Bases de données

Chapitre 4 : Dépendances Fonctionnelles et Normalisation (Partie1)

Prof. M. RAHMOUNI
md.rahmouni@yahoo.fr

Introduction

- Le but des dépendances fonctionnelles et de la théorie de la normalisation est de s'assurer que le **schéma relationnel** défini pour une base de données est correctement construit.
- Un mauvais schéma relationnel peut en effet entrainer des anomalies lors des manipulations.

•L'objectif de la normalisation est de construire un schéma de base de données cohérent et possédant certaines propriétés vérifiées par la satisfaction de formes normales.

- Pour une application spécifique, il est en effet possible de proposer plusieurs schémas.
- Les questions qui se posent alors sont les suivantes :
  - qu'est-ce qu'un bon schéma ?
  - quel schéma choisir ?
- Un mauvais schéma défini lors de la phase de conception peut conduire à un certain nombre d'anomalies pendant la phase d'exploitation de la base :
  - des redondances d'information,
  - des anomalies lors des opérations de mise à jour (insertions, suppressions, modifications).

Exemple :
Soit le schéma de la relation FOURNISSEUR (Nom_Fournisseur, Adresse, Produit, Prix).
Une relation (table) correspondant à ce schéma pourra éventuellement contenir plusieurs produits pour un même fournisseur. Dans ce cas, il faudra faire face à un certain nombre de problèmes :

- l'adresse du fournisseur sera dupliquée dans chaque n-uplet (redondance),
- si on souhaite modifier l'adresse d'un fournisseur, il faudra rechercher et mettre à jour tous les n-uplets correspondant à ce fournisseur,
- si on insère un nouveau produit pour un fournisseur déjà référencé, il faudra vérifier que l'adresse est identique,
- si on veut supprimer un fournisseur, il faudra retrouver et supprimer tous les n-uplets correspondant à ce fournisseur (pour différents produits) dans la table.

- Ces anomalies n'apparaitront pas si on décompose le schéma initial de base de données.
- Les 3 premières formes normales ont été proposées par E.F. Codd ("inventeur" du modèle relationnel) en 1972. La forme normale dite de Boyce-Codd a été proposée en 1974. Les 4ème (1977) et 5ème (1979) formes normales ont été proposées ensuite par Fagin, mais elles ne concernent que des cas rares et très spécifiques.
- Les formes normales s'appuient sur les dépendances fonctionnelles entre attributs d'un schéma de base de données.

Dépendances fonctionnelles

Définition 1 :

Un attribut (ou un groupe d'attributs) B est dit "fonctionnellement dépendant" d'un attribut (ou d'un groupe d'attributs) A si :
a1 = a2 => b1 = b2,(c-à-d si étant donné une valeur de A, lui correspond à une valeur unique de B)(quel que soit l'instant considéré)
a1, a2, b1, b2 étant des réalisations (valeurs) des attributs A et B dans des n-uplets de la base de données.
On dit alors que A "détermine" B, et on note A → B.

Dépendances fonctionnelles

Définition 2 :

Soit R(U) une relation avec U l'ensemble de ses attributs.
Soit $X,Y \subseteq U$, i.e X et Y sont deux attributs ou ensemble d'attributs de R. On dit qu'il existe une DF entre X et Y(ou que X détermine Y ou encore que Y et déterminé par X), notée X→Y si et seulement si:
$\forall$t1 et t2, deux tuples de R, si t1[X]=t2[X] alors t1[Y]=t2[Y].
Pour une DF X→Y, X est la source et Y la cible.

Dépendances fonctionnelles

Exemple :

Considérons le schéma de la relation suivante : r(A, B, C, D, E).
Cette relation est définie en extension par les tuples suivants :

A	B	C	D	E
a1	b2	c2	d3	e2
a1	b2	c2	d1	e4
a2	b3	c2	d1	e5
a2	b4	c5	d1	e5

- Parmi les DFs suivantes, lesquelles s'appliquent à r?

- [ ] E→D
- [ ] D→E
- [ ] C→A
- [ ] E→B
- [ ] E→A
- [ ] B→C
- [ ] B→D
- [ ] B→A

Dépendances fonctionnelles

Exemple :

Parmi les DFs suivantes, lesquelles s'appliquent à r?

A	B	C	D	E
a1	b2	c2	d3	e2
a1	b2	c2	d1	e4
a2	b3	c2	d1	e5
a2	b4	c5	d1	e5

A→B
A→C
B→C
A→E
D→E

Dépendances fonctionnelles

Exemple :

Soit le schéma de relation PERSONNE (No_SS, Nom, Adresse, Age, Profession).

Les dépendances fonctionnelles qui s'appliquent sur ce schéma de relation sont les suivantes :

- No_SS -> Nom,
- No_SS -> Adresse,
- No_SS -> Age,
- No_SS -> Profession.

On pourra aussi écrire : No_SS -> Nom Adresse Age Profession.

L'attribut No_SS détermine tous les attributs du schéma de relation. Il s'agit d'une propriété de la clé d'une schéma de relation.

- La normalisation est donc le processus de transformation d'une relation ayant des problèmes lors de la mise à jour vers une autre relation n'ayant pas ces problèmes
- L'objectif est donc de définir un bon schéma relationnel qui décrit d'une manière convenable le système d'information d'une entreprise.
- Le processus de normalisation d'une BD n'est pas obligatoire : outil pratique et performant

10

Exemple :

- Considérons la relation suivante

PRODUIT (Refproduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

- Elle est visiblement redondante

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

11

Cette relation présente certaines anomalies :

- Redondance : un produit apparaît autant de fois qu'il sera livré par un service
- Mise à jour : faute de redondance, les mises à jour conduiront à des risques d'incohérence et de non intégrité.
- Insertion et suppression : l'insertion et la suppression ou le transfert d'attributs pourront faire apparaître des valeurs nulles

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

12

On peut dire qu'une Base de Données relationnelle est <u>'correcte'</u> ou <u>normalisée</u> si :

- chaque relation décrit une information élémentaire avec les seuls attributs qui lui sont directement liés
- il n'y a pas de redondance d'information qui peuvent produire des problèmes de mise à jour.

- La relation Produit peut être décomposée en trois relations non redondantes : PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

Décomposition 1

PRODUIT2 (<u>RefProduit, NumService</u>, Quantité, Adresse, Capacité)

PRODUIT1(<u>RefProduit</u>, Libelle, PU)

P1	CH7	23.510
P3	VIS12	0.150

Décomposition 2

PRODUIT 21 (<u>RefProduit, NumService</u>, Quantité)

P1	S1	300
P1	S2	500
P3	S4	900

PRODUIT22 (<u>NumService</u>, Adresse, Capacité)

S1	Sousse	9000
S2	Tunis	6000
S4	Sousse	2000

PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

Décomposition 1

PRODUIT2 (<u>RefProduit, NumService</u>, Quantité, Adresse, Capacité)

PRODUIT1(<u>RefProduit</u>, Libelle, PU)

P1	CH7	23.510
P3	VIS12	0.150

Décomposition 2

PRODUIT 21 (<u>RefProduit, NumService</u>, Quantité)

P1	S1	300
P1	S2	500
P3	S4	900

PRODUIT22 (<u>NumService</u>, Adresse, Capacité)

S1	Sousse	9000
S2	Tunis	6000
S4	Sousse	2000

Le résultat final de la décomposition est donc les relations suivantes :

- PRODUIT1 (<u>RefProduit</u>, LibelleProduit, PU)
  contient les données relatives aux produits.
- PRODUIT21 (<u>RefProduit #, NumService#</u>, Quantité)
  contient les données relatives aux produits distribués par des services.
- PRODUIT22 (<u>NumService</u>, Adresse, Capacité)
  contient les données relatives aux services.

Dépendance fonctionnelle

Définition

- Un attribut ou une liste d'attributs Y dépend fonctionnellement d'un attribut ou d'une liste d'attributs X dans une relation R, si, étant donnée une valeur de X, il ne lui est associé qu'une seule valeur de Y dans tout tuple de R.
- On notera une telle dépendance fonctionnelle :

X→Y (X détermine Y ou Y dépend fonctionnellement de X).

17

Exemple

PRODUIT (RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

Pour cette relation, les dépendances fonctionnelles suivantes sont vérifiées :

RefProduit → LibelleProduit

NumService → Adresse, Capacité

RefProduit → PU

RefProduit, NumService → Quantité

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

18

Propriétés des dépendances fonctionnelles

- Des axiomes et des règles d'inférence permettent de découvrir de nouvelles dépendances à partir d'un ensemble initial. Dans ce que suit nous considérons R une relation.
- Les trois premières propriétés sont connues sous le nom « Axiomes d'Armstrong »

19

Propriété 1 : Réflexivité

$Y \subseteq X \Rightarrow X \rightarrow Y$

Tout ensemble d'attributs détermine lui-même ou une partie de lui-même.

Propriété 2 : Augmentation

$X \rightarrow Y \Rightarrow X, Z \rightarrow Y, Z$

Si X détermine Y, les deux ensembles d'attributs peuvent être enrichis par un même troisième.

Propriété 3 : Transitivité

$X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$.

Propriété 4 : Union

$X \rightarrow Y$ et $X \rightarrow Z \Rightarrow X \rightarrow Y, Z$

Propriété 5 : Pseudo-transitivité:

$X \rightarrow Y$ et $W,Y \rightarrow Z \Rightarrow W,X \rightarrow Z$

Propriété 6 : Décomposition:

$X \rightarrow Y$ et $Z \subseteq Y \Rightarrow X \rightarrow Z$

20

Dépendance fonctionnelle élémentaire

- Une Dépendance fonctionnelle X ⟶Y est élémentaire si pour tout X' ⊂ X la dépendance fonctionnelle X' ⟶Y n'est pas vraie.

  En d'autres termes, Y ne dépend pas fonctionnellement d'une partie de X (X est la plus petite quantité d'information donnant Y).

- Exemple :

  RefProduit, LibelleProduit ⟶ PU

  n'est pas élémentaire car il suffit d'avoir la référence du produit pour déterminer le prix unitaire.

21

Dépendance fonctionnelle canonique

- Une Dépendance fonctionnelle X⟶Y est canonique si sa partie droite ne comporte qu'un seul attribut.
- et un ensemble F de dépendances fonctionnelles est canonique si chacune de ses dépendances est canonique.

22

Clé d'une relation

- est l'ensemble d'attributs dont les valeurs permettent de caractériser les n-uplets de la relation de manière unique.
- Formellement :

  Un attribut ou une liste d'attributs X est une clé pour la relation R(X,Y,Z) si
  - Y et Z dépendent fonctionnellement de X dans R : X⟶ Y, Z.
  - et X⟶ Y, Z est élémentaire.

23

TD – Dépendances fonctionnelles

Exercice 1

• L'axiome de pseudo transitivité nous dit que si X→Y et YW→Z, alors XW→Z. Démontrer cet axiome à l'aide des autres axiomes d'Armstrong.

Exercice 2

En utilisant les axiomes d'Armstrong, démontrer que si X → YZ et Z → CW alors X → YZC.

Exercice 3

Soit R(A,B,C,D,E,G,H) F = { AB → C ; B → D ; CD → E ; CE → GH ; G → A }. En utilisant les axiomes d'Armstrong, montrer que l'on peut déduire de cet ensemble :

1. AB → E
2. BG→C
3. AB→G

Exercice 4

Soit R(A,B, E,G,H,I,J) et F = {AB → E; AG → J; BE → I; E → G; GI → H} En utilisant les axiomes d'Armstrong, montrer que l'on peut déduire de cet ensemble :

1. ABG→EGJ
2. AB→GH
3. BE→H

Exercice 5

- Soit R(A,B,C,D,E,G,H) et F = {AB → C, B → D, CD → E, CE → GH, G → A}. En utilisant les axiomes d'Armstrong, montrer que l'on peut déduire de cet ensemble :

1. ABC → E
2. BG → C
3. BG → GH
4. GBCE → GH
5. AB → GH

Axiomes d'Armstrong

Exercice 1

L'axiome de pseudo transitivité nous dit que si X→Y et YW→Z, alors XW→Z.
Démontrer cet axiome à l'aide des autres axiomes d'Arstrong.
X→Y alors XW→YW (accroissement)
XW→YW et YW→Z alors XW→Z (transitivité)

Exercice 2

En utilisant les axiomes d'Armstrong, démontrer que si X→YZ et Z→CW alors X →YZC
Z→CW alors Z→CWZ (accroissement)
Z→CWZ alors YZ→CWZY (accroissement)
X→YZ et YZ→CWZY donc X→CWZY(transitivité)
X→CWZY donc X→CZY (projectivité)

Exercice 3

Soit R(A,B,C,D,E,G,H) F = { AB→ C ; B→ D ; CD→ E ; CE→ GH ; G→ A }. En utilisant les axiomes d'Armstrong, montrer que l'on peut déduire de cet ensemble :

1. AB→E
   B→D donc AB→D par augmentation
   AB→C et AB→ D donc AB→CD par union
   AB→CD et CD→E donc AB→E par transitivité.
2. BG→C
   G → A donc BG → A par augmentation,
   BG → BG donc BG → B par projection,
   BG → A et BG → B donc BG → AB par union,
   BG → AB et AB → C donc BG → C par transitivité.
3. AB→G
   AB → E et AB → C donc AB → CE par additivité,
   AB → CE et CE → GH donc AB → GH par transitivité,
   AB → GH donc AB → G par projection.

Exercice 4

Soit R(A,B, E,G,H,I,J) et F = {AB→E; AG→J; BE→I; E→G; GI→H}
En utilisant les axiomes d'Armstrong, montrer que l'on peut déduire de cet ensemble :

1. ABG→EGJ
   AB→E donc ABG→EG
   AG→J donc ABG→GJ
   ABG→EJG
2. AB→GH
   AB→ E et E→G, par transitivité AB→ G
   AB→E, par augmentation AB→BE
   AB→BE et BE→I, par transitivité AB→I
   AB→G et AB→I, par union AB→GI
   AB→GI et GI→H, par transitivité AB→H
   AB→G et AB→H, par union AB→GH
3. BE→H
   E→G donc BE→G
   BE→G et BE→I donc BE→GI
   BE→GI et GI→H donc BE→H

Exercice 5

Soit R(A,B,C,D,E,G,H) et F = {AB→C, B→D, CD→E, CE→GH, G→A}.
En utilisant les axiomes d'Armstrong, montrer que l'on peut déduire de cet ensemble :

1. ABC→E
   AB→C et CD→E donc ABC→E
2. BG→C
   G→A donc BG→AB
   BG→AB et AB→C donc BG→C
3. BG→GH
   B→D donc BG→D
   BG→C et BG-D donc BG→CD
   CD→E donc CD→CE
   BG→CD et CD→CE donc BG→CE
   BG→CE et CE→GH donc BG→GH
4. GBCE→GH
   G→A donc GB→AB
   GB→AB et AB→C donc GB→C
   GB→C et CD→E donc GBC→E
   GBC→E donc GBCE→CE
   GBCE→CE et CE→GH donc GBCE→GH
5. AB→GH
   B→D donc AB→D
   AB→D et AB→C donc AB→CD
   CD→E donc CD→CE
   AB→CD et CD→CE donc AB→CE
   AB→CE et CE→GH donc AB→GH