



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Contribution à la méthodologie des bases de données temporelles

Tallier, Benoît

Award date:
1997

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix
Namur

Institut d'informatique

Contribution à la méthodologie des bases de données temporelles

Tallier Benoit

Mémoire présenté pour l'obtention du grade
de licencié et maître en
informatique

Promoteur : **HAINAUT J-L.**

Année académique 1996-1997

Contribution à la méthodologie des bases de données temporelles

Résumé

Le temps constitue une dimension très importante de la réalité. Néanmoins, le modèle relationnel actuel ne supporte pas cette dimension. Dans ce travail, nous étudions la façon de représenter cet aspect. Nous définissons tout d'abord un modèle temporel. Il s'agit d'une extension d'un modèle conceptuel de base correspondant à une restriction du modèle Entité-Association. Ensuite, nous introduisons des transformations qui expriment le temps à l'aide des concepts du modèle conceptuel. Grâce à celles-ci, nous proposons un plan qui permet de représenter le temps au niveau relationnel. Enfin, nous présentons deux exemples pour appliquer ce plan de transformation.

Abstract

The time constitutes an important dimension of reality. Nevertheless, the current relational model doesn't support the time dimension. In this work, we study how to represent this aspect. First, we define a temporal model. It extends a conceptual model corresponding to a restriction of E-R model. We introduce then transformations that express time with the concepts of conceptual model. After that, we propose a plan using the transformations. This plan allows to introduce time in the relational model. Finally, we present two examples to apply the proposed plan.

Avant-propos

Je tiens à remercier mon promoteur, Mr Jean-Luc Hainaut, pour ces conseils et son aide précieuse lors de la réalisation de ce mémoire.

Je remercie vivement ma famille qui m'a soutenu et encouragé pendant ces années d'études.

Que toutes les personnes qui ont également participé, de près ou de loin, à la réalisation de ce travail en soient très sincèrement remerciées.

Table des matières

I. INTRODUCTION	9
II. TYPES DE TEMPS ET MODÉLISATIONS DU TEMPS.....	13
A. TYPES DE TEMPS	13
B. MODÉLISATIONS DU TEMPS	14
C. CHOIX DE LA MODÉLISATION ET DES TYPES DE TEMPS.....	15
III. VUE D'ENSEMBLE DES ALGÈBRES TEMPORELLES.....	17
A. ALGÈBRE TEMPORELLE DE JONES.....	18
B. ALGÈBRE TEMPORELLE DE BEN-ZVI.....	19
C. ALGÈBRE TEMPORELLE DE NAVATHE.....	20
D. ALGÈBRE TEMPORELLE DE SADEGHI	21
E. ALGÈBRE TEMPORELLE DE SARDA	22
F. ALGÈBRE TEMPORELLE DE CLIFFORD	23
G. ALGÈBRE TEMPORELLE DE TANSEL.....	25
H. ALGÈBRE TEMPORELLE DE GADIA	26
I. ALGÈBRE TEMPORELLE DE YEUNG	27
J. ALGÈBRE TEMPORELLE DE LORENTZOS.....	28
K. ALGÈBRE TEMPORELLE DE MAC-KENZIE.....	29
L. ALGÈBRES TEMPORELLES DE TUZHILIN	30
IV. LES MODÈLES.....	31
A. MODÈLE CONCEPTUEL DE BASE	31
B. MODÈLE CONCEPTUEL TEMPOREL.....	33
C. MODÈLE RELATIONNEL	36
V. TRANSFORMATIONS DES STRUCTURES DE DONNÉES TEMPORELLES.....	39
A. TRANSFORMATIONS TER \rightarrow TER.....	39
B. TRANSFORMATIONS TER \rightarrow ER	46
C. TRANSFORMATIONS ER \rightarrow RM.....	56
VI. RÈGLES DE COHÉRENCE.....	59
A. NOTATIONS	59
B. RÈGLES DE COHÉRENCE DES SCHÉMAS TEMPORELS	59
C. RÈGLES DE COHÉRENCE POST-TRANSFORMATIONS	60
VII. PLANS DE TRANSFORMATION D'UN SCHÉMA TEMPOREL.....	65
A. INCOMPATIBILITÉS	65
B. PREMIER PLAN DE TRANSFORMATION	65
C. SECOND PLAN DE TRANSFORMATION	67

VIII. ETUDES DE CAS.....	69
A. PREMIÈRE ÉTUDE DE CAS	69
B. DEUXIÈME ÉTUDE DE CAS.....	80
IX. CONCLUSION	85
X. BIBLIOGRAPHIE.....	87

I. Introduction

Le temps constitue un aspect de la réalité très important. Il est lié à chaque événement qui se produit dans le monde réel. Chaque fait est lié à un instant précis, chaque objet possède une existence qui lui est propre. Or, les bases de données et les modèles ont pour but de capturer la réalité le plus fidèlement possible. Il serait intéressant de pouvoir représenter cette dimension temporelle à ces niveaux. D'autre part, il est également essentiel pour beaucoup d'applications informatiques de pouvoir disposer de données temporelles.

Actuellement, le modèle relationnel ne supporte pas l'aspect temporel des données. Les tables stockent des données dites instantanées c'est-à-dire qu'elles contiennent l'état des données à un instant particulier, généralement l'état courant (cfr. Figure I-1).

EMPLOYE			
NOM	ADRESSE	LOCALITE	FONCTION
Durant	Rue Albert 1er 55	Saint-Pierre	sous-directeur
Leroy	Rue dessous la ville 18	Libramont	Analyste
Dupont	Rue des pierres 41	Libramont	sous-directeur
Guillaume	Rue du monument 31	Bastogne	employe

Figure I-1 - Une table instantanée (non temporelle)

Avec de telles tables, il n'est pas possible de conserver l'état passé des données. Si nous désirons connaître l'historique des données, nous sommes obligés de modifier la structure des tables. Il est nécessaire d'ajouter deux colonnes (de nom DEBUT et FIN) pour représenter la période de validité des données. Dans notre cas, un employé est donc représenté par une succession d'états des données le concernant. L'état courant se distingue des autres par la colonne de fin de période (de nom FIN) qui contient la valeur NULL. Cette valeur représente l'infini temporel. C'est la plus grande valeur possible contenue par cette colonne FIN.

EMPLOYE					
NOM	DEBUT	FIN	ADRESSE	LOCALITE	FONCTION
Durant	10-06-1972	08-04-1979	Rue du feu 36	Saint-Pierre	ouvrier
Durant	08-04-1979	19-09-1983	Rue du feu 36	Saint-Pierre	employe
Durant	19-09-1983	14-04-1989	Rue de Tervuren 30	Bruxelles	employe
Durant	14-04-1989	29-12-1992	Grand Place 3	Bruxelles	sous- directeur
Durant	29-12-1992	NULL	Rue Albert 1er 55	Saint-Pierre	sous- directeur
Leroy	01-06-1990	NULL	Rue dessous la ville 18	Libramont	Analyste
Dupont	01-06-1972	27-09-1983	Rue des pierres 41	Libramont	vendeur
Dupont	27-09-1983	13-11-1995	Rue des pierres 41	Libramont	commercial
Dupont	13-11-1995	NULL	Rue des pierres 41	Libramont	sous- directeur
Guillaume	19-03-1983	13-06-1987	Rue des Champs 48	Libramont	employe
Guillaume	13-06-1987	15-09-1989	Rue du Serpont 301	Libramont	employe
Guillaume	15-09-1989	16-05-1993	Rue du monument 31	Bastogne	employe
Guillaume	16-05-1993	NULL	Rue du monument 31	Bastogne	sous- directeur

Figure I-2 - Une table temporelle (historisation des tuples)

Dans la Figure I-2, nous voyons par exemple que l'employé Durant a habité Saint-Pierre à partir du 10-06-1972 jusqu'au 19-09-1983 non inclus, date à laquelle il a déménagé à Bruxelles. Depuis le 29-12-1992, il est revenu à Saint-Pierre et nous constatons qu'il y habite toujours.

La technique que nous venons d'appliquer pour rendre la table temporelle réalise une *historisation des tuples*¹. Elle insère une nouvelle ligne (i.e. un nouvel état) chaque fois qu'il y a modification d'une ou plusieurs valeurs de la donnée.

Il y a également une autre technique qui réalise *l'historisation des attributs*. Elle représente chaque colonne par une table qui reprend l'historique des valeurs de cette colonne pour chaque employé. Un exemple est donné par la Figure I-3 qui renseigne l'historisation de FONCTION.

¹ Un tuple est défini comme la ligne d'une table d'une base de données

EMPLOYE			
NOM	DEBUT	FIN	FONCTION
Durant	10-06-1972	08-04-1979	ouvrier
Durant	08-04-1979	14-04-1989	employe
Durant	14-04-1989	NULL	sous-directeur
Leroy	01-06-1990	NULL	analyste
Dupont	01-06-1972	27-03-1983	vendeur
Dupont	27-03-1983	13-11-1995	commercial
Dupont	13-11-1995	NULL	sous-directeur
Guillaume	19-03-1983	16-05-1993	employe
Guillaume	16-05-1993	NULL	sous-directeur

Figure I-3 - Une table temporelle (historisation de l'attribut FONCTION)

Les bases de données temporelles intéressent de plus en plus les chercheurs. Nous observons durant ces dernières années une publication accrue sur le sujet. Le domaine qui attire à l'heure actuelle le plus de gens est le développement de modèles temporels. Grâce à ces modèles, nous pourrions représenter la dimension du temps au sein des bases de données actuelles. Les chercheurs se sont alors directement focalisés sur l'extension du modèle conceptuel, y compris l'extension de l'algèbre relationnelle. Nous rappelons que l'algèbre relationnelle est un ensemble de relations et une collection d'opérations. Une relation¹ est un sous-ensemble de produit cartésien de N ensembles D_i appelés domaines. Une relation est donc un ensemble d'éléments de la forme (v_1, v_2, \dots, v_N) , avec $v_i \in D_i$ où $1 \leq i \leq N$, que nous appelons N -uplets ou tuple. La collection d'opérations comprend cinq opérations de base : Union, Différence d'ensembles, Produit cartésien, Sélection et Projection.

Ce mémoire a pour objectif de se définir un modèle conceptuel temporel, de considérer un modèle relationnel et de développer un plan de transformation qui nous permet de représenter le temps d'une manière simple dans le modèle relationnel. Dans un premier temps, nous nous fixons un modèle conceptuel temporel. Il étend un modèle conceptuel de base correspondant à une restriction du modèle Entité-Association². Il contient donc non seulement les concepts du modèle conceptuel de base mais aussi des structures de données temporelles. Puis, nous considérons le modèle relationnel dans lequel nous souhaitons représenter l'aspect temporel des données. Nous définissons ensuite un ensemble de transformations de structures de données temporelles, qui expriment de manière simple l'aspect temporel à l'aide de concepts du modèle conceptuel de base. Grâce à ces transformations, nous développons un plan transformant un schéma temporel en un autre conforme au modèle relationnel. Néanmoins, vu la difficulté pour exprimer la dimension temporelle dans les bases de données conventionnelles, nous n'avons considéré que quelques-unes de ces structures temporelles. D'où, l'ensemble des transformations définies n'est pas exhaustif.

¹ [DELMAL1995]

² [BODART1994]

Le reste du mémoire est organisé comme suit. Dans le chapitre II, nous présentons les types de temps existants et les différentes modélisations du temps. Dans le chapitre III, nous proposons une vue d'ensemble des travaux déjà réalisés sur le sujet. Ensuite, nous décrivons dans le chapitre suivant les différents modèles utilisés dans ce travail. Dans le chapitre V, nous abordons les transformations qui expriment la dimension temporelle à l'aide des concepts du modèle de base. Dans le chapitre VI, nous évoquons les différentes règles de cohérence qu'il faut respecter suite à l'application des transformations vues au chapitre précédent. Dans le chapitre VII, nous présentons un premier plan qui transforme un schéma du modèle conceptuel temporel en un schéma conforme au modèle relationnel. Un second plan de transformation, moins rigide que le premier, est aussi proposé. Dans le chapitre VIII, nous effectuons deux études de cas pour illustrer ce que nous avons développé dans les chapitres qui précèdent. Enfin, nous terminerons ce travail par une conclusion.

II. Types de temps et modélisations du temps

Pour représenter les données temporelles, il est nécessaire de déterminer plusieurs éléments. Tout d'abord, il est important de savoir quel type de temps nous désirons incorporer dans le modèle relationnel. Il s'agit de déterminer si l'utilisateur veut connaître par exemple uniquement la validité des données ou si son intérêt se porte non seulement sur la validité mais aussi sur le moment d'enregistrement de la donnée dans la base de données. Les différents types de temps seront abordés dans le premier point qui suit. Un deuxième élément est l'identification de la modélisation du temps. Nous définirons quelles sont les deux modélisations fréquemment utilisées. Enfin, nous indiquerons ensuite les types de temps et la modélisation que nous choisissons pour notre modèle relationnel.

A. Types de temps

Dans une base de données, l'utilisateur peut y inclure un certain nombre de dimensions temporelles différentes. Elles sont au nombre de trois. Ce sont les types de temps **Valid time**, **Transaction time** et **User-defined time**.

Le Valid time d'un événement est la période durant laquelle l'événement a lieu dans le monde réel. Cette survenance est indépendante de l'enregistrement dans la base de données. Grâce à ce Valid time, l'utilisateur pourra demander quels sont les états passés d'une donnée, de voir quel est son évolution au sein de la base de données. Dans la littérature, le Valid time est parfois repris sous d'autres appellations : Intrinsic time, Effective time ou Logical time. Le Valid time est également le type de temps le plus employé dans les bases de données temporelles développées jusqu'à présent.

Le Transaction time concerne le stockage des données dans la base de données. Il identifie l'instant où la donnée a été enregistrée dans la base de données. Il permet à l'utilisateur de savoir quel était l'état de la base de données dans le passé. D'autres noms désignent également ce Transaction time dans les articles déjà publiés : Extrinsic time, Registration time, Physical time. Dans les travaux actuellement parus, peu d'auteurs ont utilisé ce type de temps dans leur modèle temporel.

Le troisième type de temps, User-defined time, est supporté et fourni par l'utilisateur. Comme son nom l'indique, la sémantique de ce type de temps est donnée par l'utilisateur

d'applications. Il l'introduit dans son modèle quand il le désire et quand il pense que son utilité lui sera bénéfique.

Actuellement, le modèle relationnel supporte seulement le User-defined time car c'est simplement un autre domaine comme INTEGER, etc. . Ce n'est pas le cas en ce qui concerne le Transaction time et le Valid time. Comme nous le verrons, nous devons recourir à toute une série de transformations pour que le modèle relationnel supporte ces deux types de temps. Nous devons observer que chacun des concepts du modèle peut supporter un ou plusieurs types de temps. Dans le cas où un concept du modèle ne supporte que le Valid time, il est appelé *concept historique*. S'il supporte le Transaction time, il est dénommé *concept rollback*. Dans le cas où il supporte les deux types de temps, il est appelé *concept temporel*. Dans beaucoup d'études publiées, il s'avère que seul le Valid time est supporté.

Dans l'exemple qui suit, nous voyons une base de données qui supporte les deux premiers types de temps que nous avons définis.

Employé					
NOM	POSTE	VALID TIME DEBUT	VALID TIME FIN	TRANSACTION TIME DEBUT	TRANSACTION TIME FIN
Durant	Cadre	01-09-1986	31-12-1993	10-09-1986	01-02-1994
Dupont	Sous-directeur	08-07-1983	03-09-1995	08-07-1983	Null

Figure II-1 - BD supportant Valid time et Transaction time

Nous observons dans ce premier tableau que l'employé Durant a occupé le poste de Cadre entre le 01-09-1986 et le 31-12-1993. Mais, cette donnée n'a été enregistrée dans la base de données que le 10 septembre 1986. D'autre part, cette donnée a été supprimée le 01-02-1994.

B. Modélisations du temps

Pour représenter le temps dans une base de données, il existe deux modèles conceptuels du temps. Ces modèles sont le **modèle discret** et le **modèle continu**.

Le premier de ces modèles proposés représente le temps comme un ensemble de points équidistants et discrets. Ces points correspondent à des unités de temps indécomposables que nous appelons **chronons**. Une autre caractéristique de ces chronons est une durée arbitraire. Cette durée est fixée par la granularité de la mesure du temps. Cette granularité peut être des années, des mois, des jours, des heures, etc. ou n'importe quel autre unité de temps appropriée. D'où, le temps dans le modèle discret peut être vu comme étant isomorphe aux nombres entiers. De plus, nous y définissons des intervalles de temps qui sont des ensembles de chronons équidistants et consécutifs. Ces intervalles sont notés par $[t_1, t_v]$ où la borne

inférieure t_1 est le premier élément de l'intervalle et la borne supérieure t_u est le dernier élément de l'intervalle.

Dans le modèle continu, le temps peut être vu comme étant isomorphe aux nombres réels. Chacun de ces nombres correspond à un point du temps. Bien que ce modèle continu représente le temps différemment du modèle discret, il partage avec celui-ci une propriété importante: le temps doit être ordonné totalement c'est-à-dire que pour deux instants t_1 et t_2 , t_1 précède t_2 ou t_2 précède t_1 .

Malgré une perception continue du temps, beaucoup de propositions de bases de données temporelles ont adopté le modèle discret du temps. Les principales raisons sont :

1. Les mesures temporelles sont imprécises. Les instruments de mesure du temps indiquent le moment où l'événement s'est produit en termes de chronons.
2. Les concepts de chronons et d'intervalles de temps permettent d'exprimer les événements qui ont une certaine durée.

C. Choix de la modélisation et des types de temps

Dans ce travail, le modèle relationnel considéré doit supporter à la fois le Valid time et le Transaction time. Nous pensons que ces deux types de temps sont nécessaires pour exprimer le comportement temporel des différents concepts concernés. Ils apportent des informations à plusieurs niveaux qui sont intéressantes pour l'utilisateur. Néanmoins, dans le développement qui suivra, nous supposons que ces deux types de temps sont identiques c'est-à-dire que le moment où la donnée est enregistrée (resp. supprimée) dans la base de données correspond exactement à l'instant du début (resp. de fin) de l'existence de celle-ci. Cette hypothèse est uniquement faite dans un but de simplification.

En ce qui concerne la modélisation du temps, nous pensons que le modèle discret s'avère plus adéquat pour représenter le temps. Comme la plupart des auteurs qui ont développé un modèle temporel, nous choisissons cette modélisation discrète du temps.

III. Vue d'ensemble des algèbres temporelles

Avant d'exposer le modèle temporel considéré dans ce travail, nous présentons les algèbres temporelles suivantes :

1. Algèbre temporelle de Jones
2. Algèbre temporelle de Ben-Zvi
3. Algèbre temporelle de Navathe
4. Algèbre temporelle de Sadeghi
5. Algèbre temporelle de Sarda
6. Algèbre temporelle de Clifford
7. Algèbre temporelle de Tansel
8. Algèbre temporelle de Gadia
9. Algèbre temporelle de Yeung
10. Algèbre temporelle de Lorentzos
11. Algèbre temporelle de Mac-Kenzie
12. Algèbre temporelle de Tuzhilin

Ces douze algèbres temporelles étendent l'algèbre relationnelle de Codd pour qu'elles supportent le Valid time et dans certains cas, le Transaction time. Ces algèbres diffèrent au niveau des types d'objets qu'elles définissent et au niveau des opérations qu'elles offrent.

Nous allons examiner en premier lieu plusieurs algèbres temporelles qui étudient la variation temporelle des tuples. Parmi celles-ci, nous avons les algèbres de Jones, de Ben-Zvi, de Navathe, de Sadeghi et de Clifford. A partir de l'algèbre de Sarda, nous constaterons que les algèbres incorporent différents opérateurs permettant de passer d'une représentation par intervalle à une représentation par chronons et inversement. L'intérêt de la représentation par intervalle est l'occupation efficace de l'espace mémoire disponible et une présentation plus appropriée. La seconde présentation est l'union de tranches de temps pouvant être manipulée par les opérateurs relationnels conventionnels. Dès lors, nous pourrons grâce à ces opérateurs choisir la représentation qui nous convient le mieux. Pour les 7 dernières algèbres présentées, nous verrons qu'elles utilisent un modèle de données qui n'est pas sous première forme normale. La non atomicité des attributs est due à la nature temporelle introduite. De plus, nous constaterons que chaque algèbre temporelle a été développée de telle sorte qu'elle reste le plus proche possible de l'algèbre de Codd, sauf Tuzhilin qui a tout redéfini.

A. Algèbre temporelle de Jones

1. Présentation

Jones¹ a développé un langage appelé LEGOL 2.0 . Ce langage a été conçu pour être utilisé au niveau des applications de bases de données dans lesquelles l'ordre temporel des événements et la validité des objets sont très importants. C'est la première algèbre temporelle qui a été définie. Son algèbre ne tient compte que du Valid time. Les autres types de temps ne sont pas considérés.

Les objets employés par le modèle de données du LEGOL 2.0 sont des relations identiques à celles du modèle relationnel classique auxquelles sont ajoutées deux attributs implicites. Ces attributs, START et STOP, correspondent aux chronons initial et final de l'intervalle d'existence de la relation dans le monde réel. Les valeurs de ces attributs seront initialisées par l'utilisateur lors de chaque introduction de données.

2. Exemple

Un exemple illustrant cette algèbre est donné par la Figure III-1. Il reprend les informations de deux clients d'une société et la localité où ils ont habité. La granularité du temps dans cet exemple est d'un jour.

CLIENT			
NOM	LOCALITE	START	STOP
Durant	Bruxelles	01-03-1980	31-05-1980
Durant	Bruxelles	01-03-1981	31-08-1981
Dupont	Anvers	01-03-1980	01-03-1981

Figure III-1 - Exemple d'algèbre temporelle de Jones

Pour le client dont le nom est Durant, nous pouvons observer qu'il a habité Bruxelles durant la période allant du 01-03-1980 au 31-05-1980 et celle entre le 01-03-1981 et le 31-08-1981.

3. Opérateurs

Cette algèbre de Jones conserve les opérateurs de l'algèbre relationnelle de Codd. Ceux-ci restent inchangés. Des nouveaux opérateurs ont été ajoutés. Ils sont liés à la dimension temporelle introduite dans l'algèbre. Ils font partie de la liste des cinq éléments qui suit.

¹ [MC KENZIE1991]

1. **TIME INTERSECTION**
2. **ONE-SIDED-TIME INTERSECTION**
3. **TIME UNION**
4. **TIME DIFFERENCE**
5. **TIME-SET MEMBERSHIP**

La première de ces opérations renvoie les tuples pour lesquels le Valid time est égal à l'intersection des Valid times de deux tuples qui se chevauchent. Quant aux autres opérateurs, ils ne sont pas encore spécifiés mais ils semblent supporter une forme limitée de sélection temporelle et de jointure temporelle.

B. Algèbre temporelle de Ben-Zvi

1. Présentation

Le modèle temporel de Ben-Zvi¹ comprend deux types d'objets différents : les relations instantanées et les relations temporelles, c'est-à-dire les relations supportant à la fois le Valid time et le Transaction time. C'est un des rares auteurs à introduire les deux types de temps dans son algèbre relationnelle. Ces relations temporelles correspondent à des ensembles de tuples contenant cinq attributs implicites supplémentaires. Ces attributs sont :

- EFFECTIVE-TIME-START
- EFFECTIVE-TIME-STOP
- REGISTRATION-TIME-START
- REGISTRATION-TIME-STOP
- DELETION-TIME

Les deux premiers reprennent la période de validité du tuple. REGISTRATION-TIME-START(resp. STOP) est égal au Transaction time de la transaction qui stocke l'EFFECTIVE-TIME-START(resp. STOP) dans la base de données. Le dernier attribut a pour valeur l'instant où un tuple incorrect introduit dans la base de données a été supprimé logiquement.

2. Exemple

Nous reprenons l'exemple des clients d'une société caractérisés par leur nom et leur localité. La granularité associée est toujours d'un jour.

¹[MC KENZIE1991] et [TANSEL1993]

CLIENT						
NOM	LOCALITE	Effective-TIME-START	Effective-TIME-STOP	Régistration-TIME-START	Régistration-time-stop	Deletion-time
Durant	Bruxelles	01-03-1980	31-05-1980	01-04-1980	31-05-1980	-
Dupont	Anvers	01-08-1980	01-03-1981	01-03-1980	09-03-1981	-

Figure III-2 - Exemple d'algèbre temporelle de Ben-Zvi

3. Opérateurs

Afin de pouvoir supporter la dimension temporelle introduite, les opérateurs d'union, de différence, de jointure, de sélection et de projection ont été étendus. Par exemple, la sélection est étendue par des prédicats tels que overlap, follows, etc. .

Un nouvel opérateur appelé **TIME VIEW** a été créé. Il est noté $TV = (t_e, t_s)$ où t_e et t_s correspondent chacun à un temps. Cet opérateur va tout d'abord sélectionner les tuples pour lesquels t_e appartient à l'intervalle de validité et t_s appartient à la période de Transaction time. Puis, celui-ci élimine les attributs implicites de la relation temporelle pour former une relation conforme à celle du modèle relationnel classique. Dans l'exemple du point deux, l'application de TIME VIEW (08-04-1980, 10-04-1980) donnerait comme solution

NOM	LOCALITE
Durant	Bruxelles

Figure III-3 - Résultat d'application de TIME VIEW

C. Algèbre temporelle de Navathe

1. Présentation

Le modèle relationnel temporel de Navathe¹ (TRM) et son algèbre associée ont été définis de telle sorte qu'ils supportent TSQL (Temporal SQL), une extension de SQL. Son modèle et TSQL offrent un framework consistant et cohérent pour la gestion des données temporelles et non temporelles.

¹ [MC KENZIE1991], [TANSEL1993] et [LORENTZOS1995]

Les objets inclus par Navathe dans son modèle temporel sont au nombre de deux :

1. Les relations instantanées.
2. Les relations historiques c'est-à-dire des relations qui ne supportent que le Valid time.

Dans ces relations historiques, nous trouvons deux attributs implicites obligatoires TIME-START et TIME-STOP. Ils constituent respectivement la borne inférieure et la borne supérieure de l'intervalle d'existence du tuple. Ces relations sont semblables à celles du LEGOL 2.0. Elles diffèrent des relations historiques du LEGOL 2.0 au niveau des tuples. Si les tuples ont une valeur identique pour chaque attribut non temporel et si les intervalles de validité de ces tuples sont contigus, ils doivent être réunis.

2. Exemple

L'exemple vu dans l'algèbre de Jones reste d'actualité. Il suffit de remplacer le nom des attributs START et STOP respectivement par TIME-START et TIME-STOP.

3. Opérateurs

Navathe maintient les cinq opérateurs de base de l'algèbre relationnel identiques. D'autre part, il y inclut cinq nouveaux opérateurs travaillant sur les relations historiques : **TIME-SLICE**, **INNER-TIME-VIEW**, **OUTER-TIME-VIEW**, **TCJOIN** et **TCNJOIN**. Les trois premiers d'entre eux sont des formes de sélection temporelle. Les deux autres réalisent des jointures utilisant la sémantique de l'intersection, c'est-à-dire que le Valid time de chaque tuple résultat est égal à l'intersection des Valid times de deux tuples d'entrée.

D. Algèbre temporelle de Sadeghi

1. Présentation

L'algèbre proposée par Sadeghi¹ est comparable à celle de Navathe. Elle a été conçue pour supporter HQL (Historical Query Language). Il n'y a qu'un seul type d'objet présent dans cette algèbre : la relation historique. Cette relation historique est obtenue en prenant la relation du modèle relationnel et en lui adjoignant deux attributs implicites START et STOP. Ces deux attributs constituent les bornes de l'intervalle de validité de chaque tuple. De plus, la structure de ces relations historiques sont similaires à celles vues précédemment dans l'algèbre de Jones. Néanmoins, si nous avons des tuples de valeurs identiques pour les attributs non temporels avec des intervalles de validité contigus, ils sont regroupés en un seul. Par exemple, si dans une relation, nous trouvons

¹ [MC KENZIE1991]

CLIENT			
NOM	LOCALITE	START	STOP
Durant	Bruxelles	01-04-1993	02-03-1995
Durant	Bruxelles	02-03-1995	08-04-1997

Figure III-4 - Illustration de fusion d'intervalles (1)

nous devons les rassembler en un seul tuple, ce qui donne

CLIENT			
NOM	LOCALITE	START	STOP
Durant	Bruxelles	01-04-1993	08-04-1997

Figure III-5 - Illustration de fusion d'intervalles (2)

2. Exemple

Une illustration de cette algèbre temporelle est donnée par la Figure III-1 vu dans l'algèbre de Jones.

3. Opérateurs

Les opérateurs du modèle relationnel classique sont étendus dans le but de supporter l'aspect temporel qui a été ajouté. D'autre part, Sadeghi a introduit deux nouveaux opérateurs : **WHEN** et **TIME-JOIN**. L'opérateur **WHEN** nous offre la possibilité de réaliser une sélection temporelle. Les résultats renvoyés par **WHEN** sont des tuples où n'apparaissent pas les attributs implicites. L'opérateur **TIME-JOIN** est comme son nom l'indique une jointure temporelle.

E. Algèbre temporelle de Sarda

1. Présentation

L'algèbre de Sarda¹ comprend les deux types d'objets que sont les relations instantanées du modèle relationnel et les relations historiques. Dans ces dernières, il représente le Valid time par **PERIOD**, un unique attribut implicite non atomique. Les intervalles qu'ils considèrent sont des intervalles semi-ouverts à droite.

¹ [MC KENZIE1991] et [TANSEL1993]

2. Exemple

Reprenons toujours l'exemple des clients d'une société. Une illustration du modèle de Sarda est présentée par la Figure III-6.

CLIENT		
NOM	LOCALITE	PERIOD
Durant	Bruxelles	01-03-1980 ... 31-05-1980
Durant	Bruxelles	01-03-1981 ... 31-08-1981
Dupont	Anvers	01-03-1980 ... 01-03-1981

Figure III-6 - Exemple d'algèbre temporelle de Sarda

3. Opérateurs

Les opérateurs classiques sont considérés tel quel dans l'algèbre temporelle de Sarda à l'exception de la sélection. Celle-ci est étendue de telle sorte qu'elle puisse permettre les prédicats temporels et non temporels. En plus des opérateurs relationnels classiques, Sarda en a introduit de nouveaux :

1. **EXPAND** : cet opérateur produit, pour tout chronon de l'intervalle de validité de chaque tuple, un tuple de valeur identique avec le chronon comme time-stamp.
2. **CONTRACT** : Il réalise l'opération inverse de EXPAND. Il réunit les tuples de valeur identique en un seul.
3. **PROJECT-AND-WIDEN** : C'est une forme de projection temporelle qui réunit les tuples ayant des valeurs identiques pour les attributs non temporels.
4. **CONCURRENT-PRODUCT** : Il s'agit du produit cartésien utilisant la sémantique d'intersection.

F. Algèbre temporelle de Clifford

1. Présentation

Dans son modèle de données HDRM (Historical Relational Data Model), Clifford¹ permet deux types d'objet : le LIFESPAN qui est un ensemble de chronons et

¹ [MC KENZIE1991], [TANSEL1993] et [LORENTZOS1995]

la relation historique dans laquelle un LIFESPAN est assigné à chaque tuple et à chaque attribut. Une relation dans son modèle est un 4-tuple ordonné qui contient un ensemble d'attributs, un ensemble d'identifiants, une fonction qui associe chaque attribut à leur LIFESPAN et une fonction qui associe les attributs à leurs domaines de valeurs.

2. Exemple

Reprenons une nouvelle fois l'exemple des clients d'une société. Supposons que la granularité soit dans cet exemple d'un trimestre. Représentons d'autre part ces trimestres par des entiers, où 1 représente le premier trimestre de 1990, 2 le second trimestre de 1990, etc. . La figure suivante constitue un exemple de l'algèbre de Clifford.

CLIENT		
NOM	LOCALITE	LIFESPAN
1 → "Dupont "	1 → "Bruxelles "	{1,2,4}
2 → "Dupont "	2 → "Bruxelles "	
4 → "Dupont "	4 → "Bruxelles "	
2 → "Laloi "	2 → "Anvers "	{2,3,5,6}
3 → "Laloi "	3 → "Anvers "	
5 → "Laloi "	5 → "Tournai "	
6 → "Laloi "	6 → "Tournai "	

Figure III-7 - Exemple d'algèbre temporelle de Clifford

3. Opérateurs

Les opérateurs d'union, de différence, d'intersection, de projection et de produit cartésien sont étendus afin qu'ils puissent supporter directement la dimension temporelle introduite.

De plus, Clifford a envisagé de nouveaux opérateurs :

1. **WHEN** : Il associe l'instance d'une relation à son LIFESPAN qui est l'union des LIFESPANS de ces tuples.
2. **SELECT-IF** : Il s'agit d'une forme de sélection temporelle. Cet opérateur sélectionne les tuples valides qui satisfont une certaine condition à un moment donné.
3. **TIME-SLICE** : C'est une forme de projection temporelle. Il renvoie les tuples pour lesquels les LIFESPANS sont restreints à une partie de leur LIFESPAN initial.

4. **SELECT-WHEN** : Cet opérateur est une variante du SELECT-IF. Il renvoie les tuples avec des LIFESPANS restreints à l'instant précis où il vérifie une condition de sélection donnée.
5. **TIME-JOIN** : Il restreint les LIFESPANS des tuples à la valeur d'un attribut temporel.

G. Algèbre temporelle de Tansel

1. Présentation

Tansel¹ ne considère que la relation historique comme type d'objet dans son algèbre. Cependant, ces relations peuvent être caractérisées par quatre types différents d'attributs. Ces attributs sont temporels ou non temporels ; ils contiennent une seule valeur ou un ensemble de valeurs. En ce qui concerne les attributs temporels contenant une seule valeur, ils sont constitués de trois composants : les deux premiers reprennent les bornes de l'intervalle d'existence tandis que le troisième contient la valeur de domaine associé à l'attribut. Si nous disposons d'un attribut temporel contenant un ensemble de valeurs, il est composé d'un ensemble des trois composants cités juste avant.

2. Exemple

Un exemple illustrant l'algèbre de Tansel est donné par la Figure III-8.

CLIENT	
NOM	LOCALITE
Durant	{ ([01-03-1980,31-05-1980), "Bruxelles"), ([01-03-1981,31-08-1981), "Bruxelles") }
Dupont	{ ([01-03-1980,01-03-1981), "Anvers") }

Figure III-8 - Exemple d'algèbre temporelle de Tansel

3. Opérateurs

Les opérateurs relationnels standards sont étendus afin qu'ils tiennent compte du temps introduit au niveau des données.

D'autre part, Tansel propose dans son algèbre quelques nouveaux opérateurs. Tout d'abord, il y a **PACK** qui combine les tuples qui ont des attributs de valeurs identiques sauf pour un seul de ces attributs. **UNPACK** réalise l'opération inverse de **PACK**. Il copie un tuple pour chaque élément d'un attribut qui en contient un ensemble.

¹ [MC KENZIE1991], [TANSEL1993] et [LORENTZOS1995]

T-DEC décompose un attribut temporel contenant une seule valeur en trois attributs atomiques non temporels qui représentent les trois composants de cet attribut atomique temporel. **T-FORM** est l'opération inverse de T-DEC. **DROP-TIME** permet, comme son nom l'indique, d'enlever la composante temporelle de l'attribut. Les trois autres nouveaux opérateurs, **SLICE**, **USLICE** et **DSLICE** sont des formes limitées de la projection temporelle. Ils déterminent le time-stamp de l'attribut temporel comme l'intersection, l'union et la différence de son time-stamp original et du time-stamp d'un autre attribut. Dans le cas où le time-stamp est vide, alors le tuple est supprimé. Enfin, il a introduit un opérateur appelé **ENUMERATION** pour supporter les opérateurs d'agrégation.

H. Algèbre temporelle de Gadia

1. Présentation

Nous trouvons dans l'algèbre de Gadia¹ deux types d'objet distincts : les relations historiques et les éléments temporels. Il définit un élément temporel comme la réunion finie d'intervalles disjoints. Pour pouvoir observer les changements de valeur des attributs, il introduit la notion d'assignation temporelle. L'assignation temporelle d'un attribut A est une fonction des éléments temporels vers le domaine de l'attribut A. Son modèle exige que chaque valeur des attributs d'un tuple donné soit fonction du même élément temporel. Cette propriété, appelée homogénéité, assure qu'un instantané au temps t d'une relation historique produit une relation sans valeur NULL.

2. Exemple

Un exemple représentant cette algèbre est donné par la Figure III-9.

CLIENT	
NOM	LOCALITE
[01-03-1980,31-05-1980) U [01-03-1981,31-08-1981) → Durant	[01-03-1980,31-05-1980) U [01-03-1981,31-08-1981) → Bruxelles
[01-03-1980,01-03-1981) U [10-04-1981,11-11-1981) → Dupont	[01-03-1980,01-03-1981) → Anvers [10-04-1981,11-11-1981) → Mons

Figure III-9 - Exemple d'algèbre temporelle de Gadia

¹ [MC KENZIE1991], [TANSEL1993] et [LORENTZOS1995]

3. Opérateurs

Gadia a défini une version historique des cinq opérateurs standards. Ceux-ci utilisent les sémantiques instantanées. Donc, pour tout opérateur historique, prendre l'instantané au temps t de la relation historique résultat doit être équivalent à appliquer la partie relationnelle de l'opérateur historique à l'instantané au temps t des relations historiques.

De plus, Gadia a défini deux nouveaux opérateurs :

1. **TDOM** : Il associe un tuple ou instance de relation à son domaine temporel. Le domaine temporel correspond à son élément temporel et le domaine temporel d'une relation est l'union des éléments temporels de ces tuples.
2. **TEMPORAL SELECTION** : C'est une forme de sélection temporelle et de projection temporelle. Cet opérateur sélectionne dans une relation, les tuples pour lesquels les éléments temporels chevauchent un élément temporel spécifié. Puis, il restreint les valeurs des attributs des tuples trouvés à l'intersection de leurs éléments temporels avec l'élément temporel spécifié.

I. Algèbre temporelle de Yeung

1. Présentation

Le modèle proposé par Yeung¹ est semblable à celui de Gadia. En effet, il considère les mêmes types d'objet à savoir la relation historique et l'élément temporel. Mais, contrairement à Gadia, il permet que les éléments temporels soient multidimensionnels. Cela lui permet de modéliser différents aspects du temps. Dans son modèle, les attributs sont toujours des fonctions d'éléments temporels dans les domaines de valeur des attributs. Néanmoins, il y a une différence par rapport au modèle de Gadia ; les valeurs d'attribut ne sont pas nécessairement des fonctions du même élément temporel. Enfin, Yeung a envisagé les relations avec des identifiants possibles. Ces identifiants ne peuvent contenir qu'une seule valeur sur l'intervalle de validité.

2. Exemple

La Figure III-9 dans l'algèbre de Gadia illustre bien cette algèbre de Yeung.

¹ [MC KENZIE1991]

3. Opérateurs

Les opérateurs standards sont étendus car ils doivent supporter la dimension temporelle introduite. Yeung a aussi pensé à deux nouveaux opérateurs : **SELECTION TEMPORELLE** et **PROJECTION TEMPORELLE**.

J. Algèbre temporelle de Lorentzos

1. Présentation

Comme pour les deux algèbres précédentes, Lorentzos¹ associe aussi les time-stamps aux valeurs d'attribut plutôt qu'aux tuples. D'autre part, ces time-stamps peuvent être reliés à un sous-ensemble non vide de valeurs d'attribut d'un tuple.

De plus, les time-stamps des attributs sont explicites c'est-à-dire qu'ils sont numériques et l'utilisateur peut les voir et les modifier comme il le désire. De manière similaire à l'algèbre de Sarda, les intervalles de validité sont des intervalles semi-ouverts à droite.

2. Exemple

Un exemple illustrant l'algèbre temporelle de Lorentzos est donné par la figure qui suit.

CLIENT					
NOM	LOCALITE	Année-Start	Année-Stop	Mois-Start	Mois-Stop
Durant	Bruxelles	1980	1981	3	7
Durant	Bruxelles	1981	1982	9	12
Dupont	Anvers	1981	1982	3	12
Dupont	Mons	1984	1986	5	9

Figure III-10 - Exemple d'algèbre temporelle de Lorentzos

3. Opérateurs

Les opérateurs standards sont inchangés dans l'algèbre proposée par Lorentzos. Mais, il y a de nouveaux opérateurs pour la manipulation de la dimension temporelle. Ces opérateurs sont **EXTEND**, **UNFOLD** et **FOLD**. L'opérateur UNFOLD transforme une relation, où les tuples possèdent deux attributs time-stamps, en une autre relation équivalente où les tuples n'en contiennent qu'un seul. L'attribut time-stamp du résultat

¹ [MC KENZIE1991] et [TANSEL1993]

représente un chronon durant lequel l'attribut est valide. Quant à l'opérateur FOLD, il effectue l'opération inverse de UNFOLD.

Grâce à ces deux opérateurs, nous disposons de deux représentations du time-stamp. En effet, UNFOLD nous donne une représentation orientée chronon tandis qu'avec FOLD, nous avons une représentation orientée intervalle.

K. Algèbre temporelle de Mac-Kenzie

1. Présentation

Mac-Kenzie¹ impose comme hypothèse aux attributs time-stamps de ne contenir qu'une seule valeur. Il définit une relation rollback comme une séquence de relations instantanées indexées par le Transaction time. Il dit aussi qu'une relation temporelle est une suite de relations historiques indexées par le Transaction time.

Il restreint la valeur d'un attribut d'une relation historique à une paire ordonnée composée d'une valeur et d'un ensemble de chronons.

De plus, il n'y a pas de contrainte d'homogénéité sur les time-stamps des valeurs d'attributs. Il n'y a pas non plus deux tuples d'une relation possédant des valeurs identiques pour tous leurs attributs.

2. Exemple

Reprenons l'exemple des clients d'une société. Comme pour l'algèbre temporelle de Clifford, la granularité du temps est d'un trimestre et chaque chronon est représenté par un nombre entier : 1 représente le premier trimestre de 1990, 2 le deuxième trimestre, etc. .

CLIENT	
NOM	LOCALITE
<Durant, {1,3,4}>	<Bruxelles, {1,3,4}>
<Dupont, {1}>	<Anvers, {1}>
<Dupont, {2}>	<Mons, {2}>

Figure III-11 - Exemple d'algèbre temporelle de Mc Kenzie

3. Opérateurs

Les opérations standards sont étendues pour supporter directement la dimension temporelle.

¹ [MC KENZIE1991] et [LORENTZOS1995]

Mac-Kenzie a ajouté quelques autres opérateurs à l'ensemble des opérateurs standards. Il y a tout d'abord la dérivation historique δ pour la sélection temporelle et la projection temporelle. Les deux opérateurs suivants, ρ et ρ' , sont les rollbacks instantanés et historiques. Ils enlèvent respectivement l'état instantané ou historique d'une relation rollback ou temporelle.

Finalement, l'utilisateur dispose aussi de deux opérateurs pour réaliser les opérations d'aggrégation unique et non unique et de deux autres pour une conversion entre une relation instantanée et une relation historique.

L. Algèbres temporelles de Tuzhilin

1. Présentation

Tuzhilin¹ propose deux algèbres TA et TA' qui s'appliquent à des objets relationnels historiques qui supportent le Valid time (discret et linéaire-borné) et un opérateur de tranche de temps qui produit des relations sous première forme normale (1°FN) sans valeur NULL.

2. Opérateurs

Les opérateurs sont étendus de telle sorte qu'ils supportent le temps.

Dans l'algèbre TA, il a introduit deux nouveaux opérateurs L_F (futur linéar recursive operator) et L_P (past linéar recursive operator). Ils déterminent les tuples valides en $t+1$ et $t-1$ à partir de trois tuples valides en t .

Dans TA', il propose six nouveaux opérateurs :

1. **UNIONS SEQUENTIELLES** SU_F et SU_P qui « compressent » l'histoire future et passée en un temps simple.
2. **INTERSECTIONS SEQUENTIELLES** SI_F et SI_P qui donnent les tuples constants dans le futur ou le passé jusqu'à maintenant.
3. **SHIFTS** SH_F et SH_P qui « shift » en avant et en arrière d'une unité de temps.

¹ [MC KENZIE1991]

IV. Les modèles

Dans ce chapitre, nous allons définir les différents modèles que nous utiliserons par la suite. Nous définissons tout d'abord un modèle de base qui contiendra une partie des concepts du modèle Entité-Association. Nous sélectionnons seulement une partie de ce modèle Entité-Association pour des raisons de simplification. Nous en profiterons pour rappeler la définition de ces différents concepts. Ensuite, nous étendons ce modèle de base en introduisant des concepts temporels pour obtenir le modèle temporel. Finalement, nous évoquerons le modèle relationnel utilisé dans ce travail.

A. Modèle conceptuel de base

1. Définition du modèle

Le modèle conceptuel de base correspond à une restriction du modèle Entité-Association. En d'autres termes, nous ne conservons que les concepts suivants :

- Les types d'entité
- Les types d'association binaire
- Les attributs atomiques
- Les attributs monovalués
- Les identifiants d'un type d'entité

Nous noterons ce modèle par ER.

2. Définition des concepts

a) Types d'entité

Un type d'entité est la classe de toutes les entités possibles du réel perçu qui vérifie la définition constitutive du type.

Dans l'exemple suivant, nous retrouvons deux types d'entité : CLIENT et COMMANDE.

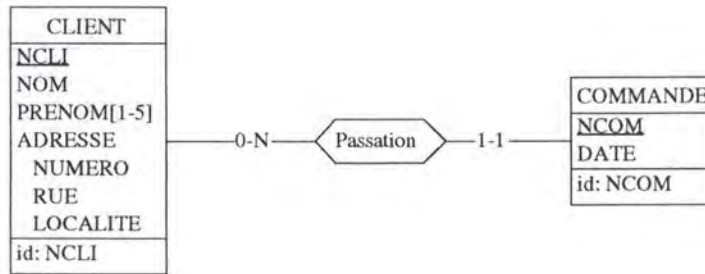


Figure IV-1 - Exemple de types d'entité

Nous supposons que chaque type d'entité possède au moins un attribut. Nous imposons également que chaque type d'entité est identifié par au moins un des attributs qui le caractérisent.

b) Type d'association binaire

Un type d'association binaire est la classe de toutes les associations binaires possibles du réel perçu qui vérifie la définition constitutive du type.

Une association binaire est une association définie par une correspondance entre deux entités où chacune assume un rôle donné. La typologie fonctionnelle de ce type d'association sera one-to-many (1-N) c'est-à-dire que le couple de cardinalités de l'association sera $([i_1 - j_1], [i_2 - 1])$ où $j_1 > 1$ (cfr. Figure IV-2).

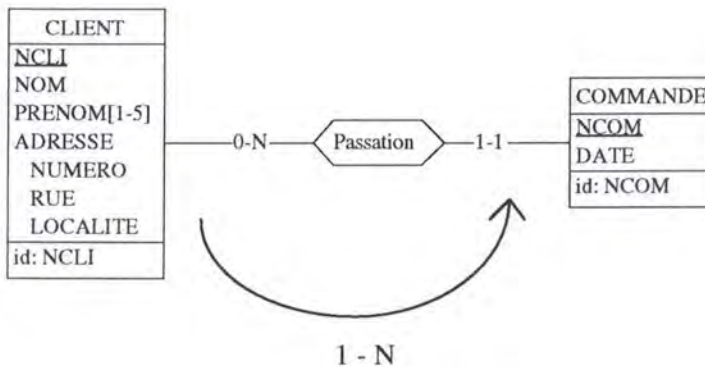


Figure IV-2 - Exemple de type d'association binaire 1-N

Nous supposons que les types d'association considérés dans ce travail ne contiennent aucun attribut. De plus, ces types d'association seront considérés comme des attributs du type d'entité jouant le rôle 1-1.

c) **Attribut atomique**

Un attribut atomique est une caractéristique ou qualité d'un entité qui ne peut être décomposée. Nous imposons que tous les attributs de chaque type d'entité soient atomiques.

Dans la Figure IV-1, les attributs NCLI, NOM et PRENOM du type d'entité CLIENT sont des exemples d'attributs atomiques.

d) **Attribut monovalué**

Un attribut monovalué est une caractéristique ou qualité d'un type d'entité qui prend au maximum une valeur. Nous imposons que tous les attributs de chaque type d'entité soient monovalués.

Les attributs NCLI et NOM du type d'entité CLIENT de la Figure IV-1 sont des illustrations de ce concept.

e) **Identifiant d'un type d'entité**

Un identifiant d'un type d'entité est un groupe minimal d'attributs et/ou de rôles tel qu'à chaque combinaison de valeurs prises par ce groupe correspond au plus une entité de ce type.

Dans la Figure IV-2, NCLI est l'identifiant du type d'entité CLIENT et NCOM est celui du type d'entité COMMANDE.

B. Modèle conceptuel temporel

1. Comportements temporels

Le comportement temporel d'une donnée peut être de deux types différents :

1. sur l'existence de cette donnée.
2. sur les modifications des valeurs de cette donnée.

Une donnée caractérisée par un comportement temporel du premier type signifie que nous désirons connaître la période pendant laquelle elle a été "active" dans la base de données. En d'autres termes, ce type de comportement ne donne pas directement la durée de vie de la donnée. Il nous indique la période pendant laquelle elle présente un intérêt. Par exemple, un employé d'une entreprise existe en tant que personne dans le monde réel. Seulement, il ne devient intéressant que lorsque l'entreprise l'engage. A partir de ce moment, elle désire enregistrer des données au sujet de l'employé. Mais, dès que l'employé quitte l'entreprise, il ne présente alors plus aucun intérêt pour celle-ci. L'entreprise va conserver les données sur cet employé tant qu'elle le désire. Ces données

concernent uniquement la période durant laquelle il apportait quelque chose à l'entreprise.

D'autre part, lorsqu'une donnée est caractérisé par le comportement temporel sur les modifications de ces valeurs, cela signifie que nous voulons connaître la succession des versions de la donnée c'est-à-dire l'évolution de l'état de la donnée au cours du temps.

La relation entre ces deux types de comportement temporel est la suivante :

Le comportement temporel sur les modifications des valeurs d'une donnée est impliquée par le comportement temporel sur l'existence de cette même donnée.

En effet, tant que la donnée n'existe pas, il est impossible d'étudier l'évolution de l'état de celle-ci puisqu'elle n'est pas présente dans la base de données.

Remarque : Le comportement temporel sur l'existence des données sera très peu évoqué dans ce travail. Nous nous intéresserons principalement au comportement du second type.

2. Définition du modèle conceptuel temporel

Le modèle temporel envisagé contient tous les concepts présentés dans le modèle conceptuel de base. De plus, les concepts temporels suivants seront ajoutés à ce modèle :

- Les types d'entité temporels
- Les attributs atomiques monovalués temporels
- Les types d'association binaires temporels

Nous noterons ce modèle temporel par TER.

3. Définition des concepts temporels

a) Types d'entité temporels

Un type d'entité temporel est un type d'entité pour lequel le comportement temporel de ces instances nous intéresse. Ce comportement peut porter soit sur l'existence des instances de ce type d'entité ou sur les modifications des valeurs des instances de ce type d'entité.

Pour distinguer les types d'entité temporels sur l'existence de ceux qui ne le sont pas, nous plaçons à la fin de leur nom le symbole "t", c'est-à-dire que nous les notons comme dans la Figure IV-3.

E''
Δ A1
A2
id: A1

Figure IV-3 - Exemple de type d'entité temporel sur l'existence

Dans le cas où nous souhaitons voir quelle est l'évolution des valeurs de chaque instance du type d'entité, nous dissocions ces types d'entité temporels des autres en plaçant à la fin de leur nom le symbole *, c'est-à-dire que nous les notons comme dans la Figure IV-4.

E*
Δ A1
A2
id: A1

Figure IV-4 - Exemple de type d'entité temporel sur les modifications

b) Attributs atomiques monovalués temporels

Un attribut atomique monovalué temporel est un attribut atomique monovalué pour lequel nous voulons connaître la succession des états au cours du temps.

Pour différencier ces attributs temporels des autres, nous plaçons le symbole * à la fin de leur nom. Dans la Figure IV-5, l'attribut A3 est une illustration de ce concept.

E
Δ A1
A2
A3*
id: A1

Figure IV-5 - Exemple d'attribut temporel

c) Types d'association binaires temporels

Un type d'association binaire temporel est un type d'association binaire pour lequel nous voulons connaître l'évolution de l'état de chaque instance.

La typologie fonctionnelle de ces types d'association temporels sera one-to-many (1-N).

Pour distinguer ce type d'association des autres, nous mettons à la fin du nom de ceux-ci le symbole * (cfr. Figure IV-6).

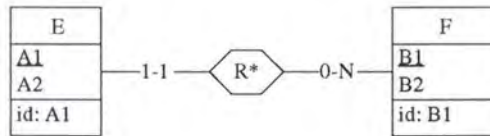


Figure IV-6 - Exemple de type d'association temporel

Nous supposons que ces types d'association ne sont caractérisés par aucun attribut. Ils seront aussi considérés comme des attributs du type d'entité jouant le rôle 1-1.

C. Modèle relationnel

Le modèle relationnel est composé des concepts appartenant à la liste suivante :

1. Les types d'entité
2. Les attributs atomiques monovalués
3. Les identifiants
4. Les attributs de référence

Au niveau physique, ces éléments sont interprétés respectivement par des tables, des colonnes, des primary keys et des foreign keys.

Une foreign key est constituée de l'identifiant d'une autre table et joue le rôle de référence à une ligne de cette autre table (cfr. Figure IV-7).

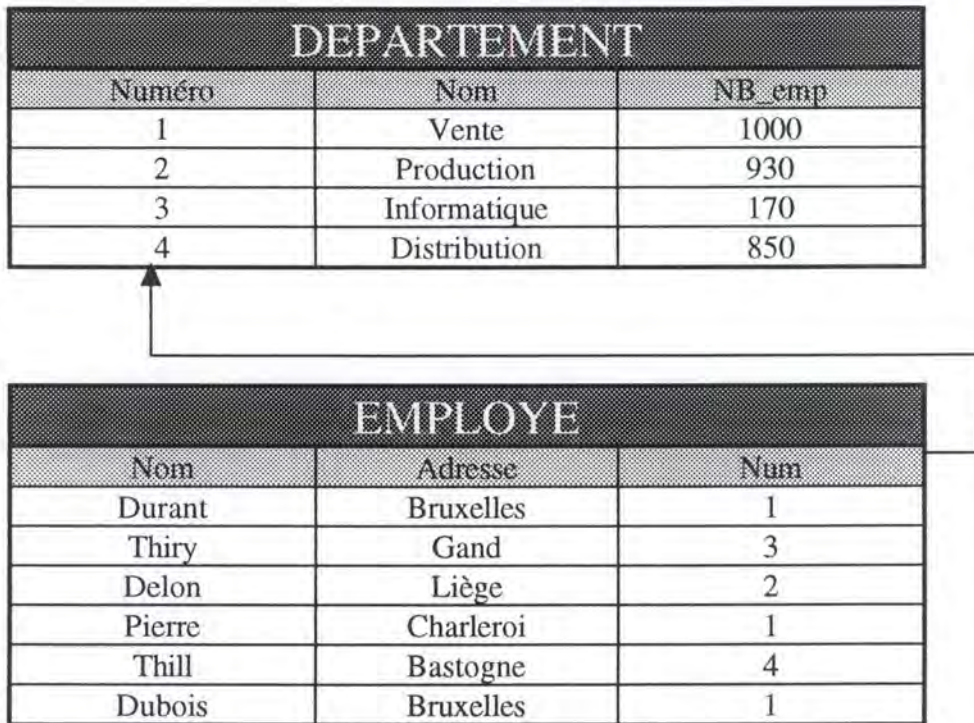


Figure IV-7 - Exemple de foreign key

Nous introduisons un nouveau concept appelé **clé étrangère temporelle**. Il s'agit d'une clé étrangère entre tables temporelles. Elle permet à une version d'une donnée de référencer la version d'une donnée d'une autre table temporelle. Cette foreign key est particulière car elle possède une valeur qui n'est pas identique à celle de l'identifiant de la version de la donnée référencée. Elle a une valeur comprise dans l'intervalle de validité de cette version (cfr. Figure IV-8). Ainsi, nous ne sommes pas obligés de mémoriser le moment exact où chaque version d'une donnée est devenue valide.

Finalement, nous complétons le modèle relationnel avec le concept *d'attribut de référence temporel* qui sera représenté au niveau physique par la foreign key temporelle. Nous la noterons comme dans la figure qui suit.

Nous devons constater dans la Figure IV-8 que la valeur NULL représente l'infini temporel.

DEPARTEMENT				
Numéro	Nom	NB_emp	Periode-debut	Periode-fin
1	Vente	800	1975	1982
1	Vente	1000	1982	NULL
2	Production	710	1973	1981
2	Production	830	1981	1990
2	Production	930	1990	NULL
3	Informatique	131	1989	1992
3	Informatique	170	1992	NULL
4	Distribution	470	1975	1980
4	Distribution	680	1980	1992
4	Distribution	850	1992	NULL

T

EMPLOYE					
Nom	Adresse	Periode-debut	Periode-fin	Num	Debut
Durant	Anvers	1980	1992	1	1981
Durant	Bruxelles	1992	NULL	1	1993
Thiry	Arlon	1973	NULL	3	1990
Delon	Liège	1970	1990	2	1978
Delon	Charleroi	1990	NULL	2	1993
Pierre	Liège	1980	1987	4	1983
Pierre	Visé	1987	NULL	4	1988
Thill	Bruxelles	1980	1983	1	1980
Thill	Gand	1983	1987	1	1984
Thill	Bruges	1987	NULL	1	1985
Dubois	Mouscron	1972	1996	1	1977
Dubois	Bruxelles	1996	NULL	1	1995

Figure IV-8 - Exemple de foreign key temporelle

Nous noterons dans la suite de ce travail ce modèle relationnel par RM.

V. Transformations des structures de données temporelles

Les schémas du modèle conceptuel temporel ne sont pas traduisibles en schémas conformes au modèle relationnel. Les structures de données temporelles ne possèdent pas d'équivalent dans le modèle relationnel. D'où, il faut les transformer en structures non temporelles équivalentes avec des concepts du modèle conceptuel de base.

Dans un premier temps, nous proposons deux transformations qui transforment un schéma temporel en un autre schéma temporel. La première d'entre elles a pour but de généraliser l'aspect temporel au type d'entité lorsque tous ses attributs sont temporels. La seconde sépare les attributs temporels d'un type d'entité de ceux qui ne le sont pas.

Dans la seconde partie, nous évoquerons des transformations qui expriment la dimension temporelle avec les concepts du modèle conceptuel de base. Nous verrons les transformations qui détemporalisent les types d'entité temporels, les attributs temporels et les types d'association temporels.

Enfin, nous parlerons des transformations qui nous permettent de transformer le schéma conceptuel conforme au modèle conceptuel de base en schéma relationnel.

A. Transformations TER \rightarrow TER

1. Transformation TEMP-TE

L'objectif d'une telle transformation est l'optimisation des schémas du modèle conceptuel temporel. Elle permet de diminuer la complexité de ces schémas.

TEMP-TE est une transformation qui porte sur les types d'entité comprenant uniquement des attributs temporels. Elle généralise l'aspect temporel au type d'entité lorsque nous nous intéressons à l'évolution des valeurs de chaque attribut de ce type d'entité.

Considérons un type d'entité E constitué d'au moins deux attributs. Soient par exemple A1, A2 et A3 les attributs de E, et envisageons que A1 est l'identifiant de ce type d'entité. Supposons également que les trois attributs sont temporels. L'application de la transformation à ce cas nous donne l'équivalence de la Figure V-1.



Figure V-1 - Transformation TEMP-TE

2. Transformation ECLAT-TE

Dans un schéma du modèle conceptuel temporel, les chances de rencontrer un type d'entité caractérisé à la fois par des attributs temporels et non temporels sont assez élevées. Il serait donc intéressant d'avoir une transformation qui sépare tous les éléments temporels présents dans le type d'entité de ceux qui ne le sont pas, puis de les regrouper au sein d'une même structure de données. Nous allons décrire une telle transformation. Ensuite, nous évoquerons les cas particuliers relatifs à cette transformation.

a) Description de la transformation

Pour pouvoir exposer cette transformation, prenons un type d'entité E contenant quatre attributs A1, A2, A3 et A4 . Faisons également le choix que A1 soit l'identifiant de E et que A3 et A4 soient temporels.

La transformation ECLAT-TE (cfr. Figure V-2) consiste à diviser le type d'entité E en deux. Le type d'entité E comprendra uniquement les attributs non temporels. Le second type d'entité, E1, sera caractérisé par les attributs temporels de E. Le type d'association entre ces deux types d'entité sera un type d'association one-to-one vu que tous les attributs temporels sont monovalués. Il est aussi temporel, car à une instance d'un type d'entité E, il doit faire correspondre l'historique des attributs qui caractérisaient cette instance. Nous observons immédiatement que, pour pouvoir appliquer cette transformation, le type d'entité doit être caractérisé par au moins deux attributs et, dans ces attributs, il y en a au moins un qui est un attribut non temporel et au moins un qui est temporel.

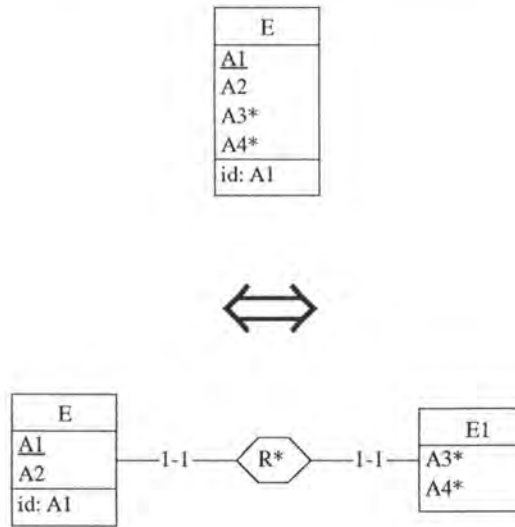


Figure V-2 - Transformation ECLAT-TE

L'avantage de cette transformation est donc de regrouper dans une table unique tous les attributs du type d'entité dont nous désirons connaître l'évolution de l'état au cours du temps. L'inconvénient présenté par cette transformation est l'augmentation du nombre de types d'entité dans le schéma conceptuel. Ceci entraîne parfois une moins bonne lisibilité du schéma et une complexité croissante de celui-ci.

Un exemple illustrant cette transformation est donné par l'équivalence suivante :

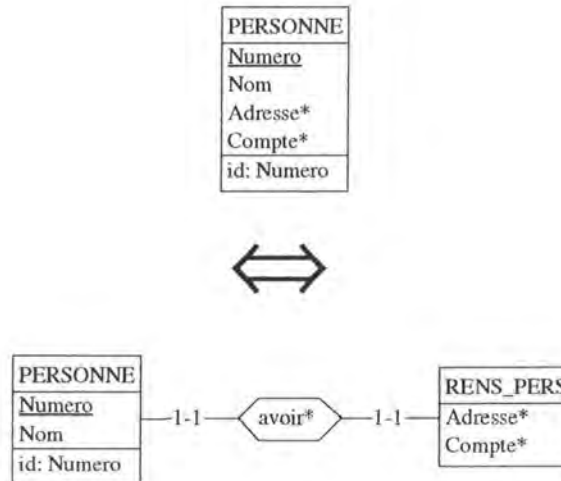


Figure V-3 - Exemple illustrant ECLAT-TE

b) Cas particuliers

1. Un premier cas particulier est donné par la Figure V-4. Cette équivalence est facilement obtenue en composant la transformation décrite précédemment avec la transformation TEMP-TE. En effet, ECLAT-TE va séparer les attributs temporels et non temporels du type d'entité et regrouper tous les attributs temporels dans un nouveau type d'entité. Puis, nous pouvons appliquer TEMP-TE à ce nouveau type d'entité créé.

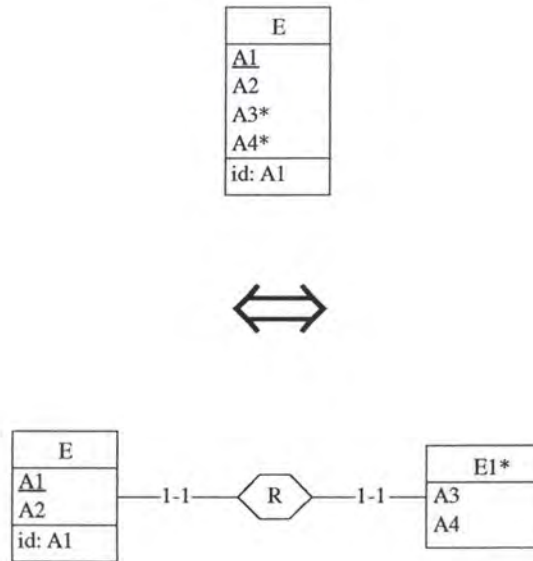


Figure V-4 - Cas particulier 1

Si nous appliquons ce cas particulier au type d'entité PERSONNE, nous obtenons l'équivalence de la Figure V-5.

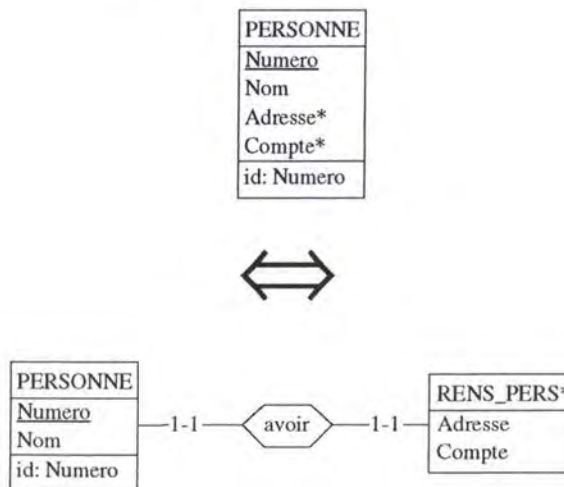


Figure V-5 - Exemple du cas particulier 1

2. Un deuxième cas particulier de la transformation ECLAT-TE est d'isoler dans des tables différentes l'évolution de l'état de chaque attribut temporel.

Ce cas particulier est représenté par la Figure V-6.

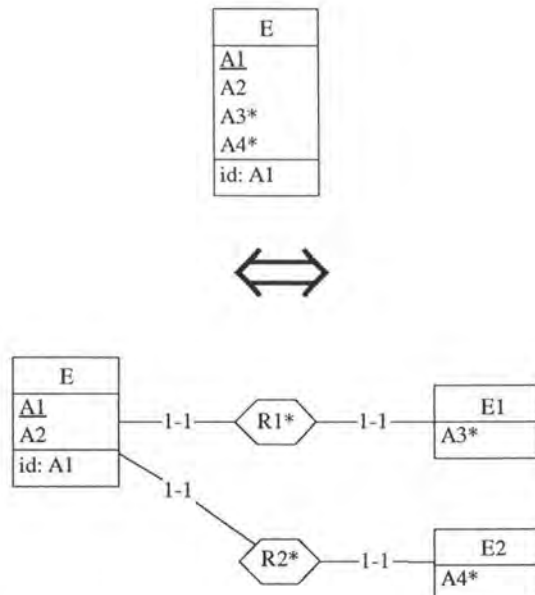


Figure V-6 - Cas particulier 2

L'avantage de ce cas particulier est qu'il suffira d'accéder au bon type d'entité pour connaître l'évolution complète de l'état d'un attribut.

L'inconvénient est que cela nous coûtera parfois plus de temps pour accéder à l'information de certains attributs.

D'autre part, par rapport au premier cas particulier, nous voyons que le nombre de tables sera encore plus élevé. Dès lors, dans des schémas temporels très complexes, nous risquons de les rendre encore moins lisibles et aussi plus complexes.

Si nous appliquons ce cas particulier au type d'entité PERSONNE, nous obtenons l'équivalence de la Figure V-7.

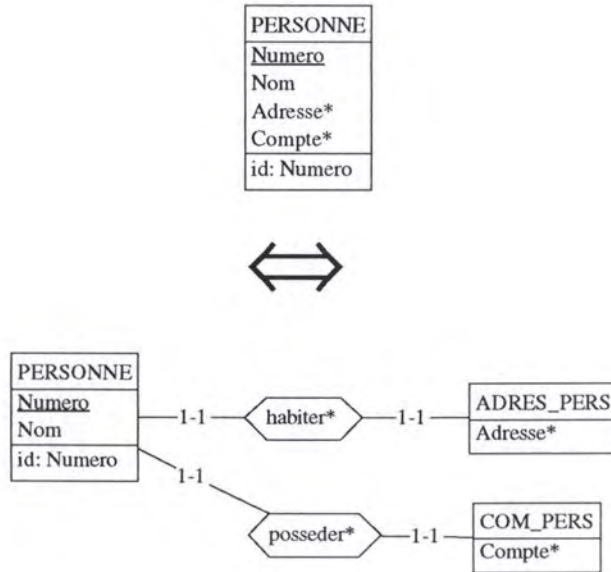


Figure V-7 - Exemple du cas particulier 2

3. Nous pouvons rencontrer le cas particulier où nous avons :

- un type d'entité F caractérisé par les attributs B1, B2 et B3 avec B1 comme identifiant
- un type d'entité E caractérisé par deux attributs non temporels, A1 et A2, et deux attributs temporels A3* et A4*. L'attribut identifiant de ce type d'entité est A1.
- Un type d'association non temporel T et un type d'association temporel R* dans lesquels E joue le rôle 1-1 et F le rôle 0-N.

Vu que les types d'association sont considérés comme des attributs du type d'entité jouant le rôle 1-1, nous pouvons envisager le cas particulier de la Figure V-8.

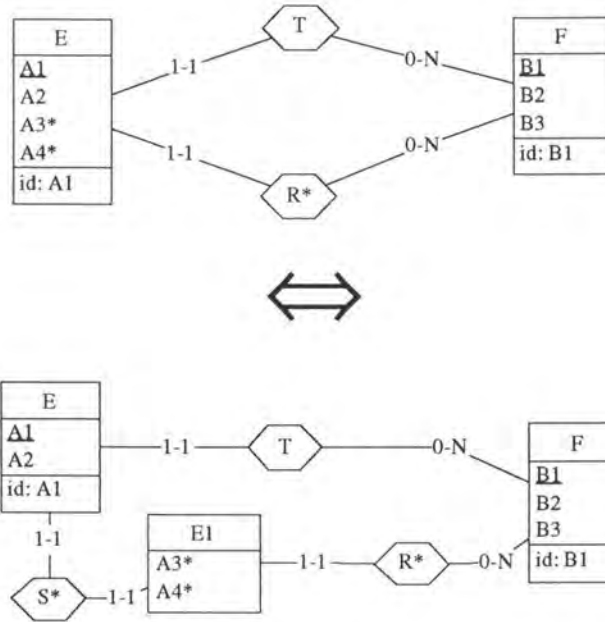


Figure V-8 - Cas particulier 3

Un exemple illustrant ce cas particulier est présenté par la figure qui suit.

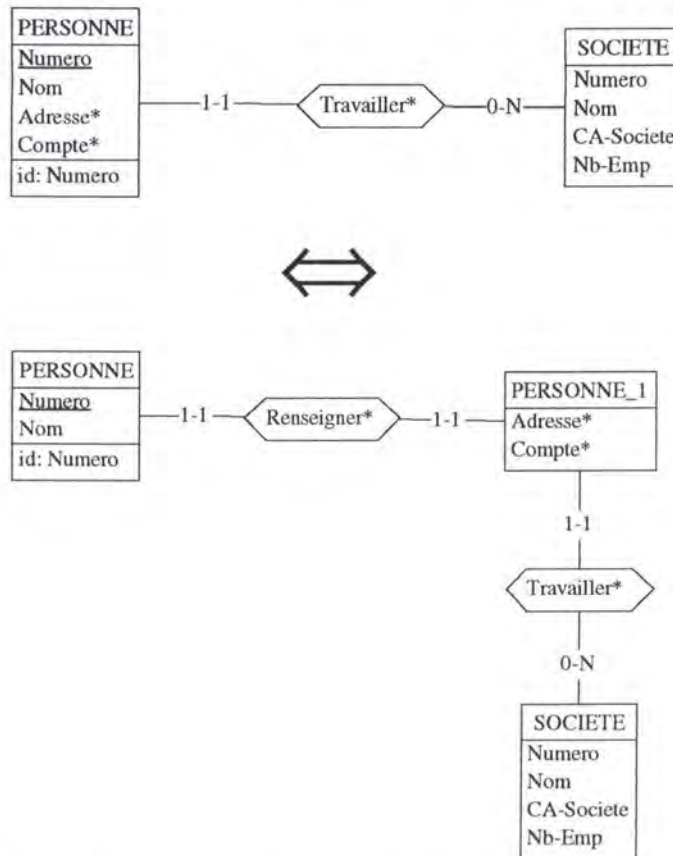


Figure V-9 - Exemple cas particulier 3

B. Transformations TER → ER

Dans ce paragraphe, nous évoquons les transformations des structures de données temporelles en structures de données non temporelles. Il s'agit donc d'exprimer l'aspect temporel à l'aide de concepts appartenant au modèle conceptuel de base.

Avant la présentation des transformations, nous rappelons ce que nous entendons par version d'une instance d'un type d'entité et d'un type d'association. Ensuite, nous allons nous intéresser à la transformation d'un type d'entité temporel. Puis, nous évoquerons une transformation qui s'occupe des attributs temporels. Enfin, nous aborderons la transformation des types d'association temporels.

1. Définition préliminaire

Une version d'une instance d'un type d'entité est l'état de cette instance pendant une certaine période de temps écoulée. Nous notons par e_i (resp. r_i) la $i^{\text{ème}}$ version de l'instance du type d'entité E (resp. l'instance du type d'association R). Nous devons aussi dire que la dernière version de l'instance d'un type d'entité correspond à l'état courant de cette instance. Nous l'appelons également version courante.

2. Transformations d'un type d'entité temporel

a) Comportement temporel sur l'existence (T-TEMPEXTE)

Dans un premier temps, nous allons transformer le type d'entité pour lequel nous souhaitons connaître le comportement temporel sur l'existence de chacune de ces instances. Cela se fera au moyen des concepts d'attributs monovalués atomiques du modèle conceptuel de base. Nous aurons ainsi un type d'entité non temporel présentant une sémantique équivalente au type d'entité temporel initial.

Pour évoquer cette transformation, nous considérons un type d'entité temporel E'' comportant deux attributs $A1$ et $A2$. Nous supposons que le premier de ces deux attributs identifie le type d'entité E'' .

La transformation exprime l'aspect temporel du type d'entité à l'aide de deux attributs monovalués atomiques. Grâce à ces deux attributs notés E -EXIST-DEBUT et E -EXIST-FIN[0-1], nous pouvons représenter la période pendant laquelle chaque instance du type d'entité existait ou existe encore.

Les valeurs contenues par ces deux attributs sont les suivantes :

E-EXIST-DEBUT contient la date à laquelle l'instance du type d'entité a commencé d'exister dans le monde réel.

$$E - \text{EXIST} - \text{FIN}[0 - 1] = \begin{cases} \text{la date à laquelle cette instance n'existe plus} \\ \infty \text{ si cette instance existe toujours.} \end{cases}$$

La valeur d'infini temporel sera représentée dans les bases de données par la valeur NULL.

La transformation T-TEMPEXTE est représentée par la Figure V-10.



Figure V-10 - Transformation T-TEMPEXTE

b) Comportement temporel sur les modifications (T-TEMPTE)

La transformation T-TEMPTE vise à exprimer la dimension temporelle à l'aide de deux attributs PERIODE-DEBUT et PERIODE-FIN[0-1]. Les valeurs contenues par ces deux attributs sont les suivantes :

PERIODE-DEBUT contient la date à partir de laquelle la version de l'instance de E a commencé d'être valide dans le monde réel.

$$\text{PERIODE} - \text{FIN}[0 - 1] = \begin{cases} \infty & \text{si la version de l'instance de E est toujours} \\ & \text{valide actuellement} \\ \text{La date à partir de laquelle la version de l'instance} \\ & \text{de E n'était plus valide sinon.} \end{cases}$$

La valeur d'infini temporel est représentée par la valeur NULL dans les bases de données.

Ces deux attributs sont les bornes de l'intervalle semi-ouvert [PERIODE-DEBUT , PERIODE-FIN[0-1]] qui permettent de décrire de manière continue l'évolution de l'état des instances du type d'entité.

Pour illustrer cette transformation, considérons un type d'entité temporelle E* constitué d'au moins un attribut. Soient par exemple A1, A2 et A3 les attributs de E* . Supposons également que A1 est l'identifiant de ce type d'entité. Cette transformation est représentée par la Figure V-11.



Figure V-11 - Transformation T-TEMPTE

Au type d'entité E que nous avons obtenu comme résultat, nous ajoutons la contrainte d'intégrité suivante :

$$\forall e \in \text{pop}(E) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$$

$$e_{(i-1)}. \text{PERIODE} - \text{FIN}[0-1] = e_i. \text{PERIODE} - \text{DEBUT}$$

En d'autres termes, cette contrainte signifie que pour chaque version d'une instance de E sauf la première, nous avons l'égalité entre les valeurs de l'attribut PERIODE-DEBUT et de l'attribut PERIODE-FIN[0-1] de la version qui précède.

Cette transformation des types d'entité temporels implique suivant les cas des changements au niveau des cardinalités des associations dans lesquelles ils assument un rôle. Nous distinguons les 3 cas suivants :

1. Association TE temporel - TE non temporel

Dans ce cas particulier, nous considérons un schéma composé d'un type d'entité temporel E* et d'un type d'entité non temporel F. Nous supposons que E* joue le rôle 1-1 d'un type d'association binaire R et que F joue le rôle 0-N de R. Si nous appliquons T-TEMPTE à ce schéma, le résultat est donné par l'équivalence suivante:

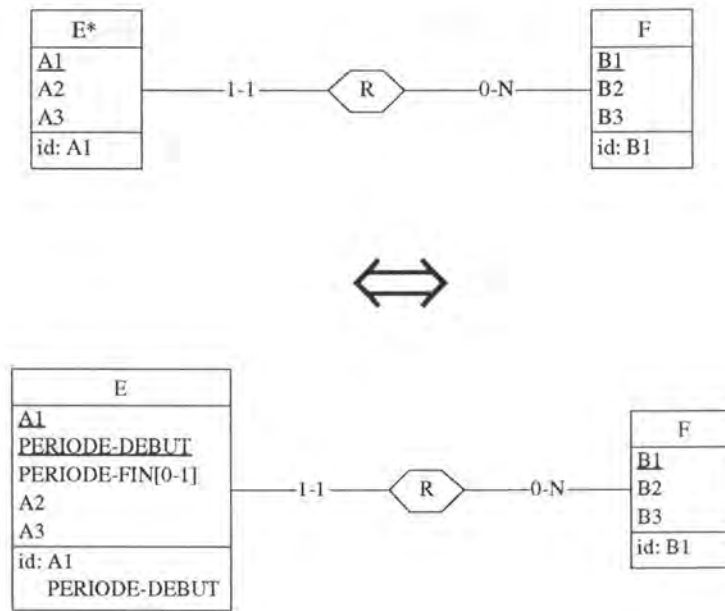


Figure V-12 - Association TE temporel -TE non temporel

Ce cas de figure n'entraîne aucune modification des cardinalités car la connectivité du rôle assumé par le type d'entité non temporel est 0-N. En effet, une instance du type d'entité F est référencé par 0 ou plusieurs versions d'instances du type d'entité E. D'où, la cardinalité du rôle est toujours 0-N.

2. Association TE temporel - TE temporel

Soient deux types d'entité temporels E* et F*. Supposons que E* joue le rôle 1-1 d'un type d'association binaire R et F* joue le rôle 0-N de R.

Après application de T-TEMPTE, nous obtenons le résultat suivant :

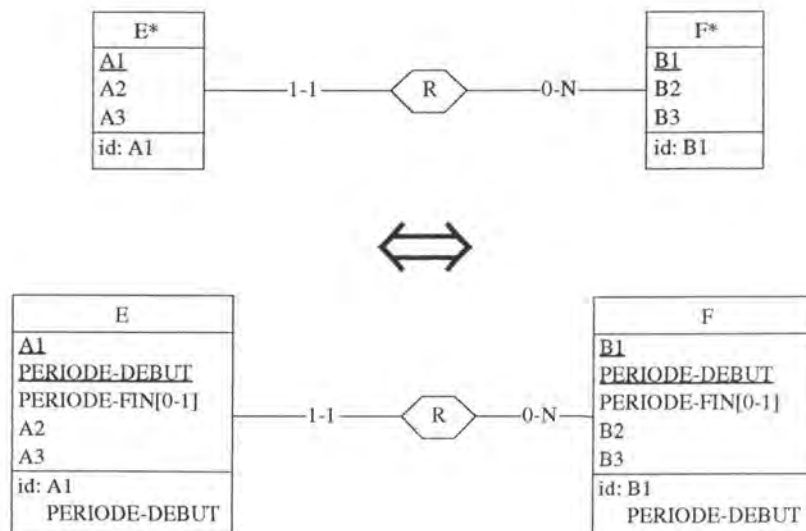


Figure V-13 - Association TE temporel - TE temporel

Dans ce cas également, les cardinalités n'ont pas changé mais nous interprétons le schéma de la manière suivante : toute version e_i d'une instance de E référence la version f_i d'une instance de F. Cette version f_i était courante à l'instant où e_i est devenue valide. Comme nous l'avons déjà vu, cette référence se fera par un attribut de référence temporel dans le modèle relationnel.

3. Association TE non temporel - TE temporel

Nous considérons ici un schéma composé d'un type d'entité non temporel E et d'un type d'entité temporel F*. Nous supposons que E joue le rôle 1-1 d'un type d'association binaire R et que F* joue le rôle 0-N de R.

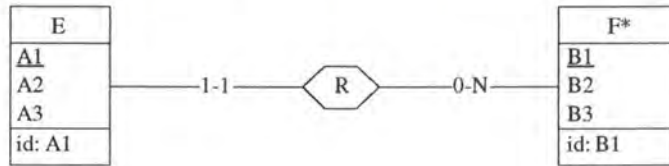


Figure V-14 - Association TE non temporel - TE temporel (1)

Si nous appliquons la transformation T-TEMPTE, il résulte le schéma suivant :

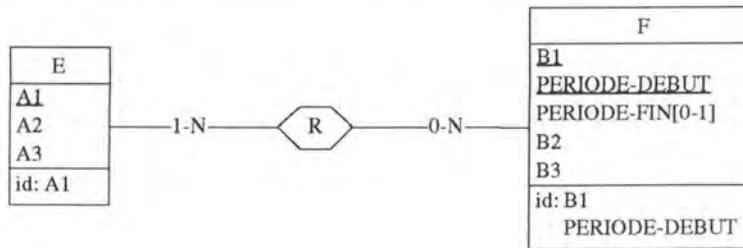


Figure V-15 - Association TE non temporel - TE temporel (2)

La cardinalité du rôle assumé par E est devenu 1-N car une instance de E ne référence par une et une seule instance de F mais un historique de cette instance. D'où, cela implique un changement de la cardinalité puisque l'historique est composé d'une ou plusieurs versions de l'instance. Nous observons dans ce cas l'apparition d'une structure non conforme au modèle conceptuel de base, à savoir le type d'association many-to-many. C'est pourquoi, dans ce cas particulier, nous proposons de considérer la transformation de la Figure V-16.

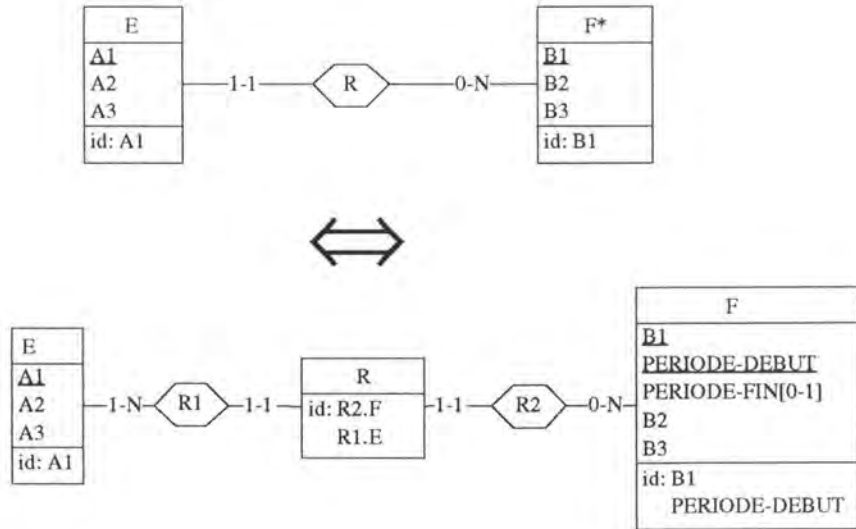


Figure V-16 - Transformation particulière des TE temporels

Pour obtenir ce résultat, il suffit de composer T-TEMPTE et la transformation d'un type d'association many-to-many en type d'entité.

De plus, nous constatons que cette transformation implique une modification de l'identifiant. Elle ajoute l'attribut PERIODE-DEBUT à l'identifiant. Considérons la table CLIENT suivante où chaque client est identifié par son numéro NCLI et où la granularité du temps est d'une année.

CLIENT					
NCLI	Nom	Periode-debut	Periode-fin[0-1]	Prénom	Adresse
1	Dupont	1973	1986	Jérôme	Rue des tilleuls, 23 Recogne
2	Durant	1965	1983	Nathalie	rue du village, 32 Libramont
1	Dupont	1986	1991	Jérôme	rue du feu, 18 Libin
2	Leroy	1983	-	Nathalie	rue du village, 32 Libramont
1	Dupont	1991	-	Jérôme	rue de la carrière, 110 Arlon

Figure V-17 - Table temporelle CLIENT

Si nous n'ajoutons pas l'attribut PERIODE-DEBUT à l'identifiant qui était NCLI, la contrainte d'intégrité d'identifiant sera violée. Nous avons dans cet exemple trois fois NCLI égale à 1 et deux fois ce même attribut égal à 2.

Nous considérons également que la partie de l'identifiant de E qui identifiait E* est stable. Dès lors, le changement de version est impliqué par la modification d'une des valeurs des attributs qui n'identifiaient pas E*. De plus, le changement de version se passe de la façon suivante, $\forall e \in \text{pop}(E)$:

$$\left\{ \begin{array}{ll} e_{d'}.A_j = e_d.A_j & \forall \text{ attributs } A_j \text{ non modifié(s)} \\ e_{d'}.A_j = A'_j & \forall \text{ attributs } A_j \text{ modifié(s)} \\ e_d.\text{PERIODE} - \text{FIN}[0 - 1] = \text{date de début de validité de } e_{d'} \\ e_{d'}.\text{PERIODE} - \text{DEBUT} = \text{date de début de validité de } e_d \\ e_{d'}.\text{PERIODE} - \text{FIN}[0 - 1] = \text{NULL} \end{array} \right.$$

où d' est l'indice de la nouvelle version courante, d est l'indice de l'ancienne version courante et le(s) A_j' est(sont) le(s) nouvelle(s) valeur(s) du(des) attribut(s) modifié(s).

3. Transformation d'un attribut temporel élémentaire monovalué (T-TEMPAT)

Pour illustrer cette transformation, nous allons considérer ici un type d'entité E contenant au moins deux attributs. Supposons qu'un seul de ces attributs est temporel et qu'il n'est pas l'identifiant du type d'entité E.

Prenons l'exemple suivant pour illustrer la transformation :

E
Δ1
A2
A3*
id: A1

Figure V-18 - Exemple illustratif de T-TEMPAT

La transformation consiste à remplacer cet attribut temporel par trois attributs monovalués non temporels ayant comme préfixe le nom de cet attribut.

Ces attributs possèdent les valeurs suivantes :

1. A3-VALEUR = la valeur de l'attribut temporel monovalué élémentaire.
2. A3-PERIODE-DEBUT = la date à partir de laquelle la valeur de l'attribut est valide.

$$3. A3-PERIODE-FIN[0-1] = \begin{cases} \infty & \text{si la valeur de l'attribut est toujours} \\ & \text{valide.} \\ \text{la date à laquelle la valeur de l'attribut} & \\ & \text{n'était plus valide sinon.} \end{cases}$$

La valeur d'infinité temporelle sera représentée par la valeur NULL dans les bases de données.

Cette transformation correspond à l'équivalence de la Figure V-19.

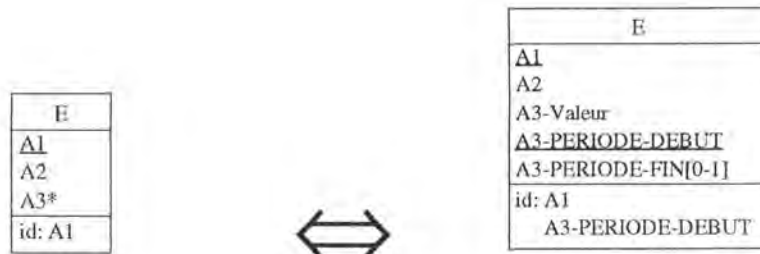


Figure V-19 - Transformation T-TEMPAT

Nous ajoutons également au résultat de la transformation la contrainte d'intégrité suivante :

$$\forall e \in \text{pop}(E) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$$

$$e_{(i-1)}.A3-FIN[0-1] = e_i.A3-DEBUT$$

En d'autres termes, cette contrainte signifie que pour chaque version d'une instance d'un type d'entité sauf la première, nous avons l'égalité entre les valeurs de l'attribut A3-DEBUT et de l'attribut A3-FIN[0-1] de la version qui précède.

Cette transformation est différente de T-TEMPTE car le changement de version d'une instance du type d'entité n'a lieu que lorsque la valeur de l'attribut temporel est modifiée. D'où, le changement de version se déroule comme suit, $\forall e \in \text{pop}(E)$:

$$\left\{ \begin{array}{l} e_{d'} \cdot A_j = e_d \cdot A_j \quad \forall \text{attribut(s) } A_j \text{ non temporel(s)} \\ e_{d'} \cdot A_k - \text{VALEUR} = A'_k \quad \text{où } A_k \text{ est l'attribut temporel} \\ e_{d'} \cdot A_k - \text{PERIODE} - \text{FIN}[0-1] = \text{date de début de validité de } A'_k \\ e_{d'} \cdot A_k - \text{PERIODE} - \text{DEBUT} = \text{date de début de validité de } A'_k \\ e_{d'} \cdot A_k - \text{PERIODE} - \text{FIN}[0-1] = \text{NULL} \end{array} \right.$$

où d' est l'indice de la nouvelle version courante, d est l'indice de l'ancienne version courante et A'_k est la nouvelle valeur de l'attribut temporel.

4. Transformations des types d'association temporels (T-TEMPREL)

Pour expliquer cette transformation, considérons deux types d'entité E et F caractérisés respectivement par les attributs A1, A2 et B1, B2. Supposons que les identifiants respectifs de E et F soient A1 et B1. Soit R^* le type d'association temporel mettant en relation les deux types d'entité définis précédemment.

La transformation d'un type d'association temporel est la suivante : cette dimension temporelle est exprimée à l'aide de deux attributs R-PERIODE-DEBUT et R-PERIODE-FIN[0-1] inclus dans le type d'association R. Ces deux attributs indiquent respectivement le début et la fin de la période durant laquelle une instance du type d'entité E est en relation avec une certaine instance du type d'entité F. Les valeurs contenues par ces attributs sont donc les suivants :

R-PERIODE-DEBUT contient la date à partir de laquelle l'association entre une instance de E et une instance de F est valide.

$$R - PERIODE - FIN[0 - 1] = \begin{cases} \infty & \text{si elle est toujours valide actuellement} \\ \text{La date à partir de laquelle l'association entre} \\ \text{l'instance de E et celle de F n'était plus} \\ \text{valide sinon} \end{cases}$$

La valeur d'infini temporel est représentée au niveau des bases de données par la valeur NULL.

Ces attributs sont les bornes de l'intervalle semi-ouvert [R-PERIODE-DEBUT, R-PERIODE-FIN[0-1]] qui permettent de décrire de manière continue l'évolution de l'état de chaque instance du type d'association.

La transformation est explicitée par la Figure V-20.

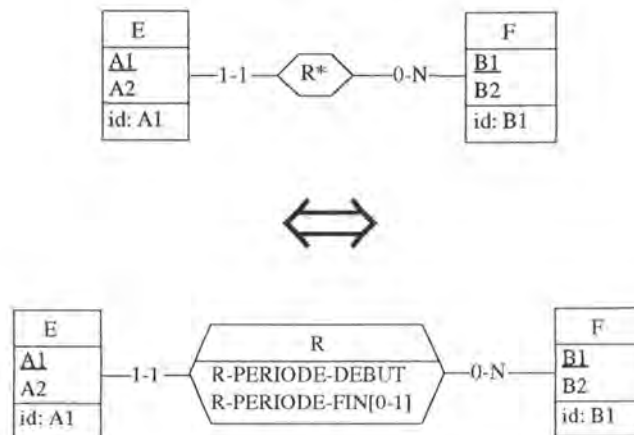


Figure V-20 - Transformation T-TEMPREL

Au résultat de la transformation, nous devons ajouter la contrainte d'intégrité suivante :

$$\forall e \in \text{pop}(E) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$$

$$r_{(i-1)}.R - PERIODE - FIN[0 - 1] = r_i.R - PERIODE - DEBUT$$

En d'autres termes, cette contrainte signifie que pour chaque version d'une instance d'un type d'association sauf la première, nous avons l'égalité entre les valeurs de l'attribut R-PERIODE-DEBUT et de l'attribut R-PERIODE-FIN[0-1] de la version qui précède.

Or, cette transformation fait apparaître dans le résultat des concepts que nous ne considérons pas dans le modèle de base, à savoir des types d'association avec attribut.

C'est la raison pour laquelle nous utiliserons la transformation présentée dans la Figure V-21 pour détemporaliser le type d'association temporel.

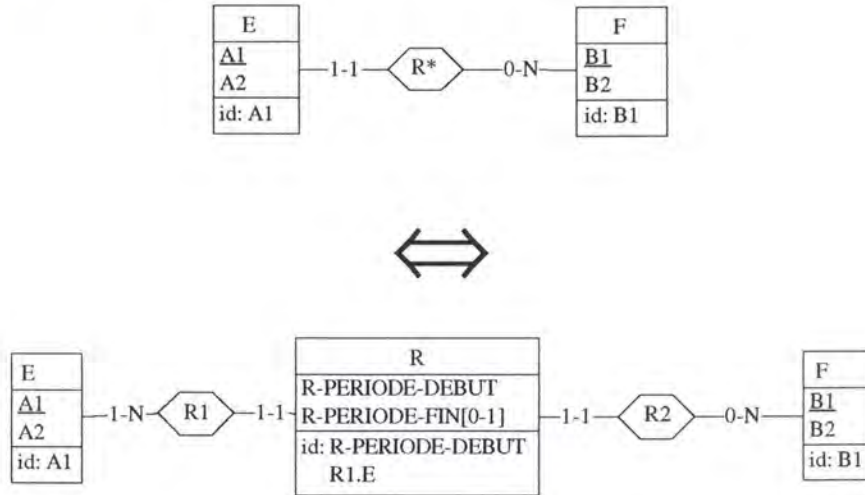


Figure V-21 - Transformation T-TEMPREL (2)

Au résultat de la transformation, nous devons aussi ajouter la contrainte d'intégrité suivante :

$$\forall r \in \text{pop}(R) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$$

$$r_{(i-1)}.R - \text{PERIODE} - \text{FIN}[0 - 1] = r_i.R - \text{PERIODE} - \text{DEBUT}$$

C. Transformations ER \rightarrow RM

Dans un schéma conforme au modèle conceptuel de base, nous y trouvons des incompatibilités par rapport au modèle relationnel. Nous appliquons donc des transformations qui éliminent ces incompatibilités. Ainsi, nous obtenons un schéma qui est conforme au modèle relationnel utilisé.

Les incompatibilités qui restent dans un schéma conceptuel sont les types d'association fonctionnels. Pour transformer les types d'association fonctionnels, nous devons appliquer l'une des transformations présentées dans la suite, suivant le cas dans lequel nous nous trouvons.

1. Le premier cas est la transformation d'un type d'association classique en attribut de référence.

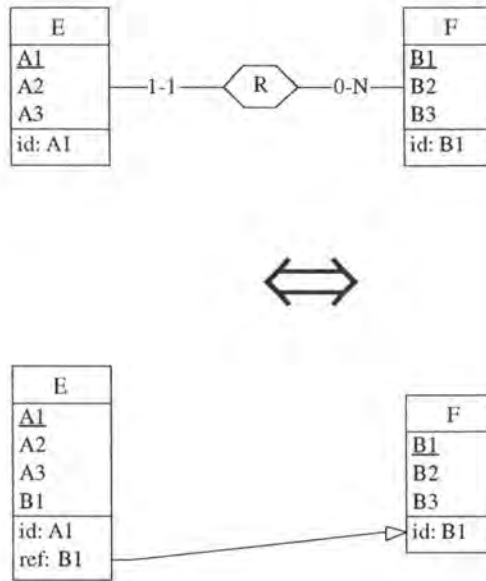


Figure V-22 - Transformation d'un TA en attribut de référence

2. Le cas suivant se présente lorsque le type d'association connecte deux types d'entité temporels. Ce type d'association est alors transformé en attribut de référence temporel.

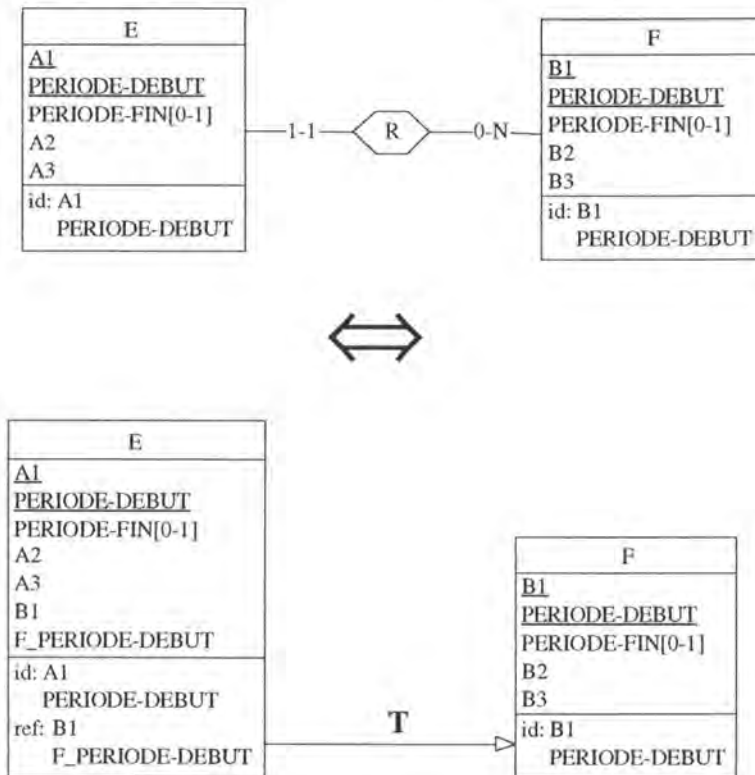


Figure V-23 - Transformation d'un TA en attribut de référence temporel

VI. Règles de cohérence

Dans le paragraphe précédent, nous avons exposé les différentes transformations qui permettent d'exprimer la temporalité à l'aide de concepts du modèle de base. Néanmoins, ces transformations s'accompagnent de règles de cohérence qu'il faut respecter. Les premières règles concernent la cohérence des schémas temporels. Ensuite, nous présenterons les règles qui doivent être respectées après avoir appliqué les transformations.

Avant de passer à ces règles de cohérence, nous indiquons les notations qui seront utilisées dans celles-ci.

A. Notations

Dans les règles de cohérence, nous pouvons retrouver les termes qui suivent :

- E désigne l'ensemble des types d'entité non temporels dans le schéma conceptuel des données.
- E^T désigne l'ensemble des types d'entité temporels dans le schéma conceptuel des données.
- d dénote la dernière version d'une structure de données.
- TA^T nomme l'ensemble des types d'association temporels dans le schéma conceptuel des données.

B. Règles de cohérence des schémas temporels

Les règles de cohérence que nous proposons dans ce paragraphe sont utilisées pour avoir un schéma temporel cohérent.

1. **Règle 1 :** Les symboles * et " sont uniquement utilisés pour indiquer la temporalité des concepts.

En d'autres termes, cela signifie que les deux symboles * et " seront considérés comme des symboles spéciaux. Ils seront destinés spécifiquement à l'indication de structures de données temporelles.

2. **Règles 2 :** Il n'existe pas de type d'entité $E^* \in \mathbf{E}^T$ tel que l'un de ces attributs est temporel.

Lorsqu'un type d'entité est temporel, c'est que nous désirons connaître la moindre évolution des valeurs de chacun de ses attributs. Il y a un changement de version à la moindre modification de valeur. D'où, il serait incohérent de demander dans ces types d'entité l'évolution d'un attribut particulier. Nous pourrions déduire l'évolution d'un attribut d'une instance à partir de l'historique de l'instance du type d'entité.

3. **Règle 3 :** Il n'existe pas de type d'entité $E^* \in \mathbf{E}^T$ tel que ce type d'entité temporel assume le rôle 1-1 d'un type d'association temporel.

Nous avons considéré que le type d'association constituait un attribut du type d'entité situé du côté de la connectivité 1-1. Dès lors, nous pensons que le type d'association ne doit pas être temporel (cfr. Règle 2).

C. Règles de cohérence post-transformations

Nous allons voir quelles sont les règles de cohérence que nous devons respecter chaque fois que nous appliquons une des transformations décrites dans le chapitre précédent.

1. **Règle 4 :**

a) Cas des types d'entité temporels

$\forall E \in \mathbf{E}^T$ tel que $E = T\text{-TEMPTE}(E^*)$, nous avons :

$\forall e \in \text{pop}(E)$ et $\forall i \quad 2 \leq i \leq \text{nombre de versions}$

$$e_{(i-1)}.PERIODE - \text{FIN}[0-1] = e_i.PERIODE - \text{DEBUT}$$

b) Cas d'un attribut temporel d'un type d'entité

$\forall E \in \mathbf{E}$ tel que $E = T\text{-TEMPAT}(E, A^*)$, nous avons :

$\forall e \in \text{pop}(E) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$

$$e_{(i-1)}.A - \text{FIN}[0-1] = e_i.A - \text{DEBUT}$$

c) Cas d'un type d'association temporel

$\forall R \in \text{TA}^T$ tel que $R = \text{T-TEMPREL}(R^*)$, nous avons :

$\forall e \in \text{pop}(E) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$

$$r_{(i-1)}.R - \text{PERIODE} - \text{FIN}[0-1] = r_i.R - \text{PERIODE} - \text{DEBUT}$$

Expliquons le premier de ces trois cas semblables. Dans le cas d'un type d'entité temporel, nous avons vu que la temporalité est exprimée par deux attributs PERIODE-DEBUT et PERIODE-FIN[0-1]. Cette contrainte impose que pour toutes les versions d'une instance sauf la première, l'attribut PERIODE-DEBUT est égal à l'attribut PERIODE-FIN[0-1] de la version qui précède.

Or, nous savons que ces attributs sont les bornes de l'intervalle d'existence de chaque version de l'instance d'un type d'entité. Dès lors, grâce à cette contrainte, nous rendons les intervalles contigus et nous retrouverons dans la base de données une évolution continue des données pour lesquelles nous souhaitons connaître l'historique.

2. Règle 5 : Les intervalles d'existence des concepts temporels sont DISJOINTS.

Cette propriété des intervalles est réalisée de la manière suivante : par la règle 4, nous avons une contiguïté des intervalles de validité et donc un recouvrement continu du temps. De plus, il faut interpréter les intervalles de bornes PERIODE-DEBUT et PERIODE-FIN[0-1] comme des intervalles semi-ouverts à droite. De cette façon, nous évitons que chaque intervalle ait une intersection.

Dès lors, il faut faire très attention lorsqu'une query possède une condition sur les attributs exprimant l'aspect temporel.

Exemple : Soit une partie d'une base de données contenant les clients d'une entreprise. Supposons qu'elle est composée d'une table reprenant le nom, l'adresse et le compte du client et supposons également que nous avons une granularité du temps qui est d'un jour (cfr. Figure VI-1).

CLIENT				
NOM	ADRESSE	COMPTE	PERIODE-DEBUT	PERIODE-FIN[0-1]
Dupont	Bruxelles	1500	01-08-1992	07-06-1994
Laloi	Arlon	300	04-12-1989	20-01-1995
Dupont	Liège	1500	07-06-1994	13-05-1996
Dupont	Anvers	1500	13-05-1996	NULL
Laloi	Virton	0	20-01-1995	NULL

Figure VI-1 - Exemple de table temporelle CLIENT

Par exemple, nous avons dans cette base de données un client de nom DUPONT qui a habité Bruxelles entre le 01-08-1992 et jusqu'au 07-06-1994 non inclus. A partir de cette seconde date jusqu'au 13-05-1996, la base de données nous renseigne que ce client a habité la localité de Liège.

3. **Règle 6** : Dans cette règle de cohérence, nous retrouvons les trois sous-cas que nous avons déjà développés dans la règle 4.

a) Cas des types d'entité temporels

$$\forall E \in \mathbf{E}^T \text{ tel que } E = T\text{-TEMPTE}(E^*) \text{ et}$$

$$\forall e \in \text{pop}(E) \text{ et } \forall i : 1 \leq i \leq d$$

$$e_i.\text{PERIODE-DEBUT} < e_i.\text{PERIODE-FIN}[0-1]$$

b) Cas d'un attribut temporel d'un type d'entité

$$\forall E \in \mathbf{E} \text{ tel que } E = T\text{-TEMPAT}(E1, A^*) \text{ et}$$

$$\forall e \in \text{pop}(E) \text{ et } \forall i \quad 2 \leq i \leq \text{nombre de versions}$$

$$e_i.A\text{-PERIODE-DEBUT} < e_i.A\text{-PERIODE-FIN}[0-1]$$

c) Cas d'un type d'association temporel

$$\forall R \in \mathbf{TA}^T \text{ tel que } R = \text{T-TEMPREL}(R^*)$$

et

$$\forall r \in \text{pop}(R) \text{ et } \forall i: 1 \leq i \leq d$$

$$r_i.\text{PERIODE} - \text{DEBUT} < r_i.\text{PERIODE} - \text{FIN}[0 - 1]$$

4. **Règle 7 :** $\forall E \in \mathbf{E}^T$ tel que $E = \text{T-TEMPTE}(E^*)$ et $\forall e \in \text{pop}(E)$, nous avons un changement de version uniquement lorsqu'au moins une valeur d'attribut est modifiée. Le changement de version s'opère de la manière suivante :

$$\left\{ \begin{array}{ll} e_{d'} \cdot A_j = e_d \cdot A_j & \forall \text{ attributs } A_j \text{ non modifié(s)} \\ e_{d'} \cdot A_j = A'_j & \forall \text{ attributs } A_j \text{ modifié(s)} \\ e_{d'}.\text{PERIODE} - \text{FIN}[0 - 1] = \text{date de début de validité de } e_{d'} \\ e_{d'}.\text{PERIODE} - \text{DEBUT} = \text{date de début de validité de } e_{d'} \\ e_{d'}.\text{PERIODE} - \text{FIN}[0 - 1] = \text{NULL} \end{array} \right.$$

où d' est l'indice de la nouvelle version courante, d est l'indice de l'ancienne version courante et le(s) A'_j est(sont) le(s) nouvelle(s) valeur(s) du(des) attribut(s) modifié(s).

En d'autres termes, nous avons une nouvelle version de l'instance d'un type d'entité temporel chaque fois qu'une ou plusieurs valeurs d'attributs sont modifiées. Cette version est une copie de l'ancienne excepté pour le(s) attribut(s) modifié(s) et les deux attributs indiquant la temporalité.

5. **Règle 8 :** $\forall E \in \mathbf{E}$ tel que $E = \text{T-TEMPAT}(E1, A^*)$ et $\forall e \in \text{pop}(E)$, nous avons un changement de version de e uniquement lorsque la valeur de l'attribut temporel est modifiée.

Le changement de version se passe de la façon suivante :

$$\left\{ \begin{array}{ll} e_{d'} \cdot A_j = e_d \cdot A_j & \forall \text{ attribut(s) } A_j \text{ non temporel(s)} \\ e_{d'} \cdot A_k - \text{VALEUR} = A'_k & \text{où } A_k \text{ est l'attribut temporel} \\ e_{d'} \cdot A_k - \text{PERIODE} - \text{FIN}[0-1] = \text{date de début de validité de } A'_k \\ e_{d'} \cdot A_k - \text{PERIODE} - \text{DEBUT} = \text{date de début de validité de } A'_k \\ e_{d'} \cdot A_k - \text{PERIODE} - \text{FIN}[0-1] = \text{NULL} \end{array} \right.$$

où d' est l'indice de la nouvelle version courante, d est l'indice de l'ancienne version courante et A'_k est la nouvelle valeur de l'attribut temporel.

Nous avons une nouvelle version de l'instance du type d'entité chaque fois que la valeur de l'attribut temporel est modifiée. Cette nouvelle version ne diffère de l'ancienne version courante que pour la valeur des différents attributs remplaçant l'attribut temporel

6. **Règle 9** : $\forall E \in \mathbf{E}$ tel que $E = T\text{-TEMPAT}(E1, A^*)$ et $\forall e \in \text{pop}(E)$, tout changement de la valeur d'un attribut non temporel est modifié uniquement dans e_d .

Cette règle implique la conséquence suivante : nous disposerons d'un historique partiel des attributs non temporels. Cet historique est partiel car il est possible que nous modifions plusieurs fois la valeur d'un attribut non temporel avant de changer de version. L'historique de ces attributs serait complet si, entre chaque modification de valeur des attributs non temporels, nous avons au moins une modification de valeur d'un attribut temporel.

VII. Plans de transformation d'un schéma temporel

Dans ce chapitre, nous allons développer un premier plan qui transforme un schéma temporel en un schéma relationnel. Ce plan de transformation assure uniquement que nous obtiendrons un schéma conforme au modèle relationnel.

Nous proposons ensuite un second plan de transformation. Celui-ci est plus souple que le premier car il dépend d'un paramètre. Le but de ce paramètre est réaliser une pré-optimisation. En fonction de la valeur de ce paramètre, certaines opérations seront effectuées ou ne le seront pas. Néanmoins, il faut constater qu'il s'agit d'une proposition. Une étude plus approfondie est nécessaire.

En premier lieu, nous listerons les incompatibilités d'un schéma conceptuel temporel par rapport au modèle relationnel que nous nous sommes fixés. Puis, nous définirons le premier plan de transformation et, enfin, nous évoquerons le second.

A. Incompatibilités

Les concepts présents dans un schéma temporel et qui constituent des incompatibilités pour le modèle relationnel sont les suivants :

1. Les attributs temporels
2. Les types d'entité temporels
3. Les types d'association temporels
4. Les types d'association non temporels
5. Les contraintes autres que les identifiants et les contraintes référentielles

B. Premier plan de transformation

Lorsque nous devons transformer un schéma temporel, nous recherchons dans le schéma les incompatibilités par rapport au modèle relationnel. Puis, nous appliquons les transformations adéquates pour les éliminer.

1. Types d'entité contenant un ou plusieurs attributs temporels et au moins un attribut non temporel.

Dans ce cas précis, nous appliquons ECLAT-TE pour regrouper au sein d'un type d'entité les attributs temporels et conserver dans le type d'entité initial les attributs non temporels.

2. Types d'entité contenant uniquement des attributs temporels.

Nous appliquons à ces types d'entité la transformation TEMP-TE qui généralise au type d'entité l'aspect temporel des attributs.

Nous devons observer que ces cas doivent être envisagés dans cet ordre, car la première opération peut entraîner l'apparition de nouveaux types d'entité contenant uniquement des attributs temporels.

Après avoir éliminé ces structures incompatibles, nous obtenons un schéma temporel où chaque structure temporelle peut être transformée à l'aide des transformations TEMPEXTE, T-TEMPTE et T-TEMPREL. Ces structures font partie de :

3. Types d'entité temporels.

Nous utilisons la transformation T-TEMPTE ou T-TEMPEXTE pour les détemporaliser.

4. Types d'association temporels.

Nous détemporalisons ces types d'association en appliquant T-TEMPREL.

Une fois ces structures transformées, nous avons un schéma conceptuel conforme au modèle conceptuel de base. Il ne reste plus qu'à appliquer les transformations qui nous permettent d'obtenir un schéma conforme au modèle relationnel fixé.

5. Types d'association non temporels

Nous transformons les types d'association fonctionnels en attributs de référence temporel si ils mettent en correspondance deux types d'entité temporels ou en attributs de référence sinon.

6. Contraintes autres que les identifiants et les contraintes référentielles

Nous en faisons des contraintes additionnelles.

Finalement, nous avons comme résultat un schéma conforme au modèle relationnel.

C. Second plan de transformation

Ce second plan est appliqué en fonction d'un paramètre que nous devons nous fixer. Il est semblable au premier plan excepté pour la première opération qui est exécutée en fonction d'un paramètre. Nous réalisons cette opération lorsque les types d'entité contiennent plus de $\alpha\%$ d'attributs temporels. De plus, nous ajoutons une nouvelle opération entre les étapes 2 et 3 du premier plan. Celle-ci va transformer les attributs temporels des types d'entité qui en contiennent moins de $\alpha\%$ en attributs non temporels. Cela se passera de manière semblable à la transformation T-TEMPAT.

D'où, le plan de transformation est le suivant :

1. Types d'entité contenant au moins $\alpha\%$ d'attributs temporels et au moins un attribut non temporel.

Application d'ECLAT-TE.

2. Types d'entité contenant uniquement des attributs temporels.

Application de TEMP-TE.

3. Attributs temporels d'un type d'entité qui en contient moins de $\alpha\%$.

Application d'une transformation semblable à T-TEMPAT

4. Types d'entité temporels.

Application de T-TEMPTE ou T-TEMPEXTE.

5. Types d'association temporels.

Application de T-TEMPREL.

6. Types d'association non temporels

Transformation des types d'association fonctionnels en attributs de référence temporel si ils mettent en correspondance deux types d'entité temporels ou en attributs de référence sinon.

7. Contraintes autres que les identifiants et les contraintes référentielles

En faire des contraintes additionnelles.

VIII. Etudes de cas

Nous allons développer un premier cas simple composé de deux types d'entité et d'un type d'association temporel. Nous étudierons ensuite un exemple plus complexe pour constater les difficultés de transformations des schémas conceptuels temporels.

A. Première étude de cas

Nous appliquerons dans cette première étude de cas le premier plan de transformation que nous venons de présenter. Nous donnerons au second point un exemple de base de données qui pourrait correspondre au schéma relationnel résultant. Enfin, nous présenterons quelques exemples de queries qui sont susceptibles d'être posées.

Dans cette première étude de cas, nous considérons le schéma conceptuel temporel constitué des types d'entité EMPLOYE et DEPARTEMENT ainsi que d'un type d'association temporel entre ces deux types d'entité. De plus, EMPLOYE est caractérisé par les attributs temporels Adresse*, Localité* et Fonction*.

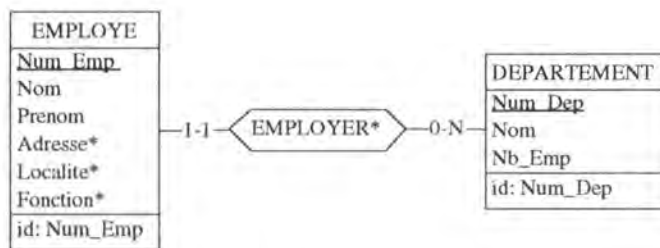


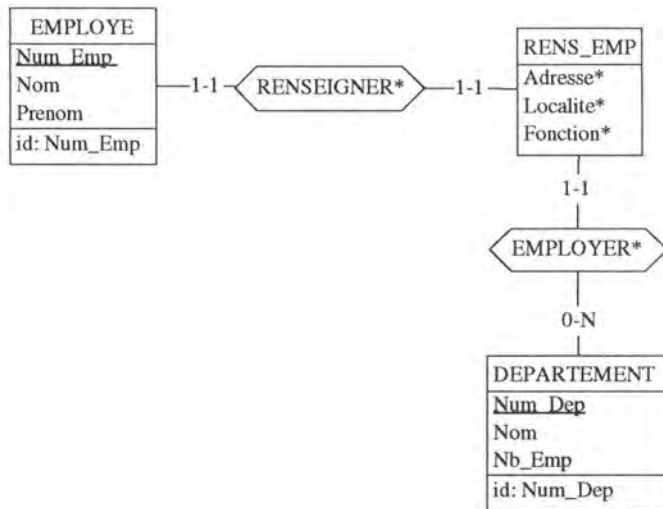
Figure VIII-1 - Etude de cas 1

1. Application du premier plan de transformation

Nous appliquons le premier plan de transformation que nous venons de décrire dans le chapitre précédent au schéma de la Figure VIII-1.

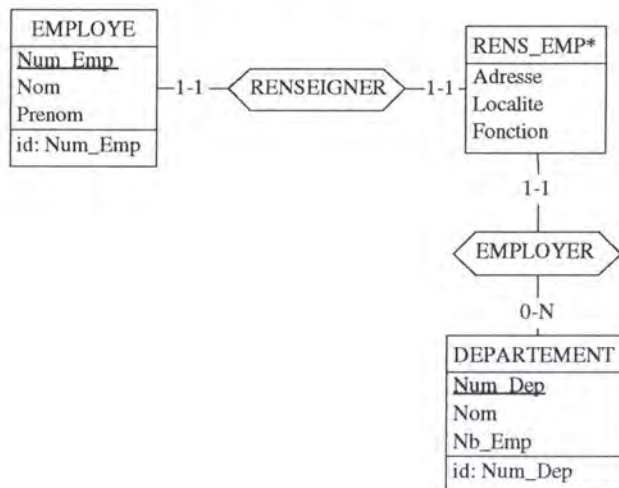
1. Types d'entité contenant plusieurs attributs temporels et au moins un attribut non temporel.

Nous voyons dans le schéma initial que le type d'entité Employe se trouve dans la situation présente. Dès lors, nous l'éclatons en appliquant le premier cas particulier de la transformation ECLAT-TE.



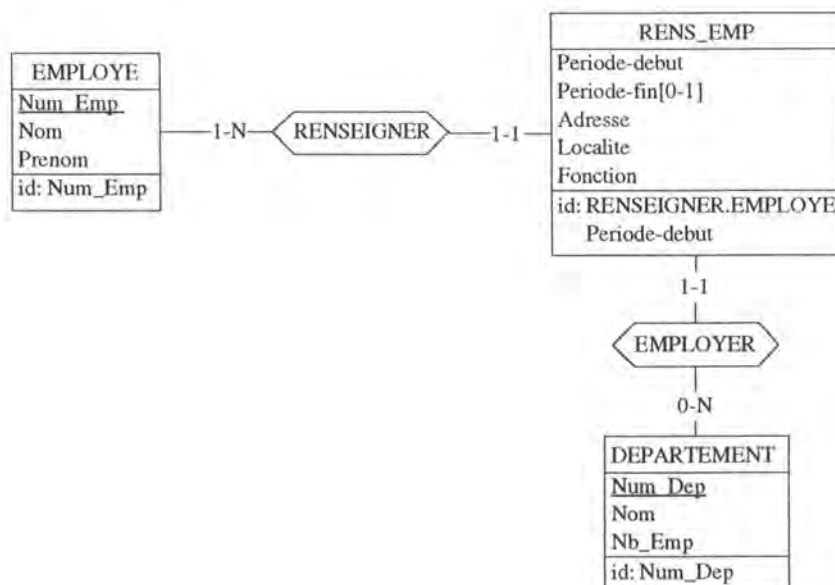
2. Types d'entité contenant uniquement des attributs temporels.

Dans le schéma résultant de la première opération, nous regardons s'il n'y a pas de types d'entité composé d'attribut uniquement temporel. Nous remarquons que RENS_EMP* est un de ces types d'entité et nous lui appliquons la transformation TEMPTE.



3. Types d'entité temporels.

Nous recherchons maintenant si le schéma contient des types d'entité temporels. C'est le cas du type d'entité RENS_EMP. Nous le détemporalisons grâce à la transformation T_TEMPTE et nous obtenons :

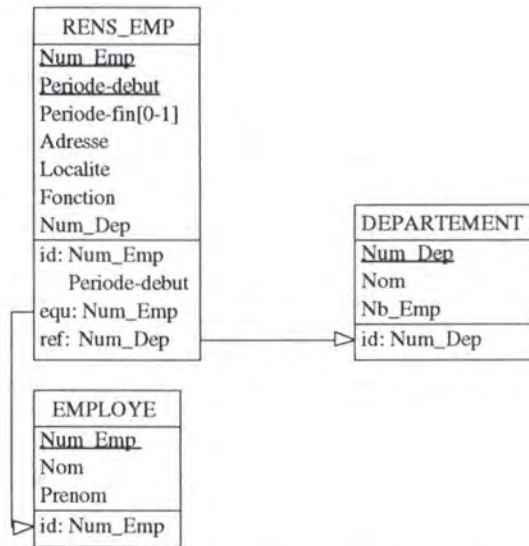


4. Types d'association temporels.

Dans le résultat précédent, nous constatons qu'il n'y a plus de types d'association temporels. Dès lors, aucun changement n'est effectué.

5. Types d'association non temporels et fonctionnels

Nous transformons ensuite les types d'association non temporels et fonctionnels dans l'étape suivante. Les types d'association RENSEIGNER et EMPLOYER sont dès lors transformés en attributs de référence.



+ Contraintes additionnelles

2. Exemple de base de données

Dans ce paragraphe, nous présentons un exemple qui illustre le schéma que nous avons obtenu une fois le plan de transformation appliqué. Nous proposons tout d'abord un exemple de table EMPLOYE, puis un exemple de table DEPARTEMENT et un exemple de table RENS_EMP.

EMPLOYE		
NUM_EMP	NOM	PRENOM
1	Durant	Philippe
2	Leroy	Pierre
3	Guillaume	Jean
4	Thill	Yves
5	Lambert	Serge
6	Toussaint	Manu
7	Libois	Thierry

Figure VIII-2 - Exemple de table EMPLOYE

DEPARTEMENT		
NUM_DEP	NOM	NB_EMP
100	Production	270
200	Vente	240
300	Informatique	300
400	Maintenance	100

Figure VIII-3 - Exemple de table DEPARTEMENT

RENS_EMP						
NUM_EMP	PERIODE_DEBUT	PERIODE_FIN	ADRESSE	LOCALITE	FONCTION	NUM_DEP
1	10-06-1972	08-04-1979	Rue du feu 36	Saint-Pierre	ouvrier	100
1	08-04-1979	19-09-1983	Rue du feu 36	Saint-Pierre	employe	100
1	19-09-1983	14-04-1989	Rue de Tervuren 30	Bruxelles	employe	100
1	14-04-1989	29-12-1992	Grand Place 3	Bruxelles	sous-directeur	100
1	29-12-1992	13-06-1993	Rue Albert 1er 55	Saint-Pierre	sous-directeur	100
1	13-06-1993	15-04-1997	Rue des Champs 98	Saint-Pierre	sous-directeur	100
1	15-04-1997	NULL	Rue des Champs 98	Saint-Pierre	directeur	100
2	01-06-1990	10-11-1993	Rue dessous la ville 18	Libramont	Encodeur	300
2	10-11-1993	13-01-1994	Rue du pressoir 19	Liège	Encodeur	300
2	13-01-1994	30-05-1995	Rue de Grandvoir 9	Charleroi	Analyste	300
2	30-05-1995	06-03-1997	Rue de Grandvoir 9	Charleroi	sous-directeur	300
2	06-03-1997	NULL	Rue des bois 33	Charleroi	sous-directeur	300
3	01-06-1972	27-09-1983	Rue des pierres 41	Libramont	vendeur	200
3	27-09-1983	13-11-1995	Rue des pierres 41	Libramont	commercial	200
3	13-11-1995	NULL	Rue des pierres 41	Libramont	sous-directeur	200
4	19-03-1983	13-06-1987	Rue des Champs 48	Libramont	employe	100
4	13-06-1987	15-09-1989	Rue du Serpont 301	Libramont	employe	100
4	15-09-1989	16-05-1993	Rue du monument 31	Bastogne	employe	100

4	16-05-1993	13-07-1997	Rue du monument 31	Bastogne	sous-directeur	100
4	13-07-1997	NULL	Rue du canal 31	Bruxelles	sous-directeur	100
5	19-04-1977	01-06-1983	Rue du fleuve 31	Liège	ouvrier	400
5	01-06-1983	08-03-1990	Rue des blés 3	Liège	ouvrier	400
5	08-03-1990	30-11-1993	Rue de Bruxelles 1	Liège	ouvrier	400
5	30-11-1993	NULL	Rue des livres 10	Charleroi	ouvrier	400
6	07-06-1990	NULL	Rue de la gare 33	Bruxelles	employe	100
7	13-05-1992	30-06-1995	Rue de la gare 44	Bruxelles	technicien	300
7	30-06-1995	NULL	Rue Neuve 101	Bruxelles	technicien	300

Figure VIII-4 - Exemple de table RENS_EMP

3. Exemples de query

Dans ce paragraphe, nous allons donner quelques exemples de queries pour constater la difficulté introduite par la temporalité.

1. Qui remplissait la fonction d'employé entre 01-01-1979 et 01-01-1986 ?

Il faut déterminer les noms des employés tels que :

1. Ceux-ci n'exercent plus la fonction d'employé mais l'exerçaient à l'époque.
2. Ceux-ci exercent toujours actuellement la fonction d'employé.

La requête correspondant à cette question est la suivante :

```
select distinct e.Nom, e.Prenom
from EMPLOYE e, RENS_EMP r
where e.Num_Emp = r.Num_Emp
and r.Fonction = 'employe'
and (( r.Periode_fin > 01-01-1979 and r.Periode_fin < 01-01-1986)
or ( r.Periode_debut > 01-01-1979 and r.Periode_debut < 01-01-1986)
or ( r.Periode_debut < 01-01-1979 and ( r.Periode_fin >01-01-1986
or r.Periode_fin is null)));
```

La solution correspondant à cette requête est donnée par la Figure VIII-5.

NOM	PRENOM
Durant	Philippe
Thill	Yves

Figure VIII-5 - Solution requête 1

2. Qui exerçait la fonction d'employé lorsque Durant est devenu sous-directeur ?

Comme pour la première question qui a été posée, nous devons déterminer les noms et prénoms des personnes qui étaient engagées comme employé au moment où Durant est devenu sous-directeur.

La requête relative à la question est donnée par :

```
select distinct e.Nom, e.Prenom
from EMPLOYE e, RENS_EMP r
where e.Num_Emp = r.Num_Emp
and r.Fonction = 'employe'
and r.Periode_debut <= (select min(r.Periode_debut)
                        from EMPLOYE e, RENS_EMP r
                        where e.Num_Emp = r.Num_Emp
                        and r.Fonction = 'sous-directeur'
                        and e.Nom = 'Durant')
and (r.Periode_fin > (select min(r.Periode_debut)
                    from EMPLOYE e, RENS_EMP r
                    where e.Num_Emp = r.Num_Emp
                    and r.Fonction = 'sous-directeur'
                    and e.Nom = 'Durant')
    or r.Periode_fin is null);
```

La solution correspondant à cette requête est donnée par la Figure VIII-6.

NOM	PRENOM
Thill	Yves

Figure VIII-6 - Solution requête 2

3. Quand a été engagé chaque personne exerçant la fonction de sous-directeur ?

La question demande le nom de chaque personne qui a exercé la fonction de sous-directeur et à quel moment celle-ci a commencé à exercer cette fonction.

La requête correspondant à cette question est la suivante :

```
select e.nom, r.Periode_debut
from EMPLOYE e, RENS_EMP r
where e.Num_Emp = r.Num_Emp
and r.Fonction = 'sous-directeur'
and r.Periode_debut in ( select min(Periode_debut)
                        from RENS_EMP
                        where Fonction = 'sous-directeur'
                        group by Num_Emp);
```

La réponse donnée par la requête est présentée dans la Figure VIII-7.

NOM	PERIODE-DEBUT
Durant	14-04-1989
Leroy	30-05-1995
Guillaume	13-11-1995
Thill	16-05-1993

Figure VIII-7 - Solution requête 3

4. Qui est passé de la fonction d'ouvrier à la fonction d'employé ? (combien de temps ont-ils exercé la fonction d'ouvrier du département ?)

Cette question demande les noms des employés qui ont exercé la fonction d'ouvrier et d'employé ainsi que la période pendant laquelle ils ont été ouvriers. Nous avons fait l'hypothèse qu'une personne exerçant la fonction d'ouvrier ne peut évoluer que vers la fonction d'employé par la suite.

La requête correspondant à cette question est la suivante :

```
select e.nom, r.Periode_debut, s.Periode_fin
from EMPLOYE e, RENS_EMP r, RENS_EMP s
where e.Num_Emp = r.Num_Emp
and e.Num_Emp = s.Num_Emp
```

```

and r.Fonction = 'ouvrier'
and s.Fonction = 'ouvrier'
and r.Periode_debut in (select min(Periode_debut)
                        from RENS_EMP
                        where Fonction = 'ouvrier'
                        group by Num_Emp)
and s.Periode_fin in ( select min(Periode_debut)
                      from RENS_EMP
                      where Fonction = 'employe'
                      group by Num_Emp) ;

```

La réponse donnée par la requête est présentée dans la Figure VIII-8.

NOM	PERIODE-DEBUT	PERIODE-FIN
Durant	01-06-1972	08-04-1979

Figure VIII-8 - Solution requête 4

5. Quels sont les employés qui exercent actuellement la fonction de sous-directeur ?

Pour résoudre cette query, il faut sélectionner les employés tels que fonction = sous-directeur et Periode-fin = NULL.

La query correspondant à cette question est la suivante :

```

select e.nom, e.prenom
from EMPLOYE e, RENS_EMP r
where e.Num_Emp = r.Num_Emp
and r.Fonction = 'sous-directeur'
and r.Periode_fin is null

```

La solution correspondant à cette query est donnée par la Figure VIII-9.

NOM	PRENOM
Leroy	Pierre
Guillaume	Jean
Thill	Yves

Figure VIII-9 - Solution requête 5

6. Quel était la fonction de Durant le 01-01-1982 ?

La query qui correspond à cette question est donné par :

```
select r.Fonction
from EMPLOYE e, RENS_EMP r
where e.Num_Emp = r.Num_Emp
and e.Nom = 'Durant'
and r.Periode_debut <= 01-01-1982
and r.Periode_fin > 01-01-1982
```

La solution de cette query est :

FONCTION
employé

Figure VIII-10 - Solution requête 7

7. Quel est l'évolution de la carrière de Durant ?

Nous allons regarder quelle fonction Durant a déjà assumé et pendant quelle période il a assumé la fonction.

Dans ce cas précis, nous demandons l'historisation de l'attribut *Fonction* pour un nom d'employé particulier. Nous ne pouvons pas trouver la réponse à cette question par une simple query. Nous devons exécuter une procédure pour trouver la solution de cette question. Nous utilisons deux tables intermédiaires appelées HISTORIQUE1 et HISTORIQUE2. Elles sont composées des colonnes *Fonction*, *Periode_debut* et *Periode_fin*. Nous supposons que toutes les tables sont placées dans une base de données appelée GESEMP.

La détermination de la solution se passe en trois étapes :

Etape 1

Au cours de cette étape, nous allons sélectionner la valeur des attributs *Fonction*, *Periode_debut* et *Periode_fin* de toutes les lignes concernant Durant et nous les insérons dans la table HISTORIQUE1. De plus, elles seront ordonnées par ordre croissant de *Periode_debut*.

Nous appliquons dès lors la requête suivante :

```
insert into HISTORIQUE1
select r.Fonction, r.Periode_debut, r.Periode_fin
from EMPLOYE e, RENS_EMP r
where e.Num_emp = r.Num_emp
and e.Nom = 'Durant'
order by r.Periode_debut
```

La solution de cette requête est donnée par la figure suivante :

HISTORIQUE1		
PERIODE_DEBUT	PERIODE_FIN	FONCTION
10-06-1972	08-04-1979	ouvrier
08-04-1979	19-09-1983	employe
19-09-1983	14-04-1989	employe
14-04-1989	29-12-1992	sous-directeur
29-12-1992	13-06-1993	sous-directeur
13-06-1993	15-04-1997	sous-directeur
15-04-1997	NULL	directeur

Figure VIII-11 - Table HISTORIQUE1

Etape 2

Dans cette seconde étape, nous devons fusionner les intervalles de validité contigus pour une même valeur de fonction. Vu qu'il est impossible de le faire par une simple requête, nous le présenterons de manière procédurale à l'aide des primitives d'accès et de mise à jour. Le résultat de la fusion sera placé dans la table HISTORIQUE2.

```

Open-DB (GESEMP)
if db-found then do begin
  get-first(HISTORIQUE1, HIST1)
  if record-found then do begin
    get-next(HISTORIQUE1,HIST2)
    while record-found do begin
      if (HIST1.Fonction = HIST2.Fonction) then do begin
        HIST1.Periode_fin = HIST2.Periode_fin
        get-next(HISTORIQUE1, HIST2)
      end
    else do begin
      HIST1.Periode_fin = HIST2.Periode_debut
      create(HISTORIQUE2, HIST1)
      HIST1.Fonction = HIST2.Fonction
      HIST1.Periode_debut = HIST2.Periode_debut
      HIST1.Periode_fin = HIST2.Periode_fin
      get-next(HISTORIQUE1, HIST2)
    end
  end
  create(HISTORIQUE2, HIST1)
end
close-DB(GESEMP)
end

```

La solution qui devrait être donnée pour cette deuxième étape est :

HISTORIQUE2		
FONCTION	PERIODE-DEBUT	PERIODE-FIN
ouvrier	10-06-1972	08-04-1979
employé	08-04-1979	14-04-1989
sous-directeur	14-04-1989	15-04-1997
directeur	15-04-1997	NULL

Figure VIII-12 - Table HISTORIQUE2

Etape 3

Au cours de cette troisième étape, nous affichons le résultat trouvé dans la table HISTORIQUE2 (cfr. Figure VIII-12). La requête est la suivante :

```
select *
from HISTORIQUE2
```

B. Deuxième étude de cas

Nous appliquons également dans cette deuxième étude de cas le premier plan de transformation. Nous donnons au second point quelques exemples de queries qui sont susceptibles d'être posées.

Dans cette deuxième étude de cas, nous considérons le schéma conceptuel constitué des types d'entité FACULTE, ENSEIGNANT et COURS ainsi que de deux types d'association temporels De* et Titulaire* qui mettent en correspondance respectivement FACULTE - ENSEIGNANT et ENSEIGNANT - COURS.

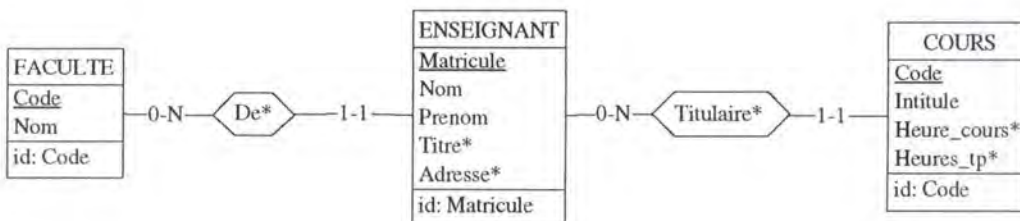
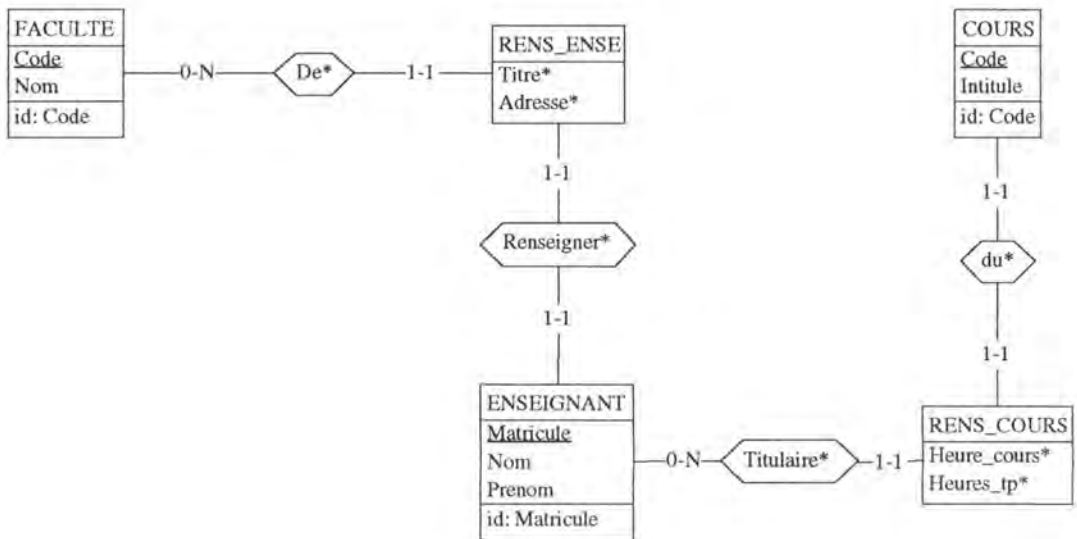


Figure VIII-13 - Deuxième étude de cas

1. Application du premier plan de transformation

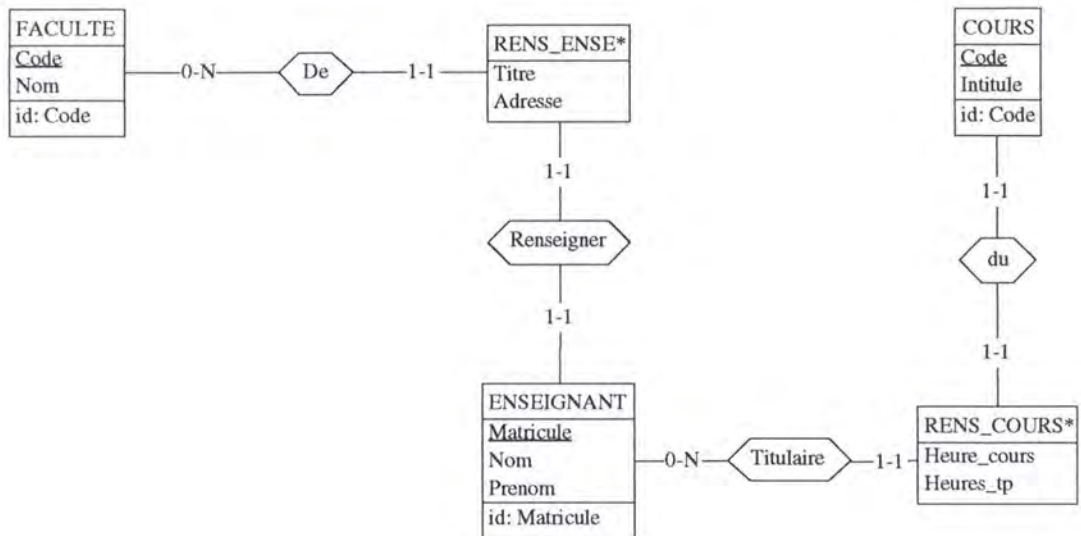
1. Types d'entité contenant plusieurs attributs temporels et au moins un attribut non temporel.

Dans cette première étape, nous recherchons ces types d'entité qui possèdent plusieurs attributs temporels sans qu'ils le soient tous. Nous voyons que les types d'entité ENSEIGNANT et COURS font partie de ce cas. Nous appliquons donc ECLAT-TE et nous obtenons le schéma suivant.



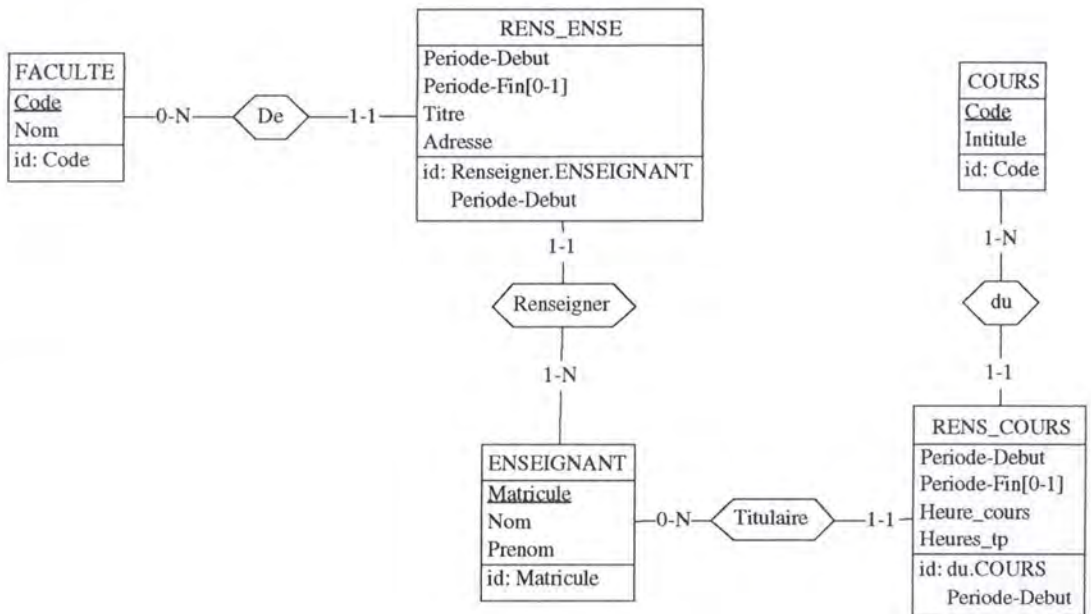
2. Types d'entité contenant uniquement des attributs temporels.

Nous recherchons maintenant les types d'entité caractérisés uniquement par des attributs temporels. L'étape précédente a fait apparaître dans le schéma deux types d'entité qui correspondent à ce cas : RENS_ENSE et RENS_COURS. Nous les transformons en type d'entité temporel grâce à TEMP_TE.



3. Types d'entité temporels.

Nous voyons dans le schéma résultant de l'étape précédente deux types d'entité temporels : RENS_ENSE et RENS_COURS. Nous les transformons en types d'entité non temporels et le résultat est :

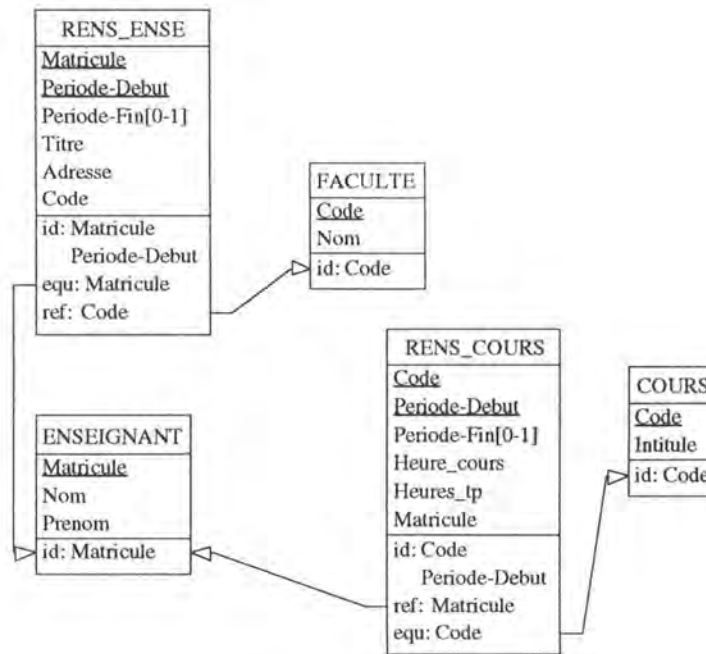


5. Types d'association temporels.

Vu qu'il n'y a plus de types d'association temporels dans le schéma, nous passons cette étape.

6. Types d'association non temporels et fonctionnels

Nous transformons en attributs de référence tous les types d'association fonctionnels lorsqu'ils n'associent pas deux types d'entité temporels. Sinon, nous les transformons en attributs de référence temporels.



+ Contraintes additionnelles

2. Exemples de query

Dans ce paragraphe, nous allons donner quelques exemples de queries.

1. **Quels sont le nom et le prénom de chaque enseignant travaillant actuellement à la faculté d'informatique ?**

La requête correspondant à cette question est la suivante :

```
select distinct e.Nom, e.Prenom
from FACULTE f, RENS_ENSE r, ENSEIGNANT e
where f.Code = r.Code
and e.Matricule = r.Matricule
and f.Nom = 'Informatique'
and r.Periode_fin is null
```

2. **Quels sont le nom et le prénom des personnes qui ont été des titulaires du cours de 'Mathématique' entre le 01-09-1972 et 01-09-1990 ?**

```
select distinct e.Nom, e.Prenom
from ENSEIGNANT e, RENS_COURS r, COURS c
where e.Matricule = r.Matricule
      and r.Code = c.Code
      and c.Intitulé = 'Mathématique'
      and (( r.Periode_fin > 01-01-1972 and r.Periode_fin < 01-01-1990)
           or ( r.Periode_debut > 01-01-1972 and r.Periode_debut < 01-01-1990)
           or ( r.Periode_debut < 01-01-1972 and ( r.Periode_fin >01-01-1990
           or r.Periode_fin is null)));
```

IX. Conclusion

L'objectif de ce mémoire était de se définir un modèle conceptuel temporel, de considérer un modèle relationnel, puis de développer un plan qui transforme un schéma temporel en un schéma relationnel correct.

Dans ce travail, nous avons considéré un modèle conceptuel de base qui est une restriction du modèle Entité-Association. Nous l'avons ensuite étendu pour obtenir le modèle temporel. Ce modèle temporel comprend quelques structures de données temporelles, à savoir les types d'entité temporels, les types d'association binaires one-to-many temporels et les attributs temporels. Nous avons ensuite défini des transformations qui permettent d'exprimer l'aspect temporel de ces structures dans le modèle relationnel. Elles utilisent les concepts d'attributs monovalués atomiques pour représenter le plus simplement possible le temps. Enfin, nous avons pu développer un plan qui utilise ces transformations et qui permet d'exprimer la dimension temporelle dans les schémas relationnels.

Les principales difficultés résidaient dans la transformation des structures de données temporelles en structures non temporelles conformes au modèle conceptuel de base. Une autre difficulté importante s'est présentée lorsque nous avons exprimé quelques queries sur un exemple car l'introduction du temps complique énormément celles-ci.

D'autre part, l'ensemble des transformations proposées n'est pas exhaustif, car nous avons simplifié le modèle conceptuel et nous n'avons considéré que quelques cas où les structures de données sont temporelles. Un travail futur envisagerait le modèle Entité-Association dans son ensemble afin de l'étendre en un modèle où toute structure de données supporterait l'aspect temporel. Ensuite, nous rechercherions toutes les transformations qui nous permettraient de représenter le temps dans le modèle Entité-Association. Enfin, nous pourrions développer un plan de transformation qui transforme tout schéma du modèle temporel en schéma relationnel.

Une autre extension de ce travail serait de considérer les types de temps comme distincts. En d'autres termes, il faudrait envisager le Valid time et le Transaction time différents.

Enfin, il serait également intéressant d'implémenter dans DB-Main des outils permettant de transformer automatiquement les schémas temporels en schémas relationnels équivalents.

X. Bibliographie

- [BODART1994] Bodart F., Pigneur Y.
Conception assistée des systèmes d'information - Méthode, modèles, outils (2th Edition).
Masson, 1994 .
- [DATE1995] Date C.,J.
An introduction to Database Systems - Volume I (6th Edition).
Addison-Wesley, 1995.
- [DELMAL1995] Delmal P.
SQL2, Point de la théorie de l'application.
De Boeck, 1995.
- [ELMASRI1990] Elmasri R.
Semantics of temporal data in an extended ER Model.
Proc. Intl. Conf. On E-R Approach, Lausanne, 1990, pp249-264.
- [LORENTZOS1995] Lorentzos N., Manolopoulos Y.
Functional requirements for historical and interval extensions to the relational model.
Data&Knowledge Engineering 17, pp59-86.
- [MC KENZIE1991] Mac Kenzie E., Snodgrass R.
Evaluation of Relational Algebras Incorporating the Time Dimension in Databases.
ACM Computing Survey, 1991, Vol 23, pp 501-543.
- [TANSEL1993] Tansel A., Clifford J., Gadia J.K., Jajodia S., Segev A., Snodgrass R.
Temporal Databases : Theory, Design and Implementation
The Benjamin/Cummings Publishing Company, 1993, pp1-207