoin d'un accès en lecture seule sur le magasin historique afin de permettre la création de jeux de données. Il doit aussi donner un accès au magasin dimensionnel afin de mettre à jour les données de ContourCube. Il a finalement besoin d'accès en lecture et écriture dans le schéma ENTLAR_METADATA pour permettre aux utilisateurs d'afficher les informations, et ceux qui ont la permission doivent aussi pouvoir mettre à jour ces informations.

ANNEXE E

Le dictionnaire de variables

LE DICTIONNAIRE DE VARIABLES

1. Introduction

Le dictionnaire de variables est une application qui sert principalement à documenter l'ensemble des variables de l'entrepôt. Les variables sont des champs qui peuvent provenir de sources diverses, dont des questionnaires et des applications Web, ou de combinaisons d'autres variables à l'aide de formules. Chaque variable possède une description, l'information sur sa provenance est la valeur des données qu'elle représente (la légende), ainsi que de nombreuses autres informations pertinentes pour l'entretien de l'entrepôt mais surtout pour l'utilisation des variables par des chercheurs. Le dictionnaire sert de documentation pour l'entrepôt, mais c'est aussi un élément actif qui affecte les données par les informations qui sont saisies sur chaque variable. Il est très important que ces données soient à jour en tout temps. Les utilisateurs de l'entrepôt qui en ont l'autorisation peuvent utiliser le dictionnaire, ainsi que son moteur de recherche de variables pour les aider dans leurs projets de recherche. Un schéma de l'entrepôt sert spécifiquement à supporter les métadonnées, incluant le dictionnaire de variables.

2. Les variables

L'entrepôt contient beaucoup de données dans ses différents schémas et magasins, et ces données peuvent être représentées sous formes de dimensions (pour les magasins à modèle dimensionnel) ou de champs. Le dictionnaire de variables documente plus spécifiquement les champs des différentes tables de l'entrepôt. Ce sont ces champs qui sont exportés avec le Dataset Maker lors de la création de jeux de données pour la recherche. Le Dataset Maker peut aussi servir à créer des jeux de données pour des applications, comme des rapports. La documentation de champs est essentielle à la réalisation de tout projet de recherche sérieux, sinon elle est à recommencer à chaque fois qu'un nouveau jeu de données est créé (un peu comme c'était le cas auparavant). En ayant un endroit bien identifié pour documenter les variables, tout le monde peut alors partir sur une base solide pour consulter tous les jeux de données créés à partir de l'entrepôt, sans avoir à chercher dans de nombreux documents la signification exacte de chaque variable, ainsi que ses particularités. Une variable peut être un champ qui est saisi à partir d'un questionnaire, un champ qui provient d'une autre base de données (alimentée par une application Web, par exemple), ou une valeur obtenue à partir d'une formule qui utilise d'autres variables. Il est important de pouvoir retracer l'origine d'une variable pour pouvoir en faire bon usage, et bien interpréter les données.

Il y a de nombreux champs d'information qui sont saisis, et ces informations peuvent être appelées à changer. Toutes les données sont conservées dans des tables, et certains types d'information utilisent des tables adaptées spécifiquement à cet effet, comme par exemple tous les questionnaires d'où peut provenir une variable qui sont

stockés dans la table DICT_VARIABLES_QUESTORIGINE. D'autres informations, comme la description et la nature de la variable, sont stockées directement dans la table générique DICT_VARIABLES. Plus de détails sur les tables de la base de données sont décrits dans la section 3.

Les variables peuvent exister sous la forme (aussi appelée nature) générale, financière ou mixte. Une variable générale provient d'un questionnaire ou d'un calcul de variables générales, une variable financière provient des états financiers d'une entreprise ou d'un calcul sur ces variables financières et une variable mixte est une variable calculée à partir de variables financières et générales. Il est important de savoir à tout moment si une variable est générale, financière ou mixte et si elle provient directement du questionnaire ou si elle est calculée. Toutes ces informations sont notées dans le dictionnaire de variables.

3. La base de données

L'entrepôt de données comporte une section réservée à la documentation, c'est le schéma ENTLAR_METADATA. Dans ce schéma, on trouve toutes les tables du dictionnaire. Une description précise de chaque table est faite dans l'annexe B. La table principale, celle qui contient la liste de toutes les variables, est DICT_VARIABLES. C'est dans cette table qu'on définit le numéro de chaque variables (NOVAR). C'est un numéro qui est utilisé comme référence principale pour identifier uniquement toutes les variables dans le dictionnaire, puis aussi pour plusieurs applications de l'entrepôt. L'association NOMVAR et NOVAR permettent de retrouver à tout moment une variable dans l'entrepôt. Pour des raisons de convivialité lors de la programmation, c'est bien le NOM de la variable qui est utilisé pour nommer les champs dans les tables de l'entrepôt, et non le numéro de la variable. Cependant, la plupart des applications utilisent le numéro de la variable pour effectuer leurs traitements. Une des raisons est qu'il est déjà arrivé que des noms identiques de variables ne représentent pas les mêmes données, et l'utilisation d'une clé numérique permet d'éviter ce genre de problème, tout en donnant la possibilité de renommer une variable en minimisant les inconvénients. Partout dans le schéma ENTLAR_METADATA, c'est le champ NOVAR qui est utilisé comme clé lointaine pour identifier une variable.

4. La conception de l'application Web

L'application Web du dictionnaire se trouve dans le répertoire /dictionnaire de l'intranet du LaRePE. Le code est fait en JSP, Java et utilise aussi du JavaScript. La page principale du dictionnaire est « recherche_variables.jsp », et c'est cette page qui appelle toutes les autres. Il y a de nombreuses pages qui sont appelées à l'intérieur d'autres pages pour effectuer des traitements spécifiques. Par exemple, la page « afficher_details.jsp » appelle la page « afficher_details_supp_quest.jsp » pour la

section des informations supplémentaires des variables provenant du questionnaire. Des validations JavaScript sont faites dans les pages, les scripts se trouvent dans le répertoire /res et /dictionnaire/res de l'intranet du LaRePE.

Les « beans » Java sont utilisés dans les page JSP pour supporter le chargement des données qui sont passées en paramètre lorsque les variables sont modifiées, ou sinon pour charger les données de chaque variable à partir de la base de données. Chaque variable est associée à un bean Java, une classe qui représente la sorte (nature) de cette variable et qui permet de charger les données ou de les conserver dans la base de données. Les classes sont VariableQM, VariableQMCalculee et VariableQMQuestionnaire du package larepe.entrepot.variable. La classe VariableQM supporte la création de nouvelles variables, ainsi que le chargement et sauvegarde des attributs communs à toutes les variables (novar, nom, description, etc.). Les classes VariableQMCalculee et VariableQMQuestionnaire supportent respectivement les variables calculées et les variables provenant du questionnaire, ainsi que les champs plus spécifiques qui leur sont associés.

La table DICT_VARIABLES possède un champ CLASSEJAVA qui est utilisé pour déterminer quelle classe est utilisée pour créer le bean de la variable dans les pages *afficher_details.jsp* et *modifier_variables.jsp*. L'utilisation de l'héritage permet de séparer le code spécifique à chaque variable, et à ne pas tenir compte des attributs qui ne sont pas concernés par chaque type de variable. Il sera ainsi possible de créer des pages d'affichage spécifiques aux variables de chaque futur projet qui sera intégré à l'entrepôt.

Lorsque des changements sont apportés à une variable avec la page *modifier_variable.jsp*, une classe de chargement, ChargeurWeb du package larepe.entrepot.variable.chargeursWeb est utilisée pour récupérer les paramètres passés par méthode POST. Ces chargeurs fonctionnent avec le même héritage que la classe VariableQM, c'est pourquoi chaque classe VariableQM doit posséder son propre chargeur pour les paramètres Web. Les chargeurs sont utilisés pour diminuer la taille des classes de variable, qui sont spécialisée dans le chargement et conservation des données de la base de données. Cependant, certains mécanismes de chargement des beans pourraient être utilisés (comme <jsp:setproperty ... />) pour diminuer le code nécessaire pour charger les données dans les sortes de variables plus simples.

Le package larepe.entrepot.variable.recherche comprend les classes qui sont utilisées par le moteur de recherche du dictionnaire. Les classes servent à récupérer les critères de recherche utilisés dans le moteur de recherche, ainsi qu'à sélectionner les variables correspondantes dans la base de données. Le panier de variables se trouve aussi dans ce package. Le fonctionnement détaillé de ces classes est décrit dans la *javadoc*.

5. Utilisation de l'application Web du dictionnaire

Afin de donner un accès aux informations du dictionnaire, une application Web a été développée et intégrée à l'intranet du LaRePE. Ce dictionnaire est sécurisé et seules les personnes qui ont le droit de l'utiliser y ont accès. Il permet de faire une recherche parmi les différentes informations conservées sur les variables, et aussi d'afficher cette information. Une liste des variables avec une description sommaire est aussi accessible.

5.1 La recherche de variables

Pour faciliter l'utilisation du dictionnaire, un moteur de recherche a été ajouté et certains critères sont disponibles. Cependant, ces critères de recherche sont préliminaires, et dépendent directement de l'information disponible dans le dictionnaire. Si les informations sont incomplètes ou incorrectes, le moteur de recherche ne pourra pas fonctionner correctement.

Il est possible de faire afficher seulement les variables que l'on désire grâce à l'outil de recherche. On peut rechercher des variables en donnant divers critères :

- Le nom d'une variable
- Un ou des mots-clés en rapport avec la ou les variables
- Un logiciel d'utilisation de la ou les variables (ex : SAS-PDG)
- Un jeu de données (data set) utilisant la ou les variables
- Le type de la ou les variables recherchées (questionnaire, calculée, etc.)
- Le questionnaire et/ou version et/ou page et/ou question où l'on retrouve la ou les variables recherchées.

Lorsqu'on accède au dictionnaire, l'interface principale (figure 1) affiche une liste des variables contenues dans le dictionnaire, ainsi que le panier de variable. La panier de variable est réservé à l'utilisation du Dataset Maker et ne sert pas spécifiquement au dictionnaire de variables. Il est possible de sélectionner une ou plusieurs variables dans la liste, et de cliquer sur le bouton Afficher pour passer en mode de visualisation des informations sur les variables.

Dictionnaire de variables

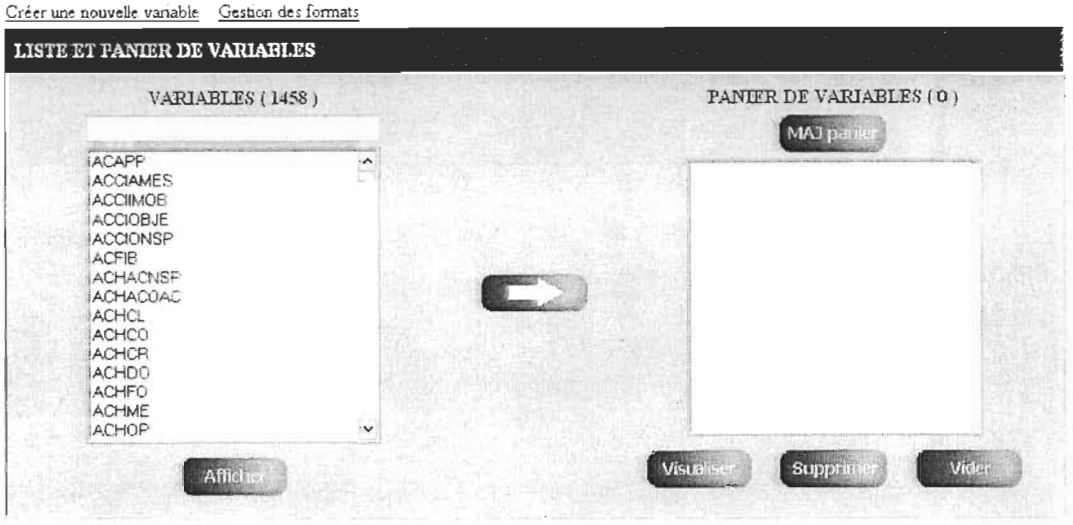


Figure 1 Interface principale du dictionnaire de variables

Le moteur de recherche (figure 2) se situe dans le bas de cette page. Les critères permettent de rechercher les variables à partir de leur nom, de mots clés, du logiciel qui les utilise, du type de variable (questionnaire, calculée) ou du questionnaire d'origine de la variable. La liste des résultats de la recherche est affiché dans la liste de variables (figure 3).

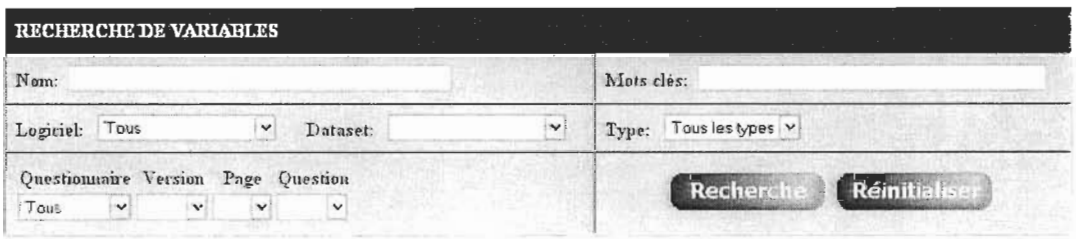


Figure 2 Interface de recherche de variables

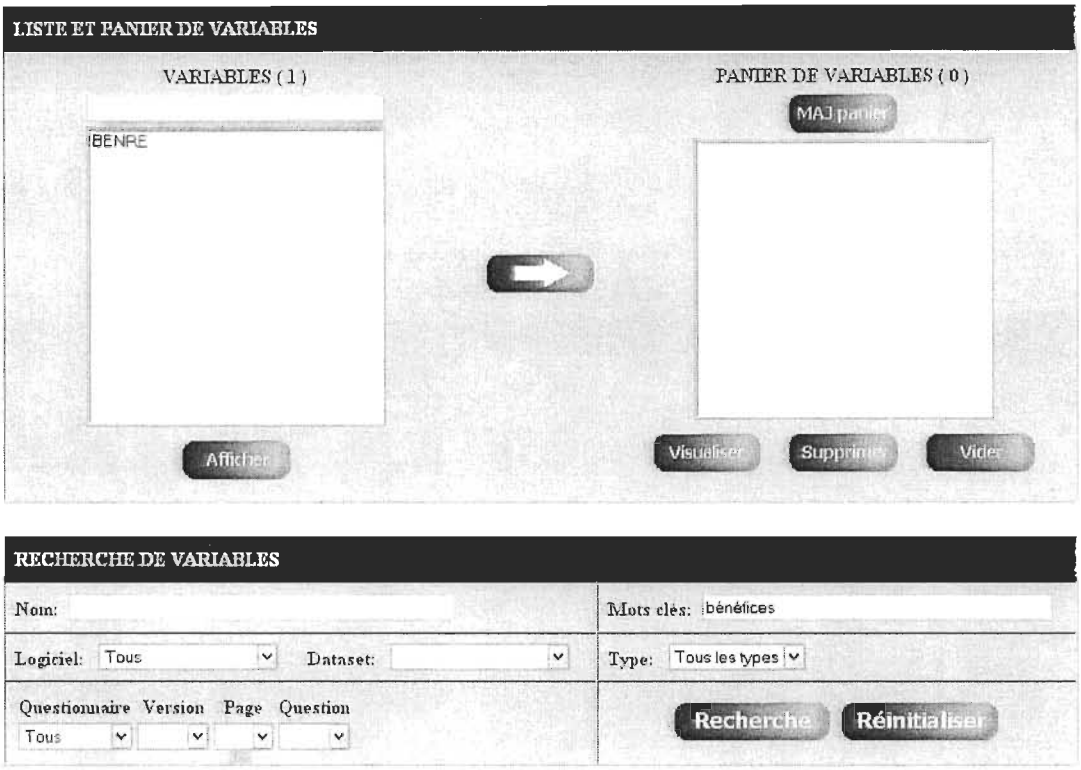


Figure 3 Recherche avec le mot clé "bénéfice"

5.2 La consultation de variables

Pour obtenir des informations sur une variable, il faut sélectionner une variable dans la liste et cliquer sur le bouton *Afficher*. Une nouvelle page s'ouvre et l'information détaillé de la variable est affichée. Les informations générales (figure 4) sont le nom de la variable, la nature (général, financier, mixte), la date de l'ajout de la variable au dictionnaire, ainsi que la personne qui l'a ajoutée, et une description courte qui est copiée dans les applications statistiques lors de la création de jeux de données.

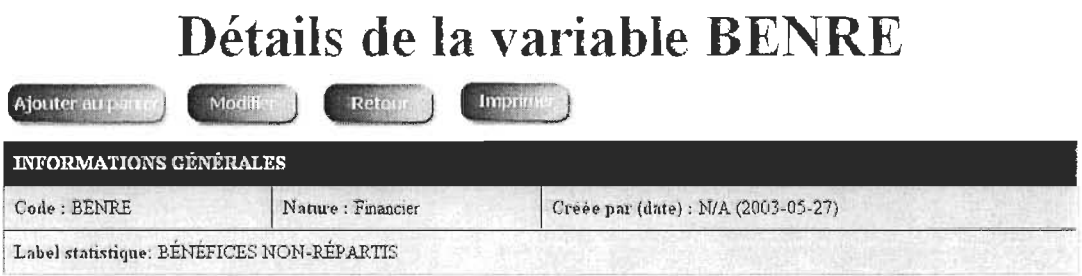
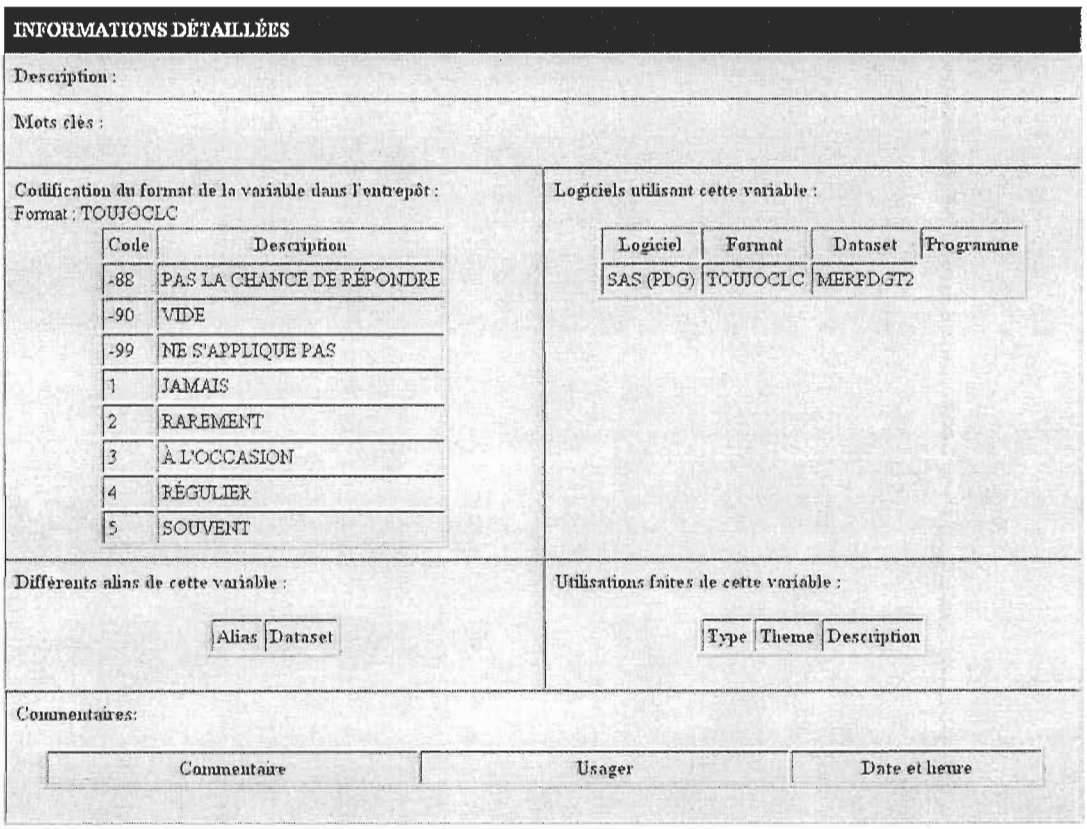


Figure 4 Section "informations générales" de la variable BENRE

L'information détaillée de la variable est affichée sous les informations générales, et servent à donner plusieurs informations pertinentes lors de l'utilisation des variables pour la recherche. Une description plus précise de la variable est faite, ainsi qu'une liste de mots clés utilisés pour trouver cette variable. Les codes utilisés pour représenter les données sont affichés (si c'est nécessaire) avec le nom du format SAS de la variable. Les logiciels qui utilisent la variable sont aussi listés. Les alias (différents noms qui peuvent être utilisés pour des raisons diverses) et les utilisations (recherches, études, mémoires, etc.) faites de la variables sont aussi listés. Finalement, des commentaires faits par diverses personnes sur cette variable peuvent aussi être affichés.



INFORMATIONS DÉTAILLÉES

Description :

Mots clés :

Codification du format de la variable dans l'entrepôt :
Format : TOUJOCLC

Code	Description
-88	PAS LA CHANCE DE RÉPONDRE
-90	VIDE
-99	NE S'APPLIQUE PAS
1	JAMAIS
2	RAREMENT
3	À L'OCCASION
4	RÉGULIER
5	SOUVENT

Logiciels utilisant cette variable :

Logiciel	Format	Dataset	Programme
SAS (PDG)	TOUJOCLC	MERPDGT2	

Différents alias de cette variable :

Alias	Dataset

Utilisations faites de cette variable :

Type	Theme	Description

Commentaires:

Commentaire	Usager	Date et heure

Figure 5 Section "information détaillée" de la variable ACHACOAC

Les informations supplémentaires (figures 5 et 6) de chaque variable dépendent de la nature de cette variable. Pour une variable qui provient du questionnaire, on affiche le nom de la table d'où provient la variable dans la base de données source ainsi que son format dans cette table. On indique si la variable est monétaire, et si oui, si on doit utiliser un taux de change STOCK ou FLUX pour sa conversion. Un champ indique si des transformations (recodage) sont faites dans SAS. Un autre indique si la variable peut suivre entre les questionnaires, et si oui, si on doit utiliser une formule pour ce suivi. Finalement, une liste des questionnaires où se retrouve cette variable est faite. Si la variable est calculée, on donne la formule utilisée, ainsi qu'une indication pour savoir si le dictionnaire a été capable de comprendre la formule (état

de la formule). Un champ indique aussi si la formule peut être calculée lors du chargement de l'entrepôt. Pour l'instant, l'entrepôt ne supporte pas encore ce type de calcul. Les variables ne sont donc jamais calculées. Lorsque le dictionnaire reconnaît la formule, il fabrique des hyperliens (figure 7) vers toutes les variables utilisées dans cette formule.

INFORMATIONS SUPPLÉMENTAIRES (VARIABLE DU QUESTIONNAIRE)

Nom table BD: MAIN_ESE_CONT_COUT	Format dans BD: NUMBER(3)
Est-ce une variable monétaire? : Non	Recodage dans SAS:
La variable peut-elle suivre? : Non applicable	

Liste des questionnaires où se retrouve la variable:

Questionnaire	Version	Section	Page	Question
Base	4.1	Production	18	14
Mise à jour	1.1	Production	11	14
Base	5	Production	19	14

Figure 6 Section "informations supplémentaires" provenant de la variable ACHACOAC

INFORMATIONS SUPPLÉMENTAIRES (VARIABLE CALCULÉE)

Formule permettant d'obtenir la variable:

```
IF AUPROFI =. THEN DO;
   AUPROTOT =.;
   END;
ELSE IF VALEN = 1 AND AUPROFI = 0 THEN DO;;
   AUPROTOT = 1;
   END;
ELSE IF VALEN = 1 AND AUPROFI = 1 THEN DO;;
   AUPROTOT = 1;
   END;
ELSE DO;
   AUPROTOT =.;
   END;

SUM = AUPROTOT;
```

État de la formule :

Correcte

Stockage de la formule :

Non, et ne jamais calculer

Figure 7 Section "informations supplémentaires" de la variable calculée AUPROTOT

La dernière section d'information est l'information technique (figure 8). Cette information sert uniquement à la maintenance de l'entrepôt et n'est pas très utile aux autres utilisateurs, comme les chercheurs ou les étudiants.

INFORMATIONS TECHNIQUES

No de la variable : 841	Format dans l'entrepôt : N/A
Table dans l'entrepôt : QM_GENERAL_01	Classe Java : larepe.entrepot.variable.VariableQMQuestionnaire

Figure 8 Section "informations techniques" de la variable ACHACOAC

5.3 Modifier une variable

Il arrive que les informations sur une variable soient incomplètes, ou changent dans le temps. Il est aussi très important que les données du dictionnaire soient le plus à jour possible. C'est pourquoi une interface a été mise en place pour permettre de modifier les variables. Cette interface est visuellement très semblable à l'affichage de l'information détaillée des variables, ce qui facilite le repérage visuel pour la personne qui fait la modification. L'utilisateur qui désire modifier la variable doit posséder une autorisation spéciale d'écriture dans le dictionnaire. Cet utilisateur peut alors accéder à la modification de toute variable déjà présente dans le dictionnaire.

L'interface de modification du dictionnaire possède des listes de choix pour certains critères, ce qui facilite la saisie des informations (voir figure 9). D'autres informations sont des listes de plusieurs valeurs (voir figure 10, logiciel d'utilisation), et chaque liste possède une case qui permet de supprimer une valeur à l'extrême gauche de chaque ligne. Les figures 9, 10, 11, 12 et 13 illustrent les différentes informations à saisir pour chaque variable.

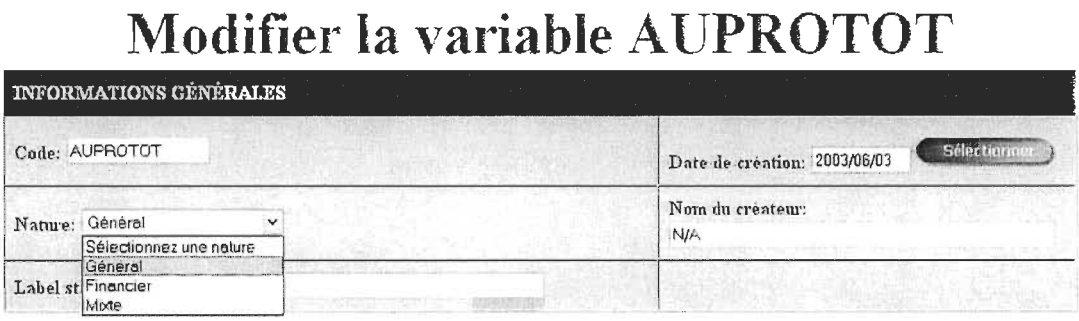


Figure 9 Modification des informations générales de la variable AUPROTOT

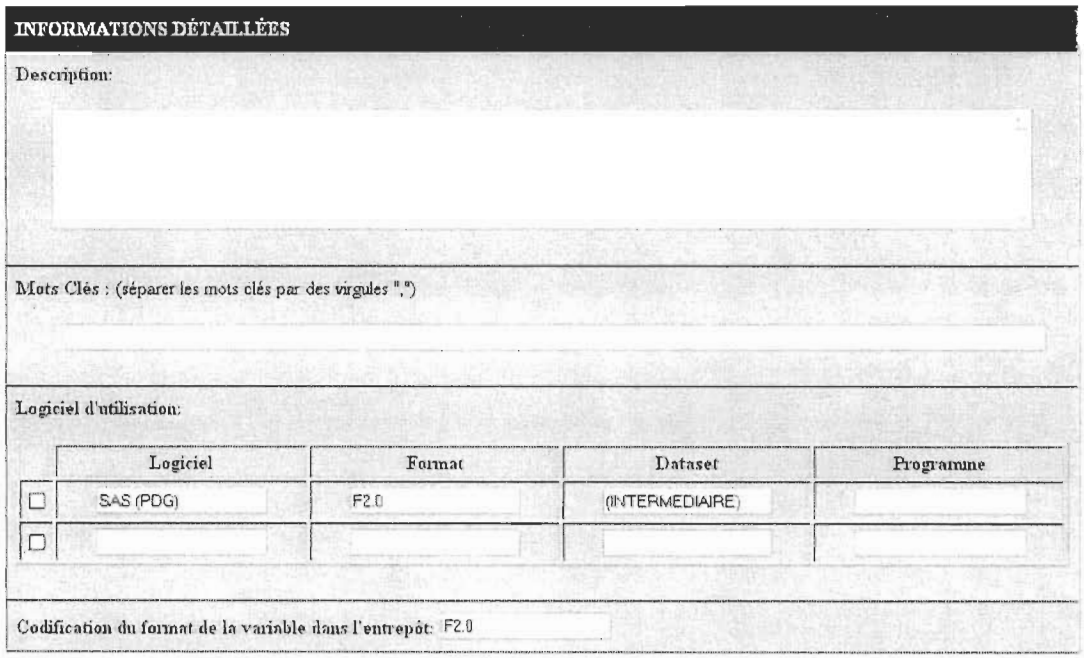


Figure 10 Modification des informations détaillées de la variable AUPROTOT (1ere partie)

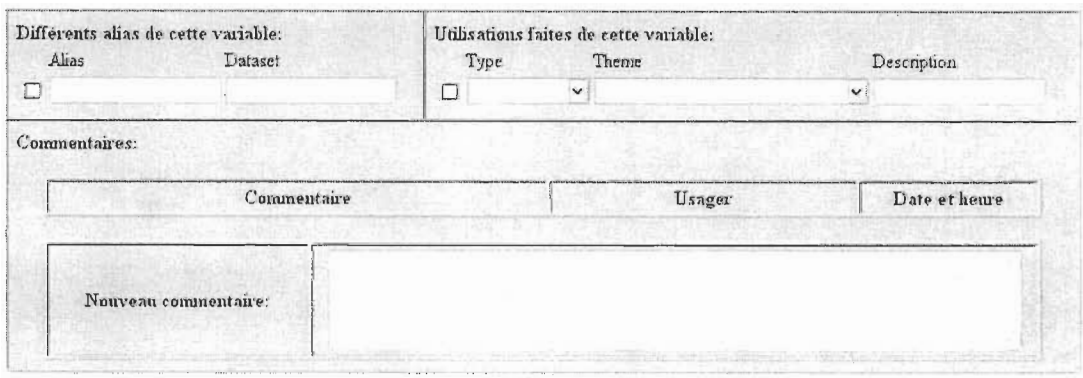


Figure 11 Modification des informations détaillées de la variable AUPROTOT (2e partie)

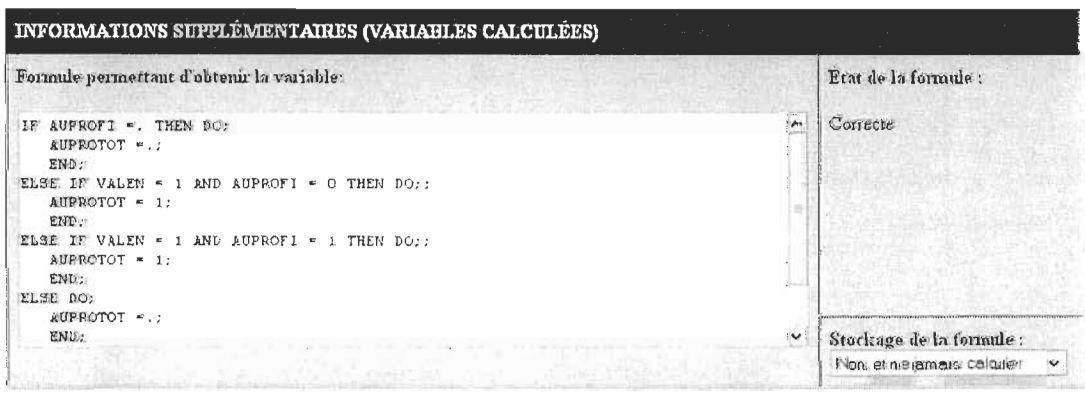


Figure 12 Modification des informations supplémentaires de la variable AUPROTOT

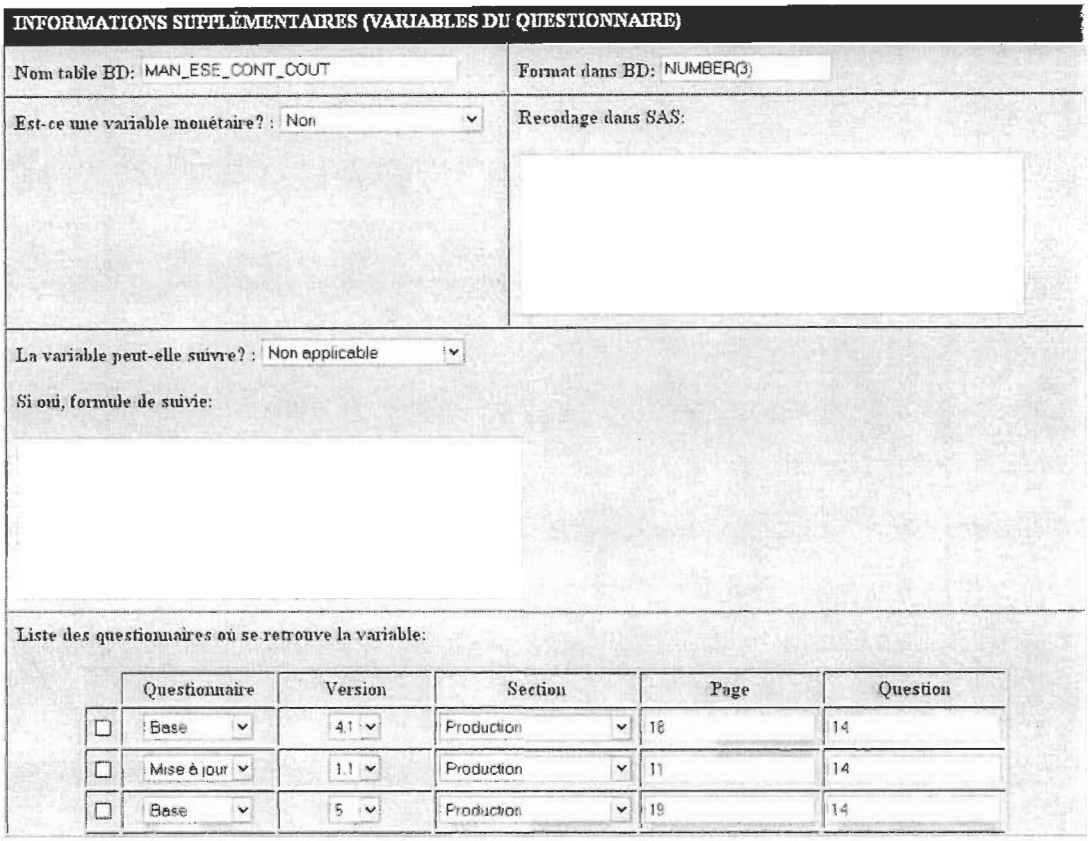


Figure 13 Modification des informations supplémentaires de la variable ACHACOAC

Il faut remarquer que les informations techniques sont créées avec la variable et ne peuvent pas être modifiées à partir de cette interface. Si jamais elles doivent être modifiées, c'est un informaticien qui s'en charge puisque ces informations servent à l'entretien et au chargement des données dans les tables de l'entrepôt.

Au bas de la page de modification, les boutons *Valider*, *Enregistrer* et *Annuler* sont affichés (figure 14). Le bouton valider permet de vérifier que toutes les informations essentielles sont saisies, et sous la bonne forme, sans toutefois modifier le contenu du dictionnaire de variables. Le bouton enregistrer valide et conserve les changements dans le dictionnaire de variable. Le bouton annuler retourne l'utilisateur à l'affichage des informations détaillées sans conserver les modifications.

Figure 14 Boutons à la fin de l'interface de modification

5.4 Créer une variable

Il peut évidemment être nécessaire d'ajouter une nouvelle variable dans le dictionnaire, qu'elle provienne d'un nouveau questionnaire ou que ce soit une nouvelle formule créée pour la recherche. C'est pourquoi un lien est disponible dans le dictionnaire pour créer une nouvelle variable. Ce lien n'est accessible qu'aux utilisateurs qui ont l'autorisation d'écrire dans le dictionnaire. Il mène vers une page (voir figure 15) qui demande quelques informations de base permettant de créer la variable. Une fois créée, elle peut être modifiée comme n'importe quelle autre variable avec l'interface de modification des variables. Les informations sur le type de variable et la nature sont très importantes, elles ne peuvent que difficilement être modifiées après la création de la variable. Il faut noter que la création d'une variable ne prépare pas encore la place pour les données de cette variable dans l'entrepôt de données. Cependant, certains champs d'information servent à gérer le comportement des données lors du chargement, comme le champ sur le suivi ou celui sur la nature monétaire de la variable.

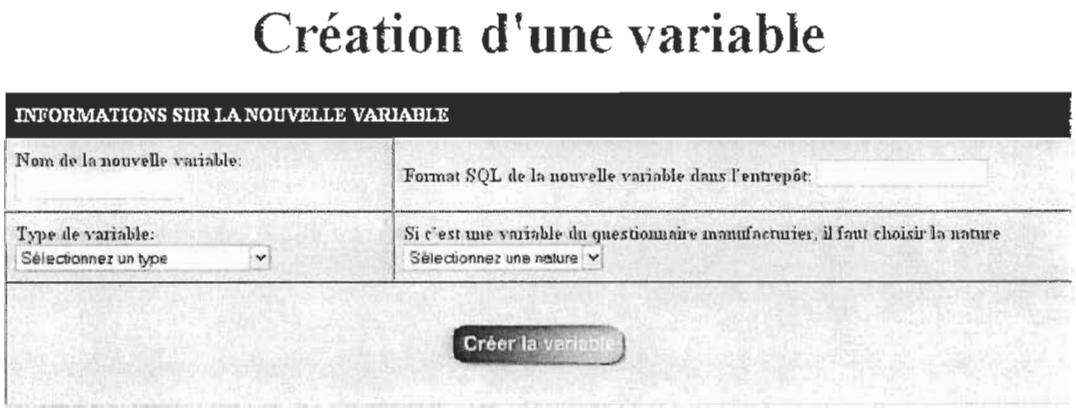


Figure 15 Création d'une nouvelle variable

5.5 Gérer les formats du dictionnaire

Les formats sont un héritage de l'application SAS, qui a toujours servi à exploiter les données pour les projets statistiques au LaRePE. Ils peuvent être gérés dans le dictionnaire à l'aide d'une interface (voir figure 16) indépendante des variables. Chaque format représente une liste de codes qui sont utilisés pour savoir ce que les données des variables signifient. Par exemple, le code 1 peut vouloir dire *oui*, le code 2 veut dire *non*, etc. Tout ces formats sont utilisés lors de la création de jeux de données.

Liste des formats existants

144 formats dans la liste

Ajouter nouveau format | Retour

Nom du format
$PERIODR
$TYPEDOS
ACCORCLA
AGEDICLA
AGEDICLB
ANAIFCLA
ANDIRCLA
ANDIRCLB

Figure 16 Affichage de la liste des formats dans le dictionnaire de variables

Il est possible de cliquer sur un format pour le modifier. Une liste des codes est alors affichée (voir figure 17), et on peut alors changer la signification des codes, en ajouter ou en enlever.

CODE	DESCRIPTION	SUPPRIMER
-88	PAS LA CHANCE DE RÉPONDRE	☒
-90	VIDE	☒
-99	NE S	☒
1	TOTAL DÉSACCORD	☒
2	DÉSACCORD	☒
3	INDÉCIS	☒
4	EN ACCORD	☒
5	TOTAL ACCORD	☒

Enregistrer

NOUVEAU CODE	NOUVELLE DESCRIPTION

Ajouter

Figure 17 Modification du format ACORCLA

Il est aussi possible de créer de nouveaux formats (voir figure 18), il faut alors saisir le nom du format ainsi que les différents codes utilisés. L'étape 1 consiste à donner le nom du format. L'étape 2 est identique à la figure 17, à l'exception que la liste des codes est alors vide lorsque le format vient tout juste d'être créé.

Ajout d'un format

Étape 1

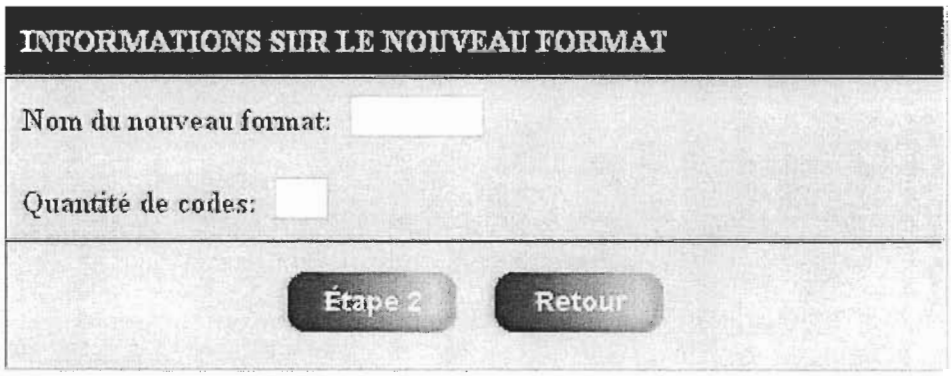


Figure 18 Création d'un nouveau format dans le dictionnaire

ANNEXE F

Le *Dataset Maker*

LE DATASET MAKER

1. Introduction

Le Dataset Maker est un outil permettant de générer des jeux de données (data set) à partir des variables choisies dans le dictionnaire de variables. Ce puissant outil permet de générer des jeux de données très rapidement (en terme de secondes). Il est rendu nécessaire à cause de la grande variété de données conservées dans l'entrepôt de données du LaRePE, et aussi pour rendre plus facile la tâche de création d'un jeu de données pour les chercheurs. De cette façon, les utilisateurs peuvent accéder eux-mêmes aux données sans devoir passer par un intermédiaire. Une certaine forme de documentation est aussi incluse aux jeux de données créés afin d'aider les utilisateurs dans leurs projets de recherche.

2. Fonctionnement du Dataset Maker

Quatre étapes sont suivies pour la création d'un jeu de données.

1- **Choix des variables**. Cette étape demande la confirmation que le choix des variables est le bon.

2- **Choix d'options et groupement des variables**. Cette étape fait la séparation des variables générales et financières. Pour les variables générales, il faut choisir l'endroit où aller chercher les données. Voici les possibilités offertes :

- Utiliser les variables qui proviennent du questionnaire de base le plus récent.
- Utiliser les variables qui proviennent du questionnaire (base ou MAJ) le plus récent. Dans ce cas, on doit aussi spécifier si dans le cas d'un questionnaire de MAJ on veut permettre aux variables autorisées à le faire suivre. C'est-à-dire que comme plusieurs variables ne se retrouvent pas dans le questionnaire de MAJ, on veut savoir si on autorise de faire passer la valeur de ces variables du questionnaire de base au questionnaire de MAJ.

Pour les variables financières, il faut choisir les variables que l'on voudra générer sous le format R1R2R3 et les variables que l'on voudra générer sous le format NOW1AN2AN.

Le format R1R2R3 générera, pour un nombre d'année spécifié, des variables se terminant par R1, R2, R3... par rapport à une année de référence. Par exemple, si on choisi la variable ACOUT pour ce type de variable et que l'on en veut pour 4 années, nous obtiendrons les variables ACOUTR1 (valeurs de la variable pour l'année de référence), ACOUTR2 (année de référence - 1), ACOUTR3 (année de référence - 2), ACOUTR4 (année de référence - 3).

Le format NOW1AN2AN générera, pour un nombre d'année spécifié, des variables se terminant par NOW, 1AN, 2AN... par rapport à la dernière année financière. Par exemple, si on choisi la variable ACOUT pour ce type de variable et que l'on en veut pour 4 années, nous obtiendrons les variables ACOUTNOW (valeurs de la variable pour la dernière année financière), ACOUT1AN (dernière année financière - 1), ACOUT2AN dernière année financière - 2), ACOUT3AN (dernière année financière - 3).

3- **Création du jeu de données**. Cette étape demande d'autres options et permet de créer le jeu de données. Il peut être créé sous plusieurs formats.

Voici les formats disponibles :

- SAS
- EXCEL
- XSL (tableau html)

Plusieurs autres options sont demandées dont la date exacte à laquelle on veut récupérer les données. Par défaut, la date du jour est utilisée mais il est possible d'aller chercher les données telles qu'elles étaient à une date antérieure. Il est aussi possible d'utiliser une clause SQL « where » pour spécifier un certain nombre seulement d'entreprises. Une vérification de la validité de la clause SQL est faite avant la création du dataset. La clause par défaut est « ETAT = 1 AND TYPE = 'M' AND LOCAT >= 1 AND LOCAT <= 17 » et elle spécifie toutes les entreprises de type manufacturier qui se trouvent au Québec.

4- **Confirmation**. Cette étape confirme que le dataset a été créé avec succès et permet de le télécharger. En effet, un fichier compressé ZIP est créé contenant les différents formats de dataset choisis. Ce fichier est sauvegarder dans l'entrepôt dans un format BLOB. À l'aide d'un servlet, il est possible de télécharger ce fichier ZIP sur un ordinateur personnel.

Le Dataset Maker permet aussi à un usager de retourner voir la liste de tous les datasets qu'il a créé dans le temps. Il lui est possible de soit télécharger à nouveau un dataset donné ou bien de recréer un dataset. Cette dernière option fera en sorte que toutes les variables qui avaient été utilisées pour un ancien dataset donné seront replacées dans le panier et toutes les options qui avaient été utilisées pour cet ancien dataset donné seront sauvegarder dans le Dataset Maker. Par conséquent, il sera possible à l'usager de recréer fidèlement le dataset ou bien de repartir de ce dataset pour en faire un autre avec plus ou moins de variables.

3. Exemple d'utilisation

Les étapes de création d'un jeu de données sont ici présentées plus en détail. La première étape (illustrée par les figures 1 et 2) permet de choisir les variables à l'aide du dictionnaire de variables, puis de confirmer le choix de ces variables.

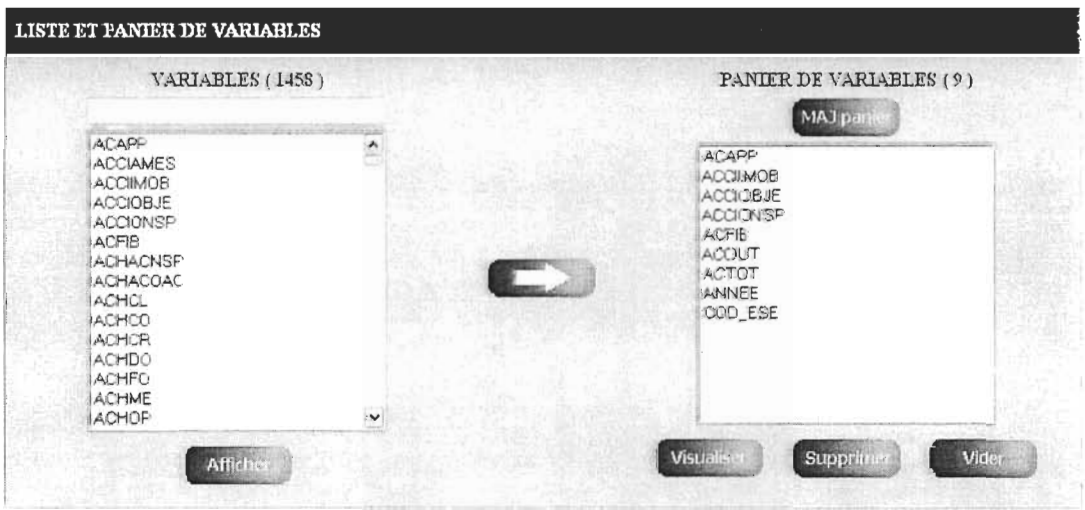


Figure 1 À partir du dictionnaire de variables, on se fabrique un panier qui est ensuite utilisé à l'étape 1 du *Dataset Maker*

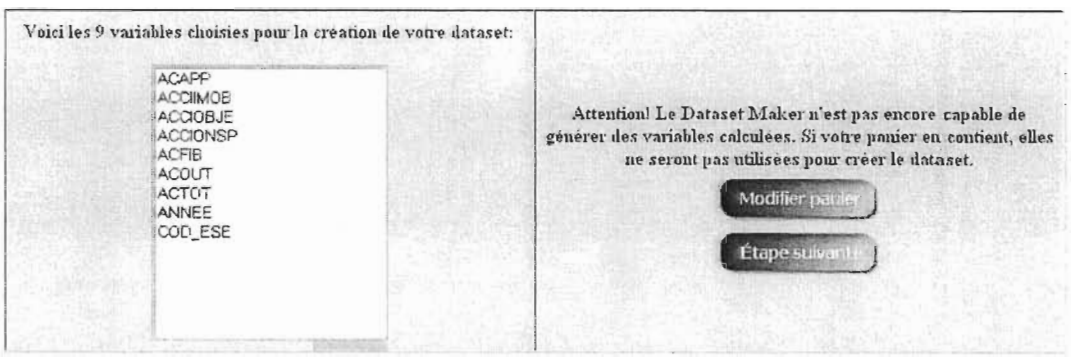


Figure 2 Étape 1 du *Dataset Maker*, confirmer le choix des variables

À la deuxième étape (voir figure 3), il faut déterminer si on veut utiliser les données du magasin historique avec ou sans le suivi. Il est aussi possible d'utiliser uniquement les questionnaires les plus complets (ceux de première année) et de ne pas utiliser les questionnaires de mise à jour. Ces deux options nécessitaient auparavant de nombreuses manipulations à la main dans les logiciels statistiques, mais maintenant elles sont disponibles et calculées automatiquement, selon les besoins de l'utilisateur. Au niveau des variables financières, elles doivent subir une transformation pour être alignées. Ces transformations peuvent être sélectionnées très facilement par l'utilisateur, il lui suffit de choisir les variables puis d'appuyer sur les flèches à l'écran qui lui permettent de les transférer dans une des cases. La case R1, R2, R3... permet de placer les variables financières sur une seule ligne en utilisant l'année de référence choisie (dans l'exemple, 2000) pour le nombre de périodes financières désirées. L'autre case, celle avec NOW, 1AN, 2AN... permet de créer des variables sur une seule ligne en partant de la plus récente année financière de chaque entreprise. Ces deux transformations de variables financières était très longue à compléter avec SAS, mais avec la structure interne de l'entrepôt de données, quelques secondes suffisent à préparer un jeu de données, même très grand.

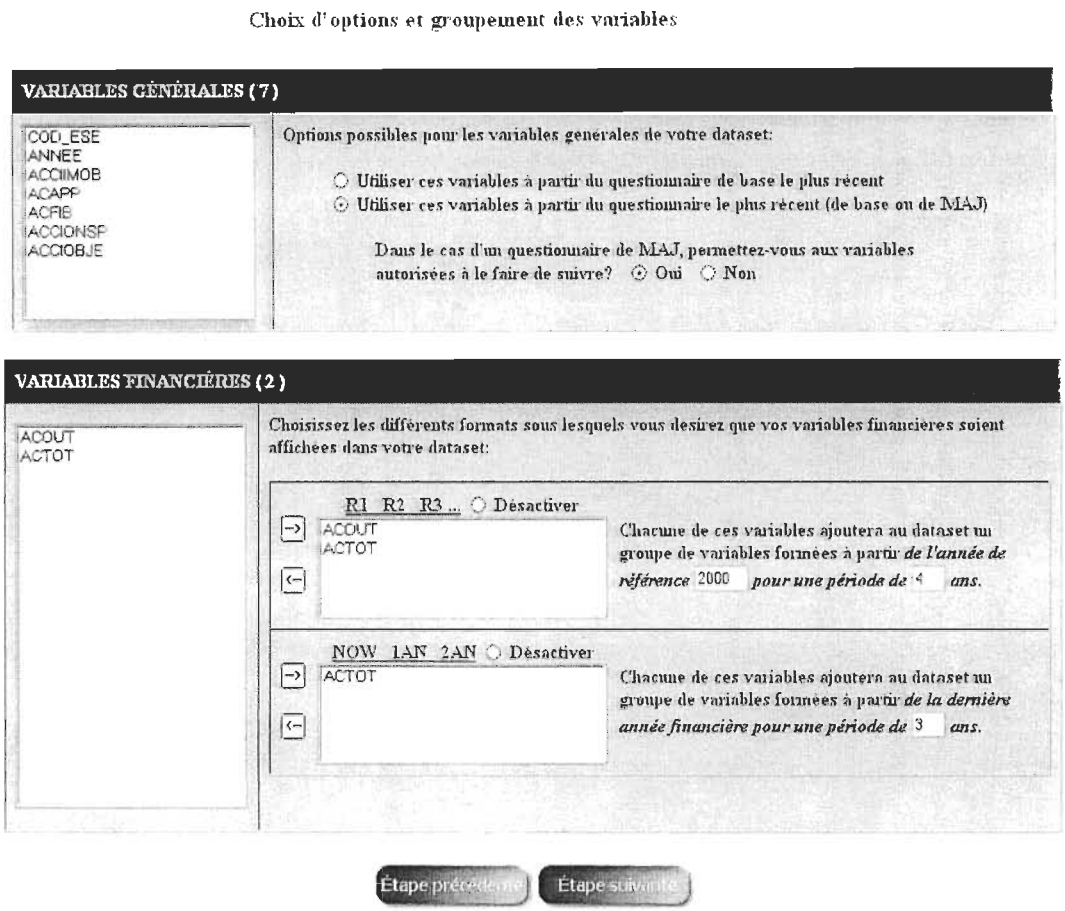


Figure 3 Étape 2 du *Dataset Maker*, déterminer les paramètres pour accéder aux données dans l'entrepôt en fonction des variables

La troisième étape (voir figure 4) permet de choisir de nombreuses options qui servent à créer le fichier du jeu de données et à le stocker dans la base de données. Certaines options ont aussi un rôle à jouer dans la sélection des données qui entrent dans la composition du jeu de données. Il faut choisir quel type de fichier sera créé parmi les choix suivant : un fichier de données SAS, un fichier XML qui peut être importé dans SAS, un fichier Excel et un fichier XML pouvant être affiché dans un navigateur. L'application exige un nom pour le jeu de données, ce nom est utilisé pour créer les fichiers de données et aussi pour stocker les fichiers dans la base de données, afin de pouvoir les indexer. Une autre information sert à indexer le fichier dans la base de données, c'est l'utilité du jeu de données qui est une sorte de description. Les autres informations sont des critères de sélection qui déterminent quelles données seront utilisées pour chaque variable. Puisque le *Dataset Maker* fonctionne à partir du magasin historique, il faut toujours donner une date pour générer les données. Normalement, c'est la date courante qui est utilisée. Il est aussi très important de spécifier la devise dans laquelle les données monétaires doivent être sélectionnées puisque l'entrepôt en supporte plusieurs. Il est possible de modifier l'année de référence sélectionnée à l'étape 2. Finalement, un champ permet d'ajouter plusieurs critères de sélection à la main, ces critères utilisent la même

syntaxe qu'une clause WHERE dans une requête SQL. Il est possible d'utiliser toutes les variables générales de l'entrepôt, y compris celles qui n'ont pas été sélectionnées pour le jeu de données. Les variables financières ne peuvent pas encore être utilisées, mais elles pourront être supportées dans le futur au besoin.

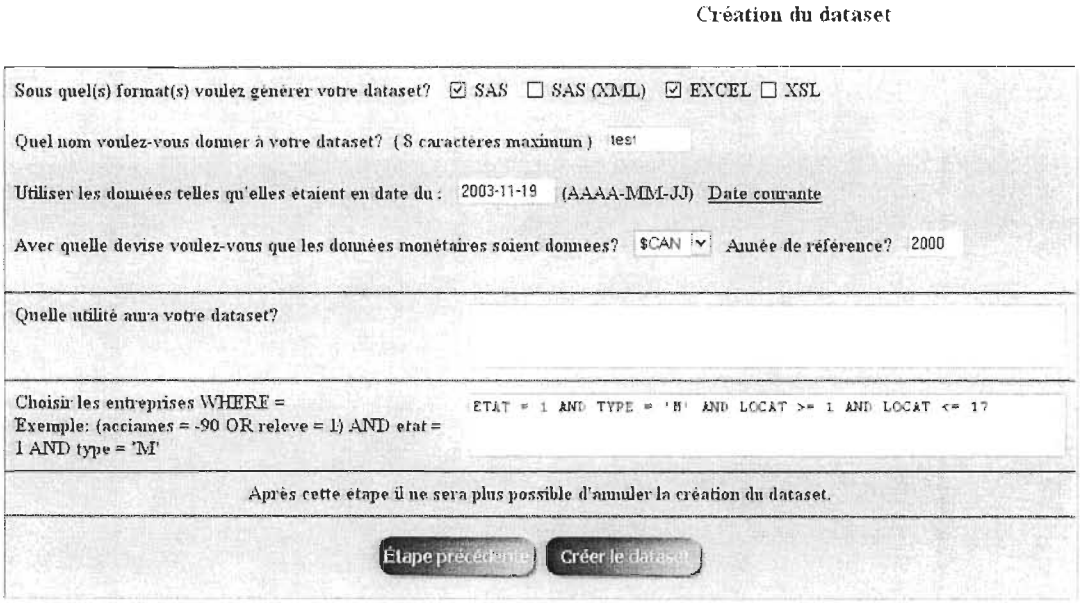


Figure 4 Étape 3 du Dataset Maker, sélectionner les options de création fichier de données à télécharger

La dernière étape de création d'un jeu de données (voir figure 5) consiste essentiellement en une confirmation de création du fichier, puis au téléchargement de ce dernier. Si une erreur survient, un message est affiché et des correctifs peuvent alors être apportés par l'utilisateur.

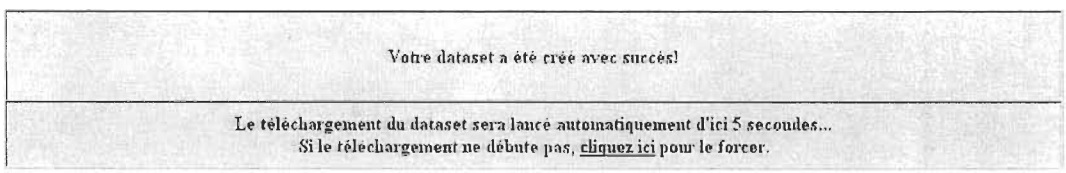


Figure 5 Etape 4 du Dataset Maker, l'application confirme que le jeu de données est prêt et démarre son téléchargement

4. Code utilisé

L'application *Dataset Maker* se sépare en deux composantes : la partie Web avec les fichiers JSP, et la partie Java avec les classes et librairies. La partie Web se situe dans le répertoire */datasetMaker* de l'application Web de l'entrepôt. Les différents fichiers JSP représentent les étapes décrites dans les sections 2 et 3 de ce document.

Un modèle général est fait dans le fichier *dataset_maker.jsp*, et le fichier de l'étape à laquelle l'utilisateur est rendu est importé pour afficher les bons choix. Il y a aussi une page qui affiche un menu pour choisir entre la création d'un nouveau jeu de données, ou le téléchargement d'un jeu de données existant (qui permet aussi de recréer un jeu de données avec des données à jour, ou de nouvelles variables).

La majorité du travail est cependant effectuée dans les classes en Java, les fichiers JSP servant principalement à l'affichage des pages Web. Elles utilisent des beans Java, comme ceux du package *larepe.entrepot.dataset* pour générer les jeux de données. La classe *DatasetMaker* contient tous les outils pour créer un nouveau jeu de données. Les différentes classes de pilotes se trouvent aussi dans ce package. La classe abstraite *Dataset* permet de créer un nouveau pilote (voir section 5).

Lorsque des variables sont sélectionnées et que les paramètres de création d'un jeu de données sont saisis, toutes ces informations sont conservés dans la classe *DatasetMaker*. Au moment où l'utilisateur confirme qu'il désire bien créer un jeu de données avec ces paramètres (étape 3), une requête SQL est générée dans la classe *DatasetMaker*. Plusieurs méthodes *creerRequete*[...] sont appelées, selon les paramètres, pour créer une très grande requête SQL. Cette requête est ensuite transmise au serveur Oracle pour qu'il retourne un *ResultSet* avec toutes les variables et données déjà alignées de la bonne façon. C'est Oracle qui fait tout le travail, il ne reste qu'à transposer ce *ResultSet* à l'aide des pilotes pour créer les fichiers de jeux de données que les utilisateurs peuvent importer dans leur application préférée.

Les fichiers sont stockés sous forme de BLOB dans l'entrepôt (base de données Oracle). Afin de permettre à l'utilisateur de télécharger ce fichier, un servlet est nécessaire. Ce servlet est downloadDataset du package larepe.entrepot.dataset, et il doit être configuré correctement dans le fichier web.xml pour pouvoir servir. L'utilisateur peut alors être dirigé vers un lien (ce lien dépend de web.xml), comme par exemple /datasetMaker/downloadDataset/mondataset.zip. Un paramètre est nécessaire, c'est le numéro du jeu de données (*noDataset*) dans la base de données. On pourrait par exemple avoir le lien */datasetMaker/downloadDataset/mondataset.zip?noDataset=10*. Ainsi, l'utilisateur télécharge le fichier mondataset.zip (qui peut être un nom arbitraire, selon la configuration de web.xml) et le contenu provient du BLOB identifié par le numéro 10 dans la base de données Oracle.

5. Principe des pilotes

Un des buts du pilote est de créer un ou des fichiers dans un *ZipOutputStream* (pour créer un fichier ZIP) pour qu'une application spécifique puisse utiliser les données provenant de l'entrepôt. Une classe pilote doit hériter de la classe abstraite *larepe.entrepot.dataset.Dataset* pour être supportée par le *Dataset Maker*. Le rôle du pilote est de récupérer un *ResultSet* créé par la classe *DatasetMaker* et de créer un fichier à partir de ce *ResultSet*, dans un fichier Zip, tout en utilisant correctement les paramètres choisis par l'utilisateur pour la création de ce jeu de données. Il peut

aussi être nécessaire au pilote d'accéder aux métadonnées pour créer de la documentation sur les données pour l'utilisateur.

À la base, le pilote effectue plusieurs boucles dans le *ResultSet* pour créer son propre fichier de données. Par exemple, pour un fichier de données SAS, il faut créer un fichier .DAT qui pourra être lu et importé par le logiciel SAS. Ce fichier est un simple fichier texte où la première ligne contient le nom de variables, et les lignes subséquentes les enregistrements avec les données. Cependant, une fois ce fichier créé, il faut aussi préparer un fichier .SAS qui permettra à l'utilisateur de facilement importer les données. Ce fichier contient quelques informations sur le jeu de données (qui l'a fait, quand, etc.). Mais ce fichier contient aussi les formats des données (numérique, nom de la classe, ...) et la description sommaire de chaque variable. Ces informations sont très utiles une fois rendu dans le logiciel SAS, elles sont affichées à plusieurs endroits. C'est de cette façon que les métadonnées de l'entrepôt peuvent être récupérées dans les autres logiciels. Les pilotes XSL et Excel ne recopient pas toutes ces informations, quoiqu'il pourrait être utile de le faire plus tard.

Comme on le voit, un pilote doit remplir un mandat, celui de créer un fichier de données à partir d'un *ResultSet*, et insérer ce fichier dans un *ZipOutputStream*. Il peut néanmoins créer d'autres informations, dans le même fichier ou en utilisant des fichiers supplémentaires, pour transmettre de l'information sur le jeu de données et les variables. Cette documentation peut devenir très utile. Et pour ajouter de nouveaux pilotes, il suffit de trouver une librairie appropriée, ou de fabriquer le format directement afin de permettre au logiciel visé d'importer facilement les données.

6. Problèmes rencontrés et solutions adoptées

Un des problèmes rencontrés est arrivé dans SAS, en essayant d'importer les fichiers à partir d'une commande INPUT directement à partir d'un fichier .SAS. Toutes les données avaient été importées dans le fichier SAS. La première constatation, c'est que SAS limite ses lignes à 256 caractères, ce qui donnait un résultat plutôt étrange. Même si les lignes avaient plus de 256 caractères, les données étaient importées. À tous les 256 caractères, sur cette ligne, le caractère était ignoré, ce qui donnait souvent de très mauvaises données importées, et les résultats étaient très difficiles à diagnostiquer. Un autre problème avec cette façon de faire est que le *buffer* de l'éditeur de texte de SAS ne semble pas être protégé, ce qui occasionne des *buffer overflow* avec de très gros jeux de données. Ces *buffer overflow* faisaient généralement geler mon ordinateur, d'autres fois il se mettait en veille et ne voulait plus se réactiver, et d'autres fois il redémarrait tout bonnement. C'est pourquoi on a choisi de créer le jeu de données à l'intérieur d'un fichier .DAT, type de fichier texte que SAS supporte et il est alors possible d'importer les données sans tous les problèmes décrits précédemment.

La performance lors de l'exécution du Dataset Maker doit rester constante, et se maintenir à un bon niveau. Il n'est pas acceptable d'attendre 5 minutes pour la

création d'un jeu de données. C'est pourquoi les requêtes SQL qui font le gros du travail se doivent d'être bien optimisées. Plusieurs requêtes imbriquées sont utilisées pour forcer Oracle à faire des MERGE SORT, ce qui est beaucoup plus rapide. Par exemple, pour l'alignement des données financières, une requête trouve l'information pour R1, une autre pour R2, etc. Ces requêtes sont générées dans la clause FROM d'une super-requête, qui doit alors aligner les données correctement. Cette super-requête prenait constamment 56 secondes pour s'exécuter, avant qu'on décide de mettre un ORDER BY dans toutes les sous-requêtes. De cette façon, l'optimiseur d'Oracle est capable de faire une jointure avec un MERGE SORT, ce qui est de loin plus rapide que de mettre toutes les requêtes dans la même table de tri temporaire. Chaque table arrive déjà triée (pas très long), et le résultat est obtenu généralement en 4 à 5 secondes.

Dans le cas du pilote Excel, il n'a pas été possible de supporter plus de 255 colonnes. Un algorithme simple qui répartit les variables sur plusieurs feuilles de calcul a été créé. Cependant, une des limitations de POI (librairie Java utilisée pour créer des fichiers Excel) est que l'ensemble du fichier doit tenir en mémoire. Il s'est avéré que pour plus de 300 variables, ce fichier n'arrivait pas à tenir dans les limites de la mémoire du serveur. C'est pourquoi il n'est pas possible de créer un jeu de données dans Excel avec plus de 253 variables (excluant COD_ESE et ANNEE qui sont toujours inclues). Cependant, si une nouvelle interface ou librairie Excel permettait de stocker les fichiers sous une forme de *stream* sans avoir à tout stocker en mémoire, il serait alors possible d'utiliser plusieurs feuilles de calcul pour mettre toutes les variables.

ANNEXE G

Analyse avec ContourCube

ContourCOMPONENTS

How to Analyze Data

with ContourCube OLAP Component

What's OLAP?

OLAP is a special method of analyzing data and building reports. Its purpose is to display data in a dynamical table that automatically summarizes it in different slices and allows users to interactively manage calculations and the form of a report. The instruments for managing reports are elements of the table itself. By dragging columns and rows, users can change the view of a report and data groupings on their own, and the OLAP engine instantly calculates new totals.

Users can drill down into data and summarize it, filter and sort the table. As a result, they can create dozens of various reports from a single table without involving programmers. By examining data at different angles, users better understand it and take effective decisions.

Instead of tedious process of developing more and more new fixed reports developers are provided with the ability to create several interactive OLAP reports, the flexibility and openness of which allow them to fully solve user queries.

ContourCube is an OLAP component that can be embedded into various systems to display their data from in the form of an OLAP table. This enhances analytical and reporting capabilities, facilitates usage, and contributes to user satisfaction.

ContourCube can access data from any flat array - relational and local databases, and arbitrary datasets.

Creating Cube from Flat Data

As mentioned above, ContourCube can use a wide range of data sources: any relational and local databases as well as arbitrary datasets. All data, pooled into the OLAP system, is subdivided into dimensions and facts (or measures), where dimensions are descriptions and facts are figures related to this description. Aggregation of fact is performed in a slice of dimensions according to a given algorithm. (Sum, average, etc.) By manipulating dimensions, users can specify procedures of calculating totals by facts. A set of facts and dimensions is called a cube.

The developer creates of a cube by subdividing source fields from a database into dimensions and facts. Only field of numeric data type can be facts, while there are no type limitations for dimensions, though, normally, field of string or date type are considered as a dimension.

It is necessary to define an area to contain a dimension and a mutual order of dimensions in the area that influences the order of the fact aggregation. There are four areas: columns area (horizontal), rows area (vertical), inactive (filters) and non-visual dimensions area.

ContourCube provides 'modifiers' - embedded field transformation functions - that can generate new dimensions (time series) for dimensions of date type, thus enabling user to see data aggregated by time. The 'modifiers' of dimensions can only transform a date into a given time period: Year, Quarter, etc.

It is necessary to define an aggregation algorithm for facts. By applying different aggregation algorithms, several basic facts can be created from a single database field, namely facts which values are fed from the data source.

Any fact can be a foundation for a calculated fact. To make it possible ContourCube provides several modifiers, including statistical algorithms, such as average, dispersion, etc.

Besides, calculated facts can be created with formulas. By using other facts, built-in and user functions, constants and operators, the developer can define the algorithm for calculating a new fact.

Calculated facts can be hierarchical, specifically every calculated fact can be foundation for one or several calculated facts.

Modified facts as well as facts calculated by formulas can be added 'on the fly'. That is why users can perform calculations of extra facts during analysis.

Finally, flat data, loaded into the cube, becomes a multidimensional cube.

ContourGrid Visual Layout

The multidimensional cube is displayed in the form of interactive table. Some elements of the table are active, in other words users can manipulate them to change groping, filter data, etc.

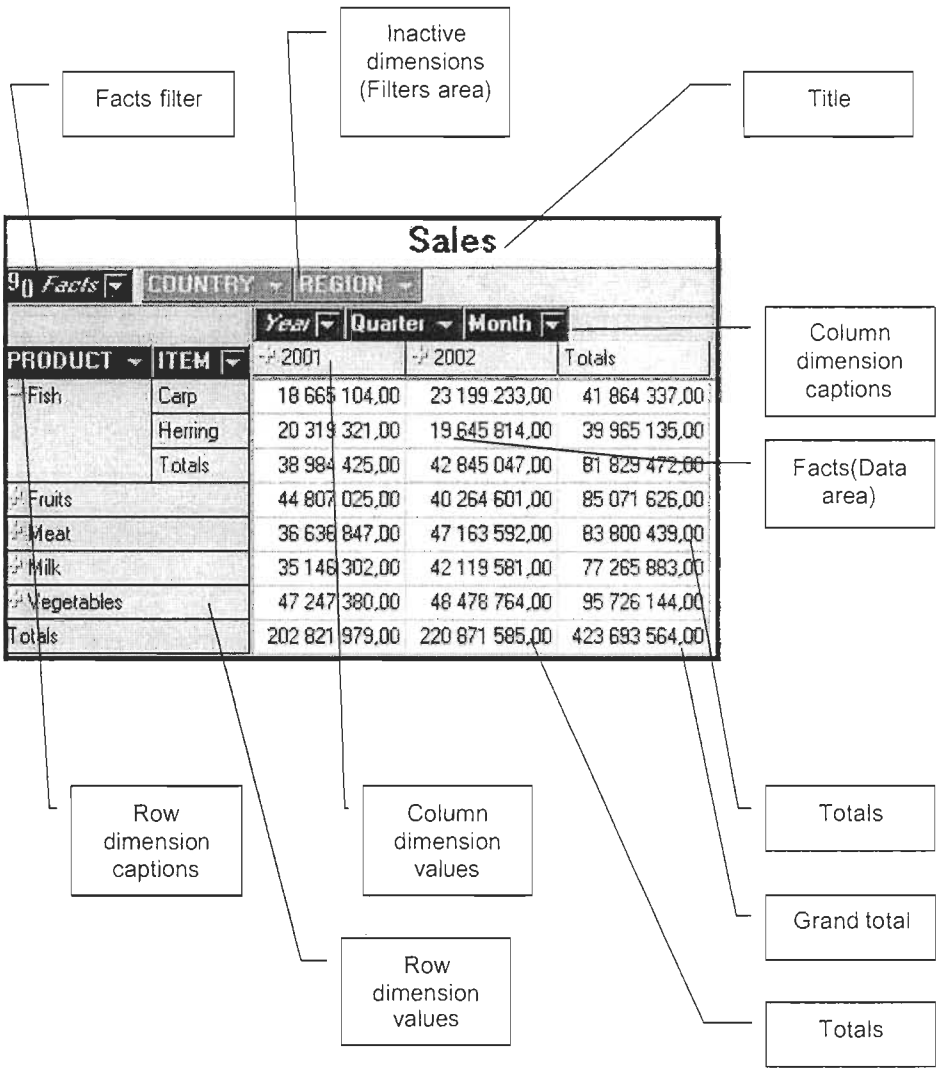


Figure 1. ContourGrid Visual Layout

ContourGrid consists of several areas; a description of each area follows:

1. **Inactive Dimensions Area**. Dimensions put in this area can be used to filter data. They can be also put in the area at design time. In this way, users can drag columns and rows to the area using their mouse. If the users drag a dimension from the area to the columns or rows area, data becomes more detailed. If the users drag an active dimension to the area, the table shows aggregated totals.
2. **Column Dimensions Area (Horizontal Dimensions)**. Dimensions in this area are active, with facts for every value of a dimension being displayed in the form of columns. Totals that OLAP engine calculates upon every value of a dimension are displayed in the bottom cell of a column.
3. **Row Dimensions Area (Vertical Dimensions)**. Dimensions in this area are active, with facts for every value of a dimension being displayed in the form of rows. Totals that OLAP engine calculates upon every value of a dimension are displayed in the right column of a row.
4. **Facts Area (Data)** shows values of facts for every intersection of a column and a row. If dimensions are fully expanded, cells display non-aggregated facts, otherwise – totals.
5. **Facts Choice Area (Facts Filter)**. The drop-down list displays all fact within in the table. (Note that there may be more facts in the cube!) This area allows you to swap facts or make unneeded facts inactive.
6. **Dimension Captions** are active cells that can be dragged. A dimension can be dragged to any other area or change place, with new totals being automatically calculated. A pull-down menu below Dimension Captions allows users to set filter conditions for the table by making values of dimensions active or inactive. To make all values active, press Ctrl+; to make them inactive, press Ctrl-. A small toolbar at the bottom of the menu allows users to make all values of a dimension active, inactive or invert their state.
7. **Dimensions Values** are also active cells. By pressing the + or - keys users can perform drill down or drill up, thus gaining more detailed or more summarized reports. To expand all values of dimensions, users can press Ctrl+; to hide the values - Ctrl-.
8. **Totals** are rows or columns in the data area that display a sum or a total, calculated in another way, at the intersection of values of dimensions along a column and row. Totals can be made inactive for a given dimension.

Drill down\Drill up

Initially all or some values of dimensions may be collapsed. In this case, the table displays totals by the highest levels in the hierarchy of dimensions. To see more detailed data, drill down can be performed by clicking the '+' sign next to the cell with a value of a dimension. Then, you will see values of the next dimension to the right of the selected value, while the totals area displays corresponding numeral values. The total for the expanded dimension will be displayed below the values of its child dimension.

To hide unneeded details and obtain a summarized report a value of a dimension can be collapsed by clicking the '-' sign.

This technology allows you to explore data by drilling drown from summarized values to their details along a desired direction. As you can see in Figure 1, all values of dimension *Product* are collapsed with drill down performed along value *Fish*. Note that sales figures for all products – except Fish - are aggregated; as for Fish, they are subdivided into species.

Pivoting

ContourCube also enable the table to be pivoted. In other words any dimension from the rows area can be dragged into the columns area. This involves change of the report layout and alters aggregation. ContourCube automatically calculates new Totals and Subtotals.

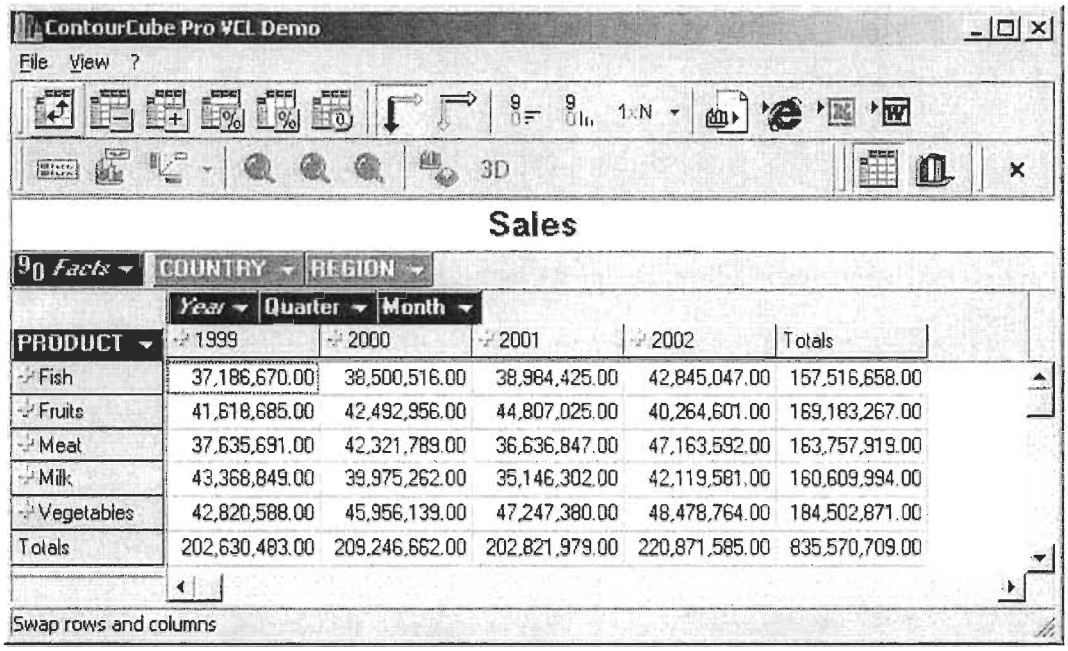


Figure 1 Before swapping rows and columns

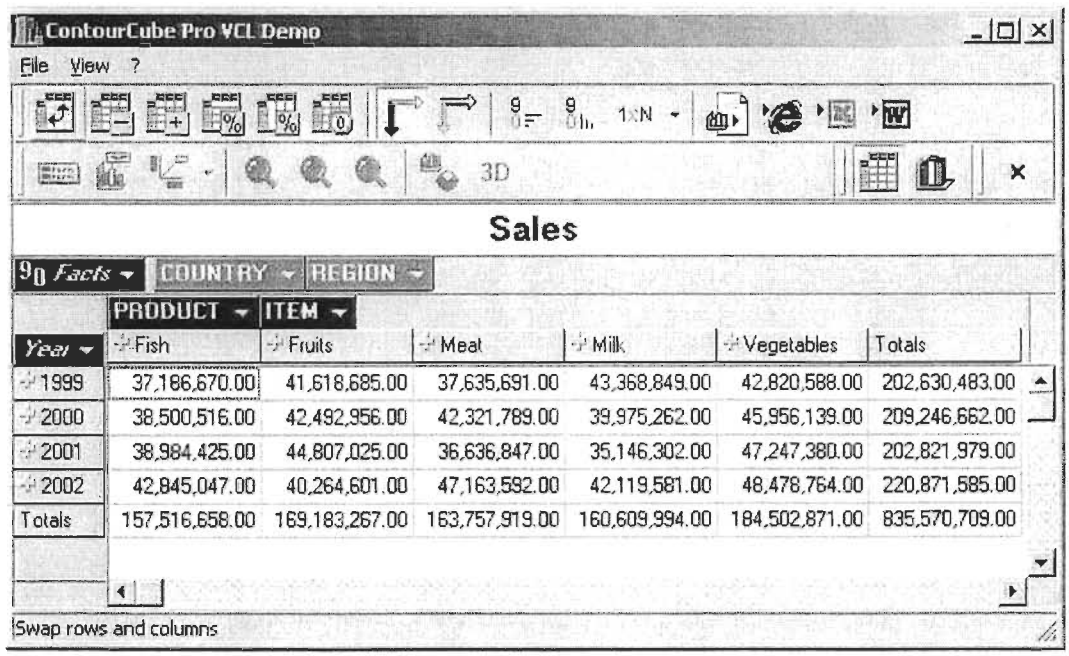


Figure 2 After swapping rows and columns

Filtering

Every dimension has a built-in popup menu that enables users to check/uncheck unneeded values of dimensions. Then, the table shows data only for checked values of dimensions.

Filtering can be performed on both active and inactive dimensions. To quickly make all values of dimensions inactive, press Ctrl-; to make them active, press Ctrl+. A small toolbar below the popup menu also allows you to make all values of dimensions active/inactive or invert your choice. For example, by setting filter to dimension *Country* to value *USA* users can obtain a report on sales in the USA.

Figure 3 Filtering

Sorting

Data in the table are always sorted. There are two options of sorting the table: by dimensions and by facts. The default option is sorting by dimensions. To specify the sorting order (ascending/descending) by every dimension users can use a local menu that appears on their mouse right clicking. The sorting order is independently specified for every dimension.

The Sorting by Facts option is applied either for columns or for rows. To sort rows by facts by rows select a fact and a column. All rows are recursively sorted by the fact values in the selected column. The sorting order for every column is the one that is specified for the corresponding dimension.

To sort columns by facts select a fact and a row. All columns are recursively sorted by the fact values in the selected row. The sorting order for every column will is the one that is specified for the corresponding dimension.

Normally, the application developer makes two 'Sorting Columns by Facts' and 'Sorting Rows by Facts' buttons and defines a fact and a row or a column by the current cursor position in the data area. That is why users should move their cursor to a desired cell and press a desired bottom to select the sorting option.

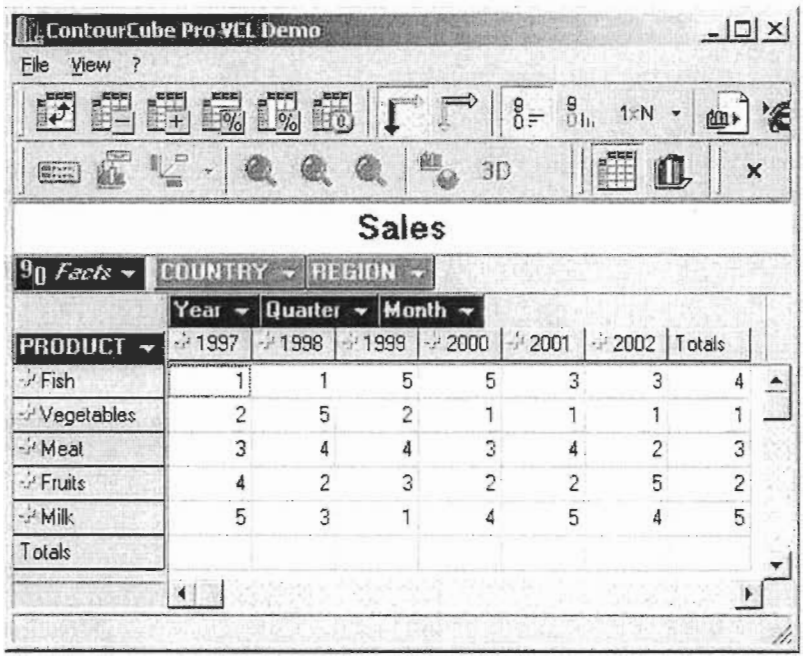


Figure 4 Products sorted by sales in 1997

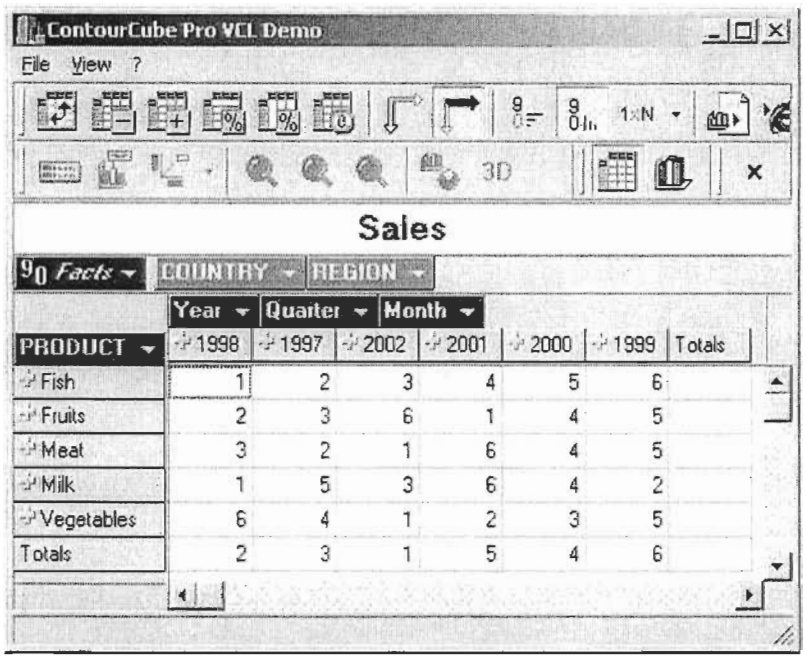


Figure 5 Years sorted by sales Fish

Export & Print

Data and its layout can be exported to HTML and loaded to Excel, Word or a browser in the same way as ContourCube displays data. Then, users can print the report.

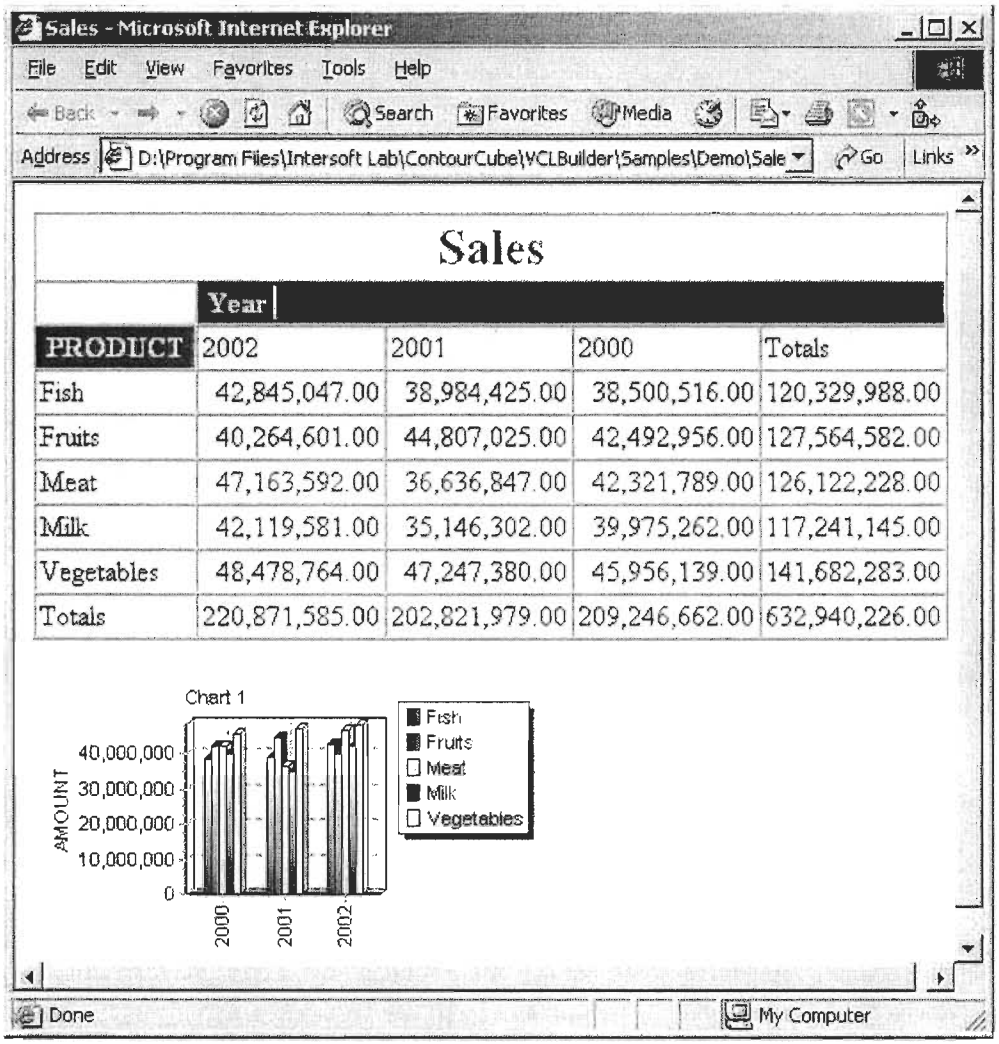


Figure 6 Report exported to html

The developer can also apply any other report component or another print method to obtain data by using ContourCube's special View object that provides data same way as ContourCube displays data.

ContourCube home site: http://www.ContourComponents.com

ContourCube (c) 2002 Intersoft Lab

COMPONENTS

Creating web-based OLAP solution

This article is on how to create affordable and powerful OLAP solution using ContourCube ActiveX OLAP software component.

Overview

OLAP technology provides users a highly interactive data analysis and reporting facilities. Users can pivot, filter, drill down and drill up data and generate numbers of views with simple mouse manipulations. Web based OLAP simplifies deployment of OLAP software, makes the solution accessible for as many people as it's needed and does not require any extra knowledge to work with. It is a best way to create public internet- and corporate intranet informational sites.

Web based OLAP can retrieve data from a database directly or from pre-built cubes - multidimensional databases. In case of using ContourCube software component, the cube files named "microcubes" are Internet-optimized. A microcube contains not only data but metadata too.

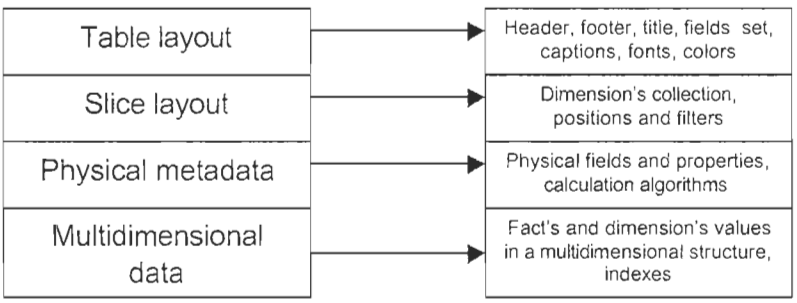


Figure 1 Microcube structure

Metadata contains property values needed for report layout definition include colors, fonts, captions and more. It also contains field calculation algorithms. So Contour microcube is a data container of an analytical application, like Excel workbook. User needs just any viewer created with ContourCube software component to display microcube contents and manipulate with number of reports.

Any application powered by ContourCube can load microcubes through http, file, and ftp protocols.

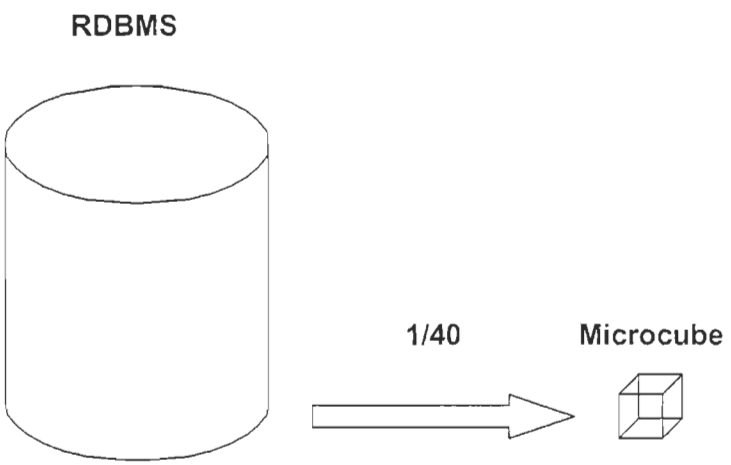


Figure 2 Data loaded from a relational database to Contour microcube will be compressed up to 40 times.

Microcube file contains data in compressed form, which helps decrease network traffic. Depending on data source structure it is possible that data is compressed from 10 up to 100 times.

Placing ContourCube ActiveX onto a web page

Place ContourCube ActiveX onto your web page. You can do it in a HTML editor like MS Frontpage, by simple inserting the control from ActiveX controls list. Or you can add this code manually:

```
<OBJECT CLASSID="clsid:CCA2C672-33FD-11D5-8D72-005004532BDF"
id="ContourCubeX1" width="746" height="386"
CODEBASE="ccubex.cab#VERSION=-1,-1,-1,-1">
```

Class ID identifies the version of the ContourCube ActiveX component that you want to insert into your web page. You can define any id value for further object reference. The next two properties **width** and **height** define size of the object on your page.

CCubeX.cab is the ContourCube ActiveX file that can be found in installation directory of the product. You should upload this file to your web server and specify path to this file in the CODEBASE attribute.

Automatic update of the ActiveX component

When a user first time visit your web OLAP page, the system will check if the component is installed on his computer and ask if the user wants to download and install it. In the following line you can specify the component version required by your web page:

```
CODEBASE="/ccubex.cab#VERSION=1,3,0,11">
```

In this case the new version of the component will be downloaded on the client computer only if it is newer than existing version or if the control isn't installed on the workstation at all. You should edit your web page each time you upload a new version of the component to your web site.

The following line shows the syntax that allows new version upload without any edition of your page.

```
CODEBASE="/ccubex.cab#VERSION=-1,-1,-1,-1">
```

This line means that new component version will be downloaded if it is not installed on the client computer or if the component version is newer than the client's one.

Supplying the component license

As a legal user of the ContourCube component, you should provide registering information on the product. This information will be contained in a LPK-file (License Package File) that can be created using Microsoft tools. Place this code in your page:

```
<OBJECT CLASSID="clsid:5220cb21-c88d-11cf-b347-00aa00a28331">
  <PARAM NAME="LPKPath" VALUE="/cx_license.lpk">
</OBJECT>
```

cx_license.lpk is your license file that contain your serial number information and certifies your right to use the ActiveX component.

To create a LPK-file, follow these instructions:

- Launch Microsoft License Package Tool LPK_Tool.exe which is supplied with the MS Internet Client SDK. The application is actually simple to use. When you start the application, it displays a combo box that lists all of the ActiveX Controls currently running on your computer.

- The next step requires that you choose the controls that you want to display on a given HTML page. You choose a control by highlighting it in the combo box and then clicking the **Add** button.

- Once you've selected all the controls that you want to display on a given page, you can create and save the actual .lpk file by clicking the **Save & Exit** button. (This causes the application to display the File Save dialog box, which allows you to specify the path and file name.)

Microcube data delivery

In most cases the best way to deliver data to web users is to place pre-built microcubes on the Internet server and to use ContourCube software component on the client side. This is the MOLAP (Multidimensional OLAP) architecture. Because users load pre-built microcubes, this architecture is faster than a direct connection to a relational database. Also, this is the simplest way of data delivery because you don't need to maintain database connection. The following example shows how to connect your web page to a Microcube server:

```
<PARAM NAME="DataSourceType" value="2">
<PARAM NAME="ConnectionString"value =
"http://www.site.com/cubes/Sales.cube">
```

The first line tells ContourCube object that it will get data from a microcube. The second line specifies the URL of the Microcube file.

If you want to schedule microcube updating you can use Contour CubeMaker command line tool that provides batch creation of microcubes based on a template microcube and controlled from an XML script.

Direct connection to a database

In corporate Intranet solutions it is possible to provide to business-users online direct connection to a relational database. It allows user to view data interactively. This solution works in ROLAP architecture and use both client and server resources. Database server will select data and eventually perform data pre-grouping and aggregation if you include Group By option in your Select statement. In this case, calculations will be shared between server and client PC. OLAP engine on the client workststion will get pre-grouped and aggregated data and then perform additional calculation of subtotals and totals according to specified aggregation algorithms.

Add this code in your web page to connect to a MS SQL database through ADO.

```
<SCRIPT LANGUAGE="VBScript">

<!--

Const CONS1 = "Provider=SQLOLEDB.1; Data Source=MYSQLSERVER; Initial
Catalog=Sales;"
Const CONS1 = "Persist Security Info=True;User ID=sa; Password=123"
CONS = CONS1 & CONS2
```

Then build your SQL Select statement:

```
Const SQL1 = "SELECT Country, Product, Date, SUM(Amount)"
Const SQL3 = "FROM Sales"
Const SQL3 = "Group By Country, Product, Date"
SQL = SQL1 & SQL2 & SQL3
```

Now all is ready to create a cube:

```
With ContourCubeX1
  'Create Dimensions and Facts in ContourCube
  .Active = False
  .ClearFields
  .CubeTitle = "Sales"
  .AddDimension "Country", "Country", 0, 2
  .AddDimension "Product", "Product", 0, 2
  .AddDimension "Country", "Country ", 0, 2
  .AddDimension "Date", "Date", 2, -1 ' xda_outside = 2
 'Create date parts from the date field
  .AddDimModifier "Year", 3, "Date", 1, "Year", 1, 1
  .AddDimModifier "Quarter", 3, "Date", 2, "Quarter", 1, 2
  .AddDimModifier "Month", 3, "Date", 3, "Month", 1, 3
  'Create data fields -'facts'
  .AddFact "Quantity", "Quantity", 1, "Quantity"
  .AddFact "Amount", " Amount ", 1, " Amount "
 'perform SQL statement
  .SQL = SQL
  .DataSourceType = 0 'xcdt_ADO = 0
  .ConnectionString = CONS
  .Active = True
End With
-->
</script>
```

Setting additional component properties

You can define colors, fonts and positions of fields and set any other ContourCube properties. Note that all properties will be overwritten with values saved in the microcube file when it is loaded. For example, you can define these properties in OnCubeLoaded event handler or create a user menu for this purpose.

```
<script for = "ContourCubeX1" event = "OnCubeLoaded()" language =
"javascript">
   ContourCubeX1.Footer.Text = "text appeared bellow the table"
</script>
```

Adding dialogs

You can allow users to switch between microcubes by name or by complex parameters. The code below shows how to add two buttons that select one of two available microcubes:

```
<button onclick = "CubeClick('Sales.cube')">Sales</button>
<button onclick = "CubeClick('Balance.cube')">Balance</button>
```

The next code on Java script loads chosen microcube:

```
<script language = "javascript">
function CubeClick(Cube)
{
  ContourCubeX1.Active = false
  ContourCubeX1.ConnectionString = " www.site.com/cubes/" + Cube
  ContourCubeX1.Active = true
}
</script>
```

Creating toolbars

If you want to equip your user with tools for quick layout changing, printing reports and making sophisticated data analysis you can create a toolbar on your web page. To do this,

you only need to place appropriate buttons and link them to ContourCube properties and methods or your own procedures and functions. 217

```
<button onclick = "Swap()">Swap </button>
<button onclick = "PrintCube(0)">Print</button>
<button onclick = "PrintCube(-1)">PrintPreview</button>
<script language = "javascript">
function Swap ()
{
  ContourCubeX1. Transposed = Not ContourCubeX1. Transposed
}
function PrintCube (preview)
{
  ContourCubeX1.PrintCube(preview,1)
}
</script>
```

Resume

ContourCube ActiveX provides a simple-to-create and high-performance web based OLAP solution. This solution allows to effectively use all PC resources and does not require too expensive servers. You can create a basic OLAP solution in a couple of hours and on the other hand, you are free in creating sophisticated advanced solutions.

The most considerable feature of such solution is its affordability: its end users are not charged for using our OLAP component.

ContourCube home site: http://www.ContourComponents.com

ANNEXE H

Tutoriel de *SAS Enterprise Miner*

SAS Tutorial: Enterprise Miner

Fine Dining Marketing Campaign.

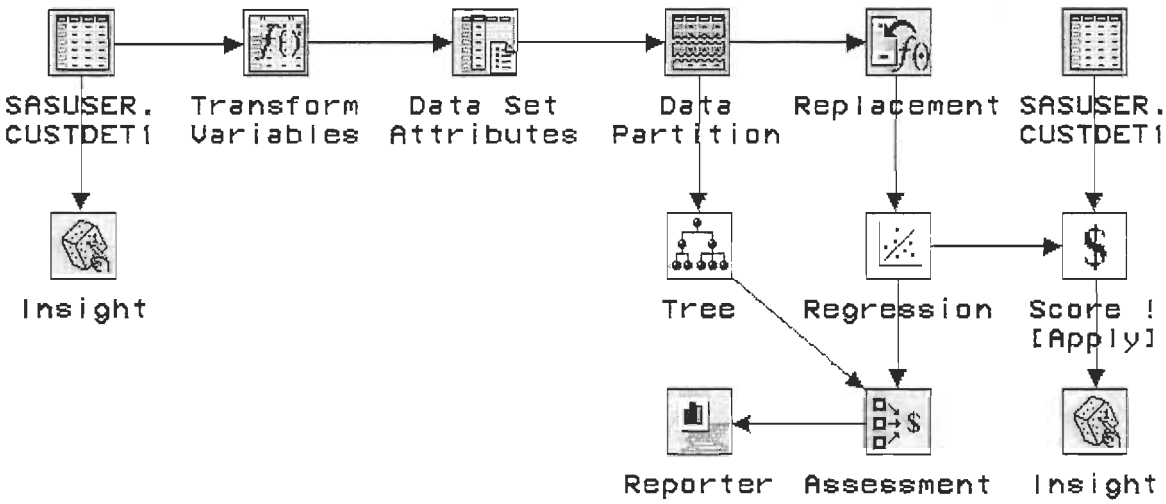
Suppose that you work for a mail order enterprise that sends out a catalog of furnishings and housewares each month. As part of an upcoming sales campaign, you want to distribute a special catalog focusing on fine dining which contains kitchenware, dishes, and flatware. Because it is too expensive to mail the catalog to all of your customers, you decide to target only those most likely to buy at least one item. Towards this end, you decide to build a targeting model from historical data and deploy the model to produce a new mailing list.

You have an extensive record of customer purchases. The data contains information on precious purchases including variables indicating whether a customer bought from a specific category such as kitchenware, dishes, or flatware in the past two years. The data set contains 49 variables with the following labels:

Purchase
Dollars Spent
Yearly Income
Home Value
Order Frequency
Recency
Married
Name Prefix
Age
Sex
Telemarket Ind.
Rents Apartment
Occupied <1 Year
Domestic Product
Apparel Purchase
Leisure Product
Luxury Items
Kitchen Product
Dishes Purchase
Flatware Purchase
Total Dining (kitch+dish+flat)
Promo: 1-7 Months
Promo: 8-13 Months
$ Value per Mailing
Country Code
Total Returns
Mens Apparel
Home Furniture
Lamps Purchase
Linens Purchase
Blankets Purchase
Towels Purchase
Outdoor Product
Coats Purchase
Ladies Coats
Ladies Apparel
His/Her Apparel
Jewelry Purchase
Date 1st Order
Telemarket Order
Account Number
State Code
Race
Heating Type
Number of Cars
Number of Kids
Travel Time
Education Level
Job Category

Enterprise Miner

Based on SAS software, Enterprise Miner combines the data mining process with graphical ease of use. It delivers a broad range of predictive and descriptive models that you can apply, test, and compare to determine the best fit for the data. Most of this is done by manipulating icons: you connect nodes in a graphical workspace, adjust settings, and run the workflow. Here's an example of the workflow:



Nodes are a key concept in Enterprise Miner; most of the time you interact with the program by dragging and dropping, right-clicking, or double-clicking the node that corresponds to a particular task.

1. Invoke Enterprise Miner
After you start the SAS System, from the SAS menu bar, you can select **Solutions** → **Analysis** → **Enterprise Miner**.

Creating a new project

2. To create a new project, select File → New → Project. When the **Create new project** window appears, enter Dining List in the **Name** field. In the location field, select browse and create a new directory **DiningList** for the project.
Click the **Create** button.
You have now created a directory for your project. EM has created three additional subdirectories: EMDATA, EMPROJ and REPORTS.

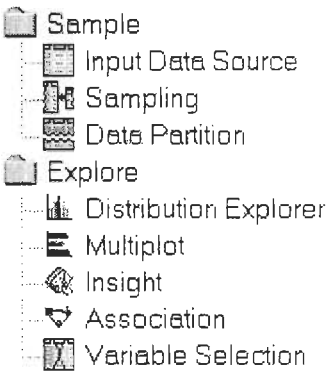
3. The **Dining List** project appears in the left window pane. Below it appears the default name of the workflow, **Untitled**. Select **Untitled** and enter the new name Propensity.

4. Go to the course web site and download the tutorial file: DataMining2003.sas7bdat and save it in the emproj subdirectory.

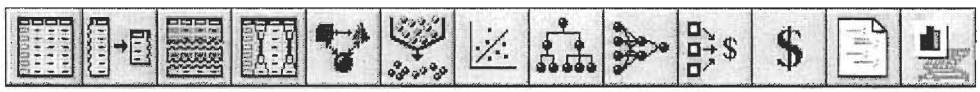
Apply workspace nodes

You must first specify the source of the data for the mailing list project. To define a data source, you drag and drop a node onto the workspace.

View the nodes by clicking the **Tools** tab at the bottom of the left window pane. A grouped list of icons and labels appears; a small section is shown below:



You can also select nodes from the Enterprise Miner menu bar at the top of the window. The bar shows a larger version of some of the icons in the node list (To read the name of a node in the menu bar, briefly hold the mouse pointer over its icon. The name will appear in a tooltip box).



Define a data source

1. Left-click the node that is labeled **Input Data Source** and drag it to the right window pane. Release the node to drop it into the workspace.

2. Double-click the new node to specify the source data. The **Input Data Source** window appears, with the **Data** tab in the foreground.

3. Click the **Select** button. The **SAS Data Set** window appears. Select the emproj directory.

4. Select **DataMining2003** from the list of tables. Click **OK** to close the window.
5. Close the **Input Data Source** window. A confirmation box appears.

6. Click **Yes** to save your changes.

Data Exploration

Apply the Insight node

Enterprise Miner includes an **Insight** node for exploring your project's data.. It allows you to explore the situation of missing values, outliers, or skewed distributions can.

1. Drag and drop the **Insight** node onto the workspace. Place the new node under the **Input Data Source** node
2. Connect the **Input Data Source** node to the **Insight** node:
    1. Hold the mouse pointer at the edge of the **Input Data Source** node until it becomes a pair of crosshairs.
    2. Left-click and quickly drag to the **Insight** node.
    3. An arrow between the nodes appears.
3. Double-click the **Insight** node. The **Insight Settings** window appears:
4. Notice that the data set name is not **DataMining2003** but a new name that has been provided by Enterprise Miner. The **Description** field displays the original data set name.
5. Because of the large data stores you might work with, the Insight node defaults to examining a sample size of 2,000 records. In the case of the **DataMining2003** data set, which has 1,966 records, click the **Entire data set** button. The **Sample size** field will change to **Data set size** to reflect the change.
6. Close the **Insight Settings** window. A confirmation box appears.
7. Click **Yes** to save your changes.

View Insight node output

8. Right-click the **Insight** node and select **Run**. A green border appears around the node as it reads your data, and then a confirmation box pops up.

9. Click **Yes** to view results. A tabular view of your data appears:
10. With the table view open, select **Analyze → Distribution** from the SAS System menu bar.

11. A window for selecting distribution variables appears. Select income. Click the **Y** button and then **OK**. A window with the distribution of the income variable appears.

Create a target – Transforming variables

You are trying to target the buyers of dining wares, but the variable **DINING** presents a problem. Because it contains the sum of **KITCHEN**, **DISHES**, and **FLATWARE**, its values range from `0` to `28`. But you are looking for the buyers of **any** dining wares, represented by all values greater than `0`. What you need, therefore, is a binary version of **DINING**, where the values greater than `0` are collapsed to `1`.

You create variables in Enterprise Miner with the **Transform Variables** node

1. Drag and drop the **Transform Variables** node onto the workspace.

2. Connect the **Input Data Source** node to the **Transform Variables** node.
3. Double-click the **Transform Variables** node. The **Transform Variables** window appears.
4. Click the **Create Variable** icon on the workspace menu bar. The **Create Variable** window appears.
5. Enter `DINEBIN` in the **Name** field.

6. Enter `DINING No/Yes` in the **Label** field.

7. Click **Define**. The **Customize** window appears.

8. Enter `dining>0` in the **DINEBIN(N)=** formula field at the bottom of the window.

9. Click **OK**. The **Create Variable** reappears with `dining>0` displayed in the **Formula** field.

10. Click **OK**. The new variable **DINEBIN** appears in the **Transform Variables** window.

11. Close the **Transform Variables** window. A confirmation box appears.

12. Select **Yes** to save your changes

Modifying Attributes

You now need to identify it as the model's target. This is done with the **Data Set Attributes** node

1. Drag and drop the **Data Set Attributes** node onto the workspace to the right of the input data source node. Connect the **Transform Variable** node to the **Data Set Attributes** node.

2. Double-click the **Data Set Attributes** node. The **Data Set Attributes** window appears.

3. Click the **Variables** tab.

   Scroll down the list of variables until **DINEBIN** appears. Notice the grayed-out column **Model Role** and the white column **New Model Role**. Grayed-out columns reflect the original data set attributes and they cannot be edited.

   Role refers to the use of each variable. Most variables are treated as input variables in an attempt to predict the target. If you scroll down the list of variables, you will see that Enterprise Miner considers certain variables unsuitable as inputs (e.g., dates, or variables with a single value). Such variables are given the role **rejected.**

4. Right-click in the column **New Model Role** to the right of the variable **DINEBIN**.

5. Select **Set New Model Role** from the pop-up menu.

6. Select **target**.
7. You are trying to target the buyers of **dining wares** (for whom the variable **Dinebin=1**). However other variables in the dataset contain the same information: **KITCHEN**, **DISHES**, and **FLATWARE DINEBIN** has value of 1 if the customer had bought any dining ware. It is therefore necessary to exclude them from the analysis (assign a "reject" status).

Note: Within the **Data Set Attributes Window**, the column **Measurement** refers to measurement level. This is the range of values that is found in each variable. There are five possible assignments:

unary - one value
for example, a variable with a particular value that was used to create a data subset

binary - two values
for example, the variable **MARITAL** that contains `No` or `Yes`

nominal - more than two non-numeric values, but no implied order
for example, **STATECOD** that contains `AK`, `AL`, `AR`, `AZ`, etc.

ordinal - more than two but not more than ten numeric values, with implied order
for example, **NUMCARS** that contains values from `0` to `3`

interval - more than ten numeric values
for example, **AMOUNT** that contains many different dollar values

A new order for values in Target Variable

When you build a model, Enterprise Miner considers the target event to be the first sorted value of the target variable. The default sort order is ascending. But the new target variable, **DINEBIN**, contains values of `0` and `1`, with `1` representing the purchase of any dining wares. The values need to be in descending order for Enterprise Miner to aim at the intended target.

To change the order of a target variable:

1. Click the **Class Variables** tab in the **Data Set Attributes** window.
2. Scroll down the list of variables until **the target variable** appears.
3. Right-click in the **New Order** column to the right of the taret variable.
4. Select **Set New Order** from the pop-up menu.
5. Select **Descending**.

A new level

1. In the case of the data in **DataMining2003**, the new **DINEBIN** variable has been assigned the wrong measurement level. Scroll down the **Data Set Attributes** window and notice that to the right of **DINEBIN**, to the column **Measurement**.. Because the variable contains only values of `0` and `1`, the correct measurement level is binary.
2. Right-click in the **New Measurement** column to the right of the variable **DINEBIN**.
3. Select **Set New Measurement** from the pop-up menu.
4. Select **binary**. The new measurement level of **DINEBIN** is reflected in the window.

Data Partition

1. You have control over how the partitions are created. The **Data Partition** node provides several options

2. Drag and drop the **Data Partition** node onto the workspace.

3. Connect the **Data Set Attributes** node to the **Data Partition** node.

4. Double-click the **Data Partition** node. Its window appears with the **Partition** tab in the foreground

Build a Decision Tree Model

1. Drag and drop the **Tree** node onto the workflow.

2. Connect the **Data Partition** node to the **Tree** node.
3. Double-click the **Tree** node to open it. The **Tree** window appears with the **Variables** tab in the foreground.

4. The **Basic** and **Advanced** tabs contain a set of criteria for tree building and evaluation.

If you have made changes to the default settings for this node, a new window prompts you to enter a model name

5. Enter `Direct-Tree` in the **Model Name** field.

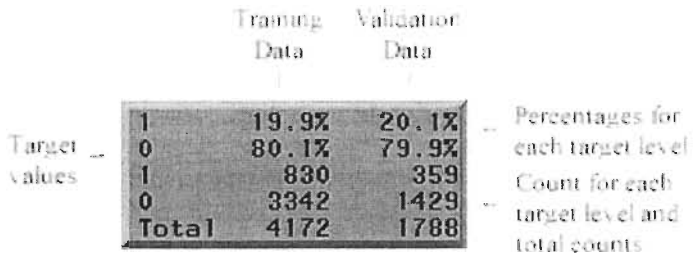
6. Enter Direct Marketing Decision Tree Model in the **Model Description** field.
7. Select **OK**. The **Save Model As** window closes and returns you to the workflow.

Views of tree structure

1. Right-click the **Tree** node.

2. Select **Run** from the pop-up menu. An alert box appears.

3. Select **Yes** to view your results. The **Tree Results** window appears with the **All** tab in the foreground.
4. Select the Plot tab to view performance on training and validation sets
5. Select View → Tree to view tree structure



Evaluation/Assessment

The **Assessment** node takes output from any modeling node and checks the model's accuracy against data in the test partition.

1. Drag and drop the **Assessment** node onto the workflow below the **Tree** node.

2. Connect the **Tree** node to the **Assessment** node.
3. Right-click the **Assessment** node.

4. Select **Run** from the pop-up menu.

As the **DataMining2003** data is processed in the workflow, each node in turn displays a thick green border. When it does, you are prompted to view the results.

5. Select **Yes**. The **Assessment Tool** window appears

Note: The **Assessment** node does not analyze all available data, it relies on sampling to produce quick results. As a consequence, different lift charts show variations even when they use the same data

Lift charts

The **Models** tab of the **Assessment Tool** window lists models that you have defined. Scrolling to the right displays statistics for each model.

1. Select the `Direct-Tree` model.

2. Select **Tools → Lift Chart** from the SAS menu bar. The **Lift Chart** window appears.

The blue baseline represents the response rate that you obtain by not using a model, but by sending the catalog to everyone in your customer database.

Diagnosis Chart (Confusion Matrix)

Select **Tools → Diagnosis chart**

Report results

The **Reporter** node is the Enterprise Miner tool for drawing a detailed map of the Data Mining process.

1. Drag and drop the **Reporter** node to the workspace.
2. Connect the **Assessment** node to the **Reporter** node.

1. Right-click the **Reporter** node.
2. Select **Run** from the pop-up menu.
5. Select **Open** when the report generation message box appears. An extensive report on your mining workflow appears in your default Web browser.

Applying the model on a data set (Testing or other)

The **Score!** node performs this task.

1. Drag and drop the **Score!** node to the workspace.
2. Connect the tree node to the score node.
3. Drag and drop the **Input Data Source** node onto the workspace above the **Score!** node.
4. Connect the **Input Data Source** node to the **Score!** node.
5. Double-click the **Input Data Source** node. The **Input Data Source** window appears.
6. As you did at the beginning of the workflow, specify **a data set** as the **Source Data**. For now, select the EMDATA directory, and select the file that begins with **TST** (plus some random characters) to be the source data. This is the test partition.
7. Change the **Role** of the data set from **RAW** to **SCORE**.
8. Close the **Input Data Source** window. A confirmation box appears.
9. Select **Yes** to save your changes.

Gain insight

1. Double-click the **Score!** node that is already in the workflow. The **Score!** window appears.

2. Select the radio button that is next to **Apply training data score code to score data set**.

3. Close the **Score!** window. A confirmation box appears.

4. Select **Yes** to save your changes.

5. Drag and drop the **Insight** node onto the workspace below the **Score!** node.

6. Connect the **Score!** node to the **Insight** node
7. Double-click the **Insight** node. The **Insight Settings** window appears.

8. Click the **Select** button that is next to the **Data set** field. The **Imports Map** window appears.You need to click through the hierarchical list of data sets to identify the data set that is associated with the score data. This data set typically has an **SD** prefix, followed by a string of random alphanumeric characters.

   Find the file

   1. Click the plus sign (+) to the left of **Score ! [Apply]** in the data set list. It expands to show **SAS_DATA_SETS**.

   2. Click the plus sign (+) to the left of **SAS_DATA_SETS**. It expands to show four data sets.

   3. Select the data set whose fileref begins with the letters **SD**. The **Role** and **Description** fields identify it as score data.
9. Click the **OK** button. You return to the **Data** tab.

10. In the **Insight based on** field, click the radio button that is next to **Entire data set**.

11. Close the **Insight Settings** window. A confirmation box appears.

12. Select **Yes** to save your changes

Examine table output

1. Right-click the new **Insight** node. Select **Run** from the pop-up menu. A message box appears.

2. Select **Yes** to view your results. The table output appears in a window
3. If you scroll to the right within the output window you can see the probability variables, **P_DINEBIN1** and **P_DINEBIN0**.

Saving and Exporting Output File

At this point, you can save the output for future use. For example, in order to save the table in excel format for further computations and analysis using Excel:

1. Click on the result icon on the window to the left. The select file → export data.
2. Select from the lists the appropriate directory (library), i.e, **EMDATA** and the appropriate DS file.
3. Click on **Next**
4. Select Microsoft Excel and the **Next**
5. Use the browse button to specify directory where the file should be saved and a file name.
6. Click on **Next** and then **Finish**.

In Excel you may calculate the profit, revenue and response rate that would be generated from the direct marketing campaign for the customers in the test set.