

Sujet de TD n°3 BASES DE DONNÉES

Formes normales (2) et couverture minimale (2)

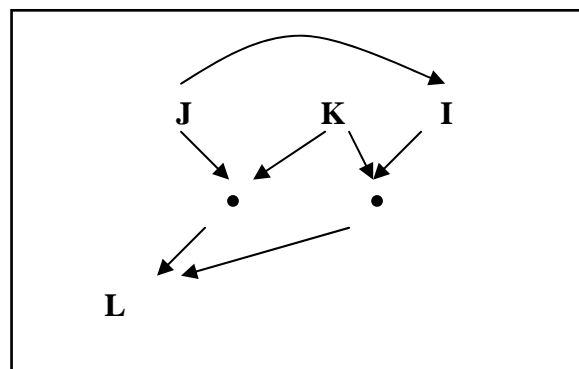
EXERCICE 1

Soit la relation $R(I, J, K, L)$ et les dépendances fonctionnelles :

- $JK \rightarrow L$
- $J \rightarrow I$
- $IK \rightarrow L$

QUESTIONS

1. Trouver une couverture minimale de F.
- Intuitivement en regardant le graphe il semble que la DF : $JK \rightarrow L$ peut être déduite....



Mais appliquons ce qui a été dit en cours :

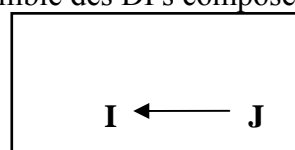
- Rendre canoniques & élémentaires les DFs qui ne le sont pas

$J \rightarrow I$ permet de dire que dans la DF $IK \rightarrow L$ l'attribut I peut être remplacé par J et donc cela donne : $JK \rightarrow L$

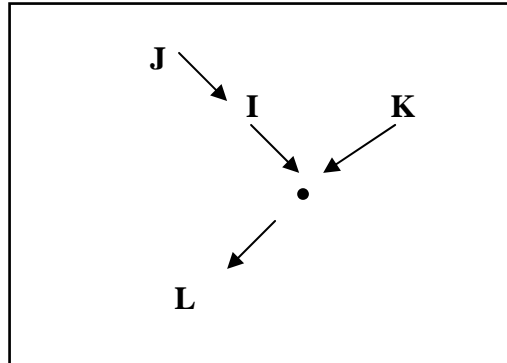
Donc si $JK \rightarrow L$ se déduit il reste les DFs suivantes :

- $IK \rightarrow L$
- $J \rightarrow I$

- Représenter les nouvelles Dfs sous forme d'un graphe dont les noeuds sont les attributs impliqués dans les Dfs et les arcs les Dfs elles-mêmes
 - Construction de l'ensemble des DFs composées d'un seul attribut source de DF



- Lister les DFs non encore intégrées (qui n'apparaissent pas dans le graphe représentant l'ensemble des DFs en 1) : $IK \rightarrow L$
- Placer les DFs avec comme source un sous-ensemble d'attributs déjà source de DF : aucunes
- Intégrer les DFs où l'un des attributs non affectés apparaît comme source :



2. En quelle forme normale est R ?

La clé candidate est J, K et donc :

- R est en 1NF
- R n'est pas en 2NF car $J \rightarrow I$ et J est un sous-ensemble de la clé candidate...

3. Proposer une décomposition de R en 3NF préservant les dépendances fonctionnelles, la décomposition comportant seulement deux relations.

On applique l'algorithme « passage au schéma relationnel du cours »

- R1 (J, K, L) où J, K est clé candidate
- R2 (J, I) où J est clé candidate

Il est évident que R2 est en 2NF, 3NF et même BCNF et que R1 est elle aussi en 2NF, 3NF et BCNF....

4. La décomposition proposée en 3 est-elle sans perte de dépendance et d'information? Si non, pouvez-vous modifier le schéma pour avoir une décomposition préservant les dépendances fonctionnelles et sans perte d'information? En quelle forme normale est chacune des relations de cette seconde décomposition?

Il n'y a pas de perte d'information (on a suivi le théorème de HEATH) mais il y a perte de dépendances : $IK \rightarrow L$

Conséquence : L'utilisateur peut saisir des couples (J,K) où J pour un même K correspond à des I différents

C'est là où on voit les limites de l'algorithme : toujours vérifier la perte de dépendance !!

On revient à la relation R et à sa clé candidate [J,K] et on essaye de ne garder que des DFs entre cette clé et I et L et d'enlever toutes les autres....

Après simplification des DFs on a vu que les seules DFs qui pouvaient a priori être enlevées sont $IK \rightarrow L$ et $J \rightarrow I$.

Note : $J \rightarrow I$ était la DF qui rend R *non 2NF* ; on a commencé ci-dessus par essayer de l'enlever (voir algorithme du cours appliqué ci-dessus) et cela n'a pas marché. On essaye maintenant de commencer par enlever la DF : $IK \rightarrow L$ pour éviter la perte de DFs. On obtient alors :

- R1 (I, K, L) est en BCNF (clé candidate [J, K])
- R2 (I, K, J) n'est pas en BCNF à cause de $J \rightarrow I$ (I est un sous-ensemble de la clé candidate et J n'appartient pas à la clé candidate)

D'où :

- R2A (J, I) et
- R2B (J, K) et
- elles sont toutes les deux en BCNF sans perte d'information.

Décomposition finale en trois relations :

- R1 (I, K, L)
- R2A (J, I)
- R2B (J, K)

EXERCICE 2

On souhaite développer une application pour gérer les habilitations d'accès aux applications informatiques d'une entreprise Y.

Dans cette application, les applications informatiques sont caractérisées par un code, un libellé et un type.

APPLICATION (Code-Application, Libellé-Application, Type-Application)
--

Les habilitations sont définies pour chaque application en fonction de compétences requises qui peuvent être obligatoires, recommandées ou simplement souhaitées. Il ne peut pas y avoir plus de 5 types d'habilitation différents pour une application donnée. En fait, une grille de compétences associées à chaque habilitation existe pour chaque application. En voici deux exemples :

<i>Application A</i>	C1	C2	C3
H1	O		
H2	R	R	
H3		O	S

<i>Application B</i>	C2	C7
H1	R	
H2	S	O

Dans cet exemple, on voit, par exemple, que l'application A peut notamment être utilisée soit avec l'habilitation 1 soit avec l'habilitation 2. Pour pouvoir demander l'habilitation 1 pour un agent, il faut que cette personne ait la compétence C1 ; pour l'habilitation H2, il est recommandé que l'agent ait les compétences C1 et C2.

A. DF : Code-Application, Code-Habilitation, Code-Compétence \rightarrow Type-requis
--

Type-Requis pouvant prendre les valeurs : *obligatoires (O)*, *recommandées (R)* ou *simplement souhaitées (S)*.

En plus des compétences requises, une habilitation est caractérisée par un code et un libellé.

HABILITATION (Code-Habilitation, Libellé-Habilitation)

Lorsqu'une habilitation est attribuée à un agent, on doit connaître la date à partir de laquelle l'habilitation est effective et éventuellement à quelle date elle doit être suspendue.

Implicitement on nous dit qu'il y a une relation décrivant un agent avec par exemple le contenu suivant :

AGENT (Matricule-Agent, Nom, Prénom)

B. DF : Matricule-Agent, Code-Habilitation, Code-Application → Date-Début-Habilitation, Date-Fin-Habilitation
--

Une compétence est caractérisée par un code et un libellé (indépendamment des applications et des agents).

COMPETENCE (Code-Compétence, Libellé-Compétence)

Dans le profil de chaque agent est indiquée la liste de ses compétences et pour chacune d'elle si c'est une compétence principale ou secondaire.

PROFIL (Matricule-Agent, Code-Compétence, Type-Compétence)

Type-Compétence pouvant prendre les valeurs : Principale ou Secondaire.

L'application doit permettre le suivi de l'état des demandes d'habilitation (accord de la direction des ressources humaines puis validation par le directeur puis saisie dans la base de données). Une demande concerne un seul agent et porte sur une habilitation donnée pour une application particulière.

DEMANDE (Matricule-Agent, Code-Habilitation, Code-Application, Etat-Demande)

QUESTIONS

1. Donner le schéma relationnel permettant de stocker les données nécessaires au fonctionnement de cette application, sachant qu'il ne peut exister plus d'une demande pour un agent, une habilitation et une application donnés (on ne gère pas le renouvellement d'habilitations).

Si on inclut maintenant les DFs dans les premières ébauches de relation :

- De la DF **A** on construit la relation :
CARACTERISATION (Code-Application, Code-Habilitation, Code-Compétence, Type-requis)
- De la DF **B** on complète la relation DEMANDE :
DEMANDE (Matricule-Agent, Code-Habilitation, Code-Application, Etat-Demande, Date-Début-Habilitation, Date-Fin-Habilitation)

On aurait pu opérer en prenant l'approche du cours : couverture minimale + algorithme pour passer du graphe au schéma relationnel mais dans un premier temps je n'ai pas voulu interpréter les DFs non explicitées entre par exemple Code-Habilitation et Libellé-Habilitation et j'ai juste proposé une relation pour encapsuler ces attributs....

On a donc le schéma relationnel suivant :

APPLICATION (Code-Application, Libellé-Application, Type-Application)
AGENT (Matricule-Agent, Nom, Prénom)
HABILITATION (Code-Habilitation, Libellé-Habilitation)
COMPETENCE (Code-Compétence, Libellé-Compétence)
PROFIL (Matricule-Agent, Code-Compétence, Type-Compétence)
CARACTERISATION (Code-Application, Code-Habilitation, Code-Compétence, Type-requis)
DEMANDE (Matricule-Agent, Code-Habilitation, Code-Application, Etat-Demande, Date-Début-Habilitation, Date-Fin-Habilitation)

APPLICATION : la clé candidate est Code-Application (relation en BCNF)

AGENT : la clé candidate est Matricule-Agent (relation en BCNF)

HABILITATION : la clé candidate est Code-Habilitation (relation en BCNF)

COMPETENCE : la clé candidate est Code-Compétence (relation en BCNF)

PROFIL : la clé candidate est [Matricule-Agent, Code-Compétence] (relation en BCNF)

CARACTERISATION : la clé candidate est [Code-Application, Code-Habilitation, Code-Compétence] (relation en BCNF)

DEMANDE : la clé candidate est [Matricule-Agent, Code-Habilitation, Code-Application] (relation en BCNF)

2. Que doit-on changer dans ce schéma pour prendre en compte le fait que l'on peut renouveler une habilitation pour un agent et une application donnés ? Donner une décomposition de R pour obtenir un schéma en 3NF sans perte de dépendances fonctionnelles et sans perte de données.

Pour pouvoir avoir plusieurs demandes concernant le même agent et la même habilitation il faut ajouter un numéro de demande ou quelque chose qui fasse office. Voici deux solutions :

- Mettre la date de la demande dans la clé
- Mettre un numéro d'ordre dans les demandes qui concernent la même habilitation et le même agent
- Ajouter un numéro de demande unique et en faire la clé candidate

Cela donne une des trois relations suivantes :

- *DEMANDE (Matricule-Agent, Code-Habilitation, Code-Application, Date-Demande, Etat-Demande, Date-Début-Habilitation, Date-Fin-Habilitation)*
- *DEMANDE (Matricule-Agent, Code-Habilitation, Code-Application, No-Ordre, Etat-Demande, Date-Début-Habilitation, Date-Fin-Habilitation)*
- *DEMANDE (Code-Demande, Matricule-Agent, Code-Habilitation, Code-Application, Etat-Demande, Date-Début-Habilitation, Date-Fin-Habilitation)*

EXERCICE 3

Soit la relation R avec les tuples suivants:

A	B	C
1	2	4
1	3	4
2	5	7
1	2	7
1	3	7
1	5	4
1	5	7

QUESTIONS

1. Quelles sont les dépendances fonctionnelles ou multivaluées compatibles avec la population de la relation R ?

- Il n'y a pas de DF entre A et C car des fois la valeur 1 dans A donne 4 ou 7 dans C
- Il n'y a pas de DF entre A et B car des fois la valeur 1 dans A donne 2 ou 3 ou 5 dans B
- Il n'y a pas de DF entre B et A car des fois la valeur 5 dans B donne 2 ou 1 dans A
- Il n'y a pas de DF entre B et C car des fois la valeur 2 dans B donne 4 ou 5 dans C
- Il n'y a pas de DF entre C et A car des fois la valeur 7 dans C donne 2 ou 1 dans A
- Il n'y a pas de DF entre C et B car des fois la valeur 4 dans C donne 2 ou 3 dans B

Sur ce que l'on peut lire du contenu de R il n'y a donc aucune DF.

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation et X et Y des sous ensembles de $\{A_1, A_2, \dots, A_n\}$.

On dit que X multi-détermine Y ($X \twoheadrightarrow Y$) **si**: étant données des valeurs de X :

Il y a un ensemble de valeurs de Y associées **et si** cet ensemble est indépendant des autres attributs $Z=R-X-Y$ de la relation R.

Si on regarde attentivement le contenu on constate que pour $A = 1$ on a que chaque valeur de B est associée à toutes les valeurs de C et réciproquement. Donc si considère les lignes pour lesquelles A vaut « 1 » on a deux DMs :

- $A \twoheadrightarrow B$ et
- $A \twoheadrightarrow C$

Pour $A = 2$ il y a qu'une seule ligne et donc la valeur pour B est indépendante de celle pour C. On ne peut donc dire qu'il y a aussi une dépendance multivaluée vers A et une autre vers C.

Au final on a deux DMs sur R :

- $A \twoheadrightarrow B$ et
- $A \twoheadrightarrow C$

Et R n'est donc pas en 4NF...

2. R est-elle en quatrième forme normale? Sinon, décomposer la relation R en quatrième forme normale.

On considère les dépendances multivaluées suivantes :

- $A \twoheadrightarrow B$ et
- $A \twoheadrightarrow C$

R ne serait pas en 4NF et on peut décomposer la relation en **R1 (A, B) et R2 (A,C)** qui sont toutes les deux en 4NF.

On note d'ailleurs qu'ici la généralisation du théorème de HEARTH

Toute relation R (X, Y, Z) est décomposable sans perte d'information en $\pi[X,Y]R$ et $\pi[X,Z]R$, s'il y a une dépendance multi-ensemble de X vers Y ($X \twoheadrightarrow Y$).

Il n'y a donc pas de perte de données (vu que l'on respecte les conditions de la décomposition et d'ailleurs si on vérifie par rapport aux données (ce n'est pas une démonstration mais une simple indication), le résultat de la jointure naturelle sur A entre R1 et R2 donne un résultat identique à R1

A	B
1	2
1	3
2	5
1	5

A	C
1	4
2	7
1	7

EXERCICE 4

Dans la base de données d'une société de transport, la relation suivante décrit pour les jours du mois en cours les affectations des conducteurs de bus aux lignes:

Affectation (No-Ligne, No-Conducteur, jour)

Plus précisément, cette relation signifie que tel jour, tel conducteur est affecté à telle ligne de bus. Considérons indépendamment les unes des autres les contraintes potentielles suivantes:

- Un conducteur ne peut pas travailler sur deux lignes différentes le même jour;
- Un conducteur fait les mêmes lignes tous les jours où il travaille;
- Un conducteur ne fait qu'une seule ligne;
- Une ligne est toujours parcourue par les mêmes conducteurs;
- Une ligne n'est parcourue que par un seul conducteur;
- Un conducteur ne fait pas la même ligne deux jours différents.

QUESTIONS

- Pour chacune de ces contraintes, définir la (ou les) dépendance(s) fonctionnelle(s) ou multivaluée(s) qui lui correspond(ent).

- No-Conducteur, jour \rightarrow No-Ligne
- No-Conducteur \twoheadrightarrow jour | No-Ligne

Exemple :

No-Conducteur	Jour	No-Ligne
1	121	41

1	121	42
1	121	43
2	122	41
2	122	42
2	122	43

- c) No-Conducteur \rightarrow No-Ligne
d) No-Ligne \rightarrow jour | No-Conducteur

Exemple :

No-Ligne	Jour	No-Conducteur
41	121	1
41	121	2
41	121	5
42	122	1
42	122	2
42	122	5

- e) No-Ligne \rightarrow No-Conducteur
f) No-Conducteur, No-Ligne \rightarrow jour

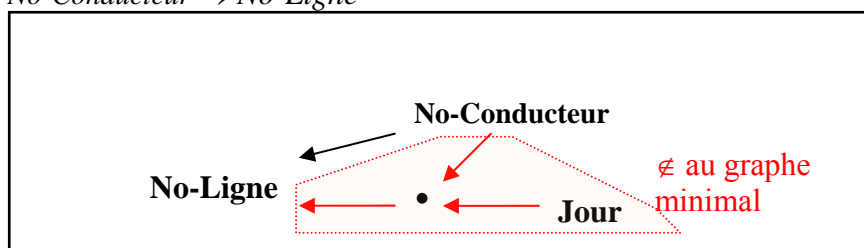
2. Pour chacun des ensembles de dépendances suivants, dessiner le graphe minimum des dépendances de la relation Affectation, préciser quel est (sont) l'identifiant de la relation, quelle est sa forme normale (en le justifiant en une ligne). Si la relation Affectation n'est pas bien normalisée proposer une décomposition en précisant les identifiants des relations obtenues, leur forme normale et s'il y a eu perte de dépendance. Dans ce dernier cas, définir en français la (les) contrainte d'intégrité que la base devra satisfaire.

3.

- L'ensemble des dépendances représentant les contraintes (a) et (c);

No-Conducteur, jour \rightarrow No-Ligne

No-Conducteur \rightarrow No-Ligne

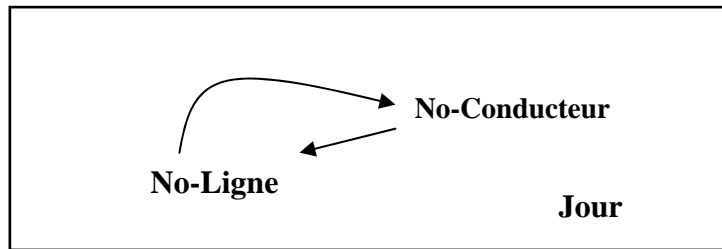


Clé candidate de R : [No-Conducteur, Jour]

Forme normale: 1NF (car No-ligne ne dépend que d'une partie de l'identifiant).

Décomposition (en BCNF et même 4NF)

- CJ (No-Conducteur, Jour)
 - CL (No-Conducteur, No-Ligne)
- L'ensemble des dépendances représentant les contraintes (c) et (e);
No-Conducteur \rightarrow No-Ligne
No-Ligne \rightarrow No-Conducteur



Clés candidates de R : [No-Conducteur, Jour] et [No-Ligne, Jour]

En 3NF (car tout attribut fait partie d'une clé candidate) mais pas BCNF (No-Ligne détermine une partie de [No-Conducteur, Jour] et No-Conducteur détermine une partie de [No-Ligne, Jour]).

Décomposition (en BCNF et même 4NF):

- CJ (No-Conducteur, Jour) avec un identifiant: [No-Conducteur, Jour]
- CL (No-Conducteur, No-Ligne) avec deux clés candidates : No-Conducteur et No-Ligne

Note Bien sur une seconde décomposition équivalente est possible en utilisant No-Ligne :

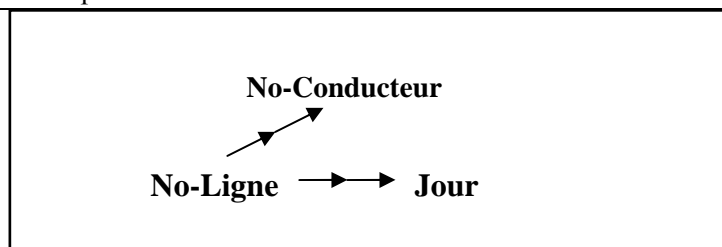
- CJ (No-Ligne, Jour) avec un identifiant: [No-Ligne, Jour]
- CL (No-Conducteur, No-Ligne) avec deux clés candidates : No-Conducteur et No-Ligne

- L'ensemble des dépendances représentant la contrainte (d);

$\text{No-Ligne} \twoheadrightarrow \text{jour} \mid \text{No-Conducteur}$

Propriété des dépendances multi-ensembles :

Si dans une relation R (X, Y, Z), on a la dépendance multi-ensemble $X \twoheadrightarrow Y$, alors on a aussi la dépendance multi-ensemble $X \twoheadrightarrow Z$.



Clé candidate : [No-Conducteur, No-Ligne, Jour]

En BCNF (car il n'y a pas de dépendance fonctionnelle), mais pas en 4NF (car il y a une dépendance multivaluée).

Décomposition (en 4NF):

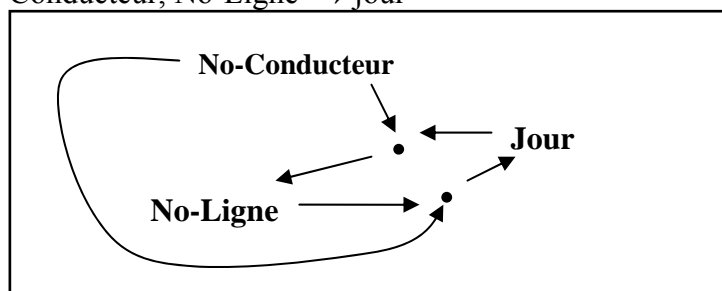
LJ (No-Ligne, Jour) avec une clé candidate : [No-Ligne, Jour]

CL (No-Conducteur, No-Ligne) avec une clé candidate : [No-Conducteur, No-Ligne]

- L'ensemble des dépendances représentant les contraintes (a) et (f).

$\text{No-Conducteur, jour} \rightarrow \text{No-Ligne}$

$\text{No-Conducteur, No-Ligne} \rightarrow \text{jour}$



Clés candidates : [No-Conducteur, Jour] et [N°conducteur, No_Ligne]
En BCNF (car les deux sources de dépendance fonctionnelle sont des identifiants entiers),
En 4NF (car il n'y a pas de dépendance multivaluée).
Pas de décomposition...