

**POLYTECHNIQUE MONTRÉAL**

affiliée à l'Université de Montréal

**Étude comparative de l'utilisabilité de trois interfaces de formulation de  
requêtes à une base de données relationnelle**

**KARINE GIRARD**

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie industriel

Avril 2019

# **POLYTECHNIQUE MONTRÉAL**

affiliée à l'Université de Montréal

Ce mémoire intitulé :

## **Étude comparative de l'utilisabilité de trois interfaces de formulation de requêtes à une base de données relationnelle**

présenté par **Karine GIRARD**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

**Martin TRÉPANIÉR**, président

**Jean-Marc ROBERT**, membre et directeur de recherche

**Samuel-Jean BASSETTO**, membre

## DÉDICACE

*À ma famille*

## REMERCIEMENTS

Je tiens à remercier Jean-Marc Robert, mon directeur de recherche, de m'avoir encouragée à faire ce projet de recherche. Sa direction, ses conseils et sa patience m'ont permis d'accomplir avec confiance le travail correspondant au diplôme de maîtrise.

J'aimerais remercier Croesus Lab. pour son aide financière et pour le temps alloué à ma recherche. Sans cela, mes études à la maîtrise n'auraient pas été possibles. Un merci tout particulier à mon collègue Maxime Dumas qui m'a apporté beaucoup de soutien et de conseils.

Merci à toutes les personnes qui ont participé aux tests d'utilisabilité pour avoir accepté de donner un peu de leur temps. Cela m'a permis de recueillir des données précieuses. Leur participation m'a aidée à produire une interface utilisateur plus intuitive.

Merci à mes collègues et mes amis pour leur soutien, leurs encouragements, leur écoute et leur patience tout au long du processus.

Pour finir, je remercie ma famille qui m'encourage dans mes projets de toutes natures.

## RÉSUMÉ

L'utilisation de bases de données relationnelles est une activité complexe pour la grande majorité des utilisateurs et cela ne fait pas exception dans une entreprise de technologie financière comme Croesus Finansoft. De nombreux chercheurs et développeurs se sont intéressés à cette activité ces dernières années pour tenter de réduire la complexité et d'améliorer l'utilisabilité des interfaces utilisateurs des systèmes de requêtes à une base de données relationnelle.

L'objectif du projet de recherche est de développer une solution d'interface utilisateur permettant aux utilisateurs d'un logiciel de gestion de portefeuilles de créer plus facilement des requêtes à une base de données qu'avec les interfaces actuelles.

La recherche a été effectuée en collaboration avec Croesus Lab. qui œuvre dans le domaine de la technologie financière et offre des produits logiciels aux grandes institutions financières canadiennes. Elle comporte une étude des problèmes rencontrés par les utilisateurs avec les interfaces d'un logiciel de gestion de portefeuille comprenant une interface de critères de recherche simplifiée basée sur une approche par filtres combinée avec une approche de langage semi-naturel sous forme de phrases structurées rigides et une interface de critères de recherche avancée basée sur une approche par filtres dans un formulaire flexible présenté sous forme hiérarchique. La recherche a aussi mené à la conception de maquettes puis d'un prototype d'interface utilisateur basée sur une approche de formulaire flexible de formulation de requêtes par diagramme. Trois cycles de tests d'utilisabilité visant à tester les maquettes et le prototype et à évaluer les résultats en termes de performance et de satisfaction humaine ont été effectués durant le processus de conception. À chaque cycle de tests, la chercheuse a observé soigneusement les comportements des utilisateurs. De plus, elle a mené des entrevues semi-dirigées pour recueillir les commentaires et les impressions des utilisateurs. Ces informations ont permis d'améliorer les maquettes.

Dix et 12 sujets ont participé respectivement au premier et au second cycle de tests d'utilisabilité portant sur les maquettes. Par la suite, le prototype fonctionnel d'interface a été développé et un troisième cycle de tests d'utilisabilité a été effectué avec celui-ci auprès de six sujets.

Pour finir, nous avons mené une étude expérimentale comparative visant à tester trois hypothèses (H) stipulant que la nouvelle interface du prototype donne de meilleurs résultats que les deux interfaces actuelles en termes de temps d'exécution des tâches (H1), de nombre d'erreurs (H2) et de nombre de demandes d'assistance (H3) de la part des utilisateurs. Douze utilisateurs connaissant

déjà les deux interfaces actuelles du logiciel de gestion de portefeuilles ont participé aux tests ; chacun devait effectuer trois tâches de création de requêtes avec chacune des trois interfaces.

De façon générale, les commentaires recueillis auprès des utilisateurs au sujet du prototype de la nouvelle interface (formulaire flexible de formulation de requêtes par diagramme) sont positifs : l'interface est considérée comme plus intuitive, elle nécessite moins d'opérations pour formuler les différentes conditions des requêtes et l'utilisateur est guidé pour l'ajout d'opérateurs logiques ET/OU. Les résultats montrent aussi que même si les utilisateurs n'ont aucune expérience avec la nouvelle interface contrairement aux deux autres interfaces, la durée de la formulation de requêtes tend à être plus courte avec la nouvelle interface pour toutes les tâches effectuées : toutefois, une seule (sur trois) offre une différence significative en termes statistiques entre la nouvelle interface et l'interface des critères de recherche avancée. L'hypothèse H1 est donc partiellement confirmée, pour une interface sur deux et pour une tâche sur trois)

De plus, la nouvelle interface a permis de réduire le nombre d'erreurs de façon significative par rapport à l'interface des critères de recherche avancée pour l'une des trois tâches. L'hypothèse H2 est partiellement confirmée (pour une interface sur deux, et pour une tâche sur trois). Enfin, il n'y a pas de différence significative entre les deux interfaces actuelles et le prototype de formulaire flexible de formulation de requêtes par diagramme quant au nombre de demandes d'assistance à la tâche. L'hypothèse H3 est donc infirmée. Dans la discussion, nous tentons d'expliquer ces résultats et dans la conclusion, nous proposons quelques pistes de recherches pour la suite.

## ABSTRACT

Exploration of relational databases is a complex activity for most people and database users of financial technology software like Croesus Finansoft are no exception. Lots of research has been done in the late years to reduce the complexity and improve the usability of interfaces used to create relational database requests.

The goal of this research is to develop a user interface solution, allowing portfolio management software users to create relational database requests in an easier way than with the actual interfaces.

The research has been done in collaboration with Croesus Lab. which works in financial technology field and offers software products to Canadians financial institutions. It includes a study of the problems encountered by the portfolio management software users offering a simplified search criteria interface based on a filter approach combined with a rigid semi-natural language approach and an advanced search criteria interface based on a filter approach presented in a hierarchical flexible form. The research has led to the conception of a mock-up and an interface prototype based on an approach of a flexible form or query formulation by diagram.

Three cycles of usability tests were done to test the mock-ups and the prototype and to evaluate the results in terms of performance and human satisfaction has been done during the conception process. In every meeting between the searcher and a subject, the searcher has taken notes about her observations of the subject's behaviour. At the end of each meeting, an interview was conducted by the searcher to collect feedback from the subjects. The information has been used to improve the mock-ups.

Ten and 12 subjects have participated respectively to the first and second usability test cycles on mock-ups. Then, the functional prototype has been developed and a third cycle of usability tests has been done with six subjects.

An experimental comparative study has been conducted to validate three hypotheses (H) saying that the new interface prototype gives better results than the two actual interfaces in terms of tasks execution time (H1), the number of errors (H2) and the amount of assistance demands (H3) from users. Twelve users already knowing the two actual interfaces have participated in the tests; each of them had to perform three tasks of requests creation with each of the three interfaces.

In general, the collected comments on the new interface prototype (interface based on an approach of a flexible form or query formulation by diagram) are positive : the interface is considered more intuitive, it requires fewer operations to formulate the different request's conditions and the user is guided during the insertion of logical operators AND/OR. The results show that even if the user has no experience with the new interface, the request formulation duration tends to be shorter with the new interface for all the tasks : however, only one task (on three) offers a significant difference in statistic terms between the new interface and the advanced search criteria. The hypothesis H1 is partially confirmed for one interface out of two and one task out of the three.

The new interface as significantly reduce the number of errors compared to the advanced search criteria interface for one task out of the three. The hypothesis H2 is partially confirmed (for one interface out of two and for one task out of the three). There is no significant difference between the two actual interfaces and the new interface in terms of the amount of assistance demands. The hypothesis H3 is informed. In the discussion, we try to explain the results and the conclusion. New search fields are proposed for the future.



## TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS .....	IV
RÉSUMÉ.....	V
ABSTRACT .....	VII
TABLE DES MATIÈRES .....	IX
LISTE DES TABLEAUX.....	XV
LISTE DES FIGURES.....	XVII
LISTE DES SIGLES ET ABRÉVIATIONS .....	XIX
LISTE DES ANNEXES .....	XX
<b>CHAPITRE 1 INTRODUCTION.....</b>	<b>1</b>
1.1 Problématiques étudiées dans le projet .....	2
1.1.1 Complexité liée aux jointures.....	2
1.1.2 Complexité liée à la charge cognitive .....	2
1.1.3 Complexité liée à la priorité des opérateurs logiques ET/OU.....	3
1.1.4 Aspect intuitif de l’interface.....	3
1.2 Question de recherche .....	3
1.3 Nature de la collaboration avec l’entreprise.....	4
1.4 Contribution .....	5
<b>CHAPITRE 2 PRÉSENTATION DU SYSTÈME ACTUEL DE FORMULATION DE REQUÊTES À UNE BASE DE DONNÉES RELATIONNELLE ET DES PROBLÈMES QUI LUI SONT RELIÉS .....</b>	<b>6</b>
2.1 Présentation du logiciel Croesus Conseiller.....	6
2.2 Définition des concepts .....	12
2.3 Problématiques rencontrées.....	19

2.3.1	Problématiques liées à l'interface de création de critères de recherches simplifiée ..	19
2.3.2	Problématiques liées à l'interface de création de critères de recherches avancée..	22
2.3.3	Problématiques de la fonctionnalité des filtres sur une liste d'information ..	25
2.3.4	Problématiques liées à l'utilisation de l'opérateur logique OU ..	26
2.3.5	Problématiques liées à la confusion entre les différentes interfaces de formulation de requêtes ..	27
2.3.6	Conclusion sur les problématiques ..	27
2.4	Statistiques de l'entreprise ..	28
<b>CHAPITRE 3 REVUE DE LITTÉRATURE SUR LES APPROCHES DE CONCEPTION DES REQUÊTES À DES BASES DE DONNÉES ET HYPOTHÈSES ..</b>		<b>30</b>
3.1	Mécanisme de recherche bibliographique ..	30
3.2	Qu'est-ce qu'un système de gestion de base de données (SGBD) ? ..	31
3.3	Aspects à considérer lors du développement d'une interface de formulation de requêtes ..	32
3.4	Approches de conception de requêtes à des bases de données relationnelles ..	33
3.4.1	Recherche par mots clés ..	34
3.4.2	Modèle de requêtes structuré souple ..	36
3.4.3	Modèle basé sur le langage naturel ..	37
3.4.4	Modèle basé sur les interfaces de type formulaires (VQI « form based ») ..	38
3.4.5	Formulation de requêtes avec un langage graphique (GQL) ..	39
3.4.6	Formulation de requêtes par filtres ..	40
3.4.7	Formulation de requêtes par diagramme ..	41
3.4.8	Formulation de requêtes par des gestes ..	42
3.4.9	Formulation de requêtes par la manipulation directe des données ..	43

3.4.10	Formulation de requêtes par recherche approximative [traduction libre de « fuzzy search »]	44
3.5	Catégorisation des irritants lors de la formulation de requêtes	44
3.5.1	Irritants avec les jointures [traduction libre de « Painful relations »]	44
3.5.2	Irritants avec les options multiples [traduction libre de « Painful options »]	45
3.5.3	Irritants avec les résultats inattendus [traduction libre de « unexpected pain »]	45
3.5.4	Irritants avec les résultats non visibles [traduction libre de « unseen pain »]	46
3.6	Catégorisation des défis rencontrés par les utilisateurs d'interface de formulation de requêtes	46
3.6.1	Manque de connaissance	47
3.6.2	Dépendance de l'information	47
3.6.3	Construction de requête de façon itérative et incrémentale	47
3.6.4	Intention imprécise	48
3.7	Principes de l'interaction guidée	48
3.8	Approche de conception centrée sur l'utilisateur	49
3.8.1	Efficacité	49
3.8.2	Efficiencie	49
3.8.3	Satisfaction	50
3.8.4	Facilité d'apprentissage et la mémorisation	51
3.9	Processus de conception centré sur l'utilisateur	51
3.10	Facteurs humains dans le design et l'ingénierie	52
3.10.1	Affichage de l'information	53
3.10.2	Encombrement visuel et densité d'information	54
3.11	Synthèse	56
3.12	Hypothèses	57

<b>CHAPITRE 4</b>	<b>MÉTHODOLOGIE DE DÉVELOPPEMENT ET DE TEST DES MAQUETTES ET DU PROTOTYPE DE LA NOUVELLE INTERFACE.....</b>	<b>59</b>
4.1	Méthodologie de développement et de test.....	59
4.2	Vue d'ensemble du développement et de l'évaluation du prototype de la nouvelle interface.....	59
4.3	Premier et deuxième cycles de tests d'utilisabilité.....	61
4.3.1	Préparation des tests d'utilisabilité.....	61
4.3.2	Sujets.....	61
4.3.3	Tâches.....	62
4.3.4	Ordre d'utilisation des interfaces.....	63
4.3.5	Procédure des tests.....	64
4.4	Troisième et quatrième cycles de tests d'utilisabilité.....	65
4.4.1	Préparation des tests d'utilisabilité.....	65
4.4.2	Sujets.....	66
4.4.3	Tâches.....	67
4.4.4	Ordre d'utilisation des interfaces.....	67
4.4.5	Procédure des tests.....	68
<b>CHAPITRE 5</b>	<b>CONCEPTION CENTRÉE SUR L'UTILISATEUR.....</b>	<b>70</b>
5.1	Premier cycle de conception - Étape 1 – Analyse des utilisateurs et du contexte.....	70
5.1.1	Utilisateurs visés.....	70
5.1.2	Contexte d'utilisation.....	71
5.2	Premier cycle de conception - Étape 2 – Spécification des besoins.....	71
5.2.1	Schéma de la base de données.....	71
5.2.2	Besoins en affichage.....	71

5.3	Premier cycle de conception - Étape 3 – Production d’une solution sous forme de captures d’écrans .....	72
5.3.1	Choix du type d’interface .....	72
5.3.2	Solution .....	73
5.4	Premier cycle de conception - Étape 4 – premier cycle de tests d’utilisabilité .....	76
5.5	Premier cycle de conception - Étape 5 - Analyse des résultats suite aux tests d’utilisabilité préliminaires .....	76
5.5.1	Évaluer la connaissance des sujets sur les opérateurs utilisés pour les ensembles .....	76
5.5.2	Opérateurs utilisés pour les conditions simples .....	77
5.6	Deuxième cycle de conception .....	77
5.6.1	Modèle logique .....	77
5.6.2	Affichage des jointures et groupements .....	78
5.6.3	Encadrés orangés pour les conditions simples .....	78
5.6.4	Affichage des opérateurs .....	78
5.7	Améliorations suite au deuxième cycle .....	78
5.8	Troisième cycle de conception .....	79
5.8.1	Analyse des résultats des tests d’utilisabilité préliminaires .....	80
5.9	Conclusion .....	84
<b>CHAPITRE 6 CONCLUSION (ET RECOMMANDATIONS) .....</b>		<b>85</b>
6.1	Résultats des tests d’utilisabilité de la phase 2 .....	85
6.1.1	Synthèse des résultats statistiques pour la phase 2 .....	85
6.1.2	Détail des calculs statistiques pour la phase 2 .....	87
6.1.3	Analyse de variance – temps de traitement .....	90
6.1.4	Analyse de la variance - nombre de demandes d’assistance .....	98
6.1.5	Résumé des résultats .....	101

6.2	Ordre de préférence et commentaires recueillis lors des entrevues .....	103
6.2.1	Opérateur logique ET/OU .....	103
6.2.2	Facilité d'utilisation.....	104
6.2.3	Capacité équivalente aux interfaces actuelles .....	104
6.2.4	Choix de couleurs.....	105
6.2.5	Modèle logique d'entités.....	105
<b>CHAPITRE 7 DISCUSSION ET CONCLUSION.....</b>		<b>107</b>
7.1	Discussion .....	107
7.1.1	Hypothèse H1 sur le temps requis de formulation de requêtes .....	107
7.1.2	Hypothèse H2 sur le nombre d'erreurs lors de formulation de requêtes.....	107
7.1.3	Hypothèse H3 le nombre de demandes d'assistance pour la formulation de requêtes .....	108
7.1.4	Autres observations.....	108
7.2	Conclusion.....	109
7.2.1	Limites de la recherche .....	109
7.2.2	Pistes de recherche .....	110
7.2.3	Contribution de notre recherche.....	111
7.2.4	Suite du projet .....	112
<b>RÉFÉRENCES.....</b>		<b>113</b>
ANNEXES .....		117

## LISTE DES TABLEAUX

Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document .....	12
Tableau 2.2 Synthèse des capacités offertes pour la formulation de requêtes selon l'interface disponible. ....	18
Tableau 4.1 Vue d'ensemble des phases de développement et des cycles de tests d'utilisabilité.	60
Tableau 4.2 Ordre des interfaces utilisées pour faire les 12 tâches.....	64
Tableau 4.3 Ordre des interfaces pendant le quatrième cycle de tests. ....	68
Tableau 6.1 Détails des tâches qui ont été soumises aux sujets lors de la phase 2 des tests.....	86
Tableau 6.2 Moyenne et écart type de la durée de chaque tâche pour chaque interface. Durée en secondes (moyenne sur 12 sujets). ....	87
Tableau 6.3 Moyenne et écart type du nombre d'erreurs pour chaque tâche et chaque interface.	88
Tableau 6.4 Moyenne et écart type du nombre de demandes d'assistance à la tâche des sujets...	89
Tableau 6.5 Analyse de variance : un facteur pour le temps de traitement de la tâche 1. ....	90
Tableau 6.6 Test de Student (ou test-t) sur des échantillons appariés.....	91
Tableau 6.7 Analyse de variance : un facteur pour le temps de traitement de la tâche 2. ....	92
Tableau 6.8 Analyse de variance : un facteur pour le temps de traitement de la tâche 3. ....	93
Tableau 6.9 Analyse de variance : un facteur pour le nombre d'erreurs de la tâche 1. ....	94
Tableau 6.10 Analyse de variance : un facteur pour le nombre d'erreurs de la tâche 2. ....	95
Tableau 6.11 Analyse de variance : un facteur pour le nombre d'erreurs de la tâche 3. ....	96
Tableau 6.12 Test de Student (ou test-t) sur des échantillons appariés.....	97
Tableau 6.13 Analyse de variance : un facteur pour le nombre de demandes d'assistance à la tâche des utilisateurs dans la tâche 1. ....	98
Tableau 6.14 Analyse de variance : un facteur pour le nombre de demandes d'assistance à la tâche des utilisateurs dans la tâche 2. ....	99

Tableau 6.15 Analyse de variance : un facteur pour le nombre de demandes d'assistance à la tâche des utilisateurs dans la tâche 3. ....	100
Tableau 6.16 Résumé des résultats de l'analyse de variance. ....	101
Tableau A.1 Données démographiques.....	118
Tableau C.1 Temps d'exécution des tâches effectuées durant la phase 1 des tests d'utilisabilité. (en secondes).....	163
Tableau C.2 Taux de succès pour les tâches du test d'utilisabilité de la phase 1 (la valeur affichée correspond au nombre de réponses valide sur un total de 11) .....	164
Tableau C.3 Taux de succès pour les tâches du test d'utilisabilité de la phase 1 (la valeur affichée correspond au nombre de réponses valide sur un total de 11) .....	165
Tableau C.4 Temps de traitement des tâches de la phase 2 en secondes .....	166
Tableau C.5 Données brutes du nombre d'erreurs pour les tâches du test d'utilisabilité .....	167
Tableau C.6 Données brutes du nombre de demandes d'assistance à la tâche des sujets pour les tâches du test d'utilisabilité .....	168
Tableau C.7 Préférence des sujets.....	169



## LISTE DES FIGURES

Figure 2.1 Page-écran principale de l'application Croesus Conseiller. Le module Clients est actif. Les icônes permettant de gérer et créer des critères de recherche sont encerclées en rouge. Le nom du critère de recherche actif sur le module est indiqué par une flèche rouge. (Les données présentées sont fictives). .....	8
Figure 2.2 Page-écran permettant de faire la gestion des critères de recherche sauvegardés. ....	9
Figure 2.3 Page-écran principale de l'application Croesus Conseiller. Le module Tableau de bord est actif. Les encadrés présentent la liste de résultats d'un critère de recherche qui est exécuté durant la nuit. (Les données présentées sont fictives). .....	10
Figure 2.4 Exemple de condition simple composée d'un champ (associé à un élément) d'un opérateur et d'une valeur. ....	13
Figure 2.5 Page-écran de l'interface de formulation des critères de recherche simplifiée lors de la création d'une nouvelle requête. Les éléments textuels affichés en bleu sont des hyperliens donnant accès à des listes de choix. L'exemple présente un critère de recherche dont le résultat attendu est une liste de comptes. ....	20
Figure 2.6 Séquence des étapes à effectuer par l'utilisateur pour formuler la requête. L'utilisateur doit actionner les hyperliens pour obtenir une liste de choix lui permettant de composer la requête à l'aide de jointures, champs, opérateurs, valeurs et opérateurs logiques. ....	21
Figure 2.7 Page-écran de l'interface de critères de recherche simplifiée. ....	22
Figure 2.8 Page-écran de l'interface de critères de recherche avancée à son état initial. On peut voir que la liste des résultats attendus est une liste de comptes. ....	23
Figure 2.9 Séquence des étapes à effectuer par l'utilisateur pour formuler la requête. La création et l'ajout d'une condition simple est effectuée dans la section tout en bas de l'interface. Par la suite, l'ajout de jointures et d'opérateurs logiques est effectué à l'aide du bouton ajouter situé à la droite de la requête. ....	24
Figure 2.10 Page-écran de l'interface de critères de recherche avancée. ....	25
Figure 2.11 Page-écran de fonctionnalité des filtres sur une liste de comptes. Les filtres de couleur orangée sont en haut à gauche de la liste. (N.B. Les données présentées sont fictives). ....	26

Figure 3.1 Diagramme qui présente le processus de conception centré sur l'utilisateur défini par la norme ISO 9241-210. (l'image est tirée du site suivant : <a href="https://www.researchgate.net/figure/Modele-UCD-tire-de-la-norme-ISO-9241-210_fig1_280851696">https://www.researchgate.net/figure/Modele-UCD-tire-de-la-norme-ISO-9241-210_fig1_280851696</a> date du 9 octobre 2018) .....	52
Figure 5.1 Interface de formulation de requêtes provenant du prototype de formulaire flexible de formulation de requêtes par diagramme.....	75
Figure 5.2 Interface de formulation de requêtes provenant du prototype suite aux changements apportés après les tests d'utilisabilité de la phase 1. ....	79
Figure 5.3 Menu contenant la liste des opérateurs disponibles pour les conditions simples avant l'amélioration. ....	80
Figure 5.4 Menu contenant la liste des opérateurs disponibles pour les conditions simples après l'amélioration. ....	81
Figure 5.5 Menu contenant la liste des opérateurs disponibles pour les conditions simples. ....	82
Figure 5.6 Interface de formulation de requêtes provenant du prototype suite aux changements apportés après les tests d'utilisabilité de la phase 2. ....	83
Figure 5.7 Graphique qui présente le modèle logique des éléments impliqués dans la formulation des requêtes. ....	84
Figure 6.1 Interface de formulation de critères de recherche provenant du prototype de formulaire flexible de formulation de requêtes par diagramme.....	104
Figure 6.2 Graphique présentant le modèle logique d'entités disponibles pour la création de requêtes.....	106
Figure B.1 Graphique présentant le modèle logique d'entités disponibles pour la création de requêtes.....	155

## LISTE DES SIGLES ET ABRÉVIATIONS

CELI	Compte Enregistré Libre d'Impôt
ET	Opérateur logique SQL (AND)
FER	Fond Enregistré de Retraite
GQL	Graphical Query Language
INVERSER	Opérateur logique SQL (NOR, NAND, NOT)
Mashup	Application qui consomme et combine des données provenant de plusieurs sources différentes (Web API)
Objet	Type d'élément retourné par la requête type d'élément sur lequel les conditions de la requête sont basées
OU	Opérateur logique SQL (OR)
Position	Titre financier qui est détenu par un compte. La position comporte l'information sur le titre, l'identifiant du compte détenteur, une quantité détenue à une date spécifique.
RÉER	Régime d'Épargne Enregistré de Retraite
Requête	Permet d'extraire de la base de données une liste d'éléments répondant tous à une série de conditions
RDBMS	Relational Database Management System
SGBD	Système de Gestion de Base de Données
SPARQL	SPARQL Protocol And RDF Query Language (acronyme récursif)
SQL	Structured Query Language
IU	Interface Utilisateur (User Interface)
VQI	Visual Query Interface
WCAG	Web Content Accessibility Guidelines
Web API	Application Programming Interface Source de données disponibles sur le web

**LISTE DES ANNEXES**

Annexe A Tests d'utilisabilité phase 1 – cycle 1 et 2 .....	117
Annexe B Tests d'utilisabilité Phase 1 – cycle 3 et phase 2 .....	152
Annexe C Résultats statistiques pour les phases 1 et 2 .....	163

## CHAPITRE 1 INTRODUCTION

L'entreprise Croesus Finansoft offre des produits et des services à de grandes institutions financières canadiennes pour la gestion des portefeuilles financiers de leurs clientèles. Les systèmes d'information offerts par cette entreprise à ses clients présentent une grande quantité d'informations sur les clients, sur leurs portefeuilles financiers et sur les titres des marchés boursiers. Ces informations sont conservées dans des bases de données complexes. Au fil des années, les utilisateurs deviennent de plus en plus exigeants et créatifs lorsqu'ils consultent et analysent les données. Ils désirent être de plus en plus autonomes et performants dans leurs tâches. Pour ce faire, ils désirent avoir un système qui leur permet entre autres de créer leurs propres requêtes à la base de données.

Durant les dernières années, l'entreprise s'est efforcée de fournir des fonctionnalités permettant aux utilisateurs de consulter les données dont ils ont besoin pour faire leur travail. Le défi rencontré est d'offrir une interface utilisateur flexible qui permet de créer des requêtes avec une sémantique complexe tout en étant facile et agréable à utiliser. Les utilisateurs des systèmes d'informations offerts par Croesus Finansoft n'ont pas de formation en informatique. Ils ne sont donc pas familiarisés avec les langages de requêtes à des bases de données tels que SQL (Langage structuré de requêtes ou « Structured Query Language ») ou SPARQL (SPARQL Protocol And RDF Query Language). L'interface doit donc guider l'utilisateur dans la création de requêtes.

Selon les informations recueillies auprès de 1027 utilisateurs (sur un total de 6012) par un sondage effectué en 2015 par la firme BIP mandatée par l'entreprise Croesus Finansoft, les interfaces utilisateurs actuellement offertes ne sont pas très intuitives ni agréables à utiliser. Il y a déjà plusieurs années que cette situation est connue dans l'entreprise. C'est pour cette raison que nous avons choisi d'étudier les problèmes d'utilisation actuels et de tenter d'améliorer les interfaces.

L'objectif du projet de recherche est de développer une solution d'interface utilisateur permettant aux utilisateurs d'un logiciel de gestion de portefeuilles de créer plus facilement des requêtes à une base de données qu'avec les interfaces actuelles.

Le document est structuré comme suit. En premier lieu, on présente une étude des problèmes d'utilisation, puis une revue de la littérature sur les approches possibles pour la consultation de données ainsi que les difficultés rencontrées par les utilisateurs lors de la consultation de base de

données. Par la suite, nous avons développé un prototype d'interface utilisateur de formulaire flexible de formulation de requêtes par diagramme pour corriger les problèmes rencontrés. Enfin, nous avons effectué une étude expérimentale qui comparait les niveaux de performance humaine dans la formulation de requêtes afin de voir si la nouvelle interface est plus performante et mieux perçue que les interfaces actuelles. Pour la conception du prototype de formulaire flexible de formulation de requêtes par diagramme, l'approche centrée sur l'utilisateur telle que définie par la norme ISO 9241-210 a été utilisée [1]. Des tests d'utilisabilité ont été effectués durant la conception. Trois cycles de conception suivis de trois cycles de tests d'utilisabilité ont été effectués avant d'effectuer l'étude expérimentale de comparaison.

## **1.1 Problématiques étudiées dans le projet**

### **1.1.1 Complexité liée aux jointures**

La première problématique étudiée dans notre projet porte sur la complexité de créer des jointures entre les éléments et des groupements de conditions sur les éléments utilisés dans les jointures. En effet, la revue de la littérature nous indique que l'utilisateur ne doit pas avoir à faire lui-même la gestion de l'ajout de parenthèses ou savoir comment grouper les conditions de jointure [2]. La nouvelle interface développée dans le cadre de ce projet doit faire en sorte que les groupements se créent automatiquement lorsque cela est nécessaire.

La solution doit donc éviter à l'utilisateur de gérer les parenthèses. L'utilisateur doit être guidé lors de la création des jointures.

### **1.1.2 Complexité liée à la charge cognitive**

La deuxième problématique étudiée porte sur la charge cognitive relative à la formulation d'une requête impliquant des jointures sur plusieurs entités. Cela nécessite une bonne compréhension du modèle relationnel entre les entités. Il est aussi nécessaire de bien comprendre l'effet de l'application de certains filtres sur les différentes entités impliquées et sur les relations qui les lient.

L'interface doit permettre à l'utilisateur de bien comprendre le modèle logique qui est offert pour lui permettre de comprendre les interactions entre les différentes entités impliquées. Il doit être facile pour l'utilisateur de comprendre l'effet des jointures sur la condition pour anticiper le résultat final.

### **1.1.3 Complexité liée à la priorité des opérateurs logiques ET/OU**

La troisième problématique étudiée porte sur la complexité de gérer les priorités des opérateurs logiques ET et OU et encore une fois les groupements de conditions. La nouvelle interface développée dans le cadre de ce projet doit gérer de façon automatique les groupements de conditions et les priorités des opérateurs logiques ET et OU. De plus, pour réduire l'utilisation des opérateurs logiques OU, les champs de type « valeur provenant d'un dictionnaire de valeurs » doivent permettre une sélection multiple des valeurs avec un opérateur « parmi » ou « excluant ». Dans le même ordre d'idée, les champs de type « chaîne de caractères » doivent permettre de spécifier plusieurs valeurs.

La solution doit réduire l'utilisation de l'opérateur logique. Pour ce faire, l'utilisateur doit être en mesure de spécifier une liste de plusieurs valeurs. Les opérateurs « égal » et « n'égal pas » sont alors remplacés par « parmi » ou « excluant ».

### **1.1.4 Aspect intuitif de l'interface**

La quatrième problématique concerne l'aspect intuitif de l'interface. Les différentes opérations requises pour la formulation de la requête doivent être effectuées de façon intuitive et les fonctionnalités qui leur correspondent doivent donc se trouver à un endroit stratégique et logique pour l'utilisateur.

La disposition des fonctionnalités doit permettre à l'utilisateur de comprendre facilement comment faire pour formuler la requête.

## **1.2 Question de recherche**

La question de recherche qui découle des problématiques étudiées dans le projet est la suivante : comment assurer la bonne utilisabilité de l'interface de création de requêtes dans une base de données relationnelle ?

Les critères utilisés pour savoir si l'objectif a été atteint avec la nouvelle interface utilisateur sont les suivants :

- le temps requis pour formuler une requête dans l'interface ;
- le nombre d'erreurs lors de la formulation d'une requête dans l'interface ;

- le nombre de demandes d'assistance à la tâche lors de la création de requêtes.

Afin d'atteindre l'objectif défini ci-dessus, différentes activités de recherche sont réalisées :

- une analyse critique des interfaces actuelles de formulation de requêtes dans l'application Croesus Conseiller (chapitre 2) ;
- une analyse des solutions possible pour corriger les situations problématiques à la lumière d'une revue de la littérature sur les interfaces de création de requêtes à une base de données relationnelle (chapitre 3) ;
- le développement d'un prototype fonctionnel d'une interface utilisateur de formulation de requêtes selon le processus de conception centrée sur l'utilisateur (norme ISO 9241-210 [1]) (Phase 1 du projet) ;

Une étude comparative de la performance des utilisateurs portant sur la nouvelle interface de création de requêtes et sur les interfaces actuelles dans l'application Croesus Conseiller (Phase 2 du projet).

### **1.3 Nature de la collaboration avec l'entreprise**

Les travaux effectués dans le cadre de la présente recherche ont été effectués en collaboration avec l'entreprise Croesus Finansoft. La chercheuse est une employée à temps plein de l'entreprise depuis mars 2000 et a profité du programme de formation personnalisé offert par l'entreprise (PIF). Par le biais de ce programme, l'entreprise a remboursé les frais de scolarité du programme de maîtrise. De plus, dans le cadre du laboratoire de recherche de l'entreprise nommé Croesus Lab, l'entreprise a versé un salaire à la chercheuse pour le temps investi à la recherche à raison de quatre jours par semaine à partir du mois de janvier 2018. Les heures supplémentaires investies ont été assumées par la chercheuse. Par ailleurs, l'entreprise a fourni à la chercheuse un ordinateur ainsi que tous les logiciels requis pour l'avancement de la recherche. Par exemple, le logiciel Visual Studio, un accès à une base de données de tests sur un serveur Sybase, un environnement de développement sur l'application principale. Pour la documentation du mémoire, l'entreprise a fourni l'accès à MS Word, MS Excel ainsi que les applications de la suite Google de l'entreprise. L'entreprise a aussi permis de faire des tests d'utilisabilité auprès de sujets qui sont des employés de l'entreprise.



La fonctionnalité visée par la recherche a été déterminée en de façon conjointe par la chercheure et l'équipe de direction de l'entreprise. L'objectif de l'entreprise étant de profiter des travaux de recherche pour améliorer une fonctionnalité problématique et complexe du logiciel. Comme la chercheure est étudiante en ergonomie du logiciel, le projet devait porter sur l'amélioration de l'interface utilisateur. Plusieurs fonctionnalités ont été considérées et c'est la fonctionnalité des critères de recherche qui a été retenue.

## **1.4 Contribution**

Les travaux de recherche ainsi que la conception et le développement de la solution logicielle ont été effectués entièrement par la chercheure. Le chef d'équipe de Croesus Lab a effectué un suivi hebdomadaire avec la chercheure pour s'assurer de l'avancement des travaux. Ce dernier a apporté une assistance à la chercheure dans deux situations. Premièrement, trois séances de remue-ménages pour le choix de la solution implémentée. Dans cette séance, des outils de requête en langage naturel ont été évalués, mais rejetés parce qu'ils ont été jugés trop compliqués à manipuler par les utilisateurs. Deuxièmement, il a assisté la chercheure pour effectuer les calculs de la variance qui sont présentés au chapitre 6 du mémoire.

Afin de créer des tests réalistes, les tâches ont été élaborés par la chercheure, basés sur des exemples de critères de recherche qui ont été sauvegardés par des utilisateurs réels. Plusieurs employés ont participé aux tests d'utilisabilité pour évaluer le temps requis et le niveau de complexité des tâches avant que nous rencontrions les sujets pour réaliser l'étude expérimentale comparative. L'entreprise a assumé le montant des salaires des employés pour le temps alloué aux tests d'utilisabilité. Le chef d'équipe de Croesus Lab. a participé à l'interprétation des résultats des tests, mais n'a entendu aucun enregistrement audio des tests d'utilisabilité.

Un designer graphique a aussi participé à l'amélioration de l'aspect visuel de l'interface pour réduire la charge visuelle. Sa contribution s'est limitée à quelques heures de travail.

## **CHAPITRE 2 PRÉSENTATION DU SYSTÈME ACTUEL DE FORMULATION DE REQUÊTES À UNE BASE DE DONNÉES RELATIONNELLE ET DES PROBLÈMES QUI LUI SONT RELIÉS**

Les fonctionnalités du produit Croesus Conseiller sont dépendantes de bases de données relationnelles complexes. Afin d'améliorer l'autonomie des utilisateurs pour la consultation des données du système, trois interfaces de création de requêtes à la base de données sont actuellement offertes dans l'application Croesus Conseiller. L'utilisation de ces interfaces est difficile pour beaucoup d'utilisateurs. Ces derniers ont alors recours aux services du Centre de services aux clients de l'entreprise pour les aider à atteindre leurs objectifs.

Dans ce chapitre, nous allons tout d'abord, présenter une brève description du logiciel Croesus Conseiller ainsi que l'historique du logiciel. Ensuite, afin d'aider le lecteur à bien comprendre le présent document, nous définissons les concepts couramment utilisés dans la formulation de requêtes et qui apparaissent tout au long du document. Par la suite, nous expliquons les problématiques rencontrées par les utilisateurs du produit Croesus Conseiller lors de l'utilisation des interfaces de formulation de requêtes. Enfin, nous présentons des statistiques sur le nombre de demandes d'assistance adressées au Centre de services aux clients et des données sur la formation offerte par l'entreprise Croesus Finansoft à ses utilisateurs.

### **2.1 Présentation du logiciel Croesus Conseiller**

Le logiciel Croesus Conseiller est offert à de grandes institutions financières canadiennes. La première version du logiciel développée en 1987 était destinée à des conseillers en placement. Par la suite, le logiciel a évolué pour répondre aux besoins d'autres catégories d'utilisateurs telles que l'assistante au conseiller, le gestionnaire de succursale, le gestionnaire de firme, l'administrateur de système, le gestionnaire de portefeuille et aussi des équipes de conformité.

La page-écran principale du logiciel présente plusieurs modules. Les modules Relations, Clients et Comptes permettent de faire la gestion de la clientèle d'un conseiller. Ils offrent des fonctions de production de rapports, d'un agenda, d'un système d'envoi de courriels et plusieurs autres fonctions relatives à cette gestion.

Un module permettant de faire l'analyse d'un portefeuille financier est aussi disponible. Le module Portefeuille permet de consulter l'état actuel d'un portefeuille en présentant chacune des positions détenues actuellement avec la quantité et la valeur de la position (titre détenu). Il permet aussi de consulter un sommaire du portefeuille et de s'assurer que l'objectif de placement du client est respecté.

Le module Transactions permet de consulter l'historique des transactions du portefeuille. Le module Titres est aussi disponible pour consulter les prix des titres ainsi que d'autres informations pertinentes.

Le module Modèles permet de créer un portefeuille cible auquel on associe des portefeuilles réels. Le système utilise l'intelligence artificielle pour analyser les portefeuilles et proposer des ordres d'achat et de vente qui permettent d'ajuster les portefeuilles réels au portefeuille cible.

Chacun des modules affiche une grille qui expose une liste d'éléments. Les utilisateurs ont besoin d'outils permettant de faire des recherches dans les différentes listes basées sur des critères précis. Tous les modules offrent une fonction de filtres et/ou de critères. Les différentes fonctionnalités varient d'un module à l'autre en fonction de l'époque où le module a été développé.

La figure 2.1 présente la page-écran principale de l'application. On peut voir sur la gauche une série d'onglets qui présentent les différents modules. On peut aussi constater que le module client est activé grâce à la couleur blanche de l'onglet. Un cercle rouge a été ajouté pour indiquer les icônes qui permettent de faire la gestion et la création d'un critère de recherche. La flèche rouge indique l'endroit où le nom du critère de recherche apparaît lorsqu'il a été activé sur la grille du module.

The screenshot displays the main interface of the Croesus - Mr. Advisor (UNI00) application. The top menu includes options like FICHIER, EDITION, MODULES, RAPPORTS, OUTILS, RECHERCHE, T-ONE, and UTILISATEURS. The left sidebar contains navigation icons for Tableau de bord, Modèles, Relations, Clients, Comptes, Portefeuille, Transactions, Titres, and Ordres. The main area is titled 'Clients' and shows a table with columns for Nom, No client, Code d..., Téléphone 1, Téléphone 2, Solde, Devise, Marge, Dernier Contact, Âge, Téléphone 3, and Téléphone. A red circle highlights search and filter icons in the top toolbar. A red arrow points to the active search criterion 'Clients qui auront 71 ans au cours de l'année' in the table header. The bottom section shows details for 'BELKADI AMADOU' with tabs for Info, Agenda, Produits & services, Profil, and Documents. The details include address, telephone numbers, and financial information.

Nom	No client	Code d...	Téléphone 1	Téléphone 2	Solde	Devise	Marge	Dernier Contact	Âge	Téléphone 3	Téléphone
✓ BELKADI AMADOU	00-002I	BD88	(450) 555-2608	(450) 555-7499	295 220,39	CAD	n/d		82		
✓ GIROUARD SABRINA	00-001F	BD77	(450) 555-4653	(450) 555-7489	289 924,03	CAD	n/d		82		
✓ LEGAULT PIERRE	00-002H	BD88	(450) 555-8495	(450) 555-8653	211 101,38	CAD	267 149,86		82		
✓ LEVY WILLIAM	00-001S	BD77	(450) 555-2897	(450) 555-4537	897 556,29	CAD	623 478,93		82	(450) 555-8782	
✓ WALKER ANDREI	00-0034	BD88	(450) 555-3560	(450) 555-6360	283 249,13	CAD	n/d		82		

Details for BELKADI AMADOU (00-002I):

- Comptes: BELKADI AMADOU 00-002I-A, BELKADI AMADOU 00-002I-T
- Relations: SMITH FAMILY 00000
- AMADOU BELKADI
- 639 NORFOLK PARK, BANFF, AB, D7I 4F7
- telephone:1: (450) 555-2608, telephone:2: (450) 555-7499
- Montants (CAD): Solde: 295 220,39, Valeur totale: 525 810,26, Marge: n/d

Bottom status bar: BELKADI AMADOU, 00-002I, (450) 555-2608, (450) 555-7499 | 0 0 0 5 | 2014/05/30

Figure 2.1 Page-écran principale de l'application Croesus Conseiller. Le module Clients est actif. Les icônes permettant de gérer et créer des critères de recherche sont encerclées en rouge. Le nom du critère de recherche actif sur le module est indiqué par une flèche rouge. (Les données présentées sont fictives).

La figure 2.2 présente la page-écran qui permet de faire la gestion des critères de recherche sauvegardés. On peut voir qu'il y a deux boutons pour ajouter un critère : le premier pour l'interface de critère de recherche simplifiée et le deuxième pour l'interface de critère de recherche avancée. Par la suite, les boutons permettent de faire la modification, la copie ou la suppression d'un critère de recherche existant. Les boutons qui se trouvent au bas de la page permettent d'appliquer le résultat du critère de recherche à la grille du module associé.

Nom	Accès	Type	Création	Module	Modifié	Généré
▶ Clients: Écart entre obj. plac. et port...	Global	Critère	Global	Clients	2007/05/25	2007/10
Clients n'ayant pas effectué de dépôt	Global	Critère	Global	Clients	2004/10/08	
Clients qui auront 71 ans au cours d...	Global	Critère	Global	Clients	2007/10/19	2007/10
Liste des clients Quebec	Global	Critère	Global	Clients	2006/04/04	2008/11
Compliance with Investment knowle...	Utilisateur	Critère	Mr. Advisor	Clients	2008/02/05	2008/03
Clients contacted prior to Dec 31 2005	Utilisateur	Critère	Mr. Advisor	Clients	2008/02/05	2008/02
Clients ayant un rendez-vous prévu	Global	Gabarit	Global	Clients	2008/04/24	
Clients à contacter bientôt	Global	Gabarit	Global	Clients	2008/04/24	
Retour sur investissement (Clients)	Global	Gabarit	Global	Clients	2013/11/15	
Clients nés apres une date spécifiqu...	Global	Gabarit	Global	Clients	2013/07/31	
Comptes détenant un des deux titres	Global	Critère	Global	Comptes	2004/10/08	2007/04
Comptes ne détenant pas un des de...	Global	Critère	Global	Comptes	2004/10/08	2008/01
Comptes détenant plus de 10% de li...	Global	Critère	Global	Comptes	2008/11/05	2007/10
Comptes: Solde de liquidation	Global	Critère	Global	Comptes	2008/03/24	2007/05
Comm. cumulées	Utilisateur	Critère	Mr. Advisor	Comptes	2007/08/02	2007/08
Clients > 1,000,000	Utilisateur	Critère	Mr. Advisor	Comptes	2007/12/14	2007/12

Figure 2.2 Page-écran permettant de faire la gestion des critères de recherche sauvegardés.

La figure 2.3 présente la page-écran principale de l'application avec le module tableau de bord. Chaque encadré présente une liste d'éléments correspondant au résultat d'un critère de recherche. Ceux-ci sont exécutés chaque nuit et permettent aux utilisateurs d'être avisés le matin lorsqu'une situation anormale s'est produite.

The screenshot shows the main dashboard of the Croesus - Mr. Advisor application. The interface is organized into a sidebar on the left and a main content area on the right. The sidebar contains icons for various modules: Tableau de bord, Modèles, Relations, Clients, Comptes, Portefeuille, Transactions, Titres, and Ordres. The main content area displays several data tables and widgets. The top table is titled ".Default Clients List" with a total of 27 records. It lists client information including No du client, Nom, Téléphone 1, and Valeur totale. Below it is a table for "Accounts with RY" with 37 records, showing details like No du client, Nom, Téléphone 1, Valeur totale, Solde, Devise, Code d..., and Assignée à. Other sections include "Sommaire de l'encaisse négative" (Total: 1), "Sommaire de l'encaisse positive" (Total: 50 +), "Objectifs de placement - écarts - Modèles" (Total: 0), and "Objectifs de placement - écarts" (Total: 0). A red message at the bottom indicates that the search results are empty: "Votre recherche ne contient aucun résultat. Tentez d'affiner votre recherche et régénérer." A calendar widget for "déc.-18" is also visible in the top right corner.

Figure 2.3 Page-écran principale de l'application Croesus Conseiller. Le module Tableau de bord est actif. Les encadrés présentent la liste de résultats d'un critère de recherche qui est exécuté durant la nuit. (Les données présentées sont fictives).

Les utilisateurs utilisent les résultats des différentes requêtes à des fins variées. Par exemple, un utilisateur pourrait obtenir la liste des clients avec lesquels il n'a eu aucun rendez-vous durant l'année. L'objectif serait alors de les contacter pour organiser une rencontre avec eux.

Une demande fréquente des utilisateurs pour la création de requêtes est d'obtenir la liste des clients qui auront 71 ans durant l'année en cours et qui détiennent un compte de type RÉER et qui n'ont

pas effectué de transaction de cotisation durant l'année. L'objectif serait de les inciter à maximiser leurs placements RÉER avant de devoir transférer ceux-ci en FER.

Un autre exemple porte sur le cas d'un employé de l'équipe de conformité qui veut identifier les portefeuilles qui ne respectent pas l'objectif de placement défini pour un client. L'objectif serait de faire le suivi avec le conseiller pour s'assurer que des corrections sont effectuées rapidement au portefeuille.

Une fois que le critère de recherche est exécuté, on obtient une liste de valeurs qui peut être utilisée de plusieurs façons. Il est entre autre possible de produire une série de rapports, d'envoyer des courriels de façon massive ou d'exporter le résultat dans un fichier MS Excel.

Les demandes varient d'un utilisateur à l'autre et d'une institution financière à l'autre. Il ne serait pas possible de satisfaire tous les besoins en matière de requête à l'aide de commandes préencodées et paramétrées (par exemple à l'aide de formulaire).

## 2.2 Définition des concepts

Le tableau 2.1 présente un glossaire du vocabulaire utilisé pour identifier les concepts liés aux requêtes qui seront utilisés tout au long du document.

Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document

<p>Critère de recherche</p>	<p>Dans le produit Croesus conseiller, le terme « critère de recherche » est utilisé pour identifier les requêtes qui sont créées par les utilisateurs pour consulter les données disponibles dans la base de données.</p> <p>La requête suivante comprend un critère de recherche simplifié (Liste de clients) composé de deux conditions simples.</p> <p>Exemple de critère simple :</p> <p>« Liste des clients ayant 71 ans et plus et possédant un compte de type RÉER »</p> <p>Le critère est composé des deux conditions simples suivantes</p> <p>« ayant 71 ans »</p> <p>« possédant un compte de type RÉER ».</p>
-----------------------------	---



Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document (suite)

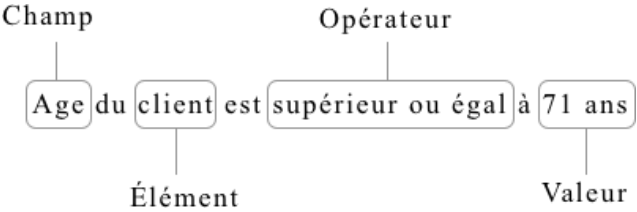
Condition	<p>Composante de base d'une requête. Elle permet de raffiner la liste d'éléments. Tous les éléments de la liste doivent répondre aux conditions de la requête pour faire partie du résultat. La condition est composée d'un champ provenant d'un élément, d'un opérateur et d'une (ou plusieurs) valeur(s).</p> <div style="text-align: center;">  <p>The diagram shows the sentence "Age du client est supérieur ou égal à 71 ans" with boxes around the words "Age", "client", "est supérieur ou égal à", and "71 ans". Lines connect these boxes to labels above or below: "Champ" above "Age", "Opérateur" above "est supérieur ou égal à", "Élément" below "client", and "Valeur" below "71 ans".</p> </div> <p>Figure 2.4 Exemple de condition simple composée d'un champ (associé à un élément) d'un opérateur et d'une valeur.</p>
Élément	<p>Entité qui regroupe plusieurs attributs (ou champs) qui permettent de l'identifier. Par exemple le terme « Client » correspond à une entité. Le client dont le nom est Tremblay et dont la date de naissance est le 3 janvier 1952 correspond à un élément de type Client.</p>
Entité	<p>Type de valeurs utilisé pour la condition et retourné dans les résultats. Par exemple, client, compte, relation, transactions, titres.</p>

Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document (suite)

<p>Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document (suite) Groupement</p>	<p>Liste de plusieurs conditions séparées par un opérateur logique qui doivent être traitées en priorité lors de l'exécution de la requête. Dans l'exemple suivant, un premier groupement est effectué sur la couleur des cheveux. Un deuxième groupement est effectué sur la couleur des yeux.</p> <p>Exemple 1 :</p> <p>Personnes ayant (les cheveux blonds OU les cheveux bruns) ET (les yeux bleus OU les yeux verts)</p> <p>Il est possible de créer une hiérarchie à plusieurs niveaux avec les groupes.</p> <p>Exemple 2 :</p> <p>Liste des personnes qui (portent des lunettes) ET (qui ont les yeux verts OU les yeux bleus) ET (     ( qui habitent dans la ville de Laval OU dans la ville de Montréal ).     ET     ( qui sont propriétaire d'un chalet ) )</p> <p>Dans cet exemple, les groupements sont représentés par des parenthèses.</p>
<p>Interface de critères de recherche avancée</p>	<p>Interface offerte aux utilisateurs de l'application Croesus Conseiller permettant de formuler des requêtes simples ou complexes. Cette interface est basé sur un formulaire flexible par filtre qui se présentent sous forme hiérarchique. Elle permet de formuler des requêtes plus élaborées sur le plan sémantique que l'interface de critères de recherche simplifiée.</p>

Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document (suite)

Interface de critères de recherche simplifiée	de de Interface offerte aux utilisateurs de l'application Croesus Conseiller permettant de formuler des requêtes simples ou complexes. Cette interface se base sur un système par filtres qui se présentent sous la forme de phrases en langage semi-naturel rigide. Elle est jugée plus simple à utiliser que l'interface avancée. (La facilité d'utilisation de l'interface n'a jamais été évaluée auprès des utilisateurs avant la présente recherche).
Interface de filtres sur une liste d'information	de Portion d'interface associée à des listes d'éléments offerte aux utilisateurs de l'application Croesus Conseiller permettant de filtrer les éléments de la liste. Cette interface permet de faire des requêtes simples uniquement.
Jointure	C'est un groupe de conditions qui s'appliquent à une entité qui est liée à l'entité de la liste des éléments qui font partie du résultat de la requête. La jointure est utilisée lorsqu'il existe une relation zéro (ou un) à plusieurs (0:N ou 1:N) entre l'entité de la liste d'éléments et l'entité liée. Elle permet de valider l'existence ou non d'éléments liés répondant à un certain nombre de conditions.  Par exemple : Liste de clients <b>ayant au moins trois</b> comptes de type RÉER.
Jointure positive	Exemple : si on veut obtenir la liste des <b>clients</b> qui <b>possèdent au moins un compte</b> et dont le(s) compte(s) répondent à n conditions, il est nécessaire de créer une jointure entre l'élément client et l'élément compte.  Exemple : liste des clients qui <b>ont au moins un</b> compte dont le type est RÉER.
Jointure négative	Exemple : si on veut obtenir la liste des <b>clients</b> qui <b>ne possèdent aucun compte</b> répondant à n conditions, il est nécessaire de créer une jointure négative entre l'entité client et l'entité compte.  Exemple : liste des clients <b>qui ne possèdent aucun</b> compte dont le type est RÉER.

Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document (suite)

<p>Liste d'éléments</p>	<p>C'est l'ensemble des éléments qui seront retournés comme résultat de la requête. La liste peut contenir zéro, un ou plusieurs éléments,</p> <p>par ex. la «Liste des clients ayant 71 ans et plus et possédant un compte de type RÉER ».</p> <p>Dans cet exemple, la liste d'éléments retournée dans le résultat sera composée de clients. Si aucun filtre et aucun critère de recherche n'est appliqué sur la liste d'éléments, la liste entière est affichée. Les entités supportées pour la liste d'éléments sont : clients, comptes, relations, titres, ordres, modèles.</p>
<p>Opérateur</p>	<p>Opérateur mathématique qui permet de comparer la valeur contenue dans le champ de l'élément avec la valeur spécifiée dans la condition. Par exemple, pour les valeurs numériques : =, &gt;=, &gt;, &lt;=, &lt;, &lt;&gt;, etc. Ex. : valeur totale &gt; 0,00 \$.</p> <p>Pour les valeurs alphanumériques : contient, ne contient pas, égale, n'égale pas, commence par, ne commence pas par, etc.</p> <p>Pour les valeurs date : ultérieure à, ultérieure ou égale à, antérieure à, antérieure ou égale à, dans les derniers x jours, etc.</p>
<p>Opérateur logique ET/OU</p>	<p>Composante de la requête qui permet de créer un lien entre les différentes conditions. Il est toujours nécessaire de spécifier un opérateur logique entre chaque condition ou groupement.</p> <p>ET : l'élément doit répondre à toutes les conditions pour faire partie du résultat.</p> <p>OU : l'élément doit répondre à au moins une des conditions pour faire partie du résultat.</p>
<p>Requête</p>	<p>Le présent document s'intéresse seulement aux requêtes qui permettent de récupérer de l'information préalablement sauvegardée dans la base de données. Les requêtes qui permettent de modifier des données existantes, d'effacer des données ou de créer de nouvelles données ne sont pas traitées par la recherche.</p>

Tableau 2.1 Glossaire du vocabulaire utilisé pour identifier les concepts utilisés dans le document (suite et fin)

<p>Requête complexe (ou critère de recherche complexe)</p>	<p>Requête composée de plusieurs conditions séparées par des opérateurs logiques ET et/ou OU ou dont les champs utilisés proviennent de plusieurs entités différentes de l'entité retournée dans la liste d'éléments du résultat.</p> <p>Par exemple : « Liste des clients qui possèdent au moins un compte dont le type de compte est REER ».</p> <p>Dans cet exemple, le type de compte est un champ appartenant à l'entité « compte » or, la liste d'éléments retournée comme résultat contient des clients. L'entité de la condition n'est donc pas la même que l'entité de la liste d'éléments du résultat.</p>
<p>Requête simple (ou critère de recherche simple)</p>	<p>Requête composée d'une ou de plusieurs conditions séparées par des opérateurs logiques ET et dont les conditions sont basées sur des champs provenant de l'élément retourné dans la liste d'éléments du résultat.</p>
<p>Schéma logique</p>	<p>Définition des entités et des champs disponibles pour créer un critère de recherche ainsi que des liens qui unissent les différentes entités entre elles. Cette notion est similaire au modèle de tables dans une base de données.</p>
<p>Valeur</p>	<p>Valeur limite qui sert de point de comparaison avec la valeur contenue dans le champ de l'élément.</p>

Le tableau 2.2 présente les différentes capacités offertes pour la formulation de requêtes dans les différentes interfaces disponibles actuellement pour les utilisateurs de Croesus Conseiller.

Tableau 2.2 Synthèse des capacités offertes pour la formulation de requêtes selon l'interface disponible.

Capacité pour la formulation de requêtes	Filtres sur liste d'éléments	Critères de recherche simplifiée	Critères de recherche avancée
Choix disponibles pour l'élément retourné par la requête	Clients, comptes et relations Historique de prix du titre, Historique de dividendes	Clients, comptes, relations, titres, modèles et ordres	Clients, comptes, relations et titres
Condition simple sur les champs de l'entité retournée	✓	✓	✓
Opérateur ET	✓	✓	✓
Opérateur OU	X	✓	✓
Un niveau de jointures	X	✓	✓
Plusieurs niveaux de jointures	X	(maximum 2, parfois)	✓
Conditions sur des champs d'une entité différente de l'entité retournée	X	✓	✓
Choix de jointures disponibles	X	Vaste choix	Choix limité
Agrégations (somme, moyenne, etc.)	X	X	!
Groupement de conditions	X	✓	✓
Inversion de condition	X	X	✓

✓ = supporté

! = partiellement supporté

X = non supporté

## 2.3 Problématiques rencontrées

Le Centre de services aux clients de l'entreprise Croesus Finansoft reçoit beaucoup de demandes d'assistance liées à l'utilisation des interfaces de formulation de requêtes. Des statistiques sur le nombre d'appels et le nombre d'employés affectés à cette tâche seront disponibles dans la section 2.4. Les questions et les plaintes portent sur différents problèmes qui seront présentés dans la présente recherche.

L'une des trois interfaces actuelles, l'interface de critères de recherches simplifiée, utilise un langage semi-naturel sous forme de phrases structurées ayant un sujet, un verbe, un complément. La deuxième interface, l'interface critères de recherches avancée, affiche l'information sous forme hiérarchique. La troisième, la fonctionnalité des filtres sur une liste d'éléments, permet de définir des conditions simples séparées par des opérateurs logiques «ET» seulement. On examine ces trois interfaces dans les pages qui suivent.

### 2.3.1 Problématiques liées à l'interface de création de critères de recherches simplifiée

La complexité du langage semi-naturel rigide utilisé dans l'interface de critères de recherches simplifiée amène les utilisateurs à souvent demander de l'assistance pour la formulation de requêtes. En effet, la notion de jointure est très difficile à comprendre pour les utilisateurs qui ne sont pas familiarisés avec le langage SQL, or l'interface force l'utilisateur à choisir une jointure chaque fois qu'il ajoute une condition. Cette étape crée beaucoup de confusion.

De plus, le système de requêtes permet de créer des jointures entre différents éléments du schéma logique. L'interface de critères de recherches simplifiée permet de définir seulement un niveau de jointures dans une requête et cela limite les utilisateurs dans la formulation de requêtes plus complexes. Par ailleurs, après avoir formulé la requête, il est difficile de distinguer les conditions qui font partie de la jointure de celles qui sont liées à l'élément de la requête. Cette difficulté découle du fait qu'il n'y a pas d'élément d'affichage visuel qui permet de distinguer les conditions qui font partie de la jointure par rapport à celles qui sont liées à l'élément principal de la requête.

Par ailleurs, pour créer des groupements de conditions, l'utilisateur doit ajouter des parenthèses. Il est difficile de savoir dans quel cas et à quel endroit les parenthèses doivent être ajoutées. Par ailleurs, les parenthèses sont difficiles à distinguer parce qu'elles utilisent un contraste de couleurs

trop faible. De plus, la zone où l'utilisateur doit cliquer pour ajouter les parenthèses est très petite. Cela ne respecte pas les normes d'accessibilité.

La figure 2.5 présente la page-écran de l'interface de critères de recherche simplifiée dans son état initial. C'est-à-dire lorsqu'elle est vide, au début de la formulation du critère de recherche. Dans cet exemple, on peut voir que la liste de résultats attendue est une liste de comptes grâce au champ « module ». Le grand encadré blanc nommé : « Définition » est destiné à la formulation de la requête désirée.

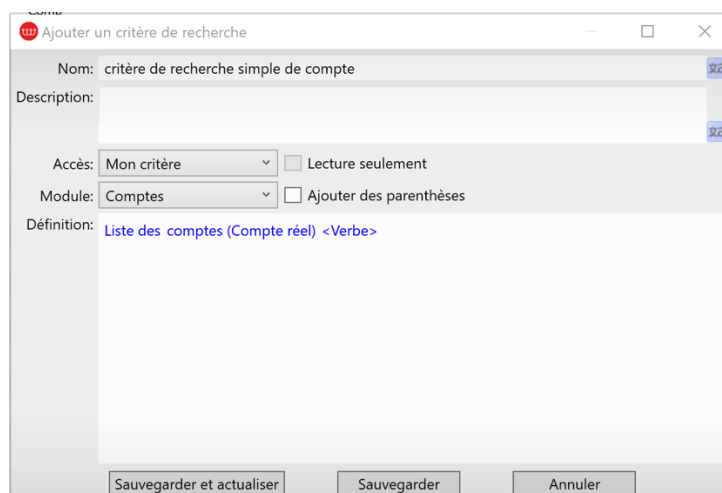


Figure 2.5 Page-écran de l'interface de formulation des critères de recherche simplifiée lors de la création d'une nouvelle requête. Les éléments textuels affichés en bleu sont des hyperliens donnant accès à des listes de choix. L'exemple présente un critère de recherche dont le résultat attendu est une liste de comptes.

La figure 2.6 montre une succession d'images qui présentent la séquence des actions à effectuer par l'utilisateur pour la formulation de la requête. À chaque étape, l'utilisateur doit actionner l'hyperlien pour obtenir une liste de choix lui permettant de spécifier les différentes composantes de la requête tels que la jointure, le champ, l'opérateur, la valeur et l'opérateur logique. Pour que la requête soit complète, elle doit se terminer par un « . ».





Figure 2.6 Séquence des étapes à effectuer par l'utilisateur pour formuler la requête. L'utilisateur doit actionner les hyperliens pour obtenir une liste de choix lui permettant de composer la requête à l'aide de jointures, champs, opérateurs, valeurs et opérateurs logiques.

La figure 2.7 présente un exemple de page-écran de l'interface de critères de recherches simplifiée portant sur une requête complète. On peut voir dans l'exemple que la requête retourne une liste de comptes. Dans la liste de conditions, il y a une jointure négative marquée par la mention « n'ayant pas » et une jointure avec l'objet « transaction » grâce à la mention « n'ayant pas de transaction avec ». Les trois conditions qui suivent la mention « n'ayant pas de transaction avec » s'appliquent à l'élément transaction. Elles font toutes partie de la même jointure entre compte et transaction.

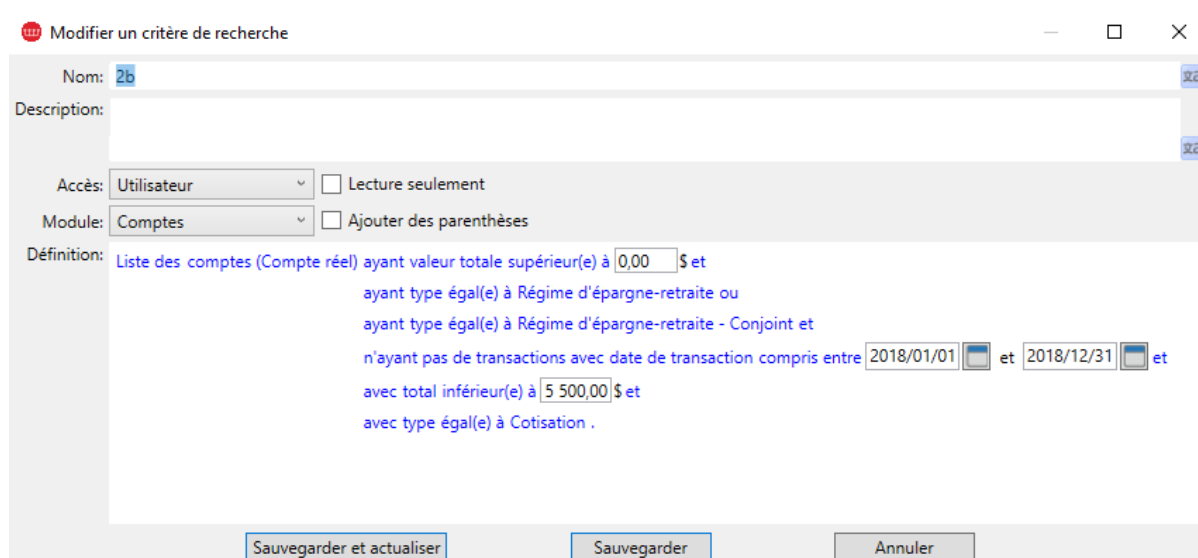


Figure 2.7 Page-écran de l'interface de critères de recherche simplifiée.

### 2.3.2 Problématiques liées à l'interface de création de critères de recherches avancée

L'interface de critères de recherches avancée permet de construire des requêtes complexes avec plusieurs jointures et plusieurs groupements de conditions. Cette interface affiche beaucoup de boutons et de contrôles. L'ajout de conditions nécessite plusieurs opérations peu intuitives. Les utilisateurs sont intimidés par la complexité et par le nombre d'étapes nécessaires pour la construction d'une requête. L'utilisation de cette interface nécessite une formation. Par ailleurs, la création de critères de recherche est une tâche qui est accomplie sporadiquement par les utilisateurs. Ces derniers ont donc le temps d'oublier le fonctionnement de l'interface entre chaque utilisation. De plus, ils se plaignent que le fonctionnement est contre-intuitif.

D'autre part, le mode d'affichage de la requête sous forme de longues phrases rapprochées crée une densité de l'information qui rend les utilisateurs inconfortables.

La création de jointures rend nécessaire la création d'un niveau hiérarchique dans la structure de la requête. Cette étape est souvent omise par les utilisateurs. Cela a un effet négatif sur le résultat retourné lors de l'exécution de la requête.

Par ailleurs, l'ajout d'opérateurs logiques est peu intuitif. Il est difficile de s'assurer que les conditions font bien partie de l'opérateur logique désiré.

La figure 2.8 présente une page-écran de l'interface des critères de recherche avancée à son état initial, c'est-à-dire lorsque vide, lors de la création d'un nouveau critère de recherche. On peut voir que le résultat attendu est composé d'une liste de comptes grâce au champ « recherche des : ».

Figure 2.8 Page-écran de l'interface de critères de recherche avancée à son état initial. On peut voir que la liste des résultats attendus est une liste de comptes.

La figure 2.9 montre une succession d'images qui présentent la séquence des actions à effectuer par l'utilisateur pour la formulation de la requête. La création d'une condition simple (champ, opérateur, valeur) est effectuée dans la section du bas « condition ». Une fois la condition formulée, elle doit être ajoutée à l'encadré principal à l'aide du bouton « ajouter ». Pour l'ajout d'opérateurs logiques et de jointures, il faut utiliser le bouton ajouter qui se trouve à la gauche de l'encadré blanc.

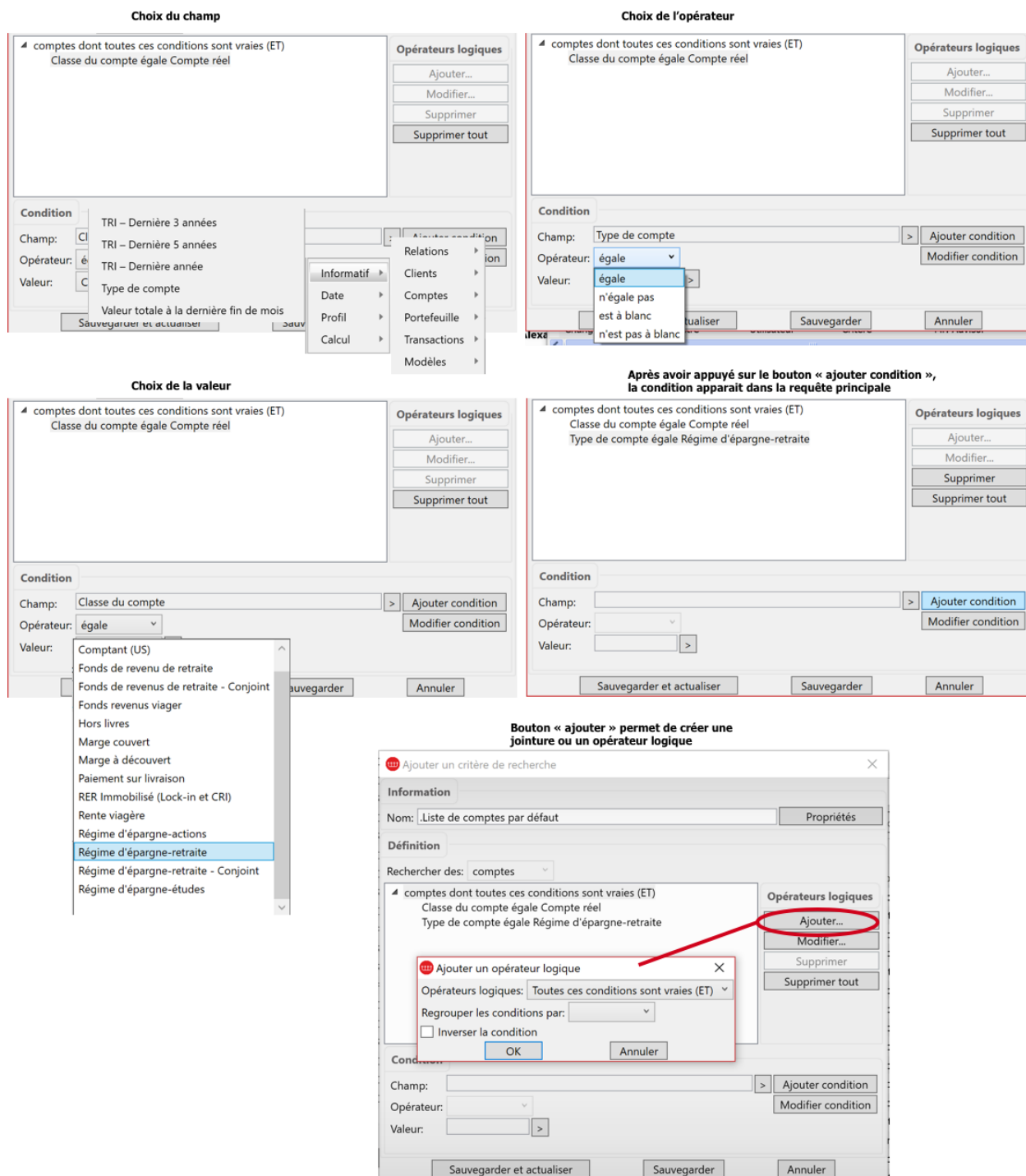


Figure 2.9 Séquence des étapes à effectuer par l'utilisateur pour formuler la requête. La création et l'ajout d'une condition simple est effectuée dans la section tout en bas de l'interface. Par la suite, l'ajout de jointures et d'opérateurs logiques est effectué à l'aide du bouton ajouter situé à la droite de la requête.

La figure 2.10 présente une capture d'écran qui montre un exemple de l'interface de critère de recherche avancée. On peut voir que la liste d'éléments désirée est composée de clients. Il y a une jointure négative correspondant à la ligne «excluant celles/ceux dont toutes ces conditions sont vraies (NON ET)». Il y a un groupe de deux conditions avec l'opérateur logique OU à la ligne «dont au moins une des conditions est vraie (OU)».

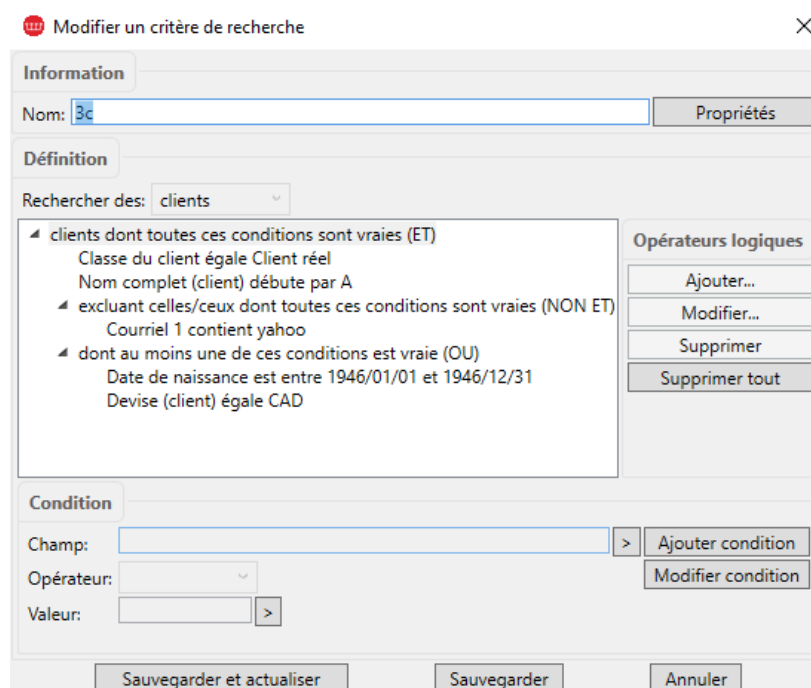


Figure 2.10 Page-écran de l'interface de critères de recherche avancée.

### 2.3.3 Problématiques de la fonctionnalité des filtres sur une liste d'information

La fonctionnalité des filtres sur une liste d'information est encore peu connue des utilisateurs puisqu'elle a été ajoutée à l'application Croesus Conseiller très récemment. Cette fonctionnalité permet de filtrer la liste, une condition à la fois. Chaque condition permet de raffiner la liste affichée. Les conditions ne permettent pas les jointures ni l'opérateur logique «OU». Cette fonctionnalité est simple à utiliser, facile à lire, mais offre peu de possibilités pour les requêtes plus complexes et une liste de champs plus limitée pour la formulation de conditions.

La figure 2.11 présente la fonctionnalité de filtres sur une liste d'éléments. Les filtres, de couleur orangée, sont situés dans la partie supérieure gauche de la grille. Toutes les conditions sont traitées avec l'opérateur ET entre chaque condition.

Figure 2.11 shows a screenshot of a software application window titled "Croesus - Mr. Advisor (UNI00)". The interface displays a list of accounts under the heading "Comptes". The list is filtered with several criteria: "Valeur totale >= 0", "Devise = CAD", and "Type = Régime d'épargne-retraite". The filters are highlighted in orange. The table below shows the resulting list of accounts with their respective details.

Nom	No compte	Code d...	Type	Régime	Téléphone 1	Téléphone 2	Solde	Devise
ALEX TEST	~E-0000C-1	BD88	Régl...				14 388,00	CAD
ARCAND ANNIE	00-0015-S	BD77	Régl...	0100...	(450) 555-2150	(450) 555-2483	65 357,85	CAD
ARCHAMBAULT CHAN	00-000P-S	BD67	Régl...	0100...	(450) 555-0069	(450) 555-3233	26 371,90	CAD
AZAR DIANE	00-0008-S	BD67	Régl...	0100...	(450) 555-1648	(450) 555-5094	100 000,00	CAD
BÉLANGER CHARLOT	00-000V-S	BD67	Régl...	0100...	(450) 555-8035	(450) 555-6789	103 427,88	CAD
BELLEY LUCIANO	00-0020-S	BD88	Régl...	0100...	(450) 555-7012	(450) 555-8957	26 941,32	CAD
BLOUIN CLAUDE	00-0025-S	BD88	Régl...	0100...	(450) 555-8762	(450) 555-3780	61 120,27	CAD
CHIASSON ESTELA	00-000M-S	BD67	Régl...	0100...	(450) 555-2377	(450) 555-0726	100 000,00	CAD
D'ERRICO NORMAND	00-0025-S	BD88	Régl...	0100...	(450) 555-7675	(450) 555-3340	673 428,96	CAD
DEAKIN JOSIE	00-000W-S	BD67	Régl...	0100...	(450) 555-0545	(450) 555-0882	98 500,99	CAD
DESROCHES PAUL	00-0027-S	BD88	Régl...	0100...	(450) 555-6754	(450) 555-6754	215 022,54	CAD
DUFORT PIERRETTE	00-000R-S	BD67	Régl...	0100...	(450) 555-5019	(450) 555-4417	148 344,35	CAD
DUSSAULT STÉPHAN	00-002X-S	BD88	Régl...	0100...	(450) 555-8640	(450) 555-2231	130 029,67	CAD
DUTRISAC ISABELL	00-0009-S	BD67	Régl...	0100...	(450) 555-3548	(450) 555-6394	100 000,00	CAD
EDWARDS RANDY	00-0028-S	BD88	Régl...	0100...	(450) 555-0069	(450) 555-9094	439 637,94	CAD
GIROUARD SABRINA	00-001F-S	BD77	Régl...	0100...	(450) 555-4653	(450) 555-7489	117 119,74	CAD
JAMES ALBERT	00-1234-S	BD77	Régl...		(780) 555-6370	(780) 555-1374	40 000,00	CAD
JAMES LISA	00-5678-S	BD77	Régl...		(780) 555-6370	(780) 555-1374	0,00	CAD
LABRECQUE FRÉDÉR	00-0029-S	BD88	Régl...	0100...	(450) 555-2959	(450) 555-4207	32 014,81	CAD
LACROIX SUZANNE	00-001R-S	BD77	Régl...	0130...	(450) 555-3515	(450) 555-6263	99 535,96	CAD
LALANDE RACHELLE	00-001P-S	BD77	Régl...	0100...	(450) 555-2701	(450) 555-3189	23 513,14	CAD
LAMOUREUX EMILE	00-0031-S	BD88	Régl...	0100...	(450) 555-2399	(450) 555-9438	83 380,07	CAD
LANDRY ANGÈLE	00-000T-S	BD67	Régl...	0100...	(450) 555-7897	(450) 555-7448	100 417,70	CAD

Figure 2.11 Page-écran de fonctionnalité des filtres sur une liste de comptes. Les filtres de couleur orangée sont en haut à gauche de la liste. (N.B. Les données présentées sont fictives).

### 2.3.4 Problématiques liées à l'utilisation de l'opérateur logique OU

Les utilisateurs de systèmes de consultation de base de données en général ont de la difficulté à déterminer si l'opérateur logique à utiliser doit être le ET ou le OU. Ces notions calquées sur les requêtes en langage SQL sont difficiles à comprendre parce que le sens des conjonctions en langage naturel diffère de la logique des opérateurs SQL. Dans les exemples suivants, en langage naturel, on peut voir que les conjonctions ET et OU sont interchangeables.

Exemple :

« Listes des comptes RÉER, CELI **ET** FER »

« Listes des comptes RÉER, CELI **OU** FER »

Dans l'interface utilisateur, la requête doit être définie avec l'opérateur logique OU comme suit :

« Listes des comptes dont le type est RÉER **OU** dont le type est CELI **OU** dont le type est FER ».

Avec un opérateur logique ET, aucun résultat ne serait retourné puisqu'un type de compte ne peut

pas avoir plus d'une valeur de façon simultanée. L'utilisation de l'opérateur logique OU est offerte dans les interfaces de critères de recherche simplifiée et avancée.

### **2.3.5 Problématiques liées à la confusion entre les différentes interfaces de formulation de requêtes**

Considérant que la fonctionnalité de formulation des critères de recherches est complexe, les utilisateurs ne veulent pas apprendre à utiliser plusieurs interfaces de critères de recherches. Ils choisissent une interface parmi celles qui sont offertes. Or, chacune des interfaces offre une liste de champs et une liste d'opérateurs et a différentes capacités. Il est donc parfois nécessaire, pour un même utilisateur, d'apprendre à utiliser les deux interfaces pour arriver à construire tous les critères de recherche désirés. Comme les utilisateurs sont souvent pressés d'obtenir le résultat désiré, ils ne veulent pas avoir à faire un choix entre les deux interfaces de critères de recherche. De plus, comme les capacités des différentes interfaces sont variables, il existe des situations où la requête souhaitée est impossible à formuler parce qu'elle nécessiterait de jumeler les capacités des différentes interfaces, ce qui est parfois impossible.

Par ailleurs, il n'est pas possible de transférer une requête qui est construite dans une interface vers une autre. Si l'utilisateur tente de créer une requête sans succès en raison de la limite de l'interface et qu'il désire transférer la requête vers l'autre interface pour poursuivre la formulation, il doit reconstruire la requête entièrement manuellement à chaque fois. Cela provoque du découragement chez les utilisateurs.

### **2.3.6 Conclusion sur les problématiques**

Toutes ces problématiques engendrent de la frustration et une surcharge de demandes de support au Centre de services aux clients de l'entreprise. Très souvent, les utilisateurs ne veulent pas apprendre à utiliser la fonctionnalité de façon autonome, ils désirent avoir un service qui crée les requêtes pour eux parce que la fonctionnalité est trop complexe. C'est ce qui les pousse à faire appel au Centre de services aux clients. Cette surcharge a un coût qui est difficile à calculer, mais qui est important et doit être réduite. Plusieurs utilisateurs préfèrent exporter des données dans un fichier MS Excel et utiliser les outils de recherches offerts par cette application pour effectuer leurs recherches.

## 2.4 Statistiques de l'entreprise

Nous avons choisi de nous intéresser à la fonctionnalité de création de requêtes de l'application principale de Croesus Conseiller parce que nous avons eu à travailler sur cette interface à plusieurs reprises dans le passé, sans jamais obtenir de résultats satisfaisants pour les utilisateurs. Les fonctionnalités offertes existent depuis plusieurs années : l'interface de critères de recherche simplifiée actuelle est disponible pour les utilisateurs depuis environ 10 ans. L'interface de critères de recherche avancée est disponible depuis environ 15 ans. La fonctionnalité de filtre sur une liste est disponible depuis 2017 pour un nombre limité d'utilisateurs.

Les deux interfaces de critères de recherches sont disponibles pour tous les utilisateurs, ce qui représente environ 14 000 utilisateurs. On estime à 7000 le nombre d'utilisateurs qui ont utilisé l'une ou l'autre des interfaces pour créer ou modifier une requête durant l'année 2017. Cela représente 50 % des utilisateurs. Généralement, les utilisateurs construisent une requête une fois et l'exécutent à répétition par la suite pour actualiser le résultat. Une requête a donc une durée de vie assez longue.

Les statistiques d'utilisation de la fonctionnalité des critères de recherches démontrent que seulement 30 % des requêtes ont été créées avec l'interface de critères de recherche avancée. 70 % des requêtes ont été créées avec l'interface de critères de recherche simplifiée. Toutefois, les utilisateurs ont la capacité de partager des critères de recherche entre eux. Autrement dit, un utilisateur crée le critère de recherche et par la suite, plusieurs utilisateurs peuvent générer le résultat dans leur environnement de travail. En 2017, les utilisateurs ont généré 50 % de critères de recherche simplifiés et 50 % de critères de recherche avancés malgré le fait qu'il existe plus de critères de recherche simplifiés que de critères de recherche avancés dans la base de données. Cela laisse supposer que les critères de recherche avancés sont plus souvent partagés entre les utilisateurs que les critères de recherche simplifiés. L'interface de filtre sur une liste de valeurs est apparue récemment. Nous n'avons donc pas encore de donnée statistique au sujet de cette interface.

Le centre de services aux clients de l'entreprise Croesus Finansoft comprend les équipes suivantes :

- Une équipe de deux formateurs à temps plein pour les différentes fonctionnalités du produit. Les formations offertes sont : Formation encadrée, Formation ad hoc, Webinaires. Il existe quatre formations webinaires (deux en anglais, deux en français) qui portent sur le sujet des



interfaces de formulation de requêtes à la base de données sur un total de 22 (11 en anglais, 11 en français). Pour un total de 27 300 téléchargements de webinaires en 2017, 8705 portaient sur les fonctionnalités de requêtes. Cela représente environ 32 % des téléchargements. L'équipe de formation de l'entreprise Croesus Finansoft s'affaire présentement à produire un autre webinaire pour expliquer la fonctionnalité des filtres sur des listes.

- Une équipe de 15 employés (un directeur, un chef d'équipe, neuf analystes de données et d'application à temps plein) pour répondre aux questions et assister les utilisateurs dans leurs tâches pour les différents produits et fonctionnalités offertes par l'entreprise Croesus Finansoft. L'équipe de services aux clients a reçu 19 300 appels en 2017 et la tendance est à la hausse. On estime à 889 le nombre des appels au sujet des outils de requêtes sur 19000 appels en 2017. Cela représente 4,7 % des appels.

En 2016 un sondage a été effectué auprès d'un certain nombre d'utilisateurs. Sur un total de 608 commentaires, 36 commentaires (6 %) portaient sur le sujet de la fonctionnalité de requêtes.

Ce chapitre a permis de résumer les problèmes rencontrés par les utilisateurs lors de la création de requêtes à la base de données dans le logiciel Croesus Conseiller et les efforts déployés par l'entreprise pour venir en aide aux utilisateurs. Une revue de la littérature a été effectuée pour savoir si les problèmes rencontrés par l'entreprise Croesus Finansoft ont déjà fait l'objet de recherches. C'est ce dont traite le prochain chapitre.

## **CHAPITRE 3 REVUE DE LITTÉRATURE SUR LES APPROCHES DE CONCEPTION DES REQUÊTES À DES BASES DE DONNÉES ET HYPOTHÈSES**

Dans le passé, pour consulter des systèmes de bases de données relationnelles, il était nécessaire de faire appel à un expert du système. De nos jours, les utilisateurs s'attendent à pouvoir faire eux-mêmes des recherches avancées sur des bases de données. Le fait de rendre les utilisateurs autonomes dans leur recherche permet de réduire l'effort requis pour porter assistance aux utilisateurs.

Dans ce chapitre, nous allons en premier lieu, présenter une brève définition de ce qu'est une base de données ainsi que les différentes approches de conception de requêtes à des bases de données relationnelles. En second lieu, nous allons présenter une catégorisation des irritants et des défis rencontrés lors de la formulation de requêtes. Par la suite, nous allons discuter des principes d'interactions guidées et de la formulation de requêtes par gestes. Pour finir, nous allons présenter l'approche de conception centrée sur l'utilisateur et quelques facteurs humains à prendre en compte dans le design des interfaces utilisateurs.

### **3.1 Mécanisme de recherche bibliographique**

Les bases de données utilisées pour effectuer la revue de littérature sont les suivantes : Compendex et Google Scholar. Dans la première phase de revue de la littérature, la chercheuse s'est concentrée sur les interfaces et les langages de formulation de requêtes basés sur des bases de données relationnelles. De plus, elle s'est intéressée aux documents dont l'année était plus récente que l'année 2000. Les documents concernant les recherches sur des fichiers XML et sur des données provenant de sources web et ontologies n'ont pas été pris en considération.

Par la suite dans une deuxième phase de revue de littérature, la chercheuse s'est intéressée à des interfaces et des langages de requête sur des données provenant de plusieurs sources sur le web et basés sur des ontologies liées à des systèmes SPARQL.

## 3.2 Qu'est-ce qu'un système de gestion de base de données (SGBD) ?

Une base de données est un ensemble de données disponibles dans un système informatique et pouvant être interrogée pour son contenu. Autrement dit, il doit être possible d'obtenir les données qui répondent à un certain nombre de conditions. Les données sont liées entre elles et conservées dans un format préétabli pour chaque type de donnée. Par exemple, un client a un nom, une date de naissance, un numéro d'assurance sociale, etc. L'information est sauvegardée sur une mémoire secondaire [3]. La présente recherche ne s'intéresse qu'à l'aspect de consultation des données et non pas à l'ajout, la modification ou la suppression de données.

L'ouvrage de Gardarin (2003) définit deux types de schémas de données : schéma interne représenté par des tables de la base de données aussi appelée « schéma physique » et le schéma externe aussi nommé « vue externe » (ou modèle logique) qui donne accès à un sous-ensemble de l'information. Ce schéma permet d'assurer une certaine sécurité des données [3]. Pour une même base de données, il ne peut exister qu'un seul schéma interne, mais il peut exister plusieurs schémas externes pour répondre aux différents besoins des utilisateurs.

Il existe une grande variété de langages interactifs qui donnent un accès aux données tels que les langages de commandes (SQL), les langages graphiques ainsi que l'interrogation par menus ou par formes. Le langage naturel est encore généralement trop complexe pour permettre d'interroger une base de données efficacement.

Depuis quelques années, les logiciels sont basés sur de multiples bases de données qui sont reliées entre elles. Il doit être possible d'interroger les différentes bases de données en utilisant des jointures entre les différentes tables. Cette opération est difficile puisqu'il est nécessaire de connaître les schémas de données et les termes utilisés dans les différentes bases de données.

Avec l'arrivée du Web 2.0, les sources de données se sont multipliées. En effet, il est maintenant possible d'avoir accès à une très grande quantité d'information sur plusieurs sources internet. L'équipe de Jarrar et Dikaiakos s'intéresse aux mashup. Un mashup est une application qui consomme et combine de l'information provenant de plusieurs sources de données différentes sur le web (web API). L'équipe de Jarrar et Dikaiakos présente le web sémantique comme une structure comparable à une base de données où les sources sont comparables à des tables et un mashup est comparable à une requête. Le langage SPARQL a été créé pour permettre d'interroger et combiner

des données provenant de plusieurs sources différentes sur le web. La création d'applications mashup nécessite une connaissance très poussée du langage informatique [4, 5].

### **3.3 Aspects à considérer lors du développement d'une interface de formulation de requêtes**

Les utilisateurs de système de formulation de requêtes sont souvent considérés à tort comme des utilisateurs novices. Or souvent, ce sont plutôt des utilisateurs qui ont une connaissance très poussée du domaine d'affaires, qui maîtrisent bien les données et qui ont des besoins très sophistiqués pour la formulation de requêtes. Toutefois, leur connaissance du langage SQL est souvent nulle ou très faible [2]. C'est pourquoi il est nécessaire de créer des interfaces faciles à utiliser, mais qui offrent la possibilité de créer des requêtes complexes incluant des opérateurs logiques ET, OU et des jointures.

Lors de la création des interfaces, il faut tenir compte des cinq objectifs suivants [2] :

- 1 Même si l'utilisateur est expert dans son domaine d'affaires, il est considéré comme n'ayant aucune connaissance du langage SQL. Il ne connaît pas non plus le schéma de données et ne devrait pas être forcé de créer des jointures.
- 2 L'utilisateur doit être en mesure de créer des conditions booléennes sans devoir utiliser de parenthèses ou tenir compte des priorités des opérateurs logiques.
- 3 La formulation de conditions quantitatives existentielles (« il existe au moins un élément qui répond à la condition », « tous les éléments répondent à la condition » ou « aucun élément ne répond à la condition ») ou de jointures est difficile pour les utilisateurs. L'interface doit offrir une façon facile de les créer.
- 4 On s'attend à ce que l'utilisateur procède par étape pour la création de la requête. Il doit donc être possible de facilement tester et raffiner la requête.
- 5 L'utilisateur doit être en mesure de grouper des enregistrements et de spécifier des agrégations telles que sommation et moyenne.

La portion de création de requêtes la plus facile pour les utilisateurs est basée sur les attributs des éléments, c'est-à-dire des conditions simples (interface par filtres). La portion plus difficile est liée

aux opérateurs logiques. La complexité d'une requête peut dépendre des cinq catégories suivantes [2] :

- 1 opérateurs logiques ET, OU, NON ET, NON OU ;
- 2 agrégation (somme, moyenne, etc.) ;
- 3 les conditions quantitatives existentielles (il existe au moins un élément qui répond à la condition, aucun élément ne répond à la condition) ou jointures ;
- 4 négation et quantification universelle (tous les éléments répondent à la condition) ;
- 5 jointures de l'élément sur lui-même ou des requêtes récursives.

### **3.4 Approches de conception de requêtes à des bases de données relationnelles**

Au cours des dernières années, les chercheurs et les développeurs ont défini plusieurs approches de conception de requêtes à des bases de données relationnelles :

- L'approche basée sur la recherche par mots clés est plus facile à utiliser, mais ne permet pas de créer des requêtes complexes et précises [6].
- L'approche basée sur un langage structuré souple, tel que SQL, est puissante, mais difficile à utiliser pour l'utilisateur moyen. Entre autres parce que l'utilisateur doit connaître le modèle des tables de la base de données qui est souvent très différent du modèle d'affaires de l'entreprise [6].
- L'approche basée sur le langage naturel permet de construire des requêtes complexes. Toutefois, pour que le langage naturel soit suffisamment expressif pour construire des requêtes complexes, il devient souvent aussi difficile à maîtriser que SQL. Par ailleurs, la recherche par langage naturel retourne parfois des résultats inappropriés [6, 7].
- L'approche impliquant une interface composée de formulaires (form-based) de requête permet de guider l'utilisateur dans la construction de la requête [2].
- L'approche basée sur le langage graphique (GQL) permet aux utilisateurs expérimentés de formuler des requêtes en format graphique. Cette approche est aussi appelée « requête par exemple » [traduction libre de « query-by-example »] [8].

- L'approche basée sur un diagramme (query-by-diagram) proposé par l'application mashQL permet de formuler la requête de façon dynamique à l'aide d'un formulaire présenté sous forme de diagramme. [9].
- L'approche de requête par filtres permet d'ajouter un filtre simple à la fois (champ, opérateur, valeur). Cette approche ne permet pas de logique plus complexe impliquant des opérateurs logiques « OU » ou des jointures.
- L'approche de formulation de requêtes par gestes. Cette approche a été développée pour être utilisée sur des appareils mobiles qui offrent une interface qui se manipule par des gestes et avec un minimum d'interactions au clavier.
- L'approche par la manipulation directe des données [10] permet de filtrer l'information directement sur la grille qui présente l'information. Cette approche est inspirée des tableaux.
- L'approche par recherche approximative [traduction libre de « fuzzy search » ou « approximate search »]. Cette approche basée sur la recherche par mot clé permet de spécifier la valeur recherche de façon imprécise. Par exemple, sans spécifier d'accents. La recherche considère les synonymes.

### **3.4.1 Recherche par mots clés**

Les moteurs de recherches disponibles sur le web sont à l'origine de la popularité actuelle de recherche par mots clés [11]. Les recherches sur le web sont effectuées sur une grande quantité de documents. Le résultat est trié en fonction de la pertinence du document. Plus il y a d'occurrences du mot clé recherché dans le document, plus le document est jugé pertinent. Lorsqu'on effectue une recherche dans une base de données, la recherche n'est pas effectuée sur des documents, mais plutôt sur des tables. Le système doit donc faire des jointures entre les différentes tables. Pour ce faire, les tables et les colonnes impliquées dans la recherche doivent être préalablement établies. Le système effectue les jointures lui permettant d'identifier des correspondances dans les mots clés spécifiés. Cette approche a plusieurs désavantages. Elle ne permet pas de faire une recherche intelligente de l'information puisque tous les mots clés spécifiés doivent être retrouvés pour que l'information soit retournée dans le résultat [11, 12]. De plus elle ne permet pas de tenir compte de l'intention précise de l'utilisateur et peut retourner un résultat peu pertinent. Enfin, tous les résultats

sont présentés de façon pêle-mêle. C'est-à-dire que les résultats de même nature ne sont pas groupés ensemble et la nature du résultat (jointures impliquées) n'est pas affichée [13].

Selon Yu et Jagadish, il ne serait pas suffisant, pour les systèmes de base de données, de simplement offrir un moteur de recherche par mots clés et de laisser le moteur mener l'interaction avec l'utilisateur. Les moteurs de recherche par mots clés sont efficaces pour le web, mais pas pour une base de données [14]. Contrairement à des recherches sur le web, sur une base de données, les utilisateurs s'attendent à un résultat très précis. Rien de plus, rien de moins. Tout écart de résultat devrait leur être expliqué. Ces attentes sont dues aux caractéristiques suivantes des systèmes de base de données [14] :

- Capacité de faire des requêtes plus sophistiquées. Les utilisateurs désirent construire des requêtes avec une sémantique complexe. Ils s'attendent à utiliser des opérateurs logiques et autres éléments sémantiques complexes. Cette sémantique ne peut pas être supportée par un système de recherche par mots clés.
- Les attentes des utilisateurs sont plus élevées. Ils s'attendent à une plus grande précision que pour une recherche sur le web où ils tolèrent des erreurs.
- Les utilisateurs s'attendent à un résultat structuré contrairement à une recherche sur le web où on s'attend à une liste de liens indépendants les uns des autres.

La première approche discutée dans la recherche de ces auteurs correspond à un modèle de recherche par mots clés sans structure sur une base de données structurée. L'utilisateur spécifie les mots clés et la sémantique est construite automatiquement par le système [6] :

- 1- Basée sur l'ancêtre commun le plus bas. C'est-à-dire qu'on identifie tous les éléments liés aux mots clés spécifiés. On conserve l'arbre de relations entre les éléments qui est le plus concis et le plus bas dans l'arborescence globale de la base de données.
- 2- Basée sur le groupe d'enregistrements le plus petit. C'est-à-dire qu'on récupère toutes les données qui contiennent les mots clés spécifiés et on conserve seulement les enregistrements uniques.

Une autre approche est la recherche par mots clés/libellés où on permet de spécifier le nom du champ associé au mot clé recherché. C'est-à-dire que l'utilisateur spécifie le libellé du champ attaché au mot clé : par ex., Région = « Asie » où « Région » est le libellé du champ qui fait partie

du schéma logique des tables de la base de données [6]. La recherche par mots clés/libellés permet de faire une recherche basée sur une sémantique plus stricte. Par conséquent, elle permet d'obtenir des résultats plus précis.

Il y a deux inconvénients à ces approches. Premièrement, le résultat est peu significatif. Deuxièmement, cette approche nécessite la consultation de la base de données entière. Cela engendre des coûts importants. De plus, il est impossible de créer des requêtes avec une structure sémantique spécifique.

Certaines études proposent une approche collaborative. Ici le système propose un formulaire à l'utilisateur basé sur des recherches passées effectuées par d'autres utilisateurs [15]. Le fait de proposer une grande quantité de formulaires crée une problématique. La recherche parmi toutes les options est fastidieuse [16]. Certaines études proposent une approche hybride entre la recherche par mots clés et la recherche par formulaire [15, 16]. L'utilisateur spécifie un certain nombre de mots clés et le système propose des formulaires appropriés parmi une liste préétablie de formulaires.

L'application Croesus Conseiller permet de faire le partage de requêtes préalablement sauvegardées entre les utilisateurs. Cette fonctionnalité est toutefois difficile à utiliser en raison de la trop grande quantité de requêtes partagées. Cette fonctionnalité devrait être améliorée pour permettre un partage plus efficace entre les utilisateurs.

### **3.4.2 Modèle de requêtes structuré souple**

Une autre approche pour la recherche par mots clés est basée sur une structure souple. C'est-à-dire qu'on utilise les mots clés sans structure dans des conditions qui font partie d'une requête qui contient aussi des conditions de structure : par ex., Tremblay et Montréal et REER où « Tremblay » réfère au nom du client, « Montréal » réfère à l'adresse du client et « REER » fait allusion à une jointure entre le client et le compte et réfère au type de compte. Cette approche a des limites. Les utilisateurs doivent spécifier les jointures à utiliser entre les éléments, et cela est difficile à exprimer correctement [6].

Il est très difficile pour des utilisateurs moyens n'ayant aucune notion du langage SQL de faire des requêtes à une base de données directement avec SQL. Entre autres parce que ça les oblige à connaître le modèle des tables de la base de données. La création de jointures dans une requête est



difficile pour la majorité des utilisateurs [17]. De plus, le modèle de tables de la base de données est une information qui doit rester cachée pour des raisons de sécurité. Par ailleurs, un modèle de tables normalisées est souvent trop complexe et trop différent du domaine d'affaires pour être compris par les utilisateurs [6, 14, 18]. Le schéma d'éléments logiques exposé aux utilisateurs doit leur permettre de se construire un modèle mental cohérent.

Pour faciliter la tâche des utilisateurs, il est recommandé de leur exposer un modèle d'éléments logiques qui est simple et plus proche du domaine d'affaires [14]. Le modèle de base de données complexe est alors simplifié sous forme d'éléments importants représentés dans un schéma logique pour le domaine d'affaires. Le défi dans cette approche est de combler les jointures qui sont cachées dans le schéma abstrait sommaire. Yu & Jagadish [6] proposent une approche basée à la fois sur le schéma sommaire et sur la recherche par mots clés sans structure pour combler les manquements.

Cette approche est déjà utilisée dans les interfaces offertes dans l'application Croesus Conseiller. Elle sera conservée pour la nouvelle interface.

### **3.4.3 Modèle basé sur le langage naturel**

Dans une approche basée sur le langage naturel, le système devrait agir comme un administrateur de base de données [19]. L'utilisateur fait sa demande dans ses mots au système et celui-ci analyse la demande. S'il y a des ambiguïtés dans les phrases ou les mots, le système valide auprès de l'utilisateur. Par la suite, ce sont les ambiguïtés relatives à la structure de la demande qui sont validées. Une fois que l'utilisateur a confirmé la compréhension de la demande initiale, le système traduit la demande dans un langage structuré tel que SQL. Par la suite, le résultat est rendu disponible à l'utilisateur.

Le développement d'une interface de requête en langage naturel comporte quelques problèmes qui sont définis ci-dessous [4, 9, 20, 21] :

- le développement est très complexe en raison des variations linguistiques et des ambiguïtés. En contrôlant le langage de requête à l'aide de dialogues, on élimine les ambiguïtés ;
- une bonne implémentation d'interface en langage naturel implique une association rigide avec le domaine (source d'information web API) ce qui rend l'interface peu adaptable ;

- le temps réponse requis pour l'exécution de la requête dépend de son association avec un domaine précis (web API). Plus l'interface est liée à un domaine, plus elle est performante ;
- même en développant une interface efficace sur des domaines (web API) variés, il n'est pas démontré si c'est utile pour les utilisateurs ;
- ce type d'interface offre peu de précision dans le résultat obtenu ;
- les utilisateurs doivent savoir ce qui est possible de demander et comment le formuler.

Kaufmann et Bernstein [22] présentent l'hypothèse de l'habitabilité (*Habitability hypothesis*) selon laquelle les utilisateurs novices sont coincés entre l'imprécision des langages par mots clés interprétés et la rigueur des systèmes de requêtes formels. Aucune des deux options ne répond à leurs besoins. Selon l'équipe, il est préférable d'imposer des restrictions lors de la saisie d'une requête en langage naturel pour guider l'utilisateur et l'aider à formuler la requête correctement. Le défi est de trouver le juste milieu acceptable entre la rigueur trop rigide et la souplesse trop imprécise.

Pour que cette approche soit intéressante pour un utilisateur, elle doit être configurée de façon très précise. C'est une approche coûteuse qui demande beaucoup de maintenance. De plus cette approche fait appel à des notions d'intelligence artificielle. Ces notions ne font pas partie du champ de compétence de la chercheuse. Pour ces raisons, cette approche ne sera pas utilisée pour la nouvelle interface.

### **3.4.4 Modèle basé sur les interfaces de type formulaires (VQI « form based »)**

Les interfaces classiques de requête à une base de données sont basées sur des formulaires rigides. Les utilisateurs spécifient les valeurs recherchées dans des champs préétablis. La sémantique de la requête est rigide. Pour permettre plusieurs sémantiques, il faut prévoir plusieurs formulaires. Cette approche est encore beaucoup utilisée. Elle permet aux utilisateurs d'avoir un accès facile à l'information [15]. Toutefois, cette approche offre très peu de souplesse pour la formulation de la requête. L'utilisateur est contraint à la sémantique imposée par le formulaire.

Il faut prendre note que les systèmes de requêtes par mots clés ne permettent pas de faire les requêtes imbriquées ni d'utiliser les jointures ou les agrégations. La formulation de requêtes à l'aide

d'interface de type formulaire permet de faire ce genre de requêtes [23] dans la mesure où l'interface a été prévue pour le faire.

Une des avenues possibles pour l'utilisateur de VQI (Visual Query Interface) serait de créer un algorithme de construction automatique d'interface de type formulaire. Le formulaire s'adapterait au besoin de l'utilisateur au fur et à mesure de la progression de la formulation de la requête.

Pour améliorer l'utilisabilité d'une interface de type formulaire, certains auteurs [23] proposent la solution suivante : permettre une recherche avec auto complétion [traduction libre de « search-as-you-type »]. La solution est de créer des tables qui contiennent les valeurs distinctes possibles pour chaque attribut (champ). Lorsque l'utilisateur saisit une valeur, le système offre la liste des valeurs possibles qui correspondent de façon automatique. Cette façon de faire permet à l'utilisateur d'explorer les données disponibles et de formuler une requête correcte plus rapidement. De cette façon il est guidé dans la construction de sa requête. Le défi de cette approche est lié à la vitesse de réaction du système [23].

Ce type d'interface est intéressant pour la formulation de requêtes n'impliquant pas de jointure ou seulement des jointures préétablies. L'approche sera utilisée pour la nouvelle interface, toutefois, elle doit être adaptée pour permettre l'utilisation plus souple de jointures.

### **3.4.5 Formulation de requêtes avec un langage graphique (GQL)**

L'équipe de Keramopoulos [8] s'est intéressée aux interfaces de formulation de requêtes graphique (GQL). Selon cette équipe, la plupart des interfaces de requêtes graphiques s'adressent davantage aux utilisateurs de niveau expert et sont mal adaptées à des utilisateurs novices. L'équipe s'est intéressée aux systèmes GOQL qui utilisent un langage graphique de requête basé sur des bases de données orientées objet. Dans ce langage, la procédure pour la création d'une requête consiste à sélectionner les objets (entités) impliqués dans la requête désirée dans un modèle présenté sous forme graphique. Par la suite, l'interface présente automatiquement une fenêtre de création séparée en deux sections : celle qui permet de construire la requête et celle qui présente les outils de formulation. La section de la construction de la requête contient les objets (entités) qui ont été sélectionnés préalablement. L'utilisateur peut sélectionner les attributs des objets sur lesquels il veut filtrer et spécifier les valeurs attendues. Cette méthode est aussi appelée « requête par exemple » [traduction libre de Query-by-Example] [24].

Cette interface permet d'utiliser une syntaxe riche, toutefois pour permettre aux utilisateurs novices de formuler des requêtes, un modèle de données simplifié doit leur être présenté alors que les utilisateurs de niveau expert préfèrent avoir une vue complète du schéma physique des tables. L'équipe de Jarrar et Dikaiakos [5, 25] ainsi que Styperek, Ciesielczyk, Szwabe et Misiorek [26] affirment que cette approche est peu utilisée par les utilisateurs finaux. Ils avancent l'hypothèse que ce type d'interface nécessite l'apprentissage et la compréhension du modèle logique, ce qui n'est pas naturel pour des utilisateurs qui ne sont pas familiarisés avec le langage de programmation.

Un système permettant de formuler des requêtes dans une interface graphique sur un appareil Android a été développé sous le nom de Db4o. Ce produit permet la formulation de requêtes simples dans une interface simple et intuitive qui s'adresse à des utilisateurs peu expérimentés avec des bases de données (RDBMS) [27]. Cette interface semble permettre de formuler des requêtes relativement complexes, c'est-à-dire qui comportent des jointures et des opérateurs logiques ET et OU.

Selon les auteurs, cette interface est peu adaptée à des utilisateurs novices. Pour cette raison, cette approche ne sera pas retenue pour la nouvelle interface.

### **3.4.6 Formulation de requêtes par filtres**

Cette approche est utilisée dans plusieurs interfaces de gestion de courriels. Elle permet de raffiner une liste de données en ajoutant des filtres un à un. Ces derniers sont utilisés avec des opérateurs logiques ET. Cette approche est limitée mais est bien comprise et beaucoup utilisée par les utilisateurs novices. L'application Croesus Conseiller offre une approche par filtres sur une liste d'éléments [4].

L'approche par filtres peut être utilisée en combinaison avec d'autres approches. C'est-à-dire que chaque condition de la requête peut être formulée à l'aide d'un filtre. La nouvelle interface va donc s'inspirer de cette approche pour la formulation des conditions en combinaison avec d'autres approches pour la sémantique de la requête.

### 3.4.7 Formulation de requêtes par diagramme

Le langage de requête par diagramme nommé MashQL est basé sur l'approche des requêtes par filtres. Les différents filtres sont affichés sous forme d'arbre d'information. Cette façon d'afficher la requête permet de mieux la comprendre. L'utilisateur doit spécifier l'entité (ou la source de données) qui doit être interrogée. Chaque branche de l'arbre correspond à une condition. Une branche peut être étendue en un sous-arbre qui interroge un sous-ensemble d'informations (jointure). Ce langage permet la formulation interactive de la requête grâce à l'utilisation de listes déroulantes pour la formulation de la requête. La complexité de la structure des données est gérée par le système et cachée à l'utilisateur. Pour formuler les conditions, l'utilisateur doit sélectionner l'attribut (champ) sur lequel il veut filtrer. Par la suite, il doit choisir l'opérateur à utiliser et spécifier la valeur. Ce langage permet aussi des jointures négatives et autres opérations sémantiques complexes qui ne font pas partie de la présente recherche (par exemple : union, intersection, etc.). Cette approche est facile à utiliser mais permet aussi une lecture proche du langage naturel, donc facile à lire [4, 5, 25]. Ce langage permet à des utilisateurs novices de créer facilement des mashup. Le langage est ensuite traduit par le système en requête SPARQL.

La limitation de ce langage tient au fait que les termes affichés proviennent de la structure de données sous-jacentes. Il est donc possible que les noms des champs soient peu significatifs pour les utilisateurs finaux [23]. De plus, ce langage force l'utilisateur à formuler la requête de façon abstraite dans une interface qui ne permet pas de voir les données. L'utilisateur doit naviguer entre l'interface de formulation de requêtes et le résultat afin de raffiner la requête pour obtenir un résultat plus précis [10].

Le langage permet d'utiliser le résultat d'une requête comme point d'entrée à une autre requête. Cette façon de faire permet la formulation de l'équivalent d'une jointure. Cette façon de casser la requête en plusieurs étapes aide les utilisateurs novices à mieux comprendre celle-ci. Cette approche est appelée pipeline par l'équipe de M. Jarrar [4].

Selon Khule et Keole [20], pour que les données disponibles sur le web soient utilisées à leur plein potentiel, les utilisateurs n'ayant aucune connaissance en programmation doivent être en mesure de formuler des requêtes efficaces rapidement et facilement.

Cette approche par diagramme est intéressante pour le développement de la nouvelle interface. Elle utilise l'approche par filtre sous forme de petits formulaires et permet de faire la gestion des jointures. Elle va donc nous inspirer pour le développement de l'interface.

### **3.4.8 Formulation de requêtes par des gestes**

Les utilisateurs de systèmes informatiques se tournent de plus en plus vers des appareils mobiles tels que téléphones intelligents et tablettes. Ces appareils fonctionnent avec des gestes différents de ceux d'un ordinateur conventionnel comprenant un clavier et une souris. Pour que l'interface de formulation de requêtes puisse être utilisée de façon agréable sur un appareil mobile, il faut prévoir une interaction par gestes et non pas par clavier [28]. C'est ce qu'on appelle une « base de données gestuelle » (traduction libre de « gestual database »).

Il existe plusieurs défis liés à la formulation de requêtes par des gestes. Premièrement, le nombre d'actions possibles est grand. Il dépend du langage utilisé (SQL), du schéma de tables de la base de données et des données contenues dans celle-ci. Une solution à ce problème est d'offrir à l'utilisateur de sélectionner le type de requête désirée et ensuite de spécifier les paramètres de la requête choisie. Si l'utilisateur est incertain du type de requête à choisir, le système doit le guider pour répondre à son besoin et obtenir le résultat désiré [28].

L'utilisation de gestes pour la formulation d'une requête engendre un nouveau paradigme. Dans les systèmes avec clavier, l'utilisateur spécifie la requête à l'aide du clavier sous forme de chaîne de caractères et ensuite l'exécute pour obtenir un résultat. Dans un système par gestes, le système donne une rétroaction constante durant la formulation de la requête. Les utilisateurs s'attendent à gérer la formulation de la requête et gérer les données par une action directe avec les données et la requête [28]. La formulation de requêtes par gestes ouvre la voie à une grande possibilité de gestes différents. Les utilisateurs peuvent se sentir perdus dans un tel système puisqu'il est très différent des systèmes classiques. Le système doit donc guider les utilisateurs dans la formulation de leur requête [29].

Cette approche ne permet pas l'utilisation de jointures. En effet, il n'est pas possible de donner une rétroaction instantanée lors de l'utilisation de jointures. Il serait difficile de donner le résultat de la sous-requête associée à chacune des jointures en plus du résultat de la requête principale. Pour cette

raison, cette approche ne sera pas utilisée pour la nouvelle interface. Toutefois la nouvelle interface doit prévoir la capacité d'être utilisée sur des appareils mobiles.

### **3.4.9 Formulation de requêtes par la manipulation directe des données**

Beaucoup d'utilisateurs aiment utiliser un tableur pour effectuer des recherches. Il y a une blague dans le domaine qui dit que le troisième bouton le plus utilisé dans un système d'intelligence d'affaires est le bouton d'exportation à MS Excel, les deux premiers étant les boutons OK et le Cancel [10]. Les utilisateurs de l'application Croesus Conseiller n'y font pas exception. Il semble que les utilisateurs apprécient le tableur pour effectuer des recherches en raison de la manipulation directe des données. En effet, dans un tableur, les données interrogées sont visibles et les actions sont réversibles et incrémentales. De plus, la manipulation est effectuée directement sur les données sans avoir recours à des commandes complexes [10, 30].

Bakke & Karger [10, 30] pensent qu'il est nécessaire de permettre une manipulation directe des données pour la consultation de base de données. En effet, l'objet d'intérêt d'une requête réside dans le résultat et non pas dans la requête elle-même. Selon eux, les conditions suivantes doivent être respectées pour développer une interface utilisateur agréable à utiliser :

- la formulation de la requête devrait se faire de façon incrémentale à l'aide de manipulation sur les données à travers une séquence d'opérations ;
- l'utilisateur doit être en mesure de voir la requête et d'effectuer des corrections sur celle-ci sans avoir à refaire toutes les étapes ;
- permettre d'utiliser une sémantique riche comparable à celle offerte par le langage SQL.

La solution proposée par Bakke & Karger permet de faire des recherches sur une base de données relationnelle impliquant des jointures. Ce qui n'est pas le cas du tableur qui ne permet de faire des recherches que sur un tableau de données à une dimension.

Dans la solution proposée, pour ajouter des filtres et des jointures il suffit de cliquer avec la souris sur l'entête des colonnes de la grille qui présente le résultat. Un menu apparaît et il est possible de formuler des conditions, des jointures et des calculs. Une série d'icônes permet de visualiser la requête. Il est possible à tout moment de modifier la requête à l'aide de la souris sur les entêtes de colonnes [10].

Cette approche implique que les données soient affichées en tout temps à l'écran sous forme de grille avec des lignes et des colonnes. Or parfois, il est possible de formuler une requête dont le résultat sera utilisé sous une autre forme, par exemple pour la production de rapports. De plus, il y a des enjeux de performances liés au fait d'afficher toute l'information d'abord, et filtrer par la suite. Pour cette raison, cette approche ne sera pas utilisée.

### **3.4.10 Formulation de requêtes par recherche approximative [traduction libre de « fuzzy search »]**

L'approche de recherche approximative consiste à faire une recherche par mot clé avec une certaine souplesse, par exemple en considérant les synonymes. Elle corrige les erreurs de saisie ou l'absence d'accents comme le ferait un système d'autocorrection. Cette approche est principalement utilisée par les moteurs de recherche sur le web. Elle est aussi utilisée pour des systèmes de recherche par reconnaissance vocale. Cette approche a l'inconvénient d'augmenter le risque d'obtenir des résultats non significatifs [31, 32]. Elle nécessite de déterminer le niveau d'erreur accepté. Il est aussi nécessaire de créer des dictionnaires de synonymes.

Cette approche serait intéressante pour l'interface développée dans notre recherche pour aider l'utilisateur à spécifier les champs et les valeurs de leur recherche. Toutefois, son implémentation nécessiterait beaucoup de temps. C'est la raison pour laquelle elle ne sera pas utilisée.

## **3.5 Catégorisation des irritants lors de la formulation de requêtes**

Certains auteurs [14] ont catégorisé les types d'irritants ou de frustrations rencontrés lors de la formulation de requêtes de la façon suivante :

### **3.5.1 Irritants avec les jointures [traduction libre de « Painful relations »]**

Lorsque les requêtes impliquent une seule table, la formulation de requêtes est relativement facile pour la majorité des utilisateurs. Lorsqu'il y a plusieurs tables ou éléments impliqués, il est nécessaire d'utiliser des jointures entre les éléments. Pour ce faire, l'utilisateur doit connaître le modèle des éléments offert. La création de jointures n'est pas naturelle pour beaucoup d'utilisateurs [17]. Il est difficile de déterminer quelle requête utiliser parmi plusieurs jointures.



### **3.5.2 Irritants avec les options multiples** [traduction libre de « Painful options »]

On pourrait croire, à tort, que le fait d'offrir différentes interfaces pour construire des requêtes serait une solution supérieure. Chaque interface pourrait fonctionner selon une logique différente et retourner un résultat différent, ce qui offrirait beaucoup de possibilités aux utilisateurs. Cette façon de faire entraîne la création de logiciels ayant trop d'options et trop d'interfaces qui créent beaucoup de confusions. De plus, les utilisateurs comparent les résultats des différentes interfaces et s'attendent au même résultat [14].

Les psychologues parlent de la théorie du « chemin non choisi » [traduction libre de « paths not taken »] qui implique un coût cognitif causé par le regret face à l'option qui n'a pas été choisie qui ne serait pas vécu si les multiples options n'étaient pas offertes [33]. Par ailleurs, trop de choix causent un sentiment de « besoin de fermeture » [traduction de « need of closure »] [34].

Vronay, Marcjan, Turski et Kott [7] proposent un système de langage naturel semi-structuré. L'utilisateur peut taper la requête à l'écran et le système le guide en lui proposant la suite de la requête. Cette approche est semblable à une recherche par mots clés/libellés où le champ auquel le mot clé est associé doit être spécifié en association avec le mot clé. Cette approche est très intéressante, mais offre les mêmes limitations que la recherche par mots clés/libellés de la recherche de Yu et Jagadish [6]. Pour permettre de faire des requêtes complètes impliquant des jointures, elle pourrait être utilisée pour la formulation de segments de requêtes simples, conjointement avec un autre système pour la gestion des jointures entre les éléments.

### **3.5.3 Irritants avec les résultats inattendus** [traduction libre de « unexpected pain »]

Un autre type de frustration pouvant être vécue par les utilisateurs survient lorsque la requête donne des résultats inattendus sans explication [14]. Les attentes face aux résultats sont basées sur la compréhension du système et du schéma logique des données. Si l'utilisateur sait qu'une certaine valeur ou donnée existe, mais que cette donnée n'est pas retournée par le système, il ressent une frustration face au système parce qu'il ne comprend pas ce qui en est la cause et ne peut donc pas corriger sa requête pour obtenir un résultat plus précis. Une règle importante de l'utilisabilité est de permettre à l'utilisateur de rester en contrôle de l'état du système. Pour atteindre cet objectif, il

faut être en mesure d'expliquer les résultats à l'utilisateur. Ce qui est difficile dans les systèmes de recherche sur base de données.

### **3.5.4 Irritants avec les résultats non visibles [traduction libre de « unseen pain »]**

Dans la majorité des interfaces de construction de requête, la construction est séparée du résultat. L'utilisateur doit construire une requête, l'exécuter, valider le résultat et recommencer jusqu'à ce que la requête soit suffisamment précise. Or, si le résultat est difficile à obtenir et qu'il nécessite plusieurs opérations pour y avoir accès, la tâche de création de requêtes devient fastidieuse. La construction de la requête implique que l'utilisateur ait la capacité de prédire le résultat. La plupart des utilisateurs ne sont pas familiers avec la construction de requête. Ils construisent la requête par étape et valident le résultat au fur et à mesure que la construction de la requête avance [13, 17]. Il est donc important que le résultat de la requête soit accessible facilement et qu'il soit possible de raffiner la requête à plusieurs reprises. Dans les systèmes de recherche par mot clé, la requête se fait facilement et le résultat est obtenu rapidement. Il est aussi facile de raffiner la recherche pour préciser le résultat. Ce sont des systèmes qui offrent une bonne utilisabilité puisque le résultat est facilement accessible et proche de la création de la requête. C'est ce qui fait défaut dans les systèmes de recherche sur base de données [14].

Le défi dans l'affichage du résultat réside dans la présentation simple de la base de données. En effet, si plusieurs éléments sont impliqués dans la requête et que les jointures sont de type un à plusieurs ou plusieurs à plusieurs, il n'est pas possible d'afficher le résultat simplement sous forme de liste. Le résultat doit aussi afficher les liens entre les éléments retournés et les éléments impliqués dans la requête. Il doit être possible pour l'utilisateur de naviguer à travers les résultats pour valider l'exactitude de ceux-ci [13, 35].

## **3.6 Catégorisation des défis rencontrés par les utilisateurs d'interface de formulation de requêtes**

Nandi & Jagadish [18] ont remarqué que les utilisateurs de systèmes de formulation de requêtes à une base de données en général peuvent passer plus de temps à formuler les requêtes que le temps requis par le système pour les exécuter. Ils pensent qu'il est possible d'utiliser la puissance du

système, non seulement pour exécuter les requêtes, mais aussi pour aider et guider les utilisateurs durant la formulation de la requête. Ils ont identifié les quatre types de défis suivants pouvant être rencontrés lors de la formulation de requêtes.

### **3.6.1 Manque de connaissance**

Le premier défi rencontré lors de la formulation de requêtes est lié à la connaissance du système. L'utilisateur doit être familiarisé avec le langage de formulation de la requête. Ensuite, il doit être familiarisé avec le schéma logique exposé. Il est aussi important d'être familiarisé avec les valeurs enregistrées, c'est-à-dire les données elles-mêmes [18]. Il est parfois nécessaire de faire des requêtes pour connaître le format des valeurs possibles et comprendre le modèle de base de données.

### **3.6.2 Dépendance de l'information**

La construction de la requête utilise parfois des éléments de structure de requête telle que des jointures, des groupements de conditions des sous-requêtes ou même parfois des calculs plus complexes. Pour construire une requête qui implique des éléments de structure de requête, il est nécessaire de connaître le modèle de table de la base de données, mais aussi le format des données elles-mêmes [18].

### **3.6.3 Construction de requête de façon itérative et incrémentale**

Il est important de prendre en considération que les capacités cognitives des utilisateurs sont limitées [36]. Il pourrait être possible de formuler une requête très complexe en une seule étape, mais l'être humain impliqué dans la formulation a une mémoire à court terme limitée. Pour remédier à cette situation, les utilisateurs construisent généralement les requêtes par petites étapes incrémentales et itératives [18].

Au départ, une simple requête est construite et exécutée. L'utilisateur s'assure que le résultat correspond à ses attentes et retourne modifier la requête en y ajoutant des conditions plus complexes. Ce processus est effectué à plusieurs reprises jusqu'à ce que la requête corresponde au besoin et que le résultat corresponde aux attentes [18].

Ce paradigme de formulation de requêtes est à la fois incrémental et itératif. La requête est constamment en évolution durant la session de formulation de requêtes. L'utilisateur franchit les

étapes « intention → requête → exécution → résultat » (traduction libre de « intent → query → execution → result »)

### 3.6.4 Intention imprécise

Les défis énumérés plus haut rendent difficile la formulation d'une intention claire pour la formulation de la requête pour un utilisateur. Celui-ci peut avoir une idée vague de la requête qu'il désire formuler dès le départ. Il peut faire plusieurs requêtes dans le but de valider ses idées avant de pouvoir formuler une intention plus précise [18].

## 3.7 Principes de l'interaction guidée

Si le système guide l'utilisateur en lui fournissant une rétroaction rapide et fluide sur ses actions, le temps requis pour formuler la requête sera réduit et la satisfaction de l'utilisateur sera augmentée [18]. Pour guider l'utilisateur dans la formulation de la requête, le système doit répondre aux règles qui suivent :

- énumération : Le système de base de données est responsable d'énumérer à l'utilisateur toutes les options de formulation de requêtes possibles. Autrement dit, l'utilisateur saisit la requête et le système propose une auto complétion automatique des valeurs possibles selon le contexte de la requête [18] ;
- renseignements internes : Le système de base de données est responsable de fournir à l'utilisateur le plus de renseignements possible à partir des données. Les informations telles que la distribution des données ou de l'information au sujet des données devraient être retournées automatiquement sans qu'elles ne soient spécifiquement demandées par l'utilisateur [18] ;
- réactif : Toutes les interactions de l'utilisateur avec le système doivent être instantanées même si imprécises. Certaines requêtes peuvent être plus complexes et plus longues à exécuter par le système. L'utilisateur doit attendre le résultat précis sans garantie que celui-ci répondra à ses attentes. Le système doit donner une rétroaction sous forme de résumé de l'information pour que l'utilisateur puisse valider sa requête en attendant que le résultat plus précis soit disponible [18].

## **3.8 Approche de conception centrée sur l'utilisateur**

Lors de la conception d'une interface utilisateur, il est toujours important de tenir compte des principes de base de l'utilisabilité afin de concevoir une interface de qualité pour les utilisateurs. La norme ISO 9241-210 [1] définit l'utilisabilité comme suit : « degré selon lequel un système, un produit ou un service peut être utilisé, par des utilisateurs spécifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié » [1]. La même norme définit l'expérience utilisateur comme suit : « perceptions et réactions d'un utilisateur qui résultent de l'utilisation effective et/ou anticipée d'un système, produit ou service ».

Cette section du chapitre permet de définir les critères de mesure de la performance humaine et de la satisfaction liés à l'utilisabilité selon la norme ISO 9241-11 [1], qui sont l'efficacité, l'efficience, la satisfaction, la facilité d'apprentissage et la mémorisation. Voici leurs définitions.

### **3.8.1 Efficacité**

« L'efficacité se réfère à la capacité d'un dispositif à atteindre un objectif donné » [37]. La norme ISO 9241 la définit comme « la précision ou degré d'achèvement selon lequel l'utilisateur atteint des objectifs spécifiés » [38]. Pour mesurer l'efficacité dans une tâche, il faut évaluer si la tâche a atteint minimalement, partiellement ou totalement l'objectif. Cette notion ne tient pas compte du processus et des étapes requises pour obtenir le résultat. Elle s'intéresse uniquement à la qualité du résultat lui-même.

Les attentes des utilisateurs face au résultat sont variables en fonction du contexte d'utilisation, du niveau d'expertise de l'utilisateur face au produit et de la fréquence d'utilisation. Il faut aussi tenir compte du point de comparaison de départ. Par exemple, si un utilisateur utilise une dactylo pour saisir et éditer un texte, lorsqu'il compare son système actuel avec un système de traitement de texte informatique moderne, il peut considérer que le nouveau système est beaucoup plus efficace que l'ancien. Ce qui ne serait peut-être pas le cas pour un utilisateur qui a l'habitude d'utiliser un système de traitement de texte moderne et qui doit le comparer à un autre tout à fait semblable [37].

### **3.8.2 Efficience**

« L'efficience est la capacité de produire une tâche donnée avec le minimum d'effort » [37]. La norme ISO 9241 définit l'efficience comme un « rapport entre les ressources dépensées et la

précision et le degré d'achèvement selon lequel l'utilisateur atteint des objectifs spécifiés » [38]. Pour mesurer l'efficacité, quatre types d'indicateurs peuvent être utilisés : le nombre et la nature des erreurs, le temps pour exécuter la tâche, le nombre d'opérations requises pour exécuter la tâche et la charge de travail [1].

Pour évaluer la nature des erreurs, il faut tenir compte de l'aspect irréversible de l'erreur. Plus l'erreur est irréversible plus elle aura d'effet sur l'efficacité du système. Par ailleurs, il y a des erreurs qui se produisent sporadiquement et d'autres qui se produisent systématiquement ou à une fréquence régulière. Les erreurs plus fréquentes doivent être considérées en priorité pour être évitées [37].

Les attentes quant au temps requis pour accomplir une tâche varient en fonction du contexte d'utilisation. Par exemple, lorsqu'on désire régler les paramètres d'un poste de radio, les attentes sont différentes si le système audio se trouve dans une voiture par rapport à celui qui se trouve dans un appartement [37].

Les systèmes offrent parfois plusieurs méthodes différentes pour accomplir une même tâche. Certaines méthodes nécessitent moins d'étapes que d'autres, mais les utilisateurs se limitent à leur connaissance du système pour le choix de la méthode pour accomplir la tâche. Lorsqu'on évalue le nombre d'étapes pour accomplir une tâche, il faut tenir compte des habitudes d'utilisation des utilisateurs [37].

La norme ISO distingue plusieurs formes de mesures de l'efficacité : l'efficacité humaine (charge de travail de l'humain pour l'utilisation du système), l'efficacité temporelle (rapport entre l'efficacité du système et le temps requis pour effectuer la tâche), l'efficacité économique (rapport entre l'efficacité du système et les coûts engendrés par l'utilisation du système) [37].

### **3.8.3 Satisfaction**

La satisfaction est liée au niveau de confort vécu durant une tâche exécutée avec le produit. Dans le monde du travail, les critères d'évaluation du niveau de satisfaction sont différents de ceux d'un produit qui s'adresse au public en général. Les utilisateurs se font souvent imposer le produit qu'ils vont utiliser. Dans ce contexte, l'évaluation de la satisfaction est basée sur la productivité de l'individu lorsqu'il utilise le produit. Dans le monde du travail, ce sont donc les critères d'efficacité et d'efficacité qui sont les plus importants pour déterminer le produit à utiliser [1]. Il existe d'autres

aspects liés à la satisfaction, tels que l'attitude face au produit, les qualités matérielles et esthétiques du produit, le besoin ressenti, le plaisir ou confort ressenti durant son utilisation.

Pour évaluer le niveau de satisfaction, il faut avoir recours à des méthodes plus subjectives. À l'aide d'un questionnaire ou d'entrevue, on peut amener l'utilisateur à commenter son expérience d'utilisation.

### **3.8.4 Facilité d'apprentissage et la mémorisation**

La facilité d'apprentissage et de mémorisation est liée au comportement de l'utilisateur. Lorsqu'une personne apprend à utiliser un produit, elle améliore son comportement face au produit. Pour évaluer la facilité d'apprentissage, on peut faire un test comparatif permettant d'évaluer l'amélioration de l'autonomie d'un sujet lors de l'utilisation d'un même système à travers le temps. On peut aussi évaluer le niveau de performance après une période d'inactivité avec le système [1].

Les notions d'affordance et de transparence sont importantes pour que l'utilisation du système soit intuitive. Lorsque l'utilisateur n'a pas accès à l'aide contextuelle ou lorsque l'utilisation est faite de façon sporadique, le système doit être conçu de façon à ce que l'utilisateur puisse accomplir sa tâche de manière autonome et intuitive [37].

## **3.9 Processus de conception centré sur l'utilisateur**

La norme ISO 9241-210 [1] définit un processus de conception de systèmes impliquant des interactions entre des humains et un système ou une machine. Ce processus a pour objectif de produire des systèmes ou des machines qui sont utiles et qui tiennent compte des contraintes, des forces et des faiblesses des utilisateurs. Le processus centré sur les utilisateurs met en application des principes d'ergonomie et des techniques d'utilisabilité et d'accessibilité. Il se divise en quatre étapes et comprend généralement plusieurs itérations [1].

- étape 1 : faire une étude du contexte d'utilisation et des utilisateurs eux-mêmes ;
- étape 2 : documenter les exigences des utilisateurs qui devront être prises en considération pour l'élaboration de la solution ;
- étape 3 : produire une solution, sous forme de prototype ou de maquettes ;
- étape 4 : effectuer des tests d'utilisabilité auprès des utilisateurs visés par le produit.

Suite à l'exécution des quatre étapes, l'équipe d'analystes doit évaluer les résultats au moyen de tests d'utilisabilité pour déterminer si la solution répond aux exigences de la tâche et aux besoins des utilisateurs dans les différents contextes d'utilisation. Selon les résultats obtenus, elle va déterminer s'il est nécessaire de faire une d'autres itérations entre les quatre mêmes étapes pour améliorer la solution [1].

La figure 3.1 présente un graphique présentant les différentes étapes du processus défini par la norme ISO 9241-210 [1]. On peut voir que les étapes sont présentées de façon circulaire pour évoquer l'aspect itératif du processus.



Figure 3.1 Diagramme qui présente le processus de conception centré sur l'utilisateur défini par la norme ISO 9241-210. (l'image est tirée du site suivant :

[https://www.researchgate.net/figure/Modele-UCD-tire-de-la-norme-ISO-9241-210\\_fig1\\_280851696](https://www.researchgate.net/figure/Modele-UCD-tire-de-la-norme-ISO-9241-210_fig1_280851696) date du 9 octobre 2018)

### 3.10 Facteurs humains dans le design et l'ingénierie

La façon de disposer les informations et les dispositifs de commande dans une interface est importante. C'est ce qui fait la différence entre une interface intuitive et agréable ou pas. Les facteurs humains doivent être pris en compte pour la conception de la nouvelle interface qui sera réalisée pendant notre recherche. Cette section présente quelques éléments fondamentaux à considérer.



### 3.10.1 Affichage de l'information

Lorsqu'on affiche de l'information à des utilisateurs, certaines règles d'affichage de l'information doivent être respectées [39].

**Affichage détectable :** pour que l'information affichée puisse être détectée par l'humain, il faut que l'affichage tienne compte du contexte et de l'environnement où l'humain se trouve. Si ce dernier se trouve dans un endroit très sombre, alors l'affichage par la couleur n'est pas approprié pour tout ce qui est affiché en dehors de l'écran. Il faut alors utiliser un affichage qui serait plus approprié au contexte d'utilisation.

**Affichage par redondance :** l'utilisation d'un affichage redondant (par ex. : taille et couleur) peut aider l'utilisateur à mieux percevoir certaines informations ou les distinctions entre les informations.

**Affichage distinguable :** les différents codes utilisés (ex., couleur, forme géométrique, taille) tout comme les différents niveaux d'un même code doivent être faciles à distinguer. Si on utilise 10 codes différents et plusieurs niveaux sur chaque code, il est probable que les utilisateurs auront vraisemblablement une grande difficulté à s'y retrouver.

**Affichage significatif :** un affichage doit utiliser autant que possible des codes qui sont significatifs pour les utilisateurs. La signification peut être inhérente à l'encodage tel que l'affichage d'une flèche courbée sur un panneau de signalisation de la route pour signifier un virage. La signification peut aussi être apprise telle que les feux rouge, jaune et vert pour les feux de circulation. C'est alors une convention acceptée par les utilisateurs ciblés.

**Standardisation de l'affichage :** lorsque le codage de l'information est utilisé par différentes personnes dans différentes situations, il est important d'utiliser un standard partout. Si un nouveau mode d'affichage est ajouté à un système qui utilise déjà plusieurs modes d'affichages, le nouveau mode d'affichage doit respecter celui qui est déjà en place.

Par ailleurs, la théorie de Gestalt nous apprend que l'humain perçoit le tout avant de percevoir les parties. Voici les six lois de la Gestalt [40] :

- Loi de la forme : l'œil regroupe les objets pour créer une forme simple, symétrique et stable ;

- Loi de la proximité : l'œil regroupe les objets proches les uns des autres ;
- Loi de la similitude : l'œil regroupe les objets semblables ;
- Loi de clôture : l'œil a plus de facilité à identifier une forme fermée comme une figure plutôt qu'une forme ouverte ;
- Loi de destin commun : l'œil identifie les objets en mouvements ayant la même trajectoire comme faisant partie d'une même forme ;
- Loi de la continuité : l'œil regroupe les objets qui partagent le même alignement.

De plus, la hiérarchie visuelle permet de mettre en évidence les éléments les plus importants. Pour ce faire, on peut utiliser différents artifices tels que la taille, la graisse de caractères, la couleur de l'objet, la couleur de l'arrière-plan, le soulignement, la police de caractères, le rapprochement, l'encadrement, etc. Il faut aussi tenir compte de l'emplacement des objets. Il faut toutefois limiter le nombre de moyens utilisés pour mettre en évidence les éléments de l'interface. En minimisant les modes de différenciation, l'interface sera plus facile à apprendre pour les utilisateurs [40].

**Groupement de l'information :** lorsque c'est possible, il est important d'utiliser des regroupements de l'information bien définis avec un nombre de groupes adéquat. S'il y a trop de groupes, il peut être difficile de retrouver l'information recherchée. Plus il y a de groupes, plus le temps requis pour retrouver l'information est long [41].

### **3.10.2 Encombrement visuel et densité d'information**

Plusieurs études ont défini l'encombrement visuel ou la densité de l'information en termes de nombre de symboles et d'éléments affichés à l'écran [42-44]. Ces définitions proviennent d'expériences psychologiques portant sur la perception et la prospection visuelle selon lesquelles le temps requis pour rechercher une information est proportionnel à la quantité de symboles à l'écran [45, 46]. En effet, l'affichage d'une quantité élevée de symboles à l'écran diminue l'efficacité de l'utilisateur lors de la recherche d'information puisque celui-ci doit y consacrer son attention de façon sérielle [44, 45, 47].

Pour évaluer l'encombrement visuel, il faut tenir compte du contexte d'utilisation et du niveau d'expertise de l'utilisateur avec le système. Il faut aussi prendre en considération la disposition des

informations à l'écran. En effet, un affichage dense, mais bien disposé peut-être bénéfique dans certaines situations [47-49].

Il existe plusieurs éléments qui contribuent à augmenter l'encombrement visuel : les attributs visuels et la qualité esthétique de l'interface [47].

Les attributs visuels sont composés de propriétés observables tels que la couleur, le contraste, la police de caractère, etc. Les études indiquent qu'il faut utiliser avec parcimonie les différents éléments visuels d'affichage. Il faut distinguer les informations les plus importantes par la taille ou la luminosité. L'abondance de stimulus visuels amplifie l'impression d'encombrement visuel. Par exemple, Mayhew [48] recommande de se limiter à huit couleurs d'affichage pour éviter une impression de surcharge pour l'utilisateur [50]. La densité de l'information peut être réduite en utilisant des abréviations appropriées, en évitant de présenter des informations inutiles, en utilisant des mots courts et des phrases concises, en utilisant un affichage avec des colonnes en groupant les informations [39].

La qualité esthétique est une propriété subjective qui caractérise la présentation de l'information de façon harmonieuse. L'équipe de Lavie et Tractinsky [51] a catégorisé l'esthétique d'une interface comme suit : esthétique classique et esthétique créative. L'esthétique classique réfère à la simplicité de l'affichage, la symétrie et l'organisation de l'interface. L'esthétique créative fait référence à la création d'effet de surprise et de nouveauté dans l'affichage. Moshagen et Thiels [50] ont catégorisé les facteurs de l'esthétique comme suit : simplicité, diversité, imagerie (colorfulness) et l'artisanat (craftsmanship).

Les éléments qui contribuent à augmenter la densité de l'information sont aussi à examiner à la lumière de leur pertinence pour la tâche. L'information qui est affichée de façon redondante (cela est souhaitable dans certains cas) ou qui n'est pas pertinente à la tâche crée une charge supplémentaire pour l'accomplissement de la tâche. Cette lourdeur est perçue comme un encombrement visuel.

Par ailleurs, l'organisation de l'information sous forme de hiérarchie ou de regroupements logiques peut réduire l'encombrement visuel. La théorie de la Gestalt offre des règles permettant de structurer l'information de façon à aider les utilisateurs à bien percevoir l'information affichée.

### 3.11 Synthèse

Dans ce chapitre, nous avons vu les différentes approches qui ont été étudiées par la communauté scientifique pour les interfaces de conception de requêtes à des bases de données relationnelles. Les besoins des utilisateurs du logiciel Croesus Conseiller requièrent beaucoup de souplesse dans l'interface utilisateur puisque les requêtes formulées utilisent une sémantique complexe impliquant des jointures et des groupes de conditions. De plus, les utilisateurs doivent avoir le contrôle sur la construction des différentes conditions, des jointures et des groupes de conditions. L'approche utilisera des contrôles de type formulaire (par ex., des boîtes combos, des champs de saisie ou des champs de dates), mais dans un format flexible qui permet d'ajouter autant de conditions, d'opérateurs logiques, de groupes de conditions et de jointures que nécessaire. Pour guider l'utilisateur dans la formulation de la requête et pour rendre la lecture plus facile, l'approche par diagramme sera utilisée comme source d'inspiration pour la nouvelle interface.

Par la suite, les différents irritants perçus par les utilisateurs lors de la conception de requêtes à des bases de données relationnelles ont été définis. De plus, les défis rencontrés par les utilisateurs lors de la création de requêtes à une base de données ont été catégorisés. Cette section a permis de confirmer que les problèmes rencontrés par les utilisateurs du logiciel Croesus Conseiller lors de la consultation de base de données relationnelle sont communs et connus de la communauté scientifique pour des utilisateurs qui n'ont pas de connaissance du langage SQL.

Les principes de l'interaction guidée ont ensuite été définis. Ces principes doivent être pris en considération lors de la conception de la nouvelle interface. L'utilisateur doit être guidé lors de l'ajout d'opérateurs logiques et de jointures puisque ces actions sont difficiles pour la plupart des utilisateurs.

Dans la dernière section, le chapitre s'intéresse à l'approche de conception centrée sur l'utilisateur ainsi qu'aux facteurs humains dans l'affichage de l'information. Le processus de conception centrée sur l'utilisateur défini par la norme ISO 9241-210 [1] sera utilisé pour la conception de la nouvelle interface. La solution qui sera produite doit tenir compte des facteurs humains : l'affichage doit être aéré et facile à lire et les utilisateurs doivent être en mesure de trouver facilement l'information importante.

L'interface actuelle de critères de recherche simplifiée utilise l'approche par filtre combinée avec une approche de formulation de phrase en langage semi-naturel structuré et rigide. L'interface guide la formulation de la phrase en offrant des verbes et des mots. La formulation s'effectue principalement à l'aide de la souris.

L'interface actuelle de critères de recherche avancée utilise une section formulaire pour la formulation des différents filtres. Elle combine donc l'approche par formulaire et l'approche par filtre. Pour les opérateurs logiques et les jointures, c'est une structure hiérarchique qui peut difficilement être comparée avec les approches qui ont été étudiées dans la revue de la littérature.

La nouvelle interface sera développée en combinant plusieurs approches. Les différentes conditions seront formulées à l'aide de formulaires selon l'approche par filtre. La création des jointures sera effectuée à l'aide d'un formulaire, mais l'affichage des jointures utilisera un système comparable à l'approche par diagramme. Les opérateurs logiques OU utiliseront aussi une approche par diagramme.

Pour respecter des contraintes de temps, le modèle logique d'entités utilisé par la nouvelle interface sera basé sur celui offert par l'interface de critères de recherche simplifiée. La formulation sera effectuée principalement à l'aide de la souris. Cette décision rend le développement du prototype plus facile et rapide. La solution pourrait toutefois être adaptée par la suite pour permettre une formulation à l'aide du clavier.

Cette approche est nommée : formulaire flexible de formulation de requêtes par diagramme.

### **3.12Hypothèses**

La recherche permet de poser les hypothèses suivantes :

- H0 (hypothèse nulle) : L'interface de formulaire flexible de formulation de requêtes par diagramme donne le même résultat que les deux interfaces actuelles de l'application Croesus Conseiller en termes de temps de formulation, de taux d'erreurs de formulation et de nombre de demandes d'assistances.
- H1 : L'interface de formulaire flexible de formulation de requêtes par diagramme nécessite moins de temps pour la formulation de requêtes à une base de données relationnelle que les deux interfaces actuelles de l'application Croesus Conseiller.

- H2 : L'interface de formulaire flexible de formulation de requêtes par diagramme provoque moins d'erreurs de formulation que les deux interfaces actuelles l'application Croesus Conseiller.
- H3 : L'interface de formulaire flexible de formulation de requêtes par diagramme engendre moins de demandes d'assistance que les deux interfaces actuelles de l'application Croesus Conseiller.

## **CHAPITRE 4 MÉTHODOLOGIE DE DÉVELOPPEMENT ET DE TEST DES MAQUETTES ET DU PROTOTYPE DE LA NOUVELLE INTERFACE**

Ce chapitre présente la méthodologie de développement et de test des maquettes et du prototype de formulaire flexible de formulation de requêtes par diagramme qui a été suivi dans ce projet.

### **4.1 Méthodologie de développement et de test**

Dans la première phase du projet, portant sur la conception et le développement des maquettes et d'un prototype de formulaire flexible de formulation de requêtes par diagramme d'interface, trois cycles de tests d'utilisabilité ont été effectués. Les tâches soumises aux utilisateurs lors des deux premiers cycles de tests sont basées sur des captures d'écrans de maquettes alors que les tâches du troisième cycle sont basées sur un prototype fonctionnel. Cette phase avait pour but de valider la solution envisagée pendant le développement du prototype de formulaire flexible de formulation de requêtes par diagramme. Les différents cycles de la phase 1 du projet de recherche sont décrits dans le chapitre 5.

Dans la deuxième phase du projet, un quatrième et dernier cycle de tests d'utilisabilité a été effectué. L'objectif était de comparer les performances humaines correspondant à l'utilisation des interfaces existantes et de l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme. Les résultats obtenus lors de ce dernier cycle sont présentés dans le chapitre 6.

### **4.2 Vue d'ensemble du développement et de l'évaluation du prototype de la nouvelle interface**

Le tableau 4.1 présente une vue d'ensemble des phases de développement du prototype de formulaire flexible de formulation de requêtes par diagramme et des cycles de tests d'utilisabilité qui ont été effectués durant ce projet.

Tableau 4.1 Vue d'ensemble des phases de développement et des cycles de tests d'utilisabilité.

<p><b>Phase 0 :</b></p>	<p>Analyse de contexte. Analyse des utilisateurs. Analyse des besoins.</p> <hr/> <p>Conception de maquettes d'interface.</p>
<p><b>Phase 1 :</b> Processus de conception centrée sur l'utilisateur</p>	<p><b>Cycle 1</b> Tests d'utilisabilité auprès d'employés de Croesus Finansoft. Tâches effectuées avec une maquette sur papier.</p> <hr/> <p>Amélioration des maquettes.</p> <hr/> <p><b>Cycle 2</b> Tests d'utilisabilité auprès d'utilisateurs réels de Croesus Conseiller. Tâches effectuées avec une maquette sur papier.</p> <hr/> <p>Développement d'un prototype fonctionnel.</p> <hr/> <p><b>Cycle 3</b> Tests d'utilisabilité auprès d'employés de Croesus Finansoft. Tâches effectuées à l'aide de l'interface du prototype fonctionnel.</p> <hr/> <p>Amélioration du prototype fonctionnel</p>
<p><b>Phase 2 :</b> Analyse comparative de la performance humaine avec trois différentes interfaces</p>	<p><b>Cycle 4</b> Tests d'utilisabilité auprès d'utilisateurs réels de Croesus Conseiller. Tâches effectuées avec l'interface du prototype fonctionnel.</p>



## **4.3 Premier et deuxième cycles de tests d'utilisabilité**

Cette section présente la méthodologie utilisée pour les deux premiers cycles de tests d'utilisabilité dans la phase 1 du projet.

### **4.3.1 Préparation des tests d'utilisabilité**

Afin de préparer les tests d'utilisabilité qui ont été réalisés dans le premier et le deuxième cycle, les tâches suivantes ont été accomplies par la chercheure :

- chercher des requêtes à une base de données relationnelle ayant été créée par des utilisateurs réels pour que les tâches de formulation de requêtes soumises aux utilisateurs lors des tests soient réalistes ;
- sélectionner des captures d'écrans ;
- préparer les valeurs possibles à présenter aux sujets pour leur permettre d'identifier les données qui sont retournées par la requête ;
- créer 12 permutations différentes des tests afin d'éviter d'avantager une interface par rapport aux autres par la position des questions dans le test ;
- imprimer les tests sur papier ;
- valider auprès des gestionnaires la possibilité d'effectuer le test avec les employés et identifier des sujets ;
- entrer en contact avec les sujets et identifier une plage horaire pour effectuer le test ;
- réserver une salle ;
- enregistrer et chronométrer les différentes tâches des tests.

### **4.3.2 Sujets**

Les sujets rencontrés lors du premier cycle n'étaient pas représentatifs des utilisateurs visés alors qu'ils l'étaient lors du deuxième cycle ; dans ce dernier cas, la chercheure s'est déplacée dans les lieux de travail pour rencontrer des utilisateurs réels de l'application Croesus Conseiller.

Lors du premier cycle, les tests d'utilisabilité ont été effectués avec des employés de Croesus Finansoft. Dix sujets ont été rencontrés, deux femmes et huit hommes âgés entre 30 et 50 ans : une

analyste de données, un analyste stratégique de recherche, une chef d'équipe, un analyste UX, un conseiller en formation, un analyste d'application et données et quatre analystes d'affaires. Leur niveau de scolarité va du niveau CÉGEP au niveau universitaire supérieur (ex., détenteur d'un diplôme d'études supérieures spécialisées). Le nombre d'années d'expérience avec le logiciel est de 3 mois pour un sujet, 6 mois pour un sujet, 1 an pour deux sujets, 2 ans pour un sujet, 3 ans pour un sujet, de 4 ans pour un sujet, de 5 ans pour un sujet, 6 ans pour un sujet et 18 ans pour un sujet.

Pour trouver des sujets, la chercheuse a eu recours l'équipe de direction de l'entreprise Croesus Finansoft. Cette équipe a formé une liste d'employés qui étaient intéressés et disponibles pour effectuer le test. La chercheuse a contacté les sujets pour valider leur intérêt et fixer une date et un lieu pour la rencontre.

Lors du deuxième cycle de test d'utilisabilité, 12 sujets qui sont des utilisateurs réguliers de l'application Croesus Conseiller ont participé à l'étude. Il s'agit de 10 femmes et deux hommes dont l'âge varie entre 24 ans et 56 ans (moyenne de 44 ans). Il y avait sept adjointes en placement et cinq analystes d'application et de données qui utilisaient l'application Croesus Conseiller depuis un certain nombre d'années (trois sujets depuis moins de 1 an, deux sujets depuis 3 à 5 ans, deux sujets depuis 5 à 10 ans et 5 sujets depuis plus de 10 ans).

Le niveau de scolarité des sujets va du secondaire à la maîtrise (un sujet de niveau secondaire, cinq sujets de niveau collégial, un sujet ayant un certificat de niveau universitaire, quatre sujets ayant un baccalauréat et un sujet ayant une maîtrise).

Pour trouver des sujets, la chercheuse a eu recours au centre de services aux clients de l'entreprise Croesus Finansoft. Ceux-ci ont désigné une liste d'employés qui étaient intéressés et disponibles pour effectuer le test. Une employée de ce centre s'est chargée d'identifier et de contacter les personnes pour leur demander si elles seraient intéressées et disponibles pour participer à la recherche. Par la suite, la chercheuse a contacté les sujets pour valider leur intérêt et fixer une date et un lieu pour la rencontre.

### **4.3.3 Tâches**

Les tests d'utilisabilité du premier et du deuxième cycle comprennent cinq séries de trois tâches. La première série de trois tâches permet au sujet de se familiariser avec le format des tâches du test. Cette série utilise une représentation graphique sous forme de diagramme de tâche pour

représenter les requêtes. Le sujet se fait présenter une série de trois tâches pour chacune des trois interfaces de formulation de requêtes actuelles et pour le prototype d'interface qui a été développé. Pour chaque tâche, l'utilisateur doit identifier le ou les enregistrements qui seront retournés parmi une liste de 11 enregistrements.

#### **4.3.4 Ordre d'utilisation des interfaces**

Pour éviter d'avantager une interface par rapport aux autres, il est important de contrôler l'ordre d'utilisation des interfaces. En effet, les sujets prennent de l'expérience au fur et à mesure de l'avancement des tests. Le sujet est susceptible d'avoir plus de facilité à utiliser la dernière interface que la première. Nous avons donc utilisé la méthode du carré latin [52] afin de nous assurer que l'ordre d'utilisation de chaque interface soit réparti de façon équitable durant le test d'utilisabilité. Pour un test effectué avec quatre interfaces, il existe 24 différentes permutations possibles. Étant donné que le test est effectué avec 12 sujets, 12 permutations différentes seront présentées aux différents sujets.

Les quatre interfaces ont été numérotées ainsi :

- 1 – la fonctionnalité des filtres
- 2 – l'interface des critères de recherche simplifiée
- 3 – l'interface des critères de recherche avancée
- 4 – l'interface de formulation de requêtes du prototype de formulaire flexible de formulation de requêtes par diagramme

Le tableau 4.2 présente l'ordre des interfaces utilisées pour faire les 12 tâches du test. Chaque interface doit se retrouver trois fois dans chacune des quatre positions.

Tableau 4.2 Ordre des interfaces utilisées pour faire les 12 tâches.

<b>Ordre d'utilisation des interfaces</b>
1-2-3-4
1-3-2-4
1-4-2-3
2-1-4-3
2-3-1-4
2-4-1-3
3-1-4-2
3-2-4-1
3-4-2-1
4-1-3-2
4-2-3-1
4-3-1-2

### **4.3.5 Procédure des tests**

La durée du test est d'environ une heure par sujet. Le test se déroule dans une salle située dans le lieu de travail du sujet. Un sujet à la fois est présent avec la chercheuse dans la salle. Au début de chaque rencontre, la chercheuse se présente et explique le contexte du projet de recherche ainsi que le déroulement des tests. La conversation qui se déroule entre le sujet et la chercheuse durant l'exécution du test est enregistrée à l'aide d'un téléphone cellulaire. Chaque tâche est chronométrée à l'aide d'un téléphone intelligent.

Avant de commencer le test, le sujet est informé qu'il peut abandonner un numéro s'il le juge trop difficile. Il lui est recommandé de ne pas parler durant l'exécution des tests, sauf en cas de question parce que la durée de chaque test est chronométrée. Entre chaque test, une pause permet au sujet de faire des commentaires s'il y a lieu.

Le test se divise en trois étapes. La première étape permet de recueillir des données démographiques sur chaque sujet : âge, sexe, occupation, niveau de scolarité (échelle de Likert), expérience avec le produit Croesus Conseiller (même échelle), expérience en finance (même échelle), expérience avec la technologie (même échelle), fréquence d'utilisation d'un outil de recherche (même échelle).

À la deuxième étape, des captures d'écrans contenant une requête déjà formulée sont présentées au sujet. Quatre interfaces différentes sont présentées. Pour chaque interface, trois captures d'écrans sont présentées pour un total de 12 captures d'écrans. Pour chaque capture d'écran, on présente une grille qui représente la base de données. La grille contient toujours 11 enregistrements. L'utilisateur doit interpréter la requête et indiquer quels enregistrements seront retournés par la requête selon sa compréhension.

La troisième étape consiste à placer les quatre interfaces en ordre de préférence et à donner un commentaire positif et un commentaire négatif pour chaque interface. Pour terminer, le sujet peut faire d'autres commentaires s'il y a lieu.

## **4.4 Troisième et quatrième cycles de tests d'utilisabilité**

Cette section présente la méthodologie utilisée pour le troisième et le quatrième cycle de conception du prototype de formulaire flexible de formulation de requêtes par diagramme.

### **4.4.1 Préparation des tests d'utilisabilité**

Afin de préparer les tests d'utilisabilité qui ont été exécutés dans le troisième et le quatrième cycle, les tâches suivantes ont été exécutées par la chercheuse :

- trouver des requêtes ayant été créées par des utilisateurs réels pour que les tâches de formulation de requêtes soumises aux utilisateurs lors des tests soient réalistes ;
- sélectionner des captures d'écrans ;
- préparer les valeurs possibles à présenter aux sujets pour leur permettre d'identifier les données qui sont retournées par la requête ;
- créer 12 permutations différentes des tests afin d'éviter d'avantager une interface par rapport aux autres par la position des questions dans le test ;

- imprimer les tests sur papier ;
- valider auprès des gestionnaires la possibilité d'effectuer le test avec les employés et identifier des sujets ;
- entrer en contact avec les sujets et identifier une plage horaire pour effectuer le test ;
- réserver une salle ;
- enregistrer et chronométrer les différentes tâches des tests.

#### **4.4.2 Sujets**

Les sujets rencontrés lors du troisième cycle n'étaient pas représentatifs des utilisateurs visés alors que ceux du quatrième cycle l'étaient ; dans ce dernier cas, la chercheuse s'est déplacée pour rencontrer des utilisateurs réels de l'application Croesus Conseiller.

Lors du troisième cycle de tests d'utilisabilité, quatre hommes et deux femmes âgés entre 30 et 55 ans ont participé aux tests d'utilisabilité du troisième cycle de conception du prototype de formulaire flexible de formulation de requêtes par diagramme. Ils occupent les différents postes suivants : directeur de recherche et développement, analyste en assurance qualité, analyste de données, spécialiste support aux ventes et produits, analyste fonctionnel et un chercheur en innovations technologiques. Leur scolarisation varie du niveau baccalauréat à celui de doctorat. Le nombre d'années d'expérience avec le logiciel était de 6 mois pour un sujet, de 2 ans pour un sujet, 7 ans pour un sujet, de 16 ans pour un sujet et de 18 ans pour deux sujets.

Lors du quatrième cycle de tests d'utilisabilité sept femmes et cinq hommes âgés entre 18 et 60 ans ont participé aux tests d'utilisabilité du quatrième cycle qui fait partie de la phase 2 du projet de recherche. Ils occupent les différents postes suivants : adjointe en placement (six sujets), analyste application et données (quatre sujets), analyste principal applicatif (un sujet) et directeur (un sujet). Leur scolarisation va du niveau secondaire au niveau du baccalauréat (deux sujets de niveau secondaire, trois sujets de niveau collégial, six sujets de niveau baccalauréat, et un sujet a un certificat de niveau universitaire). Le nombre d'années d'expérience avec le logiciel varie entre de 1 à 3 ans pour deux sujets, 5 à 10 ans pour 4 sujets, 3 à 5 ans pour un sujet et il est de plus de 10 ans pour 7 sujets.

Les sujets ayant participé au troisième cycle de tests sont tous différents des sujets ayant participé au quatrième cycle de tests. Pour trouver les sujets, la chercheuse a eu recours au Centre de services aux clients de l'entreprise Croesus Finansoft. Ce centre a établi une liste d'employés qui étaient intéressés et disponibles pour effectuer les tests d'utilisabilité. La chercheuse s'est ensuite chargée de contacter les personnes pour fixer une date et un lieu pour la rencontre.

### **4.4.3 Tâches**

Le test comprend deux tâches de familiarisation qui ne sont pas considérées dans le résultat ainsi que trois tâches dont les résultats comptent. Les mêmes tâches doivent être exécutées avec les deux interfaces de formulation de critères de recherche (simplifiée et avancée) ainsi qu'avec le prototype de formulaire flexible de formulation de requêtes par diagramme qui a été développé dans le cadre de la recherche. Il y a donc un total de neuf tâches à exécuter.

Pour chaque tâche, le critère de recherche attendu est présenté textuellement de façon structurée en français. L'utilisateur doit le créer à l'aide de l'interface.

### **4.4.4 Ordre d'utilisation des interfaces**

Pour éviter d'avantager une interface utilisateur par rapport aux autres, il est important de contrôler leur ordre d'utilisation. En effet, les sujets peuvent devenir plus efficaces au fur et à mesure de l'avancement des tests. La dernière interface utilisée pourrait donc être avantagée. Nous avons utilisé la méthode du carré latin [52] pour nous assurer que le positionnement de chaque interface dans le test soit réparti équitablement. Pour un test effectué avec trois interfaces, il existe six différentes permutations possibles. Étant donné que le test est effectué sur 12 sujets, 12 permutations seront présentées. Il y a donc deux utilisateurs pour chaque permutation possible.

Les trois interfaces ont été identifiées par une lettre :

A – l'interface de critères de recherche simplifiée

B – l'interface de critères de recherche avancée

C – l'interface de formulation de requêtes du prototype de formulaire flexible de formulation de requêtes par diagramme

Le tableau 4.3 présente les permutations utilisées pour les 12 tâches du test. Chaque interface doit se retrouver à chacune des positions quatre fois.

Tableau 4.3 Ordre des interfaces pendant le quatrième cycle de tests.

<b>Ordre des interfaces utilisées</b>
C-A-B
A-C-B
B-A-C
A-B-C
C-A-B
B-A-C
B-C-A
C-B-A
B-C-A
C-B-A
A-B-C
A-C-B

#### **4.4.5 Procédure des tests**

La durée du test varie entre une heure et une heure 30 minutes par sujet. Le test se déroule dans une salle située dans le lieu de travail du sujet. Un sujet à la fois est présent avec la chercheuse dans la salle. Au début de chaque rencontre, la chercheuse se présente et explique le contexte du projet de recherche de même que le déroulement des tests. La conversation qui se déroule durant l'exécution des tâches ainsi que les actions effectuées à l'écran sont enregistrées à l'aide de l'application Camtesia. La durée du test est chronométrée à l'aide d'un téléphone intelligent. Le test est effectué sur un ordinateur de type PC.

Le test se divise en quatre étapes. La première étape permet de recueillir des données démographiques sur chaque sujet : âge, sexe, occupation, niveau de scolarité, expérience avec le produit Croesus Conseiller, expérience en finance, expérience avec la technologie, fréquence



d'utilisation d'un outil de recherche. Toutes les données, sauf le sexe et l'occupation, ont été recueillies avec une échelle de Likert.

À la deuxième étape, le sujet se fait présenter des tâches de familiarisation. La première tâche est effectuée par la chercheuse et permet d'expliquer le fonctionnement de l'interface. C'est le moment pour l'utilisateur de poser des questions s'il y a lieu. La deuxième tâche de familiarisation est effectuée par le sujet avec l'assistance de la chercheuse. Ces deux tâches ne font pas partie des résultats de la recherche.

À la troisième étape, on demande à l'utilisateur d'effectuer trois tâches de façon autonome. La chercheuse n'intervient pas pour aider l'utilisateur sauf si celui-ci le demande spécifiquement. Le sujet a la possibilité de demander de l'aide lorsqu'il le juge nécessaire. La chercheuse répond à la question et note le nombre de fois que le sujet demande de l'aide. Les mêmes tâches de familiarisation et tâches réelles sont effectuées à répétition pour chacune des interfaces.

La quatrième étape consiste à demander au sujet de placer les trois interfaces en ordre de préférence et de faire des critiques positives et négatives au sujet des différentes interfaces.

## **CHAPITRE 5 CONCEPTION CENTRÉE SUR L'UTILISATEUR**

Ce chapitre présente les différentes activités effectuées pour la conception du prototype de formulaire flexible de formulation de requêtes par diagramme. L'élaboration de la solution et du prototype fonctionnel a été effectuée en respectant les étapes définies dans le processus de conception centrée sur l'utilisateur de la norme ISO 9241-210 [1]. Trois cycles de conception centrée sur l'utilisateur ont été effectués. Les deux premiers cycles ont permis de produire des captures d'écrans de la solution envisagée. Le troisième cycle a été effectué avec la première version du prototype fonctionnel. Les trois premiers cycles étaient préliminaires à la phase 2 du projet de recherche qui comprend une quatrième session de tests d'utilisabilité qui sera présentée dans le chapitre suivant.

Cette section décrit les étapes d'analyse et de conception qui ont mené à la production du prototype de formulaire flexible de formulation de requêtes par diagramme.

### **5.1 Premier cycle de conception - Étape 1 – Analyse des utilisateurs et du contexte**

#### **5.1.1 Utilisateurs visés**

Les utilisateurs visés par la nouvelle interface ne sont pas familiers avec la programmation ni avec le langage SQL. On peut les diviser en deux groupes. Le premier groupe inclut des utilisateurs qui sont très peu à l'aise avec les moteurs de recherche qui permettent d'utiliser des sémantiques complexes. Ces utilisateurs n'utilisent le système de requête de Croesus Conseiller que très rarement. Généralement, ils demandent de l'assistance ou copient un critère de recherche existant pour le modifier légèrement ensuite. Le second groupe est composé d'utilisateurs ayant une grande maîtrise des données et qui ont des besoins très poussés. Certains de ces utilisateurs ont pour tâche d'assurer la conformité des transactions financières effectuées par les membres de leur équipe par rapport aux lois en vigueur au Canada pour le respect du profil d'investisseur de leurs clients. Ceux-ci utilisent les fonctions les plus avancées pour la création de requêtes. Leur objectif est d'automatiser une partie de leurs tâches d'identification de situations anormales dans les transactions financières effectuées sur le marché boursier.

### **5.1.2 Contexte d'utilisation**

Les besoins des utilisateurs en matière de critères de recherche varient en fonction de la période de l'année. Par exemple, pendant la période des REER, les besoins ne sont pas les mêmes que durant la période des impôts. Les années sont divisées en périodes et le cycle est similaire d'année en année. Les utilisateurs se créent habituellement un coffre à outils de critères de recherche qui sont réutilisés de façon cyclique. D'ailleurs, ils peuvent partager les critères de recherche entre eux. Ils ont donc rarement l'occasion de créer des critères de recherche. L'interface est difficile à apprendre et les utilisateurs oublient le fonctionnement d'une utilisation à l'autre.

Dans plusieurs cas, lorsqu'ils ont un besoin spécifique, ils préfèrent filtrer la liste manuellement plutôt que d'utiliser le système de requêtes.

## **5.2 Premier cycle de conception - Étape 2 – Spécification des besoins**

### **5.2.1 Schéma de la base de données**

Pour des raisons de sécurité des données et pour rendre l'utilisation plus facile, le schéma de la base de données ne doit pas être exposé à l'utilisateur. Ce dernier doit plutôt avoir accès à un schéma logique d'éléments qui va l'aider à se construire le modèle mental que les utilisateurs devraient avoir des données du système [14]. Le schéma est exposé à travers les listes de champs et de jointures offerts pour construire les conditions simples. Le modèle logique d'entités qui est exposé est basé sur celui qui est offert pour la formulation de critères de recherche par les interfaces actuelles. En effet, l'objectif du projet est de développer une interface de formulation de requêtes qui permet de formuler des requêtes avec le même niveau de complexité et les mêmes résultats que les interfaces actuelles.

### **5.2.2 Besoins en affichage**

Pour répondre aux exigences de la tâche, le prototype de formulaire flexible de formulation de requêtes par diagramme doit prévoir différents affichages et satisfaire certaines conditions :

- l'affichage pour des conditions simples ayant des champs de plusieurs types différents : chaîne de caractères alphanumériques, valeur numérique, date, valeur booléenne, valeur provenant d'un dictionnaire de valeurs ;

- l'utilisation d'opérateurs de base et d'autres, plus complexes : égal, différent de, plus grand que, plus petit que, contenant, ne contenant pas, est entre, dans les derniers X jours, parmi (liste de valeurs sélectionnées), est nul, n'est pas nul, débute par, ne débute pas par, etc. ;
- l'affichage de conditions complexes utilisant les opérateurs logiques ET/OU ;
- l'affichage de groupements de conditions ;
- l'affichage de conditions avec des jointures sur plusieurs éléments, ce qui implique des relations zéro à n : dont tous (jointure sur le nombre d'éléments), dont au moins un (jointure positive), dont aucun (jointure négative).

Pour réduire le nombre d'étapes et aider l'utilisateur à mieux comprendre, il est préférable de réduire l'utilisation des opérateurs logiques OU en offrant la possibilité de spécifier plusieurs valeurs pour un même champ à l'intérieur d'une même condition simple à l'aide d'un opérateur « parmi » ou « excluant » et en permettant l'opérateur « est entre ».

### **5.3 Premier cycle de conception - Étape 3 – Production d'une solution sous forme de captures d'écrans**

#### **5.3.1 Choix du type d'interface**

La nouvelle interface de formulation de requêtes est basée sur une approche dérivée de l'approche par formulaires flexibles (« form-based ») proposée par Jagadish jumelée à l'approche par diagramme proposée par Jarrar. Ainsi chaque condition simple utilise des champs de formulaires tels que des boîtes de combo, des champs de saisie de date, des champs d'édition de texte, etc. L'ajout des conditions simples est basé sur l'approche par filtres. Toutefois, l'utilisateur peut contrôler la structure de la requête en ajoutant les conditions simples, les opérateurs logiques et les jointures selon son besoin.

L'approche d'interface de formulation de requêtes sur une base de données relationnelle qui est développée dans le cadre de cette recherche se nomme : formulaire flexible de formulation de requêtes par diagramme.

### 5.3.2 Solution

Pour respecter la convention établie par la fonctionnalité des filtres sur des listes d'éléments, les conditions sont affichées dans un encadré de couleur orangée. C'est le principe d'utilisation d'un standard qui a été décrit dans la revue de littérature [39].

Pour distinguer les conditions simples par rapport aux groupements et aux jointures, une couleur différente est utilisée pour chaque type d'élément de structure de la requête. L'objectif est d'attirer le regard sur les groupements qui sont des éléments importants d'une requête. Cette décision est basée sur la revue de littérature qui décrit différents modes d'encodages [39]. Les groupements ont la même utilité logique que les parenthèses d'une requête SQL, mais ils permettent de visualiser d'un coup d'œil rapide les conditions qui font partie du groupement grâce à un encadré qui apparaît autour des conditions simples qui font partie du groupement. Par ailleurs, la couleur bleue est déjà utilisée dans l'application Croesus Conseiller pour identifier les critères de recherche lorsqu'ils sont appliqués pour filtrer une liste d'éléments. Le choix de la couleur respecte donc le principe d'utilisation d'un standard pour respecter une convention [39].

L'utilisation de caractères gras pour le libellé des champs et pour les valeurs des conditions améliore la lisibilité. C'est un encodage redondant utilisant la couleur de l'arrière-plan et la graisse des polices de caractères [39].

Par ailleurs, dans le but de réduire la densité de l'affichage [39] des conditions si possible, les opérateurs mathématiques sont utilisés plutôt que leur équivalent textuel : par ex. :  $>$ ,  $<$ ,  $=$ , etc. La même logique est appliquée pour l'affichage des opérateurs logiques ET et OU. Dans l'interface développée pour le prototype, les opérateurs logiques ET et OU sont affichés seuls (sans phrase explicative). Cette façon d'afficher les opérateurs logiques permet de réduire la densité de l'information.

Dans le but de réduire la densité de l'information, les boutons et contrôles qui permettent de modifier la requête sont affichés sous forme d'icônes.

Pour réduire la complexité liée aux opérateurs logiques OU, des conditions simples basées sur des champs de type « valeur d'un dictionnaire de valeurs » peuvent utiliser l'opérateur « parmi » ou « excluant » avec une liste de valeurs possibles. Cela remplace une série de conditions simples séparées par des opérateurs logiques OU. De plus, il est possible d'utiliser l'opérateur « est entre »

pour éviter de créer une condition avec l'opérateur  $>$  et une autre avec l'opérateur  $<$  et l'opérateur logique ET. Cette façon d'afficher l'information permet de grouper l'information ce qui respecte les règles d'affichage de l'information présentée par Sanders et McCormick [39]. De plus, on évite l'information redondante dans le but de réduire la densité de l'information.

La figure 5.1 présente un exemple de requête affichée dans l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme qui a été développé dans le cadre de ce projet. L'exemple permet de distinguer les éléments suivants :

- les conditions simples sont présentées dans des encadrés de couleur orangée ;
- les groupes de conditions sont présentés sous forme d'encadrés de couleur bleue ;
- le champ « type » présente un opérateur « parmi » ou « excluant » suivi d'une liste de valeurs ;
- le champ « date de transaction » utilise l'opérateur « entre » avec deux dates ;
- les opérateurs mathématiques « = », « < » et « > » sont utilisés pour l'affichage en remplacement du texte « est plus grand que » ou « est supérieur à ».

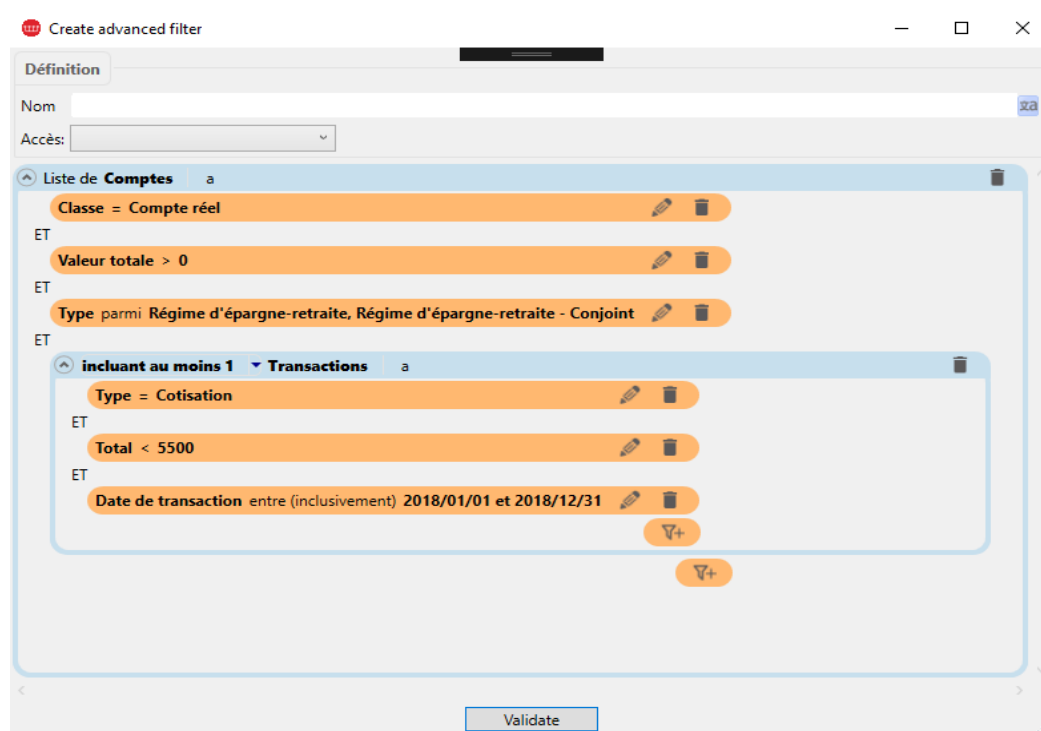


Figure 5.1 Interface de formulation de requêtes provenant du prototype de formulaire flexible de formulation de requêtes par diagramme.

## **5.4 Premier cycle de conception - Étape 4 – premier cycle de tests d'utilisabilité**

Ce cycle de tests d'utilisabilité a pour but de valider le plan de tests prévus pour le second cycle qui devait être effectué avec des utilisateurs réels de l'application Croesus Conseiller. Le premier cycle est donc préliminaire. Cette étape a permis d'ajuster le format de présentation, le nombre et le niveau de difficulté des tâches pour s'assurer que le test puisse être effectué dans une période d'environ une heure.

Les tâches effectuées étaient réparties sur quatre interfaces différentes : l'interface des filtres sur des listes de valeurs, les critères de recherche avancés, les critères de recherches simplifiés et l'interface qui a été produite dans le cadre de la présente recherche. Cela a permis de comparer le comportement et les réactions des sujets par rapport aux différentes interfaces.

Durant ce cycle, les commentaires et les observations des sujets ont permis d'apporter des améliorations à la solution présentée.

## **5.5 Premier cycle de conception - Étape 5 - Analyse des résultats suite aux tests d'utilisabilité préliminaires**

### **5.5.1 Évaluer la connaissance des sujets sur les opérateurs utilisés pour les ensembles**

Les premiers 10 sujets ont permis d'évaluer le niveau de connaissance des symboles mathématiques liés aux ensembles. Par exemple les symboles suivants «  $\exists$  » pour signifier l'existence d'une instance dans un ensemble, «  $\nexists$  » pour signifier la non-existence d'une instance dans un ensemble, «  $\forall$  » pour signifier tous les éléments d'un ensemble. Deux sujets ont mentionné avoir déjà vu les symboles dans leur formation académique, mais que c'était trop lointain pour se souvenir de leur signification. Les autres sujets n'avaient aucune connaissance de ces symboles. Cette étape a permis d'éliminer l'utilisation de ces symboles dans une solution potentielle pour l'affichage et la manipulation des jointures.



### 5.5.2 Opérateurs utilisés pour les conditions simples

Certaines conditions utilisent des opérateurs mathématiques qui ne sont pas bien compris par les sujets rencontrés. Par exemple le symbole « ~ » était utilisé pour signifier « contient » comme dans l'exemple « nom complet du client ~ Tremblay ». Plusieurs sujets ont bloqué sur ce symbole. Le test a donc été changé pour utiliser le libellé « contient » en remplacement du symbole « ~ ».

Dans le même ordre d'idée, un utilisateur a été bloqué par le symbole « < > » qu'il a interprété comme étant « égal » plutôt que « différent ». L'interface a donc été modifiée très tôt dans le processus pour utiliser le symbole suivant : « ≠ » dans certaines conditions et l'opérateur « < > » dans d'autres. Les sujets ont été sondés pour connaître leur préférence et c'est le symbole « < > » qui a été retenu.

## 5.6 Deuxième cycle de conception

Le deuxième cycle de conception a été effectué en respectant la même procédure et en utilisant les mêmes tâches que lors du premier cycle. Les tests ont été effectués auprès d'utilisateurs ayant le profil cible pour la fonctionnalité.

La section suivante présente un résumé des observations et commentaires recueillis lors des tests d'utilisabilité effectués dans le cadre de phase d'analyse et conception de la solution envisagée. Par la suite, les changements effectués pour améliorer l'interface seront expliqués.

### 5.6.1 Modèle logique

Certains sujets ont eu de la difficulté à comprendre le format des données proposées pour le résultat de la requête présenté sous forme de tableau. La représentation visuelle des données utilisée dans les tâches est très différente de l'affichage auquel ils sont habitués. La chercheuse a dû donner des explications pour que les sujets comprennent bien l'information. Il est très important que le sujet ait un modèle mental de la structure des données auxquelles la requête est appliquée et des relations « 1 à plusieurs » entre les éléments du modèle. Ça ne semble pas naturel pour plusieurs d'entre eux.

### **5.6.2 Affichage des jointures et groupements**

Plusieurs sujets (N = 8) ont effectué les tests sans tenir compte de l'information affichée dans les encadrés bleus dans l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme. Cette information concerne la jointure et est importante pour bien interpréter la requête présentée. En effet, dans certains cas, la tâche comporte une jointure négative, cela inverse donc le résultat. Cette problématique semble être causée par l'apparence des encadrés bleus. En effet, l'image est présentée comme si c'était l'entête d'une fenêtre MS Windows. Les sujets ont l'habitude de considérer l'entête comme une information de moindre importance. L'affichage devra donc être corrigé.

### **5.6.3 Encadrés orangés pour les conditions simples**

Deux des sujets rencontrés ont mentionné que la couleur choisie était agressive. La couleur a été choisie pour respecter la convention déjà en place dans la fonctionnalité des filtres sur des listes de valeurs. Pour cette raison, la couleur ne sera pas changée dans le prototype de formulaire flexible de formulation de requêtes par diagramme.

Par ailleurs, les sujets devaient faire un effort pour lire chaque élément affiché dans les listes de valeurs. Le fait d'afficher les éléments de la liste l'un après l'autre séparé d'une virgule rend la lecture plus difficile. Le prototype a donc été modifié pour que les valeurs de la liste soient affichées sur des lignes distinctes.

### **5.6.4 Affichage des opérateurs**

Tous les sujets rencontrés ont mentionné préférer l'affichage des opérateurs en format mathématique plutôt qu'en format textuel tel que c'est utilisé dans les interfaces actuelles. Ce mode d'affichage sera conservé.

## **5.7 Améliorations suite au deuxième cycle**

Suite au deuxième cycle de tests, certains changements ont été apportés à l'interface. La figure 5.2 présente l'interface suite aux changements effectués en tenant compte des commentaires recueillis pendant la première série de tests d'utilisabilité.

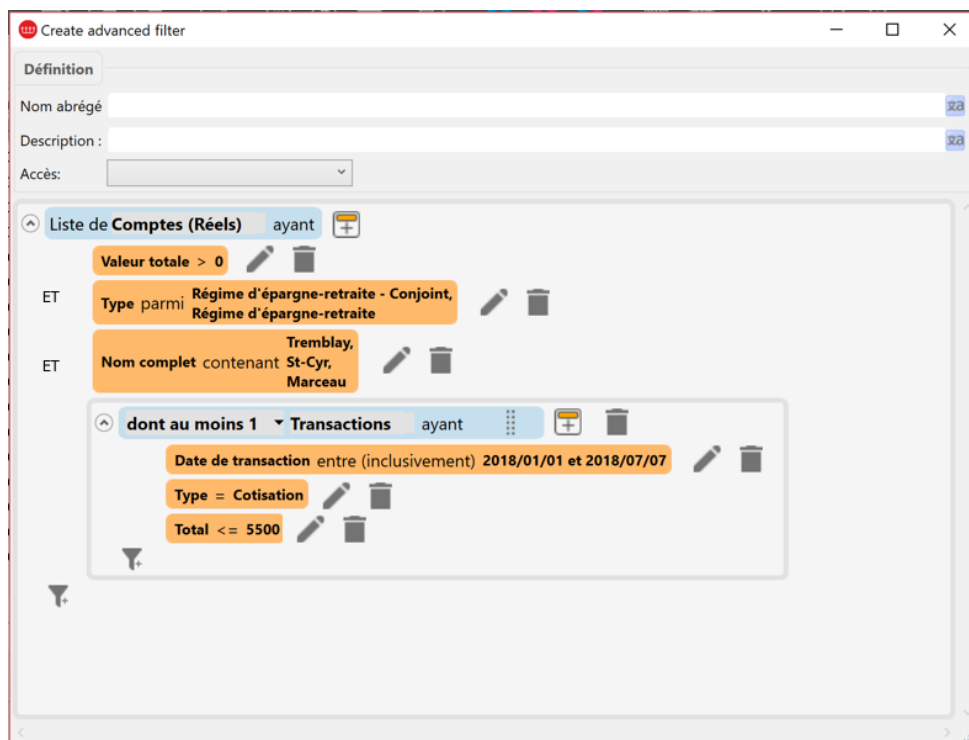


Figure 5.2 Interface de formulation de requêtes provenant du prototype suite aux changements apportés après les tests d'utilisabilité de la phase 1.

Premièrement, l'encadré orangé est moins large. Il se limite à la largeur de la condition qu'il contient. L'objectif est de réduire l'encombrement visuel.

Deuxièmement, les valeurs contenues dans une liste sont affichées sur des lignes distinctes. L'objectif étant de rendre la lecture plus facile.

Pour finir, l'encadré bleu qui permet d'afficher les jointures et les groupes de conditions a été modifié pour moins ressembler à l'entête d'une fenêtre. De plus, les combos qui s'affichent à l'intérieur de l'encadré sont de couleur différente pour attirer le regard et pour permettre de comprendre qu'il est possible de changer les valeurs.

## 5.8 Troisième cycle de conception

Le troisième cycle de conception et de test d'utilisabilité a été effectué à l'aide d'un prototype fonctionnel. Ce cycle avait pour objectif de tester la stabilité du prototype ainsi que son utilisabilité.

### 5.8.1 Analyse des résultats des tests d'utilisabilité préliminaires

Les sujets avaient de la difficulté avec la liste des opérateurs disponibles pour les conditions simples. Le menu affiché présentait l'information de façon désorganisée tel qu'on peut le voir dans la figure 5.3.

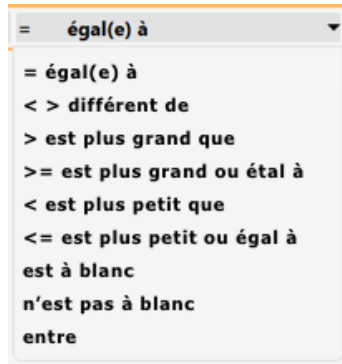


Figure 5.3 Menu contenant la liste des opérateurs disponibles pour les conditions simples avant l'amélioration.

La figure 5.4 démontre la nouvelle façon d’afficher les opérateurs dans le menu déroulant.

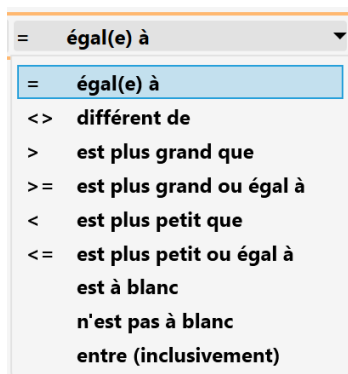


Figure 5.4 Menu contenant la liste des opérateurs disponibles pour les conditions simples après l’amélioration.

Par ailleurs, les sujets ont eu de la difficulté avec les groupements de conditions et les opérateurs logiques ET/OU. En effet, ils devaient appuyer sur le bouton « + » qui apparaît tout en haut de la fenêtre. De plus, il n’était pas possible de modifier l’opérateur sélectionné. Le sujet devait effacer le groupement et recommencer. Pour améliorer cet aspect de l’interface, la création de groupements a été automatisée. Il est possible de changer l’opérateur à l’aide d’une boîte combo. De plus, lorsqu’un utilisateur change l’opérateur logique, la condition précédente et la suivante ou la nouvelle sont automatiquement regroupées dans un nouvel encadré. De cette façon, il n’est plus nécessaire d’utiliser le bouton « + » et il y a moins de manipulation nécessaire.

L’ajout de jointure était aussi effectué à l’aide du bouton + qui se trouve tout en haut de la condition. Une fois que le groupe apparaît, l’utilisateur devait sélectionner le type de jointure dans la boîte combo affichée dans l’entête du groupement. L’ajout de jointure nécessitait donc deux opérations peu intuitives. Pour simplifier, une option a été ajoutée dans le menu qui offre la liste des champs pour la création de conditions simples. La figure 5.5 permet de voir l’option qui a été ajoutée sous forme de sous-menu. L’option « Liaison entre Compte et... » offre tous les types de jointures : client, position, relation et transactions.

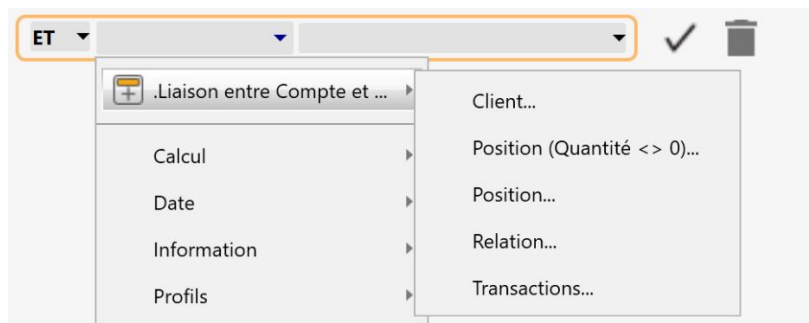


Figure 5.5 Menu contenant la liste des opérateurs disponibles pour les conditions simples.

D'autre part, les sujets ont mentionné que l'affichage de l'interface était encore trop dense. Pour alléger l'affichage, les icônes qui permettent de modifier ou de supprimer des conditions ont été cachées. Elles apparaissent lorsque le curseur de la souris se trouve sur la condition. La figure 5.5 présente l'interface suite aux changements de la troisième série de tests d'utilisabilité.

Pour finir, les utilisateurs ont mentionné le désir de pouvoir spécifier plusieurs valeurs pour les champs de type chaîne de caractères, autrement dit, ils désirent pouvoir spécifier plusieurs conditions en une seule. Par exemple, le critère suivant : « Liste des clients ayant nom contenant « Tremblay » ou ayant nom contenant « St-Cyr » ou ayant nom contenant « Corriveau » », deviendrait « Liste des clients ayant le nom contenant « Tremblay, St-Cyr, Corriveau » ». Cette façon de traiter la condition permet de réduire le nombre d'opérations et de faciliter la lecture de l'information par la suite. De plus, elle permet de réduire l'utilisation de l'opérateur logique OU.

La figure 5.6 présente l'interface du prototype suite aux changements apportés lors du troisième cycle de tests d'utilisabilité.

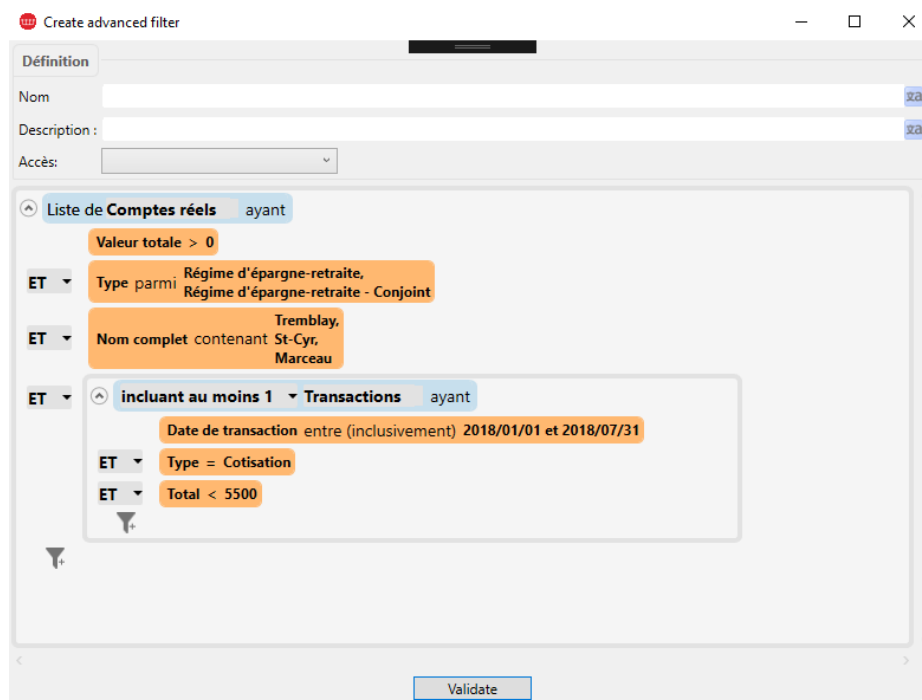


Figure 5.6 Interface de formulation de requêtes provenant du prototype suite aux changements apportés après les tests d'utilisabilité de la phase 2.

De plus, afin d'aider les sujets à comprendre le modèle logique des entités, le graphique présenté dans la figure 5.7 a été ajouté aux explications qui sont données par la chercheuse aux sujets avant de commencer le test d'utilisabilité.

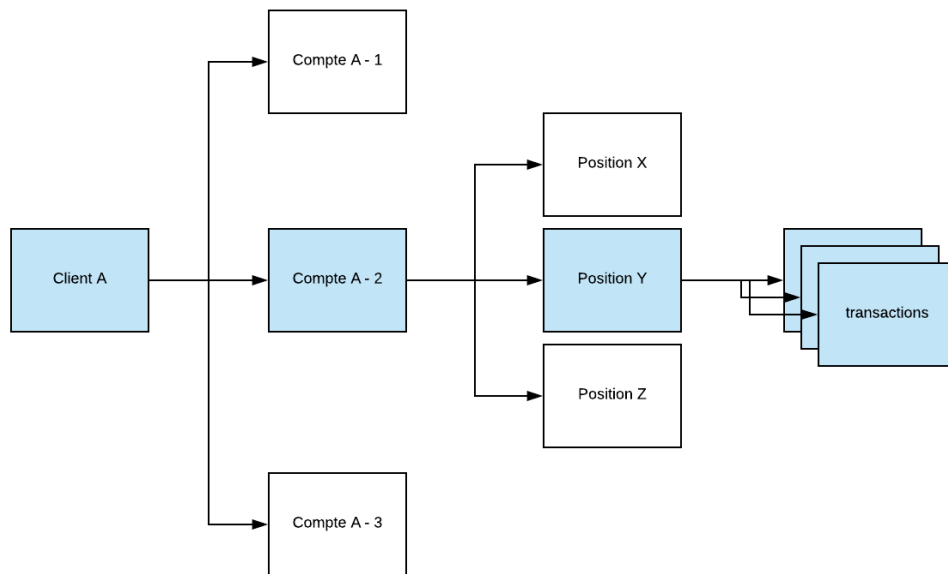


Figure 5.7 Graphique qui présente le modèle logique des éléments impliqués dans la formulation des requêtes.

## 5.9 Conclusion

Le processus de conception centré sur l'utilisateur qui a été suivi pour la conception du prototype d'interface a permis de régler plusieurs problèmes d'utilisation et d'apporter plusieurs améliorations à la solution envisagée au départ. À cette étape-ci, le prototype fonctionnel est suffisamment stable pour effectuer des tests d'utilisabilité en vue de comparer la performance des utilisateurs avec le prototype et les interfaces actuelles.



## **CHAPITRE 6 CONCLUSION (ET RECOMMANDATIONS)**

Ce chapitre présente les résultats d'une étude comparative de la performance des utilisateurs dans la formulation de requêtes à une base de données relationnelle avec trois différentes interfaces, soit les deux interfaces actuelles de critères de recherche de l'application Croesus Conseiller et la nouvelle interface que nous avons développée dans ce projet.

La phase 2 du projet utilise la même méthodologie et les mêmes tâches que celles qui ont été utilisées au troisième cycle de la phase 1 de conception décrit dans le chapitre précédent. Cependant, ici les sujets sont représentatifs des utilisateurs visés.

Les critères de comparaison utilisés dans le test d'utilisabilité sont le temps d'exécution des tâches, le nombre d'erreurs de formulation et le nombre de demandes d'assistance des utilisateurs. Les sujets qui participent aux tests sont des utilisateurs réguliers de l'application Croesus Conseiller.

### **6.1 Résultats des tests d'utilisabilité de la phase 2**

La section suivante présente les résultats d'analyse de la variance de la phase 2. L'analyse de la variance permet de savoir s'il y a une différence significative sur le plan statistique entre les niveaux de performance humaine avec les trois interfaces que nous avons comparés.

#### **6.1.1 Synthèse des résultats statistiques pour la phase 2**

Dans le but d'alléger les tableaux, les trois interfaces ont été identifiées par une lettre :

A – l'interface de critères de recherche simplifiée

B – l'interface de critères de recherche avancée

C – l'interface de formulation de requêtes du prototype de formulaire flexible de formulation de requêtes par diagramme

Pour chacune des tâches à effectuer, le sujet devait formuler une requête à l'aide des différentes interfaces testées à partir d'une requête décrite en français. Le tableau 6.1 présente la description des tâches qui ont été soumises aux sujets lors de la phase 2 des tests d'utilisabilité.

Tableau 6.1 Détails des tâches qui ont été soumises aux sujets lors de la phase 2 des tests.

Tâche 1	<p>Liste des clients réels</p> <p>dont le code de cp fait partie de la liste suivante :</p> <ul style="list-style-type: none"> <li>- AC42</li> <li>- AC44</li> <li>- AC47</li> </ul> <p>dont la marge est supérieure ou égale à 0,00</p> <p>dont la date de dernier contact est entre le 1er janvier 2017 et le 1er mai 2017 (inclusivement)</p>
Tâche 2	<p>Liste des clients réels</p> <p>dont la valeur totale est supérieure à zéro</p> <p>ayant effectué au moins une transaction</p> <ul style="list-style-type: none"> <li>- de type cotisation</li> <li>- où la date de transaction est durant l'année 2017</li> </ul> <p>comprenant au moins 1 compte (lié à) dont le type est parmi la liste suivante :</p> <ul style="list-style-type: none"> <li>- Régime d'épargne retraite</li> <li>- Régime d'épargne retraite - conjoint</li> </ul> <p>dont la devise du client est CAD</p>
Tâche 3	<p>Liste des clients réels</p> <p>dont le chargé de comptes est Albert Einstein</p> <p>ayant au moins 1 compte</p> <ul style="list-style-type: none"> <li>- dont le type est Régime d'épargne retraite</li> <li>- sur lequel il y a une transaction <ul style="list-style-type: none"> <li>o de type achat</li> </ul> </li> </ul> <p>dont la date de transaction est en 2017</p>

### 6.1.2 Détail des calculs statistiques pour la phase 2

Le tableau 6.2 présente la moyenne et l'écart type de la durée des tests d'utilisabilité selon la tâche et l'interface. La durée est en secondes.

Tableau 6.2 Moyenne et écart type de la durée de chaque tâche pour chaque interface. Durée en secondes (moyenne sur 12 sujets).

		Tâche 1	Tâche 2	Tâche 3	<b>Pour les trois tâches réunies</b>
Interface A	Moyenne	145,5	272,17	153,08	<b>190,25</b>
	Écart type	39,19	132,3	109,52	<b>114,95</b>
Interface B	Moyenne	192,33	223,67	132	<b>182,67</b>
	Écart type	112,96	106,09	40,03	<b>98,07</b>
Interface C	Moyenne	93,83	160,17	95,25	<b>116,42</b>
	Écart type	40,21	76,43	36,16	<b>61,15</b>

Le tableau 6.3 présente la moyenne et l'écart type du nombre d'erreurs pour chaque tâche et chaque interface. Par exemple, si la personne a ouvert une parenthèse sans la refermer, c'est une erreur qui est comptabilisée comme 1 point. Un autre exemple d'erreur se produit lorsque la personne sélectionne le mauvais opérateur dans une condition.

Tableau 6.3 Moyenne et écart type du nombre d'erreurs pour chaque tâche et chaque interface.

		Tâche 1	Tâche 2	Tâche 3	<b>Pour les trois tâches réunies</b>
Interface A	Moyenne	0,25	0,33	0,75	<b>0,56</b>
	Écart type	0,62	0,67	0,79	<b>0,73</b>
Interface B	Moyenne	0,17	1,75	2,08	<b>1,36</b>
	Écart type	0,45	1,96	2,43	<b>1,94</b>
Interface C	Moyenne	0,17	0,5	0,25	<b>0,42</b>
	Écart type	0,39	1,07	0,67	<b>0,77</b>

Le tableau 6.4 présente la moyenne et l'écart type du nombre de demandes d'assistance à la tâche des sujets pour chaque tâche et chaque interface.

Tableau 6.4 Moyenne et écart type du nombre de demandes d'assistance à la tâche des sujets.

		Tâche 1	Tâche 2	Tâche 3	<b>Pour les trois tâches réunies</b>
Interface A	Moyenne	0,08	1	0,17	<b>0,53</b>
	Écart type	0,39	1,60	0,39	<b>1,08</b>
Interface B	Moyenne	0,42	0,33	0,17	<b>0,53</b>
	Écart type	1,41	0,78	10,62	<b>1,03</b>
Interface C	Moyenne	0,25	0,42	0,17	<b>0,31</b>
	Écart type	0,45	1,17	0,58	<b>0,79</b>

Cette section présente les résultats des analyses de variance pour le dernier cycle des tests d'utilisabilité.

### 6.1.3 Analyse de variance – temps de traitement

Le tableau 6.5 présente les résultats de l'analyse de variance pour le temps de traitement de la tâche 1 effectuée dans les trois interfaces.

Tableau 6.5 Analyse de variance : un facteur pour le temps de traitement de la tâche 1.

#### RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 1	12	1746	145,5	1536,09
interface B - tâche 1	12	2308	192,33	12759,52
interface C - tâche 1	12	1126	93,83	1616,88

#### ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour F</i>
Entre Groupes	58260,22	2	29130,11	5,49	<b>0,008</b>	3,28
À l'intérieur des groupes	175037,33	33	5304,16			
Total	233297,56	35				

**La différence est donc significative ( $F_2 = 5,49, p < 0,05$ ).**

Étant donné que le résultat de l'analyse de la variance révèle que la différence est significative, on poursuit l'analyse des données pour identifier l'interface qui se démarque par rapport aux autres. Le tableau 6.6 présente le résultat de l'analyse Test de Student (ou test-t).

Tableau 6.6 Test de Student (ou test-t) sur des échantillons appariés.

	Interface A	Interface B
Interface B	0.1248	NA
Interface C	0.0916	<b>0.0022</b>

**Il y a une différence significative entre l'interface B et l'interface C pour la tâche 1 :** les utilisateurs sont plus rapides avec l'interface C qu'avec l'interface B. De plus, bien que la différence ne soit pas significative sur le plan statistique, les utilisateurs sont plus rapides avec l'interface C (prototype) qu'avec l'interface A (simplifiée).

Le tableau 6.7 présente les résultats de l'analyse de variance pour le temps de traitement de la tâche 2 effectuée dans les trois interfaces.

Tableau 6.7 Analyse de variance : un facteur pour le temps de traitement de la tâche 2.

RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 2	12	3266	272,17	17502,33
interface B - tâche 2	12	2684	223,67	11254,42
interface C - tâche 2	12	1922	160,17	5840,88

ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour F</i>
						3,28
Entre Groupes	75714	2	37857	3,28	<b>0,05</b>	3,28
À l'intérieur des groupes	380574	33	11532,55			
Total	456288	35				

**La différence n'est donc pas significative ( $F_2 = 3,28, p > 0,05$ ). P est égal à 0.05.**

Le tableau 6.8 présente les résultats de l'analyse de variance pour le temps de traitement de la tâche 3 effectuée dans les trois interfaces.



Tableau 6.8 Analyse de variance : un facteur pour le temps de traitement de la tâche 3.

## RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 3	12	1837	153,08	11995,36
interface B - tâche 3	12	1584	132	1851,64
interface C - tâche 3	12	1143	95,25	1307,48

## ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour</i>
						<i>F</i>
Entre Groupes	20559,05556	2	10279,53	2,03	<b>0,15</b>	3,28
À l'intérieur des groupes	166699,1667	33	5051,49			
Total	187258,2222	35				

**La différence n'est donc pas significative ( $F_2 = 2,03, p > 0,05$ ).**

## Analyse de la variance - nombre d'erreurs

Le tableau 6.9 présente les résultats de l'analyse de variance pour le nombre d'erreurs de la tâche 1 effectuée avec les trois interfaces.

Tableau 6.9 Analyse de variance : un facteur pour le nombre d'erreurs de la tâche 1.

## RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 1	12	3	0,25	0,39
interface B - tâche 1	12	3	0,25	0,20
interface C - tâche 1	12	2	0,17	0,15

## ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour</i>
						<i>F</i>
Entre Groupes	0,06	2	0,028	0,11	<b>0,89</b>	3,28
À l'intérieur des groupes	8,17	33	0,25			
Total	8,22	35				

**La différence n'est donc pas significative ( $F_2 = 0.11, p > 0,05$ ).**

Le tableau 6.10 présente les résultats de l'analyse de variance pour le nombre d'erreurs de la tâche 2 effectuée dans les trois interfaces.

Tableau 6.10 Analyse de variance : un facteur pour le nombre d'erreurs de la tâche 2.

RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 2	12	6	0,5	0,45
interface B - tâche 2	12	21	1,75	3,84
interface C - tâche 2	12	8	0,67	1,15

ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique</i>
						<i>pour F</i>
Entre Groupes	11,06	2	5,53	3,04	<b>0,06</b>	3,28
À l'intérieur des groupes	59,92	33	1,82			
Total	70,97	35				

**La différence n'est donc pas significative ( $F_2 = 3.04, p > 0,05$ ).**

Le tableau 6.11 présente les résultats de l'analyse de variance pour le nombre d'erreurs de la tâche 3 effectuée dans les trois interfaces.

Tableau 6.11 Analyse de variance : un facteur pour le nombre d'erreurs de la tâche 3.

RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 3	12	11	0,92	0,63
interface B - tâche 3	12	25	2,08	5,90
interface C - tâche 3	12	5	0,42	0,45

ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour F</i>
Entre Groupes	17,56	2	8,78	3,77	<b>0,03</b>	3,28
À l'intérieur des groupes	76,75	33	2,33			
Total	94,31	35				

**La différence est donc significative ( $F_2 = 3.77, p < 0,05$ ).**

Étant donné que le résultat de l'analyse de la variance révèle qu'il existe une différence significative entre les moyennes, on poursuit l'analyse des données pour identifier l'interface qui se démarque par rapport aux autres. Le tableau 6.12 présente le résultat de l'analyse « t tests with pooled sd ».

Tableau 6.12 Test de Student (ou test-t) sur des échantillons appariés.

	A	B
B	0.0698	NA
C	0.4277	<b>0.0115</b>

**Il y a donc une différence significative entre le résultat de l'interface B et de l'interface C pour la tâche 3. Les utilisateurs font en moyenne plus d'erreurs avec l'interface B qu'avec l'interface C.** Par contre la différence n'est pas significative entre les résultats obtenus avec l'interface A et l'interface C. On peut en conclure que l'interface C (prototype) offre une performance plus efficace pour la tâche 3 par rapport à l'interface de critères de recherche avancée.

### 6.1.4 Analyse de la variance - nombre de demandes d'assistance

Le tableau 6.13 présente les résultats de l'analyse de variance pour le nombre de demandes d'assistance à la tâche des sujets dans la tâche 1 effectuée avec les trois interfaces.

Tableau 6.13 Analyse de variance : un facteur pour le nombre de demandes d'assistance à la tâche des utilisateurs dans la tâche 1.

#### RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 1	12	2	0,17	0,15
interface B - tâche 1	12	12	1	2
interface C - tâche 1	12	3	0,25	0,20

#### ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour F</i>
Entre Groupes	5,06	2	2,53	3,22	<b>0,053</b>	3,28
À l'intérieur des groupes	25,92	33	0,79			
Total	30,97	35				

**La différence n'est donc pas significative ( $F_2 = 3.2, p > 0,05$ ).**

Le tableau 6.14 présente les résultats de l'analyse de variance pour le nombre de demandes d'assistance à la tâche des sujets dans la tâche 2 effectuée avec les trois interfaces.

Tableau 6.14 Analyse de variance : un facteur pour le nombre de demandes d'assistance à la tâche des utilisateurs dans la tâche 2.

#### RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre d'échantillo</i>			
	<i>ns</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 2	12	15	1,25	2,57
interface B - tâche 2	12	4	0,33	0,61
interface C - tâche 2	12	6	0,5	1,36

#### ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour F</i>
Entre Groupes	5,72	2	2,86	1,89	<b>0,17</b>	3,28
À l'intérieur des groupes	49,96	33	1,51			
Total	55,64	35				

**La différence n'est donc pas significative ( $F_2 = 1.89, p > 0,05$ ).**

Le tableau 6.15 présente les résultats de l'analyse de variance pour le nombre de demandes d'assistance à la tâche des sujets dans la tâche 3 effectuée avec les trois interfaces.

Tableau 6.15 Analyse de variance : un facteur pour le nombre de demandes d'assistance à la tâche des utilisateurs dans la tâche 3.

#### RAPPORT DÉTAILLÉ

<i>Groupes</i>	<i>Nombre</i>			
	<i>d'échantillons</i>	<i>Somme</i>	<i>Moyenne</i>	<i>Variance</i>
interface A - tâche 3	12	2	0,17	0,15
interface B - tâche 3	12	3	0,25	0,39
interface C - tâche 3	12	2	0,17	0,33

#### ANALYSE DE VARIANCE

<i>Source des variations</i>	<i>Somme des carrés</i>	<i>Degré de liberté</i>	<i>Moyenne des carrés</i>	<i>F</i>	<i>Probabilité</i>	<i>Valeur critique pour</i>
						<i>F</i>
Entre Groupes	0,06	2	0,03	0,10	<b>0,91</b>	3,28
À l'intérieur des groupes	9,58	33	0,29			
Total	9,64	35				

**La différence n'est donc pas significative ( $F_2 = 0.09, p > 0,05$ ).**



### 6.1.5 Résumé des résultats

Le tableau 6.16 présente un résumé des résultats de l'analyse de la variance.

Tableau 6.16 Résumé des résultats de l'analyse de variance.

	Tâche	Valeur p du calcul de la variance. (Valeur significative est : $p < 0,05$ )	Commentaires
Temps de traitement	1	<b>0,008</b>	<b>Il y a une différence significative entre l'interface B et l'interface C pour la tâche 1</b> : les utilisateurs sont plus rapides avec l'interface C qu'avec l'interface B.  De plus, bien que la différence ne soit pas significative sur le plan statistique, les utilisateurs sont plus rapides avec l'interface C (prototype) qu'avec l'interface A (simplifiée).
	2	0,05	<b>Différence non significative.</b>  Malgré le fait que la différence ne soit pas significative sur le plan statistique, l'interface du prototype présente de meilleurs résultats que les deux autres.
	3	0,15	<b>Différence non significative</b>  Malgré le fait que la différence ne soit pas significative sur le plan statistique, l'interface du prototype présente de meilleurs résultats que les deux autres.

Tableau 6.16 : Résumé des résultats de l'analyse de variance. (suite et fin)

Nombre d'erreurs	1	0,89	<b>Différence non significative</b>
	2	0,06	<b>Différence non significative</b> L'interface de critères de recherche simplifiée (A) offre une performance plus intéressante que les autres.
	3	<b>0,03</b>	<b>Il y a une différence significative entre l'interface B et l'interface C pour la tâche 3.</b> On peut en conclure que l'interface C (prototype) offre une performance plus efficace pour la tâche 3 par rapport à l'interface de critères de recherche avancée.
Nombre de demandes d'assistance	1	0,05	<b>Différence non significative</b> L'interface de critères de recherche avancée (B) offre une moins bonne performance que les deux autres interfaces.
	2	0,17	<b>Différence non significative</b> L'interface de critères de recherche avancée (B) offre une performance plus intéressante que les autres.
	3	0,91	<b>Différence non significative</b>

## 6.2 Ordre de préférence et commentaires recueillis lors des entrevues

Suite aux tests, les sujets étaient invités à classer les différentes interfaces testées par ordre de préférence : **10 sujets** sur 12 ont placé l'interface du prototype **en première** et deux sujets l'ont placée en deuxième.

Par la suite, une entrevue semi-structurée a permis de recueillir les commentaires des sujets à propos des différentes interfaces. La section qui suit présente un résumé des commentaires des utilisateurs ainsi qu'une synthèse des observations de la chercheuse à propos des comportements des sujets avec les interfaces durant l'exécution des tests.

Cette section présente les observations et les commentaires recueillis lors des tests d'utilisabilité pour l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme.

### 6.2.1 Opérateur logique ET/OU

Plusieurs sujets ( $N = 4$ ) ont mentionné que le mode d'affichage des opérateurs logiques ET et OU (voir figure 6.1) permet de les distinguer facilement, car ils sont mis en évidence par une boîte combo. De plus, il est facile pour l'utilisateur d'en ajouter et de les modifier.

La figure 6.1 présente une capture d'écran de la nouvelle interface.

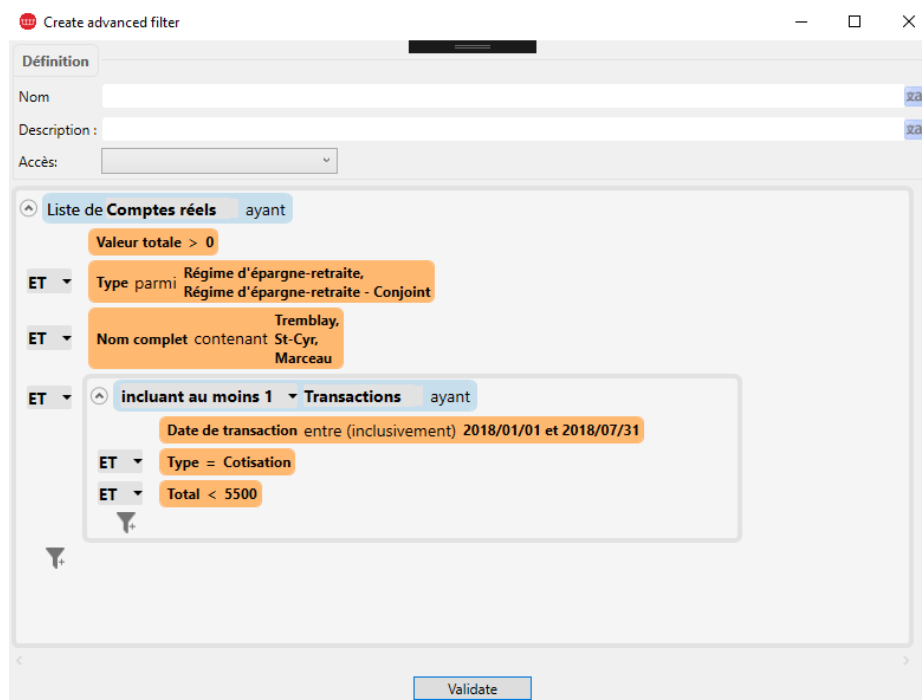


Figure 6.1 Interface de formulation de critères de recherche provenant du prototype de formulaire flexible de formulation de requêtes par diagramme.

## 6.2.2 Facilité d'utilisation

De façon générale, les sujets mentionnent que l'utilisation de l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme est plus intuitive que celle des deux autres. De plus, la formulation d'un critère de recherche nécessite moins d'étapes. Par conséquent, la formulation du critère de recherche est plus rapide.

Un sujet a mentionné qu'il y avait beaucoup moins de possibilités de faire des erreurs dans cette interface que dans les autres puisque les groupements de conditions et de jointures se font de façon automatique par l'interface. De plus, l'utilisation est plus intuitive, et ce, malgré le fait que le sujet utilisait l'interface pour la première fois.

L'apprentissage du fonctionnement de l'interface se faisait rapidement pour les différents sujets.

## 6.2.3 Capacité équivalente aux interfaces actuelles

Les cinq sujets les plus expérimentés (sur 12) ont mentionné que l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme avait une plus grande capacité pour

la formulation de requêtes grâce à une sémantique complexe comportant entre autres plusieurs niveaux de jointures. Ils considéraient que la nouvelle interface avait les mêmes capacités sémantiques que les deux autres interfaces réunies.

La chercheuse a remarqué que les sujets (N = 6) qui sont moins habitués avec les interfaces actuelles éprouvaient de la difficulté à faire des recherches dans la hiérarchie de menus pour les champs des conditions simples. En effet, la structure hiérarchique des menus est la même dans la nouvelle interface que dans l'interface simplifiée. Cet aspect pourrait être amélioré.

Par ailleurs, le fait d'afficher l'élément impliqué dans la jointure dans l'entête de l'encadré aide les sujets à connaître le contexte du menu, ce qui est un avantage par rapport aux autres interfaces. La chercheuse a remarqué que cet aspect avait aidé les sujets à accomplir leur tâche plus facilement pour trois des sujets rencontrés. Les sujets les plus expérimentés (N = 5) étaient déjà à l'aise avec le concept. Cela ne faisait donc pas de différence majeure pour eux.

#### **6.2.4 Choix de couleurs**

Un sujet a mentionné que les couleurs utilisées étaient trop présentes et trop fortes et que cela était agressant.

#### **6.2.5 Modèle logique d'entités**

Le graphique du modèle logique d'entités présenté lors des explications préalables à l'exécution des tests d'utilisabilité (figure 6.2) n'a pas semblé aider les utilisateurs à créer et manipuler les jointures plus facilement. Toutefois, cette manipulation semble plus facile dans la nouvelle interface du prototype de formulaire flexible de formulation de requêtes par diagramme que dans les autres interfaces de critère de recherche pour 10 des sujets rencontrés.

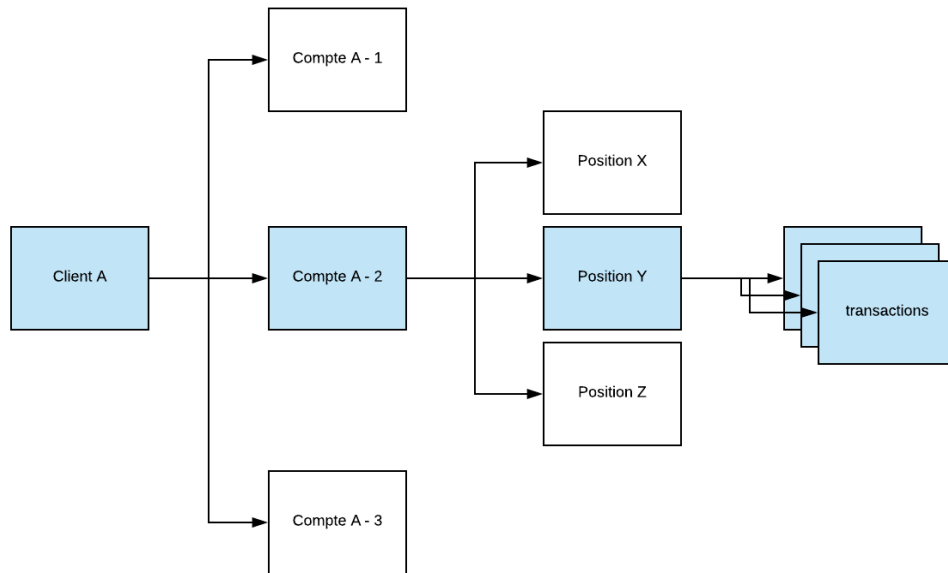


Figure 6.2 Graphique présentant le modèle logique d'entités disponibles pour la création de requêtes.

## **CHAPITRE 7 DISCUSSION ET CONCLUSION**

Ce dernier chapitre présente une discussion des résultats et la conclusion de notre recherche. La discussion fait ressortir l'essentiel des nouvelles connaissances développées dans ce projet de recherche et montre les implications des résultats obtenus pour la conception d'une interface de formulation de requêtes. Dans la conclusion, nous présentons les limites de notre recherche et nous suggérons des pistes pour de prochaines études.

### **7.1 Discussion**

L'objectif du projet de recherche est de développer une solution d'interface utilisateur permettant aux utilisateurs d'un logiciel de gestion de portefeuilles de créer plus facilement des requêtes à une base de données qu'avec les interfaces actuelles.

#### **7.1.1 Hypothèse H1 sur le temps requis de formulation de requêtes**

La recherche révèle que l'hypothèse H1 est partiellement confirmée pour la tâche 1 et ne l'est pas pour les tâches 2 et 3. Le temps de formulation de requêtes dans la tâche 1 est significativement plus court avec la nouvelle interface (93,8 sec.) comparativement à l'interface de critères de recherche avancée (192,3 sec.). Pour les trois tâches, la nouvelle interface de formulaire flexible de formulation de requêtes par diagramme obtient les meilleurs résultats mais peu de résultats sont significatifs sur le plan statistique. Considérant que la nouvelle interface offre une capacité sémantique plus poussée, grâce à la capacité d'ajouter des jointures à plusieurs niveaux, l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme est le meilleur choix.

Bien qu'en deçà de nos espérances, ces résultats vont dans la bonne direction et sont encourageants dans la mesure où les sujets n'avaient aucune expérience avec la nouvelle interface alors qu'ils avaient plusieurs années avec les deux interfaces comparaison.

#### **7.1.2 Hypothèse H2 sur le nombre d'erreurs lors de formulation de requêtes**

La recherche révèle que l'hypothèse H2 est partiellement confirmée lorsqu'on compare les résultats d'erreurs de la nouvelle interface à ceux de l'interface de critères de recherche avancée. En effet le nombre d'erreurs produites avec la nouvelle interface est plus faible que le nombre d'erreurs

produites avec l'interface de critères de recherche avancée pour la tâche 3. Les résultats ne sont pas significatifs pour les deux autres tâches. Toutefois, les résultats obtenus avec la nouvelle interface sont meilleurs que ceux obtenus avec les deux autres interfaces.

Il n'y a pas de différence significative de nombre d'erreurs entre la nouvelle interface et l'interface de critères de recherche simplifiée. Ce résultat peut s'expliquer par le fait que la chercheuse a donné des indications avant l'exécution des tests et par le fait que les sujets avaient la possibilité de demander de l'assistance pour la formulation de la requête. La création de groupes de conditions et de jointures par l'interface simplifiée nécessite l'ajout de parenthèses. L'endroit où ajouter les parenthèses et la façon de les ajouter ne sont pas évidents. Il faut l'enseigner à l'utilisateur. Dans un contexte d'utilisation normal, les utilisateurs n'ont pas facilement recours à l'aide pour la formulation de la requête. Cette notion est facilement omise et, lorsqu'elle n'est pas omise, elle est difficile à accomplir.

### **7.1.3 Hypothèse H3 le nombre de demandes d'assistance pour la formulation de requêtes**

La recherche révèle que l'hypothèse H3 n'est pas confirmée. Il n'y a pas de différence significative d'un point de vue statistique entre les trois interfaces quant au nombre de demandes d'assistances des sujets pour la formulation de requêtes.

Ce résultat est décevant. Les sujets avaient tendance à se décourager lorsqu'ils cherchaient les champs dans la liste pour formuler les différents filtres. Cet aspect de l'interface pourrait être amélioré.

### **7.1.4 Autres observations**

Lors du dernier cycle de test d'utilisabilité, nous avons remarqué que les utilisateurs perdaient beaucoup de temps à chercher les champs requis pour la formulation de la requête à travers l'arborescence de menus qui donnent accès aux champs. Pour aider les utilisateurs à formuler les requêtes plus rapidement, l'interface devrait offrir à l'utilisateur de saisir le nom du champ désiré avec une auto complétion qui permet de filtrer la liste des champs offerts. L'interface devrait supporter plusieurs synonymes pour chacun des champs offerts en fonction des termes utilisés dans le marché pour identifier chaque champ. Cette solution pourrait se baser sur les algorithmes de



recherche approximative décrits par G. Navarro [32]. Dans le même ordre d'idée, lorsque le champ utilisé dans la condition est de type « valeur provenant d'une liste de dictionnaire », l'interface devrait permettre à l'utilisateur de saisir la valeur souhaitée pour permettre de filtrer les choix offerts de façon automatique. Encore une fois, cette recherche pourrait se baser sur l'approche de recherche approximative.

Par ailleurs, la liste des opérateurs offerts pour les champs de type date est très longue. L'interface pourrait présenter les opérateurs dans une structure séparée par des séparateurs pour aider l'utilisateur à s'y retrouver dans la liste. C'est un principe qui permet de réduire l'impression de densité de l'information grâce à la structure et à la disposition de l'information.

L'une des plus grandes difficultés rencontrées par les sujets réside dans l'utilisation des jointures. En effet, pour bien utiliser les jointures, il faut avoir un modèle mental clair du modèle d'entités logiques et de l'effet de l'ajout de conditions sur des entités autres que l'entité retournée dans le résultat. Autrement dit, il faut avoir une connaissance de la théorie des ensembles provenant du monde mathématique. Nos utilisateurs ne maîtrisent pas bien ces notions.

## **7.2 Conclusion**

### **7.2.1 Limites de la recherche**

Notre recherche ne permet pas d'évaluer la facilité avec laquelle les utilisateurs peuvent exécuter, consulter et analyser le résultat de la requête pour ensuite retourner la modifier. La revue de la littérature indique que les utilisateurs procèdent de façon itérative lors de la formulation d'une requête. Il est donc important de permettre à l'utilisateur de naviguer facilement entre le résultat de la requête et l'interface de formulation et modification de la requête. Cet aspect n'a pas été abordé par la présente recherche. Le prototype qui a été développé ne permet pas de faire l'exécution de la requête.

De plus, une des étapes importantes de la formulation d'une requête à une base de données consiste à valider le résultat de la requête. Actuellement, les interfaces de critères de recherche avancée et simplifiée ainsi que l'interface du prototype de formulaire flexible de formulation de requêtes par diagramme n'offrent aucun outil pour aider l'utilisateur à mieux comprendre les résultats de la requête. Lorsqu'un utilisateur s'attend à avoir certains éléments précis dans le résultat de la requête

et que le résultat ne correspond pas à ses attentes, il est laissé à lui-même pour décortiquer les données obtenues et comprendre l'erreur qu'il a faite lors de la formulation de la requête.

Par ailleurs, lors des rencontres avec les sujets, une problématique a été mentionnée à plusieurs reprises : la fonctionnalité des critères de recherche permet aux utilisateurs de partager les critères de recherche préalablement sauvegardés. Il est aussi possible pour les administrateurs de système de créer des critères de recherche qui sont visibles par tous les utilisateurs d'une firme (institution financière). L'objectif étant de permettre aux utilisateurs de réutiliser des critères de recherche en modifiant les valeurs dans les conditions. Cette fonctionnalité n'est pas utilisée à son plein potentiel en raison de la grande quantité de critères de recherche partagés. Un outil permettant de rechercher et de catégoriser les critères de recherche permettrait d'améliorer grandement la fonctionnalité. Cette amélioration permettrait d'offrir une approche collaborative pour la formulation de requêtes tel que proposé par Nikam et par Chu, Baid, Chai, Doan et Naughton. Cela pourrait contribuer à réduire la nécessité de faire de la formation au sujet de cette fonctionnalité.

Enfin, dans le cadre de ce projet de recherche, l'approche basée sur la reconnaissance du langage naturel ne faisait pas partie du mandat de notre recherche en raison de contraintes de temps et du manque de connaissance de la chercheuse sur le sujet. Toutefois, cette approche offre un potentiel très intéressant et pourrait faire l'objet d'une recherche future.

### **7.2.2 Pistes de recherche**

Pour le futur, il serait intéressant d'étudier une approche permettant à l'utilisateur de taper au clavier les champs et les valeurs désirées avec une auto complétion offerte par le système. De plus, le système pourrait offrir un dictionnaire permettant de reconnaître plusieurs noms différents pour faire référence à un même concept similaire à la recherche approximative. De cette façon, l'utilisateur pourrait réduire le temps requis pour rechercher les champs de la requête dans les listes déroulantes.

Une autre avenue possible pour une recherche future serait de s'intéresser à l'affichage et à l'analyse des résultats. En effet, les données de la base de données sont formées de plusieurs entités qui sont en relations les unes avec les autres. En ce moment, l'affichage des résultats ne permet pas de voir les relations facilement entre les différents éléments. Avec cette approche, on pourrait

permettre à l'utilisateur de décortiquer la requête pour évaluer les jointures de façon indépendante du reste de la requête.

Afin d'exploiter davantage la capacité de partage des requêtes entre les utilisateurs, une étude pourrait être menée pour appliquer les principes proposés par Nikam et par Chu, Baid, Chai, Doan et Naughton qui permettent une approche hybride entre la recherche par mots clés et l'interface par formulaire et qui permet une collaboration entre les utilisateurs.

### **7.2.3 Contribution de notre recherche**

Dans cette recherche, nous avons créé, développé et testé une nouvelle approche de formulation de requêtes et l'avons implantée dans un prototype d'interface. Cette approche se nomme : formulaire flexible de formulation de requêtes par diagramme. C'est une solution hybride entre l'approche par formulaire dynamique et de l'approche de requête par diagramme (« query-by-diagram »). Elle utilise aussi un formulaire pour la formulation des conditions qui s'inspire de l'approche de requêtes par filtres.

L'approche proposée pourrait être utilisée avec n'importe quel format de données, mais cela nécessite une configuration préalable du modèle logique de données et de l'association avec le modèle physique. Le fait d'offrir un modèle logique préétabli permet de garder un certain contrôle sur la requête produite et ainsi offrir un temps de réponse raisonnable. De plus, l'utilisateur risque moins de se sentir perdu dans une trop grande quantité d'information ou une arborescence d'entités trop étendue. Par ailleurs, certains champs plus complexes basés sur des calculs peuvent être offerts.

L'approche étudiée pour l'interface est indépendante de la source d'information. Elle pourrait donc être utilisée autant sur une base de données avec le langage SQL que sur des API web avec le langage SPARQL. Toutefois, les interfaces basées sur des langages SPARQL nécessitent une grande souplesse pour se brancher sur une grande variété de sources de données. Cela implique une interface qui nécessite peu ou pas de configuration. Notre nouvelle interface comporte une faiblesse sur cet aspect.

La nouveauté du prototype de l'interface formulaire flexible de formulation de requêtes par diagramme réside dans la capacité offerte à l'utilisateur de structurer la requête selon son besoin, avec des groupes de conditions et des jointures. L'interface permet une sémantique très riche.

Enfin, tel que c'est le cas avec les interfaces de formulation de critères de recherche actuellement disponibles dans l'application Croesus Conseiller, le partage des requêtes entre les utilisateurs est possible. Cela permet une collaboration entre les utilisateurs pour la formulation de requêtes. Donc, lorsqu'un utilisateur réutilise une requête préalablement sauvegardée, c'est l'équivalent d'utiliser une interface de type formulaire rigide. Il n'a qu'à modifier les valeurs dans le formulaire pour adapter la requête à son besoin sans avoir à formuler une requête à partir de rien. Cela lui permet d'apprendre le langage de formulation proposé par l'interface à l'aide d'exemples.

#### **7.2.4 Suite du projet**

L'entreprise Croesus Finansoft est intéressée à poursuivre le développement de la nouvelle approche d'interface pour en faire une solution complète et l'offrir à ses utilisateurs. La solution implémentée sera inspirée de l'interface proposée par la recherche, toutefois, elle sera développée dans un langage WEB. La solution finale doit tenir compte de contraintes d'accessibilité. Elle doit aussi pouvoir être utilisée sur des appareils mobiles en plus d'un ordinateur avec souris. Cela pourrait nécessiter des modifications à la solution proposée.

## RÉFÉRENCES

- [1] (2018). *Online Browsing Platform (OBP)*. Available: <https://www.iso.org/obp/ui - iso:std:iso:9241:-11:ed-2:v1:fr>
- [2] S. El-Mahgary and E. Soisalon-Soininen, "A form-based query interface for complex queries," *Journal of Visual Languages & Computing*, vol. 29, pp. 15-53, 2015.
- [3] G. Gardarin, *Bases de données*, EYROLLES ed. 2003, p. 826.
- [4] M. Jarrar, "MashQL : A Query-by-Diagram Topping SPARQL," Napa Valley, California, USA, 2008, pp. 89-96: Citeseer.
- [5] M. Jarrar and M. D. Dikaiakos, "A data mashup language for the data web," in *The WWW2009 Workshop on Linked Data on the Web*, Madrid, Spain, 2009, vol. LDOW.
- [6] C. Yu and H. Jagadish, "Querying complex structured databases," in *Proceedings of the 33rd international conference on Very large data bases*, 2007, pp. 1010-1021: VLDB Endowment.
- [7] D. P. Vronay, C. Marcjan, A. Turski, and R. Kott, "Database query user interface to assist in efficient and accurate query construction," ed: Google Patents, 2013.
- [8] E. Keramopoulos, P. Pouyioutas, and T. Ptohos, "The GOQL Language and its Formal Specifications," *IJCSA*, vol. 5, pp. 23-51, 2008.
- [9] J. Mustafa and M. D. Dikaiakos, "MashQL: A Query-by-Diagram Topping SPARQL Towards Semantic Data Mashups," presented at the The 2nd international workshop on Ontologies and information systems for the semantic web, Napa Valley, California, USA, 2008.
- [10] E. Bakke and D. R. Karger, "Expressive query construction through direct manipulation of nested relational results," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 1377-1392: ACM.
- [11] S. Agrawal, S. Chaudhuri, and G. Das, "DBXplorer: A System for Keyword-Based Search over Relational Databases," presented at the Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, January 2002,
- [12] B. Balla and Z. Chen, "Expanding database keyword search for database exploration," *Procedia Computer Science*, vol. 17, pp. 198-205, 2013.
- [13] J. Fan, G. Li, and L. Zhou, "Interactive SQL query suggestion: Making databases user-friendly," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, 2011, pp. 351-362: IEEE.
- [14] H. Jagadish *et al.*, "Making database systems usable," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 13-24: ACM.
- [15] P. Nikam, "A review on dynamic query forms for database queries," *IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 8079-8081, 2015.

- [16] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton, "Combining keyword search and forms for ad hoc querying of databases," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 349-360: ACM.
- [17] M. Kahng, S. B. Navathe, J. T. Stasko, and D. H. P. Chau, "Interactive browsing and navigation in relational databases," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1017-1028, 2016.
- [18] A. Nandi and H. Jagadish, "Guided interaction: Rethinking the query-result paradigm," *Proceedings of the VLDB Endowment*, vol. 4, no. 12, pp. 1466-1469, 2011.
- [19] F. Li and H. Jagadish, "Understanding natural language queries over relational databases," *ACM SIGMOD Record*, vol. 45, no. 1, pp. 6-13, 2016.
- [20] R. S. Khule and R. R. Keole, "The Study of Basics for a Query Formulation Language—MashQL," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 1, pp. 134-139, 2015.
- [21] M. T. Priyanka and M. M. Waghmare, "Study of MashQL Querying Language for the Data Web," *International Journal of Engineering Research and General Science*, vol. 2, no. 6, 2014.
- [22] E. Kaufmann and A. Bernstein, "How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?," Berlin, Heidelberg, 2007, pp. 281-294: Springer Berlin Heidelberg.
- [23] G. Li, J. Fan, H. Wu, J. Wang, and J. Feng, "DBease: Making Databases User-Friendly and Easily Accessible," presented at the CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 2011.
- [24] N. Jayaram, A. Khan, C. Li, X. Yan, and R. Elmasri, "Querying knowledge graphs by example entity tuples," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2797-2811, 2015.
- [25] M. Jarrar and M. D. Dikaiakos, "A query formulation language for the data web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 783-798, 2012.
- [26] A. Styperek, M. Ciesielczyk, A. Szwabe, and P. Misiorek, "Evaluation of SPARQL-compliant semantic search user interfaces," *Vietnam Journal of Computer Science*, journal article vol. 2, no. 3, pp. 191-199, August 01 2015.
- [27] E. Keramopoulos, D. Dervos, I. Deligiannis, G. Evangelidis, and P. Pouyioutas, "Critical Analysis of Designing a Graphical Query Language," in *4th annual International Conference on Education and New Learning Technologies*, Barcelona, Spain, 2012.
- [28] A. Nandi, L. Jiang, and M. Mandel, "Gestural query specification," *Proceedings of the VLDB Endowment*, vol. 7, no. 4, pp. 289-300, 2013.
- [29] A. Nandi, "Querying Without Keyboards," in *CIDR*, 2013.
- [30] B. Shneiderman, "Direct manipulation: A step beyond programming languages," *Computer*, no. 8, pp. 57-69, 1983.
- [31] M. Rouse. (2005). *fuzzy search*. Available: <https://whatis.techtarget.com/definition/fuzzy-search>

- [32] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31-88, 2001.
- [33] J. J. Inman, J. S. Dyer, and J. Jia, "A generalized utility model of disappointment and regret effects on post-choice valuation," *Marketing Science*, vol. 16, no. 2, pp. 97-111, 1997.
- [34] A. v. Hiel and I. Mervielde, "The need for closure and the spontaneous use of complex and simple cognitive structures," *The Journal of Social Psychology*, vol. 143, no. 5, pp. 559-568, 2003.
- [35] S. Idreos, O. Papaemmanouil, and S. Chaudhuri, "Overview of data exploration techniques," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 277-281: ACM.
- [36] B. K. Britton and A. Tesser, "Effects of prior knowledge on use of cognitive capacity in three complex cognitive tasks," *Journal of verbal learning and verbal behavior*, vol. 21, no. 4, pp. 421-436, 1982.
- [37] E. Brangier and J. Barcenilla, "Concevoir un produit facile à utiliser," *Paris: Editions d'organisation*, 2003.
- [38] N. Bevan, "International standards for HCI and usability," *International journal of human-computer studies*, vol. 55, no. 4, pp. 533-552, 2001.
- [39] M. S. Sanders and E. J. McCormick, *Human factors in engineering and design*. McGRAW-HILL book company, 1987.
- [40] J.-F. Nogier and J. Leclerc, *UX Design et ergonomie des interfaces-6e éd*. Dunod, 2016.
- [41] T. S. Tullis, "Optimizing the usability of computer-generated displays," in *Proceedings of the Second Conference of the British Computer Society, human computer interaction specialist group on People and computers: designing for usability*, 1986, pp. 604-613: Cambridge University Press.
- [42] S. Baldassi, N. Megna, and D. C. Burr, "Visual clutter causes high-magnitude errors," *PLoS biology*, vol. 4, no. 3, p. e56, 2006.
- [43] M. J. Bravo and H. Farid, "Recognizing and segmenting objects in clutter," *Vision Research*, vol. 44, no. 4, pp. 385-396, 2004.
- [44] J. M. Wolfe, "Guided Search 4.0 : Current progress with a model of visual search," vol. *Integrated models of cognitive systems* (Ed.), pp. 99-119, 2007.
- [45] M. P. Eckstein, "Visual search : A retrospective," *Journal of vision*, vol. 11, 14, december 2011.
- [46] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97-136, 1980.
- [47] P. Doyon-Poulin, "Effet de l'encombrement visuel de l'écran primaire de vol sur la performance du pilote, la charge de travail et le parcours visuel," *École polytechnique de Montréal*, 2014.
- [48] D. J. Mayhew, "Screen layout and design (chap. 14)," *Principles and guidelines in software user interface design*, pp. 458-506, 1992.

- [49] T. S. Tullis, "An evaluation of alphanumeric, graphic, and color information displays," *Human Factors*, vol. 23, no. 5, pp. 541-550, 1981.
- [50] M. Moshagen and M. T. Thielsch, "Facets of visual aesthetics," *International Journal of Human-Computer Studies*, vol. 68(10), pp. 689-709, 2010.
- [51] T. Lavie, O.-G. T., and M. J., "Aesthetics and usability of in-vehicle navigation displays," *International Journal of Human-Computer Studies*, vol. 69(1-2), pp. 80-99, 2011.
- [52] (2018). *La définition de Carré latin*. Available: <https://carnets2psycho.net/dico/sens-de-carre-latin.html>



## **ANNEXE A TESTS D'UTILISABILITÉ PHASE 1 – CYCLE 1 ET 2**

### **Première étape**

La première étape consiste en un formulaire démographique.

### **Deuxième étape**

Dans la deuxième étape, un exercice de familiarisation vous est présenté. Vous devez interpréter la requête qui est présentée en format graphique et identifier les enregistrements qui seraient, selon vous, affichés par la requête. Les résultats de cette étape ne seront pas considérés dans les résultats de l'étude. Elle permet de se familiariser avec le processus du test d'utilisabilité.

Cette étape sera enregistrée et chronométrée. Vous devez donc répondre le plus rapidement possible. Entre chaque question, une période de discussion est prévue.

### **Troisième étape**

Dans la troisième étape, 12 questions vous seront soumises. Une capture d'écran contenant un critère de recherche est présentée. Vous devez interpréter le critère de recherche et identifier les enregistrements qui seront retournés dans le résultat selon votre interprétation.

Cette étape sera enregistrée et chronométrée. Vous devez donc répondre le plus rapidement possible. Entre chaque question, une période de discussion est prévue.

### **Quatrième étape**

Dans la quatrième étape, vous devez placer les différentes interfaces en ordre de préférence. Vous devez donner un commentaire positif et un commentaire négatif pour chacune des 4 interfaces.

Cette étape sera enregistrée.

## Étape 1

Tableau A.1 Données démographiques

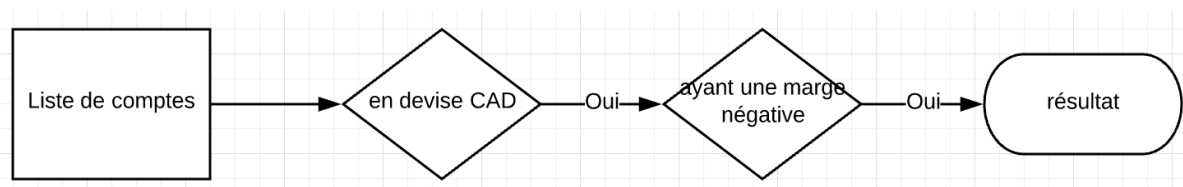
Participant	1-2-3-4				
Âge					
Genre					
Occupation					
Niveau de scolarité	Collège	Bacc	Maîtrise	Doctorat	
Nombre d'années, d'expérience avec Croesus Conseiller	Moins de 1 an	1 à 3 ans	3 à 5 ans	5 à 10 ans	plus de 10 ans
Nombre d'années d'expérience dans le domaine financier	Moins de 1 an	1 à 3 ans	3 à 5 ans	5 à 10 ans	plus de 10 ans
Niveau de familiarité avec la technologie en général	minimum	moyen	avancé	expert	
Fréquence d'utilisation de moteur de recherche	1 fois par année	tous les 3 mois	tous les mois	toutes les semaines	tous les jours

## Étape 2

## Exercice

Cette étape est un exercice de familiarisation. Les résultats de cette étape ne seront pas considérés dans la recherche.

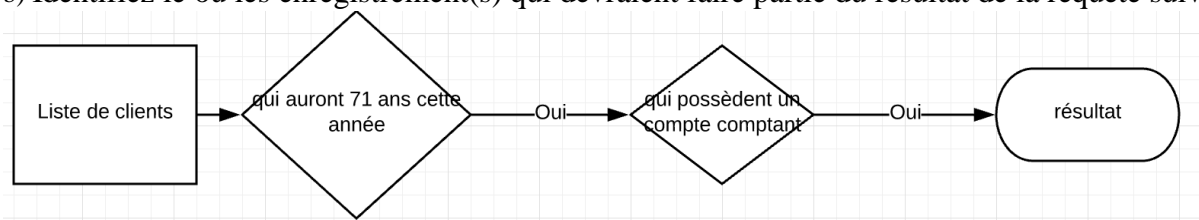
- a) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat de la requête suivante



Enregistrements :

Numéro de compte	Code de CP	Marge	Devise
À	AC44	0	CAD
B	AC45	-10000	CAD
C	ACC44	-1000	USD
D	AC43	0.01	USD
E	AC44	-0.01	USD
F	AC45	-1000	CAD
G	AC44	0.01	CAD
H	AC 44	1000	CAD
I	AC45	-8500	USD
J	AC44	1000	CAD
K	AC45	0	USD

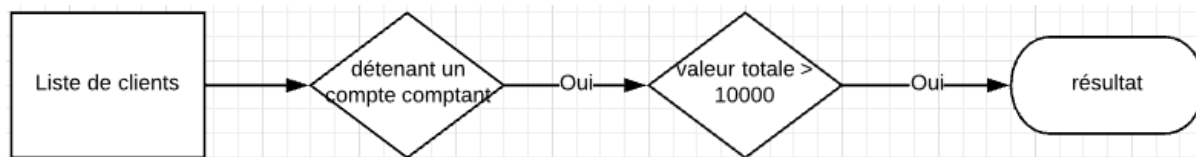
b) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat de la requête suivante



Enregistrements

No. de client	Date de naissance	Comptes	
A	31/12/1946	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
B	01/01/1947	Classe	Type
		Réel	Régime d'épargne retraite
C	31/12/1947	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
D	01/01/1975	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
		Réel	Comptant
E	01/01/1947	Classe	Type
		Fictif	Comptant

c) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat de la requête suivante



Enregistrements

No. de client	Valeur totale	Comptes	
À	-10000	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
B	10001	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
C	0	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
D	10000	Classe	Type
		Réel	Régime d'épargne retraite
		Réel	Comptant
E	-0.01	Classe	Type
		Fictif	Comptant

## Étape 3

## Interface #1

- a) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du filtre suivant



Enregistrements :

Numéro de compte	Code de CP	Valeur totale	Devise	Marge
À	AC44	0	CAD	0
B	AC45	10000	CAD	0
C	ACC44	-1000	USD	0
D	AC43	0.01	USD	100
E	AC44	-0.01	USD	1000
F	AC45	1000	CAD	0
G	AC44	0.01	CAD	50000
H	AC 44	1000	CAD	-50000
I	AC45	-8500	USD	0
J	AC44	1000	CAD	0
K	AC45	0	USD	0

b) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du filtre suivant

**Clients**











Code de CP = AC44 ET Devise = CAD ET Nom contenant CHENIER ET Valeur totale ≠ 0

Enregistrements :

No. de compte	Code de CP	Valeur	Devise	Nom
À	AC44	0	USD	Mark Chenier
B	AC45	10000	CAD	Ann Chenier
C	ACC44	-1000	USD	Martin Chenier
D	AC43	0.01	USD	Chenier
E	AC44	-0.01	USD	Chen
F	AC45	1000	CAD	Chenier Rachel
G	AC44	0	CAD	CHENIER Mark
H	AC 44	1000	CAD	CHENIER
I	AC45	-8500	USD	CHENIERE ALEX
J	AC44	1000	CAD	MARK CHENIER SMITH
K	AC45	0	USD	KARINE GIRARD

c) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du filtre suivant

**Relations**



 Type = Famille  
 ET
 
 Marge = 0  
 ET
 
 Création est parmi les derniers 6 mois  

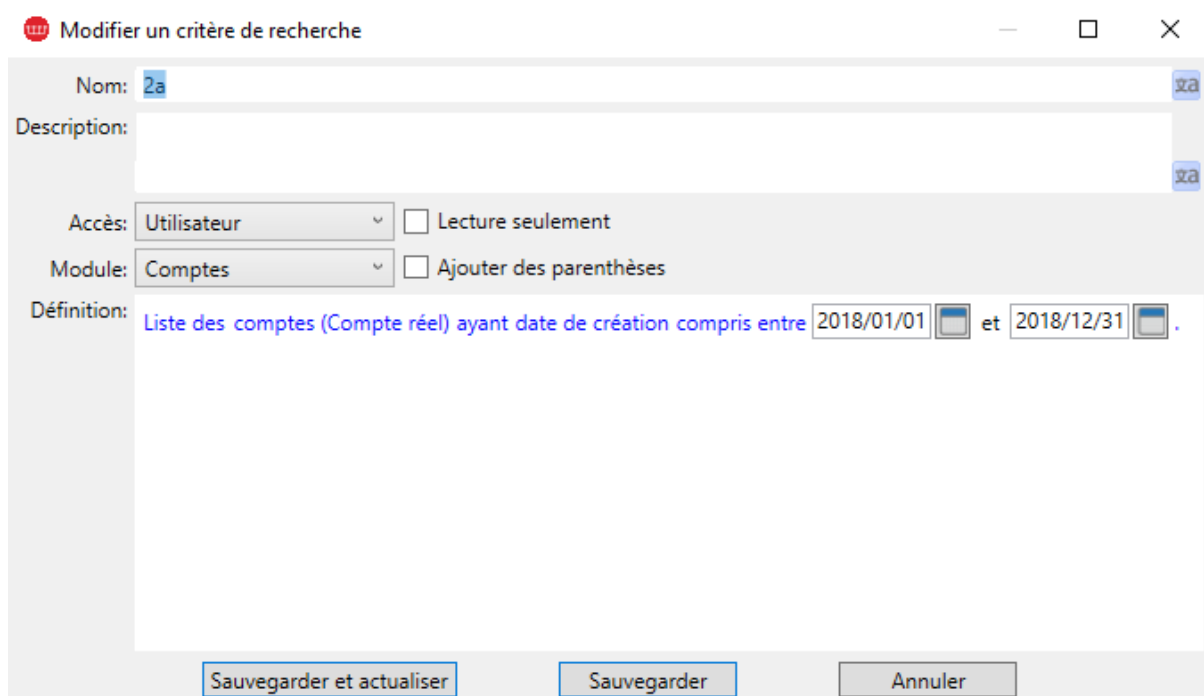
Enregistrements

No. de relation	Type	Marge	Devise	Date de création
À	Famille	0	USD	01/01/2018
B	Commercial	10000	CAD	05/01/2018
C	Commercial	-1000	USD	01/01/2001
D	Famille	0.01	USD	02/02/2005
E	Personnel	-0.01	USD	12/12/2017
F	Commercial	1000	CAD	08/08/2001
G	Famille	0.01	CAD	03/03/2018
H	Personnel	1000	CAD	04/04/2018
I	Commercial	-8500	USD	01/05/2008
J	Personnel	1000	CAD	02/15/2018
K	Famille	0	USD	02/15/2018



## Interface #2

- a) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche



The screenshot shows a window titled "Modifier un critère de recherche" with the following fields and options:

- Nom: 2a
- Description:
- Accès: Utilisateur  Lecture seulement
- Module: Comptes  Ajouter des parenthèses
- Définition: Liste des comptes (Compte réel) ayant date de création compris entre 2018/01/01 et 2018/12/31 .

Buttons at the bottom: Sauvegarder et actualiser, Sauvegarder, Annuler.

## Enregistrements

Numéro compte	de Classe	Marge	Devise	Date de création
À	Réel	0	USD	01/01/2018
B	Réel	10000	CAD	05/01/2018
C	fictif	-1000	USD	01/01/2001
D	Réel	0.01	USD	02/02/2005
E	Réel	-0.01	USD	12/12/2017
F	fictif	1000	CAD	08/08/2001
G	Réel	0.01	CAD	03/03/2018
H	Réel	1000	CAD	04/04/2018
I	fictif	-8500	USD	01/05/2008
J	Réel	1000	CAD	02/15/2018
K	Réel	0	USD	02/15/2018

b) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

Modifier un critère de recherche

Nom: 2b

Description:

Accès: Utilisateur  Lecture seulement

Module: Comptes  Ajouter des parenthèses

Définition: Liste des comptes (Compte réel) ayant valeur totale supérieur(e) à 0,00 \$ et  
ayant type égal(e) à Régime d'épargne-retraite ou  
ayant type égal(e) à Régime d'épargne-retraite - Conjoint et  
n'ayant pas de transactions avec date de transaction compris entre 2018/01/01 et 2018/12/31 et  
avec total inférieur(e) à 5 500,00 \$ et  
avec type égal(e) à Cotisation .

Sauvegarder et actualiser    Sauvegarder    Annuler

## Enregistrements

No cpt	Type	Classe	Valeur totale	Devise	Transactions			
À	Régime retraite	d'épargne	Réel	0	USD	Type	total	Date de transaction
						Cotisation	5500	02/02/2018
						Achat	100	01/01/2018
						Dépôt	1000	01/12/2018
B	Régime retraite - conjoint	d'épargne	Réel	10000	CAD	Type	total	Date de transaction
						Cotisation	1000	02/02/2017
C	Comptant		Réel	-1000	USD	Type	total	Date de transaction
						Cotisation	100000	02/02/2012
						Achat	100	01/01/2011
						Dépôt	99999	01/12/2018
D	Régime retraite	d'épargne	Réel	0.01	USD	Type	total	Date de transaction
						Cotisation	100000	02/02/2012
						Achat	100	01/01/2011
						Dépôt	99999	01/12/2018
E	Régime retraite	d'épargne	Réel	-0.01	USD	Type	total	Date de transaction
						Cotisation	999999	02/02/2018
						Cotisation	999999	01/01/2018
						Cotisation	1000	01/12/2018
F	Comptant		Réel	1000	CAD	Type	total	Date de transaction
						Dépôt	5500	02/02/2018
						Achat	100	01/01/2018

G	Régime retraite - conjoint	d'épargne	Réel	0.01	CAD	Type	total	Date de transaction
						Achat	100	01/01/2018
						Dépôt	1000	01/12/2018
H	Régime retraite	d'épargne	Réel	1000	CAD	Type	total	Date de transaction
						Cotisation	1000	02/02/2018
I	Comptant		Réel	-8500	USD	Type	total	Date de transaction
						Cotisation	500	02/02/2017
						Achat	100	01/01/2018
						Dépôt	1000	01/12/2018
J	Régime retraite	d'épargne	Réel	1000	CAD	Type	total	Date de transaction
						Cotisation	500	02/02/2017
K	Régime retraite	d'épargne	Réel	0	USD	Type	total	Date de transaction
						Achat	100	01/01/2018
						Dépôt	1000	01/12/2018

c) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

Modifier un critère de recherche

Nom: 2c

Description:

Accès: Utilisateur  Lecture seulement

Module: Clients  Ajouter des parenthèses

Définition: Liste des clients (Client réel) ayant devise égal(e) à CAD et  
ayant valeur totale supérieur(e) à 0,00 \$ et  
ayant type égal(e) à Corporatif ou  
ayant type égal(e) à Individu .

Sauvegarder et actualiser    Sauvegarder    Annuler

## Enregistrements :

Numéro de client	Classe	Type	Valeur totale	Devise	Date de création
À	Réel	Corporatif	0	USD	01/01/2018
B	Réel	Famille	10000	CAD	05/01/2018
C	Réel	Individu	-1000	USD	01/01/2001
D	Réel	Corporatif	0.01	USD	02/02/2005
E	Réel	Corporatif	-0.01	USD	12/12/2017
F	Réel	Famille	1000	CAD	08/08/2001
G	Réel	Individu	0.01	CAD	03/03/2018
H	Réel	Individu	1000	CAD	04/04/2018
I	Réel	Individu	-8500	USD	01/05/2008
J	Réel	Corporatif	1000	CAD	02/15/2018
K	Réel	Individu	0	USD	02/15/2018

## Interface #3

Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

**Modifier un critère de recherche** [X]

**Information**

Nom:

**Définition**

Rechercher des:

- comptes dont toutes ces conditions sont vraies (ET)
  - Classe du compte égale Compte réel
  - Valeur totale (compte) est plus grand que 0,00
- transactions dont toutes ces conditions sont vraies (ET)
  - Date de transaction est entre 2018/01/01 et 2018/12/31
  - Total de la transaction est plus petit que 5 500,00
  - Type de transaction égale Cotisation
- dont au moins une de ces conditions est vraie (OU)
  - Type de compte égale Régime d'épargne-retraite
  - Type de compte égale Régime d'épargne-retraite - Conjoint

**Opérateurs logiques**

**Condition**

Champ:

Opérateur:


Valeur:



No cpt	Type	Valeur totale	Devise	Classe	Transactions		
A	Comptant	1000	CAD	Réel	Type	total	Date de transaction
					Dépôt	5500	02/02/2018
					Achat	100	01/01/2018
B	Régime d'épargne retraite - conjoint	10000	CAD	Réel	Type	total	Date de transaction
					Cotisation	1000	02/02/2018
C	Comptant	-1000	USD	Réel	Type	total	Date de transaction
					Cotisation	100000	02/02/2012
					Achat	100	01/01/2011
					Dépôt	99999	01/12/2018
D	Régime d'épargne retraite	0	USD	Réel	Type	total	Date de transaction
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018
E	Régime d'épargne retraite	-0.01	USD	Réel	Type	total	Date de transaction
					Cotisation	999999	02/02/2018
					Cotisation	999999	01/01/2018
F	Régime d'épargne retraite - conjoint	0.01	CAD	Réel	Type	total	Date de transaction
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018

G	Comptant	-8500	USD	Réel	Type	total	Date de transaction
					Cotisation	5500	02/02/2018
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018
H	Régime d'épargne retraite	1000	CAD	Réel	Type	total	Date de transaction
					Cotisation	5500	02/02/2018
I	Régime d'épargne retraite	0	USD	Réel	Type	total	Date de transaction
					Cotisation	5500	02/02/2018
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018
K	Régime d'épargne retraite	0.01	USD	Réel	Type	total	Date de transaction
					Cotisation	100000	02/02/2012
					Achat	100	01/01/2011
					Dépôt	99999	01/12/2018
L	Régime d'épargne retraite	1000	CAD	Réel	Type	total	Date de transaction
					Cotisation	1000	02/02/2018

b) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

 Modifier un critère de recherche ×

**Information**

Nom:  Propriétés

**Définition**

Rechercher des:

- ▾ comptes dont toutes ces conditions sont vraies (ET)
  - Code de CP (compte) égale AC44
  - Devise (compte) égale CAD
- ▾ positions dont au moins une de ces conditions est vraie (OU)
  - Symbole égale BCE
  - Symbole égale IBM

**Opérateurs logiques**

**Condition**

Champ:  >

Opérateur:  >


Valeur:  >

## Enregistrements :

Numéro de compte	Code de CP	Marge	Devise	Positions	
A	AC44	0	USD	Quantité	symbole
				0	BCE
				0	IBM
				100	PJC
B	AC45	10000	CAD	Quantité	Symbole
				500	BCE
				0	IBM
				100	PJC
C	AC44	-1000	USD	Quantité	Symbole
				500	BCE
				500	IBM
				100	PJC
D	AC44	0.01	USD	Quantité	Symbole
				0	BCE
				500	IBM
				100	PJC
E	AC45	-0.01	USD	Quantité	Symbole
				100	PJC
F	AC44	1000	CAD	Quantité	Symbole
				500	CBE
				500	MBI
				100	PJC

G	AC44	0.01	CAD	Quantité	Symbole
				500	CBE
H	AC44	1000	CAD	Quantité	Symbole
				-500	IBM
I	AC 44	-8500	USD	Quantité	Symbole
				-500	BCE
J	AC44	1000	CAD	Quantité	Symbole
				1	BCE
K	AC45	0	USD	Quantité	Symbole
				0	BCE
				500	IBM
				0	PJC

c) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

 Modifier un critère de recherche ×

**Information**

Nom:  Propriétés

**Définition**

Rechercher des:

- ▾ clients dont toutes ces conditions sont vraies (ET)
  - Classe du client égale Client réel
  - Nom complet (client) débute par A
- ▾ excluant celles/ceux dont toutes ces conditions sont vraies (NON ET)
  - Courriel 1 contient yahoo
- ▾ dont au moins une de ces conditions est vraie (OU)
  - Date de naissance est entre 1946/01/01 et 1946/12/31
  - Devise (client) égale CAD

**Opérateurs logiques**

**Condition**

Champ:  >

Opérateur:  >

Valeur:  >

## Enregistrements

Numéro client	Classe	Nom complet	Courriel 1	Devise	Date naissance	de
A	Réel	Ann	Ann@croesus.com	USD	12/12/1946	
B	Réel	Antony	Antony@yahoo.com	CAD	25/08/1975	
C	Réel	Bertrand	Bertrand@live.com	CAD	11/11/1960	
D	fictif	Alan	Alan@croesus.com	USD	05/05/1985	
E	fictif	Claudia	Claudia@yahoo.com	CAD	04/04/2000	
F	Réel	Alex	Alex@yahoo.com	CAD	18/10/1946	
G	Réel	David	David@yahoo.com	USD	01/01/1946	
H	Réel	Anny	Anny@croesus.com	CAD	01/01/1999	
I	Réel	Erik	Eric@yahoo.com	USD	01/01/1946	
J	fictif	Frank	Frank@croesus.com	USD	31/12/1947	
K	Réel	Armand	Armand@croesus.com	CAD	05/05/1946	

## Interface #4

- a) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

The screenshot shows a window titled "Create advanced filter" with a "Définition" tab. The "Nom" field contains "a" and the "Accès" dropdown is empty. The main filter area is titled "Liste de Comptes" and contains the following conditions:

- Code de CP = AC44
- ET
- Devise = CAD
- ET
- incluant au moins 1 Positions
- OU
  - Symbole = BCE
  - Symbole = IBM

Each condition has edit and delete icons. There are also expand/collapse icons for the "incluant au moins 1 Positions" and "OU" sections. A "Validate" button is located at the bottom of the window.



## Enregistrements

Numéro compte	Code de CP	Marge	Devise	Positions	
A	AC45	-0.01	USD	Quantité	Symbole
				100	PJC
B	AC45	10000	CAD	Quantité	Symbole
				500	BCE
				0	IBM
				100	PJC
C	AC 44	-8500	USD	Quantité	Symbole
				-500	BCE
D	AC44	0.01	USD	Quantité	Symbole
				0	BCE
				500	IBM
				100	PJC
E	AC44	1000	CAD	Quantité	Symbole
				-500	IBM
F	AC44	1000	CAD	Quantité	Symbole
				500	CBE
				500	MBI
				100	PJC

G	AC44	0.01	CAD	Quantité	Symbole
				500	CBE

H	AC44	1000	CAD	Quantité	Symbole
				0	BCE

I	AC45	0	USD	Quantité	Symbole
				0	BCE
				500	IBM
				0	PJC

J	AC44	0	USD	Quantité	Symbole
				0	BCE
				0	IBM
				100	PJC

K	AC44	-1000	USD	Quantité	Symbole
				500	BCE
				500	IBM
				100	PJC

b) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche

Create advanced filter

Définition

Nom

Accès:

Liste de **Comptes** a

Classe = Compte réel

ET

Valeur totale > 0

ET

Type parmi Régime d'épargne-retraite, Régime d'épargne-retraite - Conjoint

ET

incluant au moins 1 Transactions a

Type = Cotisation

ET

Total < 5500

ET

Date de transaction entre (inclusivement) 2018/01/01 et 2018/12/31

Validate

No cpt	Type	Valeur totale	Devi se	Classe	Transactions		
A	Comptant	-8500	USD	Réel	Type	Total	date
					Cotisation	5500	02/02/2018
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018
B	Régime d'épargne retraite	0	USD	Réel	Type	Total	date
					Cotisation	5500	02/02/2018
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018
C	Régime d'épargne retraite	-0.01	USD	Réel	Type	Total	date
					Cotisation	999999	02/02/2018
					Cotisation	999999	01/01/2018
					Cotisation	1000	01/12/2018
D	Comptant	-1000	USD	Réel	Type	Total	date
					Cotisation	100000	02/02/2018
					Achat	100	01/01/2018
					Dépôt	99999	01/12/2018
E	Comptant	1000	CAD	Réel	Type	Total	date
					Dépôt	5500	02/02/2018
					Achat	100	01/01/2018

F	Régime d'épargne retraite - conjoint	0.01	CAD	Réel	Type	Total	date
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018

G	Régime d'épargne retraite	1000	CAD	Réel	Type	Total	date
					Cotisation	1000	02/02/2018

H	Régime d'épargne retraite	1000	CAD	Réel	Type	Total	date
					Cotisation	5500	02/02/2018

I	Régime d'épargne retraite	0	USD	Réel	Type	Total	date
					Achat	100	01/01/2018
					Dépôt	1000	01/12/2018

J	Régime d'épargne retraite - conjoint	10000	CAD	Réel	Type	Total	date
					Cotisation	1000	02/02/2018

K	Régime d'épargne retraite	0.01	USD	Réel	Type	Total	date
					Cotisation	100000	02/02/2018
					Achat	100	01/01/2018
					Dépôt	99999	01/12/2018

c) Identifiez le ou les enregistrement(s) qui devraient faire partie du résultat du critère de recherche



Create advanced filter

Définition

Nom


Accès:



^ Liste de **Clients** a

**Nom complet** débutant par **A**  


ET



ET

^ **excluant tous(tes)** **Clients** 



**Courriel 1** **contenant**   


ET


^ **incluant au moins 1** **Clients** a 

**Date de naissance** entre (inclusivement)  et   

OU

**Devise = CAD**  





Validate








## Enregistrements :

Numéro client	Classe	Nom Complet	Courriel 1	Devise	Date de naissance
A	Réel	Antony	Antony@yahoo.com	CAD	25/08/1975
B	Réel	David	David@yahoo.com	USD	01/01/1946
C	fictif	Alan	Alan@croesus.com	USD	05/05/1985
D	fictif	Claudia	Claudia@yahoo.com	CAD	04/04/2000
E	Réel	Alex	Alex@yahoo.com	CAD	18/10/1946
F	fictif	Frank	Frank@croesus.com	USD	31/12/1946
G	Réel	Anny	Anny@croesus.com	CAD	01/01/1999
H	Réel	Erik	Eric@yahoo.com	USD	01/01/1946
I	Réel	Ann	Ann@croesus.com	USD	12/12/1946
J	Réel	Armand	Armand@croesus.com	CAD	05/05/1946
K	Réel	Bertrand	Bertrand@live.com	CAD	11/11/1960

## Étape 4

A)

**Comptes**

▽+  Classe = Compte réel   ET Code de CP = AC44   ET Valeur totale > 0  



B)

Modifier un critère de recherche

Nom:

Description:

Accès:   Lecture seulement

Module:   Ajouter des parenthèses

Définition: Liste des comptes (Compte réel) ayant code de CP égal(e) à  et  
ayant valeur totale supérieur(e) à  \$.

C)

**Modifier un critère de recherche** [X]

**Information**

Nom:

**Définition**

Rechercher des:

▲ comptes dont toutes ces conditions sont vraies (ET)  
Classe du compte égale Compte réel  
Code de CP (compte) égale AC44  
Valeur totale (compte) est plus grand que 0,00

**Opérateurs logiques**

**Condition**

Champ:  >   
Opérateur:  >   
Valeur:  >

D)

Create advanced filter

Définition

Nom |

Accès: ▾

^ Liste de **Comptes** a

Classe = Compte réel ✎ 🗑

ET

Code de CP = AC44 ✎ 🗑

ET

Valeur totale > 0 ✎ 🗑

▽+

Validate

## **ANNEXE B TESTS D'UTILISABILITÉ PHASE 1 – CYCLE 3 ET PHASE 2**

### **Première étape**

La première étape consiste en un formulaire démographique.

### **Deuxième étape**

Dans la deuxième étape, je présente le fonctionnement des trois interfaces qui seront utilisées durant le test. Vous aurez l'occasion de poser des questions sur le fonctionnement de chacune des interfaces.

### **Troisième étape**

Dans la troisième étape, une tâche vous est soumise pour vous permettre de vous familiariser avec les différentes interfaces. Vous pourrez poser des questions. Cette étape ne sera pas considérée dans le test d'utilisabilité

### **Quatrième étape**

Dans la quatrième étape, vous aurez des tâches à exécuter sur les différentes interfaces qui font partie du projet de recherche. Vous devrez formuler des critères dans l'interface à partir d'une demande qui est formulée en français normal.

Chaque tâche sera chronométrée.

Chaque tâche sera enregistrée à l'aide de l'application Camtasia.

### **Cinquième étape**

Dans la cinquième étape, vous devez placer les différentes interfaces en ordre de préférence. Vous devez donner un commentaire positif et un commentaire négatif pour chacune des 4 interfaces. Vous serez invité à commenter votre expérience d'utilisation des interfaces de formulation de critères.

Cette étape sera enregistrée.

## Étape 1

## Données démographiques

Participant					
Occupation					
Genre	Masculin		Féminin		
Âge	18 à 30	31 à 40	41 à 50	50 à 60	60 et +
Niveau de scolarité	Secondaire	Collège	Bacc	Maîtrise	Doctorat
Nombre d'années d'expérience avec Croesus Conseiller	Moins de 1 an	1 à 3 ans	3 à 5 ans	5 à 10 ans	plus de 10 ans
Nombre d'années d'expérience dans le domaine financier	Moins de 1 an	1 à 3 ans	3 à 5 ans	5 à 10 ans	plus de 10 ans
Niveau de familiarité avec la technologie en général	minimum	moyen	avancé	expert	
Fréquence d'utilisation de moteur de recherche	1 fois par année	tous les 3 mois	tous les mois	toutes les semaines	tous les jours

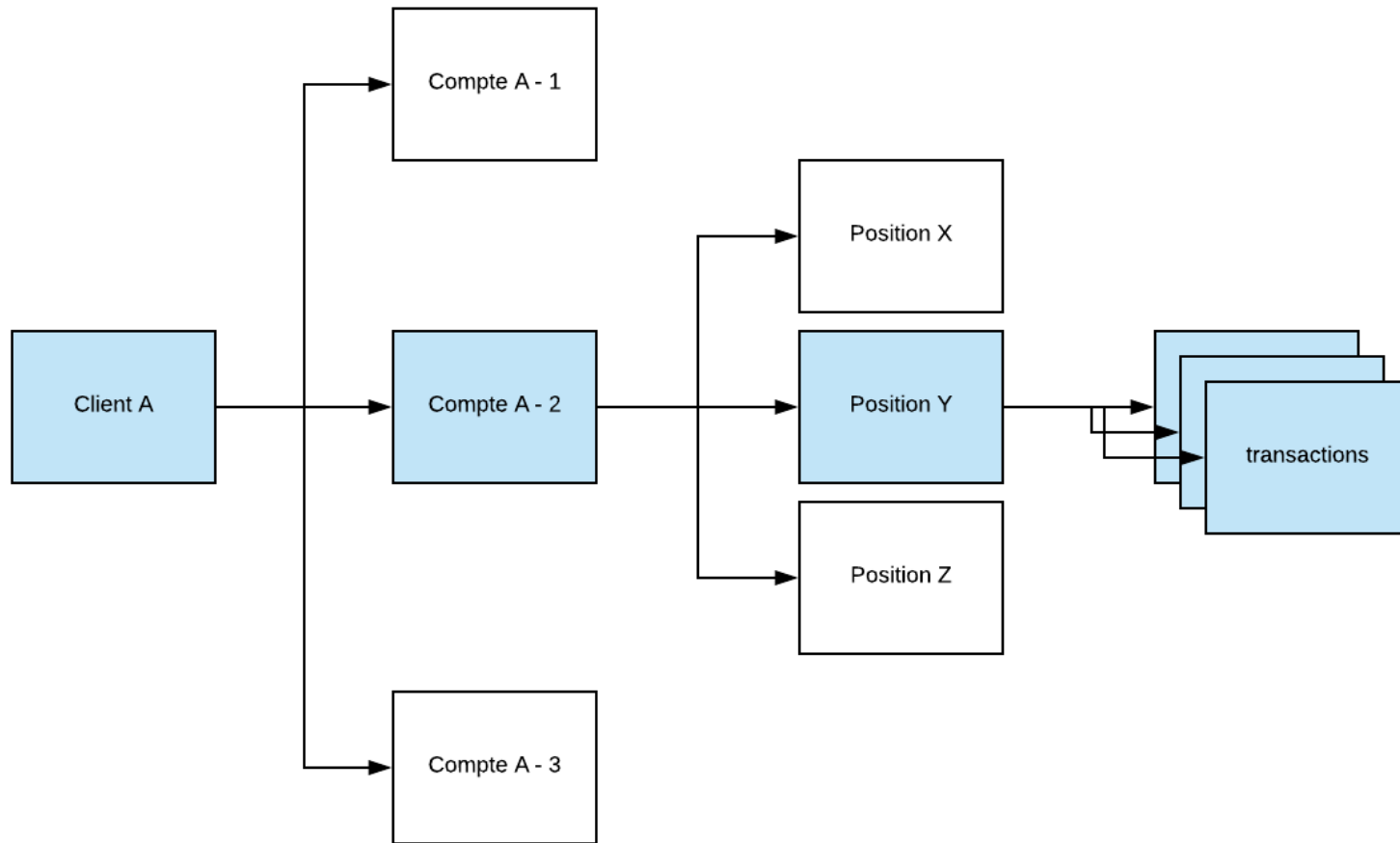


Figure B.1 Graphique présentant le modèle logique d'entités disponibles pour la création de requêtes.

**Tâche de familiarisation 1**

Liste des comptes réels

dont le type est parmi

- Régime d'épargne retraite
- Régime d'épargne retraite - conjoint

ayant fait au moins une transaction

- dont le type est achat
- dont la date de transaction est entre le 1 janvier 2017 et le 1 juillet 2017 (inclusif)

Détenant le titre PJC dans leur portefeuille (position avec quantité  $\leq 0$ )

**Tâche de familiarisation 2**

Liste des clients réels

dont le nom abrégé du client débute par l'une des lettres suivantes

- B
- C
- G

dont le numéro de client ne contient pas \$

dont le code postal débute par J7P OU dont la ville est Laval

dont la devise du client est CAD



**Tâche 1**

Liste des clients réels

dont le code de cp fait partie de la liste suivante :

- AC42
- AC44
- AC47

dont la marge est supérieure ou égale à 0,00

dont la date de dernier contact est entre le 1e janvier 2017 et le 1e mai 2017 (inclusivement)

**Tâche 2**

Liste des clients réels

dont la valeur totale est supérieure à zéro

Ayant effectué au moins une transaction

- de type cotisation
- où la date de transaction est durant l'année 2017

Comprenant au moins 1 compte (lié à) dont le type est parmi la liste suivante :

- Régime d'épargne retraite
- Régime d'épargne retraite - conjoint

dont la devise du client est CAD

**Tâche 3**

Liste des clients réels

dont le chargé de comptes est Albert Einstein

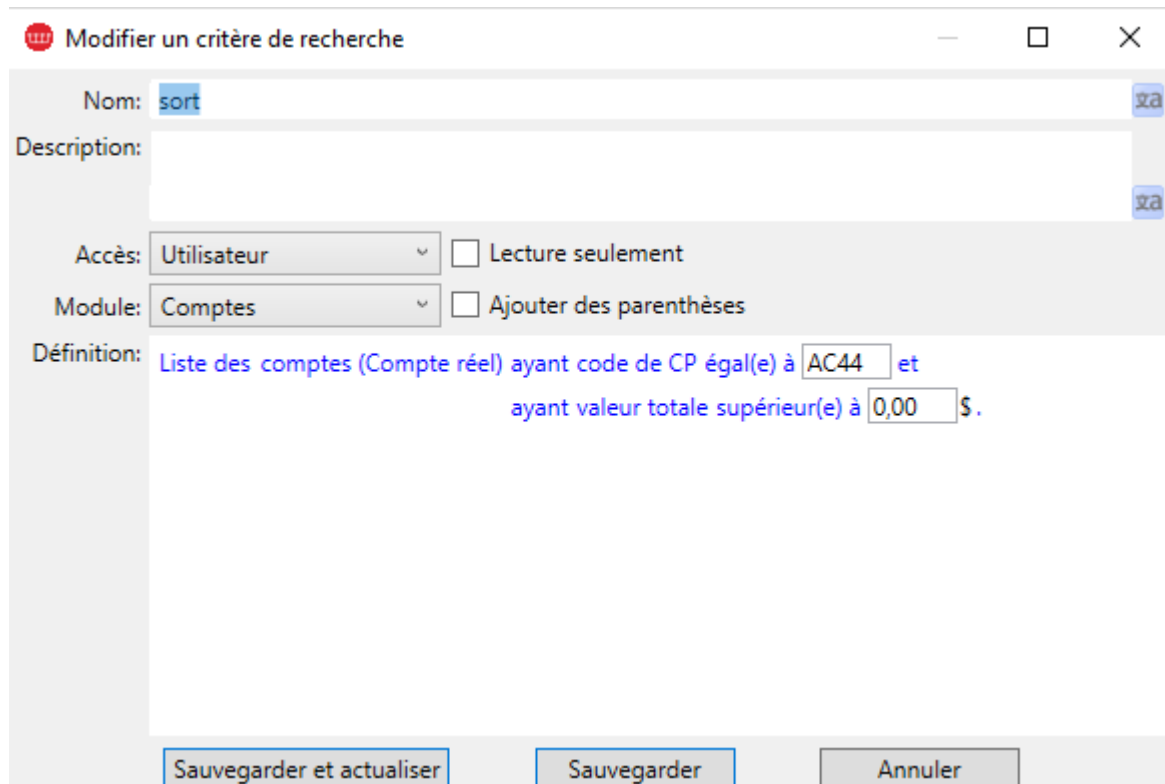
ayant au moins 1 compte

- dont le type est Régime d'épargne retraite
- sur lequel il y a une transaction
  - o de type achat
  - o dont la date de transaction est en 2017

**Tâche 4**

Placer les interfaces suivantes en ordre de préférence :

A)



The screenshot shows a window titled "Modifier un critère de recherche" with the following fields and options:

- Nom:  za
- Description:
- Accès:   Lecture seulement
- Module:   Ajouter des parenthèses
- Définition: Liste des comptes (Compte réel) ayant code de CP égal(e) à  et ayant valeur totale supérieur(e) à  \$.

Buttons at the bottom:

B)

**Modifier un critère de recherche** [X]

**Information**

Nom:

**Définition**

Rechercher des:

comptes dont toutes ces conditions sont vraies (ET)

- Classe du compte égale Compte réel
- Code de CP (compte) égale AC44
- Valeur totale (compte) est plus grand que 0,00

**Opérateurs logiques**

**Condition**

Champ:

Opérateur:

Valeur:

C)

Create advanced filter

**Définition**

Nom:  xa

Description:  xa

Accès:

Liste de **Comptes réels** ayant

- Code de CP = AC44

**ET**

- Valeur totale >= 0

Validate

**Grille de résultats**

	<b>Temps</b>	<b>Nombre de question</b>
<b>familiarisation 1 - A</b>		
<b>familiarisation 2 - A</b>		
<b>1 - A</b>		
<b>2 - A</b>		
<b>3 - A</b>		
<b>familiarisation 1 - B</b>		
<b>familiarisation 2 - B</b>		
<b>1 - B</b>		
<b>2 - B</b>		
<b>3 - B</b>		
<b>familiarisation 1 - C</b>		
<b>familiarisation 2 - C</b>		
<b>1 - C</b>		
<b>2 - C</b>		
<b>3 - C</b>		

## ANNEXE C RÉSULTATS STATISTIQUES POUR LES PHASES 1 ET 2

Le tableau C.1 présente les données brutes du temps de traitement en secondes pour toutes les tâches effectuées avec toutes les interfaces et pour tous les sujets ayant participé à la phase 1. L'interface 1 correspond à la fonctionnalité des filtres sur des listes d'éléments. L'interface 2 correspond à l'interface des critères simplifiée. L'interface 3 correspond à l'interface des critères avancée et l'interface 4 correspond à l'interface qui a été produite dans le prototype de formulaire flexible de formulation de requêtes par diagramme.

Tableau C.1 Temps d'exécution des tâches effectuées durant la phase 1 des tests d'utilisabilité.  
(en secondes)

# sujet	int. 1	int. 1	int. 1	int. 2	int. 2	int. 2	int. 3	int. 3	int. 3	int. 4	int. 4	int. 4
1-2-3-4	56	50	40	43	146	67	175	37	72	63	84	75
1-3-2-4	18	42	58	37	115	44	119	60	88	40	98	54
1-4-2-3	36	48	49	39	99	36	80	34	58	58	100	69
2-1-4-3	46	46	38	44	145	37	75	38	54	54	72	42
2-3-1-4	54	75	47	53	121	66	158	80	58	50	45	44
2-4-1-3	20	49	39	36	96	38	72	44	79	38	78	35
3-1-4-2	55	37	28	45	70	35	174	56	53	36	108	33
3-2-4-1	30	30	24	34	112	39	170	39	91	38	80	33
3-4-2-1	34	42	47	30	58	35	122	45	48	44	71	41
4-1-3-2	30	35	31	35	120	34	94	46	35	50	105	34
4-2-3-1	12	37	27	26	99	36	79	77	34	40	83	47
4-3-1-2	31	47	32	29	97	40	117	52	53	74	89	106

Le tableau C.2 présente les données brutes pour le taux de succès des différentes interfaces pour les différentes tâches pour tous les sujets. Les valeurs indiquent le nombre d'enregistrements correctement identifiés sur un total de 11.

Tableau C.2 Taux de succès pour les tâches du test d'utilisabilité de la phase 1 (la valeur affichée correspond au nombre de réponses valide sur un total de 11)

# sujet	Int 1	Int 1	Int 1	Int 2	Int 2	Int 2	Int 3	Int 3	Int 3	Int 4	Int 4	Int 4
1-2-3-4	10	6	11	6	6	8	10	8	10	7	10	10
1-3-2-4	6	9	10	11	8	9	9	10	6	11	11	7
1-4-2-3	9	8	11	11	8	9	11	11	9	11	11	7
2-1-4-3	9	9	11	10	9	9	10	11	9	11	7	7
2-3-1-4	11	8	7	6	6	9	10	11	7	8	8	11
2-4-1-3	10	8	10	11	4	10	9	11	10	11	9	7
3-1-4-2	11	9	11	10	8	9	8	10	7	11	8	7
3-2-4-1	9	8	11	10	9	9	8	10	9	11	9	9
3-4-2-1	9	9	11	10	6	8	9	11	7	11	9	7
4-2-3-1	10	8	11	11	10	8	9	10	10	10	11	10
4-1-3-2	10	7	11	11	7	9	9	11	7	11	7	6
4-3-1-2	10	8	11	11	7	9	11	11	9	11	11	9
Somme	114	97	126	118	88	106	113	125	100	124	111	97
Moyenne	9.5	8.1	10.5	9.8	7.3	8.8	9.4	10.4	8.3	10.3	9.3	8



Le tableau C.3 suivant donne les résultats de préférence des utilisateurs pour les différentes interfaces. 1 étant la préférée et 3 la moins appréciée.

Tableau C.3 Taux de succès pour les tâches du test d'utilisabilité de la phase 1 (la valeur affichée correspond au nombre de réponses valide sur un total de 11)

	Interface 1	Interface 2	Interface 3	Interface 4
1-2-3-4	1	4	3	2
1-3-2-4	2	3	4	1
1-4-2-3	1	2	3	4
2-1-4-3	3	2	4	1
2-3-1-4	2	3	4	1
2-4-1-3	2	3	4	1
3-1-4-2	2	3	4	1
3-2-4-1	1	2	4	3
3-4-2-1	1	4	3	2
4-1-3-2	1	3	4	2
4-2-3-1	1	3	4	2
4-3-1-2	3	2	4	1
	20	34	45	21

## Résultats statistiques pour le temps de traitement pour la phase 2

Le tableau C.4 présente les données brutes du temps de traitement en secondes pour toutes les tâches effectuées avec toutes les interfaces et pour tous les sujets ayant participé à la phase 2.

Tableau C.4 Temps de traitement des tâches de la phase 2 en secondes

	Int	A	Int	A	Int	A	Int	B	Int	B	Int		Int	C	Int	C	Int	C
C-A-B		101		148		90		93		135				64		116		65
A-C-B		173		170		115		116		124				92		93		71
B-A-C		137		126		87		127		149				55		72		64
A-B-C		125		180		104		106		115				88		146		87
C-A-B		185		427		193		177		245				184		266		149
B-A-C		126		355		167		167		356				83		165		76
B-C-A		157		449		151		269		289				84		211		136
A-B-C		222		415		484		412		385				167		315		158
C-B-A		89		119		91		87		154				61		87		76
B-C-A		181		356		140		396		394				93		206		127
C-B-A		110		152		96		129		153				82		93		68
A-C-B		140		369		119		229		185				73		152		66
Somme		1746		3266		1837		2308		2684				1126		1922		1143
Moyenne		145.5		272.16		153.08		192.3		223.7				93.8		160.17		92.25

Le tableau C.5 présente les données brutes du nombre d'erreurs pour toutes les tâches effectuées avec toutes les interfaces et pour tous les sujets ayant participé à la phase 2.

Tableau C.5 Données brutes du nombre d'erreurs pour les tâches du test d'utilisabilité

	A-1	A-2	A-3	B-1	B-2	B-3	C-1	C-2	C-3
C-A-B	0	0	0	0	0	0	0	0	0
A-C-B	0	0	0	0	0	0	1	2	1
B-A-C	0	0	0	0	2	0	0	0	0
A-B-C	0	0	2	0	0	0	1	0	0
C-A-B	0	1	1	0	4	4	0	0	0
B-A-C	0	1	1	1	5	5	0	0	0
B-C-A	1	1	0	0	4	5	0	0	0
A-B-C	2	0	1	0	3	5	0	1	1
C-B-A	0	0	1	1	0	0	0	0	0
B-C-A	0	1	2	0	3	5	0	3	1
C-B-A	0	0	1	0	0	1	0	0	0
A-C-B	0	2	2	1	0	0	0	2	2
Somme	3	4	9	2	21	25	2	6	3
Moyenne	0.25	0.33	0.75	0.17	1.75	2.08	0.17	0.5	0.25

Le tableau C.6 présente les données brutes du nombre de demandes d'assistance à la tâche des sujets pour toutes les tâches effectuées avec toutes les interfaces et pour tous les sujets ayant participé à la phase 2.

Tableau C.6 Données brutes du nombre de demandes d'assistance à la tâche des sujets pour les tâches du test d'utilisabilité

	A-1	A-2	A-3	B-1	B-2	B-3	C-1	C-2	C-3
C-A-B	0	0	0	0	0	0	0	0	0
A-C-B	1	0	0	0	0	0	0	0	0
B-A-C	0	0	0	0	0	0	0	0	0
A-B-C	0	0	0	0	0	0	0	0	0
C-A-B	0	5	1	2	2	0	0	4	2
B-A-C	0	1	0	0	0	0	0	0	0
B-C-A	0	3	0	1	0	0	0	1	0
A-B-C	0	1	1	2	0	0	1	0	0
C-B-A	0	0	0	0	0	0	0	0	0
B-C-A	0	1	0	4	2	2	1	0	0
C-B-A	0	1	0	0	0	0	1	0	0
A-C-B	1	3	0	3	0	1	0	1	0
Somme	1	12	2	9	4	2	3	5	2
Moyenne	0.08	1	0.17	0.42	0.33	0.17	0.25	0.41	0.17

Le tableau C.7 suivant donne les résultats de préférence des utilisateurs pour les différentes interfaces. 1 étant la préférée et 3 la moins appréciée.

Tableau C.7 Préférence des sujets

	Interface A	Interface B	Interface C
3-1-2-	3	2	1
1-3-2-	2	3	1
2-1-3-	3	1	2
1-2-3-	3	1	2
3-1-2-	3	2	1
2-1-3-	2	3	1
2-3-1-	3	2	1
1-2-3-	2	3	1
3-2-1-	2	3	1
2-3-1-	2	3	1
3-2-1-	3	2	1
1-3-2-	2	3	1
	30	28	14