



Données structurées et bases de données
Rapport de projet

15/02/2020

Groupe: L2-MIAGE 2

Chaimae Serhrouchni

chaimae.serhrouchni@etudiant.univ-rennes1.fr

Ibrahim Kalilou Simpara

ibrahim-kalilou.simpara@etudiant.univ-rennes1.fr

Rapport de projet DSB – Partie 1

1) Introduction:



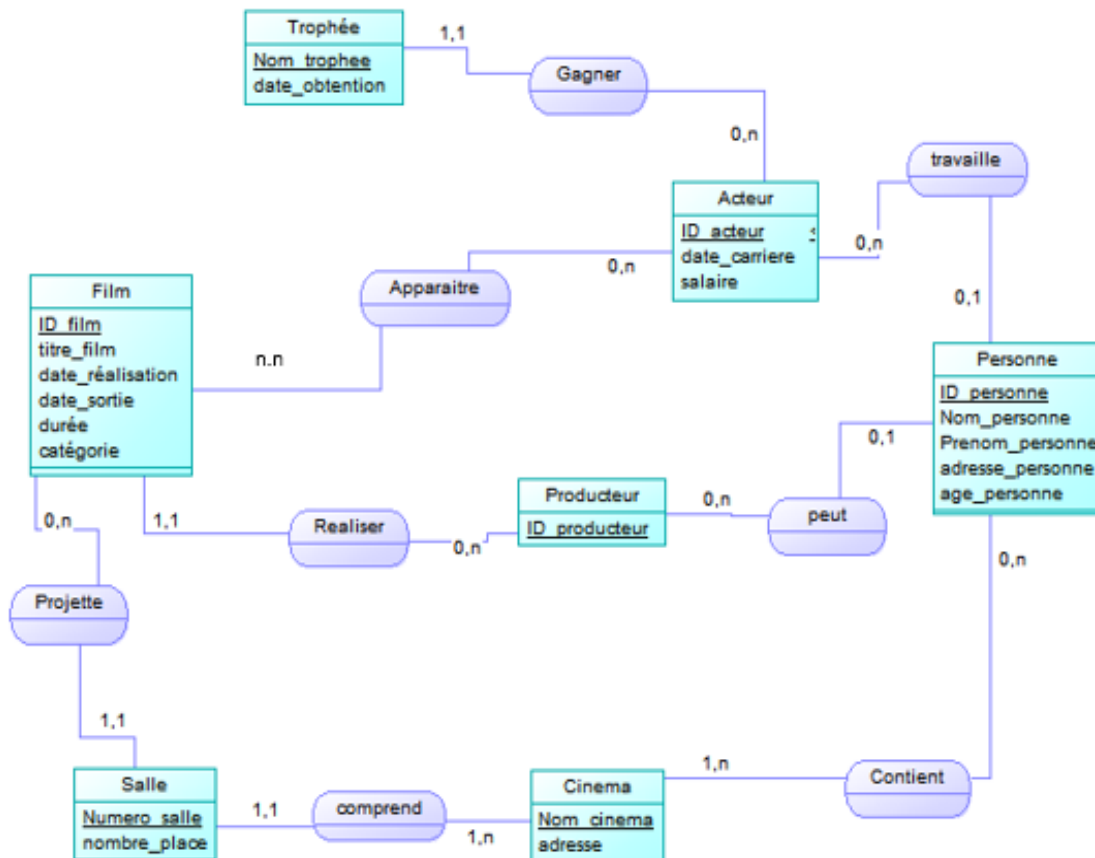
Notre projet consiste en la conception d'une base de données dans un environnement cinématographique.

Nous souhaitons mettre en place une application qui permettra de connaître le nombre de personnes présentes dans un cinéma ainsi que le nombre de salles qui la compose , les films et leurs catégories qui sont diffusés dans le cinéma ainsi que le nombre d'acteurs qui la compose. l'application pourra reconnaître aussi le producteur de tel ou tel film ainsi que les acteurs et les trophées remportés par chaque acteur .



2) Modèle conceptuel (MCD)

2-1) Schéma:



2-2) Les attributs et leurs significations:

Film:

ID_film (clé primaire)* : l'identifiant du film

titre_film: le titre du film

date_réalisation: la date de début de réalisation du film

date_sortie: la date de sortie du film dans les salles de cinéma

durée: la durée du film

Catégorie: la catégorie du film

Personne:

ID_personne (clé primaire)*: l'identifiant de la personne
Nom_personne: nom de la personne
Prénom_personne: prénom de la personne
Adresse_personne: adresse de la personne
Age_personne: age de la personne

Acteur:

Id_acteur (clé primaire)*: l'identifiant de l'acteur
date_carriere: la date du début de la carrière d'un acteur
salaire: le salaire mensuelle de l'acteur

Producteur:

ID_producteur (clé primaire)*: l'identifiant du producteur

Trophé:

nom_trophe (clé primaire)*: le nom du trophé
date: la date d'obtention du trophée

Cinéma:

nom_cinema (clé primaire)*: le nom du cinéma
Adresse: L'adresse du cinéma

Salle:

Numéro_salle (clé primaire)*: numéro de salle où le film va être projeté
Nombre_place: le nombre de place dans une salle

2-3) Associations et cardinalités:

- *l'association entre film et salle est 'Projette'*: Un film peut être projeté dans plusieurs salles comme il peut être projeté dans aucune salle, une salle projette un unique film.
- *l'association entre film et producteur est 'Réaliser'*: Un film est réalisé par un seul et unique producteur, mais un producteur peut réaliser plusieurs films.
- *l'association entre film et acteur est 'Apparaître'*: Un film ne fait pas apparaître un seul ou aucune acteur, et un acteur peut apparaître dans plusieurs films.

- *l'association entre salle et cinéma est 'comprend'*: Une salle est comprise dans un unique cinéma, alors qu'un cinéma peut avoir plusieurs ou une unique salle.
- *l'association entre cinéma et personne est 'contient'*: Un cinéma contient plusieurs personnes, et une personne peut visiter plusieurs ou aucun cinéma.
- *L'association entre personne et acteur est 'travaille'*: Une personne peut travailler autant qu'acteur ou pas, le travaille d'acteur peut être fait par plusieurs ou aucune personne.
- *l'association entre personne et producteur est 'peut'*: Une personne peut être un producteur, et plusieurs personnes peuvent être des producteurs.
- *l'association entre acteur et trophée est 'Gagner'*: Un acteur peut gagner plusieurs ou aucun trophée, un trophée peut être gagné par un seul acteur.

3) Schéma relationnelle (MLD):

NB: nous avons identifiées les clés étrangères par # à la place des doubles traits en bas car le word sur le google drive ne nous permet pas de le faire par contre les attributs obligatoire sont toujours spécifiés par *

1) Application de la règle 1 :

Trophee(nom_trophee, date_obtention)

Acteur(ID_acteur, date_carrière, salaire)

Film(ID_film, titre_film, date_realisation, date_sortie, durée, catégorie)

Salle(Numero_salle, nombre_place)

Cinema(Nom_cinema, adresse)

Personne(ID_personne, Nom_personne, Prenom_personne, adresse_personne, age_personne)

Producteur(ID_producteur) .

2) Application de la règle 2 et 2 bis :

Trophee(ID_trophe, nom_trophee*, date_obtention, *ID_acteur#)

Film(ID_film, titre_film, date_realisation, date_sortie, durée, catégorie, *ID_producteur#)

Salle(Numero_salle, nombre_place, *ID_film#, *Nom_Cinema#)

travaille(*ID_personne#, *ID_acteur#)

peut(*ID_personne#, *ID_producteur#)

3) Application de la règle 3:

Contient(*ID_personne#, *Nom_cinema#)

Apparaître(*ID_film#, *ID_acteur#)

4) Résultat final:

Trophee(ID_trophe, nom_trophee, date_obtention, *ID_acteur#)

Acteur(ID_acteur, date_carrière, salaire)

Personne(ID_personne, Nom_personne, Prenom_personne, adresse_personne, age_personne)

Cinema(Nom_cinema, adresse)

Producteur(ID_producteur) .

Film(ID_film, titre_film, date_realisation, date_sortie, durée, catégorie, *ID_producteur#)

Salle(Numero_salle, nombre_place, *ID_film#, *Nom_Cinema#)

travaille(*ID_personne#, *ID_acteur#)

peut(*ID_personne, *ID_producteur)

Contient(*ID_personne#, *Nom_cinema#)

Apparaître(*ID_film#, *ID_acteur#)

4. Schéma physique (MPD):

```
CREATE TABLE Personne(ID__personne INTEGER NOT NULL,  
                        Nom__personne VARCHAR(20),  
                        Prenom__personne VARCHAR(20),  
                        adresse__personne VARCHAR(20),  
                        age__personne INTEGER,  
                        PRIMARY KEY(ID__personne) )
```

```
CREATE TABLE Cinema(Nom__cinema VARCHAR(20) NOT NULL,  
                     adresse VARCHAR(20),  
                     PRIMARY KEY(Nom__cinema))
```

```
CREATE TABLE Produits (ID__producteur INTEGER PRIMARY KEY)
```

```
CREATE TABLE Acteur(ID__acteur INTEGER NOT NULL ,  
                     date__carrière DATE,  
                     salaire INTEGER,  
                     PRIMARY KEY( ID__acteur) )
```

```
CREATE TABLE Trophee(nom__trophee VARCHAR(20),  
                      date obtention DATE,  
                      ID__acteur INT,  
                      PRIMARY KEY (nom__trophee) ,  
                      FOREIGN KEY(ID__acteur) REFERENCES Acteur )
```

```
CREATE TABLE Film(ID__film INTEGER NOT NULL,  
                   titre__film VARCHAR(20),  
                   date__réalisation DATE,  
                   date sortie DATE,  
                   duree INTEGER,  
                   categorie VARCHAR(20),  
                   ID__producteur INTEGER,  
                   PRIMARY KEY (ID__film),  
                   FOREIGN KEY (ID__producteur) REFERENCES  
Producteur)
```

```
CREATE TABLE Salle(Numero_salle INTEGER NOT NULL,  
                    nombre_place INTEGER ,  
                    ID_film INTEGER NOT NULL,  
                    Nom_Cinema VARCHAR(20),  
                    PRIMARY KEY(Numero_salle),  
                    FOREIGN KEY (ID_film) REFERENCES Film ,  
                    FOREIGN KEY (Nom_Cinema) REFERENCES Cinema )
```

```
CREATE TABLE travaille(ID_personne INTEGER NOT NULL,  
                        ID_acteur INTEGER NOT NULL,  
                        PRIMARY KEY (ID_personne),  
                        FOREIGN KEY (ID_personne) REFERENCES  
Personne,  
                        FOREIGN KEY (ID_acteur) REFERENCES Acteur)
```

```
CREATE TABLE peut(ID_personne INTEGER NOT NULL,  
                   ID_producteur INTEGER NOT NULL ,  
                   PRIMARY KEY(ID_personne),  
                   FOREIGN KEY(ID_personne) REFERENCES Personne,  
                   FOREIGN KEY(ID_producteur) REFERENCES Producteur)
```

```
CREATE TABLE Contient(ID_personne INTEGER NOT NULL,  
                       Nom_cinema VARCHAR(20) NOT NULL,  
                       PRIMARY KEY (ID_personne, Nom_cinema),  
                       FOREIGN KEY (ID_personne) REFERENCES Personne,  
                       FOREIGN KEY (Nom_cinema) REFERENCES Cinema )
```

```
CREATE TABLE Apparaître(ID_film INTEGER NOT NULL ,  
                          ID_acteur INTEGER NOT NULL,  
                          PRIMARY KEY (ID_film,ID_acteur),  
                          FOREIGN KEY (ID_film) REFERENCES Film,  
                          FOREIGN KEY (ID_acteur) REFERENCES Acteur)
```


Rapport de projet DSB

```
CREATE TABLE Personne(ID_personne INTEGER NOT NULL, Nom_personne VARCHAR(20),
                        Prenom_personne VARCHAR(20), adresse_personne VARCHAR(20),
                        age_personne INTEGER, PRIMARY KEY(ID_personne) );

CREATE TABLE Cinema(Nom_cinema VARCHAR(20) NOT NULL, adresse VARCHAR(20), PRIMARY KEY(Nom_cinema));

CREATE TABLE Producteur (ID_producteur INTEGER PRIMARY KEY);

CREATE TABLE Acteur(ID_acteur INTEGER NOT NULL , date_carrière DATE,
                     salaire INTEGER, PRIMARY KEY( ID_Acteur) );

CREATE TABLE Trophée(nom_trophée VARCHAR(20), date_obtention DATE, ID_acteur INT,
                      PRIMARY KEY (nom_trophée) , FOREIGN KEY(ID_acteur) REFERENCES Acteur(ID_acteur));

CREATE TABLE Film(ID_film INTEGER NOT NULL, titre_film VARCHAR(20),
                   date_realisation DATE, date_sortie DATE, durée INTEGER, categorie VARCHAR(20),
                   ID_producteur INTEGER,
                   PRIMARY KEY (ID_film),
                   FOREIGN KEY (ID_producteur) REFERENCES Producteur(ID_producteur));

CREATE TABLE Salle(Nomero_salle INTEGER NOT NULL, nombre_place INTEGER , ID_film INTEGER NOT NULL,
                    Nom_Cinema VARCHAR(20),
                    PRIMARY KEY(Nomero_salle),
                    FOREIGN KEY (ID_film) REFERENCES Film (ID_film),
                    FOREIGN KEY (Nom_Cinema) REFERENCES Cinema (Nom_Cinema) );

CREATE TABLE travaille(ID_personne INTEGER NOT NULL, ID_acteur INTEGER NOT NULL,
                        PRIMARY KEY (ID_personne),
                        FOREIGN KEY (ID_personne) REFERENCES Personne(ID_personne),
                        FOREIGN KEY (ID_acteur) REFERENCES Acteur(ID_acteur));

CREATE TABLE peut(ID_personne INTEGER NOT NULL, ID_producteur INTEGER NOT NULL ,
                   PRIMARY KEY(ID_personne),
                   FOREIGN KEY(ID_personne) REFERENCES Personne(ID_personne),
                   FOREIGN KEY(ID_producteur) REFERENCES Producteur(ID_producteur));

CREATE TABLE Contient(ID_personne INTEGER NOT NULL, Nom_cinema VARCHAR(20) NOT NULL,
                       PRIMARY KEY (ID_personne, Nom_cinema),
                       FOREIGN KEY (ID_personne) REFERENCES Personne(ID_personne),
                       FOREIGN KEY (Nom_cinema) REFERENCES Cinema(Nom_cinema) );

CREATE TABLE Apparaître(ID_film INTEGER NOT NULL , ID_acteur INTEGER NOT NULL,
                          PRIMARY KEY (ID_film, ID_acteur),
                          FOREIGN KEY (ID_film) REFERENCES Film(ID_film),
                          FOREIGN KEY (ID_acteur) REFERENCES Acteur(ID_acteur) );
```

5. Peuplement des tables:

Personne:

Einstein Albert dont l'âge et l'adresse sont inconnus a comme identifiant 5.

PERSONNE (5, 'Einstein' , 'Albert' , NULL, NULL)

Jennifer Lopez dont l'adresse est inconnue a comme identifiant 3 et âge 52 ans.

PERSONNE(3, 'Lopez', 'Jennifer', NULL, 52)

Serhrouchni chaimae a 21 ans, habite à Cesson-Sévigné et son identifiant est 1.

PERSONNE (1, 'Serhrouchni', 'Chaimae', 'Cesson-sévigné', 21)

Ibrahim Kalilou Simpara a 20 ans, habite à villejean, Rennes et son identifiant est 2.

PERSONNE (2, 'Kalilou Simpara', 'Ibrahim', 'Villejean Rennes', 20)

Cinéma:

Le cinéma Gaumont dont l'adresse est :12 Rue Yvonne Jean-Haffen,35000 Rennes.

Cinema('Gaumont', '12 Rue Yvonne Jean-Haffen,35000 Rennes')

Producteur:

Le producteur dont l'identifiant est 2:

Producteur(2)

Acteur:

L'acteur dont l'identifiant est 3, a débuté sa carrière le 05 mars 1995 et son salaire est de 75000 €.

Acteur(3, '05/03/1995', 75000)

L'acteur dont l'identifiant est 1 et dont la date de début de carrière est inconnu est payer a 90000 €

Acteur(1, 'NULL', 90000)

Trophee:

Le trophée Oscar a été remporté par l'acteur qui a comme identifiant 1 le 23 février 2019.

Trophee('Oscar', 23/02/2019, 1)

Film:

Le film Avengers Endgame qui est de catégorie science-fiction et dont l'identifiant est 7 a été réalisé le 05 avril 2018 et est sorti le 03 janvier 2019. Il dure 181 minutes et il a été produit par le producteur dont l'identifiant est 2.

Film(7, 'Avengers Endgame', '05/04/2018', '03/01/2019', 181, 'science-fiction', 2).

Le film Spider-man: No way Home qui est de catégorie Fantaisie et dont l'identifiant est 8 a été réalisé le 01 février 2021 et est sorti le 15 décembre 2021. Il dure 170 minutes et a été produit par le producteur dont l'identifiant est 2.

Film (8, 'Spider-man: No way Home', '01-02-2021', '15-12-2021', 170, 'Fantaisie' 2)

Salle

La salle numéro 17 qui comporte 100 places et qui projette le film qui a comme identifiant 7 dans le cinéma Gaumont.

Salle(17, 100)

La salle numéro 20 comporte 50 places et qui projette le film qui a comme identifiant 8 dans le cinéma Gaumont..

Salle(20, 50, 8, 'Gaumont')

travail:

La personne qui a comme identifiant 3 correspond à l'acteur qui a comme identifiant 3 (c'est à dire la personne qui a comme identifiant 3 est un acteur)

Travail(3, 3)

peut:

La personne qui a comme identifiant 2 correspond au producteur qui a comme identifiant 2 (c'est-à-dire la personne qui a comme identifiant 2 est un producteur).

Peut(2, 2)

Contient:

La personne qui a comme identifiant 5 est présente dans le cinéma Gaumont.

Contient(5, 'Gaumont')

Apparaître:

L'acteur dont l'identifiant est 3 apparaît dans le film dont l'identifiant est 7
Apparaître(7, 3)

6. Requêtes:

a- une sélection avec projection:

français: Les noms et prénoms des personnes dont l'âge est inférieur à 50.

SQL: SELECT Nom_personne, Prenom_personne FROM Personne
WHERE age_personne < 50

Algèbre: $\pi_{\text{Nom_personne}, \text{Prenom_personne}}(\sigma_{\text{age} < 50}(\text{Personne}))$

b- une jointure:

français: Le nom et l'adresse des personnes qui sont des acteurs

SQL: select Nom_personne, adresse_personne from Personne, Acteur
where Personne.ID_personne = Acteur.ID_acteur

Algèbre: $\pi_{\text{Nom_personne}, \text{adresse_personne}}(\text{Personne} \bowtie \text{Acteur})$

c- une moyenne sur l'intégralité d'un attribut:

français: La moyenne des salaires des acteurs

SQL: SELECT AVG(salaire) FROM Acteur

Algèbre: Pas d'opérateur spécifié en algèbre relationnelle

d- un regroupement avec calcul:

français: Nombre de film par catégorie

SQL: SELECT categorie, COUNT(ID_film) FROM Film
GROUP BY categorie

Algèbre: Pas d'opérateur spécifié en algèbre relationnelle

e- une différence:

français: Personnes qui ne sont pas des producteurs

SQL: SELECT Nom_personne, Prenom_personne FROM Personne
WHERE ID_personne NOT IN
(SELECT ID_personne FROM Personne, Producteur WHERE
Personne.ID_personne=Producteur.ID_producteur)

Algèbre: $\pi_{ID_personne}(Personne) - \pi_{ID_producteur}(Producteur)$

f- une division:

français: Noms et prénoms des producteurs qui réalisent tous les films

SQL: SELECT Nom_personne, Prenom_personne
FROM Personne
WHERE NOT EXISTS
 (SELECT ID_producteur FROM Film
 WHERE NOT EXISTS
 (SELECT ID_producteur FROM Producteur
 WHERE Producteur.ID_producteur=Personne.IDpersonne
 AND Producteur.ID_producteur= Film.ID_producteur))

Algèbre: R1: $\pi_{ID_producteur}(Film) \div \pi_{ID_producteur}(Producteur)$

R2: $\pi_{nom_personne,prenom_personne}(Personne \bowtie R1)$

to be continued...

**Directed by
produced by**

Cast

**Serhrouchni chaimae
Simpara Ibrahim Kalilou**

**Editor
Script supervisor**

**Serhrouchni Chaimae
Simpara Ibrahim Kalilou**

Special Thanks

**Bariatti Francesco
Annie foret**