

INFO00502

Beatrice Beaujet – Thibaut Bernard

Gestion de la programmation d'un complexe de cinéma



BAUDOUIN Philippe

RENARD Stanislas

26/11/2014

TABLE DES MATIERES

A.	Analyse du problème.....	2
1.	Introduction	2
2.	Cahier des charges	2
3.	Matrice de flux	3
4.	Diagramme de flux.....	3
5.	Modèle conceptuel de traitement.....	4
6.	Dictionnaire de données	5
7.	Modèle conceptuel de données	8
8.	Modèle logique de données	9
9.	Modèle relationnel	10
10.	Normalisation	10
B.	Traitement SQL.....	11
1.	Script de création de base de données.....	11
2.	Insertion de données	11
3.	Requêtes SQL	12
C.	Conclusion	13
	Annexe	14

A. ANALYSE DU PROBLEME

1. INTRODUCTION

Dans le cadre de la matière info0502 nous avons un projet à réaliser. Celui-ci consiste à modéliser la gestion d'un complexe de cinéma.

Pour ce faire nous possédons des informations relatives sur ce complexe et nous devons à partir ses informations créer une base de données implémentable dans un SGBD (Système de Gestion de Base de Données) qui est MySQL pour notre cas.

Nous allons dans un premier temps passer par une phase d'analyse qui nous permettra de nous rapprocher du problème au plus près de la réalité, ensuite nous créerons notre base de données d'après cette analyse.

2. CAHIER DES CHARGES

Le cahier des charges étant libre dans ses choix, hormis le fait que nous devons gérer un complexe de cinéma.

Pour une meilleure analyse et évité de se perdre dans les choix futur concernant cette dernière, nous avons décidé d'imposer quelques règles supplémentaires :

- On se concentre uniquement sur les interactions avec les clients
- La possibilité à un client de pouvoir acheter sa place sur internet ou sur place
- La gestion des recettes (consommables et places de cinéma)
- La gestion de plusieurs établissements
- La gestion de plusieurs films avec son nombre de place restante pour la séance
- Ajout de la gestion des publicité pour une séance

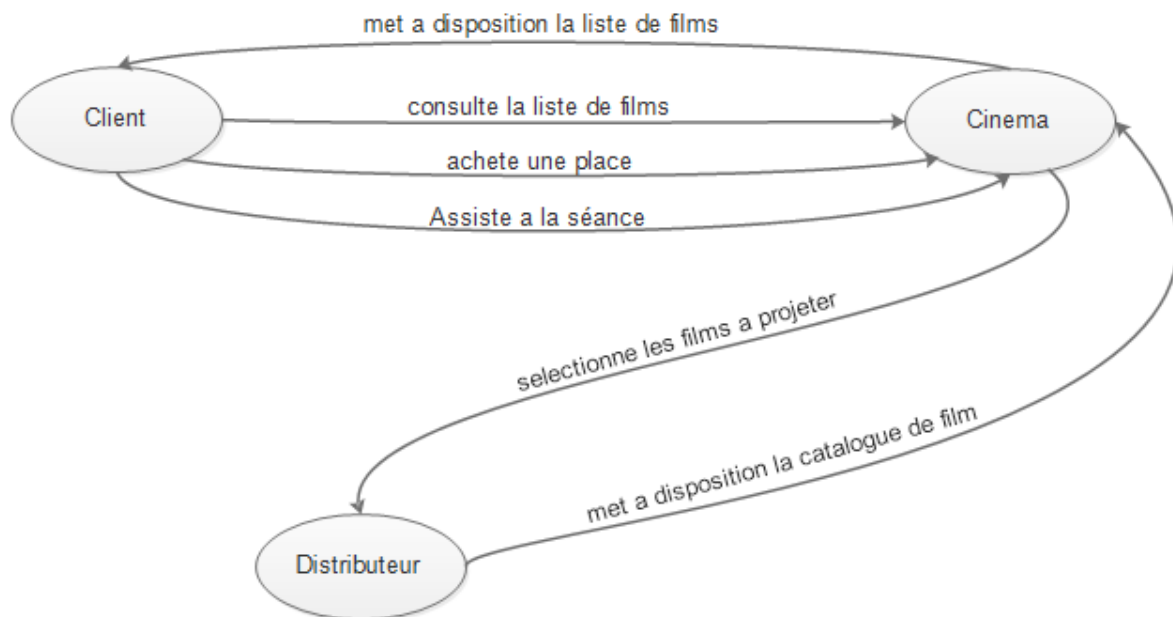
3. MATRICE DE FLUX

D'après les informations relatives au cahier des charges nous pouvons en déduire une matrice de flux qui va représenter les principales interactions entre nos acteurs.

	Client	Cinéma	Distributeur
Client		<ul style="list-style-type: none"> - consulte la liste de film - acheter une place - achète des consommables - assiste à la séance 	
Cinéma	- met à disposition la liste de film		- Sélectionne des films a projeté
Distributeur		- Met à disposition le catalogue de film	

4. DIAGRAMME DE FLUX

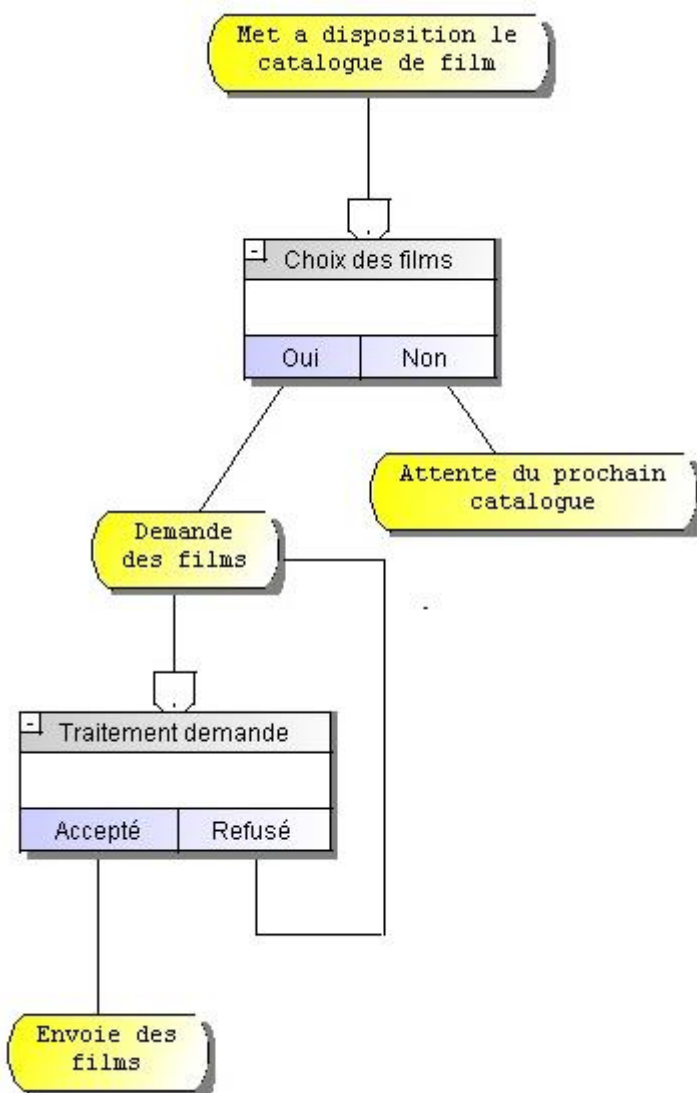
NB : l'acteur Distributeur ne sera présent dans la base de données.



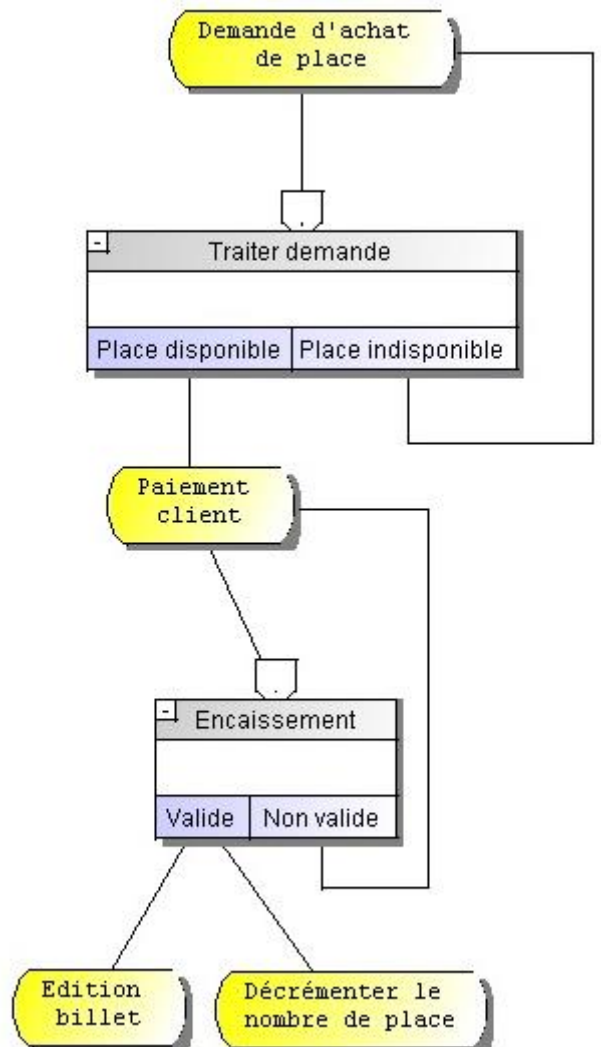
5. MODELE CONCEPTUEL DE TRAITEMENT

Le **modèle conceptuel des traitements** ou **MCT** permet de représenter de façon schématique l'activité d'un système d'information sans faire référence à des choix organisationnels ou des moyens d'exécution, c'est-à-dire qu'il permet de définir simplement ce qui doit être fait.

MCT Cinéma / Distributeur :



MCT Client / Cinéma :



6. DICTIONNAIRE DE DONNEES

Billet :

Colonne	Type	Null
<u>idBillet</u>	int	Non

Client :

Colonne	Type	Null
<u>idClient</u>	int	Non
nomClient	varchar(25)	Oui
prenomClient	varchar(25)	Oui
adresseMail	varchar(25)	Oui
mdp	varchar(25)	Oui
ddnClient	date	Oui

Consommable :

Colonne	Type	Null
<u>idConsommable</u>	int	Non
nomConso	varchar(25)	Oui
prixConso	float	Oui

Est vendu :

Colonne	Type	Null
quantite	int(11)	Non

Etablissement :

Colonne	Type	Null
<u>idEtablissement</u>	int	Non
adresse	varchar(50)	Oui
codePostal	int	Oui

Film :

Colonne	Type	Null
<u>idFilm</u>	int	Non
duree	int	Oui
pays	varchar(25)	Oui
synopsis	varchar(500)	Oui
annee	int	Oui

Joue :

Colonne	Type	Null
rôle	varchar(25)	Oui

Personne :

Colonne	Type	Null
<u>idPersonne</u>	int	Non
nomPersonne	varchar(25)	Oui
prenomPersonne	varchar(25)	Oui
sexe	tinyint	Oui
ddnPersonne	date	Oui

Publicite :

Colonne	Type	Null
<u>idPub</u>	int	Non
nomPub	varchar(25)	Oui
durationPub	int	Oui

Recette :

Colonne	Type	Null
<u>idRecette</u>	int	Non
dateRecette	date	Oui

Salle :

Colonne	Type	Null
<u>idSalle</u>	int	Non
nbPlace	int	Oui
equipement3d	tinyint	Oui

Seance :

Colonne	Type	Null
<u>idSeance</u>	int	Non
heure	date	Oui
placeRestantes	int	Oui
projection3d	tinyint	Oui

Tarif :

Colonne	Type	Null
<u>idTarif</u>	int	Non
nomTarif	varchar	Oui
prix	float	Oui

Vente :

Colonne	Type
<u>idVente</u>	int

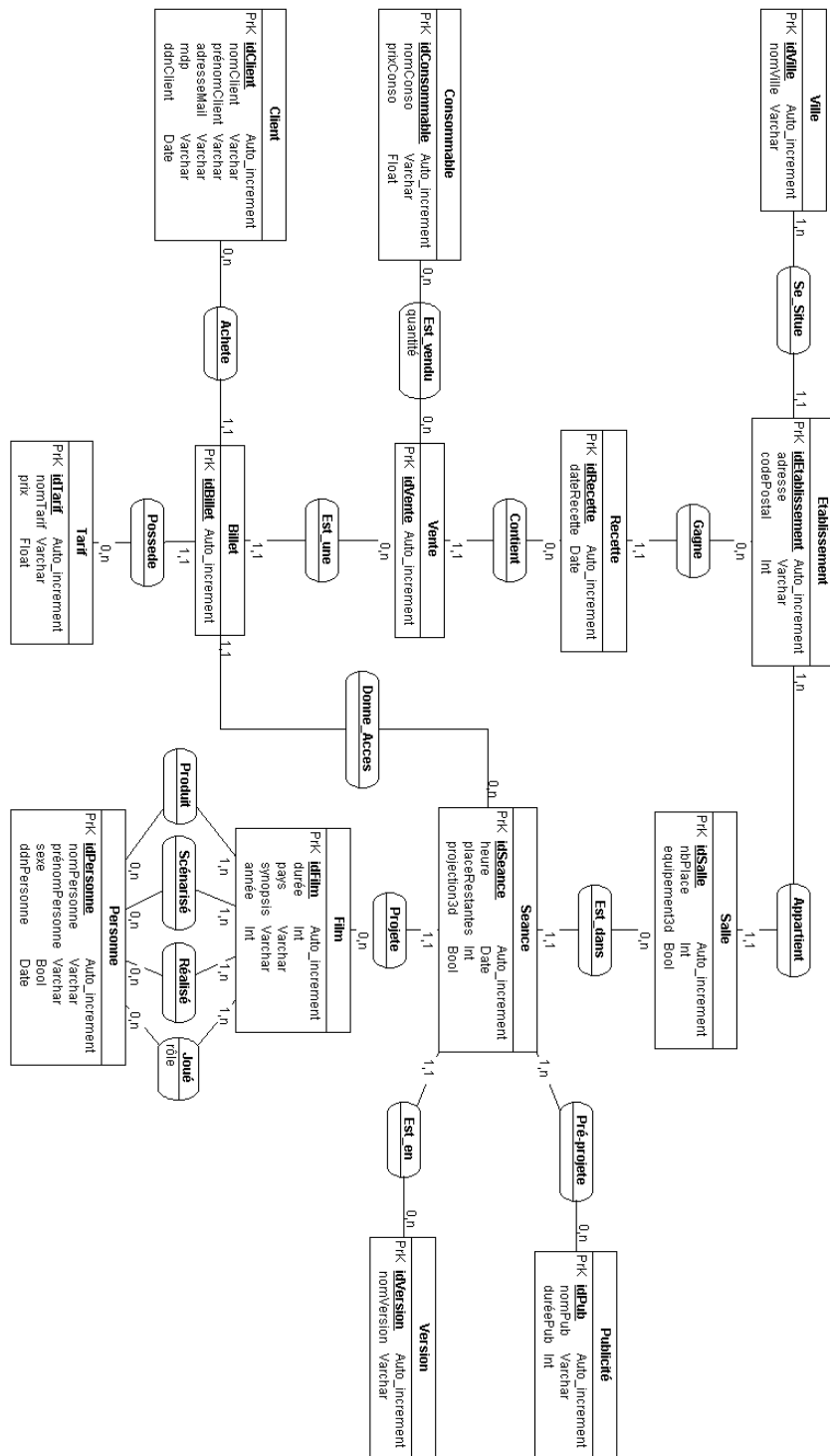
Version :

Colonne	Type	Null
<u>idVersion</u>	int	Non
nomVersion	varchar(25)	Oui

Ville :

Colonne	Type	Null
<u>idVille</u>	int	Non
nomVille	varchar(25)	Oui

7. MODELE CONCEPTUEL DE DONNEES

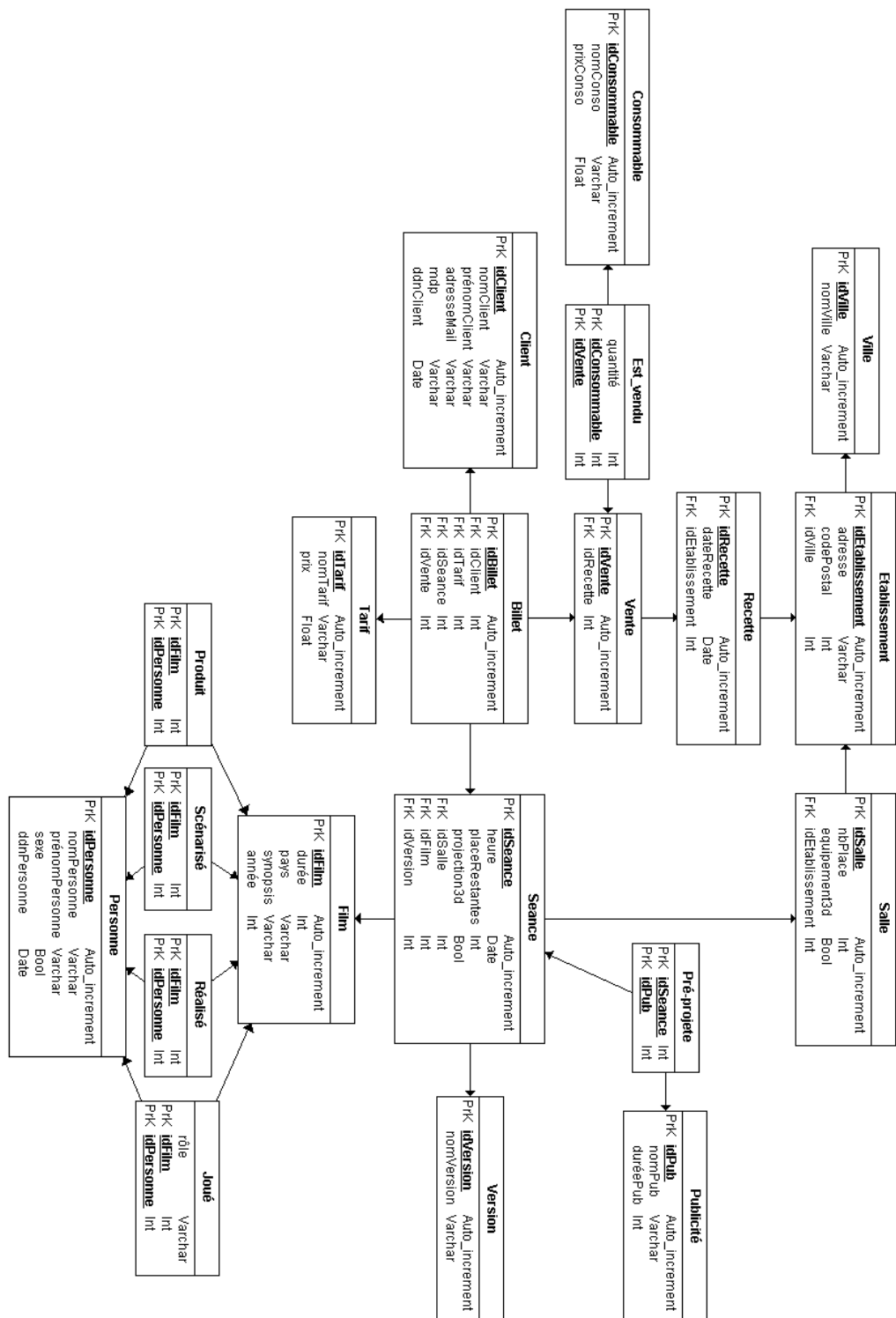


Le **MCD** (ou **Modèle Conceptuel des Données**) est utilisé pour décrire sous forme d'un schéma les données relatives au sujet (les entités, leurs attributs et les relations qu'elles entretiennent).

Les entités et les relations du **MCD** étant assez explicites nous n'avons pas besoin de les détailler ici.

On peut donc en déduire le modèle logique de données.

8. MODELE LOGIQUE DE DONNEES



Le **modèle Logique de Données** (ou **MLD**) est une image du modèle conceptuel intégrant les choix d'organisation des données, il ajoute au **MCD** la notion d'organisation. Le **MLD** indique donc comment les données sont organisées.

On remarque sur le schéma les clés primaires qui sont souligné et représenté par l'annotation PrK et les clés secondaires sont représentés par l'annotation FrK.

9. MODELE RELATIONNEL

Billet	(<u>idBillet</u> , #idClient, #idTarif, #idSeance, #idVente)
Client	(<u>idClient</u> , nomClient, prenomClient, adresseMail, mdp, ddnClient)
Consommable	(<u>idConsommable</u> , nomConso, prixConso)
Est_vendu	(quantité, #idConsommable, #idVente)
Établissement	(<u>idEtablissement</u> , adresse, codePostal, #idVille)
Film	(<u>idFilm</u> , nomFilm, durée, pays, synopsis, année)
Joué	(rôle, #idFilm, #idPersonne)
Personne	(<u>idPersonne</u> , nomPersonne, prenomPersonne, sexe, ddnPersonne)
Pré-Projeté	(#idSeance, #idPub)
Produit	(#idFilm, #idPersonne)
Publicité	(<u>idPub</u> , nomPub, duréePub)
Réalisé	(#idFilm, #idPersonne)
Recette	(<u>idRecette</u> , dateRecette, #idEtablissement)
Salle	(<u>idSalle</u> , nbPlace, equipement3d, #idEtablissement)
Scénarisé	(#idFilm, #idPersonne)
Seance	(<u>idSeance</u> , heure, placeRestantes, projection3d, #idSalle, #idFilm, #idVersion)
Tarif	(<u>idTarif</u> , nomTarif, prix)
Vente	(<u>idVente</u> , #idRecette)
Version	(<u>idVersion</u> , nomVersion)
Ville	(<u>idVille</u> , nomVille)

Le modèle relationnel est une manière de modéliser les informations contenues dans une base de données.

On retrouve comme dans le **MLD** les clés primaires qui sont soulignées et cette fois si les clés secondaires précédé d'un #.

10. NORMALISATION

Le but essentiel de la normalisation est d'éviter une mauvaise modélisation des données et ainsi éviter un certain nombre de problèmes potentiels tels que les anomalies de lecture, les anomalies d'écriture, la redondance des données et la contreperformance.

Nous rappelons au préalable les règles de normalisation que nous allons appliquer :

1NF : Une relation est en 1NF si et seulement si chacun des attributs qui la compose ne peut contenir qu'une valeur atomique

2NF : Une relation R est en 2NF si et seulement si R est 1NF et tout attribut n'appartenant pas à une clé candidate ne dépend pas d'une partie de cette clé

3NF : Une relation R est en 3NF si et seulement si R est en 2NF et tout attribut n'appartenant pas à une clé de dépend pas d'un attribut n'appartenant pas cette clé.

Notre base de données est bien en 1NF car chaque entité possède des attributs atomiques.

Elle aussi en 2NF car toutes les entités ne possède qu'une clé primaire.

Et enfin elle est en 3NF car tout attribut ne composant pas un identifiant dépend directement d'un identifiant

B. TRAITEMENT SQL

Nous arrivons maintenant à la seconde phase celle de la création de notre base de données.

1. SCRIPT DE CREATION DE BASE DE DONNEES

Prenons l'exemple de la table Client, le script de création de cette table donne la requête SQL suivante :

```
CREATE TABLE IF NOT EXISTS `Client` (  
    `idClient`      int(11)      NOT NULL AUTO_INCREMENT,  
    `nomClient`     varchar(25)   DEFAULT NULL,  
    `prenomClient`  varchar(25)   DEFAULT NULL,  
    `adresseMail`   varchar(25)   DEFAULT NULL,  
    `mdp`           varchar(25)   DEFAULT NULL,  
    `ddnClient`     date          DEFAULT NULL,  
    PRIMARY KEY (`idClient`)  
)
```

Cette requête nous permet donc de créer la table Client avec comme clé primaire idClient, comme attribut nomClient, prenomClient, adresseMail et mdp qui sont des varchar cette a dire des chaines de caractères et enfin le dernier attribut ddnClient qui est une date.

Les autres requêtes de création de table sont présentes en annexe.

2. INSERTION DE DONNEES

Gardons l'exemple de Client pour l'insertion de données, le script d'insertion dans cette table nous donne la requête SQL suivante :

```
INSERT INTO `Client` (`idClient`, `nomClient`, `prenomClient`, `adresseMail`, `mdp`, `ddnClient`) VALUES  
(1, 'nClient1', 'pClient1', 'adresse1', '3210', '2014-12-01'),  
(2, 'nClient2', 'pClient2', 'adresse2', '3210', '2014-12-01'),  
(3, 'nClient3', 'pClient3', 'adresse3', '3210', '2014-12-01'),  
(4, 'nClient3', 'pClient3', 'adresse4', '3210', '2014-12-01');
```

Dans cette requête on remarque que les insertions sont « factices » mais représentative de l'insertion de 4 clients dans la base de données.

Les autres requêtes d'insertion de données sont présentes en annexe.

3. REQUETES SQL

- **Liste des films passant à Reims :**

Requêtes SQL :

```
SELECT nomFilm
FROM Film, Seance, Salle, Etablissement, Ville
WHERE Film.idFilm = Seance.idFilm
AND Seance.idSalle = Salle.idSalle
AND Salle.idEtablissement = Etablissement.idEtablissement
AND Etablissement.idVille = Ville.idVille
AND nomVille LIKE 'Reims'
```

Algèbre relationnel :

$$\Pi_{nomFilm} (Film \bowtie \Pi_{idFilm} (Seance \bowtie \Pi_{idSalle} (Salle \bowtie \Pi_{idEtablissement} (Etablissement \bowtie \rho_{nomVille=Reims} (Ville))))))$$

- **Argent rapporté par une séance :**

Requêtes SQL :

```
SELECT SUM( prix )
FROM Tarif, Billet, Seance
WHERE Tarif.idTarif = Billet.idTarif
AND Billet.idSeance = Seance.idSeance
AND Seance.idSeance = 3
```

Algèbre relationnel :

$$\sum \Pi_{Prix} (Tarif \bowtie \Pi_{idTarif} (\rho_{idSeance=1} (Billet)))$$

- **Nombre de places restantes pour une séance :**

Requêtes SQL :

```
SELECT nbPlace - (SELECT COUNT(*) FROM Billet WHERE Billet.idSeance = 1)
FROM Salle, Seance
WHERE Salle.idSalle = Seance.idSeance
AND Seance.idSeance = 1
```

Algèbre relationnel :

$$\Pi_{nbPlace} (Salle \bowtie \rho_{idSeance=1} (Seance)) - COUNT(\rho_{idSeance=1} (Billet))$$

- **Recettes d'un établissement donné un jour donné :**

Requêtes SQL :

```
SELECT SUM( prix ) + SUM( prixConso )
FROM Recette, Vente, Billet, Tarif, Est_vendu, Consommable
WHERE dateRecette = '2014-12-12'
AND idEtablissement = 1
AND Recette.idRecette = Vente.idRecette
AND Vente.idVente = Est_vendu.idVente
AND Est_vendu.idConsommable = Consommable.idConsommable
AND Vente.idVente = Billet.idVente
AND Billet.idTarif = Tarif.idTarif
```

Algèbre relationnel :

$$R = \Pi_{idVente} (Vente \bowtie \Pi_{idRecette} (\rho_{date='2014-12-12' \ \& \ idEtablissement=1} (Recette))) \\ \sum \Pi_{nomConso} (Consommable \bowtie \Pi_{idConso} (Est_vendu \bowtie R)) + \sum \Pi_{prix} (Tarif \bowtie \Pi_{idTarif} (Billet \bowtie R))$$

- **Liste des séances à venir pour un établissement donné :**

Requêtes SQL :

```
SELECT nomFilm, heure
FROM Film, Seance, Salle
WHERE heure >= NOW()
AND Film.idFilm = Seance.idFilm
AND Seance.idSalle = Salle.idSalle
AND idEtablissement = 2
```

Algèbre relationnel :

$$\Pi_{nomFilm, heure} (Film \bowtie \rho_{heure=NOW()} (\Pi_{idFilm, heure} (Seance \bowtie \rho_{idEtablissement=1} (Salle))))$$

CONCLUSION

Ce projet nous a permis de mieux comprendre et cerner les étapes d'analyse et de conception de la méthode **MERISE**, ainsi que les tenants et les aboutissants de mise en œuvre d'une base de données fonctionnelle.

Il nous a aussi permis de prendre en main les outils de développement tel que PhpMyAdmin et le langage SQL.

CREATION ET INSERTION DE DONNEES DE LA TABLE BILLET

```

CREATE TABLE IF NOT EXISTS `Billet` (
  `idBillet`      int(11)  NOT NULL AUTO_INCREMENT,
  `idClient`      int(11)  DEFAULT NULL,
  `idTarif`       int(11)  DEFAULT NULL,
  `idSeance`      int(11)  DEFAULT NULL,
  `idVente`       int(11)  DEFAULT NULL,
  PRIMARY KEY     (`idBillet`),
  KEY `FK_Billet_idClient` (`idClient`),
  KEY `FK_Billet_idTarif`  (`idTarif`),
  KEY `FK_Billet_idSeance` (`idSeance`),
  KEY `FK_Billet_idVente`  (`idVente`)
);

INSERT INTO `Billet` (`idBillet`, `idClient`, `idTarif`, `idSeance`, `idVente`) VALUES
(1, 1, 1, 1, 1),
(2, 2, 1, 2, 2),
(3, 3, 1, 1, 3),
(4, 4, 1, 2, 4),
(5, 4, 1, 3, 5),
(6, 3, 1, 4, 6);

```

CREATION ET INSERTION DE DONNEES DE LA TABLE CLIENT

```

CREATE TABLE IF NOT EXISTS `Client` (
  `idClient`      int(11)      NOT NULL AUTO_INCREMENT,
  `nomClient`     varchar(25)   DEFAULT NULL,
  `prenomClient`  varchar(25)   DEFAULT NULL,
  `adresseMail`   varchar(25)   DEFAULT NULL,
  `mdp`           varchar(25)   DEFAULT NULL,
  `ddnClient`     date          DEFAULT NULL,
  PRIMARY KEY     (`idClient`)
)

INSERT INTO `Client` (`idClient`, `nomClient`, `prenomClient`, `adresseMail`, `mdp`, `ddnClient`) VALUES
(1, 'nClient1', 'pClient1', 'adresse1', '3210', '2014-12-01'),
(2, 'nClient2', 'pClient2', 'adresse2', '3210', '2014-12-01'),
(3, 'nClient3', 'pClient3', 'adresse3', '3210', '2014-12-01'),
(4, 'nClient3', 'pClient3', 'adresse4', '3210', '2014-12-01');

```

CREATION ET INSERTION DE DONNEES DE LA TABLE CONSOMMABLE

```
CREATE TABLE IF NOT EXISTS `Consommable` (  
  `idConsommable`      int(11)          NOT NULL AUTO_INCREMENT,  
  `nomConso`           varchar(25)       DEFAULT NULL,  
  `prixConso`          float             DEFAULT NULL,  
  PRIMARY KEY (`idConsommable`)  
);  
  
INSERT INTO `Consommable` (`idConsommable`, `nomConso`, `prixConso`) VALUES  
(1, 'popcorn', 2),  
(2, 'soda', 1);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE EST_VENDU

```
CREATE TABLE IF NOT EXISTS `Est_vendu` (  
  `quantite`           int(11)          NOT NULL,  
  `idConsommable`      int(11)          NOT NULL,  
  `idVente`            int(11)          NOT NULL,  
  PRIMARY KEY          (`idConsommable`, `idVente`),  
  KEY `FK_Est_vendu_idVente` (`idVente`)  
);  
  
INSERT INTO `Est_vendu` (`quantite`, `idConsommable`, `idVente`) VALUES  
(1, 1, 1),  
(2, 1, 3),  
(1, 2, 1),  
(1, 2, 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE ETABLISSEMENT

```
CREATE TABLE IF NOT EXISTS `Etablissement` (  
  `idEtablissement`    int(11)          NOT NULL AUTO_INCREMENT,  
  `adresse`            varchar(50)       DEFAULT NULL,  
  `codePostal`         int(11)          DEFAULT NULL,  
  `idVille`            int(11)          DEFAULT NULL,  
  PRIMARY KEY          (`idEtablissement`),  
  KEY `FK_Etablissement_idVille` (`idVille`)  
);  
  
INSERT INTO `Etablissement` (`idEtablissement`, `adresse`, `codePostal`, `idVille`) VALUES  
(1, '5 rue machin', 51100, 1),  
(2, '5 rue test', 10000, 2);
```


CREATION ET INSERTION DE DONNEES DE LA TABLE FILM

```
CREATE TABLE IF NOT EXISTS `Film` (  
  `idFilm`          int(11)          NOT NULL AUTO_INCREMENT,  
  `nomFilm`         varchar(25)      DEFAULT NULL,  
  `duree` i         nt(11)          DEFAULT NULL,  
  `pays`            varchar(25)      DEFAULT NULL,  
  `synopsis`         varchar(500)     DEFAULT NULL,  
  `annee`           int(11)          DEFAULT NULL,  
  PRIMARY KEY (`idFilm`)  
);  
  
INSERT INTO `Film` (`idFilm`, `nomFilm`, `duree`, `pays`, `synopsis`, `annee`) VALUES  
(1, 'Vol au dessus d''un nid de', 120, 'USA', 'blabla', 1995),  
(2, 'Garry Topper', 100, 'UK', 'lbalba', 2001);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE JOUE

```
CREATE TABLE IF NOT EXISTS `Joue` (  
  `rule`            varchar(25)      DEFAULT NULL,  
  `idFilm`          int(11)          NOT NULL,  
  `idPersonne`      int(11)          NOT NULL,  
  PRIMARY KEY       (`idFilm`, `idPersonne`),  
  KEY `FK_Joue_idPersonne` (`idPersonne`)  
);  
  
INSERT INTO `Joue` (`rule`, `idFilm`, `idPersonne`) VALUES  
( 'principal', 1, 1),  
( 'secondaire', 1, 5),  
( 'principal', 2, 5),  
( 'secondaire', 2, 6);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE PERSONNE

```
CREATE TABLE IF NOT EXISTS `Personne` (  
  `idPersonne`      int(11)          NOT NULL AUTO_INCREMENT,  
  `nomPersonne`     varchar(25)      DEFAULT NULL,  
  `prenomPersonne`  varchar(25)      DEFAULT NULL,  
  `sexe`            tinyint(1)       DEFAULT NULL,  
  `ddnPersonne`     date             DEFAULT NULL,  
  PRIMARY KEY (`idPersonne`)  
);  
  
INSERT INTO `Personne` (`idPersonne`, `nomPersonne`, `prenomPersonne`, `sexe`, `ddnPersonne`) VALUES  
(1, 'Garry', 'Acteur1', 0, '2014-12-03'),  
(4, 'Garry', 'Acteur2', 1, '2014-12-01'),  
(5, 'Garry', 'realistaur', 1, '2014-11-10'),  
(6, 'goudou', 'poudou', 0, '2014-12-15'),  
(7, 'gada', 'pada', 1, '2014-11-17');
```

CREATION ET INSERTION DE DONNEES DE LA TABLE PRE_PROJETE

```
CREATE TABLE IF NOT EXISTS `Pre_projete` (  
  `idSeance`          int(11)          NOT NULL,  
  `idPub`             int(11)          NOT NULL,  
  PRIMARY KEY          (`idSeance`,`idPub`),  
  KEY `FK_Pre_projete_idPub`    (`idPub`)  
);
```

```
INSERT INTO `Pre_projete` (`idSeance`, `idPub`) VALUES  
(1, 1),  
(2, 1),  
(1, 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE PRODUIT

```
CREATE TABLE IF NOT EXISTS `Produit` (  
  `idFilm`            int(11)          NOT NULL,  
  `idPersonne`        int(11)          NOT NULL,  
  PRIMARY KEY          (`idFilm`,`idPersonne`),  
  KEY `FK_Produit_idPersonne`   (`idPersonne`)  
);
```

```
INSERT INTO `Produit` (`idFilm`, `idPersonne`) VALUES  
(1, 4),  
(2, 5);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE PUBLICITE

```
CREATE TABLE IF NOT EXISTS `Publicite` (  
  `idPub`             int(11)          NOT NULL AUTO_INCREMENT,  
  `nomPub`            varchar(25)      DEFAULT NULL,  
  `dureePub`          int(11)          DEFAULT NULL,  
  PRIMARY KEY (`idPub`)  
);
```

```
INSERT INTO `Publicite` (`idPub`, `nomPub`, `dureePub`) VALUES  
(1, 'pub1', 1),  
(2, 'pub2', 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE REALISE

```
CREATE TABLE IF NOT EXISTS `Realise` (  
  `idFilm`          int(11)          NOT NULL,  
  `idPersonne`      int(11)          NOT NULL,  
  PRIMARY KEY        (`idFilm`,`idPersonne`),  
  KEY `FK_Realise_idPersonne`    (`idPersonne`)  
);
```

```
INSERT INTO `Realise` (`idFilm`, `idPersonne`) VALUES  
(2, 4),  
(1, 7);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE RECETTE

```
CREATE TABLE IF NOT EXISTS `Recette` (  
  `idRecette`        int(11)          NOT NULL AUTO_INCREMENT,  
  `dateRecette`       date             DEFAULT NULL,  
  `idEtablissement`  int(11)          DEFAULT NULL,  
  PRIMARY KEY         (`idRecette`),  
  KEY `FK_Recette_idEtablissement` (`idEtablissement`)  
);
```

```
INSERT INTO `Recette` (`idRecette`, `dateRecette`, `idEtablissement`) VALUES  
(1, '2014-12-12', 1),  
(2, '2014-12-12', 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE SALLE

```
CREATE TABLE IF NOT EXISTS `Salle` (  
  `idSalle`          int(11)          NOT NULL AUTO_INCREMENT,  
  `nbPlace`          int(11)          DEFAULT NULL,  
  `equipement3d`     tinyint(1)      DEFAULT NULL,  
  `idEtablissement`  int(11)          DEFAULT NULL,  
  PRIMARY KEY         (`idSalle`),  
  KEY `FK_Salle_idEtablissement`    (`idEtablissement`)  
);
```

```
INSERT INTO `Salle` (`idSalle`, `nbPlace`, `equipement3d`, `idEtablissement`) VALUES  
(1, 100, 1, 1),  
(2, 200, 0, 1),  
(5, 100, 1, 2),  
(6, 150, 0, 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE SCENARISE

```
CREATE TABLE IF NOT EXISTS `Scenarise` (  
  `idFilm` int(11) NOT NULL,  
  `idPersonne` int(11) NOT NULL,  
  PRIMARY KEY (`idFilm`,`idPersonne`),  
  KEY `FK_Scenarise_idPersonne` (`idPersonne`)  
)
```

```
INSERT INTO `Scenarise` (`idFilm`, `idPersonne`) VALUES  
(1, 1),  
(2, 4);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE SEANCE

```
CREATE TABLE IF NOT EXISTS `Seance` (  
  `idSeance` int(11) NOT NULL AUTO_INCREMENT,  
  `heure` datetime DEFAULT NULL,  
  `placeRestantes` int(11) DEFAULT NULL,  
  `projection3d` tinyint(1) DEFAULT NULL,  
  `idSalle` int(11) DEFAULT NULL,  
  `idFilm` int(11) DEFAULT NULL,  
  `idVersion` int(11) DEFAULT NULL,  
  PRIMARY KEY (`idSeance`),  
  KEY `FK_Seance_idSalle` (`idSalle`),  
  KEY `FK_Seance_idFilm` (`idFilm`),  
  KEY `FK_Seance_idVersion` (`idVersion`)  
);
```

```
INSERT INTO `Seance` (`idSeance`, `heure`, `placeRestantes`, `projection3d`, `idSalle`, `idFilm`, `idVersion`) VALUES  
(1, '2014-12-14 12:20:00', NULL, 0, 1, 1, 1),  
(2, '2014-12-10 16:50:00', NULL, 1, 2, 2, 1),  
(3, '2014-12-16 11:00:00', NULL, 0, 5, 1, 1),  
(4, '2014-12-16 10:00:00', NULL, 1, 6, 2, 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE TARIF

```
CREATE TABLE IF NOT EXISTS `Tarif` (  
  `idTarif` int(11) NOT NULL AUTO_INCREMENT,  
  `nomTarif` varchar(25) DEFAULT NULL,  
  `prix` float DEFAULT NULL,  
  PRIMARY KEY (`idTarif`)  
);
```

```
INSERT INTO `Tarif` (`idTarif`, `nomTarif`, `prix`) VALUES  
(1, 'Tarif normal', 8);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE VENTE

```
CREATE TABLE IF NOT EXISTS `Vente` (  
  `idVente`          int(11)          NOT NULL AUTO_INCREMENT,  
  `idRecette`        int(11)          DEFAULT NULL,  
  PRIMARY KEY        (`idVente`),  
  KEY `FK_Vente_idRecette`    (`idRecette`)  
);
```

```
INSERT INTO `Vente` (`idVente`, `idRecette`) VALUES  
(1, 1),  
(2, 1),  
(3, 1),  
(4, 1),  
(5, 2),  
(6, 2);
```

CREATION ET INSERTION DE DONNEES DE LA TABLE VERSION

```
CREATE TABLE IF NOT EXISTS `Version` (  
  `idVersion`        int(11)          NOT NULL AUTO_INCREMENT,  
  `nomVersion`        varchar(25)      DEFAULT NULL,  
  PRIMARY KEY (`idVersion`)  
);
```

```
INSERT INTO `Version` (`idVersion`, `nomVersion`) VALUES  
(1, 'VF'),  
(2, 'VOSTFR');
```

CREATION ET INSERTION DE DONNEES DE LA TABLE VILLE

```
CREATE TABLE IF NOT EXISTS `Ville` (  
  `idVille`          int(11)          NOT NULL AUTO_INCREMENT,  
  `nomVille`          varchar(25)      DEFAULT NULL,  
  PRIMARY KEY (`idVille`)  
);
```

```
INSERT INTO `Ville` (`idVille`, `nomVille`) VALUES  
(1, 'Reims'),  
(2, 'Troyes');
```

CONSTRAINTES POUR LA TABLE `BILLET`

```
ALTER TABLE `Billet`  
  ADD CONSTRAINT `FK_Billet_idClient` FOREIGN KEY (`idClient`) REFERENCES `Client` (`idClient`),  
  ADD CONSTRAINT `FK_Billet_idSeance` FOREIGN KEY (`idSeance`) REFERENCES `Seance` (`idSeance`),  
  ADD CONSTRAINT `FK_Billet_idTarif` FOREIGN KEY (`idTarif`) REFERENCES `Tarif` (`idTarif`),  
  ADD CONSTRAINT `FK_Billet_idVente` FOREIGN KEY (`idVente`) REFERENCES `Vente` (`idVente`);
```

CONSTRAINTES POUR LA TABLE `EST_VENDU`

```
ALTER TABLE `Est_vendu`  
  ADD CONSTRAINT `FK_Est_vendu_idConsommable` FOREIGN KEY (`idConsommable`) REFERENCES  
    Consommable (`idConsommable`),  
  ADD CONSTRAINT `FK_Est_vendu_idVente` FOREIGN KEY (`idVente`) REFERENCES `Vente` (`idVente`);
```

CONSTRAINTES POUR LA TABLE `ETABLISSEMENT`

```
ALTER TABLE `Etablissement`  
  ADD CONSTRAINT `FK_Etablissement_idVille` FOREIGN KEY (`idVille`) REFERENCES `Ville` (`idVille`);
```

CONSTRAINTES POUR LA TABLE `JOUE`

```
ALTER TABLE `Joue`  
  ADD CONSTRAINT `FK_Joue_idFilm` FOREIGN KEY (`idFilm`) REFERENCES `Film` (`idFilm`),  
  ADD CONSTRAINT `FK_Joue_idPersonne` FOREIGN KEY (`idPersonne`) REFERENCES `Personne` (`idPersonne`);
```

CONSTRAINTES POUR LA TABLE `PRE_PROJETE`

```
ALTER TABLE `Pre_projete`  
  ADD CONSTRAINT `FK_Pre_projete_idPub` FOREIGN KEY (`idPub`) REFERENCES `Publicite` (`idPub`),  
  ADD CONSTRAINT `FK_Pre_projete_idSeance` FOREIGN KEY (`idSeance`) REFERENCES `Seance` (`idSeance`);
```

CONSTRAINTES POUR LA TABLE `PRODUIT`

```
ALTER TABLE `Produit`  
  ADD CONSTRAINT `FK_Produit_idFilm` FOREIGN KEY (`idFilm`) REFERENCES `Film` (`idFilm`),  
  ADD CONSTRAINT `FK_Produit_idPersonne` FOREIGN KEY (`idPersonne`) REFERENCES `Personne`  
    (`idPersonne`);
```

CONSTRAINTES POUR LA TABLE `REALISE`

```
ALTER TABLE `Realise`  
  ADD CONSTRAINT `FK_Realise_idFilm` FOREIGN KEY (`idFilm`) REFERENCES `Film` (`idFilm`),  
  ADD CONSTRAINT `FK_Realise_idPersonne` FOREIGN KEY (`idPersonne`) REFERENCES `Personne`  
    (`idPersonne`);
```

CONSTRAINTES POUR LA TABLE `RECETTE`

```
ALTER TABLE `Recette`  
  ADD CONSTRAINT `FK_Recette_idEtablissement` FOREIGN KEY (`idEtablissement`) REFERENCES  
    `Etablissement` (`idEtablissement`);
```

CONSTRAINTES POUR LA TABLE `SALLE`

```
ALTER TABLE `Salle`  
  ADD CONSTRAINT `FK_Salle_idEtablissement` FOREIGN KEY (`idEtablissement`) REFERENCES `Etablissement`  
    (`idEtablissement`);
```

CONSTRAINTES POUR LA TABLE `SCENARISE`

```
ALTER TABLE `Scenarise`  
  ADD CONSTRAINT `FK_Scenarise_idFilm` FOREIGN KEY (`idFilm`) REFERENCES `Film` (`idFilm`),  
  ADD CONSTRAINT `FK_Scenarise_idPersonne` FOREIGN KEY (`idPersonne`) REFERENCES `Personne`  
    (`idPersonne`);
```

CONSTRAINTES POUR LA TABLE `SEANCE`

```
ALTER TABLE `Seance`  
  ADD CONSTRAINT `FK_Seance_idFilm` FOREIGN KEY (`idFilm`) REFERENCES `Film` (`idFilm`),  
  ADD CONSTRAINT `FK_Seance_idSalle` FOREIGN KEY (`idSalle`) REFERENCES `Salle` (`idSalle`),  
  ADD CONSTRAINT `FK_Seance_idVersion` FOREIGN KEY (`idVersion`) REFERENCES `Version` (`idVersion`);
```

CONSTRAINTES POUR LA TABLE `VENTE`

```
ALTER TABLE `Vente`  
  ADD CONSTRAINT `FK_Vente_idRecette` FOREIGN KEY (`idRecette`) REFERENCES `Recette` (`idRecette`);
```