

Le Syndicat d'Initiative de Grenoble a recruté un **spécialiste de bases de données** (concepteur, administrateur et programmeur d'applications) pour réaliser une base de données sous le système de gestion de bases de données **Postgres** afin de renseigner les usagers sur la **programmation des films du jour** (du matin 10h au soir 22h). Les données sont mises à jour chaque matin avant 10h, ainsi qu'au fil des éventuelles annulations que les cinémas signalent.

À ce jour, la base de données **Cinema** est donc opérationnelle.

La BD CINEMA est unique et peut être consultée par chacun d'entre vous via des requêtes SELECT  
Dans ce TP, vous jouerez donc le rôle d'un simple utilisateur (dans de futurs TPs, vous serez l'administrateur de votre propre base de données).

#### SCHEMA DE LA BD CINEMA

CINE	( <u>NomCine</u> , Adresse) <i>Un cinéma a une adresse et est identifié par son nom.</i>
FILM	( <u>Titre</u> , MetteurScene) <i>Un film a un metteur en scène et est identifié par son titre.</i>
SALLE	(# <u>NomCine</u> , <u>NumSalle</u> , Prix, #Titre) <i>Une salle (identifiée par le nom de son cinéma et un numéro) projette un film à un certain prix. Dans une salle d'un cinéma donné, le film et le prix sont identiques pour toutes les séances de la journée. L'attribut NomCine prend ses valeurs parmi celles de la clé primaire nomCine de CINE. L'attribut Titre prend ses valeurs parmi celles de la clé primaire Titre de FILM.</i>
SEANCE	(#( <u>NomCine</u> , <u>NumSalle</u> ), Heure) <i>Cette relation donne les horaires de projection dans les différentes salles des cinémas de la base. Le couple (NomCine, NumSalle) prend ses valeurs parmi celles du couple (NomCine, NumSalle), clé primaire de SALLE.</i>
ACTEUR	(# <u>Titre</u> , <u>NomAct</u> ) <i>Un acteur est caractérisé par son nom et joue dans un ou plusieurs des films programmés pour la journée. L'attribut Titre prend ses valeurs parmi celles de la clé primaire Titre de FILM.</i>

#### PRELIMINAIRES ET CONNEXION AU SGBD POSTGRESQL

##### PRELIMINAIRES

- Dans votre répertoire racine, créez un répertoire R1.05
- Placez-vous dans le répertoire R1.05, puis créez un répertoire TP1\_Cinema
- Double-cliquez sur le fichier **reqTP1.sql** disponible dans le parcours **Semaine3\_TP1\_SELECT**
- Ouvrez ce fichier avec un éditeur de texte (*gedit*, *atom*, ...)
- Enregistrez-le dans votre répertoire TP1\_Cinema

##### CONNEXION A LA BASE SOUS POSTGRESQL / DECONNEXION DE POSTGRESQL

- Pour vous connecter à la BD cinema sous PostgreSQL...
  - (1) Ouvrez un terminal
  - (2) tapez la commande : **psql -h postgres-info -U invite cinema**
  - (3) entrez le mot de passe : **invite**
- Pour vous déconnecter de PostgreSQL, tapez : **\q**

**NOTE** : Avant de fermer votre session linux, n'oubliez pas de vous déconnecter de PostgreSQL

## Découverte de la BD CINEMA

- Connectez-vous à PostgreSQL (cf. page précédente)
- Listez les relations de la base en tapant la commande `\d`
- Pour chaque relation :
  1. Affichez son schéma en tapant la commande `\d nom de la relation`
  2. Vérifiez que sa **clé primaire** est bien l'attribut ou l'ensemble d'attributs soulignés en 1<sup>ère</sup> page
  3. Vérifiez que les **clés étrangères** marquées d'un # en 1<sup>ère</sup> page sont bien précisées

Exemple : `\d ACTEUR`

Column	Type	Collation	Nullable	Default
titre	character varying(50)		not null	
nomact	character varying(20)		not null	
Indexes:				
"acteur_pkey" PRIMARY KEY, btree (titre, nomact)				
Foreign-key constraints:				
"acteur_titre_fkey" FOREIGN KEY (titre) REFERENCES film(titre)				

4. Affichez son contenu

Exemple : `SELECT * FROM ACTEUR;`

## Recherche d'informations dans la BD CINEMA

AVANT DE TESTER UNE REQUETE...

- ✓ Dans le cas où plusieurs relations sont requises dans la clause **FROM**, vérifiez si elles sont toutes nécessaires et, si c'est le cas, vérifiez que les jointures entre ces tables sont bien réalisées
- ✓ Demandez-vous s'il est nécessaire d'utiliser la clause **DISTINCT** pour obtenir les résultats souhaités
- ✓ Rappelez-vous qu'il est possible d'utiliser plusieurs fois une même relation R dans la clause **FROM**, sous réserve de renommer chacune des occurrences de cette relation

### Partie A : Requêtes sans GROUP BY ni sous-requêtes

POUR CHAQUE QUESTION DE CETTE PARTIE, RESPECTEZ LA PROCEDURE SUIVANTE :

- 1<sup>ère</sup> étape : Complétez le fichier **reqTP1A.sql** avec le code de la requête répondant à la question
- 2<sup>ème</sup> étape : Copiez cette requête (**ctrl C**), puis collez-la (**ctrl + Shift + V**) dans le terminal où vous êtes connectés à PostgreSQL
- 3<sup>ème</sup> étape : Vérifiez le résultat obtenu par comparaison avec celui attendu (cf. fichier **TP1A\_Résultats**)
- 4<sup>ème</sup> étape : Si vous n'obtenez pas le résultat attendu, recommencez à l'étape 1

- Écrivez et testez les requêtes qui permettent d'obtenir les informations suivantes :
  - (a) Numéro des salles et heure des séances du cinéma *La Nef*
  - (b) Pour chaque salle du cinéma *Pathé* : numéro de la salle, titre du film projeté et prix à payer
  - (c) Nom des acteurs qui jouent dans des films mis en scène par *Peter Jackson*
  - (d) Nom et adresse des cinémas qui passent des films mis en scène par *George Lucas*
  - (e) Par ordre alphabétique, nom des acteurs qui jouent dans au moins un des films *Star Wars* (le titre de ces films doit commencer par *Star Wars*)
  - (f) Nom des cinémas où passe un film mis en scène par un réalisateur dont le nom contient *Porum*
  - (g) Heure, numéro de la salle et prix à payer pour chaque salle du cinéma *La Nef* où est projeté le film *Molière* - résultat ordonné chronologiquement
  - (h) Nom des acteurs qui jouent dans *Eternal Sunshine of the Spotless Mind* et aussi dans *The Virgin Suicides* - une version avec opérateur ensembliste et une version sans opérateur ensembliste

- (i) Nom des acteurs qui jouent dans *Eternal Sunshine of the Spotless Mind* mais pas dans *The Virgin Suicides*
- (j) Nom des acteurs qui jouent dans *The Virgin Suicides* mais pas dans *Eternal Sunshine of the Spotless Mind*
- (k) Titre des films interprétés par *Romain Duris* ou par *Fabrice Luchini*  
- une version avec opérateur ensembliste et une version sans opérateur ensembliste
- (l) Nombre de séances où est projeté un film interprété par *Romain Duris* ou par *Fabrice Luchini*
- (m) Pour toutes les projections d'un film interprété par *Romain Duris* ou par *Fabrice Luchini* : titre du film, nom du cinéma, numéro de la salle, heure de la projection et prix à payer - résultat ordonné par cinéma, puis par heure de projection
- (n) Écrivez une requête qui affiche les informations suivantes :

titre	nomact	titre	metteurscene
Les Climats	Nuri Bilge Ceylan	Les Climats	Nuri Bilge Ceylan
Mystic River	Sean Penn	Crossing Guard	Sean Penn
CQ	Sofia Coppola	The Virgin Suicides	Sofia Coppola
CQ	Sofia Coppola	Lost in Translation	Sofia Coppola

(4 rows)

- (o) Inspirez-vous de la requête produite en (n) pour écrire des requêtes affichant :
  - le nombre de films dans lesquels joue un metteur en scène
  - le titre des films où le metteur en scène fait partie des interprètes

## Partie B : Nouvelles recherches d'informations

Dans cette partie, vous serez éventuellement amenés à utiliser des **sous-requêtes** et/ou des requêtes comportant une clause **GROUP BY**, pouvant être complétée d'une clause **HAVING**.

POUR CHAQUE QUESTION DE CETTE PARTIE, RESPECTEZ LA PROCEDURE SUIVANTE :

- 1<sup>ère</sup> étape : Complétez le fichier **reqTP1B.sql** avec le code de la requête répondant à la question
- 2<sup>ème</sup> étape : Copiez cette requête (**ctrl C**), puis collez-la (**ctrl + Shift + V**) dans le terminal où vous êtes connectés à PostgreSQL
- 3<sup>ème</sup> étape : Vérifiez le résultat obtenu par comparaison avec celui attendu (cf. fichier **TP1B\_Résultats**)
- 4<sup>ème</sup> étape : Si vous n'obtenez pas le résultat attendu, recommencez à l'étape 1

- Écrivez et testez les requêtes qui permettent d'obtenir les informations suivantes :
  - (a) Réécrivez en utilisant une ou des sous-requêtes, les requêtes des questions (h) et (i) de la partie A
  - (b) Nombre d'acteurs par films - une version avec **GROUP BY** + une version avec sous-requêtes ; résultat ordonné par nombre d'acteurs décroissant, puis par titre
  - (c) Programmation en cours : pour chaque cinéma, nombre de films projetés, sachant que chaque cinéma projette au moins un film - résultat ordonné par nombre de films décroissant, puis par nom de cinéma
  - (d) Nom des cinémas projetant au moins 2 films Star Wars différents mis en scène par *George Lucas* (leur titre commence par Star Wars)
  - (e) Prix maximum, prix minimum et prix moyen arrondi à 2 décimales, des projections programmées par chaque cinéma - résultat ordonné par nom de cinéma
  - (f) Titre et nom du metteur en scène des films qui ne sont pas programmés pour la journée en cours - 1 version sans sous-requête et une version avec sous-requête
  - (g) Pour afficher le nombre de séances débutant après 20h dans chaque cinéma, la requête suivante a été proposée...
 

```
SELECT NomCine, count(*) AS Nb_Seances_Soir
FROM SEANCE
WHERE Heure > 20
GROUP BY NomCine
ORDER BY NomCine;
```

    - Exécutez-la...
    - Justifiez pourquoi elle n'est pas satisfaisante
    - Proposez une solution recevable

- (h) Pour chaque cinéma, nombre de salles qui projettent un film interprété par *Gérard Depardieu* ou par *Catherine Deneuve* - résultat ordonné par nom de cinéma
- (i) Des usagers ont fait part de leur besoin de connaître les détails des séances de projection proposées après 18h à un prix inférieur à la moyenne des prix pratiqués : heure de séance, prix à payer, titre du film projeté, nom du cinéma et numéro de la salle
- Pour y répondre, nous vous proposons de procéder par étapes :
1. Écrire une requête affichant la moyenne des prix pratiqués (moyenne arrondie à 2 décimales)
  2. En utilisant la requête précédente, affichez les informations désirées - résultat ordonné par heure, prix, puis par nom de cinéma
- (j) On s'intéresse aux films interprétés par *Fabrice Luchini*.  
Écrivez les requêtes affichant les informations suivantes :
1. Nombre de séances où est projeté un film interprété par *Fabrice Luchini*
  2. Répartition de ces séances par cinéma – résultat ordonné sur le nom des cinémas
  3. Pour chaque cinéma où est projeté un film interprété par *Fabrice Luchini*, son nom, le nombre de salles qui proposent ce film et le prix maximum pratiqué parmi ces salles
  4. Nombre de projections pour chaque film interprété par *Fabrice Luchini*
  5. Pour chaque film interprété par *Fabrice Luchini*, nombre de projections par cinéma passant ce film - résultat ordonné par nombre de projections décroissant, puis par titre
- (k) Nom et adresse des cinémas qui projettent plus de 6 films différents
- (l) Nom et adresse des cinémas qui projettent le plus de films différents