RAPPORT BDD

SOMMAIRE

Table des matières

[INTRODUCTION 3](#_Toc145521691)

[CREATION DE LA BASE DE DONNEES 4](#_Toc145521692)

[INSERTION 5](#_Toc145521693)

[REQUÊTES 6](#_Toc145521694)

[**1.** **\* Retrouver la liste de tous les films.** 6](#_Toc145521695)

[**2.** **\* Donner la liste de tous les genres de films.** 6](#_Toc145521696)

[**3.** **Pour chaque film de Spielberg (titre, année), donner le total des salaires des acteurs.** 6](#_Toc145521697)

[**4.** **\* Trouver le titre et l’année des films de science-fiction dont le budget dépasse $5.000.000.** 6](#_Toc145521698)

[**5.** **Donner le nombre de films par genre.** 6](#_Toc145521699)

[**6.** **\* Trouver le titre des films réalisés par Georges Lucas.** 6](#_Toc145521700)

[**7.** **\* Trouver le titre des films qui ne passent dans aucun cinéma de la compagnie FOX.** 6](#_Toc145521701)

[**8.** **\* Quels sont les acteurs comiques (nom, prénom) qui ont joué dans un film de Spielberg.** 6](#_Toc145521702)

[**9.** **Trouver le titre et l’année du film le plus long.** 6](#_Toc145521703)

[**10.** **\* Donner le nom et prénom des acteurs qui ont joué Gavroche dans les différentes versions des ”Misérables” avec les dates correspondantes.** 6](#_Toc145521704)

[**11.** **Donner le titre des films qui ont reçu au moins trois récompenses.** 6](#_Toc145521705)

[**12.** **\* Retrouver la liste des films dont la longueur dépasse 180 minutes.** 6](#_Toc145521706)

[**13.** **\* Donner le nom et le prénom des réalisateurs qui ont joué dans au moins un de leurs propres films.** 6](#_Toc145521707)

[**14.** **Lister les cinémas qui ont exclusivement passé des films primés.** 6](#_Toc145521708)

[**15.** **Quel est le total des salaires des acteurs du film ”La guerre des étoiles” ?** 6](#_Toc145521709)

[**16.** **Donner la moyenne des salaires des acteurs par film, avec le titre et l’année correspondant.** 6](#_Toc145521710)

[**17.** **Trouver le genre des films des années 90 dont le budget moyen dépasse $200.000.** 6](#_Toc145521711)

[**18.** **Lister les cinémas dont la taille moyenne de l’écran est supérieure à 40 mètres carrés.** 6](#_Toc145521712)

[**19.** **Donner le nom des cinémas passant tous les films primés cette année au festival de Cannes.** 6](#_Toc145521713)

[**20.** **\* Quels sont les cinémas new-yorkais de la Fox qui passent un film de Peter Jackson avant 22 heures dans une salle d’au moins 200 places et un écran de taille supérieure à 30 mètres carrés (donner aussi le nom des films correspondant).** 6](#_Toc145521714)

[**21.** **Donner le nombre de films sortis en 1980 par genre.** 7](#_Toc145521715)

[**22.** **\* Trouver le nom et le prénom des acteurs qui, pour un film donné, ont eu un salaire plus important que le salaire du réalisateur (attention un acteur peut jouer plusieurs rôles).** 7](#_Toc145521716)

[**23.** **\* Trouver les couples acteur-réalisateur (noms et prénoms) tels que l’un a dirigé l’autre sur un film et vice-versa sur un autre.** 7](#_Toc145521717)

[**24.** **Trouver le nom, le prénom, le numéro des acteurs qui ont joué dans tous les films de Luc Besson.** 7](#_Toc145521718)

[**25.** **Pour chaque film de Peter Jackson, trouver le nom et le prénom de l’acteur qui a eu le plus gros salaire.** 7](#_Toc145521719)

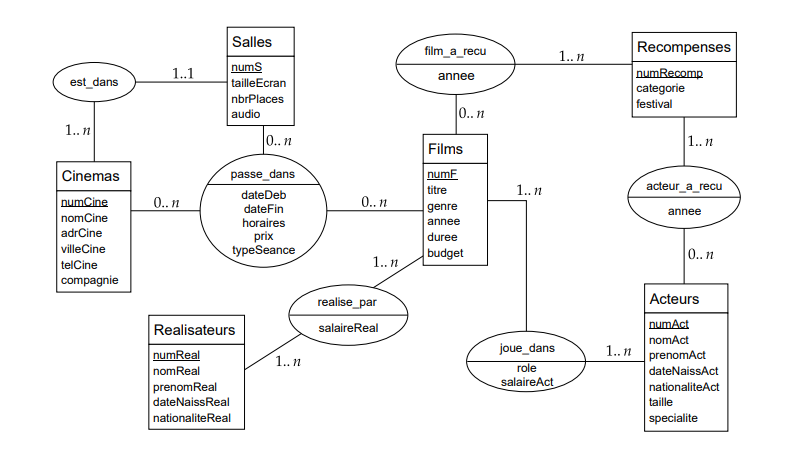
[**26.** **Donner le titre des films qui ont été primés au moins une fois (en comptant aussi les récompenses des acteurs jouant dans le film).** 7](#_Toc145521720)

[**27.** **Donner le prix moyen des films passant dans des salles en 3D (cf. typeSeance dans l’association passe dans).** 7](#_Toc145521721)

INTRODUCTION

Le but de ce TP est de nous familiariser avec la création d’une base de donnée à partir d’un schéma, d’insérer des données dans celui-ci et d’effectuer des requêtes.

Le premier objectif est d’établir une base de donnée nommée CINEMAS avec le schéma relationnel suivant :



Le deuxième objectif est de remplir cette base de données en utilisant l’instruction INSERT INTO.

Enfin, il faut interroger et modifier la base de données avec des requêtes SQL, modifier les données pour les vérifier deux fois avec les instructions UPDATE et/ou DELETE et faire des requêtes algébriques pour certaines requêtes indiquées.

CREATION DE LA BASE DE DONNEES

Pour créer notre base de données, nous avons traduit le schéma relation selon les 3 règles :

Règle 1, Tout type d'entité devient une relation. Ses propriétés deviennent des attributs. L’identifiant du type d’entité devient la clé de la relation.

Films(numF, titre, genre, annee, duree, budget)

Recompenses(numRecomp, categorie, festival)

Acteurs(numAct, nomAct, prenomAct, dateNaissAct, nationaliteAct, taille, specialite)

Realisateurs(numReal, nomReal, prenomReal, dateNaissReal, nationaliteReal)

Cinemas(numCine, nomCine, adrCine, villeCine, telCine, compagnie)

Salles(numS, tailleEcran, nbrPlaces, audio)

Règle 2, Pour toute association de cardinalité 1 d’un côté, par exemple celui de E1, on ajoute aux attributs de E2 : l’identifiant de E1 (appelé clé étrangère) ainsi que les propriétés de l'association.

Films(numF, titre, genre, annee, duree, budget)

Recompenses(numRecomp, categorie, festival)

Acteurs(numAct, nomAct, prenomAct, dateNaissAct, nationaliteAct, taille, specialite)

Realisateurs(numReal, nomReal, prenomReal, dateNaissReal, nationaliteReal)

Cinemas(numCine, nomCine, adrCine, villeCine, telCine, compagnie)

Salles(numS, tailleEcran, nbrPlaces, audio, #numCine)

Règle 3, Toute association entre deux types d'entités E1 et E2, de cardinalité 0..\* ou 1..\* des deux côtés de l’association devient une relation R. Sa clé est formée à la fois de l’identifiant de E1 et de celui de E2. Les propriétés de l’association deviennent des attributs de la relation R.

Films(numF, titre, genre, annee, duree, budget)

Recompenses(numRecomp, categorie, festival)

Acteurs(numAct, nomAct, prenomAct, dateNaissAct, nationaliteAct, taille, specialite)

Realisateurs(numReal, nomReal, prenomReal, dateNaissReal, nationaliteReal)

Cinemas(numCine, nomCine, adrCine, villeCine, telCine, compagnie)

Salles(numS, tailleEcran, nbrPlaces, audio, #numCine)

Joue\_dans(numF, numAct, rôle, salaireAct)

Film\_a\_recu(numF, numRecomp, annee)

Acteur\_a\_recu(numRecomp, numAct, annee)

Realise\_par(numF, numReal, salaireReal)

Passe\_dans(numF, numS, numCine, dateDeb, dateFin, horaires, prix, typeSceance)

Ce qui nous donne la création de la base de données suivante :

/\* creation du schema \*/

drop database if exists `cinemas`;

create database cinemas;

/\* créations des tables \*/

create table `cinemas`.films

(

    numF int PRIMARY KEY,

    titre varchar(80),

    genre varchar(20),

    annee year,

    duree int,

    budget int

);

use `cinemas`;

create table recompenses

(

    numRecomp int PRIMARY KEY,

    categorie varchar(50),

    festival varchar(50)

);

create table acteurs

(

    numAct int,

    nomAct varchar(30),

    prenomAct varchar(30),

    dateNaissAct date,

    nationaliteAct varchar(30),

    taille int,

    specialite varchar(50),

    primary key(numAct)

);

create table realisateurs

(

    numReal int,

    nomReal varchar(30),

    prenomReal varchar(30),

    dateNaissReal date,

    nationaliteReal varchar(30),

    primary key(numReal)

);

create table cinemas

(

    numCine int,

    nomCine varchar(30),

    adrCine varchar(30),

    villeCine varchar(300),

    telCine varchar(20),

    compagnie varchar(30),

    primary key(numCine)

);

create table salles

(

    numS int,

    tailleEcran int,

    nbrPlaces int,

    audio varchar(15),

    primary key(numS),

    numCine int,

    foreign key(numCine) references cinemas(numCine)

);

create table joue\_dans

(

    numF int,

    numAct int,

    rôle varchar(30),

    salaireAct int,

    primary key(numF,numAct),

    foreign key(numF) references films(numF),

    foreign key(numAct) references acteurs(numAct)

);

create table film\_a\_recu

(

    numF int,

    numRecomp int,

    annee varchar(4),

    primary key(numF, numRecomp),

    foreign key(numF) references films(numF),

    foreign key(numRecomp) references recompenses(numRecomp)

);

create table acteur\_a\_recu

(

    numRecomp int,

    numAct int,

    annee year,

    primary key(numRecomp,numAct),

    foreign key(numRecomp) references recompenses(numRecomp),

    foreign key(numAct) references acteurs(numAct)

);

create table realise\_par

(

    numF int,

    numReal int,

    salaireReal int,

    primary key(numF,numReal),

    foreign key(numF) references films(numF),

    foreign key(numReal) references realisateurs(numReal)

);

create table passe\_dans

(

    numF int,

    numS int,

    numCine int,

    dateDeb date,

    dateFin date,

    horaires time,

    prix int,

    typeSceance varchar(2),

    primary key(numF,numS,numCine)

);

INSERTION

Voici les valeurs que nous avons mis dans la base de données :

INSERT INTO Films (numF, titre, genre, annee, duree, budget)

VALUES

  (1, 'Inception', 'Science-Fiction', 2010, 148, 160000000),

  (2, 'The Shawshank Redemption', 'Drama', 1994, 142, 25000000),

  (3, 'Avatar', 'Action', 2009, 162, 237000000),

  (4, 'Pulp Fiction', 'Crime', 1994, 154, 8000000),

  (5, 'Jurassic Park', 'Adventure', 1993, 127, 63000000),

  (6, 'La guerre des étoiles : Un nouvel espoir', 'Science-Fiction', 1977, 121, 11000000),

  (7, 'Les Misérables', 'Drama', 1997, 185, 1200000),

  (8, 'La guerre des étoiles : L empire contre-attaque', 'Science-Fiction', 1980, 124, 18000000),

  (9, 'Le Seigneur des anneaux : La Communauté de l anneau', 'Fantasy', 2001, 178, 93000000),

  (10, 'Le Seigneur des anneaux : Les Deux Tours', 'Fantasy', 2002, 179, 94000000),

  (11, 'Le Seigneur des anneaux : Le Retour du roi', 'Fantasy', 2003, 201, 94000000),

  (12, 'Barbie', 'Aventure', 2023, 115, 145000000),

  (13, 'Oppenheimer', 'Biographie', 2023, 180, 100000000),

  (14, 'Le Cinquième Élément', 'Science-Fiction', 1997, 126, 93000000);

INSERT INTO Recompenses (numRecomp, categorie, festival)

VALUES

  (1, 'Meilleur Film', 'Festival du Cinéma'),

  (2, 'Meilleur Acteur', 'Oscars'),

  (3, 'Meilleur Réalisateur', 'Cannes'),

  (4, 'Meilleur Scénario', 'BAFTA'),

  (5, 'Meilleure Musique Originale', 'Golden Globes'),

  (6, 'Meilleur Film', 'Cannes');

INSERT INTO Acteurs (numAct, nomAct, prenomAct, dateNaissAct, nationaliteAct, taille, specialite)

VALUES

  (1, 'DiCaprio', 'Leonardo', '1974-11-11', 'Américaine', 183, 'Narration'),

  (2, 'Freeman', 'Morgan', '1937-06-01', 'Américaine', 188, 'Narration'),

  (3, 'Weaver', 'Sigourney', '1949-10-08', 'Américaine', 182, 'Science Fiction'),

  (4, 'Travolta', 'John', '1954-02-18', 'Américaine', 188, 'Dancing'),

  (5, 'Neill', 'Sam', '1947-09-14', 'Néo-zélandaise', 183, 'Comique'),

  (6, 'Harrison', 'Ford', '1942-07-13', 'Américaine', 185, 'Aventure'),

  (7, 'Thénardier', 'Sacha', '1981-01-25', 'Française', 175, 'Dramatique'),

  (8, 'Robert', 'Hossein', '1927-12-30', 'Française', 175, 'réalisateur/second rôle'),

  (9, 'Jejouedans', 'Laguerredesatoiles', '1987-12-26', 'Française', 190, 'etrela'),

  (10, 'Gosling', 'Ryan', '1980-11-12', 'Canadienne', 184, 'Acteur Principal'),

  (11, 'Murphy', 'Cillian', '1976-05-25', 'Irlandaise', 175, 'Acteur Principal'),

  (12, 'Willis', 'Bruce', '1955-03-19', 'Américaine', 183, 'Action'),

  (13, 'Wood', 'Elijah', '1985-05-20', 'Américaine', 155, 'Acteur Principal'),

  (14, 'Bloom', 'Orlando', '1840-06-30' , 'Américaine', 180, 'seconde rôle');

INSERT INTO Realisateurs (numReal, nomReal, prenomReal, dateNaissReal, nationaliteReal)

VALUES

  (1, 'Nolan', 'Christopher', '1970-07-30', 'Britannique'),

  (2, 'Darabont', 'Frank', '1959-01-28', 'Américaine'),

  (3, 'Cameron', 'James', '1954-08-16', 'Canadienne'),

  (4, 'Tarantino', 'Quentin', '1963-03-27', 'Américaine'),

  (5, 'Spielberg', 'Steven', '1946-12-18', 'Américaine'),

  (6, 'Lucas', 'George', '1944-05-14', 'Américaine'),

  (7, 'Robert', 'Hossein', '1927-12-30', 'française'),

  (8, 'Jackson', 'Peter', '1961-10-31', 'Néo-Zélandais'),

  (10, 'Nolan', 'Christopher', '1970-07-30', 'Britinique'),

  (11, 'Gerwin', 'Greta', '1983-08-04', 'Américaine'),

  (12, 'Besson', 'Luc', '1959-03-18', 'Française');

INSERT INTO Cinemas (numCine, nomCine, adrCine, villeCine, telCine, compagnie)

VALUES

  (1, 'Netflux', '123 Rue Principale', 'Nouille Orc', '123-456-7890', 'Nootflox'),

  (2, 'Cinémoa', '456 Avenue Centrale', 'Taumatawhakatangihangakoauauotamateaturipukakapiki-maungahoronukupokaiwhenuakitnatahu', '987-654-3210', 'CinéDuNom'),

  (3, 'Cinécity', '789 Rue Étoilée', 'Nouméa', '555-555-5555', 'Cinécity'),

  (4, 'VenezRegarder', '101 Old Street', 'Ouegoa', '111-222-3333', 'Nollywood'),

  (5, 'UnlimitedMovies', '555 Action Avenue', 'Bloxcity', '777-888-9999', 'Fox'),

  (6, 'jepasserien', 'nulpar', 'somewhereovertherainbow', '118712', 'boring\_compagnie'),

  (7, "LeCineDeN-Y", "3rd street of the communist", "New-York", "212-421-3600", "Fox"),

  (8, "PrimedCINECITY", "Primedroad", "Primed-City", "122-378-4890", "RobuxForLife");

INSERT INTO Salles (numS, tailleEcran, nbrPlaces, audio, numCine)

VALUES

  (1, 20, 150, 'Dolby Digital', 1),

  (2, 80, 200, 'DTS', 2),

  (3, 180, 100, 'Stereo', 3),

  (4, 40, 250, 'Dolby Atmos', 4),

  (5, 20, 180, 'Surround', 5);

INSERT INTO Joue\_dans (numF, numAct, rôle, salaireAct)

VALUES

  (1, 1, 'Cobb', 2000000),

  (2, 2, 'Red', 3500000),

  (3, 3, 'Neytiri', 3000000),

  (4, 4, 'Vincent Vega', 1800000),

  (5, 5, 'Dr. Alan Grant', 2200000),

  (6, 6, 'Han Solo', 2500000),

  (7, 7, 'Gavroche', 100000),

  (7, 8, 'figurant', 200000),

  (6, 9, 'ilestlaaumoins', 200000),

  (9, 13, 'Frodon', 200000),

  (10, 13, 'Frodon', 205500),

  (10, 14, 'Legolas', 400000),

  (11, 13, 'Frodon', 500000),

  (12, 10, 'Ken', 2300000),

  (13, 11, 'Oppenheimer', 24000000),

  (14, 12, 'Korben Dallas', 10000000);

INSERT INTO Film\_a\_recu (numF, numRecomp, annee)

VALUES

  (1, 1, 2010),

  (2, 2, 1995),

  (3, 3, 2010),

  (4, 4, 1995),

  (5, 5, 1994),

  (6, 1, 1978),

  (7, 1, 1997),

  (7, 2, 1997),

  (7, 3, 1997),

  (12, 6, 2023);

INSERT INTO Acteur\_a\_recu (numRecomp, numAct, annee)

VALUES

  (1, 1, 2010),

  (2, 2, 1995),

  (3, 3, 2010),

  (4, 4, 1995),

  (5, 5, 1994),

  (1, 6, 1980),

  (1, 7, 2023);

INSERT INTO Realise\_par (numF, numReal, salaireReal)

VALUES

  (1, 1, 5000000),

  (2, 2, 3000000),

  (3, 3, 8000000),

  (4, 4, 4000000),

  (5, 5, 6000000),

  (6, 6, 7000000),

  (7, 7, 6000000),

  (8, 6, 2000000),

  (9, 8, 10000000),

  (10, 8, 10000000),

  (11, 8, 10000000),

  (12, 11, 89000000),

  (13, 10, 89000001),

  (14, 12, 7000000);

INSERT INTO Passe\_dans (numF, numS, numCine, dateDeb, dateFin, horaires, prix, typeSceance)

VALUES

  (1, 1, 1, '2023-08-25', '2023-08-31', '18:00:00', 1500, '2D'),

  (2, 2, 2, '2023-08-26', '2023-09-01', '19:15:00', 1200, '2D'),

  (3, 3, 3, '2023-08-27', '2023-09-02', '20:00:00', 1200, '3D'),

  (4, 4, 4, '2023-08-28', '2023-09-03', '17:30:00', 1350, '2D'),

  (5, 5, 5, '2023-08-29', '2023-09-04', '16:45:00', 8435, '4D'),

  (6, 6, 6, '2023-09-01', '2023-09-07', '16:00:30', 1800, '3D'),

  (9, 4, 7, '2023-02-14', '2023-04-22', '17:00:00', 2000, '3D'),

  (12, 3, 8, '2023-02-14', '2023-04-22', '17:00:00', 2000, '3D');

REQUÊTES

1. **\* Retrouver la liste de tous les films.**

Requête :

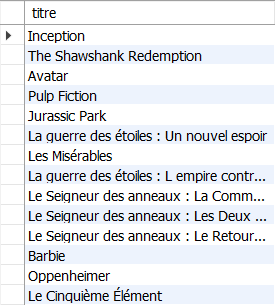
select titre

from films;

Requête algébrique :

π titre(films)

Résultat :



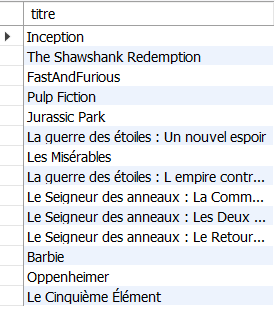
*Modifications :*

update films

set titre = 'FastAndFurious'

where numF = 3;

Résultat de la requête après modification :



1. **\* Donner la liste de tous les genres de films.**

Requête :

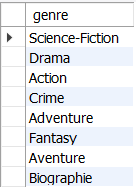
select distinct(genre)

from films;

Requête algébrique :

π genre(films)

Résultat :



*Modifications :*

delete

from realise\_par

where numF = 5;

delete

from film\_a\_recu

where numF = 5;

delete

from joue\_dans

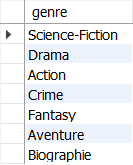
where numF = 5;

delete

from films

where numF = 5;

Résultat de la requête après modification :



1. **Pour chaque film de Spielberg (titre, année), donner le total des salaires des acteurs.**

Requête :

select titre, annee, SUM(salaireAct)

from films

natural join realise\_par

natural join realisateurs

natural join joue\_dans

where nomReal = 'Spielberg'

group by numF;

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151397486234193950/image.png

*Modifications :*

update joue\_dans

set numF = 5

where numAct = 6;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151475546438512680/image.png

1. **\* Trouver le titre et l’année des films de science-fiction dont le budget dépasse $5.000.000.**

Requête :

select titre, annee

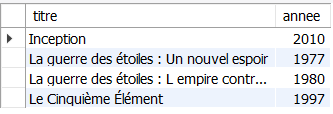
from films

where genre = 'science-fiction'

and budget > 5000000;

Requête algébrique :

Résultat :



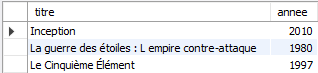
*Modifications :*

update films

set genre = 'Action'

where numF = 6;

Résultat de la requête après modification :



1. **Donner le nombre de films par genre.**

Requête :

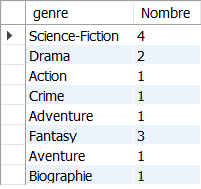
select genre ,count(genre) as Nombre

from films

group by genre;

Requête algébrique :

Résultat :



*Modifications :*

update films

set genre = 'Action'

where numF = 6;

Résultat de la requête après modification :



1. **\* Trouver le titre des films réalisés par Georges Lucas.**

Requête :

select titre

from films

natural join realise\_par

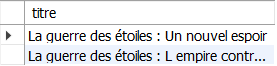
natural join realisateurs

where realisateurs.prenomReal = "George"

and realisateurs.nomReal = "Lucas";

Requête algébrique :

Résultat :



*Modifications :*

update Realise\_par

set numReal = 7

where numF = 8

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151478864485302272/image.png

1. **\* Trouver le titre des films qui ne passent dans aucun cinéma de la compagnie FOX.**

Requête :

select titre

from films

where numF not in (

    select distinct numF

    from passe\_dans

    natural join cinemas

    where compagnie = 'Fox'

);

Requête algébrique :

Résultat :



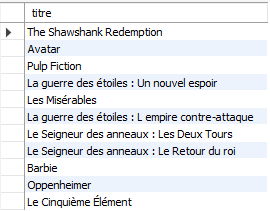
*Modifications :*

update Passe\_dans

set numCine = 5

where numF = 1

Résultat de la requête après modification :



1. **\* Quels sont les acteurs comiques (nom, prénom) qui ont joué dans un film de Spielberg.**

Requête :

select nomAct, prenomAct

from acteurs

natural join joue\_dans

natural join films

natural join realise\_par

natural join realisateurs

where realisateurs.nomReal = "Spielberg"

and acteurs.specialite = "comique";

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151399171916582912/image.png

*Modifications :*

update joue\_dans

set numF = 5

where numAct = 7;

update Acteurs

set specialite = 'comique'

where numAct = 7;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151482442767663124/image.png

1. **Trouver le titre et l’année du film le plus long.**

Requête :

select titre, annee

from cinemas.films

where duree = (select max(duree) from cinemas.films);

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151399171916582912/image.png

*Modifications :*

update films

set duree = 1111

where numF = 12;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151483413233152080/image.png

1. **\* Donner le nom et prénom des acteurs qui ont joué Gavroche dans les différentes versions des ”Misérables” avec les dates correspondantes.**

Requête :

select nomAct, prenomAct, annee

from Acteurs

natural join joue\_dans

natural join films

where rôle = 'Gavroche'

and titre = 'Les Misérables';

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151484345333321800/image.png

*Modifications :*

update Joue\_dans

set rôle = 'Gavroche'

where numAct = 1;

update Joue\_dans

set numF = 7

where numAct =1;

delete from Joue\_dans

where numAct = 7;

update Films

set annee = 2024

where numF = 7

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151487279089590372/image.png

1. **Donner le titre des films qui ont reçu au moins trois récompenses.**

Requête :

select distinct titre

from films

where numF in (

    select numF

    from film\_a\_recu

    group by numF

    Having count(numRecomp) >= 3

);

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151399678672371743/image.png

*Modifications :*

update film\_a\_recu

set numF = 1

where numF= 2;

update film\_a\_recu

set numF = 1

where numF= 3

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151489655791947837/image.png

1. **\* Retrouver la liste des films dont la longueur dépasse 180 minutes.**

Requête :

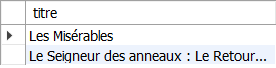
select titre

from films

where duree > 180 ;

Requête algébrique :

Résultat :



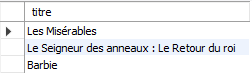
*Modifications :*

update films

set duree = 1111

where numF = 12;

Résultat de la requête après modification :



1. **\* Donner le nom et le prénom des réalisateurs qui ont joué dans au moins un de leurs propres films.**

Requête :

select distinct nomReal, prenomReal

from realisateurs

natural join realise\_par

natural join films

natural join joue\_dans

natural join acteurs

where realisateurs.nomReal = acteurs.nomAct

and realisateurs.prenomReal = acteurs.prenomAct

and films.numF = realise\_par.numF

and realisateurs.numReal = realise\_par.numReal;

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151400294358462484/image.png

*Modifications :*

update Realisateurs

set nomReal = 'Freeman'

where numReal = 2;

update Realisateurs

set prenomReal = 'Morgan'

where numReal = 2;

update Realisateurs

set dateNaissReal = '1937-06-01'

where numReal = 2;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151494403614048287/image.png

1. **Lister les cinémas qui ont exclusivement passé des films primés.**

Requête :

select nomCine

from Cinemas

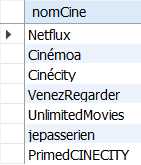
natural join passe\_dans

natural join film\_a\_recu

where passe\_dans.numF = film\_a\_recu.numF;

Requête algébrique :

Résultat :



*Modifications :*

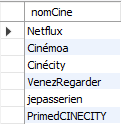
delete from passe\_dans

where numCine = 5

and numS = 5

and numF = 5;

Résultat de la requête après modification :



1. **Quel est le total des salaires des acteurs du film ”La guerre des étoiles” ?**

Requête :

select sum(salaireAct)

from joue\_dans

natural join films

where films.numF = joue\_dans.numF

and titre like '%La guerre des étoiles%';

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151400819741167686/image.png

*Modifications :*

update Joue\_dans

set salaireAct = 5800580

where numAct = 9;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151501087459774605/image.png

1. **Donner la moyenne des salaires des acteurs par film, avec le titre et l’année correspondant.**

Requête :

select avg(salaireAct) as moyenne\_salaires, titre, annee

from films

natural join joue\_dans

group by films.titre, films.annee;

Requête algébrique :

Résultat :



*Modifications :*

update Joue\_dans

set salaireAct = 5800580

where numAct = 9;

Résultat de la requête après modification :



1. **Trouver le genre des films des années 90 dont le budget moyen dépasse $200.000.**

Requête :

select distinct(genre)

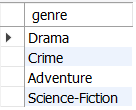
from films

where annee between 1990 and 1999

and budget > 200000;

Requête algébrique :

Résultat :



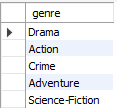
*Modifications :*

update films

set annee = 1998

where numF = 3;

Résultat de la requête après modification :



1. **Lister les cinémas dont la taille moyenne de l’écran est supérieure à 40 mètres carrés.**

Requête :

select nomCine

from cinemas

natural join salles

where salles.tailleEcran > 40;

Requête algébrique :

Résultat :



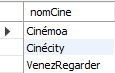
*Modifications :*

update salles

set tailleEcran = 1250

where numCine = 4

Résultat de la requête après modification :



1. **Donner le nom des cinémas passant tous les films primés cette année au festival de Cannes.**

Requête :

select c.nomCine

from cinemas c

natural join passe\_dans

natural join film\_a\_recu

natural join recompenses

where recompenses.festival = 'Cannes'

and film\_a\_recu.annee = 2023

group by c.numCine

having count() = (

    select count(\*)

    from film\_a\_recu

    natural join recompenses

    where recompenses.festival = 'Cannes'

    and annee = 2023

    );

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151401865402138644/image.png

*Modifications :*

update passe\_dans

set numF = 12

where numF = 9

and numCine = 7

and numS = 4;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151511529846480966/image.png

1. **\* Quels sont les cinémas new-yorkais de la Fox qui passent un film de Peter Jackson avant 22 heures dans une salle d’au moins 200 places et un écran de taille supérieure à 30 mètres carrés (donner aussi le nom des films correspondant).**

Requête :

select C.nomCine, F.titre as TitreFilm

from cinemas C

join passe\_dans PD on C.numCine = PD.numCine

join films F on PD.numF = F.numF

join realise\_par RP on F.numF = RP.numF

join realisateurs R on RP.numReal = R.numReal

join salles S on PD.numS = S.numS

where C.villeCine = 'New-York'

  and C.compagnie = 'Fox'

  and R.nomReal = 'Jackson'

  and R.prenomReal = 'Peter'

  and PD.horaires < '22:00:00'

  and S.nbrPlaces >= 200

  and S.tailleEcran > 30;

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151402037519581264/image.png

*Modifications :*

update cinemas

set villeCine = 'New-York'

where numCine = 5;

update salles

set nbrPlaces = 500

where numS = 5;

update salles

set tailleEcran = 100

where numS = 5;

update Passe\_dans

set numF = 10

where numF = 5

and numCine = 5

and numS = 5;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151515614926602312/image.png

1. **Donner le nombre de films sortis en 1980 par genre.**

Requête :

select genre, count(titre) as nombre\_film

from films

where annee = 1980

group by genre;

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151402239538249728/image.png

*Modifications :*

update films

set annee = 1980

where numF = 13;

update films

set annee = 1980

where numF = 14;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151517880370876466/image.png

1. **\* Trouver le nom et le prénom des acteurs qui, pour un film donné, ont eu un salaire plus important que le salaire du réalisateur (attention un acteur peut jouer plusieurs rôles).**

Requête :

select distinct nomAct, prenomAct

from acteurs

inner join joue\_dans on acteurs.numAct = joue\_dans.numAct

inner join realise\_par on joue\_dans.numF = realise\_par.numF

where realise\_par.numF = 2

and joue\_dans.salaireAct > realise\_par.salaireReal;

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151402451526754334/image.png

*Modifications :*

(Pour la modification, nous allons changer le numéro du film choisi dans la requête)

update Joue\_dans

set salaireAct = 1215485269

where numF = 12

and numAct = 10;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151523561438203964/image.png

1. **\* Trouver les couples acteur-réalisateur (noms et prénoms) tels que l’un a dirigé l’autre sur un film et vice-versa sur un autre.**

Requête :

select distinct concat(A1.nomAct, ' ', A1.prenomAct) as Acteur, concat(R1.nomReal, ' ', R1.prenomReal) as Realisateur

from joue\_dans JD1

join realise\_par RP1 on JD1.numF = RP1.numF

join acteurs A1 on JD1.numAct = A1.numAct

join realisateurs R1 on RP1.numReal = R1.numReal

where exists (

    select 1

    from joue\_dans JD2

    join realise\_par RP2 on JD2.numF = RP2.numF

    where JD2.numAct = RP1.numReal

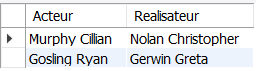
    and JD2.numF != JD1.numF

    and RP2.numReal = JD1.numAct

);

Requête algébrique :

Résultat :



*Modifications :*

update Acteurs

set nomAct = 'PeppaPig'

where numAct = 10

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151529770207690872/image.png

1. **Trouver le nom, le prénom, le numéro des acteurs qui ont joué dans tous les films de Luc Besson.**

Requête :

select distinct A.numAct, A.nomAct, A.prenomAct

from acteurs A

where exists (

    select 1

    from joue\_dans JD

    inner join films F on JD.numF = F.numF

    inner join realise\_par RP on F.numF = RP.numF

    inner join realisateurs R on RP.numReal = R.numReal

    where R.nomReal = 'Besson' and R.prenomReal = 'Luc'

    and JD.numAct = A.numAct

) and A.numAct is not null;

Requête algébrique :

Résultat :

**C:\Users\utilisateur\Desktop\image.png**

*Modifications :*

update Joue\_dans

set numF = 14

where numF = 12

and numAct = 10;

Résultat de la requête après modification :

https://cdn.discordapp.com/attachments/1095844008451899462/1151534834322264135/image.png

1. **Pour chaque film de Peter Jackson, trouver le nom et le prénom de l’acteur qui a eu le plus gros salaire.**

Requête :

select F.titre as NomFilm,

       A.nomAct as NomActeur,

       A.prenomAct as PrenomActeur,

       JD.SalaireAct as SalaireActeur

from films F

inner join realise\_par RP on F.numF = RP.numF

inner join realisateurs R on RP.numReal = R.numReal

inner join joue\_dans JD on F.numF = JD.numF

inner join acteurs A on JD.numAct = A.numAct

where R.nomReal = 'Jackson' and R.prenomReal = 'Peter'

and JD.salaireAct = (

    select MAX(JD2.salaireAct)

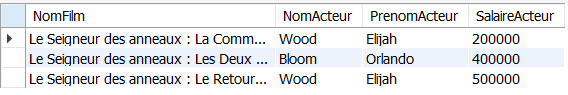
    from joue\_dans JD2

    where JD2.numF = F.numF

);

Requête algébrique :

Résultat :



*Modifications :*

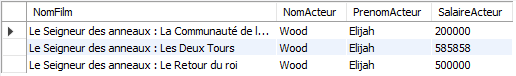
update Joue\_dans

set salaireAct = 585858

where numAct = 13

and numF = 10;

Résultat de la requête après modification :



1. **Donner le titre des films qui ont été primés au moins une fois (en comptant aussi les récompenses des acteurs jouant dans le film).**

Requête :

select distinct F.titre as TitreDuFilm

from films F

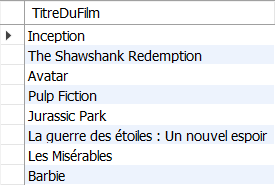
left join film\_a\_recu on F.numF = film\_a\_recu.numF

left join acteur\_a\_recu on film\_a\_recu.numRecomp = acteur\_a\_recu.numRecomp

where film\_a\_recu.numRecomp is not null or acteur\_a\_recu.numRecomp is not null;

Requête algébrique :

Résultat :

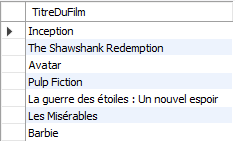


*Modifications :*

delete from film\_a\_recu

where numF = 5

Résultat de la requête après modification :



1. **Donner le prix moyen des films passant dans des salles en 3D (cf. typeSeance dans l’association passe dans).**

Requête :

select avg(passe\_dans.prix) as PrixMoyen

from passe\_dans

inner join salles S on passe\_dans.numS = S.numS

where passe\_dans.typeSceance = '3D';

Requête algébrique :

Résultat :

https://cdn.discordapp.com/attachments/1095844008451899462/1151539581561344060/image.png

*Modifications :*

Résultat de la requête après modification :