

LICENCES MATHÉMATIQUES ET INFORMATIQUE  
 3ÈME ANNÉE - FORMATION INITIALE ET PAR APPRENTISSAGE  
 BASES DE DONNÉES RELATIONNELLES  
 POLYCOPIÉ DE COURS - FORMES NORMALES

Maude Manouvrier

*La reproduction de ce document par tout moyen que ce soit est interdite conformément aux articles L111-1 et L122-4 du code de la propriété intellectuelle.*

## Table des matières

<b>10 Formes normales</b>	<b>2</b>
10.1 BCNF ( <i>Boyce-Codd Normal Form</i> ) . . . . .	2
10.1.1 Définition . . . . .	2
10.1.2 Propriétés d'une relation de schéma BCNF . . . . .	2
10.1.3 Décomposition de schéma BCNF sans perte d'information . . . . .	3
10.2 Troisième Forme Normale . . . . .	4
10.2.1 Définitions . . . . .	4
10.2.2 Redondance . . . . .	5
10.2.3 Décomposition 3NF sans perte de dépendances . . . . .	5
10.2.4 Mise en garde sur la décomposition . . . . .	7
10.3 Première et deuxième formes normales . . . . .	8
10.3.1 Définitions . . . . .	8
10.3.2 Autre définition de la 3ème forme normale . . . . .	8
10.4 Conclusion sur les formes normales . . . . .	9
<b>Bibliographie</b>	<b>10</b>

# Chapitre 10

## Formes normales

Afin d'être certain de construire un "bon schéma", la théorie de la normalisation a été proposée. Cette théorie permet de [16] :

- minimiser les données redondantes,
- éviter les anomalies de mise à jour,
- réduire les données incohérentes,
- concevoir des structures de données pour faciliter la maintenance.

La théorie de la normalisation permet de décrire la structure souhaitée pour les relations d'une base de données selon des **formes normales**.

### 10.1 BCNF (*Boyce-Codd Normal Form*)

#### 10.1.1 Définition

Un schéma de relation est **BCNF** si et seulement si les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé entière détermine un attribut [9].

Un schéma de relation est donc BCNF ssi pour toute dépendance fonctionnelle non triviale  $X \rightarrow A$  avec  $A \not\subseteq X$  alors  $X$  est une surclé (i.e. est une clé ou contient une clé) **et**  $A$  n'est pas une partie de clé.

Par exemple (repris de [9]) la relation Localisation(Cru,Pays,Region,Qualité) associée aux dépendances fonctionnelles suivantes :

$Cru, Pays \rightarrow Region$

$Cru, Pays \rightarrow Qualite$

$Region \rightarrow Pays$

n'est pas BCNF car les clés minimales possibles sont  $(Cru, Pays)$  et  $(Cru, Rgion)$ .

La dépendance fonctionnelle  $Region \rightarrow Pays$  n'est donc pas de la forme, un clé (ou une surclé) détermine un attribut non clé.

#### 10.1.2 Propriétés d'une relation de schéma BCNF

**Pas de redondance d'information** :

La forme normale BCNF évite la redondance d'information : aucune redondance d'information ne peut être détectée en utilisant les dépendances fonctionnelles [18]. En effet, soit  $R$  un schéma de relation BCNF et deux nuplets identiques sur un ensemble d'attributs  $X$  et différents sur un ensemble d'attributs  $Y$ . Soit

TABLE 10.1 – Pas de redondance d'information pour la forme normale BCNF.

X	Y	A
$x$	$y_1$	$a$
$x$	$y_2$	?

A un attribut tel que sa valeur dans le premier nuplet soit  $a$  (voir tableau 10.1).

Il y aurait redondance d'information si on avait la dépendance fonctionnelle  $X \rightarrow A$ , car de cette dépendance on pourrait déduire que la valeur de l'attribut  $A$  dans le deuxième nuplet est  $a$ . Or, si la relation est BCNF et que la dépendance fonctionnelle  $X \rightarrow A$  est associée à la relation alors  $X$  est une surclé. Si  $X$  est une surclé, alors on a forcément  $y_1 = y_2$ , et les deux nuplets sont égaux. Or une relation ne peut pas contenir deux nuplets égaux. Par conséquent, une telle situation ne peut pas apparaître dans une relation BCNF.

**Remarque** : Tout schéma de relation ayant deux attributs est BCNF.

Prenons par exemple la relation  $R(A, B)$ .

Les seules familles de dépendances non triviales pouvant être associées à cette relation sont :

$F_1 = \{A \rightarrow B\}$  ou  $F_2 = \{B \rightarrow A\}$  ou  $F_3 = \{A \rightarrow B, B \rightarrow A\}$  ou  $F_4 = \emptyset$ .

Si la famille associée est  $F_1$  alors la clé de la relation est  $A$  et  $R$  est BCNF.

Si la famille associée est  $F_2$  alors la clé de la relation est  $B$  et  $R$  est BCNF.

Si la famille associée est  $F_3$  alors la clé de la relation est  $A$  ou  $B$  et  $R$  est BCNF ( $A \rightarrow B$  possède une surclé à gauche lorsque la clé est  $A$  et  $B \rightarrow A$  possède une surclé à gauche lorsque la clé est  $B$ ).

Si la famille associée est  $F_4$  alors la clé de la relation est  $AB$  et  $R$  est BCNF.

**Relation non BCNF** :

Si  $R$  n'est pas BCNF alors il existe une dépendance fonctionnelle  $X \rightarrow A$  non triviale ( $A \not\subseteq X$ ) telle que  $X$  n'est pas une surclé.

### 10.1.3 Décomposition de schéma BCNF sans perte d'information

**Algorithme**

**Entrée** :  $R$  un schéma de relation et  $F$  une famille minimale de dépendances fonctionnelles associées à  $R$ , non BCNF.

**Sortie** : une décomposition SPI de  $R$  en  $n$  relations BCNF

**Principe** :

1. On choisit  $X \rightarrow A$  telle que la relation  $R$  soit récursivement décomposée en deux relations :  
 $R_1(X, A)$  à laquelle on associe  $F = \{X \rightarrow A\} \cup \{\text{les DFs portant sur } X \text{ et } A\}$ , BCNF  
et  $R_2(Z - \{A\})$  où  $Z$  est l'ensemble des attributs de  $R$ .
2. Si le schéma de la relation  $R_2$  n'est pas BCNF alors on la décompose à son tour en appliquant l'étape 1.

Cette décomposition est sans perte d'information. En effet, à chaque étape de la décomposition, on prend une dépendance fonctionnelle  $X \rightarrow A$ , et on crée  $R_1(X, A)$  et  $R_2(Z - A)$ . La jointure entre  $r_1$  et  $r_2$  va se faire naturellement sur  $X$ , donc comme on a  $X \rightarrow A$ , on va forcément avoir que  $r = r_1 \bowtie r_2$ .

**Remarque** : La décomposition BCNF sans perte d'information MAIS ne préserve pas forcément les dépendances fonctionnelles.

Par exemple la relation Localisation(Cru,Pays,Region,Qualité) précédemment citée n'est pas BCNF. En effet, les dépendances fonctionnelles sont :

$Cru, Pays \longrightarrow Region$ ,  $Cru, Pays \longrightarrow Qualite$  et  $Region \longrightarrow Pays$ .

La dépendance qui viole la forme normale BCNF est  $Region \longrightarrow Pays$ .

On peut donc décomposer la relation en deux relations BCNF, en appliquant l'algorithme de décomposition BCNF. On part de  $Cru, Pays \longrightarrow Qualite$  (car si on part de  $Cru, Pays \longrightarrow Region$ ,  $F_1$  va contenir les 2 premières DF de  $F$  et ne sera pas BCNF) :

$R_1(\underline{Cru}, \underline{Pays}, \underline{Qualité})$  avec  $F_1 = \{Cru, Pays \longrightarrow Qualite\}$

$R_2(\underline{Cru}, \underline{Pays}, \underline{Région})$  avec  $F_2 = \{Cru, Pays \longrightarrow Region; Region \longrightarrow Pays\}$  avec  $K_2 = (Cru, Pays)$  ou  $(Cru, Region)$  non BCNF à cause de  $Region \longrightarrow Pays$ .

On décompose donc  $R_2$  : on prend  $Region \longrightarrow Pays$  car  $Cru, Pays \longrightarrow Region$  ne permet pas de décomposer  $R_2$ .

$R_{21}(\underline{Region}, \underline{Pays})$  avec  $F_{21} = \{Region \longrightarrow Pays\}$  BCNF

$R_{22}(\underline{Cru}, \underline{Region})$  avec  $F_{22} = \emptyset$  BCNF

Mais la dépendance  $Cru, Pays \longrightarrow Region$  est perdue.

Par conséquent, on préférera, dans certains cas, une décomposition en 3ème forme normale qu'en BCNF, car la décomposition en 3ème forme normale est sans perte d'information et sans perte de dépendance.

## 10.2 Troisième Forme Normale

### 10.2.1 Définitions

Un schéma de relation est en **3ème forme normale** si tout attribut n'appartenant pas à un clé ne dépend pas d'un autre attribut non clé.

Un schéma de relation  $R$  muni d'une famille minimale de dépendances fonctionnelles  $F$  est en 3ème forme normale ssi pour toute dépendance non triviale de  $F$ ,  $X \longrightarrow A$ , alors on a  $X$  est une surclé **ou**  $A$  appartient à une clé.

La relation Localisation(Cru,Pays,Region,Qualité) est en 3ème forme normale puisque dans la dépendance  $Region \longrightarrow Pays$ , l'attribut  $Pays$  appartient à une clé  $(Cru, Pays)$ .

**Remarque** : Toute relation en forme normale BCNF est en 3ème forme normale (mais l'inverse n'est évidemment pas vrai). La 3ème forme normale est moins exigeante que la forme normale BCNF.

#### Relation dont le schéma n'est pas 3NF :

Une relation n'est pas en 3ème forme normale signifie qu'il existe une dépendance fonctionnelle  $X \longrightarrow A$  telle que  $\forall K \neg(X \text{ est une surclé ou } A \text{ appartient à une clé})$ , c'est-à-dire telle que  $X$  n'est pas une surclé et  $A$  n'appartient pas à une clé. Deux cas apparaissent :

1.  $X$  est une partie de clé ( $\forall K$ ) mais  $A$  ne l'est pas : on dit que  $X \longrightarrow A$  est une **dépendance fonctionnelle partielle**.
2.  $X$  n'est pas une partie de clé ( $\forall K$ ) et  $A$  non plus : on dit que  $X \longrightarrow A$  est une **dépendance fonctionnelle transitive**.

TABLE 10.2 – Une instance de la relation *Localisation*

Cru	Pays	Region	Qualite
<i>Gewurtz</i>	<i>France</i>	<i>Alsace</i>	<i>Extra</i>
<i>Riesling</i>	?	<i>Alsace</i>	<i>Extra</i>

Dans le deuxième cas, on dit que la dépendance est transitive, car par exemple, s'il existe  $A \rightarrow C$  et  $C \rightarrow B$  alors  $B$  est dépendant de manière transitive de  $A$  via  $C$ . La dépendance  $A \rightarrow B$  existe par transitivité via l'attribut  $C$  [4].

Une relation  $R$  est 3NF si aucun attribut non clé n'est en dépendance transitive avec les clés.

Par exemple la relation  $R(\text{Élève}, \text{Professeur}, \text{Matière}, \text{Note})$ , de famille de dépendances fonctionnelles  $\{E, M \rightarrow P, N; P \rightarrow M\}$  est 3NF mais non BCNF (à cause de  $P \rightarrow M$ , les clés possibles étant  $EM$  ou  $EP$ ). La dépendance  $P \rightarrow M$  possède un attribut clé à droite lorsque  $EM$  est clé.

### 10.2.2 Redondance

La 3ème forme normale n'empêche pas la redondance d'information. En effet prenons par exemple une instance de la relation *Localisation* (voir tableau 10.2), il est possible de déterminer la valeur remplaçant le point d'interrogation, qui est *France* car  $Region \rightarrow Pays$ . Tous les nuplets correspondant à la même région auront donc la même valeur pour l'attribut *Pays*.

Il est à noter en revanche que si plusieurs les nuplets sont associés à la même valeur de l'attribut *Qualite*, ce n'est pas de la redondance d'information. En effet, cette même valeur est associé à différents couples  $(Cru, Pays)$  ou  $(Cru, Region)$ .

### 10.2.3 Décomposition 3NF sans perte de dépendances

**Entrée** :  $R$  un schéma de relation et  $F$  une famille minimale de dépendances fonctionnelles associées à  $R$ , non 3NF (ou 3NF mais non BCNF).

**Sortie** : une décomposition SPI et SPD de  $R$  en  $n$  relations 3NF ou BCNF

**Principe** :

La décomposition de  $R$  en relations en 3ème forme normale consiste :

1. à associer à chaque dépendance  $X \rightarrow A$  une relation de la décomposition contenant les attributs  $X$  et  $A$ ;
2. et à créer une dernière relation contenant une clé minimale de  $R$  si aucune relation issue de l'étape 1 ne contient de clé minimale de  $R$ .

**Attention** : Si aucune relation issue de l'étape 1 ne contient de clé minimale de  $R$ , il ne faut surtout pas oublier de faire l'étape 2 (i.e. de créer une dernière relation contenant une clé de  $R$ ), sinon la décomposition n'est pas SPI.

**Exemple 1** :

Soit la relation :

$\text{Banque}(\text{Agence}, \text{\#Emprunt}, \text{Montant}, \text{\#Compte}, \text{Solde}, \text{Client})$  et les dépendances fonctionnelles suivantes :

$\#Emprunt \longrightarrow Montant$   
 $\#Emprunt \longrightarrow Agence$   
 $\#Compte \longrightarrow Solde$   
 $\#Compte \longrightarrow Agence$

la clé de la relation est  $(\#Emprunt, \#Compte, Client)$ .  $R$  n'est donc pas 3NF car les dépendances fonctionnelles sont partielles.

Une décomposition 3NF sans perte d'information ni perte de dépendances est :

1.  $R_1(\#Emprunt, Montant)$ ,  $R_2(\#Emprunt, Agence)$ ,  $R_3(\#Compte, Solde)$ ,  $R_4(\#Compte, Agence)$
2. et  $R_5(\#Emprunt, \#Compte, Client)$

Cette décomposition permet d'obtenir en réalité des relations dont le schéma est BCNF. Elle est plus intéressante que celle que nous aurions obtenue en suivant l'algorithme de décomposition BCNF, car elle est sans perte de dépendance et sans perte d'information.

En effet en décomposant via l'algorithme de décomposition BCNF nous aurions obtenu :

$R_1(\#Emprunt, Montant)$  et  $R_2(Agence, \#Emprunt, \#Compte, Solde, Client)$   
 Puis  $R_{21}(\#Emprunt, Agence)$  et  $R_{22}(\#Emprunt, \#Compte, Solde, Client)$   
 Puis  $R_{221}(\#Compte, Solde)$  et  $R_{222}(\#Emprunt, \#Compte, Client)$   
 La dépendance fonctionnelle  $\#Compte \longrightarrow Agence$  aurait donc été perdue.

**Fusion de relations pour diminuer le nombre de relations issues de la décomposition :** Il est possible de diminuer le nombre de relations issues de la décomposition en fusionnant certaines relations, à condition de conserver des relations de schéma BCNF ou au minimum 3NF.

Par exemple,  $R_1$  et  $R_2$  dans l'exemple précédent peuvent être fusionnées en  $R'_1(\#Emprunt, Montant, Agence)$ , qui est BCNF, et  $R_3$  et  $R_4$  peuvent être fusionnées en  $R'_2(\#Compte, Solde, Agence)$ , qui est BCNF.

### Exemple 2 :

Considérons la relation  $R(A, B, C, D, E, G)$  et une famille minimale  $F$  contenant les dépendances fonctionnelles suivantes :

$AB \longrightarrow C$   
 $C \longrightarrow A$   
 $D \longrightarrow E$   
 $AB \longrightarrow G$

Les clés minimales possibles sont  $(A, B, D)$  et  $(B, C, D)$ . La relation n'est donc pas BCNF, puisque il y a plusieurs dépendances fonctionnelles partielles. Elle n'est pas non plus en 3ème forme normale (ex.  $D \longrightarrow E$  n'a pas de clé à gauche ni d'attribut appartenant à une clé à droite).

Application de l'algorithme de décomposition 3NF :

On décompose donc la relation en :

$R_1(A, B, C)$  dont les dépendances fonctionnelles associés sont  $AB \longrightarrow C$  et  $C \longrightarrow A$ , dont le schéma est donc 3NF.

$R_2(D, E)$  de dépendance fonctionnelle  $D \longrightarrow E$ , BCNF.

$R_3(A, B, G)$  de dépendance fonctionnelle  $AB \longrightarrow G$ , BCNF.

$R_4(A, B, D)$ , avec  $F_4 = \emptyset$  BCNF, car aucune des relations précédentes ne contiennent de clé minimales de  $R$ .

$R_1$  est en troisième forme normale, car la dépendance fonctionnelle  $C \rightarrow A$  n'est pas telle que  $C$  est une surclé et  $A$  n'appartient pas à une clé. Les autres sont BCNF.

Comme  $R_4$  et  $R_1$  ont des dépendances fonctionnelles ayant la même partie gauche ( $AB \rightarrow C$  et  $AB \rightarrow C$ ), il est possible de fusionner leur schéma. On obtient par conséquent la décomposition de  $R$  en 3 relations seulement :

$R'_1(A, B, C, G)$  dont les dépendances fonctionnelles sont  $AB \rightarrow GC$  et  $C \rightarrow A$ .

$R'_2(D, E)$  de dépendance fonctionnelle  $D \rightarrow E$ .

$R'_3(A, B, D)$ .

$R'_1$  est en 3ème forme normale. Les autres relations sont BCNF. On a donc une décomposition SPI et SPD en 3ème forme normale dont le nombre de relations obtenues est minimal.

On peut noter qu'il n'est pas possible dans cet exemple de décomposer  $R$  en relations qui sont toutes BCNF en ayant une décomposition SPI et SPD. Si on applique l'algorithme de décomposition BCNF on obtient en revanche uniquement des relations BCNF mais on perd des dépendances.

Application de l'algorithme de décomposition BCNF :

Si on applique l'algorithme de décomposition BCNF, on ne peut pas commencer par prendre la dépendance fonctionnelle  $AB \rightarrow C$  et créer  $R_1(A, B, C)$  BCNF car  $C \rightarrow A$  s'applique également à  $R_1$  (qui est donc 3NF). Il faut donc commencer par une autre dépendance, par exemple  $D \rightarrow E$ .

On a donc :

$R_1(D, E)$  associée à  $F_1 = \{D \rightarrow E\}$ , BNCF

Et  $R_2(A, B, C, D, G)$  associée à  $F_2 = \{AB \rightarrow C; C \rightarrow A; AB \rightarrow G\}$ ,  $\neg$ BCNF.

On décompose  $R_2$ , en prenant  $AB \rightarrow G$ , et on obtient :

$R_{21}(A, B, G)$  associée à  $F_{21} = \{AB \rightarrow G\}$ , BNCF

Et  $R_{22}(A, B, C, D)$  associée à  $F_{22} = \{AB \rightarrow C; C \rightarrow A\}$ ,  $\neg$ BCNF.

On décompose  $R_{22}$ , en prenant  $C \rightarrow A$ , et on obtient :

$R_{221}(A, C)$  associée à  $F_{221} = \{C \rightarrow A\}$ , BNCF

Et  $R_{222}(B, C)$  associée à  $F_{222} = \emptyset$ , BCNF.

On a donc au final :  $R$  décomposée en 4 relations BCNF,  $R_1(D, E)$ ,  $R_{21}(A, B, G)$ ,  $R_{221}(A, C)$  et  $R_{222}(B, C)$  mais on a perdu la dépendance  $AB \rightarrow C$ .

#### 10.2.4 Mise en garde sur la décomposition

Il faut faire attention de ne pas décomposer une relation qui est déjà BNCF ou 3NF. Par exemple si on a la relation **Etudiant**(#Etudiant, Nom, Prenom, Adresse) avec les dépendances fonctionnelles suivantes :

#*Etudiant*  $\rightarrow$  *Nom*

#*Etudiant*  $\rightarrow$  *Prenom*

#*Etudiant*  $\rightarrow$  *Adresse*

Il est inutile de décomposer la relation car elle est BNCF. Une décomposition en trois relations en-

traînerait une répétition de la clé (Attention : ce n'est pas de la redondance<sup>1</sup>, mais on perd de la place) et obligerait à utiliser des jointures dans les requêtes.

Les trois objectifs principaux lors de conception d'un schéma relationnel sont [19] :

1. **La forme normale de Boyce-Codd**
2. **La non perte d'information**
3. **La non perte de dépendances**

Si ces trois objectifs ne peuvent être atteints alors on préférera sacrifier la forme normale BNCF et avoir :

1. **La 3ème forme normale**
2. **La non perte d'information**
3. **La non perte de dépendances**

## 10.3 Première et deuxième formes normales

### 10.3.1 Définitions

Un schéma de relation est en **première forme normale** si tout attribut contient une valeur atomique (c'est la définition de relation donnée au début du cours de BD). Toute relation du modèle relationnelle est en 1NF.

Le fait que le schéma de relation soit en 1ère forme normale dépend aussi du domaine de l'attribut. Si par exemple on souhaite connaître, pour chaque personne, la liste de ses enfants, et ceci dans une seule relation, on peut vouloir créer un domaine pour l'attribut *Enfant* qui soit une liste de Personnes (c'est ce qui est fait en relationnel objet). Dans ce cas, le schéma de la relation est bien en 1ère forme normale car l'attribut *Enfant* prend une valeur atomique dans le domaine liste de personnes.

Un schéma de relation est en **2ème forme normale** si et seulement si [9] :

- Le schéma est en 1ère forme normale.
- Tout attribut n'appartenant pas à une clé ne dépend pas d'une partie de clé, c'est-à-dire s'il n'existe pas de dépendance partielle.

Par exemple [9], la relation `Fournisseur(Nom, Adresse, Article, Prix)` associée aux dépendances fonctionnelles suivantes :  $Nom, Article \rightarrow Prix$  et  $Nom \rightarrow Adresse$  n'est pas en deuxième forme normale. En effet, la clé de la relation étant  $(Nom, Article)$ , la dépendance fonctionnelle  $Nom \rightarrow Adresse$  est partielle.

### 10.3.2 Autre définition de la 3ème forme normale

Un schéma de relation est en 3ème forme normale si et seulement si il est en 2ème forme normale et s'il n'a pas de dépendance fonctionnelle transitive, c'est-à-dire une dépendance  $X \rightarrow A$  telle que  $X$  et  $A$  soient des attributs non clé.

Par exemple la relation  $R(Magasin, Article, Rayon, Chef)$  dont les dépendances fonctionnelles sont :  $Magasin, Article \rightarrow Rayon$  et  $Rayon \rightarrow Chef$

Cette relation ne possède pas de dépendance partielle, elle est donc en 2ème forme normale, mais elle possède une dépendance transitive  $Rayon \rightarrow Chef$ . Elle n'est donc pas en 3ème forme normale. On

1. Si on a une relation  $R(X, Y, A)$ , cette relation est redondance s'il existe une dépendance fonctionnelle  $X \rightarrow A$  et si une même valeur de  $A$  est répétée plusieurs fois dans  $R$  : en effet, à une valeur de  $X$  correspond une valeur de  $A$ .



aurait pu le voir en utilisant la première définition de la 3ème forme normale, puisque *Chef* n'appartient pas à une clé et *Rayon* n'est pas une clé.

## 10.4 Conclusion sur les formes normales

Il est possible de vouloir gérer des relations qui ne sont pas en première forme normale, par exemple lorsque l'on souhaite obtenir les enfants de toutes les personnes de la base. Dans ce cas on utilisera le relationnel-objet.

La 1ère forme normale correspond à toutes les notions nécessaires à la théorie du modèle relationnel et à son implantation sur machine.

La deuxième forme normale est nécessaire mais pas suffisante.

La 3ème forme normale est désirable si on ne peut pas obtenir de forme BCNF.

La forme normale BCNF est la meilleure en particulier en ce qui concerne la redondance d'information (si on tient compte uniquement de l'information contenue dans les dépendances fonctionnelles).

# Bibliographie

- [1] D. Austin, *Using Oracle8<sup>TM</sup>*, Simple Solutions - Essential Skills, QUE, 1998, ISBN : 0-7897-1653-4
- [2] R. Chapuis, *Oracle 8*, Editions Dunes et Laser, 1998, ISBN : 2-913010-07-5
- [3] P. Chen, *The Entity-Relationship Model—Toward a Unified View of Data*, ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, Pages 9-36, [http ://www.csc.lsu.edu/~chen/pdf/erd.pdf](http://www.csc.lsu.edu/~chen/pdf/erd.pdf)
- [4] T. Connolly, C. Begg et A. Strachan, *Database Systems - A pratical Approach to Design, Implementation and Management*, Addison-Wesley, 1998, ISBN : 0-201-34287-1, disponible à la BU 055.7 CON
- [5] C.J. Date, *Introduction aux bases de données*, 6ème édition, Thomson Publishing, 1998, ISBN : 2-84180-964-1, disponible à la BU 005.7 DAT
- [6] R. Elamsri et S.B. Navathe, *Fundamentals of Database Systems*, 3ème édition, Addison Wesley-disponible à la BU 005.7 ELM
- [7] P. Delmal, *SQL2 - Application à Oracle, Access et RDB*, 2ème édition, Bibliothèque des Universités - Informatique, De Boeck Université, 1988, ISBN : 2-8041-2995-0,disponible à la BU 005.74 SQL
- [8] S. Feuerstein, B. Pribyl et C. Dawes, *Oracle PL/SQL - précis et concis*, O'Reilly, 2000, ISBN : 2-84177-108-3
- [9] G. Gardarin, *Bases de Données - objet & relationnel*, Eyrolles, 1999, ISBN : 2-212-09060-9, disponible à la BU 005.74 GAR
- [10] R. Grin, *Introduction aux bases de données, modèle relationnel*, Université Sophia-Antipolis, jan. 2000
- [11] R. Grin, *Langage SQL*, Université Sophia-Antipolis, jan. 2000
- [12] G. Gardarin et O. Gardarin *Le Client-Serveur*, Eyrolles, 1999, disponible à la BU
- [13] H. Garcia-Molina, J.D. Ulmann et J. Widow, *Database SYstem Implementation*, Prentice Hall, 2000, ISBN :0-13-040264-8, disponible à la BU 005.7 GAR
- [14] H. Garcia-Molina, J.D. Ulmann et J. Widow, *Database Systems - The Complete Book*, Prentice Hall, 2002, ISBN :0-13-031995-3
- [15] S. Krakowiak, *Gestion Physique des données*, Ecole Thématique "Jeunes Chercheurs" en Base de Données, Volume 2, Port Camargue, mars 1997
- [16] D. Lockman, *Oracle8<sup>TM</sup> Développement de bases de données*, Le programmeur - Formation en 21 jours, Editions Simon et Schuster Macmillan (S&SM), 1997, ISBN : 2-7440-0350-6, disponible à la BU 005.74 ORA
- [17] P.J. Pratt, *Initiation à SQL - Cours et exercices corrigés*, Eyrolles, 2001, ISBN : 2-212-09285-7
- [18] R. Ramakrishnan et J. Gehrke, *Database Management Systems*, Second Edition ; McGraw-Hill, 2000, ISBN : 0-07-232206-3, disponible à la BU 055.7 RAM
- [19] A. Silberschatz, H.F. Korth et S. Sudarshan, *Database System Concepts*, Third Edition, McGraw-Hill, 1996, ISBN : 0-07-114810-8, disponible à la BU 005.7 DAT

- 
- [20] C. Soutou, *De UML à SQL - Conception de bases de données*, Eyrolles, 2002, ISBN : 2-212-11098-7
  - [21] J.D. Ullman et J. Widom, *A first Course in Database Systems*, Prentice Hall, 1997, ISBN : 0-13-887647-9, disponible à la BU 005.7 ULL