LE MODÈLE RELATIONNEL

Introduction

Le modèle relationnel a été défini par E.F Codd dans les années 70. Les premiers systèmes de gestion de base de données (SGBD ou DBMS en anglais) bâtis sur ce modèle ont été SQL/DS et DB2 de IBM, d'où est né le langage de manipulation de bases relationnelles, SQL (Structured Query Language). Bien d'autres implémentations ont suivi et, actuellement, la plupart des SGBD commercialisés, aussi bien sur micros que sur grands systèmes, se réclament du modèle relationnel.

L'explication d'un tel succès tient dans la simplicité que ce modèle apporte, par rapport à ses prédécesseurs de type "hiérarchique", "réseau" ou Codasyl et C.J Date.

Simplicité tout d'abord pour l'utilisateur dans la conception, la définition, l'installation de la base de données.

- ✓ Simplicité de la structure des données :
    1. Une base relationnelle est composée de tables et est perçue par l'utilisateur comme un ensemble de tables et rien d'autre. Une table est un tableau à deux dimensions Dans une table, une ligne (ou tuple) correspond à un enregistrement et une colonne (un attribut) à un champ de cet enregistrement.
    2. Simplicité des opérateurs : Toute opération relationnelle (Algèbre Relationnelle ou Langage SQL) sur une table génère une nouvelle table Les opérateurs relationnels permettent de décrire le résultat que l'on veut obtenir sans avoir à décrire la procédure nécessaire pour arriver au résultat.
    - L'utilisateur disposera donc en relationnel d'un langage de manipulation des données simple, d'apprentissage facile

1. Le modèle relationnel

a) Concepts du modelé :

- Le modèle relationnel inclut des concepts pour la **description** de données, ainsi que des concepts pour la **manipulation** de données.
- Le modèle relationnel permet de représenter les données que l'on va gérer à l'aide d'un très petit nombre de concepts très simples :
- Les relations ou tables : des lignes et des colonnes

- Les domaines de valeurs : chaque case d'une table prend une unique valeur dans un domaine pré&défini.
- Les clés : il existe des cases dont les valeurs doivent être uniques et non nulles
  Les clés étrangères : il existe des cases qui doivent prendre une valeur existante dans les cases d'une autre table.

Une relation peut être définie en extension ou en intention.

Définition en *extension* = vision tabulaire (*tuple*, *attribut*, *domaine*).

Nom de relation Nom d'attribut Nom de domaine

Notation : $R(A_1{:}D_1, A_2{:}D_2, \ldots, A_n{:}D_n)$ ou $R\ \{A_1, A_2, \ldots, A_n\}$

$R \in (\ dom(A_1)\ \mathbf{x}\ dom(A_2)\ \mathbf{x}\ \ldots\ \mathbf{x}\ dom(A_n)\ )$; Une relation est un sous ensemble du produit cartésien des domaines de ses attributs.

Tuple : $t = <V_1, V_2, \ldots, V_n>$ où V_i = valeur $\in dom(A_i)$ ou valeur nulle.

Exemple :

NOPIECE	QUANTITE	NOMFOUR	ADRESSE
11	41	Ben Ali	11 , Rue des Rais - Alger
12	22	Amiden	44 , Rue de Ziroud - Constantine
10	15	Kessoul	33, Belle vue Oran
13	26	Hamid	122, Rue Didouche – Alger
12	43	Ben Ali	11 , Rue des Rais - Alger

La définition en *intention* d'une relation R repose sur trois éléments :

- Un ensemble d'attributs $\{A_1, A_2, \ldots, A_n\}$;
- L'affectation d'un domaine $dom(A_i)$ à chaque attribut A_i ;
- Un ensemble des contraintes d'intégrité formulées sur R.

Relation = représentation tabulaire (définition en extension).

Schéma relationnel = représentation par attributs et par contraintes (définition en intention) de plusieurs relations.

b) Représentations des entités et des associations

Clé primaire = clé d'une relation choisie parmi les clés candidates.

Domaine primaire = domaine sur lequel une clé primaire est définie.
Clé étrangère = attribut défini sur un domaine primaire et qui n'est pas clé primaire.

Les clés étrangères sont une forme de redondance obligatoire qui doit être contrôlée par le SGBD.

c) Problème de la redondance

[En dehors des clés étrangères].

Ex. La relation COMMANDE (NOPIECE, QUANTITE, NOMFOUR, ADRESSE).

COMMANDE

NOPIECE	QUANTITE	NOMFOUR	ADRESSE
11	41	Ben Ali	11 , Rue des Rais - Alger
12	22	Amiden	44 , Rue de Ziroud - Constantine
11	15	Kessoul	33, Belle vue Oran
13	26	Hamid	122, Rue Didouche – Alger
12	43	Ben Ali	11 , Rue des Rais - Alger

Schéma avec anomalies

Cette relation présente différentes anomalies.

- Anomalies de modification : si l'on souhaite mettre à jour l'adresse d'un fournisseur, il faut le faire pour tous les tuples concernés.
- Anomalies d'insertion : pour introduire le nom et l'adresse d'un fournisseur, il faut également fournir une valeur pour chacun des attributs NOPIECE et QUANTITE, ou introduire des valeurs nulles, ce qui pose d'autres problèmes.
- Anomalies de suppression : par exemple, la suppression de la commande de la pièce n° 11 fait perdre toute information concernant le fournisseur Kessoul.

d) Décomposition d'une relation

La cause des anomalies identifiées au c) est que la relation contient des attributs qui ne sont pas caractéristiques d'une même entité. Pour les supprimer, il faut décomposer la relation COMMANDE en deux autres relations :

PIECECOM (<u>NOPIECE</u>, <u>NOMFOUR</u>, QUANTITE)
FOURNISSEUR (<u>NOMFOUR</u>, ADRESSE)

PIECECOM

NOPIECE		QUANTITE
11	Ben Ali	44
12	Amiden	22
11	Kessoul	15
13	Hamid	26
11	Ben Ali	43

FOURNISSEUR

NOMFOUR	ADRESSE
Ben Al	11 , Rue des Rais - Alger
Amiden	44 , Rue de Ziroud - Constantine
Kessoul	33, Belle vue Oran

Une décomposition du schéma précédent

On appelle *décomposition* le remplacement d'une relation R par une collection de relations R_1, R_2, ..., R_n obtenues par des projections de R et telle que la relation résultant des jointures R_1 R_2 … R_n constitue un schéma relationnel équivalent à R.

Attention ! Des décompositions arbitraires peuvent entraîner des pertes de proprié- tés du schéma relationnel initial.

e) Problème des valeurs nulles

Diverses interprétations de la valeur nulle peuvent être faites :

- l'attribut ne s'applique pas au tuple créé ;
- la valeur de l'attribut n'est pas connue pour ce tuple ;
- la valeur est connue mais n'a pas encore été enregistrée.

La présence de valeurs nulles entraîne donc des difficultés pour certains traite- ments.

2. Les dépendances fonctionnelles (DF)

Notations : Soit X un ensemble d'attributs et x la valeur de X.
$X \cup Y$ sera noté X, Y.

a) Définition

Soit R(X, Y, Z) une relation où X, Y, Z sont des ensembles d'attributs, Z pouvant être vide.

Y dépend fonctionnellement de X (notation :$X \rightarrow Y$) si c'est toujours la même valeur de Y qui est associée à une valeur donnée de X dans la relation.

Autrement dit, si (x, y, z) et (x, y', z') sont deux tuples de R, alors y = y'.

Exemple : relation COMPAGNIE (VOL, AVION, PILOTE)

VOL → AVION
VOL → PILOTE
AVION → PILOTE

b) Propriétés des dépendances fonctionnelles

- **Réflexivité :** si $Y \subseteq X$ alors $X \rightarrow Y$.
- **Augmentation :** si $W \subseteq Z$ et $X \rightarrow Y$ alors $X, Z \rightarrow Y, W$.
- **Transitivité :** si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$.
- Pseudo-transitivité : si $X \rightarrow Y$ et $Y, Z \rightarrow T$ alors $X, Z \rightarrow T$.
- Union : si $X \rightarrow Y$ et $X \rightarrow Z$ alors $X \rightarrow Y, Z$.
- Décomposition : si $Z \subseteq Y$ et $X \rightarrow Y$ alors $X \rightarrow Z$.

c) Définitions complémentaires

La dépendance $X \rightarrow Y$ est *élémentaire* s'il n'existe pas $X' \subset X$ tel que $X' \rightarrow Y$ (il n'y a pas d'attributs superflus dans la partie gauche de la dépendance).

La dépendance $X \rightarrow Y$ est *directe* s 'il n'existe pas Z dans R distinct de X et Y tel que $X \rightarrow Z$ et $Z \rightarrow Y$ (la dépendance n'est pas obtenue par transitivité).

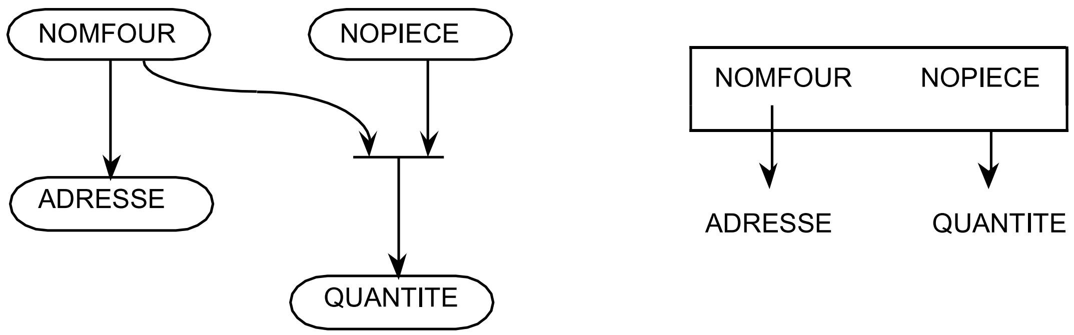
La dépendance $X \rightarrow Y$ est *triviale* si $Y - X$ est vide.

Une dépendance fonctionnelle est *simple* si elle ne comporte qu'un seul attribut en partie droite et si elle n'est pas triviale.

$X \rightarrow A_1, A_2, ..., A_n \Leftrightarrow \{ X \rightarrow A_1 ; X \rightarrow A_2 ; ... ; X \rightarrow A_n \}$
(en appliquant d'une façon itérative la propriété de décomposition).

Il est toujours possible de présenter les dépendances fonctionnelles sous forme simple.

Représentations sous forme de graphes :



Représentation par graphes des dépendances fonctionnelles d'une relation

d) Fermeture et couverture d'un ensemble de DF

On dira qu'un ensemble de dépendances D défini sur un ensemble de constituants X *implique logiquement* une dépendance d, ou que d est une *conséquence logique* de D, si l'ensemble de dépendances D étant vérifié sur X, alors d est aussi vérifiée sur X. On écrira $D \rightarrow d$.

La *fermeture* D^+ de D est obtenue en inférant jusqu'à saturation les propriétés des DF sur l'ensemble initial D de dépendances.

Couverture minimale :

*Dans un ensemble de dépendances fonctionnelles DF, il se peut qu'un certain nombre de ces dépendances se déduisent d'autres dépendances par application des propriétés des dépendances fonctionnelles. si on élimine toutes ces dépendances fonctionnelles on obtient un ensemble minimal de DF, cette ensemble est appelé **couverture minimale** des dépendances fonctionnelles* *exemple* *soit l'ensemble de DF: {A->B, B->C,A->C}, alors l'ensemble DF1:{A->B, B->C} forme une couverture minimale car aucune dépendance fonctionnelle n'est déductible de l'autre.* *d'un point de vue pratique, la couverture minimale nous servira dans la décomposition des relations sans perte d'information*

Couverture minimale ou *couverture irredondante* de D = sous-ensemble D° de D tel que pour toute dépendance d $\in$ D°, D°$-${d} n'implique pas d.

Propriétés de la couverture minimale D° de D :

- les dépendances de D° sont élémentaires et simples ;
- la fermeture de D° est égale à la fermeture de D ;
- il n'existe pas de partie stricte D' de D° dont la fermeture est égale à la fermeture de D.

Algorithme pour élaborer une couverture irredondante D° d'un ensemble D de dépendances :

```
G := D ;
Pour chaque f ∈ G faire
    Si G-{f} implique f alors
        G := G-{f} ;
    Finsi
Finpour;
D° := G ;
```

e) Déterminants et clés

Constituant = tout sous-ensemble de l'ensemble U des attributs d'une relation R.

Le constituant Y est *totalement dépendant* de X si la dépendance fonctionnelle X $\rightarrow$ Y est élémentaire.

Déterminant = tout constituant V tel qu'il existe un constituant totalement dépendant de V.

Clé = tout constituant X de R ($A_1, A_2, ..., A_n$) tel que :

- $X \rightarrow A_1, A_2, ..., A_n$;
- il n'existe pas $Y \subset X$ tel que $Y \rightarrow A_1, A_2, ..., A_n$.

Constituant clé = constituant qui appartient à l'une des clés candidates de R.

Superclé = tout constituant qui contient une clé de R.

3. Formes normales d'une relation

a) Première forme normale (1FN)

Une relation est en *première forme normale* si tout attribut ne peut prendre que des valeurs atomiques (n'est pas décomposable).

Exemples : Les relations ETUDIANT (NOM, PRENOM, MATIERES, AGE) et DEPARTE- MENT (NOM, ADRESSE, TELEPHONE) ne sont pas en 1FN si les attributs MATIERES et ADRESSE peuvent être du type [BD,GL,SE] ou [44 , Rue de Ziroud - Constantine].

b) Deuxième forme normale (2FN)

Une relation R (A, B, C, D, E) est en *deuxième forme normale* si et seulement si elle est en 1FN et tout constituant non clé est totalement dépendant de la clé de R (aucun des attributs C, D et E ne dépend d'une partie de la clé).

Autrement dit, il faut éviter la configuration suivante :

$A, B \rightarrow D, E$;

$B \rightarrow C$.

Exemples :

- La relation EMPLOYE (MATRICULE, NOM, PRENOM, AGE) est en 2FN.

- La relation COMMANDE (NOPIECE, QUANTITE, NOMFOUR, ADRESSE) n'est pas sous 2FN car on a :

NOPIECE, NOMFOUR $\rightarrow$ QUANTITE ;
NOMFOUR $\rightarrow$ ADRESSE.

La décomposition suivante conduit à deux relations en 2FN sans anomalies de mise à jour.

PIECE (NOPIECE, NOMFOUR, QUANTITE)
FOURNISSEUR (NOMFOUR, ADRESSE)

• La relation COMPAGNIE (VOL, AVION, PILOTE) avec les DF :

VOL $\rightarrow$ AVION,
AVION $\rightarrow$ PILOTE,
VOL $\rightarrow$ PILOTE,

est en 2FN avec des anomalies de mise à jour (il n'est pas possible d'introduire un nouvel avion sur un nouveau vol sans préciser le pilote correspondant). La 3FN résout ce problème.

c) Troisième forme normale (3FN)

Une relation R (A, B, C, D, E) est en *troisième forme normale* si et seulement si elle est en 2FN et s'il n'existe aucune dépendance transitive entre une clé et un des attri- buts non clé (toutes les DF sont directes). Autrement dit, R est en 3FN s'il n'existe au- cune dépendance entre deux attributs non clés. Comme R possède plusieurs clés, la défi- nition doit être vérifiée pour chaque clé.

Il faut donc éviter la configuration suivante :

A, B $\rightarrow$ C, D, E ;

C $\rightarrow$ E.

Exemple : La relation COMPAGNIE(VOL, AVION, PILOTE)

n'est pas en 3FN. La décomposition

R1 (VOL, AVION)
R2 (VOL, PILOTE)

conduit à deux relations en 3FN. Mais cette décomposition n'est pas totalement satisfaisante car elle fait perdre la dépendance fonctionnelle AVION $\rightarrow$ PILOTE. Les problèmes de mises à jour déjà signalés proviennent de cette situation. Ils disparaissent en adoptant la décomposition suivante.

R1 (VOL, AVION)
R3 (AVION, PILOTE)

Cette décomposition, grâce à la propriété de transitivité, permet effectivement de retrouver toutes les dépendances fonctionnelles initiales.

d) Intérêt de la normalisation

• Suppression des problèmes de mise à jour.
• Minimisation de l'espace de stockage pour une relation (en éliminant les redondances) : la taille des fichiers normalisés croît de façon arithmétique alors que la taille des fichiers non normalisés croît de façon géométrique.

4. Décomposition d’une relation

a) Formulation du problème

On note R = <U, D> un schéma relationnel où U est un ensemble de constituants et D un ensemble de dépendances sur ces constituants.

Schéma universel = schéma constitué d'une seule relation rassemblant tous les attributs de U et toutes les dépendances de D.

En effectuant une modélisation avec le modèle relationnel, on choisit un ensemble de relations susceptibles de représenter au mieux le monde réel et ne comportant pas d’anomalies de mises à jour. En faisant l'hypothèse que le monde réel est modélisable par un schéma relationnel R = <U, D> non normalisé, est-il possible de remplacer (on dit aussi décomposer) le schéma R par un ensemble de schémas normalisés $S = \{R_1, R_2, ..., R_k\}$ où $R_i = <U_i, D_i>$? Dans le cas général, la réponse est négative car il n’est pas toujours possible de préserver le contenu et (ou) les dépendances.

b) Préservation du contenu et des dépendances

<u>Préservation du contenu</u> :

Un algorithme de décomposition préserve le contenu si la relation initiale R peut être reconstruite par jointure à partir des relations R_i :

$R_1 \infty R_2 \infty ... \infty R_n = R$. ($\infty$ symbol de la jointure)

Les tuples obtenus par jointures doivent être exactement ceux de R : la recomposition par jointures ne doit ni enlever de tuples, ni ajouter de tuples.

Préservation des dépendances :

Un algorithme de décomposition préserve les dépendances si les dépendances initiales de D peuvent être reconstruites à partir des D_i. Autrement dit, si la fermeture de D est identique à la fermeture de l'union des D_i :

$(D_1 \cup D_2 \cup ... \cup D_n)^+ = D^+$

c) Algorithme de décomposition en 3FN par agrégation

Cet algorithme consiste à rechercher une couverture minimale des dépendances fonctionnelles, puis à réaliser une partition de cette couverture de façon à associer chaque élément de la partition à une relation normalisée.

Algorithme d'agrégation en 3FN d'un schéma R = <U, D> :

1) Rechercher une couverture minimale G de D.
2) Partitionner G en groupes de dépendances G_i ayant la même partie gauche.
3) Fusionner les groupes G_i et G_j possédant des parties gauches X_i et X_j équivalentes (c'est-à-dire ceux pour lesquels on a $X_i \rightarrow X_j$ et $X_j \rightarrow X_i$, $X_i \in G_i$, $X_j \in G_j$).
4) Associer à chaque groupe G_i un schéma $R_i = \langle U_i, D_i \rangle$. Autrement dit, construire une relation R_i pour chaque G_i dont la clé est la partie gauche des DF de D_i et les constituants non clés est la partie droite des DF de D_i.
5) Si aucun des schémas précédents ne contient une clé K de R, créer un schéma supplémentaire $\langle K, \varnothing \rangle$.

Les étapes 2 et 3 permettent de répartir toutes les dépendances fonctionnelles dans des schémas en 3FN. Cet algorithme assure donc bien la préservation des dépendances. La préservation du contenu est garantie par l'étape 5.

Exemple : Considérons le schéma relationnel R = < U, D >.

U = {A, B, C, Q, E, F}

D = $\{A \rightarrow B ; B \rightarrow A ; A \rightarrow F ; A, C \rightarrow Q ; C \rightarrow E\}$

- Quelle est la forme Normale de R ?
- Appliquer l'algorithme de décomposition pour décomposer R.

Solution :

- Clé de R ?

  On commence par le calcul d'une couverture minimale D^0 de l'emble D.

  - On remarque que D est un ensemble de dépendances élémentaires non redondantes ; donc $D^0 = D$.
  - Les attributs F,Q,E sont des attributs non clé (Toujours à droites des dfs

de D^0).

Attribut non clé

- La relation R possède deux clés candidates : AC et BC ;
- La relation R est en 1FN, elle n'est pas en 2FN à cause de la df $A \rightarrow F$

Partie de clé

- **Décomposition de R :**

  - Application de l'algorithme :

D est une couverture minimale. On peut donc prendre G = D.

L'étape 2 fournit les groupes suivants.

$G1 = \{C \rightarrow E\}$
$G2 = \{A, C \rightarrow Q\}$
$G3 = \{A \rightarrow B \; ; A \rightarrow F\}$
$G4 = \{B \rightarrow A\}$

L'étape 3 conduit à fusionner G3 et G4. On obtient un nouveau groupe G3.

$G1 = \{C \rightarrow E\}$
$G2 = \{A, C \rightarrow Q\}$
$G3 = \{A \rightarrow B \; ; B \rightarrow A \; ; A \rightarrow F\}$

L'étape 4 génère les schémas suivants.

$R1 = <\{C, E\}, \{C \rightarrow E\}>$
$R2 = <\{A, C, Q\}, \{A, C \rightarrow Q\}>$
$R3 = <\{A, B, F\}, \{A \rightarrow B \; ; B \rightarrow A \; ; A \rightarrow F\}>$

Une clé de la relation universelle est A, C. Puisque R2 contient déjà cette clé, le schéma {R1, R2, R3} préserve le contenu.

Exercice : Normalisation

Soit la relation universelle suivante :

EMPRUNTER (Code, Titre, Auteur, ISBN, **Matricule**, Nom, Date_Nais, Date_Emprunt, Date_retour) ;
Cette relation modélise l'opération d'emprunt des livres par des étudiants.
Et soit l'ensemble des dépendances Fonctionnelles associées ; F= {Code →Titre, Auteur, ISBN ; ISBN → Code ; Matricul**e** →Nom, Date_Nais ; ***Code***, ***Matricule*** →Date_Emprunt, Date_retour}

Questions/

1- Clés de la relation ?
2- Forme Normale de la relation ?
3- Décomposer la relation si elle n'est pas en 3FN.

Réponse :

1- Calcul d'une couverture F^O de F :

F^o = {Code →Titre ; Code →Auteur ; Code → ISBN ; ISBN → Code ; Matricule →Nom ; Matricule →Date_Nais ; ***Code***, ***Matricule*** →Date_Emprunt ; ***Code***, ***Matricule*** → Date_retour}.

La relation EMPRUNTER possède deux clés candidates :

Code, ***Matricule ;*** ***ISBN***, ***Matricule*** .

On choisit (Code,Matricule) clé primaire.

2- La relation est en 1FN, elle n,est pas en 2 FN à cause de la DF Code →Auteur(par exemple).

Une partie de clé détermine Un attribut non clé

Donc la relation EMPRUNTER n'est pas en 3FN.

3- Décomposition de la relation EMPRUNTER :

LIVRE (**Code**, Titre, Auteur, ISBN)

DFs associées : Code →Titre ; Code →Auteur ; Code → ISBN ; ISBN → Code ;

ETUDIANT (**Matricule**, Nom, Date_Nais)

DFs associées : Matricule →Nom ; Matricule →Date_Nais ;

EMPRUNT (***Code***, ***Matricule***, Date_Emprunt, Date_retour)

DFs associées ***Code***, ***Matricule*** →Date_Emprunt ; ***Code***, ***Matricule*** → Date_retour