

Base de données : movie_library_cinema_session

Cette base de données est une filmothèque à laquelle s'ajoute la possibilité de voir les séances cinéma pour les films à l'affiche.

Cette filmothèque est destinée à accueillir tous les films répertoriés à ce jour.

Dans le cadre de l'exercice, les datas importées correspondent principalement à des films de l'année 2020. Les données de ces films ont été prises sur Allo Ciné et IMDB.

Dans le schéma MySQLWorkBench :

- les tables pivots sont de couleur jaune,
- les tables satellites sont de couleur verte,
- les tables de liaison sont de couleur rouge.

La description de chaque table ainsi que les datatype de chaque colonne sont expliqués dans les zones commentaires de MySQLWorkbench, quand le choix de ces derniers nécessite un commentaire.

La table pivot principale est **movie**.

Elle comporte les films et leurs caractéristiques.

Les tables satellites de la table movie sont :

- **genre** : cette table contient le nom de tous les genres de films existants dans le cinéma.

Un film peut appartenir à plusieurs genres et un genre peut s'appliquer à divers films d'où la création d'une table de liaison **movie_genre**. La clé primaire est composée des deux champs id afin que la relation d'un film avec un genre n'existe qu'une fois.

- **director**: cette table contient les réalisateurs de films et les données personnelles les concernant.

Même si cela est rare, un film peut être réalisé par plusieurs réalisateurs. Bien évidemment un réalisateur peut réaliser plusieurs films d'où la création d'une table de liaison **movie_director**. La clé primaire de la table de liaison est composée des deux champs id afin que la relation d'un film avec un director n'existe qu'une fois.

Cette table peut être également considérée comme une petite table pivot car elle a elle-même une table satellite : **country** (La nationalité d'un réalisateur détermine la nationalité du film).

- **actor**: cette table contient les acteurs de films et les données personnelles les concernant.

Un film peut avoir plusieurs acteurs et un acteur peut jouer dans plusieurs films d'où la création d'une table de liaison **movie_actor**. La clé primaire de la table de liaison est composée des deux champs id afin que la relation d'un film avec un acteur n'existe qu'une fois.

Cette table peut être également considérée comme une petite table pivot car elle a elle-même une table satellite : **country**.

- **production_compagny**: cette table contient les noms des sociétés de production.

Un film peut avoir plusieurs sociétés de production et une société de production peut produire plusieurs films d'où la création d'une table de liaison **movie_production_compagny**. La clé primaire de la table de liaison est composée des deux champs id afin que la relation d'un film avec une société de production n'existe qu'une fois.

- **user**: cette table contient les utilisateurs du site et les données les concernant. Un utilisateur peut noter un film et/ou laisser un commentaire. Un utilisateur n'est pas obligé d'avoir un compte pour laisser un commentaire.

Un film peut avoir plusieurs notes et/ou commentaires et un utilisateur peut noter et/ou commenter plusieurs films d'où la création d'une table de liaison **chronicle_movie_user**. La clé primaire de la table de liaison est composée des deux champs id afin que la relation d'un film avec un user n'existe qu'une fois. Elle comprend également deux champs pouvant contenir la critique et/ou la note.

- **cinema** : cette table contient les cinémas de France et les données les concernant (liste établie sur la base des établissements actifs en 2018).

Un film peut être diffusé dans plusieurs cinémas et un cinéma peut diffuser plusieurs films d'où la création d'une table de liaison **movie_cinema**. Pas de clé primaire sur la table de liaison afin que la relation d'un film avec un cinéma puisse exister plusieurs fois car différentes dates de diffusion à différents horaires. Elle comprend également trois champs contenant le jour, l'heure et la version de diffusion.

La table cinéma est à elle seule une autre table pivot car elle possède plusieurs tables satellites.

Les tables satellites de la table **cinema** sont :

- **city** : cette table contient la liste des villes de France et les données y afférent.

Cette table peut être également considérée comme une petite table pivot car elle a elle-même une table satellite : **departement** qui contient les départements de France.

- **staff** : cette table contient l'ensemble du personnel de tous les cinémas et les données les concernant.

Un cinéma peut avoir plusieurs salariés mais on considère qu'un salarié ne peut travailler que dans un cinéma. La table staff devient donc une table pivot et la table cinéma sa table satellite.

Elle a également la table city en satellite.

- **customer**: cette table contient l'ensemble des clients de tous les cinémas ayant un programme de fidélité et les données les concernant.

Un cinéma peut ne pas avoir de programme de fidélité mais dans le cas inverse, il peut avoir plusieurs clients et dans l'absolu, un client peut l'être dans différents cinémas d'où la création d'une table de liaison **cinema_customer**. La clé primaire de la table de liaison est composée des deux champs id afin que la relation d'un cinéma avec un client n'existe qu'une fois.

Cette table peut être également considérée comme une petite table pivot car elle a elle-même une table satellite : city.

La table **country** contient la liste des pays officiels et se base sur leur code officiel à 3 caractères.

C'est une table satellite des tables director et actor.

La table movie n'est pas reliée à la table country car elle l'est implicitement via la table director.

La table **picture** est une base de stockage de photos.

Elle contient les affiches de films et les photos des réalisateurs et acteurs.

Pas de relation concrètement définie car dans ce cas la relation est induite par la présence de l'id de chaque table concernée.

Vues et requêtes

use movie_library_cinema_session;

View 1

-- *Vue sur le nom des films actuellement à l'affiche*

-- réponse :

-- Bad Boys For Life

-- Bad Lieutenant

-- Underwater

SELECT title FROM movie WHERE isShowing=1 ORDER BY title;

View 2

-- *Vue sur les cinémas du département d'Indre et Loire ayant des écrans 3D*

-- réponse :

-- Amboise, Cinea

-- Chateau-Renault, Le Balzac

-- Chinon, Le Rabelais

-- L'Ile-Bouchard, Le Cube

-- Langeais, Espace Jean-Hugues Anglade (Ex-Foyer Communal)

-- Loches, Vigny

-- Monnaie, Foyer Rural

-- Richelieu, Majestic

-- Sainte-Maure-De-Touraine, Salle Patrice Leconte

-- Tours, Cine Loire

-- Tours, Mega Cgr 2 Lions

-- Tours, Mega Cgr Centre

-- Tours, Studio Des Ursulines

SELECT city.name as Ville, cinema.name as Cinema FROM cinema, city

WHERE cinema.id_city=city.id

AND code_department = 37

AND 3D_screen = 1

ORDER BY Ville, Cinema;

Requête 1

-- *Dans quel genre y-a-t-il le plus de films français et combien ?*

-- réponse : Comédie, 6

```
SELECT G.name as Genre, COUNT(M.id) as NombreFilms FROM director as D, movie_director as MD, movie as M,
movie_genre as MG, genre as G
WHERE D.id=MD.id_director AND MD.id_movie=M.id AND M.id=MG.id_movie AND MG.id_genre=G.id
AND code_country = 'FRA'
GROUP BY Genre
HAVING NombreFilms=(SELECT COUNT(M.id) as NombreFilms FROM director as D, movie_director as MD,
movie as M, movie_genre as MG, genre as G
WHERE D.id=MD.id_director AND MD.id_movie=M.id AND
M.id=MG.id_movie AND MG.id_genre=G.id
AND code_country = 'FRA'
GROUP BY G.name
ORDER BY NombreFilms DESC
LIMIT 1);
```

Requête 2

-- *Quelle société de production a produit le plus grand nombre de films sur le mois de janvier 2020 et combien ?*

-- réponse : Sony Pictures, 3

```
SELECT PC.name as Production, COUNT(M.id) as NombreFilms FROM production_compagny as PC
JOIN(movie_production_compagny as MPC, movie as M)
ON(PC.id=MPC.id_production_compagny AND MPC.id_movie=M.id)
WHERE release_year BETWEEN "2020-01-01" AND "2020-01-31"
GROUP BY Production
HAVING NombreFilms=(SELECT COUNT(M.id) as NombreFilms FROM production_compagny as PC
JOIN(movie_production_compagny as MPC, movie as M)
ON(PC.id=MPC.id_production_compagny AND MPC.id_movie=M.id)
WHERE release_year BETWEEN "2020-01-01" AND "2020-01-31"
GROUP BY PC.name
ORDER BY NombreFilms DESC
LIMIT 1);
```

Requête 3

-- *Quel film a la moyenne de notes la plus haute et quelle est cette moyenne ?*

-- réponse :

-- Amour-Eux, 6.40

-- Echo, 6.40

```
SELECT M.title as Film, ROUND(AVG(rating),2) as MoyenneNotes FROM movie as M, chronicle_movie_user as
CMU, user as U
WHERE M.id=CMU.id_movie AND CMU.id_user=U.id
GROUP BY Film
HAVING MoyenneNotes=(SELECT ROUND(AVG(rating),2) as MoyenneNote FROM movie as M,
chronicle_movie_user as CMU, user as U
WHERE M.id=CMU.id_movie AND CMU.id_user=U.id
GROUP BY M.title
ORDER BY MoyenneNote DESC
LIMIT 1)
ORDER BY M.title;
```

Requête 4

-- Quel acteur a réalisé le film dans lequel il a joué, quel est le nom du film et le pays?

-- réponse :

-- Charlotte Dauphin, L'Autre, France

-- Jeanne Balibar, Merveilles à Montfermeil, France

-- Jézabel Marques, SOL, France

SELECT A.name as Acteur_Realisateur, M.title as Film, C.name as Pays FROM actor as A JOIN(movie_actor as MA, movie as M, country as C)

ON(A.id=MA.id_actor AND MA.id_movie=M.id AND C.code=A.code_country)

WHERE A.name IN (SELECT A.name as Acteur_Realisateur FROM actor as A JOIN(movie_actor as MA, movie as M, director as D, movie_director as MD)

ON A.id=MA.id_actor AND MA.id_movie=M.id and D.id=MD.id_director and

MD.id_movie=M.id

WHERE A.name=D.name)

ORDER BY Acteur_Realisateur;

Requête 5

-- Quel est le film le plus cosmopolite (nombre d'acteurs de nationalités différentes) ?

-- réponse :

-- Go Go Tales, 4

-- Les Filles du Docteur March, 4

SELECT title as Film, COUNT(DISTINCT(code_country)) as NombrePays FROM movie as M, movie_actor as MA, actor as A

WHERE M.id=MA.id_movie AND MA.id_actor=A.id

GROUP BY Film

HAVING NombrePays=(SELECT COUNT(DISTINCT(code_country)) as NombrePays FROM movie as M, movie_actor as MA, actor as A

WHERE M.id=MA.id_movie AND MA.id_actor=A.id

GROUP BY title

ORDER BY NombrePays DESC

LIMIT 1);

Requête 6

-- Quel est le pays le plus représenté dans les réalisateurs et combien en compte-il ?

-- réponse : France, 10

SELECT C.name as Pays, COUNT(D.id) as NombreRealisateur FROM director as D JOIN country as C

ON C.code=D.code_country

GROUP BY Pays

HAVING NombreRealisateur=(SELECT COUNT(D.id) as NombreRealisateur FROM director as D JOIN country as C

ON C.code=D.code_country

GROUP BY code_country

ORDER BY NombreRealisateur DESC

LIMIT 1)

ORDER BY NombreRealisateur DESC;

Requête 7

-- A quelles dates et quelles heures sont diffusées les séances en VO du film Underwater dans le MEGA CGR 2 lions sur le mois de mai 2020 ?

-- réponse :

-- 2020-05-27, 11:30:00

-- 2020-05-28, 11:30:00

-- 2020-05-29, 11:30:00

-- 2020-05-30, 11:30:00

-- 2020-05-31, 11:30:00

SELECT showing_date as Date, showing_hour as Heure FROM movie as M, movie_cinema as MC, cinema as C

WHERE M.id=MC.id_movie AND MC.id_cinema=C.id

AND title='Underwater'

AND C.name LIKE '%lions%'

AND showing_date BETWEEN '2020-05-01' AND '2020-05-31'

AND showing_version = 'VO';

Requête 8

-- Dans quelle ville se situe le cinéma qui a le plus d'employés (nom du cinéma et combien d'employés) ?

-- réponse :

-- Capdenac-Gare, Atmosphere, 8
-- Dinard, 2 Alizes Salle La Mouette, 8
-- Eaubonne, L'Orange Bleue, 8
-- Graulhet, Cinema Vertigo, 8
-- Haguenau, Megarex, 8
-- La Ciotat, Cinema Eden Theatre, 8
-- Les Ancizes-Comps, Cinema De La Viouze, 8
-- Woippy, Salle Michel Bonnet, 8

SELECT Cy.name as Ville, Ci.name as Cinema, COUNT(S.id) as NombreEmployes FROM cinema as Ci JOIN(staff as S, city as Cy)

ON(S.id_cinema = Ci.id AND Ci.id_city=Cy.id)

GROUP BY Ci.id

HAVING NombreEmployes=(SELECT COUNT(S.id_cinema) as NombreEmployes FROM cinema as C JOIN staff as S

ON S.id_cinema = C.id

GROUP BY C.id

ORDER BY NombreEmployes DESC

LIMIT 1)

ORDER BY Cy.name;

Requête 9

-- Combien de salariés a le cinéma (nom et ville) qui a le plus de clients ?

-- réponse :Cinema Le Mont d'Or, Metabief, 2

SELECT Ci.name as Cinema, Cy.name as Ville, COUNT(S.id) as NombreEmployes FROM cinema as Ci, staff as S, city as Cy

WHERE Ci.id=S.id_cinema AND Cy.id=Ci.id_city

GROUP BY Ci.id

HAVING Cinema IN (SELECT Ci.name FROM cinema as Ci, cinema_customer as CC, customer as Cu

WHERE Ci.id=CC.id_cinema AND CC.id_customer=Cu.id

GROUP BY Ci.id

HAVING COUNT(Cu.id)=(SELECT COUNT(CC.id_customer) as NombreClients

FROM cinema as C, cinema_customer as CC, customer as Cu

WHERE C.id=CC.id_cinema AND CC.id_customer=Cu.id

GROUP BY C.id

ORDER BY NombreClients DESC

LIMIT 1))

ORDER BY Ci.name;

Requête 10

-- Quel cinéma se rapproche le plus de la moyenne du nombre de sièges, dans quelle ville est-il et combien a-t-il de sièges?

-- réponse :

-- Calypso A, Viry-Chatillon, 551
-- Rex, Chatenay-Malabry, 551
-- Utopia, Bordeaux, 553

SELECT Ci.name as Cinema, Cy.name as Ville, seat_count as NombreSieges FROM cinema as Ci JOIN city as Cy

ON Cy.id=Ci.id_city

WHERE ABS(seat_count - (SELECT ROUND(AVG(seat_count)) FROM cinema))=(SELECT MIN(ABS(seat_count - (SELECT ROUND(AVG(seat_count)) FROM cinema))) FROM cinema)

ORDER BY Cinema;