


SOMMAIRE

SOMMAIRE.....	1
PRESENTATION	3
STRUCTURES DE DONNEES	4
1) NOTION DE DOMAINE	4
2) NOTION DE RELATION	5
3) NOTION D'ATTRIBUT	6
REGLES (OU CONTRAINTES) D'INTEGRITE	7
1) UNICITE DE CLE	7
2) INTEGRITE REFERENTIELLE	7
3) INTEGRITE D'ENTITE (OU DE RELATION)	8
ALGEBRE RELATIONNELLE.....	9
1) UNION	9
2) DIFFERENCE	10
3) PRODUIT CARTESIEN	10
4) PROJECTION	11
5) RESTRICTION	12
6) JOINTURE	13
7) INTERSECTION.....	14
NORMALISATION DES RELATIONS	15
1) REDONDANCE	15
2) ANOMALIES DE STOCKAGE	15
3) REGLES DE NORMALISATION	16
1ère forme normale	16
2ème forme normale	16
3ème forme normale	17
GESTION DES TRANSACTIONS	18
1) CONTROLE DE CONCURRENCE.....	18
2) FIABILITE DE TRANSACTION	18

 afpa ©	auteur	centre	formation	module	séq/item	type doc	millésime	page 2
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc

PRESENTATION

La séquence précédente vous a permis d'appréhender les principes de base définis par Codd et inhérents à tout SGBD relationnel.

Ce chapitre va présenter plus en détail ces différents concepts structuraux, permettant de modéliser les données sous forme de tables à deux dimensions. Il souligne les règles de cohérence qu'impose le modèle et introduit l'algèbre relationnelle, outil formel indispensable à la manipulation des relations. Il aborde également les contraintes de normalisation et les caractéristiques du modèle quant au contrôle de la concurrence, de la reprise sur panne et des performances.

afpa ©	auteur	centre	formation	module	séq/item	type doc	millésime	page 3
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc

STRUCTURES DE DONNEES

Rappelons tout d'abord comment peut être défini simplement le modèle relationnel. Une base de données relationnelle se présente comme un ensemble de relations, ou tables. Toute relation a un schéma qui décrit sa structure, et une extension, qui correspond à l'état de cette relation à un instant donné. Le schéma est composé d'un ensemble d'attributs (ou colonnes) dont un sous-ensemble constitue la clé. La clé permet d'identifier les autres attributs de la relation. Les valeurs des attributs sont atomiques, c'est-à-dire non décomposables, et appartiennent au même domaine.

Cependant, pour être capable de manipuler correctement les données d'une base relationnelle, des définitions plus précises et rigoureuses sont nécessaires. En fait, c'est à partir de la théorie des ensembles, base formelle sur laquelle repose le modèle relationnel, que se contruisent les différents concepts.

1) NOTION DE DOMAINE

Le premier concept à définir est celui qui permet de préciser les ensembles de départ, ou domaines.

Un domaine représente l'ensemble des valeurs admissibles pour un composant d'une relation; il est caractérisé par un nom.

Comme tout ensemble, un domaine peut être défini en extension (c'est-à-dire en donnant la liste exhaustive de toutes les valeurs qui le composent), ou en intension (en définissant une propriété caractéristique du domaine).

Exemple

il est identique de parler de l'ensemble {1,3,5,7,9} ou de l'ensemble {entiers impairs positifs et inférieurs à 10}.

Ce concept de domaine est primordial pour assurer un contrôle lors de la manipulation de la base : toute comparaison de valeurs n'appartenant pas au même domaine de définition doit se voir rejetée.

Exemple

on peut imaginer un système d'information contenant des noms de viticulteurs (type alphabétique - longueur 25 caractères) et les désignations des crus qu'ils produisent (type et longueur identiques). Un SGBD relationnel, appliquant parfaitement toutes les règles énoncées par Codd, devrait interdire une question du genre

" quels sont les viticulteurs produisant des crus ayant même nom qu'eux ?"

Cependant, malgré l'importance de cette notion, la plupart des SGBD relationnels commercialisés ne la gèrent pas !!

afpa ©	auteur	centre	formation	module	séq/item	type doc	millésime	page 4
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc

2) NOTION DE RELATION

A partir de deux ou plusieurs domaines (D_1, D_2, \dots, D_n), on sait obtenir leur produit cartésien. Il est constitué de l'ensemble des n-uplets ou tuples $\langle V_1, V_2, \dots, V_n \rangle$ où V_1 représente une valeur de D_1 , V_2 une valeur de D_2 , ..., V_n une valeur de D_n .

Exemple

si l'on considère : le domaine COULEUR = { rouge, vert, jaune }

et le domaine FORME = { cube, sphère, cône },

le produit cartésien des deux domaines est composé des neuf tuples

{ cube rouge, cube vert, cube jaune, sphère rouge, sphère verte, ... }.

Au sens mathématique de Codd, la relation peut être définie comme un sous-ensemble, caractérisé par un nom, du produit cartésien d'une liste de domaines.

On utilise généralement une représentation commode sous forme d'une table à deux dimensions, où chaque ligne correspond à un tuple alors que chaque colonne correspond à un domaine du produit cartésien utilisé.

Dans la liste de domaines, un même domaine peut bien sûr apparaître plusieurs fois.

Exemple:

nous pouvons composer la relation suivante

Objets-à-construire	FORME	COULEUR
	cube	rouge
	cube	jaune
	sphère	rouge
	sphère	vert
	sphère	jaune
	cône	rouge

qui matérialise le fait que tous les objets ne sont pas construits dans toutes les couleurs.

3) NOTION D'ATTRIBUT

Le concept d'attribut a été introduit pour expliciter le rôle joué par un domaine dans une relation : il permet de distinguer les différentes colonnes, tout en permettant plusieurs colonnes du même domaine.


Exemple:

une relation VOL peut être constituée d'un n° de vol, d'une ville de départ, d'une ville d'arrivée..., ces deux attributs appartenant au même domaine "villes desservies par un aéroport".

Un attribut est une colonne d'une relation caractérisée par un nom.

Remarque

les tuples correspondent à un ensemble de valeurs prises par les attributs d'une table pour représenter un objet, ou un lien entre les objets du monde réel. Pour un objet donné, et à un instant donné, nous pouvons ne pas disposer de la totalité des informations permettant la description de cet objet ou de ce lien. Autrement dit, les valeurs de quelques attributs peuvent avoir des valeurs "inconnues" : elles sont dites NULL

	auteur	centre	formation	module	séq/item	type doc	millésime	page 6
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc

REGLES (OU CONTRAINTES) D'INTEGRITE

Les règles d'intégrité sont les assertions que doivent vérifier toutes les données stockées dans une base. Certaines de ces contraintes sont qualifiées de "structurelles" : elles sont inhérentes au modèle, c'est-à-dire nécessaires à sa mise en oeuvre, à son bon fonctionnement. D'autres peuvent apparaître comme étant des règles de "comportement", propres au schéma particulier d'un système d'information, d'un type d'applications.

En fait, le modèle relationnel impose a priori une règle structurelle, l'unicité des clés. Il est commode, et courant, d'y ajouter deux types de règles d'intégrité supplémentaires - les contraintes de référence et les contraintes d'entité - ceci afin d'obtenir les règles d'intégrité minimum supportées par le modèle relationnel. Pour être parfaitement proche des recommandations de Codd, il conviendrait d'y adjoindre la contrainte d'intégrité de domaine; celle-ci concerne le contrôle syntaxique et sémantique d'une donnée quelconque en se référant au type de définition du domaine.

1) UNICITE DE CLE

Par définition, une relation est un ensemble de tuples. Un ensemble n'ayant pas d'élément en double, il ne peut donc exister deux fois le même tuple dans une relation. Afin d'identifier les tuples d'une relation sans en fournir toutes les valeurs, la notion de clé est utilisée.

La clé est l'ensemble d'attributs minimum dont la connaissance des valeurs permet d'identifier, de manière unique, un tuple de la relation considérée.

Toute relation doit posséder au moins une clé. Dans le cas où il en existe plusieurs, on en choisit généralement une de manière arbitraire, qui est appelée clé primaire (PRIMARY KEY). Il est à noter que la détermination d'une clé au sein de la relation nécessite une réflexion importante : il s'agit de prévoir toutes les extensions possibles de la relation et de s'assurer que l'ensemble des attributs retenus pour constituer la clé permettra toujours l'identification d'un tuple unique !

2) INTEGRITE REFERENTIELLE

Ce type de contrainte apparaît clairement dès lors que l'on réfléchit aux principes de conception d'un modèle relationnel. Celui-ci est utilisé pour conceptualiser des entités du monde réel (qui sont les objets ayant une existence propre), et des associations entre ces objets.

En terme de base de données, une entité correspond à un tuple dans une relation, qui comporte la clé de l'entité (identifiant) et ses caractéristiques (sous forme d'attributs). Une association est également modélisée par une relation : celle-ci doit comporter les clés des entités participantes (pour assurer son unicité) ainsi que les caractéristiques propres à l'association.

afpa ©	auteur	centre	formation	module	séq/item	type doc	millésime	page 7
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc

On distingue en fait deux catégories de relations : les relations indépendantes ou statiques (matérialisant des entités) et les relations dépendantes ou dynamiques (matérialisant des associations) . Cette dépendance est caractérisée par la présence de clés étrangères (FOREIGN KEY), c'est-à-dire d'attributs qui sont clés primaires dans d'autres relations.

La contrainte de référence est une contrainte d'intégrité qui, portant sur une relation R1, consiste à imposer que la valeur d'un groupe d'attributs de R1 apparaisse comme valeur de clé primaire dans une (ou plusieurs) autre (s) relation (s).

Exemple

Considérons un système d'information partielle concernant la gestion commerciale d'une entreprise. On peut imaginer les relations suivantes :

CLIENT [n° client, références bancaires, adresse de livraison,...]

PRODUIT [désignation, code TVA, n° produit, unité de conditionnement,...]

et la relation COMMANDE-ENREGISTREE

[n° client, n° produit, quantité commandée...]

où les attributs n° client et n° produit sont des clés étrangères.

Cette intégrité référentielle doit induire certains contrôles automatiques de la part d'un SGBD relationnel. Ainsi, on ne peut pas mémoriser, dans la relation COMMANDE_ENREGISTREE, de tuple comportant un n° client inexistant dans la relation CLIENT ou un n° produit inconnu dans la relation PRODUIT.

3) INTEGRITE D'ENTITE (OU DE RELATION)

Lors de l'insertion de tuples dans une relation, il arrive fréquemment qu'un attribut soit inconnu, ou vide de sens, sous certaines conditions; on est alors conduit à introduire dans la relation une valeur conventionnelle, dite NULL (à ne pas confondre avec l'initialisation à zéro ou à blanc d'un attribut numérique ou alphabétique !).

Il est bien évident que tout attribut dans une relation ne peut pas prendre une valeur NULL; en effet, l'existence d'une clé unique, par exemple, impose la connaissance de la clé pour pouvoir vérifier que cette valeur de clé n'existe pas déjà.

La contrainte d'entité (ou de relation) est une contrainte d'intégrité imposant que tout attribut qui participe à une clé primaire soit non NULL.

ALGEBRE RELATIONNELLE

Inventée par Codd, elle consiste en une collection d'opérations formelles qui agissent sur des relations et produisent de nouvelles relations comme résultats. Elle constitue la partie manipulation élémentaire associée au modèle relationnel, elle est considérée comme en étant l'un de ses avantages et l'une de ses forces principales. Elle comporte sept opérations de base.

1) UNION

C'est l'opération classique de la théorie des ensembles, adaptée exclusivement aux relations de même schéma.

On appelle UNION l'opération portant sur deux relations R1 et R2 de même schéma, et consistant à construire une nouvelle relation R3, de même schéma, ayant pour tuples ceux appartenant à R1, ou à R2, ou aux deux à la fois.

Exemple

Soient les deux relations VINS1 et VINS2 représentées ci-dessous.

VINS1

N° de vin	Cru	Millésime	Degré
100	Volnay	1978	12.5
110	Chablis	1979	12.0
120	Sancerre	1980	12.5
130	Tokay	1980	12.5

VINS2

N° de vin	Cru	Millésime	Degré
130	Tokay	1980	12.5
140	Chenas	1981	12.7
150	Volnay	1978	12.5

L'union de VINS1 et VINS2 donne la relation VINS

N° de vin	Cru	Millésime	Degré
100	Volnay	1978	12.5
110	Chablis	1979	12.0
120	Sancerre	1980	12.5
130	Tokay	1980	12.5
140	Chenas	1981	12.7
150	Volnay	1978	12.5

2) DIFFERENCE

Cette opération est également adaptée exclusivement aux relations de même schéma.

On appelle DIFFERENCE l'opération portant sur deux relations R1 et R2 de même schéma, et consistant à construire une nouvelle relation R3, de même schéma, ayant pour tuples ceux appartenant à R1 sans appartenir à R2.

Il apparaît clairement que, dans cette opération, l'ordre des opérandes est important!
La différence est un opérateur non commutatif.

Exemple

La différence de VINS1 et VINS2 donne la relation VINS

N° de vin	Cru	Millésime	Degré
100	Volnay	1978	12.5
110	Chablis	1979	12.0
120	Sancerre	1980	12.5

3) PRODUIT CARTESIEN

C'est une opération ensembliste qui ne nécessite pas des relations initiales de même schéma.

On appelle PRODUIT CARTESIEN l'opération portant sur deux relations R1 et R2, ayant pour schéma la juxtaposition des schémas de R1 et de R2, et pour tuples toutes les combinaisons possibles des tuples des opérations opérandes.

Exemple

Soient les deux relations CRUS et ANNEE représentées ci-dessous.

CRUS

Cru	Région	Pays
Chenas	Beaujolais	France
Volnay	Bourgogne	France
Chanturgues	Auvergne	France

ANNEE

Millésime	Qualité
1979	Bonne
1980	Passable

Le produit cartésien de CRUS et ANNEE donne la relation VIGNOBLE

Cru	Région	Pays	Millésime	Qualité
Chenas	Beaujolais	France	1979	Bonne
Volnay	Bourgogne	France	1979	Bonne
Chanturgues	Auvergne	France	1979	Bonne
Chenas	Beaujolais	France	1980	Passable
Volnay	Bourgogne	France	1980	Passable
Chanturgues	Auvergne	France	1980	Passable

4) PROJECTION

C'est une opération unaire (portant sur une seule relation) permettant de supprimer des attributs (colonnes) d'une relation.

On appelle PROJECTION l'opération portant sur une relation R1, consistant à composer une relation R2, en enlevant à la relation initiale tous les attributs non mentionnés en opérands, et en éliminant les tuples en double (conservés une seule fois).

Exemple

La projection de la relation VINS consistant à créer une relation ANNEE ne comportant que les attributs Millésime et Qualité fournit le résultat :

VINS

N° de vin	Cru	Millésime	Qualité	Degré
100	Volnay	1979	Bonne	12.7
110	Chablis	1980	Passable	11.8
120	Tokay	1981	Excellente	12.1
130	Chenas	1979	Bonne	12.0
140	Volnay	1980	Passable	11.9

ANNEE

Millésime	Qualité
1979	Bonne
1981	Excellente
1980	Passable

5) RESTRICTION

Il s'agit également d'une opération unaire qui produit une nouvelle relation en ôtant des tuples à la relation opérande (suppression de lignes).

On appelle RESTRICTION l'opération portant sur une relation R1, produisant une relation R2 de même schéma, mais comportant les seuls tuples qui vérifient la (les) condition (s) précisée (s) en opérande (s).

Exemple

La restriction de la relation VINS consistant à créer une relation BONS_VINS par la condition Qualité = "Bonne" fournit le résultat :

VINS

N° de vin	Cru	Millésime	Qualité	Degré
100	Volnay	1979	Bonne	12.7
110	Chablis	1980	Passable	11.8
120	Tokay	1981	Excellente	12.1
130	Chenas	1979	Bonne	12.0
140	Volnay	1980	Passable	11.9

BONS_VINS

N° de vin	Cru	Millésime	Qualité	Degré
100	Volnay	1979	Bonne	12.7
130	Chenas	1979	Bonne	12.0

6) JOINTURE

Elle représente l'une des opérations essentielles de l'algèbre relationnelle, sans doute la plus difficile à réaliser dans les systèmes. C'est également la plus pénalisante au niveau des performances et des résultats, si son utilisation n'est pas parfaitement maîtrisée !

On appelle JOINTURE l'opération consistant à rapprocher selon une condition les tuples de deux relations R1 et R2 afin de former une nouvelle relation R3 qui contient l'ensemble de tous les tuples obtenus en concaténant un tuple de R1 et un tuple de R2 vérifiant la condition de rapprochement.

Les relations peuvent être rapprochées par égalité sur un attribut (équi-jointure) ou par un test de comparaison par inégalité type <, >, ... (thêta-jointure).

Exemple

Soient deux relations VINS et LOCALISATION définies ci-dessous. Déterminons la jointure de ces deux relations selon la condition Qualité > Qualité moyenne.

VINS

N° de vin	Cru	Millésime	Qualité
100	Chenas	1977	Bonne
200	Chenas	1980	Excellente
300	Chablis	1977	Bonne
400	Chablis	1978	Médiocre
500	Volnay	1980	Moyenne

LOCALISATION

Région	Cru	Qualité moyen
Beaujolais	Chenas	Bonne
Bourgogne	Chablis	Moyenne
Californie	Chablis	Médiocre

Résultat de jointure

N° de vin	Cru	Millésime	Qualité	Région	Qualité moy.
100	Chenas	1977	Bonne	Beaujolais	Bonne
200	Chenas	1980	Excellente	Beaujolais	Bonne
300	Chablis	1977	Bonne	Bourgogne	Moyenne
300	Chablis	1977	Bonne	Californie	Médiocre
400	Chablis	1978	Médiocre	Bourgogne	Moyenne
400	Chablis	1978	Médiocre	Californie	Médiocre

7) INTERSECTION

Cette opération peut être considérée comme redondante avec les opérations de base énoncées précédemment, en ce sens qu'il est possible de l'obtenir à partir de la différence.

On appelle INTERSECTION l'opération portant sur deux relations de même schéma R1 et R2, consistant à construire une nouvelle relation R3, de même schéma, ayant pour tuples ceux appartenant à la fois aux deux relations R1 et R2.

NORMALISATION DES RELATIONS

Le besoin de normalisation peut être illustré de façon simple par un exemple.

Supposons que l'on ait défini une relation AVION, dont la clé primaire est le n°_avion et comportant les attributs suivants : nom de l'avion, capacité et localisation physique.

N°_avion	Nom	Capacité	Localisation
100	Airbus	250	Toulouse
101	Airbus	250	Toulouse
102	Airbus	250	Paris
103	Caravelle	100	Toulouse
104	Boeing 707	150	Paris
105	Boeing 707	150	Paris

Supposons que, dans cette compagnie aérienne, il existe la contrainte suivante :

" tous les avions de même nom ont la même capacité".

A l'inverse, deux avions de nom différent peuvent bien entendu avoir la même capacité (dépendance fonctionnelle).

De ce fait, la relation AVION, telle que définie ci-dessus, provoque les problèmes suivants :

- redondance logique
- anomalies de stockage.

1) REDONDANCE

Dans notre exemple, le couple de valeurs (Airbus, 250) apparaît 3 fois.

Cette multiplication de données identiques pose, en premier lieu, des problèmes de volumes, auxquels se greffent ceux de cohérence.

2) ANOMALIES DE STOCKAGE

Il s'agit d'anomalies relatives aux opérations d'insertion, de suppression et de modification de la base de données.

Si la compagnie acquiert un Boeing 727 de 150 places, on ne peut introduire cette donnée nouvelle dans la relation qu'en lui affectant une nouvelle valeur de clé primaire. Sinon, l'insertion est impossible (anomalie d'insertion).

Supposons que la compagnie supprime de son parc l'avion n° 103. Si l'on détruit la ligne correspondante dans la relation, on perd un lien unique, à savoir qu'une Caravelle comporte 100 places (anomalie de suppression).

Supposons que la capacité des Airbus soit portée de 250 à 275. Si la modification est répercutée dans un seul tuple, alors la relation devient incohérente. A l'inverse, si la modification est bien répercutée à tous les endroits nécessaires, après "balayage" complet de la relation, alors le coût de la modification est très élevé (anomalie de modification).

3) REGLES DE NORMALISATION

La théorie sur la normalisation repose sur l'observation que certaines relations ont de meilleures propriétés dans un environnement de mise à jour, que d'autres relations équivalentes en terme de contenant les mêmes données. Cette théorie fournit un cadre rigoureux pour la définition du schéma relationnel.

La normalisation est basée sur un processus de décomposition réversible des relations de base, qui consiste à isoler, par une opération de projection, tout lien intra-relation. La réversibilité garantit que la relation de départ peut être retrouvée à tout moment par utilisation de l'opérateur de jointure, sans qu'aucune information n'ait été perdue.

Il paraît superflu de charger ce cours d'explications approfondies, ou de démonstrations mathématiques, que les personnes intéressées peuvent retrouver facilement dans les manuels de méthodologie traitant de ces contraintes.

Ce support propose simplement une présentation succincte des trois principales règles énoncées (on les qualifie de "formes normales").

1ère forme normale

Une relation est dite en 1ère forme normale, ou "normalisée", si aucun attribut qui la compose n'est lui-même une relation, c'est-à-dire si tout attribut est atomique (non décomposable).

2ème forme normale

Une relation est dite en 2ème forme normale si les attributs n'appartenant pas à la clé primaire sont en dépendance fonctionnelle totale avec la clé primaire (la connaissance d'une valeur de clé permet de déterminer de manière unique une ligne de la relation).

afpa ©	auteur	centre	formation	module	séq/item	type doc	millésime	page 16
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc

3ème forme normale

Une relation est dite en 3ème forme normale si chaque déterminant est une clé (primaire ou candidate).

GESTION DES TRANSACTIONS

La gestion de transactions s'inscrit dans un environnement multi-usagers (concurrent) en la présence de pannes du système ou des unités de stockage. La cohérence de la base peut être altérée par les opérations de mise à jour. Afin de garantir cette cohérence, les différentes mises à jour d'un utilisateur sont groupées dans une transaction qui devient l'unité de cohérence contrôlée par le système.

Informellement, une transaction est une unité d'exécution qui, appliquée à un état cohérent, restitue un état cohérent, mais modifié, de la base.

Une transaction est atomique : elle est soit exécutée entièrement (commise), soit pas exécutée du tout (annulée).

La gestion des transactions inclut deux aspects majeurs : contrôle de concurrence et fiabilité.

1) CONTROLE DE CONCURRENCE

Son objectif est de garantir que l'exécution pseudo-parallèle de transactions multiples produit le même résultat que leur exécution séquentielle. Ainsi, le contrôle de concurrence rend l'accès multi-usager aux données transparent à chaque utilisateur.

La solution la plus employée est le verrouillage, dérivé de la méthode d'allocation des ressources dans les systèmes opératoires. Les ressources peuvent être allouées, ou désallouées, aux transactions.

Le problème d'un algorithme de verrouillage est l'occurrence d'étreinte fatale ou verrou mortel (dead lock). La solution générale de détection des verrous mortels génère l'abandon de certaines transactions.

2) FIABILITE DE TRANSACTION

Son objectif est de garantir que l'exécution de transactions s'effectue correctement en présence de pannes. Elle permet la reprise d'un mode de travail normal après différents types de pannes (transaction, système ou disque).

Son but est de restituer un état cohérent de la base après panne, tout en limitant l'assistance de l'utilisateur. Elle est fondée sur un principe unique : la duplication des données. Les données détériorées ou perdues lors d'une panne, doivent pouvoir être restituées à partir de copies.

Les structures de données les plus fréquemment employées sont les journaux (sur disque) et les archives périodiques (sur bande).

Un journal enregistre l'information concernant l'état des transactions et des données mises à jour multiples produit le même résultat que leur exécution séquentielle. Ainsi, le contrôle de concurrence rend l'accès multi-usager aux données transparent à chaque utilisateur.

afpa ©	auteur	centre	formation	module	séq/item	type doc	millésime	page 18
		Dijon				sup. form.	18/06/2008	Modele Relationnel.doc