

## Sujet de TD n°2 BASES DE DONNÉES

### Correction

#### *Formes normales et couverture minimale*

### EXERCICE 1

Soit le schéma relationnel suivant :

**video**(No-Client, Nom-Client, Prénom-Client, Adresse-Client,  
DateDébut-AbonnementClient, No-Cassette, Titre-Film, PrixAchat-Cassette,  
Nom-Editeur, Adresse-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur,  
AnnéeSortie-Film, Genre, DateLocation-Cassette, Durée-Location)

#### Hypothèse :

- on considère que les titres de film sont uniques (pas de remake...).
- on souhaite garder un historique des locations de vidéo.

#### QUESTIONS

1. Expliquer pourquoi cette relation n'est pas en 2NF.

« Une relation est en 2NF ssi elle est en 1NF et ssi tout attribut qui n'appartient pas à une clé *candidate* ne dépend pas d'un sous-ensemble strict des attributs d'une clé candidate »

Il faut rechercher la clé candidate : vraisemblablement étant donné que l'on n'a pas explicité de DF particulière et que l'on a la contrainte de l'historique il semble que la clé [No-Cassette, DateLocation-Cassette] soit clé candidate, ce qui implique que la relation video n'est pas mono-attribut et donc qu'elle n'est pas forcément 2NF.

Selon toute logique on recense les dépendances fonctionnelles suivantes :

- No-Cassette → Titre-Film, PrixAchat-Cassette, Nom-Editeur, Adresse-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Genre
- No-Cassette, DateLocation-Cassette → Durée-Location, No-Client, Nom-Client, Prénom-Client, Adresse-Client, DateDébut-AbonnementClient

Plus finement on pourrait définir l'arbre des DFs :

- No-Cassette → Titre-Film, PrixAchat-Cassette
  - Titre-Film → Nom-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Genre
    - Nom-Editeur → Adresse-Editeur
- No-Cassette, DateLocation-Cassette → No-Client, Durée-Location
  - No-Client → Nom-Client, Prénom-Client, Adresse-Client, DateDébut-AbonnementClient

Dans tous les cas le fait que l'on ait la DF suivante :

- « No-Cassette → Titre-Film, PrixAchat-Cassette » ou bien à un degré plus grossier,
- « No-Cassette → Titre-Film, PrixAchat-Cassette, Nom-Editeur, Adresse-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Genre »

implique que la relation n'est pas en 2NF

2. Décomposer **Video** pour obtenir un schéma relationnel en 2NF tout en préservant les dépendances. Attention, on ne demande ici que la 2NF.

- R1 (No-Cassette, Titre-Film, PrixAchat-Cassette, Nom-Editeur, Adresse-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Genre)
- R2 (No-Cassette, DateLocation-Cassette, Durée-Location, No-Client, Nom-Client, Prénom-Client, Adresse-Client, DateDébut-AbonnementClient)

3. Les relations obtenues sont-elles en 3NF ? Expliquer votre réponse. Si ce n'est pas le cas, modifier le schéma afin d'obtenir un résultat en 3NF.

« Une relation est en 3NF ssi elle est en 2NF et ssi chaque attribut *qui n'appartient pas à une clé candidate* ne dépend pas fonctionnellement d'un attribut *qui n'appartient pas à une clé candidate* »

Si on considère R1, il y a une DF entre Titre-Film et d'autres attributs :

- Titre-Film → Nom-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Adresse-Editeur

On décompose R1 :

- R1' (No-Cassette, Titre-Film, PrixAchat-Cassette)
- R1'' (Titre-Film, Nom-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Adresse-Editeur, Genre)

Pour que R1'' soit en 3NF il faut encore la décomposer

- R1''' (Titre-Film, Nom-Editeur, Durée-Film, Nationalité-Film, Nom-Réalisateur, AnnéeSortie-Film, Genre)
- R1'''' (Nom-Editeur, Adresse-Editeur)

R1', R1''', R1'''' sont en 3NF

Si on considère maintenant R2, il y a une DF entre No-Client et d'autres attributs :

No-Client → Nom-Client, Prénom-Client, Adresse-Client, DateDébut-AbonnementClient

On décompose R2 :

- R2' (No-Cassette, DateLocation-Cassette, Durée-Location, No-Client)
- R2'' (No-Client, Nom-Client, Prénom-Client, Adresse-Client, DateDébut-AbonnementClient)

R2' et R2'' sont en 3NF

4. Les relations obtenues sont-elles en BCNF ? Expliquer votre réponse. Si ce n'est pas le cas, modifier le schéma afin d'obtenir un résultat en BCNF.

« Une relation est BCNF ssi elle est en 3NF et s'il n'y a pas d'autres DF élémentaires que celles issues des clés candidates ».

Il faut regarder s'il existe des DF entre un attribut qui ne fait pas partie de la clé candidate et avec un attribut qui appartient à la clé candidate

R2', R2'', R1', R1'', R1''' sont en BCNF

A vérifier : Perte de données ? et Perte de DF ? même approche que pour le TD1

## EXERCICE 2

Soit le schéma relationnel suivant :

**Agence ( No-Locataire, Nom-Locataire, No-Appartement, Adresse-Appartement, Nombre-Pièce, DateDébut-Location, DateFin-location, Montant-Loyer, ModePaiement-Loyer, No-Propriétaire, Nom-Propriétaire, FraisGestion-Appartement, Nb-tantième)**

dans lequel on a représenté une partie du système d'information d'une agence immobilière.

Hypothèses du système :

- Le loyer d'un appartement donné peut être renégocié entre deux locations, mais le mode de paiement du loyer (virement, chèque, ...) est imposé par le propriétaire indépendamment du locataire et indépendamment de l'appartement.

No-Appartement, DateDébut-Location → Montant-Loyer  
No-Propriétaire → ModePaiement-Loyer, Nom-Propriétaire  
*La DF de No-Propriétaire vers Nom-proprietaire est implicite basée sur la compréhension*

- Le loyer est versé par le locataire à l'agence qui reverse ensuite à chaque propriétaire la partie qui lui est due (déduction faite des frais de gestion qui sont payés par l'agence).

*Confirme la DF : No-Appartement, DateDébut-Location → Montant-Loyer*  
No-Appartement → FraisGestion-Appartement, Adresse-Appartement, Nombre-Pièce  
*La DF de No-appartement vers Adresse-Appartement, Nombre-Pièce est implicite basée sur la compréhension*  
*Cette hypothèse semble indiquer qu'il faut construire un attribut calculé : Montant-Loyer - FraisGestion-Appartement*  
*Le propriétaire est payé par l'agence et il peut donc apparaître plusieurs fois (autant de fois qu'il a d'appartement) :*  
No-propriétaire, n°appartement → ... (on aurait envie d'y mettre l'attribut calculé)

- Il n'est pas nécessaire de conserver le mode de paiement dans les historiques des locations.

*Cela signifie que ModePaiement-Loyer peut simplement être associé à No-propriétaire comme indiqué plus haut.*

- Les frais de gestion de l'appartement sont à la charge du propriétaire et fonction de l'appartement (taille, situation, ...).

*Confirme la remarque sur la construction d'un attribut calculé*

- Un locataire peut bien sûr louer plusieurs appartements, mais on ne prendra pas en compte la collocation dans cet exercice (il n'y a qu'un seul locataire pour un appartement donné à un instant donné).

*On ajoute un élément en partie droite de la DF vue plus haut :*

No-Appartement, DateDébut-Location → No-Locataire, Montant-Loyer

No-Locataire → Nom-Locataire

*La DF de No-Locataire vers Nom-Locataire est implicite basée sur la compréhension*

- Il peut, par contre, y avoir plusieurs propriétaires. Chacun détient alors un pourcentage du tantième correspondant à l'appartement. Le tantième de l'appartement indique les caractéristiques de l'appartement dans l'immeuble (situation, superficie, ...) de façon chiffrée.

No-propiétaire, n°appartement → Nb-tantième

- Il faut garder l'historique des locations enregistrées dans le système (et pas uniquement celles en cours) mais pas des propriétaires des appartements (on ne souhaite conserver que les propriétaires actuels). Enfin, on considère qu'il n'y a pas de changement de propriétaire pendant une location.

Confirme la DF No-Appartement, DateDébut-Location → Montant-Loyer

Elle devient :

No-Appartement, DateDébut-Location → Montant-Loyer, No-Locataire, DateFin-Location

*La DF de [No-Appartement, No-Locataire, DateDébut-Location] vers DateFin-Location est implicite basée sur la compréhension*

*La remarque ci-dessus fait que l'on n'ajoute pas la DateDébut-Location en partie gauche de la DF No-propiétaire, n°appartement → Nb-tantième*

Si on résume le graphe de couverture minimale :

No-Appartement, DateDébut-Location → No-Locataire, Montant-Loyer, DateFin-Location

No-Locataire → Nom-locataire

No-propiétaire, n°appartement → Nb-tantième

No-propiétaire → ModePaiement-Loyer, Nom-propiétaire

No-Appartement → FraisGestion-Appartement, Adresse-Appartement, Nombre-Pièce

## **QUESTIONS**

1. Indiquer en quelle forme normale est la relation ci-dessus. Justifier votre réponse.

Il faut déterminer quelle est la clé candidate :

D'après les DFs mentionnées ci-dessus il semble que la seule clé candidate soit :

No-Appartement, No-propiétaire, DateDébut-Location

Intuitivement c'est l'union des parties gauche du premier niveau du graphe

En pratique on montre qu'il y a une DF vers tous les autres attributs et que si on en enlève un on ne peut identifier tous les tuples de la relation.

La relation Agence est :

- 1NF (pas d'attributs multivalués)
- N'est pas 2NF car elle ne vérifie : « Une relation est en 2NF ssi elle est en 1NF et ssi tout attribut qui n'appartient pas à une clé *candidate* ne dépend pas d'un sous-ensemble strict des attributs d'une clé candidate »

Les 4 DF suivantes empêchent la relation Agence d'être en 2NF

No-propriétaire, n°appartement → Nb-tantième
No-Appartement, DateDébut-Location → No-Locataire, Montant-Loyer, DateFin-Location, Nom-Locataire
No-propriétaire → ModePaiement-Loyer, Nom-propriétaire
No-Appartement → FraisGestion-Appartement, Adresse-Appartement, Nombre-Pièce

Pour transformer cette relation en 2NF il faut la décomposer en appliquant la règle énoncée dans le cours :

Décomposition de $R(X, Y, Z)$ tel que $X \rightarrow Y$
---

Si la DF $X \rightarrow Y$ est élémentaire :
--

- |   |
|---|
| <ul style="list-style-type: none"><li>• En créant une relation <math>R'</math> de clé <math>X</math> et d'attributs <math>Y</math></li><li>• En supprimant dans <math>R</math> les attributs <math>Y</math></li></ul> |
|---|

Si la DF $X \rightarrow Y$ est non élémentaire ( <i>pas minimale</i> ) :
--

$X'$ est la partie de $X$ en DF élémentaire avec $Y$
--

- |  |
|--|
| <ul style="list-style-type: none"><li>• En créant une relation <math>R'</math> de clé <math>X'</math> et d'attributs <math>Y</math></li><li>• En supprimant dans <math>R</math> les attributs <math>Y</math></li></ul> |
|--|

$R'$ (No-Appartement, No-propriétaire, DateDébut-Location) *reste de Agence*

R1 (No-Appartement, DateDébut-Location, No-Locataire, Montant-Loyer, DateFin-Location, Nom-Locataire)<sup>1</sup>

R2(No-propriétaire, ModePaiement-Loyer, Nom-propriétaire)

R3(No-Appartement, FraisGestion-Appartement, Adresse-Appartement, Nombre-Pièce)

R4 (No-propriétaire, n°appartement, Nb-tantième)

2. Donner les transformations successives à effectuer sur la relation pour qu'elle soit en BCNF. Indiquer clairement les transformations nécessaires pour chaque forme normale (si la relation est par exemple en 2NF, il faut montrer comment transformer la relation pour qu'elle soit en 3NF et donner la/les relation(s) obtenue(s). Puis il faut donner les transformations nécessaires et la/les relation(s) obtenue(s) pour qu'elle soit en BCNF).

On montre facilement que  $R'$ , R1, R2, R3, R4 sont en 2NF (plusieurs ont une clé candidate mono-attribut).

---

<sup>1</sup> J'ai une différence avec la correction papier d'Isabelle qui met aussi « quantième » dans R1... mais je ne vois pas la raison ... ou alors je lis mal ☺

« Une relation est en 3NF ssi elle est en 2NF et ssi chaque attribut *qui n'appartient pas à une clé candidate* ne dépend pas fonctionnellement d'un attribut *qui n'appartient pas à une clé candidate* »

- R' est 3NF et en BCNF
- R1 : clé candidate : No-Appartement. R1 n'est pas en 3NF car elle contient la DF No-Locataire → Nom-locataire  
R1A(No-Appartement, DateDébut-Location, No-Locataire, Montant-Loyer, DateFin-Location)  
R1B(No-Locataire, Nom-Locataire)
- R2 (clé candidate No-Propriétaire) est 3NF et BCNF
- R3 (clé candidate No-Appartement) est 3NF et BCNF
- R4 (clé candidate No-Propriétaire, No-Appartement) est 3NF et BCNF

Après décomposition en R1A et R1B, ces relations sont en 3NF et BCNF.

3. Démontrer qu'il n'y a pas de perte de données sur le schéma relationnel normalisé.

On montre que les DFs mises en évidence dans :

No-Appartement, DateDébut-Location → No-Locataire, Montant-Loyer, DateFin-Location No-Locataire → Nom-locataire No-propriétaire, n°appartement → Nb-tantième No-propriétaire → ModePaiement-Loyer, Nom-propriétaire No-Appartement → FraisGestion-Appartement, Adresse-Appartement, Nombre-Pièce
--

Sont toujours présentes (DF entre la clé candidate et chacun des attributs)

En ce qui concerne la perte de données, on peut avoir une approche expérimentale en faisant la jointure sur un ensemble de tuples mais il faut vérifier pour chaque décomposition de Agence vers R', R1A, R1B, R2, R3 et R4 le théorème de HEATH : « Toute relation R(X, Y, Z) est décomposable sans perte d'information en

- R1 = projection (X, Y) de R et
- R2 = projection (X, Z) de R

S'il y a dans R une dépendance fonctionnelle de X vers Y ( $X \rightarrow Y$ ) »

On montre facilement que c'est le cas.

## EXERCICE 3

Soit une relation :

<b>R(A, B, C, D, E)</b>
-------------------------

et F, l'ensemble des dépendances fonctionnelles associées à R :

A → B, B → D, C → A, AD → C, CB → E,
--

**AEC → D**

**QUESTIONS**

1. Donner une décomposition de R pour obtenir un schéma en 3NF sans perte de dépendances fonctionnelles et sans perte de données.

On cherche à simplifier les trois dépendances ayant une source multiple dans le but de ne garder si possible que des DF ayant une seule source :

**AD → C**

Par le théorème de transitivité  $A \rightarrow B$  et  $B \rightarrow D$  on obtient :  $A \rightarrow D$

Donc :  $AA \rightarrow C$  (on a remplacé D par A puisque « A détermine D »)

Donc :  $A \rightarrow C$  (qui remplace  $AD \rightarrow C$ )

**CB → E**

Par le théorème de transitivité  $C \rightarrow A$  et  $A \rightarrow B$  on obtient :  $C \rightarrow B$

Donc :  $CC \rightarrow E$  (on a remplacé B par C puisque « C détermine B »)

Donc :  $C \rightarrow E$  (qui remplace  $CB \rightarrow E$ )

**AEC → D**

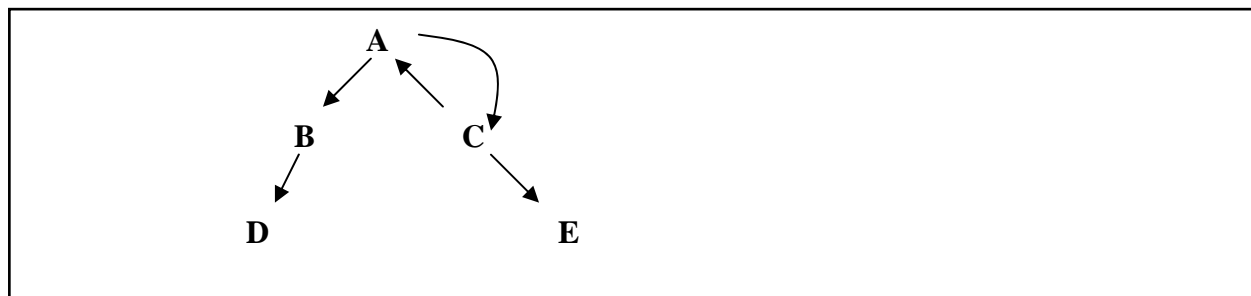
On a par hypothèse :  $C \rightarrow A$  et  $C \rightarrow E$

Donc :  $CCC \rightarrow D$  (on a remplacé A par C puisque « C détermine A »  
et on a remplacé E par C puisque « C détermine E »)

On obtient donc en simplifiant les 'C' :  $C \rightarrow D$  qui est déjà une DF déduite par transitivité des trois DF suivantes :  $C \rightarrow A \rightarrow B \rightarrow D$

L'utilisation des différents théorèmes permet de supprimer les DFs avec plusieurs sources que l'on peut déduire à partir de DFs plus simples qui peuvent exister.

Le graphe minimal est donc :



Commentaires sur le graphe et les DFs déduites :

**$A \rightarrow B$** , (dans l'énoncé)

**$B \rightarrow D$** , (dans l'énoncé)

**$C \rightarrow A$** , (dans l'énoncé)

**$A \rightarrow C$** , (remplace  $AD \rightarrow C$  pouvant être déduit de A sans l'ajout de D car A détermine D par transitivité)

**$C \rightarrow E$** , (remplace  $CB \rightarrow E$  qui peut être déduit de C sans l'ajout de B car C détermine B par transitivité)

Note :  **$AEC \rightarrow D$**  n'est pas utile car :

C détermine D par transitivité : C détermine A qui détermine B qui détermine D (DFs existantes)

C détermine A par transitivité : C détermine E et C détermine A (DFs existantes)

Comme on le voit sur le graphe, A et C sont deux clés candidates puisqu'à partir de l'une ou de l'autre on peut atteindre tous les autres attributs. R (A, B, C, D, E) avec A ou C qui peuvent devenir l'une ou l'autre la clé primaire.

Pour être en 3NF il faut supprimer la dépendance suivante :  $B \rightarrow D$ . On décompose en utilisant de manière maintenant classique le théorème de Heath :

- R1 (A, B, C, E) ou R1 (A, B, C, E)
- R2 (B, D)

La normalisation se fait par rapport aux clés candidates et non par rapport aux clés primaires. Etant donné que A et C sont deux clés candidates il serait nuisible de vouloir enlever par exemple la DF  $C \rightarrow E$  dans le cas où on choisit A comme clé primaire ou par exemple  $A \rightarrow B$  dans le cas où c'est C qui est la clé primaire car cela reviendrait à (inutilement) trop décomposer et à nécessiter des jointures supplémentaires qui nuiraient aux performances.

Ces relations sont en 3NF. On remarque que toutes les dépendances (non déduites) mentionnées dans le graphe minimal sont présentes dans R1 et R2. On a appliqué le théorème de Heath pour faire la décomposition et donc il n'y a pas de perte de données.