

Labo 3 : SQL

Objectifs

Ce laboratoire a pour but l'apprentissage du langage de requête SQL, ainsi que des langages associés DDL et DML.

Indications

Importez le schéma fourni dans Workbench, pour créer la base de données nécessaire à ce laboratoire (`sakila-schema.sql`), puis importez les données (`sakila-data.sql`) afin de la peupler.

Ce laboratoire est à rendre pour :

- Le 23.11.2015 à 23h55 pour la **classe BDR-1-A-L1**.
- Le 24.11.2015 à 23h55 pour la **classe BDR-1-B-L1**.

Consignes générales

- Le fichier `sakila_schema.pdf` représente la base de données Sakila. Prenez le temps de bien comprendre ce schéma. Le document `sakila.pdf` regroupe toutes les informations concernant cette base de données.
- Vous devez exécuter les requêtes « à la main », c'est-à-dire qu'il est obligatoire d'écrire le code SQL de chaque requête, afin de pouvoir l'ajouter au rapport final.
- Tous les mots clés du langage (`SELECT`, `FROM`, `WHERE`, ...) doivent être écrit en majuscule, le reste, en minuscule, et les requêtes doivent être **correctement formatées et indentées**.

```
SELECT
    first_name,
    last_name,
    email
FROM customer
WHERE email IS NOT NULL
    AND store_id = (
        SELECT
            MAX(store_id)
        FROM store
    )
ORDER BY customer.last_name;
```

- Sauf mention contraire, les jointures se font avec le mot clé `JOIN`, pas dans le `WHERE`, ni à l'aide d'un `IN`.
- Le mot clé `DISTINCT` ne sera utilisé que lorsqu'il est **vraiment nécessaire**.
- Exécutez toutes les requêtes d'interrogation de la base de données, **avant** celles qui la modifient, sinon, les résultats seront différents de ceux voulus (ou alors, pensez à restaurer l'état de la base de données entre chaque requête).

- Seules les informations demandées doivent être affichées dans les résultats des requêtes.
- Il ne doit y avoir aucun doublon dans les réponses.
- Lorsque dans la donnée, une liste d'attributs, entre parenthèse, est donnée, les attributs correspondant seront représentés dans les résultats des requêtes, en les renommant si nécessaire. Par exemple, si une colonne se nomme `title` et que dans la donnée, il est demandé `titre`, il faudra renommer la colonne dans le résultat, à l'aide de code SQL.
- Si le résultat d'une requête fournit un grand nombre de lignes, seules les 20 premières du résultat seront reportées dans le rapport, avec le nombre de lignes total.

titre	nombre_acteurs
LUCKY FLYING	10
OLEANDER CLUE	10
INSIDER ARIZONA	9
WIZARD COLDBLOODED	9
PERSONAL LADYBUGS	9
ALONE TRIP	8
RUNNER MADIGAN	8
CHAMBER ITALIAN	7
TELEGRAPH VOYAGE	7
UNCUT SUICIDES	7
DRIVING POLISH	7
MONSTER SPARTACUS	7
ALASKA PHANTOM	7
HANOVER GALAXY	7
BOOGIE AMELIE	6
HOME PITY	6
CLUE GRAIL	6
TAXI KICK	6
WORDS HUNTER	6
AMELIE HELLFIGHTERS	6
✓ 82 17:49:51	51 row(s) returned

- Complétez le fichier `student_NOM1_NOM2.sql` au fur et à mesure que vous complétez les exercices. Ajoutez chaque exercice entre les balises correspondantes. Attention, c'est ce fichier qui sera exécuté pour corriger le laboratoire, veillez à ce qu'il soit fonctionnel et exécutable.
- La requête 8 est optionnelle. Vous pouvez gagner un bonus de 0.3pts à la note finale, si cette dernière est juste.

Requêtes

1. Donnez l'ensemble des films (`title`, `release_year`) classés (`rating`) G, ayant une durée de plus de 100 minutes et dont les coûts de remplacements sont 29.99\$, en les ordonnant par titre.
2. Donnez le N°, le nom et le prénom (`customer_id`, `first_name`, `last_name`) des clientes, dont le prénom est TRACY, qui sont rattachées au magasin N°1, ordonnées par N° de client décroissant.
3. Listez tous les clients actifs (`numéro`, `prénom`, `nom`) habitant la ville 321, et rattachés au magasin N°2. Triez-les par nom de famille croissant.
4. Listez le pays, la ville et le N° postal (`pays`, `ville`, `npa`) des villes françaises, ainsi que des villes dont le N° de pays est entre 50 et 58 (bornes non comprises), en les ordonnant par pays, puis par ville et en dernier par npa. N'utilisez pas de BETWEEN.
5. Donnez les films de science-fiction (`film_id`, `title`, `langue`), dans lesquels joue au moins un acteur dont le prénom est ALAN ou BEN, en les ordonnant par N° de film décroissant.
6. Donnez le N°, le nom et le prénom (`customer1_id`, `customer1_first_name`, `customer1_last_name`, `customer2_id`, `customer2_first_name`, `customer2_last_name`) des clients qui ont loué au moins une fois le même film (par exemple, si ALAN et BEN ont loué le film MATRIX, mais pas TRACY, seuls ALAN et BEN doivent être listé).
7. Donnez le nom et le prénom des acteurs ayant joué dans un film d'action, dont le prénom commence par B, ou dont le nom de famille commence par A. Donnez deux versions de cette requête :
 - a. En écrivant les jointures à l'aide du mot clé JOIN.
 - b. Sans utiliser le mot clé JOIN, et en évitant les sous-requêtes.
8. (BONUS) Donnez le N°, le nom et le prénom (`customer_id`, `first_name`, `last_name`) des clients qui ont loué tous les films de l'actrice EMILY DEE.
9. Donnez le titre des films et le nombre d'acteurs (`titre`, `nombre_acteurs`), des films de musique :
 - a. En les triant par le nombre d'acteur décroissant.
 - b. Seulement les films ayant plus de 7 acteurs, en les triant par le nombre d'acteur décroissant.
10. Listez les catégories (`id`, `nom`, `nb_films_associes`) de films associées à plus de 60 films, sans utiliser de sous-requête, et en les ordonnant par nom de catégorie.
11. Affichez le film (`id_min`, `titre_min`, `durée_min`) ayant la durée la plus courte. Si plusieurs films ont la même durée (la plus courte), il faut tous les afficher.
12. Listez les acteurs (`actor_id`, `nombre_films`) qui ont joué dans plus de 35 films, sans utiliser de sous-requêtes.

13. Listez les films (`id`, `titre`) dont l'identifiant est inférieur à 100, dans lesquels joue au moins un acteur qui a joué dans plus de 35 films, en les ordonnant par `id`. Donnez deux versions de cette requête :
- En utilisant le mot clé `IN`.
 - Sans utiliser le mot clé `IN`. *Indication : une sous-requête peut être utilisée comme une table, sur laquelle on peut donc effectuer des jointures.*
- Quelle requête est la plus rapide ? A votre avis, pourquoi ?
14. Donnez les films (`film_id`, `title`, `prix`), dont le prix de location est inférieur à 2.00\$, qui n'ont jamais été loués.
15. Un fou décide de regarder l'ensemble des films qui sont présents dans la base de données. Etablissez une requête qui donne le nombre de jours (`jours`) qu'il devra y consacrer, s'il dispose de 16h par jour.
16. Affichez tous les clients (`id`, `nom`, `prénom`, `pays`, `nombre_films_total`, `total_dépense`, `dépense_moyenne`) résidant en Inde, au Japon ou au Maroc, dont la dépense moyenne par film loué est supérieure à 3.4, en les ordonnant par `pays`, puis par `nom`. *Indication : commencez par établir une requête affichant tous les clients avec leur dépense moyenne pour les films loués, puis, une nouvelle requête qui ne retourne que les clients dont la dépense moyenne par film est supérieure à 3.4, en utilisant la première requête comme sous-requête.*
17. Donnez la liste des clients (`id`, `nom`, `prénom`, `pays`) japonais et français, qui n'ont pas encore rendu tous les films qu'ils ont empruntés, en les ordonnant par `pays`, puis par `nom`. Donnez trois versions de cette requête :
- Utilisez `EXISTS` (pas de `GROUP BY`, ni de `IN` ou `NOT IN`).
 - Utilisez `IN` (pas de `GROUP BY`, ni de `EXISTS` ou `NOT EXISTS`).
 - N'utilisez ni `GROUP BY`, ni `IN`, ni `NOT IN`, ni `EXISTS`, ni `NOT EXISTS`.

Avant de commencer les requêtes suivantes, sauvegardez la base si nécessaire.

18. Donnez les trois requêtes suivantes, ainsi que le résultat de la première et de la dernière :
- Listez le nombre de paiements dont la valeur est supérieure à 11.
 - Effacez ces paiements.
 - Listez à nouveau ces paiements pour vérifier que l'opération a bien eu lieu.
19. En une seule requête, modifiez les paiements comme suit :
- Chaque paiement de plus de 5 est majoré de 50%.
 - La date de paiement est mise à jour avec la date courante du serveur.

Client (requête 20) :

M. Marcel Rochat

Adresse : Rue du centre, 1260 Nyon

Pays : Suisse / Switzerland

Tél. : 022/360.00.00

E-mail : mr@bluewin.ch

Rattaché au magasin N°1.

20. Insérez le nouveau client dans la base de données, avec toutes les informations requises pour que vous puissiez louer des films.
- Spécifiez les attributs (colonnes) lors de l'insertion. *Indication : plusieurs requêtes sont nécessaires.*
 - Pour chaque nouveau tuple, la base de données doit générer l'id, pourquoi ne pouvez-vous pas le faire ?
 - Pour chaque clé étrangère pour laquelle une valeur est requise, une requête doit donner cette valeur. On considère que l'ensemble des requêtes nécessaires sera fait dans une transaction, ainsi, seules vos modifications de la base de données seront effectives (pas de soucis de concurrence avec une éventuelle autre application).
 - Ecrivez **une** requête d'interrogation, qui montrera que l'ensemble des opérations s'est bien déroulé. *Indication : « Marcel Rochat » est un identifiant unique suffisant.*