

Labo 3 : SQL

1 Objectif

Il s'agira de mettre en pratique vos connaissances concernant le langage de requête SQL.

1.1 A rendre

Vous devrez nous rendre une archive ZIP contenant les éléments suivants:

1. Un fichier SQL contenant les différentes requêtes à réaliser
2. Un rapport dans lequel vous présenterez les différents résultats de chaque requête

Vous nommerez votre rapport dans le format suivant: **Lab3__[A|B|C]__Nom1__Nom2.pdf** où **A, B ou C** représente le nom de la classe, sans oublier vos noms dans l'ordre alphabétique. Le laboratoire sera comme d'habitude rendu sur Cyberlearn avant le:

Jeudi 24 Novembre 2022 à 23h59

1.2 Indications

Importez le schéma fourni (**pagila-schema.sql**) dans Datagrip pour créer la base de données nécessaire à cette partie du laboratoire, puis importez le fichier **pagila-import.sql** pour remplir cette base nouvellement créée.

Dans ce laboratoire on utilise une version modifiée de la base de donnée d'exemple *pagila* pour Postgres (basé sur la base de donnée *sakila* pour MySQL).

1.3 Consignes générales

1. Le fichier **pagila-schema.sql** contient les requêtes définissant la structure de la base de données. Une fois importé, prenez le temps de bien comprendre le schéma résultant.
2. Vous devez exécuter les requêtes *à la main*, c'est-à-dire qu'il est obligatoire d'écrire le code SQL de chaque requête. Celui-ci devra être inclus en annexe dans le rapport final.
3. Sauf mentions contraire, les jointures se font avec le mot clé **JOIN**, pas dans le **WHERE** ni avec un **IN**.
4. Le mot clé **DISTINCT** ne sera utilisé que lorsque **vraiment nécessaire**.
5. Exécutez les requêtes de type **SELECT** avant toutes celles qui modifieront la base de données (les dernières requêtes) ! Sinon les résultats obtenus seront différents de ceux attendus, ou alors pensez à rétablir l'état de la base de données comme au départ, juste après l'importation des fichiers SQL fournis.
6. Tous les mots clé du langage (**SELECT**, **FROM**, **WHERE**, ...) doivent être écrits en **majuscule** et le reste en minuscule. Les requêtes doivent également être formatées et indentées correctement, par exemple:

```
SELECT
    first_name,
    last_name,
    email
FROM customer
WHERE email IS NOT NULL
    AND store_id = (
    SELECT
```

```
MAX(store_id)
FROM store
)
ORDER BY last_name;
```

7. Seules les informations demandées doivent être affichées dans les résultats des requêtes.
8. Il ne doit y avoir **aucun doublon** dans les résultats de n'importe quelle requête (**peu importe le contenu de la base de données**).
9. Lorsque la donnée précise une liste d'attributs pour une requête donnée, ils doivent être présentés dans les résultats de celle-ci, et posséder les mêmes noms. Par exemple une colonne nommée *title* dans le schéma mais attendue en tant que *titre* devra être nommée ainsi dans le résultat.
10. Si le résultat d'une requête produit un grand nombre de lignes, vous ne reporterez que les 20 premières dans le rapport tout en précisant bien le nombre de lignes total (que votre capture d'écran le montre bien).
11. La requête 9 est optionnelle, mais en la réalisant vous pouvez gagner un bonus de 0.2 points à la note finale si cette dernière est correcte.

1.4 Requetes

1. Donnez le numéro, le nom et l'email (*customer_id*, *nom*, *email*) des clients dont le prénom est PHYLLIS, qui sont rattachés au magasin numéro 1, ordonnés par numéro de client décroissant.
2. Donnez l'ensemble des films (*titre*, *annee_sortie*) classés (*rating*) R, ayant une durée de moins de 60 minutes et dont les coûts de remplacements sont 12.99\$, en les ordonnant par titre.
3. Listez le pays, la ville et le numéro postal (*country*, *city*, *postal_code*) des villes française, ainsi que des villes dont le numéro de pays est entre 63 et 67 (bornes comprises), en les ordonnant par pays puis par ville et finalement par code postal. **N'utilisez pas de BETWEEN.**
4. Listez tous les clients actifs (*customer_id*, *prenom*, *nom*) habitant la ville 171, et rattachés au magasin numéro 1. Triez-les par ordre alphabétique des prénoms.
5. Donnez le nom et le prénom (*prenom_1*, *nom_1*, *prenom_2*, *nom_2*) 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 listés).
6. Donnez le nom et le prénom des acteurs (*nom*, *prenom*) ayant joué dans un film d'horreur, dont le prénom commence par K, ou dont le nom de famille commence par D **sans utiliser le mot clé JOIN.**
7. Donnez les films (*id*, *titre*, *prix_de_location_par_jour*) dont le prix de location par jour est inférieur ou égal à 1.00\$ et qui n'ont jamais été loués. Écrire la requête de 2 différentes façons (changer les clauses pour exprimer l'exclusion).
Indication : Le prix de location par jour doit être calculé.
8. Donnez la liste des clients (*id*, *nom*, *prenom*) espagnols qui n'ont pas encore rendu tous les films qu'ils ont empruntés, en les ordonnant par nom. Donnez 3 versions de cette requête :
 - (a) En utilisant **EXISTS** (pas de **GROUP BY**, ni de **IN** ou **NOT IN**).
 - (b) En utilisant **IN** (pas de **GROUP BY**, ni de **EXISTS** ou **NOT EXISTS**).
 - (c) En utilisant aucun des mot-clés précédent (c'est à dire pas de **GROUP BY**, **IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**).
9. **BONUS :** Donnez le numéro, le nom et le prénom (*customer_id*, *prenom*, *nom*) des clients qui ont loué tous les films de l'actrice EMILY DEE.
10. Donnez le titre des films et le nombre d'acteurs (*titre*, *nb_acteurs*) des films dramatiques en les triant par le nombre d'acteur décroissant. Retenez uniquement les films avec moins de 5 acteurs.
11. Listez les catégories (*id*, *nom*, *nb_films*) de films associées à plus de 65 films, sans utiliser de sous-requête, et en les ordonnant par nombre de films.
12. Affichez le(s) film(s) (*id*, *titre*, *duree*) ayant la durée la moins longue. Si plusieurs films ont la même

durée (la moins longue), il faut afficher l'ensemble de ces derniers.

13. Listez les film (**id**, **titre**) dans lesquels jouent au moins un acteur qui a joué dans plus de 40 films, en les ordonnant par titre. Donnez 2 versions de cette requête :
 - (a) En utilisant le mot-clé **IN**
 - (b) Sans utiliser le mot-clé **IN**

Indication : une sous-requête peut être utilisée comme une table, sur laquelle on peut donc effectuer des jointures
14. Un fou furieux décide de regarder l'ensemble des films qui sont présents dans la base de données. Etablissez une requête qui donne le nombre de jours (**nb_jours**) qu'il devra y consacrer sachant qu'il dispose de 8 h par jour.
15. Affichez tous les clients (**id**, **nom**, **email**, **pays**, **nb_locations**, **depense_totale**, **depense_moyenne**) résidant en Suisse, en France ou en Allemagne, dont la dépense moyenne (montant payé) par location est supérieure à 3.0, 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. Ensuite, une nouvelle requête qui ne retourne que les clients dont la dépense moyenne par film est supérieure à 3.0, en utilisant la première requête comme sous-requête. Avant de commencer les requêtes suivantes, sauvegardez la base de données si nécessaire.
16. Donnez les 3 requêtes suivantes ainsi que le résultat de la première et de la dernière :
 - (a) Comptez les paiements dont la valeur est inférieure ou égale à 9.
 - (b) Effacez ces mêmes paiements.
 - (c) Comptez à nouveau ces mêmes paiements pour vérifier que l'opération a bien eu lieu.
17. **En une seule requête**, modifiez les paiements comme suit :
 - (a) Chaque paiement de plus de 4\$ est majoré de 50 %.
 - (b) La date de paiement est mise à jour avec la date courante du serveur.
18. Un nouveau client possédant les informations suivantes souhaite louer des films :

M. Guillaume Ransome

Adresse : Rue du centre, 1260 Nyon

Pays : Suisse / Switzerland

Téléphone : 021/360.00.00

E-mail : gr@bluewin.ch

Ce client est rattaché au magasin 1.

Insérez-le dans la base de données, avec toutes les informations requises pour lui permettre de louer des films.

- (a) Spécifiez les attributs (colonnes) lors de l'insertion.

Indication : plusieurs requêtes sont nécessaires
- (b) Pour chaque nouveau tuple, la base de données doit générer l'id. Pourquoi ne pouvez-vous pas le faire ?
- (c) 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).
- (d) Ecrivez **une** requête d'interrogation, qui montrera que l'ensemble des opérations s'est bien déroulé.

*Indication : **Guillaume Ransome** est un identifiant unique suffisant.*