

MAHIER Loïc
JEHANNO Clément
JAMET Félix
PHALAVANDISHVILI Demetre

groupe 601B

Rapport préliminaire de projet *

*rapport réalisé sous L^AT_EX

Sommaire

1	Introduction	3
2	Répartition des tâches	3
3	Table de base	4
4	Dépendances fonctionnelles	4
5	Algorithme de Bernstein	5
5.1	Calcul de CV(DF)	5
5.1.1	Pas 1	5
5.1.2	Pas 2	6
5.1.3	Pas 3	8
5.2	Partitionnement de la CV et construction des schémas	10
6	Algorithme de décomposition	12
7	Schéma de nos tables	14
8	Conclusion préliminaire	15
9	Annexe	16

Introduction

Dans le cadre de ce projet nous devons créer une base de données. Nous avons décidé de modéliser la gestion de cinémas sur une grande échelle. Par exemple nous voulons savoir quels sont les cinémas de France, à qui ils appartiennent (Pathé,UGC, etc.) et ce qu'ils proposent. Comme notre modèle se base sur une certaine réalité, voici comment nous avons décomposé la chose, prenons l'exemple d'un cinéma :

Le cinéma Pathé à Atlantis, dans la ville de Nantes. Tout d'abord on voit que un cinéma est identifié par une adresse et une ville. Ensuite, notre cinéma possède des salles dans lesquelles seront diffusés des films. Chaque film est composé d'une équipe d'acteurs, d'un réalisateur et d'une date de sortie. Il peut être compatible ou non avec la 3D.

En ce qui concerne nos salles, elles possèdent un certain nombre de places qui sont réparties entre les places "standards", les places "handicapés" ainsi que les nouveaux sièges dBox (sièges bougeant en même temps que le film). Si elles sont compatibles, elles ont la possibilité de diffuser en 3D.

Une séance dans un cinéma donné correspond à une date de projection d'un certain film dans une salle spécifique à un horaire précis, le film étant diffusé ou non en 3D. Aujourd'hui si on va au cinéma il est possible de réserver sa séance, autrement dit, on réserve un certain nombre de places pour un film, à un horaire précis, dans un cinéma donné.

Répartition des tâches

Voici comment nous nous sommes organisés pour répartir les tâches :

Tout d'abord après les premières semaines de cours nous nous sommes réunis pour décider ensemble d'un sujet. L'idée du cinéma est venue assez naturellement et nous paraissait être à la fois concrète et proche de la réalité.

Ensuite nous avons défini tous les attributs de notre table, en se demandant ensemble : Que voulons-nous faire ? Comment voulons nous le faire ? Un cinéma fonctionne t-il vraiment comme ça ? L'ajout de tel ou tel attribut est-il pertinent ? etc etc. Une fois nos attributs répartis nous avons chacun pris un cinéma (on en a 4) et chaque personne a rempli la partie du tableau qui correspondait à un cinéma. Après cela on a observé nos tuples sur tableur et on a relevé nos dépendances fonctionnelles.

Ensuite Demetre et Félix ont fait l'algorithme de décomposition et Loïc et Clément ont fait l'algorithme de Bernstein. On a mis en commun le résultat des deux algorithmes afin de voir si on avait la même chose ou non, et pourquoi. Pour finir, nous avons testé la normalisation de notre schéma avec l'outil mis à notre disposition en question 5.

Table de base

Vous trouverez en annexe la table (4) contenant tous nos attributs ainsi que tous nos tuples. Celle-ci est en quatre parties à cause de sa taille conséquente.

Dépendances fonctionnelles

A partir de la table ci-dessous contenant tous nos attributs, nous avons déduit les douze dépendances fonctionnelles ci-dessous. Pour ce faire nous avons ajouté des attributs (id) nous permettant de simplifier nos relations.

- (1) idCine \rightarrow adresse, ville
- (2) adresse, ville \rightarrow franchise, nbSalle
- (3) idCine \rightarrow franchise, nbSalles
- (4) idCine, numSalle \rightarrow salleCompatibleEn3D, nbPlaceStandard, nbPlaceHandicape, nbDbox
- (5) idFilm \rightarrow nomFilm, dateSortie
- (6) nomFilm, dateSortie \rightarrow public, idReal, duree, compatible3D
- (7) idFilm, role \rightarrow idAct
- (8) idReal \rightarrow nomR, prenomR
- (9) idAct \rightarrow nomA, prenomA
- (10) idClient \rightarrow nomC, prenomC
- (11) idClient, numReservation \rightarrow nbPlaceStandardRes, nbPlaceHandicapeRes, nbPlaceDBoxRes, idSeance
- (12) idSeance, idCine \rightarrow horaire, dateProjection, numSalle, idFilm, diffusionEn3D

Avec les dépendances fonctionnelles ci-dessus, nous obtenons le graphe des dépendances en annexe (8). De cela nous déterminons la clé suivante : {idCine, idClient, numReservation, role}.

Algorithme de Bernstein

L'algo de Bernstein se fait en 4 parties :

- Calculer la CV(DF) et les clés. Si R est en 3FN on s'arrête.
- Partitionner CV(DF) en groupe DF_i ($1 \leq i \leq k$) tel que toutes les dfs d'un même groupe aient la même partie gauche.
- Construire un schéma $\langle R_i(U_i), DF_i \rangle$ pour chaque groupe DF_i, où U_i est l'ensemble des attributs apparaissant dans DF_i.
- Si aucun des schémas définis ne contient de clé X de R, rajouter un schéma $\langle R_{k+1}(X), \{\} \rangle$.

Calcul de CV(DF)

La couverture minimale se fait en trois parties :

- Toutes les dépendances doivent être élémentaires; les décomposer si nécessaire.
- Eliminer les attributs superflus du côté gauche de la df.
- Eliminer les dfs redondantes.

Pas 1

On décompose chacune des dfs en dfe :

- (1) idCine \rightarrow ville
- (1) idCine \rightarrow adresse
- (2) adresse, ville \rightarrow franchise
- (2) adresse, ville \rightarrow nbSalle
- (3) idCine \rightarrow franchise
- (3) idCine \rightarrow nbSalles
- (4) idCine, numSalle \rightarrow salleCompatibleEn3D

- (4) idCine, numSalle \rightarrow nbPlaceStandard
- (4) idCine, numSalle \rightarrow nbPlaceHandicape
- (4) idCine, numSalle \rightarrow nbDbox
- (5) idFilm \rightarrow nomFilm
- (5) idFilm \rightarrow dateSortie
- (6) nomFilm, dateSortie \rightarrow public
- (6) nomFilm, dateSortie \rightarrow idReal
- (6) nomFilm, dateSortie \rightarrow duree
- (6) nomFilm, dateSortie \rightarrow compatible3D
- (7) idFilm, role \rightarrow idAct
- (8) idReal \rightarrow nomR
- (8) idReal \rightarrow prenomR
- (9) idAct \rightarrow nomA
- (9) idAct \rightarrow prenomA
- (10) idClient \rightarrow nomC
- (10) idClient \rightarrow prenomC
- (11) idClient, numReservation \rightarrow idSeance
- (11) idClient, numReservation \rightarrow nbPlaceStandardRes
- (11) idClient, numReservation \rightarrow nbPlaceHandicapeRes
- (11) idClient, numReservation \rightarrow nbPlaceDBoxRes
- (12) idSeance, idCine \rightarrow horaire
- (12) idSeance, idCine \rightarrow dateProjection
- (12) idSeance, idCine \rightarrow numSalle
- (12) idSeance, idCine \rightarrow idFilm
- (12) idSeance idCine \rightarrow diffusionEn3D

Pas 2

On prend toutes les dfs qui ont plus d'un attribut à gauche et on calcul leur fermeture. On élimine l'autre attribut si l'attribut de droite de la df apparaît dans le résultat, ou s'il apparaît dans le résultat de la fermeture.

- (2) adresse, ville \rightarrow franchise, nbSalle

adresse+
 adresse
ville+
 ville

→ Les deux attributs sont nécessaires.

- (4) idCine, numSalle → salleCompatibleEn3D, nbPlaceStandard, nbPlaceHandicape, nbDbox

idCine +
idCine / adresse / ville / franchise / nbSalle
numSalle +
numSalle

→ Les deux attributs sont nécessaires.

- (6) nomFilm, dateSortie → public, idReal, duree, compatible3D

nomFilm +
nomFilm
dateSortie +
dateSortie

→ Les deux attributs sont nécessaires.

- (7) idFilm, role → idAct

idFilm +
idFilm / nomFilm / dateSortie / public / idReal / duree / compatible3D /
nomA / prenomA
role +
role

→ Les deux attributs sont nécessaires.

- (11) idClient, numReservation → nbPlaceStandardRes, nbPlaceHandicapeRes, nbPlaceDBoxRes, idSeance

idClient +
idClient / nomC / prenomC
numReservation +
numReservation

→ Les deux attributs sont nécessaires.

- (12) idSeance, idCine → horaire, dateProjection, numSalle, idFilm, diffusionEn3D

idSeance +

idSeance

idCine+ adresse / ville / franchise / nbSalle

→ Les deux attributs sont nécessaires.

Pas 3

Éliminons tout d'abord les dfs qui sont préservées par transitivité :

- (1) idCine \rightarrow adresse, ville
- (2) adresse, ville \rightarrow franchise, nbSalle
- (3) idCine \rightarrow franchise, nbSalles

Si l'on prend les dfs 1, 2 et 3, on remarque que l'on peut supprimer la 3 car on peut retrouver celle-ci par transitivité. Reprenons donc nos dfs restantes :

- (1) idCine \rightarrow adresse, ville
- (2) adresse, ville \rightarrow franchise, nbSalle
- (3) idCine, numSalle \rightarrow salleCompatibleEn3D, nbPlaceStandard, nbPlaceHandicape, nbDbox
- (4) idFilm \rightarrow nomFilm, dateSortie
- (5) nomFilm, dateSortie \rightarrow public, idReal, duree, compatible3D
- (6) idFilm, role \rightarrow idAct
- (7) idReal \rightarrow nomR, prenomR
- (8) idAct \rightarrow nomA, prenomA
- (9) idClient \rightarrow nomC, prenomC
- (10) idClient, numReservation \rightarrow nbPlaceStandardRes, nbPlaceHandicapeRes, nbPlaceDBoxRes, idSeance
- (11) idSeance, idCine \rightarrow horaire, dateProjection, numSalle, idFilm, diffusionEn3D

A présent, analysons chaque dépendance fonctionnelle une par une :

- (1) idCine \rightarrow adresse, ville

idCine+

idCine

→ La df est préservée.

- (2) adresse, ville \rightarrow franchise, nbSalle

$\frac{\text{adresse, ville}+}{\text{adresse, ville}}$
 \rightarrow La df est préservée.

- (3) idCine, numSalle \rightarrow salleCompatibleEn3D, nbPlaceStandard, nbPlaceHandicape, nbDbox

$\frac{\text{idCine, numSalle}+}{\text{idCine} / \text{adresse} / \text{ville} / \text{franchise} / \text{nbSalle} / \text{numSalle}}$
 \rightarrow La df est préservée.

- (4) idFilm \rightarrow nomFilm, dateSortie

$\frac{\text{idFilm}+}{\text{idFilm}}$
 \rightarrow La df est préservée.

- (5) nomFilm, dateSortie \rightarrow public, idReal, duree, compatible3D

$\frac{\text{nomFilm, dateSortie}+}{\text{nomFilm} / \text{dateSortie}}$
 \rightarrow La df est préservée.

- (6) idFilm, role \rightarrow idAct

$\frac{\text{idFilm, role}+}{\text{idFilm} / \text{nomFilm} / \text{dateSortie} / \text{public} / \text{idReal} / \text{duree} / \text{compatible3D} / \text{nomR} / \text{prenomR} / \text{role}}$
 \rightarrow La df est préservée.

- (7) idReal \rightarrow nomP, prenomP

$\frac{\text{idReal}+}{\text{idReal}}$
 \rightarrow La df est préservée.

- (8) idAct \rightarrow nomP, prenomP

idAct+

idAct

→ La df est préservée.

- (9) idClient → nomC, prenomC

idClient+

idClient

→ La df est préservée.

- (10) idClient, numReservation → nbPlaceStandardRes, nbPlaceHandicapeRes, nbPlaceDBoxRes, idSeance

idClient, numReservation+

idClient / nomC / prenomC / numReservation

→ La df est préservée.

- (11) idSeance, idCine → horaire, dateProjection, numSalle, idFilm, diffusionEn3D

idSeance, idCine+

idSeance / idCine / adresse / ville / franchise / nbSalle

→ La df est préservée.

Ainsi, hormis la suppression d'une dépendance fonctionnelle transitive, nos dépendances fonctionnelles ne changent pas.

On constate que l'on est bien en 1FN, ainsi qu'en 2FN. Cependant nous ne sommes pas en 3FN. En effet, nous avons des attributs non clés, qui déterminent d'autres attributs non clés. Par exemple, adresse et ville sont deux attributs non clés qui déterminent franchise et nbSalle qui sont eux aussi non clés (df(2)).

Partitionnement de la CV et construction des schémas

R1 = {idCine, adresse, ville}

DF1 = {idCine → adresse, ville}

$R2 = \{\text{adresse}, \text{ville}, \text{franchise}, \text{nbSalle}\}$
 $DF2 = \{\text{adresse}, \text{ville} \rightarrow \text{franchise}, \text{nbSalle}\}$

$R3 = \{\text{idCine}, \text{numSalle}, \text{salleCompatibleEn3D}, \text{nbPlaceStandard}, \text{nbPlaceHandicape}, \text{nbDbox}\}$
 $DF3 = \{\text{idCine}, \text{numSalle}, \rightarrow \text{salleCompatibleEn3D}, \text{nbPlaceStandard}, \text{nbPlaceHandicape}, \text{nbDbox}\}$

$R4 = \{\text{idFilm}, \text{nomFilm}, \text{dateSortie}\}$
 $DF4 = \{\text{idFilm} \rightarrow \text{nomFilm}, \text{dateSortie}\}$

$R5 = \{\text{nomFilm}, \text{dateSortie}, \text{public}, \text{idReal}, \text{duree}, \text{compatible3D}\}$
 $DF5 = \{\text{nomFilm}, \text{dateSortie} \rightarrow \text{public}, \text{idReal}, \text{duree}, \text{compatible3D}\}$

$R6 = \{\text{idFilm}, \text{role}, \text{idAct}\}$
 $DF6 = \{\text{idFilm}, \text{role} \rightarrow \text{idAct}\}$

$R7 = \{\text{idReal}, \text{nomR}, \text{prenomR}\}$
 $DF7 = \{\text{idReal} \rightarrow \text{nomR}, \text{prenomR}\}$

$R8 = \{\text{idAct}, \text{nomA}, \text{prenomA}\}$
 $DF8 = \{\text{idAct} \rightarrow \text{nomA}, \text{prenomA}\}$

$R9 = \{\text{idClient}, \text{nomC}, \text{prenomC}\}$
 $DF9 = \{\text{idClient} \rightarrow \text{nomC}, \text{prenomC}\}$

$R10 = \{\text{idClient}, \text{numReservation}, \text{nbPlaceStandardRes}, \text{nbPlaceHandicapeRes}, \text{nbPlaceDBoxRes}, \text{idSeance}\}$
 $DF10 = \{\text{idClient}, \text{numReservation} \rightarrow \text{nbPlaceStandardRes}, \text{nbPlaceHandicapeRes}, \text{nbPlaceDBoxRes}, \text{idSeance}\}$

$R11 = \{\underline{\text{idSeance}}, \underline{\text{idCine}}, \text{horaire}, \text{dateProjection}, \text{numSalle}, \text{idFilm}, \text{diffusionEn3D}\}$

$DF11 = \{\text{idSeance}, \text{idCine} \rightarrow \text{horaire}, \text{dateProjection}, \text{numSalle}, \text{idFilm}, \text{diffusionEn3D}\}$

On constate que nous n'avons pas de relation contenant toutes nos clés, c'est pourquoi nous devons créer une relation pour cela, avec une dépendance fonctionnelle associée vide.

$R12 = \{\underline{\text{idCine}}, \underline{\text{idClient}}, \underline{\text{numReservation}}, \underline{\text{role}}\}$

$DF12 = \{\}$

Algorithme de décomposition

Nous avons pris tous nos attributs puis nous avons suivi l'algorithme de décomposition. C'est à dire que en fonction de nos attributs et de nos dépendances fonctionnelles, nous avons pris une dépendance et avons fait une relation en fonction de cette dépendance. Puis nous avons retiré nos attributs non clés du reste de la relation originale. Ensuite on a recommencé jusqu'à arriver à une partie ne contenant que des clés et plus de dépendance fonctionnelle.

On a commencé par les dépendances fonctionnelles qui n'avaient qu'un attribut à gauche puis nous avons fini par celles qui avaient plusieurs attributs à gauche.

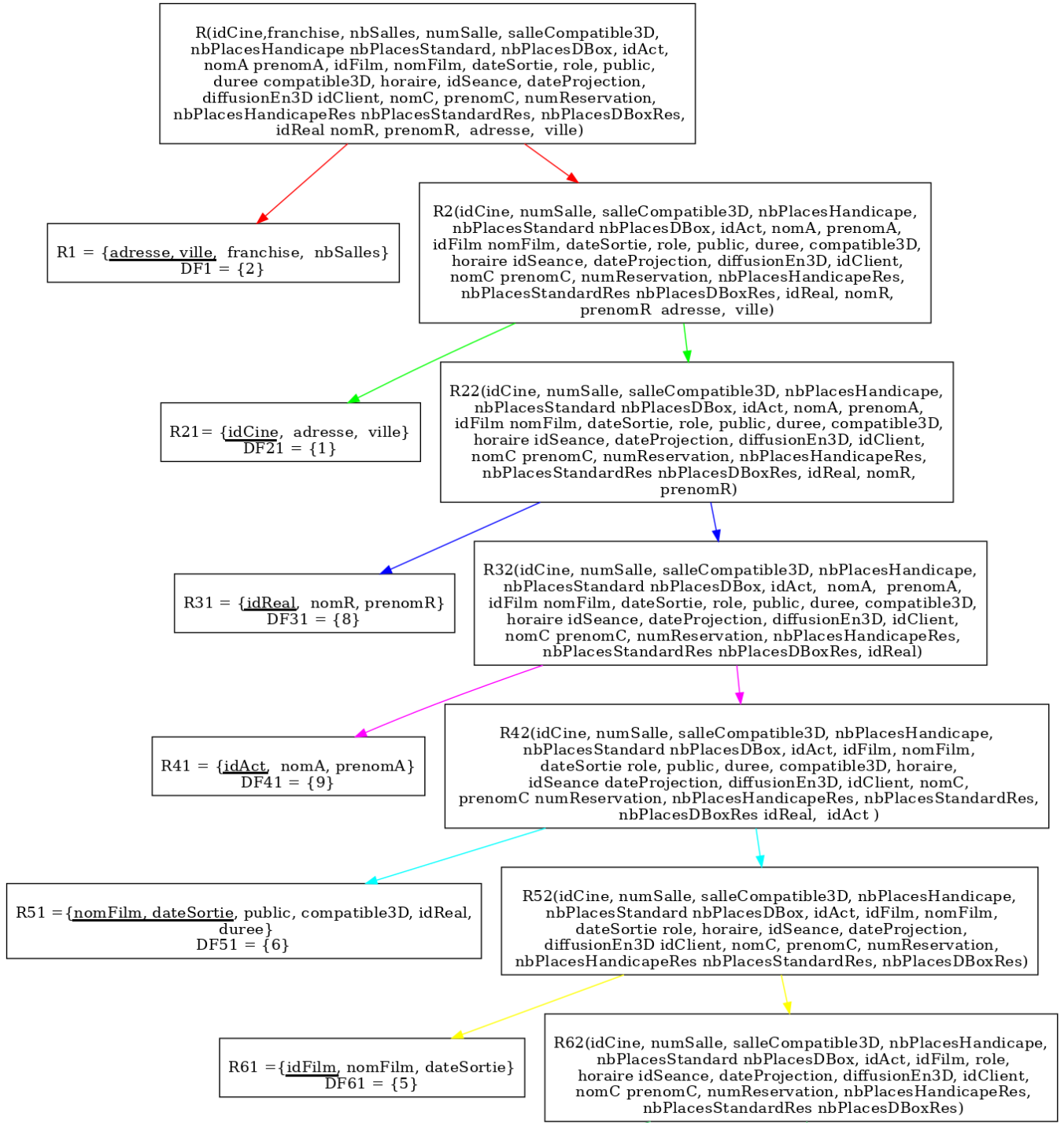


FIGURE 1 – Première partie de notre algorithme de décomposition

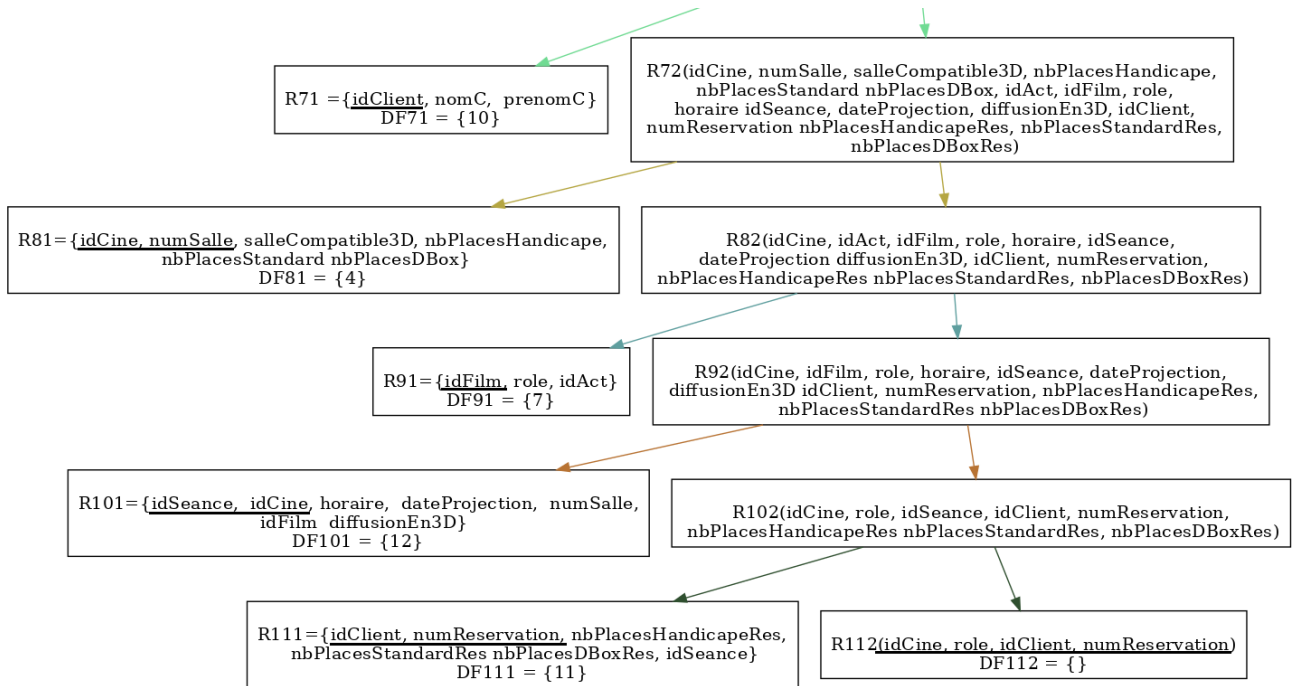


FIGURE 2 – Deuxième partie de notre algorithme de décomposition

Schéma de nos tables

Pour le diagramme nous nous sommes basés sur le résultat de l'algorithme de décomposition qui nous a fourni des tables équivalentes aux 11 relations R1..R11. Cependant nous avons fusionné deux fois deux tables : la table R1 et R21 car cela nous paraît plus logique d'avoir pour chaque adresse, ville, franchise et nbSalles un seul idCine qui détermine un unique cinéma.

Cette table devient donc la table Cinema, ce qui donne plus de sens à notre modèle. Nous avons aussi fusionné R51 et R61 pour en faire une seule et même table Film pour les mêmes raisons.

Pour finir on a renommé toutes les autres tables pour leur donner un nom plus parlant : R31 est devenue Realisateur, R41 est devenue Acteur, R71 est devenue Client, R81 est devenue Salle, R91 est devenue Casting, R101 est devenue Seance et 111 est devenue Reservation.

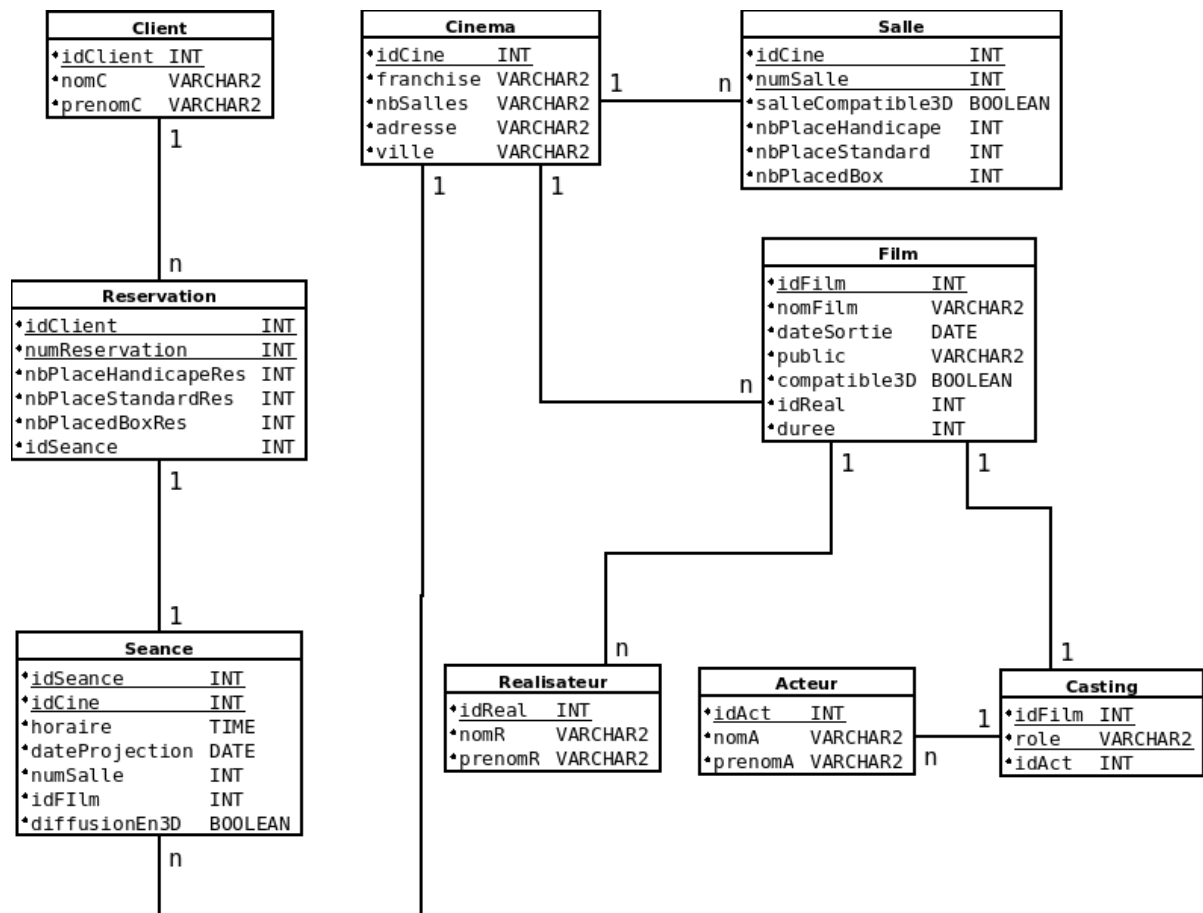


FIGURE 3 – Diagramme UML de nos tables

Conclusion préliminaire

Cette première partie nous a permis de bien poser les bases de notre base de donnée. De plus, l'algorithme de décomposition nous permet d'obtenir des tables qui nous semblent cohérentes avec la réalité. Pour la suite, nous avons déjà quelques idées de contraintes, de triggers et de fonctions qu'il nous faudra intégrer pour bien gérer nos cinémas.

Annexe

FIGURE 4 – Première partie de notre table contenant tous les attributs et quelques tuples

nomR	prenomR	idAct	nomA	prenomA	rôle	idSeance	dateProjection	horaire	diffusionEn3d	idClient
Caruso	Daniel John	1	diesel	vin	xander cage	1	24/01/17	20h	oui	1
Caruso	Daniel John	2	yan	donnie	xiang	1	24/01/17	20h	oui	1
Caruso	Daniel John	3	padukone	deepika	serena unger	1	24/01/17	20h	oui	1
Caruso	Daniel John	4	wu	kris	nicks	1	24/01/17	20h	oui	1
Caruso	Daniel John	5	rose	ruby	adele wofl	1	24/01/17	20h	oui	1
Caruso	Daniel John	6	McCann	rory	temyson torch	1	24/01/17	20h	oui	1
Caruso	Daniel John	1	diesel	vin	xander cage	1	24/01/17	20h	oui	2
Caruso	Daniel John	2	yan	donnie	xiang	1	24/01/17	20h	oui	2
Caruso	Daniel John	3	padukone	deepika	serena unger	1	24/01/17	20h	oui	2
Caruso	Daniel John	4	wu	kris	nicks	1	24/01/17	20h	oui	2
Caruso	Daniel John	5	rose	ruby	adele wofl	1	24/01/17	20h	oui	2
Caruso	Daniel John	6	McCann	rory	temyson torch	1	24/01/17	20h	oui	2
Caruso	Daniel John	1	diesel	vin	xander cage	2	27/01/17	16h30	Non	3
Caruso	Daniel John	2	yan	donnie	xiang	2	27/01/17	16h30	Non	3
Caruso	Daniel John	3	padukone	deepika	serena unger	2	27/01/17	16h30	Non	3
Caruso	Daniel John	4	wu	kris	nicks	2	27/01/17	16h30	Non	3
Caruso	Daniel John	5	rose	ruby	adele wofl	2	27/01/17	16h30	Non	3
Caruso	Daniel John	6	McCann	rory	temyson torch	2	27/01/17	16h30	Non	3
Kurzel	Justin	7	fassbender	michael	cal lynch	3	10/01/17	11h00	Non	4
Kurzel	Justin	8	fassbender	michael	aguiar da nerha	3	10/01/17	11h00	Non	4
Kurzel	Justin	9	cotillard	marion	sofia	3	10/01/17	11h00	Non	4
Kurzel	Justin	10	irons	jeremy	rikkin	3	10/01/17	11h00	Non	4
Kurzel	Justin	11	gleeson	brendan	joseph lynch	3	10/01/17	11h00	Non	4
Kurzel	Justin	12	labeled	charlotte	Ellen Kaye	3	10/01/17	11h00	Non	4
Chazelle	Damien	13	gosling	Anane	Maria	4	10/01/17	11h00	Non	4
Chazelle	Damien	14	stone	ryan	Sebastian	4	10/01/17	22h	Non	5
Chazelle	Damien	15	legend	emma	Mia	4	10/01/17	22h	Non	5
Chazelle	Damien	16	DeWitt	john	Keith	4	10/01/17	22h	Non	5
Chazelle	Damien	17	Wittrock	Rosemarie	Laura	4	10/01/17	22h	Non	5
Chazelle	Damien	18	Simmons	Finn	Greg	4	10/01/17	22h	Non	5
Chazelle	Damien	18	Simmons	J.K	Bill	4	10/01/17	22h	Non	5

FIGURE 6 – Troisième partie de notre table contenant tous les attributs et quelques tuples

nomC	prenomC	numRes	nbPlaceStandardRes	nbPlaceHandicapeRes	nbDboxRes
Martin	herve	4	2	0	0
Martin	herve	4	2	0	0
Martin	herve	4	2	0	0
Martin	herve	4	2	0	0
Martin	herve	4	2	0	0
Martin	herve	4	2	0	0
Dupont	Marc	8	2	1	0
Dupont	Marc	8	2	1	0
Dupont	Marc	8	2	1	0
Dupont	Marc	8	2	1	0
Dupont	Marc	8	2	1	0
Dubois	Jeremie	15	2	0	2
Dubois	Jeremie	15	2	0	2
Dubois	Jeremie	15	2	0	2
Dubois	Jeremie	15	2	0	2
Dubois	Jeremie	15	2	0	2
Morgan	philippe	16	1	0	2
Morgan	philippe	16	1	0	2
Morgan	philippe	16	1	0	2
Morgan	philippe	16	1	0	2
Morgan	philippe	16	1	0	2
Morgan	philippe	16	1	0	2
David	jean	23	0	0	2
David	jean	23	0	0	2
David	jean	23	0	0	2
David	jean	23	0	0	2
David	jean	23	0	0	2

FIGURE 7 – Quatrième partie de notre table contenant tous les attributs et quelques tuples

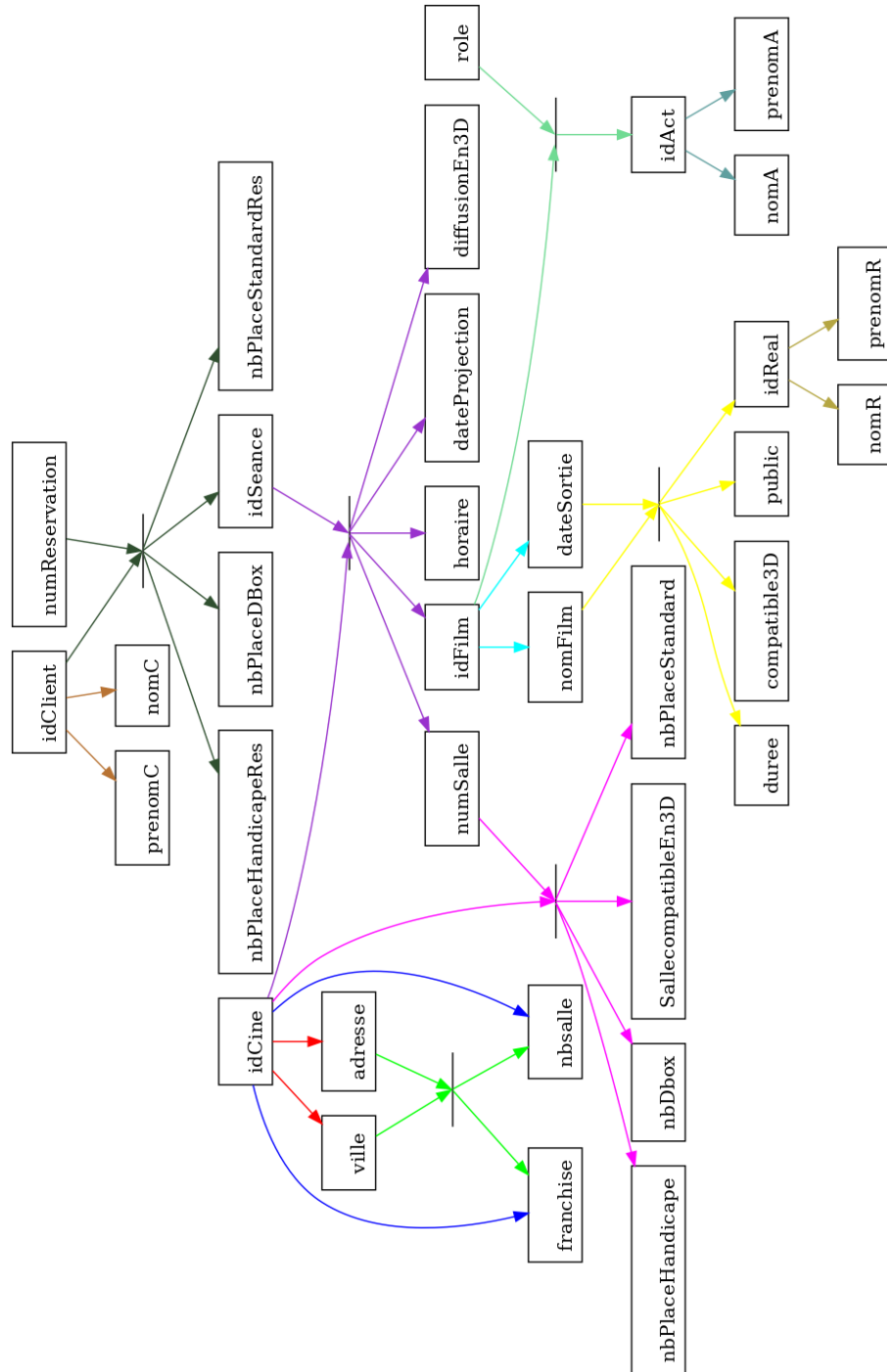


FIGURE 8 – Graphe des dépendances