

# Introduction aux bases de données

## Laboratoire N° 3

**Avertissement:** Les laboratoires sont progressifs. Il est donc fortement recommandé de faire **tous** les points de **tous** les laboratoires pour ne pas rester bloqué par la suite ! !

### 1 Objectifs

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

### Références

- MySQL : <http://easyphp.org/> et <http://dev.mysql.com/doc/refman/5.0/fr/index.html>
- Cours « Bases de données Relationnelles »

### 2 Indications

- Commencer par créer une base de données pour le labo dans MySQL, et importer le schéma et les données. Attention : il faut disposer au minimum de la version 5 de MySQL, et l'importation doit se faire en ligne de commande (plusieurs interfaces graphiques posent des problèmes d'encodage).
- Prenez le temps de bien comprendre le schéma de la base. Référez-vous à la documentation remise.
- Les requêtes sont conçues "à la main".
- Aucun doublon dans les réponses.
- Chaque requête sera donnée correctement formatée. Les 20 premiers tuples du résultat de la requête seront reportés dans le rapport (screenshot, ou résultat obtenu), avec le nombre de tuples total.
- Tous les mots clés SQL (SELECT, FROM, WHERE, ORDER BY,...) sont en majuscule, le reste en minuscule.
- Il n'y a pas de " ou de ' inutile dans les requêtes.
- Seul les informations demandées sont affichées.
- Le mot clé DISTINCT ne sera pas utilisé s'il n'est pas utile.
- Sauf mention contraire, les jointures se font avec le mot clé JOIN (pas dans le WHERE, ni à l'aide d'un IN / NOT IN).
- Toutes les requêtes d'interrogation sur la base seront faites avant les requêtes de modification de données ou de schéma.

Bases de données Relationnelles Prof. Nastaran Fatemi / Sébastien Noir

- Lorsque l'on indique entre parenthèse une liste d'attributs, les attributs correspondant seront listés dans le résultat de la requête, en les renommant si nécessaire.
- Les requêtes sont indentées. Les principales parties de la requête (SELECT, FROM, WHERE, INNER JOIN, LEFT JOIN, ORDER BY, GROUP BY, sous-requêtes,...) commence sur une nouvelle ligne. Exemple (bien formaté, mais pas forcément bien conçu ;-):

```
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;
```

### 3 Requêtes

1. Donner l'ensemble des films (title, release\_year) classés (rating) G durant plus de 100 minutes, dont les coûts de remplacements sont 29.99\$
2. Donner le nom et le prénom (first\_name, last\_name) des clientes nommées TRACY attachées au magasin numéro 1, ordonnées par numéro de client décroissant.
3. Donner les films d'action (film\_id, title) durant moins de 55 minutes en les ordonnant par film\_id croissant.
4. Donner les films (film\_id, title, langue) des films de science fiction dans lesquels joue au moins un acteur dont le prénom est ALAN ou BEN en les ordonnant par numéro de film décroissant.
5. Lister tous les clients actifs (prenom, nom) habitant la ville 321, et attachés au magasin 2. Trier par nom de famille.
6. Lister le pays, la ville, le numéro postal (pays, ville, npa) des villes françaises et des villes dont le numéro postal est entre 50 et 58 (bornes non comprises). Ne pas utiliser BETWEEN. Ordonner par pays, ville, npa.
7. Donner 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.  
Donner deux versions de cette requêtes : en écrivant les jointures à l'aide du mot clé JOIN (standard ainsi), et sans le mot clé JOIN (syntaxe Microsoft), en évitant les sous requêtes.
8. Donner le titre des films (titre) et le nombre d'acteurs (nombre\_acteurs) des films de musique, en les triant pas nombre d'acteur décroissant.
9. Même question, mais la requête ne doit retourner que les films qui utilisent plus de 7 acteurs.

Bases de données Relationnelles Prof. Nastaran Fatemi / Sébastien Noir

10. Lister les catégories (id, nom, nombre de films associés) de films associées à plus de 60 films, sans utiliser de sous requête.
11. Afficher le film (ou les films si plusieurs films ont la même durée minimum) le plus court (id\_min, titre\_min, duree\_min).
12. Lister les acteurs (actor\_id, nombre\_films) qui ont joué dans plus de 35 films, sans utiliser de sous requêtes.
13. Lister les films (id, titre) ordonnés par id dans lesquels joue au moins un acteur qui a joué dans plus de 35 films. Utiliser le mot clé IN.
14. Même question, mais sans utiliser le mot clé IN. Indication : une sous-requête peut être utilisée comme un table, et donc être jointe. Quelle requête est la plus rapide (celle-ci ou la précédente)? A votre avis, pourquoi?
15. Un fou décide de regarder l'ensemble des films qui sont présents dans la base de données. Etablir une requête qui donne le nombre de jours (jours) qu'il devra y consacrer, s'il dispose de 16h par jour.
16. Afficher tous les clients résidant en Inde, au Japon, ou au Maroc, dont la dépense moyenne pour les films loués est supérieure à 3.4. Ordonner par pays puis par nom. Afficher les informations suivantes : id, nom, prenom, pays, nombre\_films\_total, total\_depense, depense\_moyenne.  
Indication : Commencer par établir une requête pour affichant tous les clients avec leur dépense moyenne pour les films loués. Ensuite, créer une nouvelle requête qui ne retourne que les clients dont la dépense moyenne est supérieure à 3.4, en utilisant la requête initiale comme sous-requête.
17. Donner la liste des clients japonais et français (id, nom, prenom, pays) qui n'ont pas encore rendu tous les films qu'ils ont empruntés. Utilisez EXISTS, ne pas utiliser de GROUP BY, ni de IN / NOT IN.
18. Même question. Utiliser IN, ne pas utiliser de GROUP BY, ni de EXISTS / NOT EXISTS.
19. Même question. Ne pas utiliser de GROUP BY, de IN / NOT IN, ni de EXISTS / NOT EXISTS.
20. Avant de commencer les requêtes suivantes, sauvegarder la base si nécessaire. Lister le nombre de paiements dont la valeur supérieure est à 11.  
Effacer ces paiements. Lister à nouveau pour vérifier que l'opération a bien eu lieu.  
Donner les trois requêtes et les résultats de la première et de la troisième.
20. En une seule requête, modifier les paiements comme suit : Chaque paiement de plus de 5 est majoré de 50% et la date de paiement est mise à jour à la date courante du serveur.

Bases de données Relationnelles Prof. Nastaran Fatemi / Sébastien Noir

21. Insérez le nouveau client actif dans la base, avec toutes les informations requises pour que vous puissiez louer des films.

Indications : 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 la valeur. On considère que l'ensemble des requêtes nécessaires sera fait dans une transaction : ainsi, les seules modifications dans les données de la base pendant la transaction seront les vôtres (pas de risque que les données soient modifiées par une autre application pendant l'opération). Finalement, écrire UNE requête d'interrogation qui montrera que l'ensemble des opérations s'est bien déroulé (considérer que "Marcel Rochat" est un identifiant suffisant).

Monsieur Marcel Rochat  
Rue du centre  
1260 Nyon  
022 360 00 00  
Suisse / Switzerland  
mr@bluewin.ch  
Attaché au store numéro 1