

# Base de Données 2

## Gestion d'un cinéma

Alexis Giraudet

François Hallereau

Valery Morozov

19 mars 2015

# Table des matières

<b>Introduction</b>	<b>4</b>
<b>1 Présentation</b>	<b>5</b>
1.1 Répartition des tâches . . . . .	5
1.2 Diagramme de la base de données . . . . .	5
<b>2 Normalisation</b>	<b>7</b>
2.1 Algorithme de Bernstein . . . . .	7
2.2 Algorithme de décomposition . . . . .	8
<b>Conclusion</b>	<b>9</b>
<b>Annexe</b>	<b>10</b>

## Table des figures

1	Schéma des dépendances fonctionnelles . . . . .	10
2	MCD de la base de données . . . . .	10
3	Algorithme de décomposition . . . . .	11

## Introduction

Dans le cadre du cours de Base de données 2, nous avons décidé de réaliser une base de données pour la gestion d'un cinéma.

Dans ce premier rendu, nous présenterons la table générale initiale, puis comment nous l'avons découpé en plusieurs tables à l'aide de dépendances fonctionnelles et des différents algorithmes de normalisation vu en cours et enfin, le schéma de la base résultant.

# 1 Présentation

## 1.1 Répartition des tâches

Pour ce projet nous avons principalement travaillé en groupe. La première étape a été de trouver le sujet que l'on allait modéliser. Cette étape cruciale est une étape à ne pas prendre à la légère. En effet, avoir un sujet libre ouvre la voie à de multiples possibilités et donc à de bonnes et mauvaises idées. C'est donc ensemble que nous avons pris la décision de partir sur le choix d'un cinéma puisque c'est un sujet concret et précis et qui laisse porte ouverte à des extensions comme nous le verrons plus tard.

Ensuite, nous avons ensemble posé les bases de notre système. Notamment, les entités qui composeraient notre base données. En a découlé la création d'une table reprenant toutes les informations de notre cinéma. Puis, nous avons travaillé sur les dépendances fonctionnelles de notre base de données. Valery a réalisé un diagramme de notre base données, François a fait l'algorithme de Bernstein et Alexis a fait l'algorithme de décomposition. Enfin, nous avons mis en commun nos travaux pour y apporter un regard critique et effectuer des améliorations.

## 1.2 Diagramme de la base de données

Notre base de données est composée de plusieurs entités.

- un **Film** est décrit par un numéro d'identification unique, un titre, une date de sortie en salle et une date de fin de projection. La version indique si le film est en version originale ou française et s'il dispose de sous-titres. Il dispose aussi d'un genre et d'une durée et enfin d'une restriction telle que Interdit aux -16 ans, Tout public, etc.
- une **Salle** est caractérisée par son numéro et le nombre de spectateurs qu'elle peut accueillir.
- une **Séance** dispose d'une date indiquant le jour et l'heure de la projection, une référence au film concerné et une référence à la salle de projection assignée
- un **Tarif** associe un libellé à un prix. Ex : Tarif étudiant = 5€.
- un **Achat** dispose d'un identifiant unique associé à une séance et un tarif.

On en déduit les tuples suivants :

Film (id\_film, titre, date\_début, date\_fin, version, genre, restriction, durée)  
Salle (no\_salle, capacité)  
Séance (date\_séance, no\_salle, id\_film)  
Achat (id\_achat, date\_séance, no\_salle, type\_tarif, date\_achat)  
Tarif (type\_tarif, lib\_tarif, prix)

Auxquels viennent s'ajouter les dépendances fonctionnelles voir figure 1 :

1. id\_film  $\rightarrow$  titre, date\_début, date\_fin, version, genre, restriction, durée
2. no\_salle  $\rightarrow$  capacité
3. id\_achat  $\rightarrow$  date\_séance, no\_salle, type\_tarif date\_achat
4. type\_tarif  $\rightarrow$  lib\_tarif, prix
5. date\_séance, no\_salle  $\rightarrow$  id\_film

Enfin, nous avons modélisé ces données dans le diagramme. Cf. figure 2.

## 2 Normalisation

### 2.1 Algorithme de Bernstein

Nous utilisons l'algorithme sur la relation suivante :

$R\{\text{id\_film}, \text{titre}, \text{date\_début}, \text{date\_fin}, \text{version}, \text{genre}, \text{restriction}, \text{durée}, \text{no\_salle}, \text{capacité}, \text{date\_séance}, \text{no\_salle}, \text{id\_achat}, \text{type\_tarif}, \text{prix}, \text{date\_achat}\}$

avec la clé suivante  $\langle \text{id\_film}, \text{no\_salle}, \text{id\_achat}, \text{type\_tarif}, \text{date\_séance} \rangle$

Couverture minimale :

1.  $\text{id\_film} \rightarrow \text{titre}$
2.  $\text{id\_film} \rightarrow \text{date\_début}$
3.  $\text{id\_film} \rightarrow \text{date\_fin}$
4.  $\text{id\_film} \rightarrow \text{version}$
5.  $\text{id\_film} \rightarrow \text{genre}$
6.  $\text{id\_film} \rightarrow \text{restriction}$
7.  $\text{id\_film} \rightarrow \text{durée}$
8.  $\text{no\_salle} \rightarrow \text{capacité}$
9.  $\text{id\_achat} \rightarrow \text{date\_séance}$
10.  $\text{id\_achat} \rightarrow \text{no\_salle},$
11.  $\text{id\_achat} \rightarrow \text{type\_tarif}$
12.  $\text{id\_achat} \rightarrow \text{date\_achat}$
13.  $\text{type\_tarif} \rightarrow \text{lib\_tarif}, \text{prix}$
14.  $\text{date\_séance}, \text{no\_salle} \rightarrow \text{id\_film}$

À ce stade, on a  $PC \rightarrow NC$  or c'est interdit par la 2FN

### Partition :

$R_1(\text{id\_film}, \text{titre}, \text{date\_début}, \text{date\_fin}, \text{durée}, \text{version}, \text{genre}, \text{restriction})$  DF{1,2,3,4,5,6,7}

$R_2(\text{no\_salle}, \text{capacité})$  DF{8}

$R_3(\text{date\_séance}, \text{no\_salle}, \text{id\_film})$  DF{14}

$R_4(\text{id\_achat}, \text{date\_séance}, \text{no\_salle}, \text{type\_tarif})$  DF{9,10,11,12}

$R_5(\text{type\_tarif}, \text{lib\_tarif}, \text{prix})$  DF{13}

Puis, on ajoute une relation qui comprend toutes nos clés :

$R_6(\text{id\_film}, \text{no\_salle}, \text{id\_achat}, \text{type\_tarif}, \text{date\_séance})$  DF{}

À l'issue de cet algorithme, on obtient des schémas en 3<sup>ieme</sup> forme normale. En effet, on retrouve dans certains schémas  $PC \rightarrow PC$ .

De plus, cet algorithme nous offre une solution spi puisque chacun des schémas ont des attributs en commun entre eux, ainsi que spdf car nous avons conservé toutes les dépendances fonctionnelles.

## 2.2 Algorithme de décomposition

Nous avons ensuite appliqué l'algorithme de décomposition sur notre ensemble de dépendances fonctionnelles et nous obtenons l'arbre de décomposition inclus dans l'annexe, figure 3.

On remarque que cette décomposition est la même que celle obtenue avec l'algorithme de Bernstein, et comme nous l'avons vu précédemment, cette décomposition est en 3<sup>ieme</sup> forme normale.



## Conclusion

Au cours de cette première partie, nous avons élaboré notre projet de base de données et défini le schéma correspondant. Après avoir normalisé notre base de données et effectué des tests avec [via le site fourni dans le sujet du projet](#) (Database Design Tool), nous sommes désormais en mesure de développer notre projet avec Oracle.

## Annexe

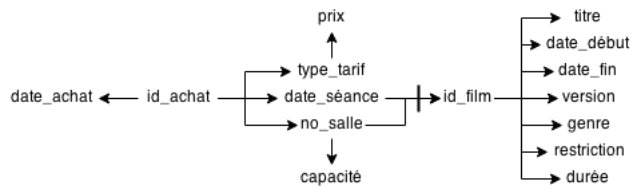


FIGURE 1 – Schéma des dépendances fonctionnelles

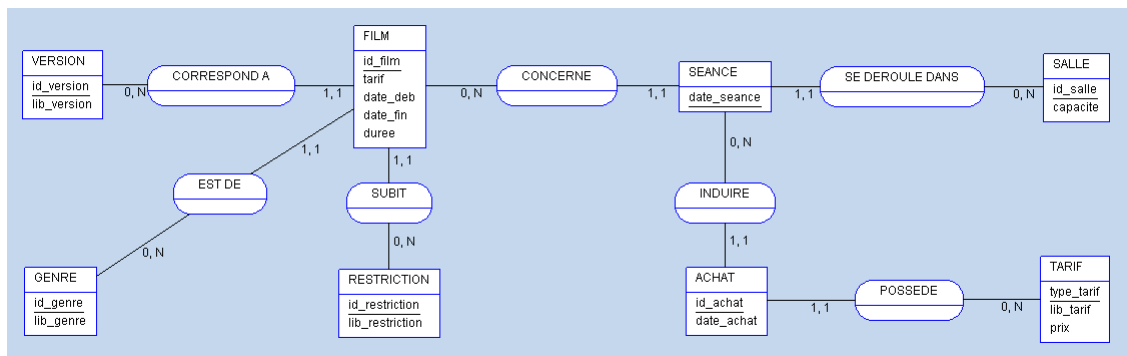


FIGURE 2 – MCD de la base de données

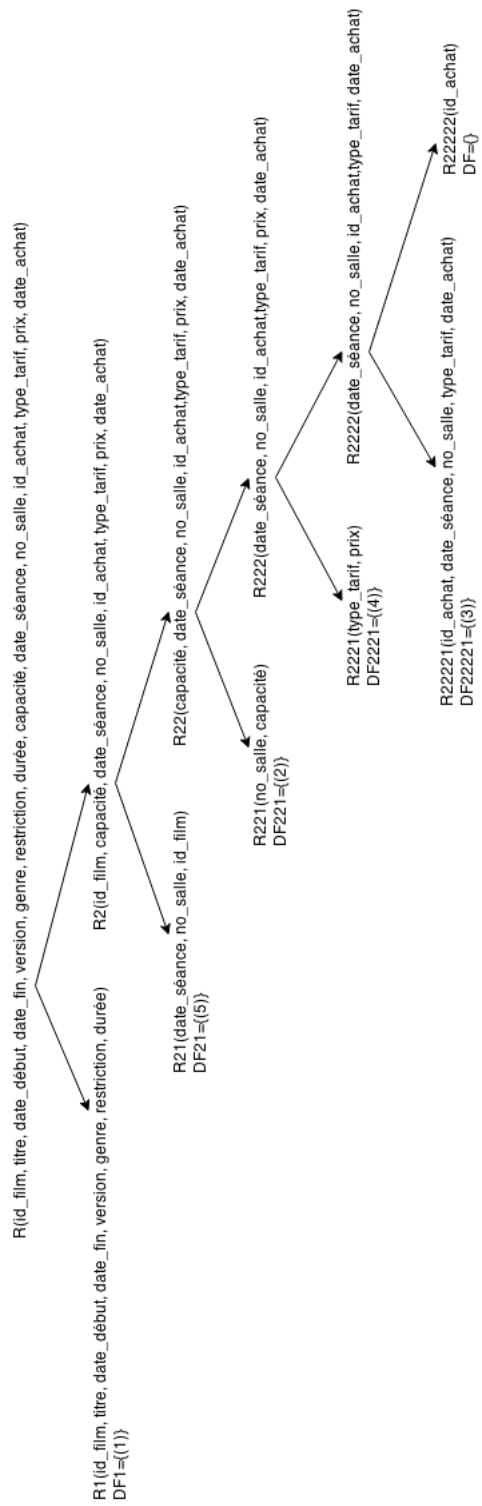


FIGURE 3 – Algorithme de décomposition