

Compte rendu :

Atelier Environnement Client Serveur

Exercice Partie I, Partie II et Partie III

Élaboré par :

- William Smith

Année scolaire 2023 / 2024

Sommaire

1. Partie I : Environnement client, serveur, procédures, triggers et transactions.
 - a. Exercice 1
 - i. Syntaxe création de la table FILM.
 - ii. Mettre un index primaire sur la colonne IDENT_FILM puis un index unique sur la colonne GENRE1 associée à Pays.
 - b. Exercice 2
 - i. Ajouter une colonne complémentaire nommée NUM_REAL.
 - ii. Cette colonne est une clé étrangère sur la table RÉALISATEUR
 - iii. Ajouter une valeur par défaut sur la colonne RECETTE avec la valeur 0.
 - iv. Mettre les colonnes TITRE et PAYS en NOT NULL.
 - v. Supprimer la contrainte.
 - c. Exercice 3
 - i. Créer une vue FILM 3 à partir de la table FILM contenant les quatre premières colonnes de la table FILM ainsi que la colonne RESUME.
 - ii. Supprimer cette vue.
 - d. Exercice 4
 - i. Créer une séquence sur la colonne IDENT_FILM qui commence à 12 et avec une valeur maximum de 9 999.
 - ii. Renommer la table FILM en FILMOLD.
 - iii. Créer une nouvelle table FILM de la table FILMOLD.
 - iv. Supprimer la table FILM et la table FILMOLD.

2. Partie II : Création d'une base de données pour gérer la location de matériel de ski.

- a. Exercice 1
 - i. Identification des clés primaires et des clés secondaires.
- b. Exercice 2
 - i. Identification des formes normales.
- c. Exercice 3
 - i. Identification des contraintes d'intégrité référentielle
- d. Exercice 4
 - i. Réalisation du dictionnaire des données.
- e. Exercice 5
 - i. La liste des clients dont le nom commence par un 'D'.
 - ii. Les noms et prénoms de tous les clients.
 - iii. La liste des fiches pour les clients qui habitent en Loire-Atlantique.
 - iv. Le détail de la fiche n°1002.
 - v. Le prix journalier moyen de location par gamme.
 - vi. Les articles qui ont été loués au moins trois fois.
 - vii. Le détail de la fiche n°1002 avec le total.

3. Partie III : Environnement client serveur, procédures, triggers et transactions.

a. Exercice 1

- i. Création de la table FILM.
- ii. Création de la table ACTEUR.
- iii. Création de la table STATISTIQUE.

b. Exercice 2

- i. Création de la table REALISATEUR.
- ii. Création de la table CASTING.
- iii. Création de la table PAYS.

c. Exercice 3

- i. Sélectionner toutes les informations sur les films réalisés par un réalisateur français triés par le nom du film.
- ii. Sélectionner le nom du film, la date de sortie, le nom du réalisateur, le nom des acteurs, leur date de naissance, ainsi que le budget du film. Le tout trié en ordre descendant par titre du film et nom des acteurs.
- iii. Trouver le nombre d'acteurs par film dans la base de données. Afficher le titre, la date de sortie, le nom du réalisateur et le distributeur.
- iv. Sélectionner le titre du film, la date de sortie, le nom et prénom du réalisateur, le nom et prénom de l'acteur, sa date de naissance, le budget du film et le nombre d'entrées en France des films qui ont un acteur algérien
- v. Sélectionner le film qui a réalisé la recette la plus élevée dans le monde.
- vi. Sélectionner l'acteur qui a joué dans deux films différents.
- vii. Sélectionner la personne qui est à la fois réalisateur et acteur.
- viii. Sélectionner les acteurs qui ont joué dans des films dont le nom du film commence par la lettre S. Indiquer leur rôle et leur nationalité.
- ix. Sélectionner les acteurs qui sont nés entre 1948 et mai 1978 ainsi que le nombre de jours total de tournage qu'ils ont réalisés.

4. Liens Github.

Partie 1 : Environnement client, serveur, procédures, triggers et transactions.

Exercice 1

1. Syntaxe de la création de la table FILM :

```
CREATE TABLE FILM (  
    Ident_Film INT PRIMARY KEY,  
    Titre VARCHAR(250),  
    Genre1 VARCHAR(50),  
    Recette DECIMAL(12,2),  
    Date_Sortie DATE,  
    Pays INT,  
    NB_Entree INT,  
    Date_Heure_Saisie DATETIME,  
    Resume TEXT  
);
```

2. Ajout des Index sur la colonne IDENT_FILM et GENRE1 :

```
CREATE UNIQUE INDEX Index_Ident_Film on FILM (Ident_Film);  
  
CREATE INDEX Index_Genre1_Film on FILM (GENRE1, PAYS);
```

Exercice 2

1. Ajout de la colonne complémentaire NUM_REAL :

```
ALTER TABLE FILM  
ADD COLUMN Num_Real INT;
```

2. Ajout de la clé étrangère sur la table REALISATEUR :

```
ALTER TABLE FILM  
ADD CONSTRAINT Fk_Realisateur  
FOREIGN KEY (Num_Real) REFERENCES Realisateur(Ident_Realisateur);
```

3. Ajout de la valeur par défaut de la colonne RECETTE :

```
ALTER TABLE FILM  
ALTER COLUMN Recette SET DEFAULT 0;
```

4. Ajout des valeurs par défaut des colonnes TITRE et PAYS :

```
ALTER TABLE FILM  
ALTER COLUMN TITRE VARCHAR(250) NOT NULL;  
  
ALTER TABLE FILM  
ALTER COLUMN PAYS INT NOT NULL;
```

5. Suppression de la contrainte de la colonne REALISATEUR :

```
ALTER TABLE FILM  
DROP FOREIGN KEY Fk_Realisateur;
```

Exercice 3

1. Création de la vue FILM2 :

```
CREATE VIEW FILM2 AS  
Select Ident_Film, Titre, Genre1, Recette, Resume  
FROM FILM;
```

2. Suppression de la vue FILM2 :

```
DROP VIEW FILM2;
```

Exercice 4

1. Création d'une séquence sur la colonne IDENT_FILM :

```
CREATE SEQUENCE Sequence_Ident_Film  
START WITH 12  
INCREMENT BY 1  
MAXVALUE 9999;
```

2. Renommage de la table FILM :

```
ALTER TABLE FILM RENAME TO FILMOLD;
```

3. Création de la nouvelle table FILM à partir de la table FILMOLD :

```
CREATE TABLE FILM AS SELECT * FROM FILMOLD;
```

4. Suppression des deux tables :

```
DROP TABLE FILM;  
  
DROP TABLE FILMOLD;
```


Partie 2 : Création d'une base de données pour gérer la location de matériel de ski.

Exercice 1

La clé primaire de la table **Clients** est la colonne **noCli**, car cette colonne est l'identifiant unique pour chaque client, les autres colonnes stockent les détails de chaque client comme leur adresse ou la ville. Dans cette table, il n'y a pas de clé étrangère, puisqu'aucune colonne se réfère à une autre table.

La clé primaire de la table **Gammes** est la colonne **codeGam**, car cette colonne est l'identifiant unique pour chaque gamme de produit, l'autre colonne stocke le détail de la gamme. Dans cette table, il n'y a aucune clé étrangère, parce qu'aucune colonne se réfère à une autre table.

La clé primaire de la table **Tarifs** est la colonne **codeTarif**, puisque cette colonne est l'identifiant unique pour chaque tarif, les colonnes différentes stockent le libelle et le prix par jour. Dans cette table, il n'y a aucune clé étrangère, comme aucune colonne se réfère à une table différente.

La clé primaire de la table **Catégories** est la colonne **codeCate**, car cette colonne stocke l'identifiant unique pour chaque catégorie de produits, l'autre colonne stocke le détail de la catégorie. Dans cette table, il n'y a aucune clé étrangère, puisqu'aucune colonne se réfère à une autre table.

La clé primaire de la table **GrilleTarifs** n'existe pas. Effectivement, aucun enregistrement ne possède une donnée unique. Une clé primaire doit être unique afin de pouvoir identifier chaque enregistrement. Cependant, cette table possède trois clés étrangères, **codeGam**, qui se réfère à la table Gammes, **codeCate**, qui se réfère à la table Catégories, et la colonne **codeTarifs**, qui se réfère la table Tarifs.

La clé primaire de la table **Articles** est la colonne **refart**, car cette colonne est l'identifiant unique qui permet d'identifier chaque article enregistré dans la table. Les autres colonnes stockent les informations de l'article comme sa désignation et le code de sa catégorie. Cette table possède deux clés étrangères, **codeGam**, qui se réfère à la table Gammes et la colonne **codeCate**, qui se réfère à la table Catégorie.

La clé primaire de la table **Fiches** n'existe pas. Effectivement, si la colonne **noFic** est définie comme clé primaire, cela signifie que chaque valeur doit être unique pour chaque enregistrement, ce qui permet d'assurer l'unicité des données. Or, des données apparaissent plusieurs dans la table. La table possède également deux clés étrangères, la colonne **noCli**, qui se réfère à la table Clients, et la colonne **refart**, qui se réfère à la table Articles.

Exercice 2

Les quatre tables qui respectent les trois formes normales sont la table **Gammes, Tarifs, Categories et Clients**, car elles respectent la première forme normale qui déclare que chaque colonne doit contenir des valeurs atomiques, c'est-à-dire une seule valeur et non un ensemble, et unique. Elles respectent la deuxième forme normale puisque qu'il n'y a pas de colonne non clé et donc qu'il n'y a pas de dépendance partielle, ainsi que la troisième forme normale qui stipule qu'il n'y a pas de dépendance dans la clé primaire.

Cependant, la table **Articles** ne respecte pas les trois formes normales. La table ne respecte pas la troisième forme normale, car les colonnes **CodeGam** et **CodeCat** ont une relation directe entre elles sans passer par des tables intermédiaires pour enlever les dépendances transitives.

La table **Fiches** ne respecte pas non plus les trois formes normales. La table ne respecte pas la première forme normale puisqu'elle ne possède pas de clé primaire, qui est une des conditions de la première forme, afin de pouvoir identifier de manière unique chaque enregistrement. La table ne respecte pas non plus la troisième forme normale. C'est-à-dire que les colonnes qui contiennent une date possède une dépendance indirecte, il est préférable de les placer dans une autre table avec d'éviter toute dépendance indirecte. La table possède également des clés étrangères avec des relations directes sans passer des tables intermédiaires.

La table **GrilleTarifs** ne respecte pas les trois formes normales, car la table ne possède pas de clé primaire, elle ne respecte pas la première forme normale. Elle ne respecte pas non plus la troisième forme normale parce que la table possède deux clés étrangères avec des liaisons directes sans passer par des tables de correspondances.

Exercice 3

Dans la table **Clients**, la contrainte d'intégrité est la colonne **NoCli**, utilisée comme clé primaire. Il n'y a pas d'autre contrainte nécessaire.

Dans la table **Fiches**, les contraintes d'intégrités sont :

- La colonne **NoCli** est une clé étrangère qui doit se référer à la table **Clients**.
- La colonne **RefArt** est une clé étrangère qui doit se référer à la table **Articles**.
- Les données dans la colonne **Retour** ne doit pas être supérieur à la donnée contenu dans la colonne **Depart**.

Dans la table **Article**, les contraintes d'intégrités sont :

- La colonne **RefArt**, utilisée comme clé primaire.
- La colonne **CodeGam** est une clé étrangère qui doit se référer à la table **Gammes**.
- La colonne **CodeCat** est une clé étrangère qui doit se référer à la table **Catégories**.

Dans la table **GrilleTarifs**, les contraintes d'intégrités sont :

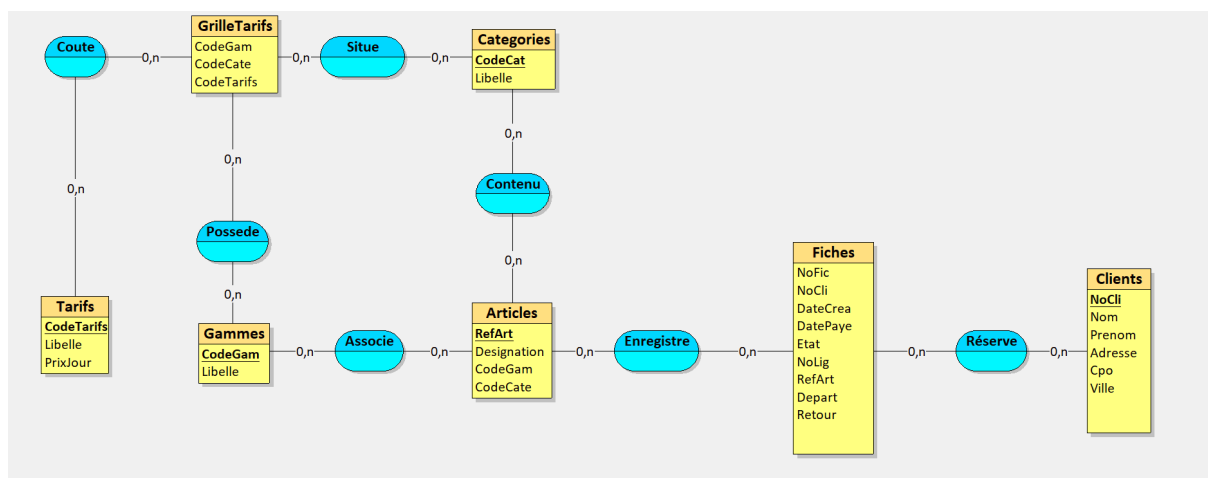
- La colonne **CodeCat** est une clé étrangère qui doit se référer à la table **Catégories**.
- La colonne **CodeTarif** est une clé étrangère qui doit se référer à la table **Tarifs**.

Dans la table **Gammes**, la contrainte d'intégrité est la colonne **CodeGam**, utilisée comme clé primaire.

Dans la table **Categories**, la contrainte d'intégrité est la colonne **CodeCat**, utilisée comme clé primaire.

Dans la table **Tarifs**, la contrainte d'intégrité est la colonne **CodeTarif**, utilisée comme clé primaire.

Schéma de la base de donnée :



Exercice 4

Le dictionnaire ci-dessous présente une description détaillée des tables, attributs, types de données, clés primaires et étrangères ainsi que les contraintes de nullité pour chaque entité.

Attribut	Table	Type de données	Clé	Nullité
NoCli	Clients	INT	PK	Not Null
Nom	Clients	VARCHAR(50)		Not Null
Prenom	Clients	VARCHAR(50)		Not Null
Adresse	Clients	VARCHAR(300)		Not Null
Cpo	Clients	INT		Not Null
Ville	Clients	VARCHAR(100)		Not Null
CodeGam	Gammes	VARCHAR(10)	PK	Not Null
Libelle	Gammes	VARCHAR(100)		Not Null
CodeTarifs	Tarifs	VARCHAR(10)	PK	Not Null
Libelle	Tarifs	VARCHAR(100)		Not Null
PrixJour	Tarifs	INT		Not Null
CodeCate	Categories	VARCHAR(10)	PK	Not Null
Libelle	Categories	VARCHAR(100)		Not Null
CodeGam	GrilleTarifs	VARCHAR(10)	FK	Not Null
CodeCate	GrilleTarifs	VARCHAR(10)	FK	Not Null
CodeTarif	GrilleTarifs	VARCHAR(10)	FK	Not Null
RefArt	Articles	VARCHAR(10)	PK	Not Null
Designation	Articles	VARCHAR(100)		Not Null
CodeGam	Articles	VARCHAR(10)	FK	Not Null
CodeCat	Articles	VARCHAR(10)	FK	Not Null
NoFic	Fiches	INT		Not Null
NoCli	Fiches	INT	FK	Not Null
DateCrea	Fiches	VARCHAR(100)		Not Null

DatePaye	Fiches	VARCHAR(100)		
Etat	Fiches	VARCHAR(100)		Not Null
NoLig	Fiches	INT		Not Null
RefArt	Fiches	VARCHAR(10)	FK	Not Null
Depart	Fiches	VARCHAR(100)		
Retour	Fiches	VARCHAR(100)		

Légende :

- 'PK' : Primary Key.
- 'FK' Foreign Key.

Exercice 5 :

1. La liste des clients dont le nom commence par un 'D'.

```
SELECT *  
FROM Clients  
WHERE nom LIKE 'D%';
```

NoCli	Nom	Prenom	Adresse	Cpo	Ville
3	Dupond	Camille	Rue Crébillon	44000	Nantes
4	Desmoulins	Daniel	Rue descendante	21000	Dijon
9	Dupond	Jean	Rue des mimosas	75018	Paris

2. Les noms et prénoms de tous les clients.

```
SELECT nom, prenom  
FROM Clients;
```

nom	prenom
Albert	Anatole
Bernard	Barnabé
Dupond	Camille
Desmoulins	Daniel
Ernest	Etienne
Ferdinand	François
Dupond	Jean
Boutaud	Sabine

3. La liste des fiches pour les clients qui habitent en Loire-Atlantique.

```
SELECT C.nom, C.prenom, F.NoFic, F.Etat
FROM Clients AS C
INNER JOIN Fiches AS F ON C.NoCli = F.NoCli
WHERE C.Cpo = 44000;
```

nom	prenom	NoFic	Etat
Dupond	Camille	1005	EC
Dupond	Camille	1005	EC

4. Le détail de la fiche n°1002.

```
SELECT *
FROM Fiches
WHERE NoFic = 1002;
```

NoFic	NoCli	DateCrea	DatePaye	Etat	NoLig	RefArt	Depart	Retour
1002	4	DDJ-13	NULL	EC	1	A03	DDJ-13	DDJ-9
1002	4	DDJ-13	NULL	EC	2	A04	DDJ-12	DDJ-7
1002	4	DDJ-13	NULL	EC	3	S03	DDJ-8	NULL

5. Le prix journalier moyen de location par gamme.

```
SELECT G.Libelle AS Gamme, AVG(T.PrixJour) AS TarifMoyen
FROM GrilleTarifs AS GT
INNER JOIN Tarifs AS T ON GT.CodeTarif = T.CodeTarif
INNER JOIN Gammes AS G ON GT.CodeGam = G.CodeGam
GROUP BY G.Libelle;
```

Gamme	TarifMoyen
Entrée de gamme	10.8333
Moyenne gamme	15.8333
Haut de gamme	27.5000
Matériel Professionnel	80.0000

6. Les articles qui ont été loués au moins trois fois.

```
SELECT F.RefArt, A.Designation, COUNT(F.RefArt) AS NombreDeLocations
FROM Fiches AS F
INNER JOIN Articles AS A ON F.RefArt = A.RefArt
GROUP BY F.RefArt, A.Designation
HAVING COUNT(F.RefArt) >= 3;
```

RefArt	Designation	NombreDeLocations
F05	Fischer Cruiser	4
F50	Fischer SOSSkating VASA	3

7. Le détail de la fiche, n°1002 avec le total.

```
SELECT C.nom, C.prenom, F.RefArt, A.Designation, F.Depart, F.Retour,
T.PrixJour, SUM(T.PrixJour) OVER() AS Total
FROM Fiches AS F
INNER JOIN Clients AS C ON F.NoCli = C.noCli
INNER JOIN Articles AS A ON F.RefArt = A.RefArt
INNER JOIN GrilleTarifs AS GT ON A.CodeGam = GT.CodeGam AND A.CodeCate
= GT.CodeCate
INNER JOIN Tarifs AS T ON GT.CodeTarif = T.CodeTarif
WHERE F.NoFic = 1002;
```

nom	prenom	RefArt	Designation	Depart	Retour	PrixJour	Total
Desmoulins	Daniel	S03	Décathlon Apparition	DDJ-8	NULL	10	30
Desmoulins	Daniel	A04	Salomon 24X+Z12	DDJ-12	DDJ-7	10	30
Desmoulins	Daniel	A03	Salomon 24X+Z12	DDJ-13	DDJ-9	10	30

Partie III : Environnement client serveur, procédures, triggers et transactions.

Exercice 1

1. Création de la table FILM.

```
CREATE TABLE Film (  
    Ident_Film INT PRIMARY KEY AUTO_INCREMENT,  
    Titre VARCHAR(100) NOT NULL,  
    Genre1 VARCHAR(50) NOT NULL,  
    Genre2 VARCHAR(50),  
    Date_Sortie DATE NOT NULL,  
    Pays INT NOT NULL,  
    Ident_Realisateur INT NOT NULL,  
    Distributeur VARCHAR(50) NOT NULL,  
    Resume TEXT NOT NULL,  
    FOREIGN KEY (Pays) REFERENCES Pays(Ident_Pays),  
    FOREIGN KEY (Ident_Realisateur) REFERENCES Realisateur  
(Ident_Realisateur)  
);
```

```
INSERT INTO Film (Titre, Genre1, Genre2, Date_Sortie, Pays,  
Ident_Realisateur, Distributeur, Resume)  
VALUES  
(  
'SUBWAY', 'POLICIER', 'DRAME', '1985-04-10', 1, 1, 'GAUMONT', 'Conte  
les aventures de la population souterraine dans les couloirs du métro  
parisien'),  
(  
'NIKITA', 'DRAME', 'ROMANTIQUE', '1990-02-21', 1, 1, 'GAUMONT',  
'Nikita condamnée à la prison à perpétuité est contrainte de travailler  
secrètement pour le gouvernement en tant que agent hautement qualifié  
des services secrets.'),  
(  
'STAR WARS 6- LE RETOUR DU JEDI', 'ACTION', 'SF', '1983-10-19', 2, 2,  
'20th Century Fox', 'L''Empire galactique est plus puissant que jamais:  
la construction de la nouvelle arme, l''Etoile de la Mort, menace  
l''univers tout entier.'),  
(  
'AVATAR', 'ACTION', 'SF', '2009-10-16', 2, 3, '20th Century Fox',  
'Malgré sa paralysie, Jake Sully, un ancien marine immobilisé dans un  
fauteuil roulant, est resté un combattant au plus profond'),  
(  
'BIENVENUE CHEZ LES CH''TIS', 'COMEDIE', NULL, '2008-02-27', 1, 4,  
'PATHE', 'Philippe Abrams est directeur de la poste de  
Salon-de-Provence est muté dans le Nord.');
```

2. Création de la table ACTEUR.

```
CREATE TABLE Acteur (  
    Ident_Acteur INT PRIMARY KEY AUTO_INCREMENT,  
    Nom VARCHAR(50) NOT NULL,  
    Prenom VARCHAR(50) NOT NULL,  
    Date_Naissance DATE NOT NULL,  
    Nb_Film INT NOT NULL,  
    Nationalite INT NOT NULL,  
    FOREIGN KEY (Nationalite) REFERENCES Pays(Ident_Film)  
);
```

```
INSERT INTO Acteur (Nom, Prenom, Date_Naissance, Nb_Film, Nationalite)  
VALUES  
( 'ADJANI', 'ISABELLE', '1955-06-27', 42, 1),  
( 'LAMBERT', 'CHRISTOPHE', '1957-03-29', 64, 1),  
( 'BOHRINGER', 'RICHARD', '1942-06-16', 132, 1),  
( 'GALABRU', 'MICHEL', '1922-10-27', 277, 1),  
( 'PARILLAUD', 'ANNE', '1960-05-06', 35, 1),  
( 'FORD', 'HARRISON', '1942-06-13', 64, 2),  
( 'FISHER', 'CARRIE', '1956-10-21', 74, 2),  
( 'SALDANA', 'ZOE', '1978-06-19', 31, 2),  
( 'WEAVER', 'SIGOURNEY', '1949-10-08', 66, 2),  
( 'RENO', 'JEAN', '1948-06-30', 75, 1),  
( 'BOON', 'DANY', '1966-06-26', 23, 1),  
( 'MERAD', 'KAD', '1964-03-27', 55, 3);
```

3. Création de la table STATISTIQUE.

```
CREATE TABLE Statistique (  
    Ident_Film INT PRIMARY KEY NOT NULL,  
    Duree INT NOT NULL,  
    Nb_Entree_France INT NOT NULL,  
    Recette_USA INT NOT NULL,  
    Recette_Monde INT NOT NULL,  
    Budget DECIMAL(6,2) NOT NULL,  
    FOREIGN KEY (Ident_Film) REFERENCES Film(Ident_Film)  
);
```

```
INSERT INTO Statistique (Ident_Film, Duree, Nb_Entree_France,  
Recette_USA, Recette_Monde, Budget)  
VALUES  
(1, 104, 2917562, 390659, 1272637, 2.6),  
(2, 118, 3787845, 5017971, 0, 7.6),  
(3, 133, 4263000, 191648000, 472000000, 32),  
(4, 170, 12018251, 760505847, 2946271769, 237),  
(5, 100, 21000000, 0, 245000000, 11);
```

Exercice 2 :

1. Création de la table REALISATEUR.

```
CREATE TABLE Realisateur (  
    Ident_Realisateur INT PRIMARY KEY AUTO_INCREMENT,  
    Nom VARCHAR(50),  
    Prenom VARCHAR(50),  
    Date_Naissance DATE NOT NULL,  
    Nb_Film_Ecrit INT,  
    Nb_Film_Produit INT,  
    Nationalite INT NOT NULL,  
    FOREIGN KEY (Nationalite) REFERENCES pays(Ident_Pays)  
);
```

```
INSERT INTO Realisateur (Nom, Prenom, Date_Naissance, Nb_Film_Ecrit,  
Nb_Film_Produit, Nationalite)  
VALUES  
( 'BESSON', 'LUC', '1959-03-18', 40, 99, 1),  
( 'LUCAS', 'GEORGES', '1944-05-14', 79, 64, 2),  
( 'CAMERON', 'JAMES', '1954-08-16', 22, 23, 2),  
( 'BOON', 'DANY', '1966-06-26', 5, 1, 1);
```

2. Création de la table CASTING.

```
CREATE TABLE Casting (  
    Ident_Film INT NOT NULL,  
    Ident_Acteur INT NOT NULL,  
    Role VARCHAR(50) NOT NULL,  
    Nb_Jour_Tournage INT,  
    FOREIGN KEY (Ident_Film) REFERENCES Film(Ident_Film),  
    FOREIGN KEY (Ident_Acteur) REFERENCES Acteur(Ident_Acteur)  
);
```

```
INSERT INTO Casting (Ident_Film, Ident_Acteur, Role, Nb_Jour_Tournage)  
VALUES  
(1, 1, 'HELENA', 100),  
(1, 2, 'FRED', 100),  
(1, 3, 'INSPECTEUR GESBERG', NULL),  
(1, 4, 'LE FLEURISTE', 35),  
(1, 10, 'LE BATTEUR', 20),  
(2, 5, 'NIKITA', 68),  
(2, 10, 'VICTOR LE NETTOYEUR', 9),  
(3, 6, 'HAN SOLO', 201),  
(3, 7, 'PRINCESSE LEIA', 203),  
(4, 8, 'NEYTIRI', 50),  
(4, 9, 'Dr. Grace Augustine', 45),  
(5, 11, 'ANTOINE BAILLEUL', 125),  
(5, 12, 'PHILIPPE ABRAMS', 126);
```

3. Création de la table PAYS.

```
CREATE TABLE Pays (  
    Ident_Pays INT PRIMARY KEY NOT NULL,  
    Libelle VARCHAR(100)  
);
```

```
INSERT INTO Pays (Ident_Pays, Libelle)  
VALUES  
(1, 'FRANCE'),  
(2, 'USA'),  
(3, 'ALGERIE');
```

Exercice 3

1. Sélectionner toutes les informations sur les films réalisés par un réalisateur français triés par le nom du film.

```
SELECT F.Titre, F.Genre1, F.Genre2, F.Date_Sortie, F.Distributeur,
F.Resume, R.Nom AS Nom_Réalisateur, R.Prenom AS Prenom_Réalisateur,
P.Libelle AS Nationalite_Réalisateur
FROM Film AS F
INNER JOIN Realisateur AS R ON F.Ident_Réalisateur =
R.Ident_Réalisateur
INNER JOIN Pays AS P ON R.Nationalite = P.Ident_Pays
WHERE P.Libelle = 'FRANCE'
ORDER BY F.Titre;
```

Titre	Genre1	Genre2	Date_Sortie	Distributeur	Resume	Nom_Réalisateur	Prenom_Réalisateur	Nationalite_Réalisateur
BIENVENUE CHEZ LES CHTIS	COMEDIE	NULL	2008-02-27	PATHE	Philippe Abrams est directeur de la poste de Salon...	BOON	DANY	FRANCE
NIKITA	DRAME	ROMANTIQUE	1990-02-21	GAUMONT	Nikita condamnée à la prison à perpétuité est cont...	BESSON	LUC	FRANCE
SUBWAY	POLICIER	DRAME	1985-04-10	GAUMONT	Conte les aventures de la population souterraine d...	BESSON	LUC	FRANCE

- Sélectionner le nom du film, la date de sortie, le nom du réalisateur, le nom des acteurs, leur date de naissance, ainsi que le budget du film. Le tout trié en ordre descendant par titre du film et nom des acteurs.

```
SELECT F.Titre AS Nom_film, F.Date_Sortie, R.Nom AS Nom_Realisateur,
A.Nom AS Nom_Acteur, A.Date_Naissance, S.Budget
FROM Film AS F
INNER JOIN Realisateur AS R ON F.Ident_Realisateur =
R.Ident_Realisateur
INNER JOIN Casting AS C ON F.Ident_Film = C.Ident_Film
INNER JOIN Acteur AS A ON C.Ident_Acteur = A.Ident_Acteur
INNER JOIN Statistique AS S ON F.Ident_Film = S.Ident_Film
ORDER BY F.Titre DESC, A.Nom DESC;
```

Nom_film	Date_Sortie	Nom_Realisateur	Nom_Acteur	Date_Naissance	Budget
SUBWAY	1985-04-10	BESSON	RENO	1948-06-30	2.60
SUBWAY	1985-04-10	BESSON	LAMBERT	1957-03-29	2.60
SUBWAY	1985-04-10	BESSON	GALABRU	1922-10-27	2.60
SUBWAY	1985-04-10	BESSON	BOHRINGER	1942-06-16	2.60
SUBWAY	1985-04-10	BESSON	ADJANI	1955-06-27	2.60
STAR WARS 6- LE RETOUR DU JEDI	1983-10-19	LUCAS	FORD	1942-06-13	32.00
STAR WARS 6- LE RETOUR DU JEDI	1983-10-19	LUCAS	FISHER	1956-10-21	32.00
NIKITA	1990-02-21	BESSON	RENO	1948-06-30	7.60
NIKITA	1990-02-21	BESSON	PARILLAUD	1960-05-06	7.60
BIENVENUE CHEZ LES CH'TIS	2008-02-27	BOON	MERAD	1964-03-27	11.00
BIENVENUE CHEZ LES CH'TIS	2008-02-27	BOON	BOON	1966-06-26	11.00
AVATAR	2009-10-16	CAMERON	WEAVER	1949-10-08	237.00
AVATAR	2009-10-16	CAMERON	SALDANA	1978-06-19	237.00

3. Trouver le nombre d'acteurs par film dans la base de données. Afficher le titre, la date de sortie, le nom du réalisateur et le distributeur.

```
SELECT F.Titre AS Nom_film, F.Date_Sortie, R.Nom AS Nom_Réalisateur,
F.Distributeur, COUNT(C.Ident_Acteur) AS Nombre_Acteurs
FROM Film AS F
JOIN Realisateur AS R ON F.Ident_Réalisateur = R.Ident_Réalisateur
JOIN Casting AS C ON F.Ident_Film = C.Ident_Film
GROUP BY F.Titre, F.Date_Sortie, R.Nom, F.Distributeur
ORDER BY F.Titre;
```

Nom_film	Date_Sortie	Nom_Réalisateur	Distributeur	Nombre_Acteurs
AVATAR	2009-10-16	CAMERON	20th Century Fox	2
BIENVENUE CHEZ LES CH'TIS	2008-02-27	BOON	PATHE	2
NIKITA	1990-02-21	BESSON	GAUMONT	2
STAR WARS 6- LE RETOUR DU JEDI	1983-10-19	LUCAS	20th Century Fox	2
SUBWAY	1985-04-10	BESSON	GAUMONT	5

4. Sélectionner le titre du film, la date de sortie, le nom et prénom du réalisateur, le nom et prénom de l'acteur, sa date de naissance, le budget du film et le nombre d'entrées en France des films qui ont un acteur algérien.

```
SELECT F.Titre AS Nom_film, F.Date_Sortie, R.Nom AS Nom_Réalisateur,
R.Prenom AS Prenom_Réalisateur, A.Nom AS Nom_Acteur, A.Prenom AS
Prenom_Acteur, A.Date_Naissance, S.Budget, S.Nb_Entree_France
FROM Film AS F
INNER JOIN Realisateur AS R ON F.Ident_Réalisateur =
R.Ident_Réalisateur
INNER JOIN Casting AS C ON F.Ident_Film = C.Ident_Film
INNER JOIN Acteur AS A ON C.Ident_Acteur = A.Ident_Acteur
INNER JOIN Statistique AS S ON F.Ident_Film = S.Ident_Film
INNER JOIN Pays AS P ON A.Nationalite = P.Ident_Pays
WHERE P.Libelle = 'ALGERIE'
ORDER BY F.Titre;
```

Nom_film	Date_Sortie	Nom_Réalisateur	Prenom_Réalisateur	Nom_Acteur	Prenom_Acteur	Date_Naissance	Budget	Nb_Entree_France
BIENVENUE CHEZ LES CH'TIS	2008-02-27	BOON	DANY	MERAD	KAD	1964-03-27	11.00	21000000

5. Sélectionner le film qui a réalisé la recette la plus élevée dans le monde.

```
SELECT F.Titre AS Nom_film, S.Recette_Monde
FROM Film AS F
INNER JOIN Statistique AS S ON F.Ident_Film = S.Ident_Film
ORDER BY S.Recette_Monde DESC
LIMIT 1;
```

Nom_film	Recette_Monde
AVATAR	2147483647

6. Sélectionner l'acteur qui a joué dans deux films différents.

```
SELECT A.Nom AS Nom_Acteur, A.Prenom AS Prenom_Acteur,
COUNT(C.Ident_Film) AS Nombre_Films
FROM Acteur AS A
INNER JOIN Casting AS C ON A.Ident_Acteur = C.Ident_Acteur
GROUP BY A.Ident_Acteur, A.Nom, A.Prenom
HAVING COUNT(C.Ident_Film) >= 2;
```

Nom_Acteur	Prenom_Acteur	Nombre_Films
RENO	JEAN	2

7. Sélectionner la personne qui est à la fois réalisateur et acteur.

```
SELECT DISTINCT R.Nom AS Nom_Realisateur, R.Prenom AS
Prenom_Realisateur
FROM Realisateur AS R
INNER JOIN Acteur AS A ON R.Nom = A.Nom AND R.Prenom = A.Prenom;
```

Nom_Realisateur	Prenom_Realisateur
BOON	DANY

8. Sélectionner les acteurs qui ont joué dans des films dont le nom du film commence par la lettre S. Indiquer leur rôle et leur nationalité.

```
SELECT A.Nom AS Nom_Acteur, A.Prenom AS Prenom_Acteur, F.Titre AS
Titre_Film, C.Role, P.Libelle AS Nationalite
FROM Acteur AS A
INNER JOIN Casting AS C ON A.Ident_Acteur = C.Ident_Acteur
INNER JOIN Film AS F ON C.Ident_Film = F.Ident_Film
INNER JOIN Pays AS P ON A.Nationalite = P.Ident_Pays
WHERE F.Titre LIKE 'S%'
ORDER BY F.Titre;
```

Nom_Acteur	Prenom_Acteur	Titre_Film	Role	Nationalite
FISHER	CARRIE	STAR WARS 6- LE RETOUR DU JEDI	PRINCESSE LEIA	USA
FORD	HARRISON	STAR WARS 6- LE RETOUR DU JEDI	HAN SOLO	USA
RENO	JEAN	SUBWAY	LE BATTEUR	FRANCE
GALABRU	MICHEL	SUBWAY	LE FLEURISTE	FRANCE
BOHRINGER	RICHARD	SUBWAY	INSPECTEUR GESBERG	FRANCE
LAMBERT	CHRISTOPHE	SUBWAY	FRED	FRANCE
ADJANI	ISABELLE	SUBWAY	HELENA	FRANCE

9. Sélectionner les acteurs qui sont nés entre 1948 et mai 1978 ainsi que le nombre de jours total de tournage qu'ils ont réalisés.

```
SELECT A.Nom AS Nom_Acteur, A.Prenom AS Prenom_Acteur,  
A.Date_Naissance, SUM(C.Nb_Jour_Tournage) AS Total_Jours_Tournage  
FROM Acteur AS A  
INNER JOIN Casting AS C ON A.Ident_Acteur = C.Ident_Acteur  
WHERE A.Date_Naissance BETWEEN '1948-01-01' AND '1978-05-31'  
GROUP BY A.Nom, A.Prenom, A.Date_Naissance  
ORDER BY Total_Jours_Tournage DESC;
```

Nom_Acteur	Prenom_Acteur	Date_Naissance	Total_Jours_Tournage ▾ 1
FISHER	CARRIE	1956-10-21	203
MERAD	KAD	1964-03-27	126
BOON	DANY	1966-06-26	125
ADJANI	ISABELLE	1955-06-27	100
LAMBERT	CHRISTOPHE	1957-03-29	100
PARILLAUD	ANNE	1960-05-06	68
WEAVER	SIGOURNEY	1949-10-08	45
RENO	JEAN	1948-06-30	29

Partie IV : Liens Github.

Ci-dessous se trouve le lien vers le dépôt GitHub contenant l'ensemble des fichiers relatifs au compte rendu. Ce dépôt comprend tous les documents, fichier SQL et autres ressources nécessaires pour une référence complète du rapport. Vous y trouverez l'intégralité des éléments constituant le rapport, permettant ainsi une consultation complète de tous les aspects abordés dans le compte rendu. Pour accéder au dépôt, veuillez cliquer sur le lien suivant : https://github.com/WillSmith351/EPSIB3_AtlEnvClient-Serveur