

# LICENCE 3 INFORMATIQUE

FINANCE

HISTOIRE

GÉOGRAPHIE

INFORMATIQUE

MATHÉMATIQUES

SCIENCES POUR L'INGÉNIEUR

FRANÇAIS LANGUE ÉTRANGÈRE

ADMINISTRATION ÉCONOMIQUE ET SOCIALE

DIPLÔME D'ACCÈS AUX ÉTUDES UNIVERSITAIRES

LICENCE MENTION INFORMATIQUE



Centre de Télé-enseignement  
Universitaire

<http://ctu.univ-fcomte.fr>

## FILIÈRE INFORMATIQUE

● **VVISEBD**

Bases de données

**Mme DAMY - SYLVIE**  
*sylvie.damy@univ-fcomte.fr*



**UNIVERSITÉ DE  
FRANCHE-COMTÉ**





# Préface

Les exercices présentés dans le cours sont des exercices d'application directe des notions présentées. Pour faire des exercices plus complets il faut regarder les TDs.

**Attention**, ces exercices *jouet* pour la plupart sont à faire au fur et à mesure que vous avancez dans le cours. On travaille la notion que l'on vient de découvrir. Ces mêmes exercices vus en fin de travail sur le chapitre peuvent paraître inadaptés, voire incorrects.

Le cours de base de données est composé de quatre parties subdivisées en sept chapitres, correspondant à l'introduction de différentes notions très générales comme dans la partie 1 (chapitre 1), ou beaucoup plus précises telles que dans la partie 3 (chapitre 5) qui présente le langage SQL (DML).

Seuls les chapitres 2, 5, et 7 donne lieu à la présentation d'exercices. Les corrigés de ces exercices sont disponibles dans ce document respectivement aux chapitres 1 (modélisation relationnelle), 2 (SQL) et 3 (normalisation).

**ATTENTION CE DOCUMENT NE REMPLACE PAS CELUI DE TRAVAUX DIRIGES.**

Il ne présente que les corrigés des exercices présentés dans le cours.



# Chapter 1

## L'exemple du cinéma

Exemple fil rouge du cours

### 1.1 Présentation

La base de données **CINEMAS** doit permettre de gérer l'ensemble des cinémas français, ainsi que les films qui y sont à l'affiche.

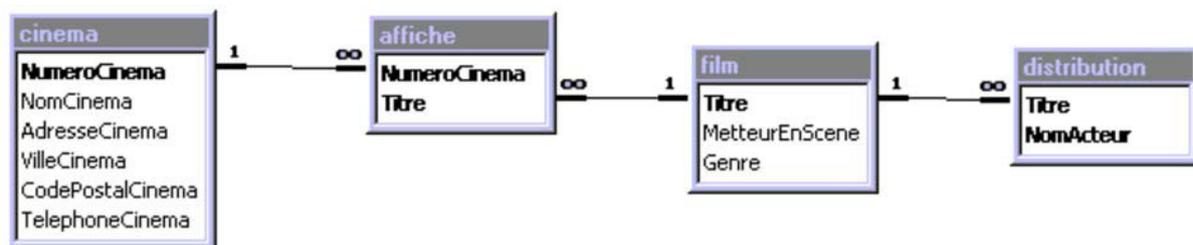
Intuitivement nous allons décrire les cinémas, les films, ainsi que les affiches de chaque cinéma, et la distribution de chaque film.

Il est bien évidemment possible de réaliser une base de données comprenant plus d'informations. Mais nous nous limiterons à une base de données nous permettant de connaître les films à l'affiche dans tel cinéma, les cinémas qui ont à l'affiche tel cinéma, les films de tel genre, dans telle ville, les films avec tel acteur,...

### 1.2 Le modèle

La base de données CINEMAS est composée de 4 relations :

- CINEMA
- AFFICHE
- FILM
- DISTRIBUTION



### 1.3 La relation CINEMA

Numero Cinema	Nom Cinema	Adresse Cinema	Ville Cinema	CodePostal Cinema	Telephone Cinema
1	Vox	Grande Rue	Besançon	25000	03 81 82 05 69
2	Plazza	Rue Des Granges	Besançon	25000	03 81 26 35 98
3	Plazza	Rue Ronchoux	Dijon	21000	03 80 26 53 25
4	Royal	Rue Villarceau	Dijon	21000	03 80 23 52 58

### 1.4 La relation FILM

Titre	MetteurEnScene	Genre
Alerte	Peterson	Aventure
Bad Boys	Bay	Policier
Braveheart	Gibson	Aventure
Gazon Maudit	Balasko	Comedie
Junior	Reitman	Comedie
L'Effaceur	Russel	Action
Les Voleurs	Techine	Comedie Dramatique
Ma Saison Preferee	Techine	Comedie Dramatique

### 1.5 La relation DISTRIBUTION

Titre	NomActeur
Alerte	Freeman
Alerte	Hoffman
Alerte	Russo
Bad Boys	Lawrence
Bad Boys	Loni
Bad Boys	Smith
Braveheart	Gibson
Braveheart	Marceau
Gazon Maudit	Abril
Gazon Maudit	Balasko
Gazon Maudit	Chabat
Junior	De Mornay
Junior	De Niro
Junior	Schwarzenegger
L'Effaceur	Schwarzenegger

### 1.6 La relation AFFICHE

NumeroCinema	Titre
1	Alerte
1	Junior
2	Alerte
2	Gazon Maudit
2	L'Effaceur
3	Alerte
3	Junior
4	Alerte
4	Braveheart
4	Les Voleurs



## Chapter 2

# Le modèle relationnel

Corrigés des exercices du chapitre 2

### 2.1 Exercice 2.1 La notion de relation

Les attributs de la relation **FILM** sont : **Titre**, **MetteurEnScene**, et **Genre**. Ces trois attributs sont des chaînes de caractères.

Cardinalité et degré de la relation : Cardinalité = 8 et Degré = 3

Le domaine de l'attribut **Genre** : Le domaine de l'attribut genre est l'ensemble des genres de films existants. Cet ensemble contient entre autres {comédie, Aventure, Policier, Action, Comédie Dramatique, Fantastique, ...}. Il s'agit de chaînes de caractères.

### 2.2 Exercice 2.2 Les propriétés des relations

La relation proposée n'est pas correcte car certains attributs ne sont pas atomiques. L'intersection de la première ligne et la deuxième colonne par exemple, contient deux valeurs : *Alerte* et *Junior*. Cette relation n'est pas en première forme normale.

Une version normalisée de cette relation est la relation **AFFICHE** proposée dans le tableau 2.1 (et que nous retrouverons dans la base **CINEMA**).

NumeroCinema	Titre
1	Alerte
1	Junior
2	Alerte
2	Gazon Maudit
2	L'Effaceur
3	Alerte
3	Junior
4	Alerte
4	Braveheart
4	Les Voleurs

Table 2.1 : Relation **AUTREAFFICHE** normalisée

## 2.3 Exercice 2.3 : Les super-clés de CINEMA

Tout sous-ensemble d'attributs de la relation CINEMA contenant l'attribut NumeroCinema est une super-clé. On a ainsi la liste suivante de super-clés :

- (NumeroCinema),
- (NumeroCinema, NomCinema),
- (NumeroCinema, NomCinema, AdresseCinema),
- (NumeroCinema, NomCinema, AdresseCinema, VilleCinema),
- (NumeroCinema, NomCinema, AdresseCinema, VilleCinema, CodePostalCinema),
- (NumeroCinema, NomCinema, AdresseCinema, VilleCinema, CodePostalCinema, TelephoneCinema),
- (NumeroCinema, NomCinema, VilleCinema),
- (NumeroCinema, NomCinema, VilleCinema, CodePostalCinema),
- (NumeroCinema, NomCinema, VilleCinema, CodePostalCinema, TelephoneCinema),
- ...

Remarquons au passage que pour toute relation l'entête complète de la relation est une super-clé.

## 2.4 Exercice 2.4 : Les clés candidates

1 - Titre est unique

Si le titre est unique, il existe une seule clé candidate pour la relation FILM qui est {Titre}

2 - Titre n'est pas unique

Dans ce cas l'unique clé candidate est : {Titre, MetteurEnScene}

3 - Titre est unique

{Titre, MetteurEnScene} n'est pas une clé candidate car ce sous-ensemble d'attributs ne respecte pas la propriété d'irréductibilité. En effet {Titre} possède la propriété d'unicité.

## 2.5 Exercice 2.5 : Définition de liens entre relations

Le lien définit entre les relations GENRE et FILM est le suivant :

GENRE(\*NumeroGenre, Genre) [NumeroGenre](1..n) --> [NumeroGenre] FILM(\*Titre, MetteurEnScene, NumeroGenre)

## 2.6 Exercice 2.6 : Définition de liens entre relations

1 - Lien CINEMA - DIRECTEUR

Il s'agit d'un lien 1..1, puisque pour un cinéma on a un directeur, et qu'une personne correspond à un directeur. Le lien définit est le suivant :

CINEMA(\*NumeroCinema, NomCinema, ...) [NumeroCinema] (1 .. 1) --> [NumeroCinema] DIRECTEUR(\*NumeroDirecteur, Nom, Prenom, TelephoneDirecteur, NumeroCinema)

Le lien est défini de la relation CINEMA (la clé primaire) vers la relation DIRECTEUR (la clé étrangère).

## 2 - Sans table DIRECTEUR

Il est bien évidemment possible de modifier simplement la relation CINEMA de la façon suivante :  
 CINEMA(\*NumeroCinema, NomCinema, AdresseCinema, VilleCinema, CodePostalCinema, telephoneCinema, NumeroDirecteur, Nom, Prenom, TelephoneDirecteur)

## 3 - Comparaison des solutions

Les deux solutions peuvent paraître équivalentes, cependant au niveau de la mise à jour des données dans la base il peut y avoir des différences. Supposons en effet que les directeurs de cinémas puissent être mutés d'un cinéma vers un autre chaque année.

Dans la première solution il suffira de mettre à jour dans la relation DIRECTEUR les numéros de cinéma.

Dans la deuxième solution, il faudra ressaisir toutes les informations pour tous les cinémas qui changent de directeur.

## 2.7 Exercice 2.7 : La règle de l'unicité de clé

### 1 - Alerte existe déjà

Le titre Alerte est déjà présent dans la relation FILM. Etant donné que cet attribut est la clé de notre relation il est impossible d'y ajouter un t-uplet dont cet attribut aurait la même valeur.

### 2 - Zorro n'existe pas

Ici la valeur de l'attribut Titre : "Zorro" n'existe pas encore dans la relation FILM. On peut donc ajouter ce t-uplet.

## 2.8 Exercice 2.8 : Les contraintes de domaine

Une contrainte de domaine possible est : Le fait que l'année de sortie doit être inférieure ou égale à l'année en cours, ou que cette année soit supérieure à 1900 par exemple.

$1900 \leq \text{AnnéeSortie} \leq \text{Année en cours}$

## 2.9 Exercice 2.9 : Gestion de NULL

### 1 - L'attribut Nom

L'attribut Nom ne fait pas partie de la clé primaire de la relation DIRECTEUR, il peut donc être autorisé à prendre la valeur NULL.

### 2 - L'attribut NumeroDirecteur

L'attribut NumeroDirecteur fait partie de la clé primaire de la relation DIRECTEUR, il ne peut donc pas être autorisé à prendre la valeur NULL, d'après la règle d'intégrité des entités.

### 3 - L'attribut NumeroCinema

L'attribut NumeroCinema ne fait pas partie de la clé primaire de la relation DIRECTEUR, il peut donc être autorisé à prendre la valeur NULL. Le fait d'accepter que cet attribut prenne la valeur NULL signifie que l'on peut décrire un directeur sans connaître le cinéma qu'il dirige. Ce qui correspond à l'extension que nous avons proposée pour la règle d'intégrité référentielle.

## 2.10 Exercice 2.10 : Clé primaire de CINEMA

### 1 - les relations

L'attribut `NumeroCinema` se trouve dans la relation `AFFICHE` du modèle proposé dans le cours. Nous allons donc modifier cette relation de la façon suivante :

`AFFICHE(*NomCinema, *VilleCinema, *Titre)`

et le lien entre les relations `CINEMA` et `AFFICHE` devient :

`CINEMA(*NomCinema, AdresseCinema,*VilleCinema,..)[NomCinema, VilleCinema] (1 .. n) -> [NomCinema, VilleCinema] AFFICHE(*NomCinema, *VilleCinema, *Titre)`

C'est la première fois que nous rencontrons une clé étrangère composée de plusieurs attributs. Comme les clés candidates, elles peuvent contenir un nombre quelconque d'attributs. La représentation graphique d'un tel lien est réalisée de la façon suivante :

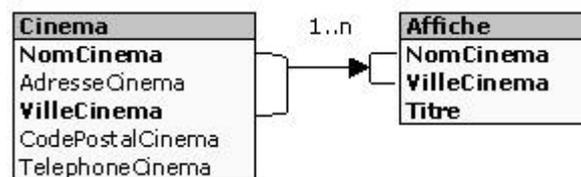


Figure 2.1 : Lien entre CINEMA et AFFICHE

Les autres relations restent inchangées.

### 2 - Comparaison des deux solutions

Le fait de prendre comme clé primaire un attribut de type "simple" comme `NumeroCinema`, nous permet ensuite de définir les relations où cet attribut apparaît comme clé étrangère de façon simple aussi. Dans la solution que nous venons de proposer, il faudra introduire l'ensemble d'attributs `NomCinema`, `VilleCinema`, dans toutes les relations concernées. Le premier effet est une perte de place importante. Cet aspect n'est pas négligeable lorsque l'on définit une base de données. La deuxième conséquence est la lourdeur des expressions qu'il faudra écrire lors de l'écriture de requêtes de manipulations des données.

## 2.11 Exercice 2.11 : Clé candidate de FILM

### 1 - MetteurEnScene : Clé primaire de FILM

Le fait que `MetteurEnScene` soit une clé candidate de la relation `FILM` signifie que deux films ne peuvent pas avoir le même metteur en scène. Un metteur en scène ne peut réaliser qu'un seul film !

### 2 - MetteurEnScene, Genre : Clé primaire de FILM

Le fait que `MetteurEnScene`, `Genre` soit une clé candidate de la relation `FILM` signifie que deux films ne peuvent pas avoir le même metteur en scène et le même genre. Un metteur en scène ne peut réaliser qu'un seul film pour un genre donné !

## 2.12 Exercice 2.12 : L'intégrité référentielle

### Ajout de données

1 - Dans `AFFICHE` : L'ajout dans la relation `AFFICHE` du t-uplet : (1, Bad Boys) est réalisé. La

relation AFFICHE a alors 11 t-uplets.

2 - Dans DISTRIBUTION : L'ajout dans la relation DISTRIBUTION du t-uplet: ("Alien", "Weaver"), ne peut pas être réalisé car la règle d'intégrité référentielle n'est pas respectée : Il n'y a aucun t-uplet dans la relation FILM ayant comme valeur de l'attribut Titre : "Alien".

3 - Dans AFFICHE : L'ajout dans la relation AFFICHE du t-uplet : (8, "Bad Boys"), ne peut pas être réalisé car la règle d'intégrité référentielle n'est pas respectée : Il n'y a aucun t-uplet dans la relation CINEMA ayant comme numéro de cinéma 8. En d'autres termes on ne peut pas mettre un film à l'affiche d'un cinéma qui n'existe pas.

### Mise à jour de données

4 - Le cinéma N°2 : Le remplacement du numéro du Cinéma 2 par 1 n'est pas possible, car cela nous conduirait à une situation où deux t-uplets ont la même valeur de clé primaire ce qui est impossible.

5 - Changement de numéro de cinéma : Si la règle de mise à jour est CASCADE, on peut remplacer le numéro du cinéma 2 par 8. Cette mise à jour sera alors réalisée dans la relation CINEMA ainsi que dans la relation AFFICHE. Ce qui donne les relations suivantes :

NumeroCinema	Titre
1	Alerte
1	Junior
8	Alerte
8	Gazon Maudit
8	L'Effaceur
3	Alerte
4	Junior
4	Alerte
4	Braveheart
4	Les Voleurs

Table 2.2 : Relation AFFICHE

et la relation CINEMA :

NumeroCinema	NomCinema ...
1	Vox
8	Plazza
3	Plazza
4	Royal

Table 2.3 : Relation CINEMA

### Suppression de données

6 - Suppression du cinéma N°2 : Si la règle d'effacement est NO ACTION, on ne peut pas supprimer le cinéma 2, puisqu'il est référencé dans la relation AFFICHE.

7 - Suppression du cinéma N°2 : Si la règle d'effacement est CASCADE, on peut supprimer le cinéma 2. Un t-uplet sera supprimé de la relation CINEMA, et trois t-uplets seront supprimés de la relation

AFFICHE.

On obtient la relation CINEMA suivante :

NumeroCinema	NomCinema ...
1	Vox
3	Plazza
4	Royal

Table 2.4 : Relation CINEMA

et la relation AFFICHE :

NumeroCinema	Titre
1	Alerte
1	Junior
3	Alerte
4	Junior
4	Alerte
4	Braveheart
4	Les Voleurs

Table 2.5 : Relation AFFICHE

8 - Suppression du cinema N°2 : Si la règle de suppression est SET DEFAULT, on peut supprimer le cinéma 2. Un t-uplet sera supprimé de la relation CINEMA, et trois t-uplets seront mis à jour dans la relation AFFICHE.

On obtient la relation CINEMA suivante :

NumeroCinema	NomCinema ...
1	Vox
3	Plazza
4	Royal

Table 2.6 : Relation CINEMA

et en supposant que la valeur par défaut est 0, la relation AFFICHE :

NumeroCinema	Titre
1	Alerte
1	Junior
0	Alerte
0	Gazon Maudit
0	L'Effaceur
3	Alerte
4	Junior
4	Alerte
4	Braveheart
4	Les Voleurs

Table 2.7 : Relation AFFICHE

Mais ici nous avons un **problème**, cette politique de mise à jour de type DEFAULT ne peut pas être mise en oeuvre. En effet la clé étrangère NumeroCinema fait partie de la clé de la table AFFICHE. On risque ainsi d'avoir, après plusieurs suppressions dans CINEMA, plusieurs t-uplets de la table AFFICHE avec la même clé primaire.

9 - Suppression du cinema N°2 : Si la règle de suppression est NO CHECK, on peut supprimer le cinéma 2. Un t-uplet sera supprimé de la relation CINEMA.

On obtient la relation CINEMA suivante :

NumeroCinema	NomCinema ...
1	Vox
3	Plazza
4	Royal

Table 2.8 : Relation CINEMA

et la relation AFFICHE reste inchangée. D'où l'introduction d'incohérences dans la base de données.

NumeroCinema	Titre
1	Alerte
1	Junior
2	Alerte
2	Gazon Maudit
2	L'Effaceur
3	Alerte
4	Junior
4	Alerte
4	Braveheart
4	Les Voleurs

Table 2.9 : Relation AFFICHE

## 2.13 Exercice 2.13 : Les opérateurs ensemblistes

### 1. Union, Intersection et Différence

### 2.13.1 L'union

L'union de FILMA et FILMF donne une relation qui représente l'ensemble des films Français ou Américains.

$FILMAOUF = FILMA \cup FILMF$

La relation résultat est :

FILMAOUF
Alerte
Congo
BadBoys
Candyman2
Casper
Harcelement
Gazon maudit
Ma saison préférée
Les voleurs

Table 2.10 : FILMAOUF

### 2.13.2 L'intersection

L'intersection de FILMA et FILMF donne l'ensemble des films co-produits par la France et les USA.

$FILMAETF = FILMA \cap FILMF$

La relation résultat est :

FILMAETF
Casper
Harcelement

Table 2.11 : FILMAETF

### 2.13.3 La différence

La différence de FILMA et FILMF donne l'ensemble des films produits par les USA uniquement.

$FILMANONF = FILMA - FILMF$

La relation résultat est :

FILMANONF
Alerte
Congo
BadBoys
Candyman2

Table 2.12 : FILMANONF

2 - La symétrie des opérateurs ensemblistes

Il est immédiat que l'union et l'intersection sont des opérateurs symétriques. Par contre la différence ne l'est pas. La relation

$FILMFNONA = FILMF - FILMA$

a pour valeur :

Ce qui correspond à l'ensemble des films produits uniquement par la France.

FILMANONF
Gazon maudit
Ma saison préférée
Les voleurs

Table 2.13 : FILMANONF

## 2.14 Exercice 2.14 : La sélection

1- Les acteurs jouant dans Alerte 2.14

ALERTE = S(Titre = "Alerte")DISTRIBUTION

Titre	NomActeur
Alerte	Freeman
Alerte	Hoffman
Alerte	Russo

Table 2.14 : Les acteurs de Alerte

2- Les titres des films à l'affiche du cinéma N°1 2.15

FILMS1 = S(NumeroCinema=1)AFFICHE

NumeroCinema	Titre
1	Alerte
1	Junior

Table 2.15 : Affiche du cinema 1

3- Les films mis en scène par Téchiné ??

FILMSTECHINE = S(MetteurEnScene="Techine")FILM

Titre	MetteurEnScene	Genre
Les Voleurs	Techine	Comedie Dramatique
Ma Saison Preferee	Techine	Comedie Dramatique

Table 2.16 : Films de Téchiné

4- Les films d'aventure ou mis en scène par Téchiné

FILMSAVENTUREOUTECHINE = S(MetteurEnScene="Techine" OU Genre="Aventure")FILM

Titre	MetteurEnScene	Genre
Les Voleurs	Techine	Comedie Dramatique
ma Saison Preferee	Techine	Comedie Dramatique
Alerte	Peterson	Aventure
Braveheart	Gibson	Aventure

Table 2.17 : Aventure ou Téchiné

5- Les comédies de Balasko

COMEDIEBALASKO = S(MetteurEnScene="Balasko" ET Genre="Comedie")FILM

Titre	MetteurEnScene	Genre
Gazon maudit	Balasko	Comedie

Table 2.18 : Comédies de Balasko

## 2.15 Exercice 2.15 : La projection

1 - Les noms des metteurs en scène

METTEURENSCENES = [MetteurEnScene]FILM

MetteurEnScene
Peterson
Bay
Gibson
Balasko
Reitman
Russel
Techine

Table 2.19 : Nom des metteurs en scène

2 - Les titres des films à l’affiche du cinéma N°1

FILMCINE1 = S(NumeroCinema=1)AFFICHE

TITREFILMCINE1 = [Titre]FILMCINE1

les relations obtenues sont FILMCINE1 2.20 :

NumeroCinema	Titre
1	Alerte
1	Junior

Table 2.20 : FILMCINE1

et TITREFILMCINE1 2.21 :

Titre
Alerte
Junior

Table 2.21 : TitreFilmcine1

3 - Comparaison de requêtes

On obtient : TITREFILM ??

Puis il n’est plus possible de réaliser la sélection demandée car nous n’avons plus aucune information sur le numéro de cinéma. Nous pouvons remarquer sur ce premier exemple de composition d’opérateurs relationnels, que l’ordre d’évaluation des opérateurs à une importance.

4 - Le titre des films qui sont à l’affiche dans au moins un cinéma

Il s’agit ici simplement de réaliser une projection sur l’attribut Titre de la relation AFFICHE.

FILMAFFICHE = [Titre]AFFICHE

Titre
Alerte
Junior
Gazon Maudit
L'Effaceur
Braveheart
Les Voleurs

Table 2.22 : TitreFilmcine

qui nous donne la relation FILMAFFICHE :

Titre
Alerte
Junior
Gazon Maudit
L'Effaceur
Braveheart
Les Voleurs

Table 2.23 : FILMAFFICHE

5 - Le nom des metteurs en scène des films d'aventure

Il s'agit de réaliser une sélection sur les films d'aventure et ensuite de projeter le nom des metteurs en scène.

FILMAVENTURE = S(Genre="Aventure")FILM  
 METTEURAVENTURE= [MetteurEnScene]FILMAVENTURE

Ce qui donne comme relation FILMAVENTURE et la relation METTEURAVENTURE :

Titre	MetteurEnScene	Genre
Alerte	Peterson	Aventure Braveheart
Gibson	Aventure	

Table 2.24 : FILMAVENTURE

MetteurEnScene
Peterson
Gibson

Table 2.25 : METTEURAVENTURE

## 2.16 Exercice 2.16 : La Jointure

1 - Développons chaque étape de la requête. Nous obtenons la relation FILMAVENTURE :

puis la relation DISTAVENTURE :

Titre	MetteurEnScene	Genre
Alerte	Peterson	Aventure
Braveheart	Gibson	Aventure

Table 2.26 : FILMAVENTURE

Titre	NomActeur	MetteurEnScene	Genre
Alerte	Freeman	Peterson	Aventure
Alerte	Hoffman	Peterson	Aventure
Alerte	Russo	Peterson	Aventure
Braveheart	Gibson	Gibson	Aventure
Braveheart	Marceau	Gibson	Aventure

Table 2.27 : DISTAVENTURE

et enfin DISTAVENTURE

NomActeur
Freeman
Hoffman
Russo
Gibson
Marceau

Table 2.28 : DISTAVENTURE

On obtient le nom des acteurs qui jouent dans un film d'aventure.

2 - Pour obtenir les titres des films à l'affiche du Vox, nous sélectionnons les numéros des cinémas dont le nom est Vox, puis réalisons la jointure de la relation obtenue avec la relation AFFICHE, et enfin projetons la relation obtenue sur l'attribut titre.

NUMEROVOX = S(NomCinema="Vox")CINEMA

NUMEROVOX = [NumeroCinema] NUMEROVOX

AFFICHEVOX = NUMEROVOX [NUMEROVOX.NumeroCinema = AFFICHE.NumeroCinema]AFFICHE

AFFICHEVOX = [titre] AFFICHEVOX

Ce qui nous donne au niveau des tables :

NUMEROVOX

NumeroCinema	NomCinema	AdresseCinema	...
1	Vox	Grande Rue	

Table 2.29 : NUMEROVOX

puis NUMEROVOX

NumeroCinema
1

Table 2.30 : NUMEROVOX

et AFFICHEVOX

NumeroCinema	Titre
1	Alerte
1	Junior

Table 2.31 : AFFICHEVOX

et enfin AFFICHEVOX

Titre
Alerte
Junior

Table 2.32 : AFFICHEVOX

**Remarque :** La projection :  $\text{NUMEROVOX} = [\text{NumeroCinema}] \text{NUMEROVOX}$  n'est pas obligatoire. Elle permet simplement ici de travailler sur des relations plus faciles à manipuler. La solution ci-dessous est tout à fait correcte :

```
NUMEROVOX = S(NomCinema="Vox")CINEMA
AFFICHEVOX = NUMEROVOX [NUMEROVOX.NumeroCinema = AFFICHE.NumeroCinema]AFFICHE
AFFICHEVOX = [titre] AFFICHEVOX
```

3 - Pour obtenir le nom des cinémas ayant à l'affiche "Junior", on peut sélectionner dans la relation AFFICHE, les t-uplets pour lesquels `titre = "junior"`, puis effectuer une jointure avec la relation CINEMA pour enfin projeter sur l'attribut `NomCinema`.

```
AFFICHEJUNIOR = S(Titre="Junior")AFFICHE
CINEJUNIOR = AFFICHEJUNIOR [AFFICHEJUNIOR.NumeroCinema = CINEMA.NumeroCinema] CINEMA
CINEJUNIOR = [NomCinema] CINEJUNIOR
```

Ce qui donne en termes de relations :

AFFICHEJUNIOR

NumeroCinema	Titre
1	Junior
3	Junior

Table 2.33 : AFFICHEJUNIOR

puis CINEJUNIOR

NumeroCinema	NomCinema	...	Titre
1	Vox		Junior
3	Plazza		Junior

Table 2.34 : CINEJUNIOR

et enfin CINEJUNIOR :

NomCinema
Vox
Plazza

Table 2.35 : CINEJUNIOR

## 2.17 Exercice 2.17 : L'intersection

1 - Si nous utilisons l'expression suivante pour calculer `FILMA INTER FILMB`.

`FILMA - (FILMA - FILMB)`

Nous obtenons tout d'abord la relation `FILMA-FILMB`

Titre
Alerte
Congo
BadBoys
Candyman2

Table 2.36 : FILMA-FILMB

puis la relation `FILMA -(FILMA - FILMB)` :

Titre
Casper
Harcèlement

Table 2.37 : FILMA -(FILMA - FILMB)

Ce qui correspond bien au résultat attendu.

## 2.18 Exercice 2.18 : La division

1- Les noms acteurs jouant dans tous les films

Pour obtenir le nom des acteurs jouant dans tous les films, il faut d'abord construire une relation contenant les titres de tous les films, puis diviser la relation `DISTRIBUTION` par cette table.

`TITREFILM = [Titre] FILM`

`ACTEURSTOUSFILM = DISTRIBUTION [DISTRIBUTION.Titre/ TITREFILM.Titre] TITREFILM`

Dans le cas de l'exemple de la base `CINEMAS` aucun acteur ne joue dans tous les films, le résultat sera donc une table vide.

2 - Les villes dans lesquelles tous les films sont à l'affiche.

Comme précédemment nous allons définir une relation contenant les titres de tous les films, puis diviser la relation AFFICHE jointe avec CINEMA par cette table.

TITREFILM = [Titre] FILM

CINEAFFICHE = CINEMA [CINEMA.NumeroCinema = AFFICHE.NumeroCinema] AFFICHE

CINEAFFICHE = [Titre, VilleCinema] CINEAFFICHE

VILLETOUTSFILMS = CINEAFFICHE[ CINEAFFICHE.Titre/ TITREFILM.Titre] TITREFILM

## 2.19 Exercice 2.19 : La Jointure externe

Pour définir pour chaque film la liste des noms de cinémas dans lesquels il est à l'affiche avec éventuellement la valeur NULL si le film n'est à l'affiche nulle part, nous avons besoin d'utiliser une jointure externe. Mais attention, cette jointure externe ne doit pas être faite à n'importe quel moment. Si nous commençons pas faire une jointure externe entre FILM et AFFICHE et qu'ensuite nous faisons une jointure interne avec CINEMA pour obtenir le nom du CINEMA nous perdrons le bénéfice de la jointure externe. Nous commençons donc pas faire les jointures nécessaires pour obtenir les informations dont nous avons besoin sur les AFFICHE et les CINEMA et ensuite seulement nous ferons la jointure externe avec FILM. D'où la solution suivante : Pour obtenir pour chaque film, la liste des noms de cinémas dans lesquels il est à l'affiche nous commençons par réaliser une jointure interne des relations CINEMA et AFFICHE. Puis une projection sur les nom et numéro de cinéma. Ce qui nous donne :

NOMCINE = CINEMA [CINEMA.NumeroCinema = AFFICHE.NumeroCinema] AFFICHE

NOMCINE =[NumeroCinema, NomCinema, Titre] NOMCINE

Supposons que les relations AFFICHE

NumeroCinema	Titre
1	Alerte
1	Junior
2	Alerte
2	Gazon Maudit
2	L'Effaceur
4	Alerte

Table 2.38 : AFFICHE

CINEMA,

NumeroCinema	NomCinema	...
1	Vox	
2	Plazza	
3	Plazza	
4	Royal	

Table 2.39 : CINEMA

et FILM, de départ soient les suivantes :

Titre	MetteurEnScene	Genre
Alerte	Peterson	Aventure
Bad Boys	Bay	Policier
Braveheart	Gibson	Aventure
Gazon Maudit	Balasko	Comedie
Junior	Reitman	Comedie
L'Effaceur	Russel	Action

Table 2.40 : FILM

Le résultat de la première requête décrite est :

NumeroCinema	Titre	Nomcinema
1	Alerte	Vox
1	Junior	Vox
2	Alerte	Plazza
2	Gazon Maudit	Plazza
2	L'Effaceur	Plazza
4	Alerte	Royal

Table 2.41 : NOMCINE

Ensuite la jointure externe gauche de la relation FILMTITRE avec la relation obtenue donne :

FILMTITRE = [Titre] FILM

FILMCINE = FILMTITRE + [FILMTITRE.Titre = NOMCINE.titre] NOMCINE

D'où les résultats suivants : FILMTITRE

Titre
Alerte
Bad Boys
Braveheart
Gazon Maudit
Junior
L'Effaceur

Table 2.42 : FILMTITRE

et FILMCINE :

---

NumeroCinema	Titre	Nomcinema
1	Alerte	Vox
1	Junior	Vox
2	Alerte	Plazza
2	Gazon Maudit	Plazza
2	L'Effaceur	Plazza
4	Alerte	Royal
NULL	Bad Boys	NULL
NULL	Braveheart	NULL

Table 2.43 : FILMCINE



## Chapter 3

# SQL : Le langage de définition des données

Corrigés des exercices du chapitre 4

### 3.1 Exercice 4.1 Création simple des tables de la base de données CINEMAS

#### 3.1.1 Création de la base de données Cinemas

Sous MariaDB commencez par créer la base de données Cinemas.



Figure 3.1 : Création d'une BD Cinemas

#### 3.1.2 Création des tables

Puis créez les tables. Pour cela il faut aller sur l'onglet SQL de MariaDB 3.2 .

```
CREATE TABLE 'cinema' (  
  'NumeroCinema' int(11) NOT NULL,  
  'NomCinema' varchar(30) DEFAULT NULL,  
  'AdresseCinema' varchar(30) DEFAULT NULL,  
  'CodePostalCinema' int(11) DEFAULT NULL,  
  'VilleCinema' varchar(20) DEFAULT NULL,  
  'TelephoneCinema' varchar(14) DEFAULT NULL);
```



Figure 3.2 : Création de la table Cinema

```
CREATE TABLE 'affiche' (
  'NumeroCinema' int(11) NOT NULL,
  'Titre' varchar(100) NOT NULL);

CREATE TABLE 'distribution' (
  'Titre' varchar(100) NOT NULL DEFAULT '',
  'NomActeur' varchar(30) NOT NULL DEFAULT '');

CREATE TABLE 'film' (
  'Titre' varchar(100) NOT NULL DEFAULT '',
  'MetteurEnScene' varchar(30) DEFAULT NULL,
  'Genre' varchar(20) DEFAULT NULL);
```

### 3.2 Exercice 4.2

Créez sous Oracle et/ou mysql les requêtes de MODIFICATION des tables de la base de données CINEMAS pour définir l'intégrité référentielle.

Sous Oracle on a les requêtes suivantes :

```
ALTER TABLE CINEMA
ADD CONSTRAINT PK_CINEMA PRIMARY KEY(NUMEROCINEMA)

ALTER TABLE FILM
ADD CONSTRAINT PK_FILM PRIMARY KEY (TITRE)

ALTER TABLE AFFICHE
ADD CONSTRAINT PK_AFFICHE PRIMARY KEY (NUMEROCINEMA, TITRE),
CONSTRAINT FK_FILM_AFFICHE FOREIGN KEY (TITRE)
REFERENCES FILM (TITRE) ON DELETE CASCADE,
CONSTRAINT FK_CINEMA_AFFICHE FOREIGN KEY (NUMEROCINEMA)
REFERENCES CINEMA (NUMEROCINEMA) ON DELETE CASCADE
```

```
ALTER TABLE DISTRIBUTION
ADD CONSTRAINT PK_DISTRIBUTION PRIMARY KEY (TITRE, NOMACTEUR),
CONSTRAINT FK_FILM_DISTRIBUTION FOREIGN KEY (TITRE)
REFERENCES FILM (TITRE) ON DELETE CASCADE
```

Et sous MySQL :

```
ALTER TABLE CINEMA
ADD CONSTRAINT PK_CINEMA PRIMARY KEY(NUMEROCINEMA)
```

```
ALTER TABLE FILM
ADD CONSTRAINT PK_FILM PRIMARY KEY (TITRE)
```

```
ALTER TABLE AFFICHE
ADD CONSTRAINT PK_AFFICHE PRIMARY KEY (NUMEROCINEMA, TITRE),
ADD CONSTRAINT FK_FILM_AFFICHE FOREIGN KEY (TITRE)
REFERENCES FILM (TITRE) ON DELETE CASCADE,
ADD CONSTRAINT FK_CINEMA_AFFICHE FOREIGN KEY (NUMEROCINEMA)
REFERENCES CINEMA (NUMEROCINEMA) ON DELETE CASCADE
```

```
ALTER TABLE DISTRIBUTION
ADD CONSTRAINT PK_DISTRIBUTION PRIMARY KEY (TITRE, NOMACTEUR),
ADD CONSTRAINT FK_FILM_DISTRIBUTION FOREIGN KEY (TITRE)
REFERENCES FILM (TITRE) ON DELETE CASCADE
```

Sous MySQL la syntaxe est légèrement différente :

Relations

Colonne	Contrainte de clé étrangère (INNODB)		
NumeroCinema	cinema	cinema	NUMEROCINEMA
	Nom de la contrainte: FK_CINEMA_AFFICHE		
	ON DELETE	CASCADE	
	ON UPDATE	RESTRICT	
Titre	cinema	film	TITRE
	Nom de la contrainte: FK_FILM_AFFICHE		
	ON DELETE	CASCADE	
	ON UPDATE	RESTRICT	

Figure 3.3 : Vue Des liens entre la table affiche et Cinema et Film



## Chapter 4

# Le langage de manipulation de données

Corrigés des exercices du chapitre 5

Tous les exercices sont faits avec les tables suivantes : La table CINEMA :

La table FILM :

NUMEROCINEMA	NOMCINEMA	ADRESSECINEMA	CODEPOSTALCINEMA	VILLECINEMA	TELEPHONECINEMA
1	Vox	Grande Rue	25000	Besançon	03 81 82 05 69
2	Plazza	Rue des granges	25000	Besançon	03 81 26 35 98
3	Eldorado	Rue St. Lambert	75015	Paris	01 23 56 98 56
4	Pax	Avenue d'Italie	75013	Paris	01 23 56 98 78

Figure 4.1 : Table CINEMA

Les tables AFFICHE et DISTRIBUTION :

TITRE	METTEURENSCENE	GENRE
Casper	Silberling	Comédie
Harcelement	Levinson	Comédie Dramatique
Junior	Reitman	Comédie
Alerte	Peterson	Aventure
Gazon Maudit	Balasko	Comédie
L'Effaceur	Russel	Action
Les Voleurs	Téchiné	Comédie dramatique
Ma saison préférée	Téchiné	Comédie dramatique

Figure 4.2 : Table FILM

TITRE	NOMACTEUR	NUMEROCHINEMA	TITRE
Alerte	Freeman	1	Alerte
Alerte	Hoffman	1	Junior
Alerte	Russo	2	Alerte
Casper	Pullman	2	Gazon Maudit
Casper	Ricci	2	Junior
Gazon Maudit	Abril	2	L'Effaceur
Gazon Maudit	Balasko	3	Alerte
Gazon Maudit	Chabat	3	Junior
Harcelement	Douglas	4	Alerte
Harcelement	Moore	4	Les Voleurs
Junior	De Mornay		
Junior	De Niro		
Junior	Schwarzenegger		
L'Effaceur	Schwarzenegger		

#### 4.1 Exercice 5.1 : Premières requêtes

1 - La liste des noms des metteurs en scène

```
SELECT MetteurEnScene
FROM FILM
```

Pour ne pas avoir deux fois la valeur Téchiné il faut écrire la requête SQL suivante :

```
SELECT DISTINCT MetteurEnScene
FROM FILM
```

METTEURENSCENE
Reitman
Levinson
Balasko
Peterson
Téchiné
Russel
Silberling

2 - Le titre des films qui sont à l'affiche dans au moins un cinéma.

Il s'agit ici simplement de réaliser une projection sur l'attribut **Titre** de la relation **AFFICHE**. Ce qui peut être réalisé par la requête :

```
SELECT DISTINCT Titre
FROM AFFICHE
```

Celle-ci donne le résultat suivant :

TITRE
Les Voleurs
Gazon Maudit
Alerte
Junior
L'Effaceur

3 - La liste de toutes les affiches possibles.

Il faut réaliser le produit cartésien des relations CINEMA et FILM pour obtenir toutes les affiches possibles, une projection sur les attributs adéquats donnera ensuite le résultat demandé.

```
SELECT NumeroCinema, Titre
FROM CINEMA, FILM
```

## 4.2 Exercice 5.2 : Requêtes avec une clause WHERE

1- Les acteurs jouant dans Alerte

```
SELECT *
FROM DISTRIBUTION
WHERE Titre = 'Alerte'
```

Rappelons que cette requête a été décrite avec les opérateurs relationnels par :

```
ALERTE = S(FTitre = "Alerte")DISTRIBUTION
```

2 - Les titres des films à l'affiche du cinéma N°1

```
SELECT Titre
FROM AFFICHE
WHERE NumeroCinema = 1
```

3 - Les titres des films mis en scène par Téchiné

```
SELECT Titre
FROM FILM
WHERE MetteurEnScene = 'Téchiné'
```

4 - Les films d'aventure ou mis en scène par Téchiné

```
SELECT *
FROM FILM
WHERE MetteurEnScene='Téchiné'
OR Genre='Aventure'
```

5 - Les comédies ayant Balasko comme metteur en scène

```
SELECT *
FROM FILM
WHERE MetteurEnScene='Balasko'
AND Genre='Comédie'
```

6 - Le nom des acteurs qui commencent par "D"

```
SELECT NomActeur
FROM DISTRIBUTION
WHERE NomActeur LIKE 'D%'
```

Le critère de sélection fait appel à l'opérateur LIKE qui signifie que le nom de l'acteur doit avoir la même forme que la chaîne de caractères qui suit ce terme. Ici 'D%', ce qui veut dire que l'on a à faire à une chaîne de caractères qui commence par la lettre D, et qui contient ensuite n'importe quelle chaîne de caractères. On a comme résultat :

NOMACTEUR
Douglas
De Mornay
De Niro

Figure 4.3 : Exo 2 - Question 6

7 - Le nom et le numéro des cinémas pour lesquels on ne connaît pas le numéro de téléphone

```
SELECT NomCinema, NumeroCinema
FROM CINEMA
WHERE TelephoneCinema IS NULL
```

On ne peut pas écrire directement `NumeroTelephone = NULL` ou `<> NULL`. Il faut utiliser les expressions `IS NULL` ou `IS NOT NULL`. 8 - Le nom des cinémas dont le code postal est compris entre 25000 et 25999

Nous avons ici deux possibilités pour traduire le critère de sélection :

```
SELECT NomCinema
FROM CINEMA
WHERE CodePostalCinema >= 25000
AND CodePostalCinema <= 25999
ou
SELECT NomCinema
FROM CINEMA
WHERE CodePostalCinema BETWEEN 25000 AND 25999
```

### 4.3 Exercice 5.3 : La jointure

Jointures internes

1 - Le nom des cinémas ayant au moins un film à l'affiche

Pour obtenir le nom des cinémas nous avons besoin de la table CINEMA, et pour savoir si un cinéma a au moins un film à l'affiche nous avons besoin de la table AFFICHE

```
SELECT DISTINCT NomCinema
FROM CINEMA, AFFICHE
WHERE CINEMA.NumeroCinema = AFFICHE.NumeroCinema
```

ou une solution utilisant l'ordre JOIN :

```
SELECT DISTINCT NomCinema
FROM CINEMA INNER JOIN AFFICHE
ON CINEMA.NumeroCinema = AFFICHE.NumeroCinema
```

2 - Les titres des films à l'affiche du Vox

NOMCINEMA
Eldorado
Vox
Plazza
Pax

```
SELECT Titre
FROM CINEMA, AFFICHE
WHERE NomCinema = 'Vox'
AND CINEMA.NumeroCinema = AFFICHE.NumeroCinema
```

ou une solution utilisant l'ordre JOIN :

```
SELECT Titre
FROM CINEMA INNER JOIN AFFICHE ON CINEMA.NumeroCinema = AFFICHE.NumeroCinema
WHERE NomCinema = 'Vox'
```

TITRE
Alerte
Junior

3 - Le nom des cinémas ayant à l'affiche Junior

```
SELECT NomCinema
FROM CINEMA, AFFICHE
WHERE
CINEMA.NumeroCinema = AFFICHE.NumeroCinema
AND Titre = 'Junior'
```

ou une solution utilisant l'ordre JOIN :

```
SELECT NomCinema
FROM CINEMA INNER JOIN AFFICHE
ON CINEMA.NumeroCinema = AFFICHE.NumeroCinema
WHERE Titre = 'Junior'
```

Jointures externes

4 - Pour chaque film les noms des acteurs faisant partie de la distribution

Pour obtenir les informations demandées il est nécessaire de réaliser une jointure externe gauche de la table FILM avec la table DISTRIBUTION, ce qui nous donne :

```
SELECT FILM.Titre, NomActeur
FROM FILM LEFT OUTER JOIN DISTRIBUTION
ON FILM.Titre = DISTRIBUTION.Titre
```

Ici il n'y a pas de solution avec une expression de la jointure dans la clause WHERE, puisque ce type de solution ne permet d'exprimer que des jointures internes.

## 4.4 Exercice 5.4 : Les fonctions de groupe

1 - Nombre de films de genre comédie

TITRE	NOMACTEUR
Alerte	Freeman
Alerte	Hoffman
Alerte	Russo
Casper	Pullman
Casper	Ricci
Gazon Maudit	Abril
Gazon Maudit	Balasko
Gazon Maudit	Chabat
Harcelement	Douglas
Harcelement	Moore
Junior	De Mornay
Junior	De Niro
Junior	Schwarzenegger
L'Effaceur	Schwarzenegger
Les Voleurs	-
Ma saison préférée	-

La solution est immédiate, il suffit de compter dans la table FILM, le nombre de films ayant comme genre "Aventure".

```
SELECT COUNT(Titre)
FROM FILM
WHERE Genre = 'Comédie'
```

Remarquons cependant que cette solution est suffisante, car **Titre** est la clé primaire de la table FILM, et donc qu'un même titre ne peut pas apparaître deux fois dans cette table.

2 - Nombre de films à l'affiche à Besançon

Pour obtenir le nombre de films à l'affiche à Besançon, il faut sélectionner les t-uplets correspondants dans la jointure des tables AFFICHE et CINEMA, et les compter. Ce qui nous donne :

```
SELECT COUNT(DISTINCT Titre)
FROM AFFICHE NATURAL JOIN CINEMA
WHERE VilleCinema = 'Besançon'
```

On compte bien évidemment le nombre de titres différents à l'affiche à Besançon.

3 - Le plus petit numéro de cinéma dans lequel "Alerte" est à l'affiche

```
SELECT MIN(NumeroCinema)
FROM AFFICHE
WHERE Titre = 'Alerte'
```

## 4.5 Exercice 5.5 : Le regroupement

Regroupements sur une table

1 - Pour chaque acteur le nombre de films dans lesquels il joue.

Pour obtenir le nombre de films dans lesquels joue chaque acteur, il faut regrouper la table DISTRIBUTION par rapport à l'attribut NomActeur, et ensuite compter combien on a de t-uplets pour chaque groupe.

```
SELECT NomActeur, COUNT(*) AS NBFilms
FROM DISTRIBUTION
GROUP BY NomActeur
```

Le regroupement donne la table intermédiaire 4.4

NOMACTEUR	TITRE
Freeman	Alerte
Hoffman	Alerte
Russo	Alerte
Pullman	Casper
Ricci	Casper
Abril	Gazon Maudit
Balasko	Gazon Maudit
Chabat	Gazon Maudit
Douglas	Harcelement
Moore	Harcelement
De Mornay	Junior
De Niro	Junior
Schwarzenegger	Junior
	L'Effaceur

Figure 4.4 : Table intermédiaire

et pour finir le résultat 4.5

2 - Pour chaque film, le premier acteur par ordre alphabétique.

Ici au contraire nous allons effectuer un regroupement sur les titres.

```
SELECT Titre, Min(NomActeur)
FROM DISTRIBUTION
GROUP BY Titre
```

Le regroupement donne pour résultat 4.6 :

### Regroupement et jointure

3 - Pour chaque cinéma, le nombre de films à l'affiche.

Pour bien expliquer les problèmes posés par cet exemple, nous ajoutons dans la table CINEMA, pour notre exemple, un Cinéma N°5 appelé Stella.

NOMACTEUR	NBFILMS
Hoffman	1
Pullman	1
Balasko	1
Russo	1
Moore	1
Schwarzenegger	2
De Mornay	1
Douglas	1
De Niro	1
Freeman	1
Ricci	1
Chabat	1
Abril	1

Figure 4.5 : Exercice 5 - Question 1

TITRE	MIN(NOMACTEUR)
Alerte	Freeman
Casper	Pullman
Gazon Maudit	Abril
Harcelement	Douglas
Junior	De Mornay
L'Effaceur	Schwarzenegger

Figure 4.6 : Requête 2 - Exercice 5

Pour obtenir le nombre de titres à l’affiche de chaque cinéma, ainsi que le nom du cinéma, nous devons partir de la jointure CINEMA et AFFICHE, puis regrouper par cinéma (NumeroCinema, et NomCinema), et enfin compter le nombre de t-uplets dans chaque groupe.

```
SELECT NomCinema, COUNT(*)AS NBTitres
FROM CINEMA JOIN AFFICHE ON CINEMA.NumeroCinema = AFFICHE.NumeroCinema
GROUP BY CINEMA.NumeroCinema, NomCinema
```

Le regroupement doit porter sur les deux attributs NumeroCinema, et NomCinema, car :

- on veut pouvoir donner en résultat le nom du cinéma, ce qui n’est pas possible s’il ne fait pas partie des attributs de regroupement,
- d’autre part si le regroupement est réalisé sur seulement le nom de cinéma on risque d’avoir des problèmes car un même nom peut correspondre à plusieurs cinémas.

On obtient ainsi 4.7

Cette première solution donne pour chaque cinéma le nombre de films qui y sont à l’affiche à la condition qu’il y ait au moins un film à l’affiche. Pour pouvoir traiter le cas où il n’y aurait pas de film à l’affiche d’un cinéma (cinéma N°5), il faut réaliser une jointure externe, et ensuite compter

NOMCINEMA	NBTITRES
Vox	2
Plazza	4
Eldorado	2
Pax	2

Figure 4.7 : Requête 3-1 Exercice 5

dans le regroupement combien il y a de Titres différents de NULL. Ce qui nous donne la requête suivante :

```
SELECT NomCinema, COUNT(Titre)AS NBTitres
FROM CINEMA LEFT OUTER JOIN AFFICHE ON CINEMA.numeroCinema = AFFICHE.numeroCinema
GROUP BY CINEMA.NumeroCinema, NomCinema
```

La fonction COUNT porte ici sur l'attribut Titre car on veut savoir combien de fois la valeur de l'attribut Titre sera différente de NULL.

On obtient ainsi 4.8 :

NOMCINEMA	NBTITRES
Vox	2
Plazza	4
Eldorado	2
Pax	2
Stella	0

Figure 4.8 : Exercice 5 - Question 3 2° solution

## 4.6 Exercice 5.6 : La clause HAVING

1 - le nom des acteurs qui jouent dans au moins deux films

Pour obtenir le nom des acteurs qui jouent dans au moins deux films, il faut regrouper la table DISTRIBUTION par acteur, et ensuite compter combien on a de titres pour chaque acteur, et enfin sélectionner les groupes représentant un acteur qui joue dans au moins deux films.

```
SELECT NomActeur
FROM DISTRIBUTION
GROUP BY NomActeur
HAVING COUNT(Titre) >=2
```

On aurait pu aussi écrire la solution suivante :

```
SELECT NomActeur
FROM DISTRIBUTION
GROUP BY NomActeur
HAVING COUNT(*) >=2
```

qui est équivalente, car pour un acteur donné, un même titre ne peut pas apparaître deux fois ( (Titre, NomActeur) étant la clé de la table DISTRIBUTION). Compter le nombre de ligne :

COUNT(\*) ou compter le nombre de titres : COUNT(Titre), revient au même dans ce cas particulier.

2 - les titres des films qui ont au plus deux acteurs dans leur distribution

```
SELECT Titre
FROM DISTRIBUTION
GROUP BY Titre
HAVING COUNT(NomActeur) <=2
```

De la même façon nous pouvons utiliser ici un COUNT(\*) car pour un même titre un nom d'acteur ne peut apparaître qu'une seule fois.

## 4.7 Exercice 5.7 : La clause ORDER BY

1 - Pour chaque acteur le nombre de films dans lesquels il joue, dans l'ordre décroissant du nombre de films.

```
SELECT NomActeur, COUNT(Titre) AS NBFilms
FROM DISTRIBUTION
GROUP BY NomActeur
ORDER BY NBFilms DESC
```

NOMACTEUR	NBFILMS
Schwarzenegger	2
Pullman	1
Balasko	1
Russo	1
Moore	1
Abril	1
Douglas	1
De Niro	1
Freeman	1
Ricci	1
Chabat	1
Hoffman	1
De Mornay	1

Figure 4.9 : Exercice 6 - Question 1

2 - les informations concernant les films en les triant par genre, et par titre

```
SELECT *
FROM FILM
ORDER BY Genre, Titre
```

TITRE	METTEURENSCENE	GENRE
L'Effaceur	Russel	Action
Alerte	Peterson	Aventure
Casper	Silberling	Comédie
Gazon Maudit	Balasko	Comédie
Junior	Reitman	Comédie
Harcelement	Levinson	Comédie Dramatique
Les Voleurs	Téchiné	Comédie dramatique
Ma saison préférée	Téchiné	Comédie dramatique

#### 4.8 Exercice 5.8 : Les sous-requêtes indépendantes

1 - Les noms des acteurs des films mis en scène par Balasko

Nous décrivons ici une sous-requête qui donne les titres des films de Téchiné, à partir de là, la requête principale précise simplement que l'acteur doit faire partie de la distribution d'un de ces films.

```
SELECT NomActeur
FROM DISTRIBUTION
WHERE Titre IN
(SELECT Titre
FROM FILM
WHERE MetteurEnScene = 'Balasko')
```

On peut aussi proposer la même solution avec l'opérateur ANY. ce qui donne :

```
SELECT NomActeur
FROM DISTRIBUTION
WHERE Titre = ANY
(SELECT Titre
FROM FILM
WHERE MetteurEnScene = 'Balasko')
```

NOMACTEUR
Abril
Balasko
Chabat

Figure 4.10 : Exercice 7 - Question 1

2 - Les titres des films à l'affiche du cinéma Vox

```
SELECT Titre
FROM AFFICHE
WHERE NumeroCinema IN
(SELECT NumeroCinema
FROM CINEMA
WHERE NomCinema = 'Vox')
```

Attention, il n'est pas possible ici d'utiliser directement l'opérateur =, car la sous-requête peut donner plusieurs t-uplets, le nom d'un cinéma n'étant pas unique.

3 - Les acteurs qui n'ont jamais joué dans un film mis en scène par Téchiné

```
SELECT NomActeur
FROM DISTRIBUTION
WHERE NomActeur NOT IN
  (SELECT NomActeur
   FROM FILM, DISTRIBUTION
   WHERE DISTRIBUTION.Titre = FILM.Titre
   AND MetteurEnScene = 'Téchiné')
```

ou encore avec l'opérateur ALL

```
SELECT NomActeur
FROM DISTRIBUTION
WHERE NomActeur <> ALL
  (SELECT NomActeur
   FROM FILM, DISTRIBUTION
   WHERE DISTRIBUTION.Titre = FILM.Titre
   AND MetteurEnScene = 'Téchiné')
```

4 - Les acteurs jouant dans un film avec un acteur qui a joué dans L'Effaceur

La solution suppose d'utiliser deux sous-requêtes imbriquées. La première donne la liste des acteurs participant au film L'Effaceur, la deuxième donne les titres des films qui ont des acteurs résultat de la première requête.

```
SELECT DISTINCT NomActeur
FROM DISTRIBUTION
WHERE Titre in
  (SELECT Titre
   FROM DISTRIBUTION
   WHERE NomActeur IN
     (SELECT NomActeur
      FROM DISTRIBUTION
      WHERE Titre = 'L''Effaceur'))
```

## 4.9 Exercice 5.9 : Les sous-requêtes synchronisées

1 - Les noms des acteurs des films mis en scène par Téchiné

```
SELECT NomActeur
FROM DISTRIBUTION
WHERE EXISTS
  (SELECT Titre
   FROM FILM
   WHERE MetteurEnScene = 'Téchiné'
   AND DISTRIBUTION.Titre = FILM.Titre)
```

2 - Les titres des films à l'affiche du cinéma Vox

```
SELECT Titre
FROM AFFICHE
WHERE EXISTS
  (SELECT *
   FROM CINEMA
```

```
WHERE NomCinema = 'Vox'
AND AFFICHE.NumeroCinema =
  CINEMA.NumeroCinema)
```

TITRE
Alerte
Junior

Figure 4.11 : Exercice 9 - Question 1

**Remarque :** les attributs projetés dans la requête synchronisés n'ont aucune importance. En effet ce qui nous intéresse lors de l'évaluation de la sous-requête c'est de savoir si elle retourne au moins un t-uplet.

3 - Les noms des acteurs qui n'ont jamais joué dans un film mis en scène par Reitman

```
SELECT NomActeur
FROM DISTRIBUTION D1
WHERE NOT EXISTS
  (SELECT * FROM DISTRIBUTION D2, FILM F
   WHERE F.MetteurEnScene = 'Reitman' AND F.Titre = D2.Titre
   AND D2.NomActeur = D1.NomActeur)
```

NOMACTEUR
Pullman
Hoffman
Russo
Abril
Douglas
Ricci
Freeman
Balasko
Chabat
Moore

Figure 4.12 : Exercice 9 - Question 2

4 - Le nom du premier acteur par ordre alphabétique

```
SELECT D1.NomActeur
FROM DISTRIBUTION D1
WHERE NOT EXISTS
  (SELECT NomActeur
   FROM DISTRIBUTION D2
   WHERE D2.NomActeur < D1.NomActeur)
```

NOMACTEUR
Abril

Figure 4.13 : Exercice 9 - Question 4

## 4.10 Exercice 5.10 : Les opérateurs ensemblistes

1- Les noms des acteurs des films mis en scène par Téchiné ou Peterson

Il s'agit ici de réaliser l'union de deux requêtes retournant les films mis en scène, respectivement, par Téchiné et Peterson.

```
SELECT NomActeur
FROM DISTRIBUTION, FILM
WHERE DISTRIBUTION.Titre = FILM.Titre
AND MetteurEnScene = 'Téchiné'
UNION
SELECT NomActeur
FROM DISTRIBUTION, FILM
WHERE DISTRIBUTION.Titre = FILM.Titre
AND MetteurEnScene = 'Peterson'
```

2 - Les titres des films à l'affiche du cinéma Vox et du cinéma N° 3

Il s'agit ici de réaliser l'intersection de deux requêtes retournant les films à l'affiche du Vox et du cinéma N°3.

```
SELECT Titre
FROM AFFICHE
WHERE NumeroCinema = 3
INTERSECT
SELECT Titre
FROM CINEMA NATURALJOIN Affiche
WHERE CINEMA.NomCinema = 'Vox'
```

3 - Les noms des acteurs qui n'ont jamais joué dans un film mis en scène par Reitman

Nous finissons par une différence entre la liste des acteurs et ceux qui ont déjà joué pour Reitman.

```
SELECT DISTINCT NomActeur
FROM DISTRIBUTION
MINUS
SELECT NomActeur
FROM DISTRIBUTION, FILM
WHERE DISTRIBUTION.Titre = FILM.Titre
AND MetteurEnScene = 'Reitman'
```

## 4.11 Exercice 5.11 : La division

1- Les noms acteurs jouant dans tous les films

Pour obtenir les noms des acteurs jouant dans tous les films, il faut diviser la table distribution par une table donnant tous les titres de films. Ce que nous exprimons en SQL des deux façons suivantes :

```
SELECT NomActeur
FROM DISTRIBUTION
```

```
GROUP BY NomActeur
Having COUNT(*) = (SELECT COUNT(*) FROM FILM)
```

ou

```
SELECT D1.NomActeur
FROM DISTRIBUTION D1
WHERE NOT EXISTS
  (SELECT Titre
   FROM FILM
   WHERE NOT EXISTS
    (SELECT *
     FROM DISTRIBUTION D2
     WHERE D1.NomActeur = D2.NomActeur
     AND FILM.Titre = D2.Titre))
```

2 - Les villes dans lesquelles tous les films sont à l'affiche

```
SELECT VilleCinema
FROM CINEMA, AFFICHE
WHERE CINEMA.NumeroCinema=AFFICHE.NumeroCinema
GROUP BY VilleCinema
Having COUNT(DISTINCT Titre) =
  (SELECT COUNT(*) FROM FILM)
```

ou

```
SELECT C1.VilleCinema
FROM CINEMA C1 INNER JOIN AFFICHE A1 ON C1.NumeroCinema=A1.NumeroCinema
WHERE NOT EXISTS
  (SELECT Titre
   FROM FILM
   WHERE NOT EXISTS
    (SELECT *
     FROM CINEMA C2 INNER JOIN AFFICHE A2 ON C2.NumeroCinema = A2.NumeroCinema
     WHERE C1.VilleCinema = C2.VilleCinema
     AND FILM.Titre = A2.Titre))
```

3 - Le nom des acteurs qui ont déjà joué dans tous les genres de films

```
SELECT NomActeur
FROM DISTRIBUTION, FILM
WHERE DISTRIBUTION.Titre = FILM.Titre
GROUP BY NomActeur
HAVING COUNT(DISTINCT Genre) =
  (SELECT COUNT(DISTINCT Genre) FROM FILM)
```

ou

```
SELECT D1.NomActeur
FROM DISTRIBUTION D1
WHERE NOT EXISTS
  (SELECT F1.Genre
   FROM FILM F1
   WHERE NOT EXISTS
    (SELECT D2.NomActeur
     FROM DISTRIBUTION D2 INNER JOIN FILM F2 ON D2.Titre = F2.Titre
```

```
WHERE F1.Genre = F2.Genre
AND D2.NomActeur = D1.NomActeur))
```

## 4.12 Exercice 5.12 : Les expressions

1 - Liste des titres mieux notés par le public que par la critique.

```
SELECT Titre
FROM FILM
WHERE NoteCritique <= NotePublic
```

TITRE
Casper
Junior
Gazon Maudit
L'Effaceur
Ma saison préférée

Figure 4.14 : Exercice 12 - Question 1

2 - Liste des films qui ont une note moyenne supérieure à 12, avec leur moyenne

```
SELECT Titre, (NoteCritique+NotePublic)/2 AS Moyenne
FROM FILM
WHERE (NoteCritique +NotePublic)/2 >= 12
```

TITRE	MOYENNE
Harcelement	12,5
Alerte	13,5
Gazon Maudit	14,5
Ma saison préférée	12

Figure 4.15 : Exercice 12 - Question 2

# Chapter 5

## La normalisation

Corrigés des exercices du chapitre 7

### 5.1 Exercice 7.1 : Les dépendances fonctionnelles

#### 1 - Les dépendances fonctionnelles

Nous avons déjà donné dans le cours deux dépendances fonctionnelles :

- `NumeroCinema`  $\rightarrow$  `NomCinema`
- `NumeroCinema`  $\rightarrow$  `AdresseCinema`, `VilleCinema`, `CodePostalCinema`, `TelephoneCinema`

Une première remarque concernant les dépendances fonctionnelles. La deuxième dépendance peut être lue de la façon suivante :

`NumeroCinema`  $\rightarrow$  `AdresseCinema`

et

`NumeroCinema`  $\rightarrow$  `VilleCinema`

et ...

Les deux premières dépendances fonctionnelles ont le même déterminant et peuvent donc être écrites ensemble (en une seule DF). Maintenant considérons le reste des informations dans la relation `CINEMA`, et trouvons éventuellement d'autres dépendances fonctionnelles.

Nous avons la dépendance fonctionnelle suivante :

`Titre`  $\rightarrow$  `MetteurEnScene`, `Genre`

Ce qui signifie qu'un film a un unique metteur en scène, et n'appartient qu'à un seul genre.

Autre dépendance : Nous supposons qu'un code postal correspond à une unique ville, d'où :

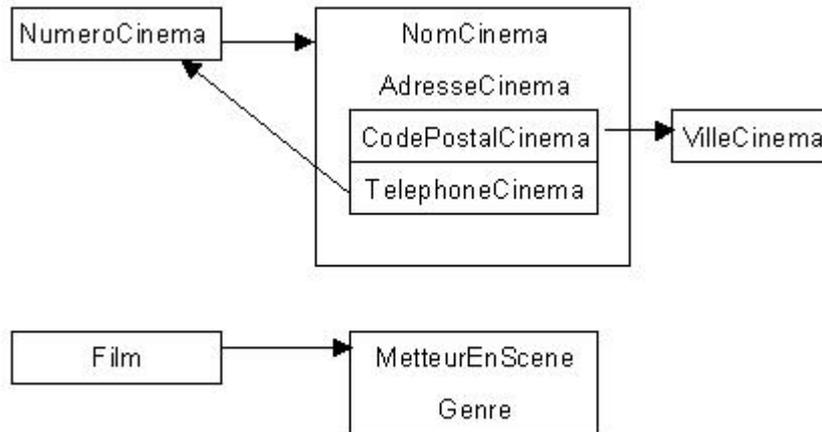
`CodePostalCinema`  $\rightarrow$  `VilleCinema`

**Remarque :** Dans la majorité des cas un code postal correspond à une unique ville. Pour notre exemple nous supposons que ceci est toujours le cas.

De même si nous supposons qu'un numéro de téléphone ne peut appartenir qu'à un seul Cinema nous avons :

TelephoneCinema  $\rightarrow$  NumeroCinema

Ce que nous pouvons représenter de la façon suivante :



## 2 - Dépendances fonctionnelles totales de la relation CINEMA

Nous avons trouvées les DFs suivantes :

- TelephoneCinema  $\rightarrow$  NumeroCinema
- CodePostalCinema  $\rightarrow$  VilleCinema
- NumeroCinema  $\rightarrow$  NomCinema
- NumeroCinema  $\rightarrow$  AdresseCinema, VilleCinema, CodePostalCinema, TelephoneCinema
- Titre  $\rightarrow$  MetteurEnScene, Genre

Toutes ces dépendances sont totales, car le déterminant est toujours limité à un seul attribut.

## 5.2 Exercice 7.2 Les règles d'Amstrong

On initialise l'ensemble de dépendances à  $A^+=A$ , B, ensuite nous répétons pour chaque dépendance fonctionnelle donnée le traitement décrit dans la boucle :

### Premier Passage

DF1 :  $A \rightarrow BC$ , ici  $A \in A^+$ , donc nous ajoutons à  $A^+$ , B, C (partie droite de la DF).

$A^+ = A, B, C$

DF2 :  $E \rightarrow CF$ ,  $E \notin A^+$  donc  $A^+$  reste inchangé.

DF3 :  $B \rightarrow E$ ,  $B \in A^+$  donc nous ajoutons E à  $A^+$ .

$A^+ = A, B, C, E$

DF4 :  $CD \rightarrow EF$ ,  $D \notin A^+$ , donc  $A^+$  reste inchangé.

### Deuxième passage

A+ a été modifié lors du premier passage, nous recommençons donc le traitement avec toutes les DFs et la nouvelle valeur de A+.

DF1 : marquée

DF2 :  $E \rightarrow CF$ ,  $E \in A+$ , on ajoute donc F à A+.

A+= A, B, C, E, F

DF3 marquée

A l'issue de ce passage toutes les DFs ne sont pas marquées et il y a eu un changement dans A+.

Nous avons A+ = A, B, C, E, F

**Troisième passage**

DF1 : marquée

DF2 : marquée

DF3 : marquée

DF4 n'est pas marquée, mais D n'appartient toujours pas à A+.

A l'issue de ce passage toutes les DFs ne sont pas marquées et il n'y a pas eu un changement dans A+. Nous avons A+ = A, B, C, E, F  $D \notin A+$ , l'ensemble d'attributs A,B ne définit donc pas fonctionnellement tous les attributs de la relation R : A, B n'est pas une clé candidate de la relation R.

### 5.3 Exercice 7.3 : Le théorème de décomposition

Décomposition de la relation CINEMA en utilisant la dépendance fonctionnelle DF1. Le théorème de décomposition nous permet de décomposer la relation CINEMA en deux relations :

- CINE : qui contient tous les attributs apparaissant dans la dépendance fonctionnelle DF1, avec comme clé le déterminant de DF1, c'est à dire : NumeroCinema.
- CINEMA1 : qui contient tous les attributs de départ sauf ceux apparaissant en partie droite de la dépendance fonctionnelle DF1.

Nous obtenons donc les 2 relations suivantes :

CINE(\*NumeroCinema, NomCinema, AdresseCinema, VilleCinema, CodePostalCinema, TelephoneCinema)

CINEMA1(\*NumeroCinema ,\* Titre, Genre, MetteurEnScene, \*NomActeur)

Décomposition de la relation CINEMA1 en utilisant la dépendance fonctionnelle DF2. Comme précédemment le théorème de décomposition nous permet de décomposer la relation CINEMA1 en deux relations:

- FILM : qui contient tous les attributs apparaissant dans la dépendance fonctionnelle DF2, avec comme clé le déterminant de DF2, c'est à dire : Titre.
- CINEMA2 : qui contient tous les attributs de la relation CINEMA1 sauf ceux apparaissant en partie droite de la dépendance fonctionnelle DF2.

Nous obtenons donc les 2 relations suivantes :

FILM(\* Titre, Genre, MetteurEnScene)

CINEMA2(\*NumeroCinema,\* Titre,\*NomActeur)

Ce qui nous donne pour finir l'ensemble de relations :

CINE(\*NumeroCinema, NomCinema, AdresseCinema, VilleCinema, CodePostalCinema,

TelephoneCinema)  
 FILM(\* Titre, Genre, MetteurEnScene)  
 CINEMA2(\*NumeroCinema,\* Titre,\*NomActeur)

Les tables obtenues contiennent les informations suivantes, si l'on prend comme point de départ la relation CINEMA présentée dans le cours :

NumeroCinema	NomCinema	AdresseCinema	VilleCinema
1	Vox	Grande Rue	Besançon
2	Plazza	Rue Des Granges	Besançon
3	Eldorado	Rue St. Lambert	Paris
4	Pax	Avenue d'Italie	Paris

Table 5.1 : Table CINE

*Titre	MetteurEnScene	Genre
Junior		
Alerte		
L'Effaceur		
Gazon Maudit		
Casper		
Harcelement		

Table 5.2 : Table FILM

La jointure de ces trois tables :

$X = \text{CINEMA2}[\text{CINEMA2.NomCinema} = \text{CINE.NomCinema}]\text{CINE}$   
 $Y = X[X.Titre = \text{FILM.Titre}]\text{FILM}$

donne la table CINEMA de départ.

## 5.4 Exercice 7.4 : La forme 2NF

### 5.4.1 Les prêts de K7

La relation PRETK7 n'est pas 2NF à cause des DF : DF1 et DF2. En effet Ces deux DFs ont pour déterminant une partie de la clé de PRETK7. Il n'y a donc pas de dépendance totale entre la clé de cette relation et les attributs non clés.

Décomposition de la relation PRETK7 en utilisant la dépendance fonctionnelle DF1

Le théorème de décomposition nous permet de décomposer la relation PRETK7 en deux relations :

- CLIENT : qui contient tous les attributs apparaissant dans la dépendance fonctionnelle DF1, avec comme clé le déterminant de DF1, c'est à dire : Numcli.
- PRET1 : qui contient tous les attributs de départ sauf ceux apparaissant en partie droite de la dépendance fonctionnelle DF1.

Nous obtenons donc les 2 relations suivantes :

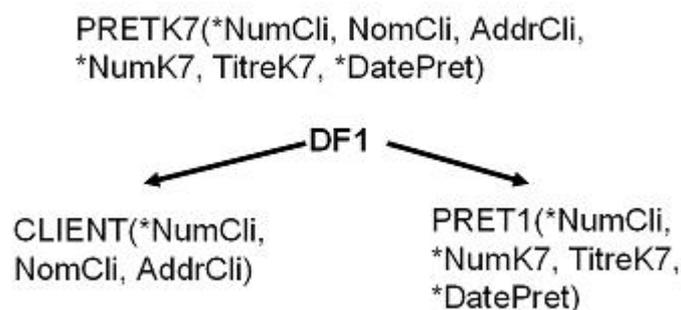
- CLIENT(\*NumCli, NomCli, AddrCli)

*NumeroCinema	Titre	NomActeur
1	Junior	Schwarzenegger
1	Junior	De Mornay
1	Junior	De Niro
1	Alerte	Hoffman
1	Alerte	Russo
1	Alerte	Freeman
2	Alerte	Hoffman
2	Alerte	Russo
2	Alerte	Freeman
3	Alerte	Hoffman
3	Alerte	Russo
3	Alerte	Freeman
4	Alerte	Hoffman
4	Alerte	Russo
4	Alerte	Freeman
2	L'Effaceur	Schwarzenegger
2	Gazon Maudit	Balasko
2	Gazon Maudit	Abril
2	Gazon Maudit	Chabat
3	Casper	Pullman
3	Casper	Ricci
4	Casper	Pullman
4	Casper	Ricci
4	Harcelement	Moore
4	Harcelement	Douglas

Table 5.3 : Table CINEMA2

–  $PRET1(*NumCli, *NumK7, TitreK7, *DatePret)$

Ce que nous pouvons schématiser de la façon suivante :

Figure 5.1 : Décomposition de  $PRETK7$  -  $DF1$ 

Décomposition de la relation  $PRET1$  en utilisant la dépendance fonctionnelle  $DF2$

Comme précédemment le théorème de décomposition nous permet de décomposer la relation  $PRET1$  en deux relations:

- K7 : qui contient tous les attributs apparaissant dans la dépendance fonctionnelle DF2, avec comme clé le déterminant de DF2, c'est à dire : NumK7.
- PRET : qui contient tous les attributs de la relation PRET1 sauf ceux apparaissant en partie droite de la dépendance fonctionnelle DF2.

Nous obtenons donc les 2 relations suivantes :

- K7(\*NumK7, TitreK7)
- PRET(\*NumCli, \*NumK7, \*DatePret)

Ce que nous pouvons schématiser de la façon suivante :

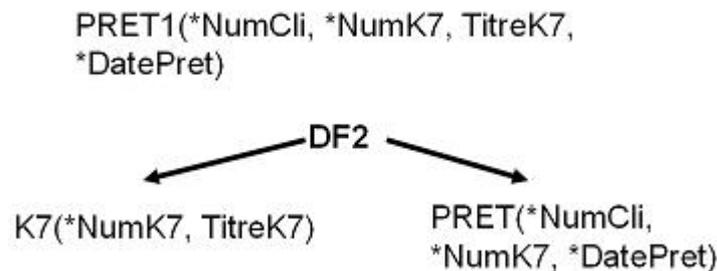


Figure 5.2 : Décomposition de PRET1 - DF2

Ce qui nous donne pour finir l'ensemble de relations :

- CLIENT(\*NumCli, NomCli, AddrCli)
- K7(\*NumK7, TitreK7)
- PRET(\*NumCli, \*NumK7, \*DatePret)

## 5.4.2 Étude de relations non 2NF

### Suppression de t-uplets

Si l'on supprime tous les t-uplets de la relation CINEMA1 faisant référence au titre *Alerte* on obtient la table suivante :

NumeroCinema	Titre	Metteur	Genre	NomActeur
1	Junior	Reitman	Comedie	Schwarzenegger
1	Junior	Reitman	Comedie	De Mornay
1	Junior	Reitman	Comedie	De Niro

Le problème est que l'on a perdu l'information concernant le cinéma *numéro 2*. En retirant des informations concernant des films, on perd en même temps des informations concernant des cinémas. Ceci vient du fait que notre relation contient des informations sur les films et les cinémas; en supprimant des informations concernant un film on risque de supprimer en même temps des informations sur les cinémas.

**Modification du nom d'un Metteur en scène : *Peterson* → *Pëetrsen***

La valeur *Peterson* apparaît plusieurs fois dans notre relation CINEMA1. Cette redondance risque de poser des problèmes lors de mises à jour. Ainsi, la modification proposée ici, peut être réalisée de deux façons :

- Rechercher tous les t-uplets contenant la valeur *Peterson*, et les modifier (ce qui peut prendre beaucoup de temps).
- Générer une table incohérente dans laquelle on pourrait trouver des t-uplets avec la valeur *Peterson* et d'autres avec la valeur *Pëtrson*.

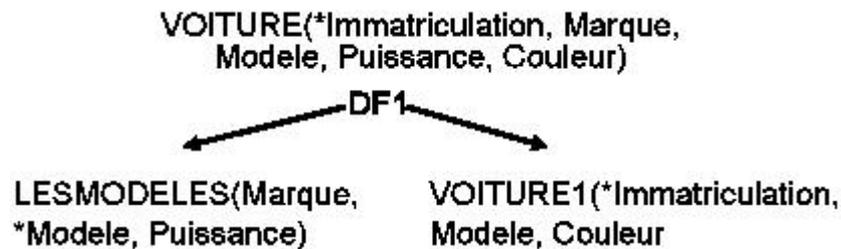
Ceci permet de décrire de façon informelle la décomposition des relations : on place séparément des informations logiquement différentes.

## 5.5 Exercice 7.5 : Les voitures

Considérons que la relation **VOITURE** est 1NF et 2NF. La dépendance fonctionnelle DF1 a pour déterminant un attribut non clé et comme partie droite des attributs non clés. Nous avons donc une dépendance entre attributs non clés. la relation n'est donc pas 3NF. Décomposition de la relation **VOITURE** en utilisant la dépendance fonctionnelle DF1. Le théorème de décomposition nous permet de décomposer la relation **VOITURE** en deux relations :

- **LESMODELES** : qui contient tous les attributs apparaissant dans la dépendance fonctionnelle DF1, avec comme clé le déterminant de DF1, c'est à dire : **Modele**.
- **VOITURE1** : qui contient tous les attributs de départ sauf ceux apparaissant en partie droite de la dépendance fonctionnelle DF1.

Ce que nous pouvons schématiser par :



Nous obtenons donc les 2 relations suivantes :

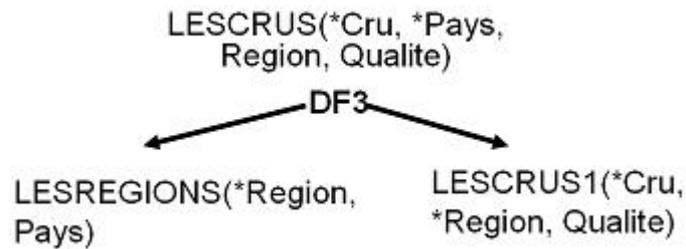
LESMODELES(Marque, \*Modele, Puissance)  
VOITURE1(\*Immatriculation, Modele, Couleur)

## 5.6 Exercice 7.6 : Les crus

La dépendance fonctionnelle DF3, implique que notre relation **LESCRUS** n'est pas BCNF. Le théorème de décomposition nous propose une décomposition en deux relations comme suit :

- **LESREGIONS** : qui contient tous les attributs apparaissant dans la dépendance fonctionnelle DF3, avec comme clé le déterminant de DF3, c'est à dire : **Region**.
- **LESCRUS1** : qui contient tous les attributs de départ sauf ceux apparaissant en partie droite de la dépendance fonctionnelle DF3.

Nous obtenons donc les 2 relations suivantes : **LESREGIONS**(\*Region, Pays)  
**LESCRUS1**(\*Cru, \*Region, Qualite)  
Que nous pouvons schématiser par :



## 5.7 Exercice 7.7 : Les cours et les sports

La relation ETUDESSPORT est-elle 4 NF? Avons nous une dépendance multi-valuée dans la relation ETUDESSPORT ? Si oui laquelle ?

1. Avons nous par exemple la dépendance ?  $\text{Cours} \twoheadrightarrow \text{NumeroEtudiant} / \text{Sport}$   
NON, car s'il l'on considère les t-uplets  $(11, BD1, Tennis)$  et  $(10, BD1, Vélo)$ . La DM devrait impliquer l'existence du t-uplet  $(11, BD1, Vélo)$  ce qui n'est pas le cas.
2. Avons nous par exemple la dépendance ?  $\text{Sport} \twoheadrightarrow \text{NumeroEtudiant} / \text{Cours}$   
NON, car s'il l'on considère les t-uplets  $(11, BD1, Tennis)$  et  $(10, MCOO, Tennis)$ . La DM devrait impliquer l'existence du t-uplet  $(11, MCOO, Tennis)$  ce qui n'est pas le cas.
3. Il est immédiat que nous avons une dépendance multivaluée de la forme :  
DM1 :  $\text{NumeroEtudiant} \twoheadrightarrow \text{Cours} / \text{Sport}$

Le déterminant de cette dépendance multivaluée :  $\text{NumeroEtudiant}$ , n'est pas une super-clé de la relation ETUDESSPORT. Cette relation n'est donc pas 4NF.

**Transformation en 4 NF :**

Nous décomposons notre relation en deux relations en utilisant le théorème de Fagin, nous obtenons :

- LESCOURS : qui contient tous les attributs apparaissant dans la première partie droite de la dépendance multivaluée, avec comme clé le déterminant de cette dépendance, c'est à dire :  $\text{NumeroEtudiant}$ .
- LESSPORTS : qui contient tous les attributs apparaissant dans la deuxième partie droite de la dépendance multivaluée (dépendance complémentaire), avec comme clé le déterminant de cette dépendance, c'est à dire :  $\text{NumeroEtudiant}$ .

Nous obtenons donc les 2 relations suivantes 5.3 :

LESCOURS(\*NumeroEtudiant, \*Cours)  
LESSPORTS(\*NumeroEtudiant, \*Sport)

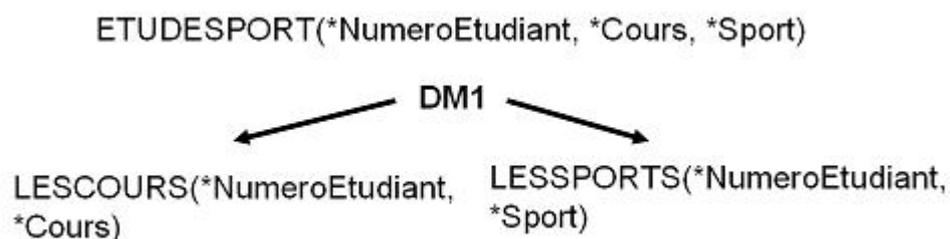


Figure 5.3 : Décomposition en 4NF

---

## 5.8 Exercice 7.8 : Fournisseurs - Pièces et Projets

### Les anomalies d'insertion

Il est dit en hypothèse que la relation FPP satisfait la dépendance de jointure  $((\ast\text{Fournisseur}, \ast\text{Piece}), (\ast\text{Fournisseur}, \ast\text{Projet}), (\ast\text{Piece}, \ast\text{Projet}))$ . Ce qui signifie que nos informations peuvent être stockées grâce aux relations suivantes :

*Fournisseur	*Piece
Dupont	X23
Dupont	X30

*Fournisseur	*Projet
Dupont	P01
Dupont	P02

*Piece	*Projet
X23	P01
X30	P02

Figure 5.4 : Décomposition de FPP

Si l'on veut introduire le t-uplet  $(Durand, X23, P02)$ , on ajoute dans les relations ci-dessus les t-uplets :

*Fournisseur	*Piece
Dupont	X23
Dupont	X30
Durand	X23

*Fournisseur	*Projet
Dupont	P01
Dupont	P02
Durand	P02

*Piece	*Projet
X23	P01
X30	P02
X23	P02

Figure 5.5 : Anomalie d'insertion

Si on recompose ces informations, on voit bien que dans la dernière table nous avons introduit le fait que le projet  $P02$  utilise la pièce  $X23$ , ce que l'on doit retrouver quelque soit le fournisseur du projet  $P02$ , et donc le fournisseur  $Dupont$ . On obtient par recombinaison la table FPP :

*Fournisseur	*Piece	*Projet
Dupont	X23	P01
Dupont	X30	P02
Durand	X23	P01
Dupont	X23	P02

Table 5.4 : Relation FPP

On peut remarquer que lorsque que l'on a une telle dépendance alors l'existence d'un t-uplet dans la relation est dans certains cas prévisible. En effet, si il existe dans la relation FPP :

- un t-uplet  $(f1, p1, pr2)$
- un autre t-uplet  $(f2, p1, pr1)$ ,
- et un autre t-uplet  $(f1, p2, pr1)$ ,
- alors le t-uplet  $(f1, p1, pr1)$  appartient à la relation FPP.