1. Introduction

Nous avons déjà mentionné qu'il existe trois grandes catégories de modèles de données : le modèle réseau, le modèle hiérarchique et le modèle relationnel. Les deux premiers sont de plus en plus abandonnés actuellement.

Le modèle relationnel a été inventé par CODD à IBM-San Jose en 1970. Il est basé sur la théorie de la relation et de la normalisation des relations.

2. Les concepts de base du modèle

Le concept de relation découle directement de la théorie des ensembles. Nous allons en rappeler la définition qui nécessite tout d'abord l'introduction de la notion de domaine.

Notion de Domaine

Ensemble de valeurs caractérisées par un nom.

A titre d'exemple, on peut considérer le domaine des entiers, celui des réels, celui des valeurs logiques {vrai, faux}, celui des jours de la semaine {samedi, dimanche, lundi, mardi, mercredi, jeudi, vendredi}, celui des couleurs du drapeau algérien {rouge, vert, blanc}…etc.

Rappelons que le produit cartésien d'un ensemble de domaines D1, D2, ….,Dn, noté D1xD2x….xDn, est l'ensemble des tuples $<v_1, v_2, \ldots, v_n>$ tels que $v_i \in$ Di.
C'est à dire c'est la totalité des combinaisons possibles que l'on peut tirer de ces domaines.
Par exemple, le produit cartésien des domaines pays={Algérie, France, Espagne} et capitale={Alger, Paris, Madrid} est composé des tuples suivants :

Algérie	Alger
Algérie	Paris
Algérie	Madrid
France	Alger
France	Paris
France	Madrid
Espagne	Alger
Espagne	Paris
Espagne	Madrid

Nous pouvons maintenant introduire la notion de relation

Notion de relation

sous-ensemble du produit cartésien d'une liste de domaines
Une relation est généralement caractérisée par un nom.

Par exemple, à partir des domaines :
D1={Algérie, France, Espagne} et D2={Alger, Paris, Madrid}
On peut composer une relation qui nous intéresse, c'est à dire celui qui a un sens par rapport à la réalité :

Pays_capital	**D1**	**D2**
	Algérie	Alger
	France	Paris
	Espagne	Madrid

- Plus simplement, une relation peut être vue comme un tableau à deux dimensions dont les colonnes correspondent aux domaines et les lignes contiennent les tuples. On parle parfois de ***table relationnelle***.

Notion de tuple (n-uplet)
est une ligne d'une relation

Par exemple, (Algérie, Alger)

- Afin de rendre l'ordre des colonnes sans importance tout en permettant plusieurs colonnes de même domaine, on associe un nom à chaque colonne. On parle parfois *d'enregistrement.*

Notion d'attribut
C'est une colonne d'une relation caractérisée par un nom

- On parle parfois de *constituant* ou *composant*

Notion de schéma de relation
C'est le nom d'une relation suivi de la liste des attributs qui la constituent.

Par exemple, une relation décrivant des voitures et comportant les attributs : N°V, marque, type, puissance, couleur, aura pour schéma :
VOITURE(N°V, marque, type, puissance, couleur)

- A toute relation, il est possible d'associer un schéma de relation qui représente l'intention de la relation, alors que le tableau avec les tuples représente une extension.

Nous pouvons maintenant introduire la notion de base de données relationnelle

Notion de base de données relationnelle
C'est une base de données dont le schéma est un ensemble de schémas de relations et dont les occurrences sont des tuples de ces relations.

3. Introduction a la conception de schémas relationnels

3.1 Perception du monde réel

Avant de construire le modèle relationnel, le monde réel peut être modélisé au niveau conceptuel à l'aide du modèle Entité/Association (E/A) qu'est une représentation graphique des données.
Les concepts utilisés dans le modèle E/A sont-les :
Entité : qui représente un objet ayant une existence visible
Association : est une relation qui s'établit entre deux entités ou plus.
Cardinalité : c'est le nombre minimal et maximal d'occurrences d'une association
Propriété : est une information décrivant une caractéristique d'une entité ou d'une association.

A titre d'exemple, soit des données modélisant des entités 'personne' et 'voiture', et le type d'association 'possede' qui traduit le fait qu'une personne soit propriétaire d'une ou plusieurs voitures. Une personne est caractérisée par un numéro de sécurité sociale (N°SS), un nom et un prénom alors qu'une voiture est caractérisée par les propriétés déjà vues N°V, marque, type, puissance, et couleur. Chaque personne est identifiée par une occurrence du N°SS alors que chaque voiture est identifiée par son N°V. A chaque occurrence d'association correspond par exemple une date d'achat (date) et un prix d'achat (prix). La figure.III.1illustre la partie du monde réel modélisé par le modèle E/A en représentant une entité avec leurs propriétés par un rectangle, une association avec leurs propriétés par un hexagone.

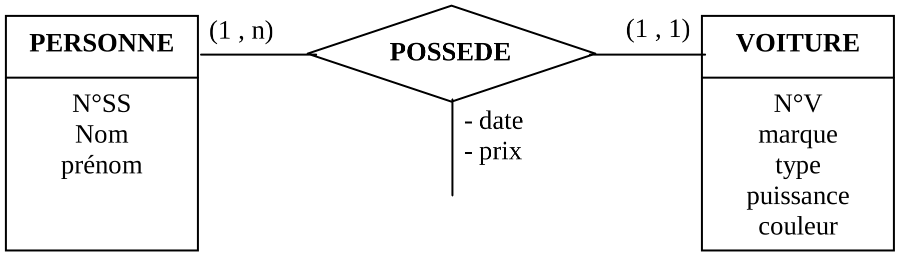


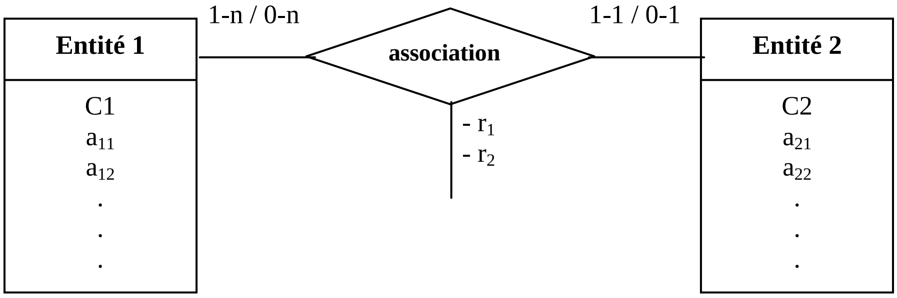
Fig.III.1-représentation du monde réel par des entités et des associations

3.2 Les règles de passage du niveau conceptuel au modèle relationnel

Le modèle relationnel se prête bien à la représentation des entités et des associations :

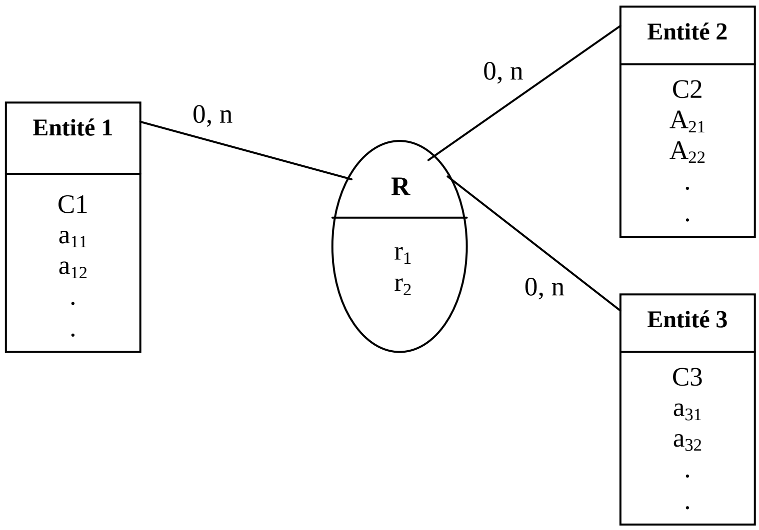
- Une propriété devient un attribut
- Une entité est représentée par une relation (table) dont le schéma est le nom de l'entité suivi de la liste des attributs de l'entité et son identifiant devient sa clé primaire.
- Une association devient une relation et on distingue deux cas selon le type de l'association :
  a) Relation binaire de type père fils : ce sont donc des relations binaires (2 dimensions) pour les quelles les cardinalités de l'entité père sont 0-N / 1-N et les cardinalités de l'entité fils sont 0-1 / 1-1
     - ➢ les propriétés de l'association deviennent des attributs de la relation fils
     - ➢ la relation disparaît
     - ➢ l'identifiant de l'entité père devient la clé secondaire (étrangère) de la relation fils

  A titre d'exemple :



Résultat : relation entité1(<u>C1</u>, a_{11}, a_{12}, …..)
relation entité2(<u>C2</u>, C1, a_{21}, a_{22}, …..r_1, r_2)

b) autre relation : Pour les autres relations, la clé d'une telle relation est composée de l'ensemble des identifiants des entités qui participent à l'association du formalisme individuel.



Résultat : relation entité1(<u>C1</u>, a_{11}, a_{12}, …..)
relation entité2(<u>C2</u>, a_{21}, a_{22}, …..)
relation entité3(<u>C3</u>, a_{31}, a_{32}, …..)
relation R(<u>C1</u>, <u>C2</u>, <u>C3</u>, r_1, r_2, …)

3.3 Problèmes soulevés par une mauvaise perception du monde réel

Une mauvaise conception des entités et des associations représentant le monde réel modélisé conduit à des relations problématiques. Imaginons par exemple que l'on concevra une entité unique propriétaire contenant tous les attributs des trois relations personne, voiture et possede. La figure suivante représente une extension possible de cette relation.

N°V	Marque	Type	Puissance	Couleur	N°SS	Nom	Prénom	Date	Prix
342.185	Peugeot	406	9	Verte	100	Omari	M^ed	12-12-88	400000DA
155.188	Renault	R12	6	Grise	150	Hamidi	Rafik	25-05-89	300000DA
226.186	Renault	R19	9	Rouge	100	Omari	M^ed	19-04-90	600000DA
145.187	Renault	R21	9	Jaune	150	Hamidi	Rafik	19-05-88	500000DA
697.188	Peugeot	305	9	Verte	200	Saleh	Ahmed	10-10-89	450000DA

Cette relation souffre de plusieurs types d'anomalies :
- Tout d'abord, des données sont redondantes ; par exemple, Omari et Hamidi apparaissent deux fois ; plus généralement, une personne apparaît autant de fois qu'elle possède de voitures
- Ces redondances conduisent à des risques d'incohérence lors d'une mise à jour ; par exemple, si l'on s'aperçoive que le prénom de Hamidi n'est pas Rafik mais Mostefa, il faudra veiller à mettre à jour tous les tuples contenant Hamidi
- Il est nécessaire d'autoriser la présence de valeurs nulles dans une telle relation afin de pouvoir conserver les personnes qui n'ont plus de voitures.

En résumé, une relation qui ne représente pas de vraies entités ou associations semble donc souffrir de présence de données redondantes et d'incohérences potentielles et nécessite le codage de valeurs nulles. Il y a tout intérêt à éliminer ces anomalies afin de faciliter la manipulation des relations.

3.4 L'approche par décomposition

L'approche par décomposition pour concevoir des schémas relationnels tend à partir d'une relation composée de tous les attributs, appelée relation universelle, à décomposer cette relation en sous-relations qui ne souffriraient pas des anomalies précédemment signalées. Il doit être réalisé par un algorithme à partir d'une bonne compréhension des propriétés sémantiques des données. Cette approche (fig.III.2) utilise un algorithme de décomposition.

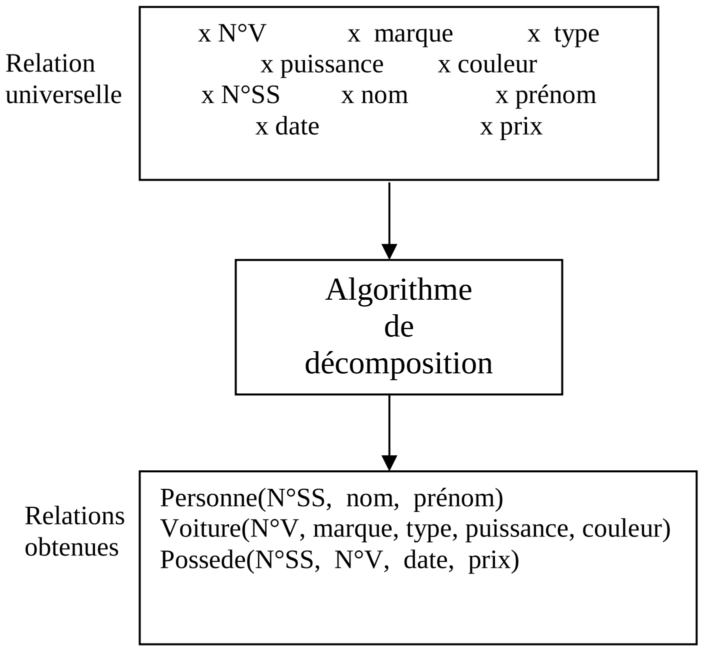


Fig.III.2 décomposition d'une relation universelle

La compréhension de la théorie de décomposition des relations nécessite la connaissance de deux opérations de base : la projection et la jointure.

La projection est l'opération qui consiste à supprimer des attributs d'une relation et à éliminer les tuples en double qui risquent d'apparaître dans la nouvelle relation obtenue. La projection de la relation R de schéma $R(A_1, A_2, \ldots, A_n)$ sur les attributs $A_1, A_2, \ldots, A_p$ est une relation R' de schéma $R'(A_1, A_2, \ldots, A_p)$ obtenue par élimination des attributs de R n'appartenant pas à R' et suppression des tuples en doubles.
Une telle projection R' sera notée : $R' = \Pi\ (A_1, A_2, \ldots, A_p)[R]$
Par exemple, la projection de la relation propriétaire sur les attributs nom et prénom donne la relation suivante :

Π (nom, prénom)[propriétaire]	Nom	Prénom
	Omari	M^{ed}
	Hamidi	Rafik
	Saleh	Ahmed

La jointure naturelle ou simplement jointure, est l'opération inverse de la projection. La jointure de deux relations R et S, de schéma respectif $R(A_1, A_2, \ldots, A_n)$ et $S(B_1, B_2, \ldots, B_p)$ est une relation T, notée $R \bowtie S$ ayant pour attribut l'union des attributs de R et S, soit
$\{A_1, A_2, \ldots, A_n\} \cup \{B_1, B_2, \ldots, B_p\}$
et pour tuples tous ceux obtenus par concaténation des tuples de R et S ayant même valeurs pour les attributs de même nom. Ainsi, on a :
$\Pi\ A_1, A_2, \ldots, A_n(T) = R$

Π $B_1, B_2, \ldots, B_p(T)=S$
L'exemple suivant fait la jointure de deux relations R et S :

R	marque	couleur
	Renault	Rouge
	Peugeot	Verte
	Citroen	Bleue
	Renault	Verte

S	couleur	puissance
	Rouge	6
	Verte	9
	Bleue	2
	Bleue	5
	verte	6

R $\bowtie$ S	marque	couleur	puissance
	Renault	Rouge	6
	Peugeot	Verte	9
	Peugeot	Verte	6
	Citroen	Bleue	2
	Citroen	Bleue	5
	Renault	Verte	9
	Renault	verte	6

Il est maintenant possible de définir précisément la notion de décomposition :

Notion de décomposition
Un schéma de relation $R(A_1, A_2, \ldots A_n)$ est convenablement décomposée en une collection de relations R1, R2,…,Rp, obtenues par des projections de R ,si la jointure R1$\bowtie$R2$\bowtie$... ….$\bowtie$Rp ait même schéma que R.

Par suite, lors d'une décomposition, le schéma de relation $R(A_1, A_2, \ldots A_n)$ est remplacé par une collection de schémas dont l'union des attributs est $(A_1, A_2, \ldots A_n)$. Il est important que la décomposition se fasse sans perte d'informations.

Notion de décomposition sans perte
Décomposition d'une relation R en R1, R2,…,Rp, telleque pour toute extension de R, on ait :
R=R1$\bowtie$R2$\bowtie$ …$\bowtie$Rp

C'est à dire, après la décomposition de relation R en une collection de relations R1, R2,…,Rp, si la jointure R1$\bowtie$R2$\bowtie$ …$\bowtie$Rp donne la même extension que R on dit que la décomposition sans perte, sinon, si donne seulement les mêmes attributs en dits que c'est une décomposition.

Le problème de la conception des bases de données relationnelles peut donc être vu comme celui de décomposer la relation universelle composée de tous les attributs en sous-relations ne souffrant pas des anomalies (vues au précédent) et de sorte à obtenir une décomposition sans perte. Nous allons ci-dessous étudier les principales méthodes proposées pour effectuer une telle décomposition qui doit permettre de déterminer des entités et associations canoniques du monde réel, donc en fait de construire un schéma conceptuel.

4. Les dépendances fonctionnelles(DF)

4.1 Notion de dépendance fonctionnelle
La notion de dépendance fonctionnelle fut introduite par CODD afin de caractériser des relations qui peuvent être décomposées sans perte d'informations.

Dans une relation, un attribut B dépend fonctionnellement d'un attribut A, si pour une valeur de A, il ne peut exister, qu'une valeur de B.

B dépend fonctionnellement de A, si, connaissant A, je connais une valeur de B.

Définition

Soit $R(A_1, A_2, \ldots A_n)$ un schéma de relation et X et Y des sous-ensembles de $\{A_1, A_2, \ldots A_n\}$. On dit que X détermine Y ou Y dépend fonctionnellement de X, si étant donné une valeur de X, il lui correspond une valeur unique de Y(et ceci quelque soit l'instant considéré).

A titre d'exemple, dans la relation voiture, on a les dépendances fonctionnelles :

N°V $\rightarrow$ couleur
type $\rightarrow$ marque
type$\rightarrow$ puissance
(type,marque) $\rightarrow$ puissance

Il est très important de remarquer qu'une dépendance fonctionnelle (DF) doit être vraie sur toutes les valeurs possibles et non sur les valeurs actuelles. Autrement dit, il est impossible de déduire les DF d'une réalisation particulière d'une relation. La seule manière de déterminer une DF est de regarder soigneusement ce que signifient les attributs car ce sont les déductions sur le monde réel qui lient les valeurs possibles des attributs entre elles. Les DF doivent être décelées par le concepteur au niveau du schéma conceptuel.

4.2 Propriétés des dépendances fonctionnelles

Les DF obéissent à plusieurs règles :

1. **Réflexivité :** $Y \subseteq X \quad \Rightarrow \quad X \rightarrow Y$; cette règle stipule que tout ensemble d'attributs détermine lui-même ou une partie de lui-même.

2. **Augmentation :** $X \rightarrow Y \;\Rightarrow\; XZ \rightarrow YZ$; cette règle signifie que si X détermine Y, les deux ensembles d'attributs peuvent être enrichis par un troisième.

3. **Transitivité :** $X \rightarrow Y$ et $Y \rightarrow Z \;\Rightarrow\; X \rightarrow Z$

Les trois règles précédentes composent les axiomes des DF. Plusieurs autres règles se déduisent des axiomes :

4. **Union :** $X \rightarrow Y$ et $X \rightarrow Z \;\Rightarrow\; X \rightarrow YZ$

5. **Pseudo-transitivité :** $X \rightarrow Y$ et $WY \rightarrow Z \;\Rightarrow\; WX \rightarrow Z$

6. **Décomposition :** $X \rightarrow Y$ et $Z \subseteq Y \;\Rightarrow\; X \rightarrow Z$

A partir de ces règles, il est possible d'introduire la notion de dépendance fonctionnelle élémentaire :

Notion de Dépendance Fonctionnelle Elémentaire

C'est une DF de la forme $X \rightarrow A$, où A est un attribut unique non inclus dans X ($A \not\subset X$) et où il n'existe pas $X' \subset X$ tel que $X' \rightarrow A$

La seule règle d'inférence qui s'applique aux dépendances fonctionnelles élémentaires est la transitivité.

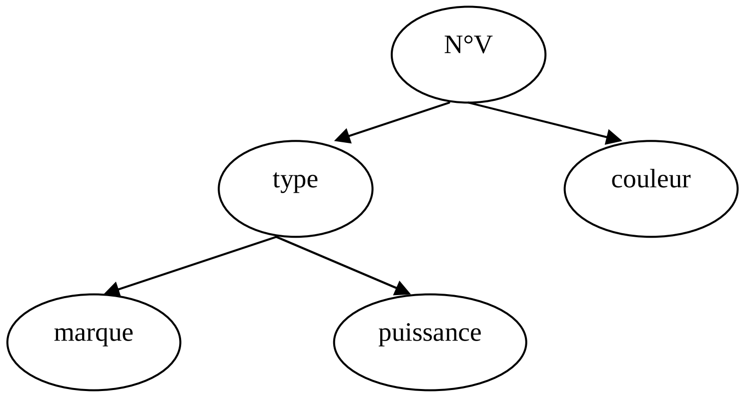
4.3 Graphe des dépendances fonctionnelles

Pour mieux identifier les dépendances fonctionnelles élémentaires, on a recours à une représentation graphique.

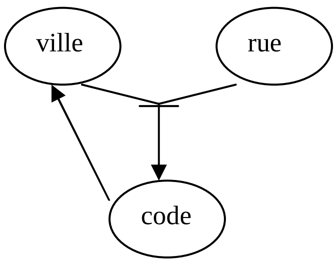
Par exemple, considérons les dépendances fonctionnelles entre les attributs de la relation VOITURE :

F={N°V→type, type→marque, type→puissance, N°V→couleur}

La figure suivante représente le graphe associé aux dépendances fonctionnelles de F



Autre exemple : soit la relation suivante code_postal(code, ville, rue), comportant les DF suivantes :
(ville, rue) → code
code → ville
le graphe de ces dépendances fonctionnelles est représenté par la figure suivante :



4.4 Fermeture transitive et couverture minimale

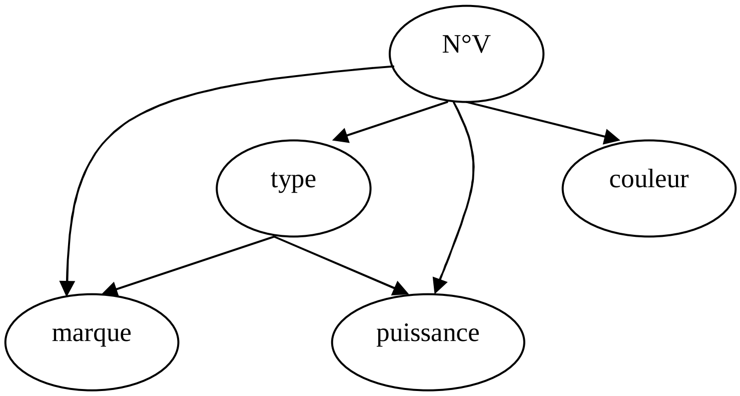
A partir d'un ensemble de DF élémentaires, on peut composer par transitivité d'autres DF élémentaires. On aboutit ainsi à la notion de fermeture transitive d'un ensemble F de DF élémentaires.

Notion de fermeture transitive

C'est l'ensemble des DF élémentaires considérées enrichis de toutes les DF élémentaires déduites par transitivité.

Par exemple, à partir de l'ensemble de DF :
F={N°V→type, type→marque, type→puissance, N°V→couleur}
On déduit la fermeture transitive :
$F^+=F\cup$ {N°V→marque, N°V→puissance}

Le graphe correspondant à fermeture transitive de F est :



A partir de la notion de fermeture transitive, il est possible de définir l'équivalence de deux ensembles de DF élémentaires : deux ensembles sont équivalents s'ils ont la même fermeture transitive.

D'autre part, il est intéressant de déterminer un sous-ensemble minimum de DF permettant de générer toutes les autres.

Notion de couverture minimale
C'est une ensemble de DF élémentaires associé à un ensemble d'attributs vérifiant les propriétés suivantes.
1-aucune DF n'est redondante.
2-toute DF élémentaire des attributs est dans la fermeture transitive.

Par exemple : F={N°V→type, type→marque, type→puissance, N°V→couleur}
est une couverture minimale pour l'ensemble des DF de la base de données voiture.

La couverture minimale est essentielle pour décomposer les relations sans perte d'informations.

5. Notion de clé et trois premières formes normales

5.1 Notion de clé d'une relation

C'est un sous-ensemble X des attributs d'une relation $R(A_1, A_2, \ldots, A_n)$ telque
1-$X \rightarrow A_1, A_2, \ldots, A_n$
2-Il n'existe pas de sous-ensemble $Y \subset X$ telque $Y \rightarrow A_1, A_2, \ldots, A_n$

En clair, une clé est un ensemble minimum d'attributs qui détermine tous les autres. Par exemple, N°V est une clé de la relation voiture alors que (N°V, type) n'est pas une clé.
Il peut y avoir plusieurs clés pour une relation, on les appelle clés candidates, parmi elles, l'une sera choisie pour être la *clé primaire*, les autres clés candidates sont des clés secondaires.

5.2 Définition des trois premières formes normales

Les trois premières formes normales ont pour objectif de supprimer toutes les redondances et les anomalies dans la base de données et de permettre la décomposition de relations en plusieurs relations plus petites sans perdre d'informations, à partir de la notion de dépendance fonctionnelle.

🕮 Notons qu'en arrivant au stade du MCD validé dans la méthode de merise, ce dernier est en 2èmeforme normale d'après les règles de la normalisation au niveau du MCD brut. Mais, il reste la 3èmeforme normale malgré qu'il en existe cinq formes normales, nous ne intéressons pas dans ce qui suit aux 4èmeforme normale et 5èmeforme normale.

💣 Notons aussi que si une table relationnelle doit être en 3èmeforme normale, il faudrait qu'elle soit en 1èreforme normale et 2èmeforme normale.

La première forme normale permet simplement d'obtenir des valeurs d'attributs non décomposable.

Notion de première forme normale (1FN)

Une relation est en première forme normale, si tout attribut contient une valeur atomique.

Cette forme normale consiste simplement à éviter les domaines composés de plusieurs valeurs. Par exemple, la relation personne(nom, prénom) sera ainsi décomposée en personne1(nom, prenom1) et personne2(nom, prenom2).

La deuxième forme normale permet d'assurer l'élimination de certaines redondances en garantissant qu'aucun attribut n'est déterminé seulement par une partie de la clé.

Notion de deuxième forme normale (2FN)

Une relation est en deuxième forme normale si et seulement si :
1-elle est en première forme normale
2-chaque attribut non identifiant, dépend fonctionnellement de la clé dans sa totalité

Par exemple : soit la relation etudiant
ETUDIANT(<u>nom</u>, adresse, <u>matière</u>, note)
Nous remarquons qu'il existe une anomalie dans la relation etudiant, en effet la 2èmeFN n'est pas vérifiée :
L'attribut adresse dépend d'une partie de la clé 'nom' et non pas de toute la clé
La solution est d'éclater la relation etudiant en deux tables relationnelles en 2FN :
ETUDIANT(<u>nom</u>, adresse)
EXAMEN(<u>nom</u>, <u>matière</u>, note)

La troisième forme normale permet d'assurer l'élimination des redondances dues aux dépendances transitives.

Notion de troisième forme normale (3FN)

Une relation est en troisième forme normale, si et seulement si :
1-elle est en deuxième forme normale
2-tout attributs n'appartenant pas à une clé ne dépendent pas d'un attribut non clé.

Cela signifie qu'il n y a pas de dépendance fonctionnelle transitive dans la relation.
A titre d'exemple, la relation :
VOITURE(<u>N°V</u>, marque, type, puissance, couleur)

n'est pas en troisième forme normale. En effet, l'attribut non clé type détermine marque et aussi puissance. Cette relation peut être décomposée en deux relations :
VOITURE(<u>N°V</u>, type, couleur)
MODELE(<u>type</u>, marque, puissance)

5.3 Propriétés d'une décomposition en troisième forme normale

Les DF sont des règles indépendantes du temps que doivent vérifier les valeurs des attributs. Il est nécessaire qu'une décomposition préserve ces règles.

Notion de décomposition préservant les DF

Décomposition {R1, R2, …..,Rn} d'une relation R telleque la fermeture transitive des DF de R est la même que celle de l'union des DF de {R1, R2, …..,Rn}.

D'où l'importance de la troisième forme normale. En effet, toute relation a au moins une décomposition en troisième forme normale telle que :
1-la décomposition préserve les DF
2-la décomposition est sans perte
cette décomposition peut ne pas être unique

5.4 Algorithme de décomposition en troisième forme normale

Il existe donc au moins une décomposition en troisième forme normale préservant les DF et sans perte. Une telle décomposition peut être produite par un algorithme ayant pour entrées l'ensemble des attributs ainsi que les DF (voir figure suivante).

Le principe d'un tel algorithme consiste à partir d'une couverture minimale des DF, puis à éditer ensemble les attributs isolés dans une relation dont tous les attributs sont clés. Ensuite, on recherche le plus grand ensemble X d'attributs qui détermine d'autres attributs A_1, A_2, ….,A_n et l'on sort la relation (X, A_1, A_2, ….,A_n). Une telle relation de cl é X est bien en 3FN, les DF
$X \rightarrow A_1$, A_2, ….,A_n sont alors éliminés de la couverture minimale, ainsi que les attributs isolés (non source ou cible de DF) de l'ensemble des attributs.

5. 5 Notion de forme normale de Boyce-Codd (BCNF)

Une relation est en forme normale de Boyce Codd, si et seulement si, les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut.

Exemple :
R (<u>matière</u> , <u>N°classe</u> , code_prof)
N'est pas en BCNF, car matière (partie de l'identifiant) dépend de la propriété code_prof