VII. CONCEPTION AVEC LE MODÈLE RELATIONNEL

1. Le modèle relationnel

a) *Rappels des concepts*

Une relation peut être définie en extension ou en intention.

Définition en *extension* = vision tabulaire (*tuple*, *attribut*, *domaine*).
Notation : $R(A_1{:}D_1, A_2{:}D_2, \ldots, A_n{:}D_n)$ ou $R\ \{A_1, A_2, \ldots, A_n\}$
$R \in (\ dom(A_1)\ \mathbf{x}\ dom(A_2)\ \mathbf{x}\ ...\ \mathbf{x}\ dom(A_n)\ )$
Tuple : $t = <V_1, V_2, ..., V_n>$ où V_i = valeur $\in$ $dom(A_i)$ ou valeur nulle.

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 prédicat [R(t)], qui pour chaque tuple t, fournit une valeur vraie ou fausse, permettant de décider de l'appartenance de t à R. R(t) = assertion équivalente à l'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.

⇒ Problème de l'équivalence entre deux schémas relationnels. Intuitivement, deux schémas relationnels seront dits *équivalents* s'ils assurent la même représentation sémantique du monde réel.

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
101	10	DURAND	10 , Rue des Gras - Clermont
102	20	DUPONT	86 , Rue de la République - Moulins
101	30	BRUNEAU	26 , Rue des Dômes - Vichy
103	10	DURAND	10 , Rue des Gras - Clermont

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° 102 fait perdre toute information concernant le fournisseur DUPONT.

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	NOMFOUR	QUANTITE
101	BRUNEAU	30
101	DURAND	10
102	DUPONT	20
103	DURAND	10

FOURNISSEUR

NOMFOUR	ADRESSE
BRUNEAU	26 , Rue des Dômes - Vichy
DUPONT	86 , Rue de la République - Moulins
DURAND	10 , Rue des Gras - Clermont

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 $R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n$ constitue un schéma relationnel équivalent à R.

Attention ! Des décompositions arbitraires peuvent entraîner des pertes de propriétés du schéma relationnel initial.

<u>Exemple 1</u> : Décomposition de la relation EMPLOYE.

La recomposition par jointure de E1 et E2 introduit des tuples supplémentaires non valides. On dit qu'on a une jointure *avec perte*. D'une façon générale, toute jointure basée sur des attributs non clés est avec perte.

<u>Exemple 2</u> : Décomposition de la relation COMPAGNIE (VOL, AVION, PILOTE) avec les hypothèses suivantes :

- un seul avion est affecté à un vol ;
- un avion est toujours piloté par la même personne.

EMPLOYE

MATRICULE	NOM	AGE	SERVICE
1100	DUPONT Charles	31	Comptabilité
19	DURAND Jean	46	Production
110	HARDY Lucie	31	Informatique
102	BRUNEAU Yves	40	Informatique

PROJECTION

E1

MATRICULE	NOM	AGE
1100	DUPONT Charles	31
19	DURAND Jean	46
110	HARDY Lucie	31
102	BRUNEAU Yves	40

E2

AGE	SERVICE
31	Comptabilité
46	Production
31	Informatique
40	Informatique

JOINTURE

MATRICULE	NOM	AGE	SERVICE
1100	DUPONT Charles	31	Comptabilité
1100	DUPONT Charles	31	Informatique
19	DURAND Jean	46	Production
110	HARDY Lucie	31	Comptabilité
110	HARDY Lucie	31	Informatique
102	BRUNEAU Yves	40	Informatique

Exemple de recomposition par une jointure avec perte

COMPAGNIE

VOL	AVION	PILOTE
102	FOK	HARDY
210	FOK	HARDY
300	BOE	ROUSSEL
400	CAR	ROUSSEL

R1

VOL	AVION
102	FOK
210	FOK
300	BOE
400	CAR

R2

VOL	PILOTE
102	HARDY
210	HARDY
300	ROUSSEL
400	ROUSSEL

Décomposition non équivalente d'un schéma

Si on désire dans R1 effectuer une modification consistant à affecter l'avion FOK au vol n° 400, on constate en recombinant les deux relations R1 et R2, que l'avion FOK a maintenant deux pilotes. Cette décomposition n'est donc pas acceptable.

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 traitements.

2. Les dépendances fonctionnelles (DF)

<u>Notations</u> : 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'.

<u>Exemple</u> : relation COMPAGNIE (VOL, AVION, PILOTE)

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

b) Propriétés des dépendances fonctionnelles

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

Les trois premières propriétés sont appelées règles d'inférence d'Armstrong. Elles forment un ensemble valide et complet.

- *Validité* : étant donné un ensemble F de dépendances vérifiées sur un schéma R, alors toute dépendance inférée de F grâce à ces propriétés est aussi vérifiée sur R.
- *Complétude* : toute dépendance qui est la conséquence d'un ensemble F, peut être obtenue par applications répétées de ces trois règles.

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.

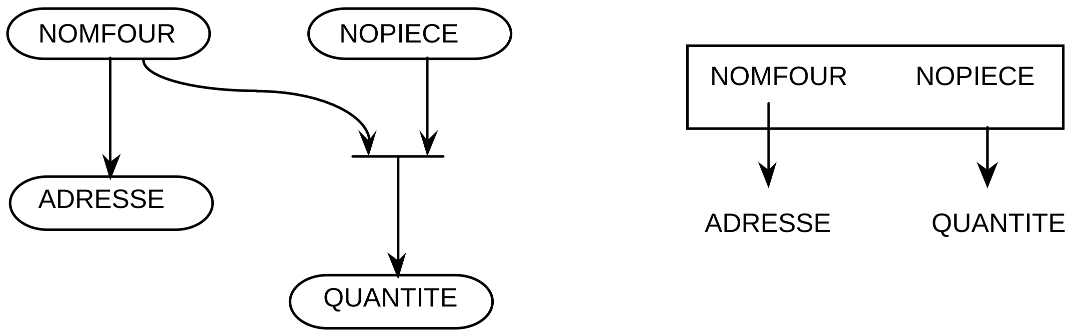
Une dépendance fonctionnelle est *simple* si elle ne comporte qu'un seul attribut en partie droite et si elle n'est pas triviale.

$X \rightarrow A_1, A_2, ..., A_n \Leftrightarrow \{ X \rightarrow A_1 ; X \rightarrow A_2 ; ... ; X \rightarrow A_n \}$

(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 ➔ 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 ou *couverture irredondante* de D = sous-ensemble D° de D tel que pour toute dépendance d ∈ 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 PERSONNE (NOM, PRENOMS, AGE) et DEPARTEMENT (NOM, ADRESSE, TELEPHONE) ne sont pas en 1FN si les attributs PRENOMS et ADRESSE peuvent être du type [Jean, Paul] ou [Avenue des Landais, Aubière].

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 → C, D, E ;
B → 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 → QUANTITE ;
NOMFOUR → 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 → AVION,
AVION → PILOTE,
VOL → 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 (<u>A, B</u>, 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 attributs non clé (toutes les DF sont directes). Autrement dit, R est en 3FN s'il n'existe aucune dépendance entre deux attributs non clés. Comme R possède plusieurs clés, la définition doit être vérifiée pour chaque clé.

Il faut donc éviter la configuration suivante :

A, B → C, D, E ;
C → E.

<u>Exemple</u> : La relation COMPAGNIE(<u>VOL</u>, AVION, PILOTE) du b) n'est pas en 3FN. La décomposition

R1 (<u>VOL</u>, AVION)
R2 (<u>VOL</u>, PILOTE)

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

R1 (<u>VOL</u>, AVION)
R3 (<u>AVION,</u> 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.

<u>Exemple</u> : Dans une relation VENTES, si on ajoute un nouveau VRP qui vend *a* produits pour *b* entreprises fabriquant chacune ces produits, il faut ajouter *a* + *b* nouveaux enregistrements dans la forme normalisée, mais *a* x *b* nouveaux enregistrements dans la forme non normalisée.

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 :

$$R_1 \bowtie R_2 \bowtie \dots \bowtie R_n = R.$$

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.

THEOREME : Une décomposition R = {R1, R2} préserve le contenu si et seulement si :

$R1 \cap R2 \rightarrow R1 - R2$

et $R1 \cap R2 \rightarrow R2 - R1$.

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 $R_i = <U_i, D_i>$. 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 $<K, \emptyset>$.

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

A : numéro de fournisseur
B : nom de fournisseur
C : numéro de pièce

Q : quantité commandée
E : désignation de la pièce
F : adresse du fournisseur.

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.

5. Processus de modélisation

a) Remarques préliminaires

Toute la sémantique est exprimée à travers les dépendances sur un ensemble d’attributs. Or, il n’est pas forcément aisé de déterminer les dépendances. L’ajout ou la suppression d’une dépendance peut changer complètement la décomposition.

⇒ Problèmes : modélisation des rôles, modélisation des sous-classes, modélisation des associations M-N.

b) Modélisation des rôles

Un attribut doit être introduit pour chaque rôle.

Un attribut de rôle prend ses valeurs dans le domaine de l'attribut générique. Cette contrainte est spécifiée par une dépendance d'inclusion (par exemple, pour la base FABRICATION on a NOPA $\subset$ NOP et NOPC $\subset$ NOP).

Les dépendances sont alors exprimées sur l'ensemble de tous les attributs, y compris les attributs de rôles.

<u>Exemple</u> : la base FABRICATION.

NOP $\rightarrow$ DESIGNATION, COULEUR, POIDS
NOS $\rightarrow$ INTITULE, LOCALISATION
NOP, NOS $\rightarrow$ QUANTITE1
NOPA, NOPC$\rightarrow$ QUANTITE2

L'algorithme d'agrégation donne le schéma relationnel normalisé 3FN suivant.

PIECE (<u>NOP</u>, DESIGNATION, COULEUR, POIDS)
SERVICE (<u>NOS</u>, INTITULE, LOCALISATION)
ORDRE (<u>*NOP*</u>, <u>*NOS*</u>, QUANTITE1)
NOMENCLATURE (<u>*NOPA*</u>, <u>*NOPC*</u>, QUANTITE2).

c) Modélisation des sous-classes

Une sous-classe C de la classe G est modélisée par une relation C séparée dont la clé prend ses valeurs sur le domaine de la clé de G. Il est nécessaire d'introduire un identifieur pour la sous-classe et de spécifier la dépendance d'inclusion correspondante. Les dépendances fonctionnelles doivent être établies sur l'ensemble des attributs, y compris les attributs identifieurs.

<u>Exemple</u> :

On souhaite mettre en évidence dans la base FABRICATION la sous-classe des pièces obsolètes. Un attribut NOPO est donc introduit pour identifier la sous-classe. Introduisons un autre attribut, DATE-PEREMPTION, pour mémoriser la date de péremption des pièces obsolètes. Il n'est valide que pour la sous-classe. On a maintenant une dépendance fonctionnelle supplémentaire :

NOPO → DATE-PEREMPTION.

Schéma relationnel normalisé 3FN obtenu par l'algorithme d'agrégation :

PIECE (NOP, DESIGNATION, COULEUR, POIDS)
SERVICE (NOS, INTITULE, LOCALISATION)
ORDRE (*NOP*, *NOS*, QUANTITE1)
NOMENCLATURE (*NOPA*, *NOPC*, QUANTITE2)
PIECE-OBS (*NOPO*, DATE-PEREMPTION)

6. Exercices

I. On considère la relation R (A, B, C) avec l'ensemble de DF { $A \rightarrow B$; $B \rightarrow C$ }.

1) Quelle est la clé primaire de R ? Dans quelle forme normale se trouve cette relation ?
2) L'extension de la relation R' suivante est-elle une extension de R ?

R'	A	B	C
	A1	B1	C1
	A2	B1	C2
	A3	B2	C1
	A4	B3	C3

3) Trouver une extension R'' conforme à R, à partir de R'.
4) Proposer une décomposition en 3FN de R sans perte d'information.

II. Soit le schéma relationnel R = <U, D>.

U = {A, B, C, D, E, G, H, I, J, K, L}
D = {A, B → C ; B → D, E ; B, D, L → K ; B, H, J, L → C ; C → A, B ;
C, E, L → K ; C, I, L → G, K ; D → B, E ; E, I, K → C, G, L}

1) Donner une couverture irredondante des dépendances.
2) Proposer une normalisation 3FN.

III. Une entreprise comprend différents services, chacun étant caractérisé par un numéro (NOSER) et un nom (NOMS) de service supposés uniques et par le numéro (NORES) et le nom (NOMR) de son responsable. Un budget (BUSER) est attribué à un service.

Chaque service gère un ou plusieurs projets, mais un projet est géré par un seul service. Un projet est caractérisé par un numéro (NOPRO) supposé unique et un nom (NOMP). Un budget (BUPRO) est attribué à un projet.

Les employés de l'entreprise sont affectés à un instant donné à un seul projet. Un employé est caractérisé par un numéro (NOEMP) supposé unique et un nom (NOME). Chaque employé peut être joint par l'intermédiaire d'un numéro de téléphone (NOTEL). Un numéro de téléphone peut être partagé entre plusieurs employés.

Un employé est installé dans un bureau caractérisé par un numéro unique (NOBUR). Un bureau peut accueillir plusieurs employés et plusieurs appareils téléphoniques. La localisation d'un bureau est repérée par le nom de son bâtiment (NOMB). Un bureau est rattaché pour gestion à un seul service.

1) Déterminer les dépendances fonctionnelles qui constituent la couverture irredondante des dépendances existant sur l'ensemble des attributs ainsi définis.
2) Donner une représentation de la base sous forme d'un ensemble de relations en 3FN.

Correction des exercices

I. 1) Par transitivité, A → C. Donc A → B, C. A est donc clé primaire.
R est en 2FN, mais pas en 3FN car B → C.

2) La DF B → C n'est pas respectée dans l'extension de R'
(B1 → C1, B1 → C2).

3)

R''	A	B	C
	A1	B1	C1
	A2	B1	C1
	A3	B2	C1
	A4	B3	C3

4) Couverture minimale : { A → B ; B → C }
G1 = { A → B }, G2 = { B → C }
R1 = < { <u>A</u>, B }, { A → B } >, R2 = < { <u>B</u>, C }, { B → C } >

II. 1) A, B → C
B → D
B, D, L → K
B, H, J, L → C
C → A
C → B
C, E, L → K
C, I, L → G
C, I, L → K
D → B
D → E
E, I, K → C
E, I, K → G
E, I, K → L

2) $G1 = \{ A, B \rightarrow C \}$
$G2 = \{ B \rightarrow D \}$
$G3 = \{ B, D, L \rightarrow K \}$
$G4 = \{ B, H, J, L \rightarrow C \}$
$G5 = \{ C \rightarrow A ; C \rightarrow B \}$
$G6 = \{ C, E, L \rightarrow K \}$
$G7 = \{ C, I, L \rightarrow G ; C, I, L \rightarrow K \}$
$G8 = \{ D \rightarrow B ; D \rightarrow E \}$
$G9 = \{ E, I, K \rightarrow C ; E, I, K \rightarrow G ; E, I, K \rightarrow L \}$

3) $G15 = \{ A, B \rightarrow C ; C \rightarrow A, B \}$
$G28 = \{ B \rightarrow D ; D \rightarrow B ; D \rightarrow E \}$
$G3 = \{ B, D, L \rightarrow K \}$
$G4 = \{ B, H, J, L \rightarrow C \}$
$G6 = \{ C, E, L \rightarrow K \}$
$G7 = \{ C, I, L \rightarrow G ; C, I, L \rightarrow K \}$
$G9 = \{ E, I, K \rightarrow C ; E, I, K \rightarrow G ; E, I, K \rightarrow L \}$

4) $R15 = < \{ \underline{A, B}, C \}, \{ A, B \rightarrow C ; C \rightarrow A, B \} >$
$R28 = < \{ B, \underline{D}, E \}, \{ B \rightarrow D ; D \rightarrow B ; D \rightarrow E \} >$
$R3 = < \{ \underline{B, D, L}, K \}, \{ B, D, L \rightarrow K \} >$
$R4 = < \{ \underline{B, H, J, L}, C \}, \{ B, H, J, L \rightarrow C \} >$
$R6 = < \{ \underline{C, E, L}, K \}, \{ C, E, L \rightarrow K \} >$
$R7 = < \{ \underline{C, I, L}, G, K \}, \{ C, I, L \rightarrow G ; C, I, L \rightarrow K \} >$
$R9 = < \{ \underline{E, I, K}, C, G, L \}, \{ E, I, K \rightarrow C ; E, I, K \rightarrow G ; E, I, K \rightarrow L \} >$

5) $R0 = < \{ \underline{A, B, C, D, E, H, I, J, K, L} \}, \emptyset >$

III. 1) NOSER $\rightarrow$ NOMS
NOSER $\rightarrow$ NORES
NORES $\rightarrow$ NOMR
NOSER $\rightarrow$ BUSER
NOPRO $\rightarrow$ NOSER
NOPRO $\rightarrow$ NOMP
NOPRO $\rightarrow$ BUPRO
NOEMP $\rightarrow$ NOPRO
NOEMP $\rightarrow$ NOME
NOEMP $\rightarrow$ NOTEL

NOEMP → NOBUR
NOBUR → NOSER
NOBUR → NOMB

2) G1 = { NOSER → NOMS ; NOSER → NORES ; NOSER → BUSER }
G2 = { NORES → NOMR }
G3 = { NOPRO → NOSER ; NOPRO → NOMP ; NOPRO → BUPRO }
G4 = { NOEMP → NOPRO ; NOEMP → NOME ; NOEMP → NOTEL ;
NOEMP → NOBUR }
G5 = { NOBUR → NOSER ; NOBUR → NOMB }

R1 = < { <u>NOSER</u>, NOMS, NORES, BUSER }, G1 >
R2 = < { <u>NORES</u>, NOMR }, G2 >
R3 = < { <u>NOPRO</u>, NOSER, NOMP, BUPRO }, G3 >
R4 = < { <u>NOEMP</u>, NOPRO, NOME, NOTEL, NOBUR }, G4 >
R5 = < { <u>NOBUR</u>, NOSER, NOMB}, G5 >

<u>Clé universelle</u> : NOEMP