

In Bases de données, Le relationnel: mythe et réalité, Editions Eyrolles 1986,
édité par A. Flory et M. Bouzeghoub, p.158-166

Bases de données Relationnelles et Connaissances

Bernard ESPINASSE , Robert MANTHA

Faculté des Sciences de l'Administration, Département des Systèmes d'Information Organisationnels,
Université LAVAL, QUEBEC, G1K 7P4, CANADA.

RESUME

Alors que les systèmes de gestion de bases de données relationnels sont de plus en plus utilisés pour la réalisation de systèmes d'information organisationnels informatisés, certaines faiblesses de ces outils se font sentir. Nous associons ces faiblesses des SGBD relationnels à des limites conceptuelles du modèle relationnel. En effet, le modèle relationnel, par sa vision essentiellement orienté mémorisation et structuration des données, nous apparaît incapable de prendre en compte l'aspect transformation des données. Ne prenant pas en compte cet aspect de transformation, le soin d'assurer l'intégrité de la base est laissé aux programmes d'application. Pour aborder cette transformation des données, il nous semble nécessaire de définir un cadre de réflexion qui nous permettra d'établir la nature des contraintes associées à ces transformations. Pour outrepasser ces limites, les systèmes relationnels doivent incorporer davantage de connaissance de façon explicite. Depuis une dizaine d'années, des progrès importants ont été faits dans le domaine de l'intelligence artificielle, notamment concernant les systèmes à base de connaissances. Ce nouveau type d'outil ouvrent de nouvelles voies prometteuses pour la conception de SGBD relationnels.

1 - INTRODUCTION

Les systèmes de gestion de bases de données relationnels ont un impact grandissant dans les milieux informatiques. Un nombre croissant d'entreprises adoptent le relationnel comme étant la solution à leurs problèmes de développement de systèmes pour satisfaire les demandes toujours plus nombreuses des divers acteurs dans leur organisation. Malheureusement, les SGBDs relationnels présentent un certain nombre de faiblesses bien souvent escamotées dans la littérature professionnelle.

2 - FORCES ET FAIBLESSES DES SGBD RELATIONNELS

Dans le développement de systèmes d'information, les systèmes de gestion de bases de données (SGBD) relationnels sont de plus en plus populaires. Ces derniers présentent un certain nombre de forces qui sont largement exploitées commercialement, citons les principales :

- Les SGBDs relationnels permettent aux usagers de mieux comprendre la nature et la structure des données contenues dans la base. Ceci tient principalement du fait que le concept de tableau (relation) est facile à comprendre.

Certains SGBDs relationnels permettent la définition de "vues" ou sous-schémas de la base qui peuvent représenter plus fidèlement les données qui sont d'un intérêt pour un ou un groupe d'usagers.

- Les SGBDs relationnels facilitent de manipulation des données notamment les requêtes et la production de rapports de gestion à l'aide de langages non-procéduraux du type "calcul relationnel" (SQL en est un exemple qui devient progressivement un standard) et du type "algèbre relationnel". Cette facilité de définir des requêtes à la base signifie qu'un nombre grandissant d'utilisateurs peuvent eux-même extraire l'information dont ils ont besoin, diminuant ainsi les pressions auprès du service informatique et augmentant leur satisfaction et leur autonomie.

- Les SGBDs relationnels facilitent l'évolution de la base de données en permettant l'ajout de nouveaux tableaux (relations) à la base comportant de nouvelles données et/ou de nouveaux liens entre les données sans qu'il soit nécessaire de recompiler le schéma de la base.

Quoique ces caractéristiques soient intéressantes, elles n'en restent pas moins insuffisantes. Aussi de nombreux SGBDs relationnels commercialisés disposent en plus d'outils complémentaires. Ceux-ci permettent de générer plus facilement des écrans afin de structurer l'interaction de l'utilisateur avec la base de données, d'outils permettant de générer des rapports à la suite de requêtes et des langages dits de "quatrième génération" comportant des instructions pour accéder aux données de la base et faire des calculs arithmétiques et/ou logiques. Ces langages disposent de structures de contrôle semblables mais souvent moins complètes que celles que l'on retrouve dans les langages évolués (ou de "troisième génération").

Ces types d'outils disponibles dans la plupart des SGBD relationnels, générateurs d'écran, dictionnaires de données intégrés et langages de quatrième génération permettent d'accélérer le développement d'applications consistant principalement de requêtes. Cependant, ces outils ne nous paraissent pas faciliter de façon adéquate le développement d'applications liées à des transformations de données consistant principalement à traiter des transactions et effectuer des mises-à-jour.

Nous n'appuyons pas notre argument sur le fait que le traitement de transactions à l'aide de SGBDs relationnels soit souvent moins "performant" (ce qui peut se défendre) mais sur la constatation que les développeurs d'applications (analystes et programmeurs) doivent faire de gros efforts pour assurer l'intégrité de la base de données.

On peut ainsi constater qu'une grande partie des instructions de programmes d'application écrits en langage

de quatrième génération associés à des écrans de saisie et à des transactions de mises-à-jour, ont pour objet d'assurer l'intégrité de la base de données; par exemple de vérifier que les données saisies ne violent pas les contraintes établies lors de l'analyse et la conception de l'application. Ces contraintes peuvent être de différents types et liées notamment :

- au type des données saisies
- à l'étendue des valeurs admissibles
- à des contraintes associées au modèle de données (cardinalités, dépendances fonctionnelles intra et inter relations, etc.)
- à des règles de gestion régissant les modifications aux données

Les SGBDs actuellement commercialisés supportent que très rarement de telles contraintes. Les plus performants sont capables de tenir compte de contraintes des deux premiers types, très rare sont ceux qui tiennent compte de celles du troisième type et encore plus rare pour le quatrième type.

Nous associons ces faiblesses des SGBD relationnels à des limites conceptuelles du modèle relationnel. En effet, si le modèle relationnel, par sa vision essentiellement orienté mémorisation et structuration des données permet de traiter de façon satisfaisante les consultations à une base de données, il nous apparaît incapable de traiter correctement les mises-à-jour, ne pouvant prendre en compte l'aspect transformation des données. Ne prenant pas en compte cet aspect transformation de données, le soin d'assurer l'intégrité de la base est laissé aux programmes d'application. Pour aborder cette transformation des données, il nous semble nécessaire de définir un nouveau cadre conceptuel permettant d'établir la nature des contraintes associées à ces transformations.

3 - UN NOUVEAU CADRE CONCEPTUEL POUR LES SGBD RELATIONNELS

Le cadre conceptuel permettant d'aborder cet aspect transformation des données que nous proposons s'inspire du rapport ISO [van Griethuysen, 1982]. L'un des apports majeurs du groupe de travail ISO TC97/SC5/WG3 est la reconnaissance que le contenu et le schéma de la base de données doivent évoluer pour refléter les changements qui ont lieu dans l'univers du discours (i.e. l'organisation). Ces changements conduisent à des transformations au niveau des données pouvant être de natures différentes, notamment:

- des entités peuvent apparaître ou disparaître de l'univers du discours
- les liens entre les entités peuvent changer
- l'état d'une entité peut changer (modification des propriétés)
- le champ d'intérêt du schéma peut varier par rapport au discours
- la représentation des entités peut varier, etc.

Tous ces changements sont régis par des contraintes ou des règles établies lors de l'analyse et la conception du

système d'information. L'un des principes fondamentaux établis par les membres du groupe est le "100% principe" qui décrète que tous les aspects statiques et dynamiques (i.e. toutes les règles, lois, etc.) de l'univers du discours doivent être inscrits dans le schéma conceptuel. Le système d'information ne peut être tenu responsable de ne pas avoir rencontré les contraintes ou les règles décrites ailleurs, en particulier celles qui se trouvent dans les programmes d'application.

L'interaction entre l'environnement et le système d'information se fait par l'entremise d'un processeur d'information [Tardieu et al., 1983] qui traite des messages (commandes) provenant de l'environnement. Le processeur d'information refusera une commande si elle ne peut être effectuée selon les règles de comportement établies dans le schéma. Le rapport ISO établi qu'il y a deux types de règles régissant le comportement du système: les règles qui concernent les interactions entre l'environnement et le système et les règles qui concernent les manipulations du schéma et de la base d'informations.

Deux types de moniteurs permettent de gérer le comportement du système; le moniteur dynamique qui s'occupe de l'interaction de l'environnement et du système d'information et le moniteur transitionnel qui s'occupe d'effectuer les opérations sur la base [Tardieu et al. 81]. Les deux moniteurs doivent s'assurer qu'un certain nombre de règles de transformation soient respectées. Notons que si la distinction entre les règles associées au moniteur dynamique et celles associées au moniteur de transition n'est pas facile à faire, ces moniteurs nous apparaissent essentiels au point que des mécanismes de support permettant d'assurer leurs fonctions devraient être inclus dans les outils de développements relationnels.

En fait de tels mécanismes de support ont déjà été conçus pour assurer une indépendance entre les données et les traitements dont le bien fondé n'est plus à démontrer. Les moniteurs transitionnel et dynamique nous semble à leur tour permettre une certaine indépendance entre les règles de transformation de données et les programmes d'application. Cette indépendance nous apparaît souhaitable pour obtenir une meilleure intégrité, une économie de code (redondance) et permettre une évolution continue de ces règles de transformation par rapport à l'univers du discours.

L'indépendance des données des traitements se réalise dans un dictionnaire et des utilitaires opérant sur ce dictionnaire. Nous nous proposons d'adopter une solution analogue en séparant les moniteurs des règles de transformation, règles qui pourraient s'exprimer de façon déclarative au moyen de règles de production telles que définies en intelligence artificielle. Ce pose alors le problème de la représentation de ces règles de transformation en tant que connaissances liées à la base données.

4 - CONNAISSANCES LIÉES AUX TRANSFORMATIONS DES DONNÉES

Les connaissances liées aux transformations des données concernent les règles et contraintes permettant d'assurer que la base de données demeure intègre par rapport à l'univers du discours [Wiederhold, 1984]. Ces connaissances sont de différentes nature, on peut distinguer entre autre:

- Les connaissances structurelles : Ces connaissances sont surtout mis en évidence lors de la conception même de la base de données. Elles concernent des contraintes d'intégrité liées aux données, citons parmi ces contraintes:

- les contraintes de références ou contraintes d'intégrité syntaxiques qui lient les occurrences référées d'un record à des occurrences référantes d'une autre record. Par exemple, soit les deux records:

- Employé (nom, ..., #département,...)
- Département (#département,).

On dit que l'attribut #département d'une occurrence d'Employé doit référer à une occurrence de Département dont l'attribut a la même valeur.

A cet aspect statique sont associées des règles dynamiques d'insertion-destruction:

- l'insertion d'une occurrence référante est interdite s'il n'existe pas dans la base d'occurrence référée
- la destruction d'une occurrence référée est interdite s'il y a présence d'occurrence référantes.

Ces règles peuvent être plus nuancées. On notera que ces contraintes de références sont déjà traitées dans de nombreux systèmes de gestion de bases de données de type réseau (voir les recommandations Codasyl).

- Les connaissances du domaine de définition des données: Elles donne lieu à des contraintes d'intégrité sémantiques. Elles concernent d'une part le type de la donnée (entier, alphanumérique, masculin /féminin,.....) et l'ensemble des valeurs qu'elle peut prendre (age compris entre 1 et 150,...).

Elle peuvent aussi avoir un aspect plus dynamique en précisant un certain nombre de restrictions dans les changements possibles de certaines données, (par exemple valeurs du statut marital).

- Les connaissances spécifiques à une application: Un large corps de connaissances relatives à une application particulière et à une activité de l'organisation peuvent être mises en relation étroite avec une base de données, ces connaissances peuvent être contenues dans un programme, ou sous forme déclarative dans une base de règles (règles de gestion).

L'explicitation de ces connaissances nous conduit à aborder la conception des outils relationnels sous un angle différent, utilisant les concepts et outils de l'intelligence artificielle. Nous présentons deux exemples d'une telle utilisation, relatives aux langages de quatrième génération et concernant la gestion des transactions, le contrôles d'intégrité et le développement.

5 - LA GESTION DES TRANSACTIONS ET CONTROLES D'INTEGRITE

Les changements d'états d'une base de données entraînés par des transactions de mise-à-jour doivent respecter les contraintes d'intégrité que nous avons vues précédemment. L'utilisation de plus en plus grande de langages de quatrième génération en développement d'applications par

les usagers eux-même, pose plus encore le problème des contrôles d'intégrité. Actuellement, les SGBD les plus performants sont capables d'emmagasiner et d'assurer quelques contraintes d'intégrité, par exemple la création et la suppression de set dans certains systèmes de type réseau. Les spécifications de contraintes sont dans ce cas incorporées dans le schéma de la base.

Un contrôle plus puissant, plus intelligent, peut être réalisé par l'introduction de connaissance dans les systèmes de gestion de base de données relationnels. Cette connaissance peut être introduite sous forme de clauses Prolog [Vassiliou et al., 1984] ou de règles de production [Stonebraker 1984]. Le traitement de transactions d'usager ne conduisant pas à un état de la base parfaitement intégré, pourrait alors être envisagé selon différentes alternatives:

- soit le SGBD rejette la transaction effectuée par l'usager
- soit accepte cette dernière sans pour autant effectuer de changement d'état à la base de données. Le système attend pour les effectuer que l'usager ait fait d'autres transactions qui assureront l'intégrité
- enfin, le SGBD génère et déclenche lui même les transactions qui laisseront la base dans un état cohérent.

6 - LANGAGES DE QUATRIEME GENERATION ET DEVELOPPEMENT

Les langages de quatrième génération prennent une place de plus en plus grande dans le développement de systèmes d'information organisationnels. Sans rentrer dans une définition précise de ces langages, on distinguera d'une part les langages de requêtes (dont certains peuvent être non-procéduraux), facilitant l'interrogation de la base et les langages orientés développement (généralement procéduraux), permettant de réaliser des applications spécifiques autour d'une base de données. Pour notre propos nous ne nous intéresserons qu'à ces derniers liés à l'aspect transformation des données.

L'usage de langage de manipulation de données de très haut niveaux de spécification autour de bases de données a considérablement facilité le développement d'applications réalisées d'une part par les informaticiens et d'autre part par les utilisateurs eux-même. On peut cependant penser qu'une approche orientée connaissance appliquée aux langages de quatrième génération permettrait de réduire le temps et le coût de développement, de concevoir des applications dont la maintenance et l'évolution seraient plus faciles.

Par exemple, dans le domaine de la paie, les systèmes informatiques sont souvent très lourds à maintenir, du fait qu'ils doivent constamment s'adapter à des nouvelles réglementations. De bons paramètres ne suffisent pas, souvent d'importantes ressources humaines du département informatique sont requises pour assurer ces mises-à-jour qui affectent souvent le code des programmes.

Une approche orientée connaissance, dans une certaine mesure résoud ces problèmes: les réglementations seraient mémorisées sous formes de règles déclaratives que l'usager pourrait modifier aisément, un calcul de paie serait assuré par un moteur d'inférence qui mettrait en œuvre ces règles sur la base de données des employés.

7 - CONCLUSION

Les systèmes de gestion de bases de données relationnels présentent un certain nombre de faiblesses liées principalement à un aspect transformation des données. Pour outrepasser ces limites, les systèmes relationnels doivent incorporer davantage de connaissance de façon explicite. Depuis une dizaine d'années, des progrès importants ont été faits dans le domaine de l'intelligence artificielle, notamment concernant les systèmes à base de connaissances. Ce nouveau type de système ouvrent de nouvelles voies prometteuses pour la conception de nouveaux outils relationnels. Une réflexion profonde, que nous avons ici à peine esquissée, sur les connaissances liées à la gestion et l'exploitation des bases de données permettrait de définir de tels outils.

BIBLIOGRAPHIE

- TARDIEU H., HECKENROTH H., ESPINASSE B., "Modèles et outils pour la représentation de la cinématique dans un système d'information", Rapport de recherche INRIA n°310, janvier 1981.
- TARDIEU H., ROCHFELD A., COLLETTI R., "La méthode MERISE", Edition d'Organisation, 1983.
- STONEBRAKER.M., "Adding Semantic Knowledge to a Relational Database System", in "On Conceptual Modeling", Springer-Verlag, 1984.
- VAN GRIETHUYSEN, 1982, "Concepts and Terminology for The Conceptual Schema and the Information Base", ISO TC 97/SC5/WG3, mars 1982.
- VASSILIOU.Y., CLIFFORD.J., JARKE.M, "Access to spécifique declarative knowledge by expert systems; the impact of the logic programming", C.R.I.S., New York University, 1984 .
- WIEDERHOLD G., "Knowledge and Database Management", IEEE Software, January 1984.