

Exercices SQL

1) Enoncés.....	2
1.1) Exercices simples sur la projection	2
1.2) Exercices simples sur la restriction	2
1.3) Premiers exercices avec jointures.....	2
1.4) Premiers exercices avec fonctions de calcul	3
1.5) Quelques exercices piégés ou difficiles sur les jointures	3
1.6) Exercices sur les fonctions de MySQL.....	4
1.7) Exercices sur les UPDATE, DELETE et INSERT	5
1.8) Exercices sur les jointures externes	6
2) Corrections.....	6
2.1) Exercices simples sur la projection	6
2.2) Exercices simples sur la restriction	7
2.3) Premiers exercices avec jointures.....	9
2.4) Premiers exercices avec fonctions de calcul	10
2.5) Exercices sur les fonctions de MySQL.....	12
2.6) Exercices sur les UPDATE, DELETE et INSERT	14
2.7) Exercices sur les jointures externes	15

Tous les exercices ne sont pas corrigés, mais n'hésitez pas à regarder les corrections des autres exercices pour trouver des idées ou vous rappeler la syntaxe...

1) Enoncés

1.1) Exercices simples sur la projection

- 1) FILM : Donner la liste de tous les acteurs.
- 2) FILM : Donner la liste des films.
- 3) HOTEL : Donner le nom et le prénom de tous les clients, et les trier par ordre alphabétique. (Le tri par ordre alphabétique se fait simplement en ajoutant à la fin de la requête une clause *ORDER BY champ1, champ2*, qui triera premièrement par *champ1*, puis par *champ2*).
- 4) HOTEL : Afficher le nombre de lits dans chaque chambre.
- 5) LIBRAIRIE : Lister les auteurs.
- 6) LIBRAIRIE : lister les livres ainsi que leur prix.
- 7) UNIVERSITE : Donner la liste des étudiants avec leur adresse complète.
- 8) UNIVERSITE : Donner la liste des étudiants par ordre de naissance.

1.2) Exercices simples sur la restriction

- 1) FILM : Donner la liste de tous les films sortie avant l'année 2000.
- 2) FILM : Afficher tous les acteurs dont le nom commence par la lettre V. (Pour comparer une chaîne de caractère, on peut utiliser l'opérateur LIKE, qui s'utilise comme ceci : ... *WHERE champ LIKE 'chaîne de caractères' ...* . Le symbole "%" signifie "n'importe quel caractère". Par exemple, *titre LIKE "%2"* nous donnera tous les enregistrements dont le champ titre finit par 2.)
- 3) FILM : Sélectionner tous les rôles dont le nom contient la chaîne de caractères "de".
- 4) HOTEL : Sélectionner toutes les réservations dont la date est inférieure au 1er janvier 2005.
- 5) HOTEL : Sélectionner toutes les chambres qui ont la télévision.
- 6) LIBRAIRIE : Sélectionner les livres dont la quantité en stock est inférieure à 5.
- 7) LIBRAIRIE : Sélectionner les livres dont la quantité en stock est inférieure à 3 et le prix est supérieur à 20.
- 8) LIBRAIRIE : Sélectionner les livres dont le titre commence ou se termine par la lettre t.
- 9) UNIVERSITE : Sélectionner les étudiants qui sont nés après 1974.
- 10) UNIVERSITE : Sélectionner les modules quand le champ description est vide. (Pour vérifier si un champ a une valeur ou pas, on peut utiliser la syntaxe *champ IS NULL* ou *champ IS NOT NULL*).

1.3) Premiers exercices avec jointures

- 1) LIBRAIRIE : Lister les livres écrits par Albert BLAKE.
- 2) LIBRAIRIE : Lister les romans.
- 3) LIBRAIRIE : Donner la liste des livres édités par Larousse.
- 4) LIBRAIRIE : Pour chaque livre, donner le titre, l'auteur, le genre et l'éditeur.
- 5) LIBRAIRIE : Quel est le genre des livres écrits par Marcel PROUST ?
- 6) FILM : Quels rôles a joué Christian CLAVIER ?
- 7) FILM : Dans quel(s) film(s) a joué Gérard DEPARDEU ?
- 8) FILM : Quels rôles sont répertoriés pour "Les Visiteurs" ?
- 9) FILM : Quels rôles sont répertoriés pour "Les Visiteurs" ou "Les visiteurs 2, les couloirs du temps" ? (faire la restriction dans un premier temps avec %, puis avec un OR).
- 10) FILM : Dans quels films intervient le personnage de Godefroy de Montmirail ?
- 10,5) UNIVERSITE : Sélectionner les enseignants qui s'occupent d'un module du semestre S1.
- 11) UNIVERSITE : Sélectionner tous les étudiants inscrits à au moins 1 module du semestre S1.
- 12) UNIVERSITE : Voir toutes les notes obtenues au module 1.
- 13) UNIVERSITE : Extraire la date, le type et le coefficient de chaque évaluation passée par l'étudiant Jules Cesar, ainsi que ses notes.
- 14) UNIVERSITE : Lister toutes les notes obtenues par Tryphon TOURNESOL.
- 15) SONDAGE : Donner l'âge et le sexe de chacun des sondés.
- 16) SONDAGE : Sélectionner l'âge des personnes qui mangent très souvent des bananes.
- 17) SONDAGE : Sélectionner les fréquences de lecture de bandes dessinées des hommes.

1.4) Premiers exercices avec fonctions de calcul

- 1) UNIVERSITE : Combien y a-t-il de modules ?
- 2) UNIVERSITE : Combien y a-t-il de modules dont le nom commence par la lettre "p" ?
- 3) UNIVERSITE : Combien y a-t-il de modules au semestre S1 ?
- 4) UNIVERSITE : Combien y a-t-il d'étudiants inscrits au module équitation ?
- 5) UNIVERSITE : Combien y a-t-il d'étudiants dans chaque module ?
- 6) UNIVERSITE : Quelle est la moyenne des notes obtenues à l'évaluation du 2005-04-03 ?
- 7) UNIVERSITE : Quelle est la moyenne des notes obtenues à chacune des évaluations de chaque module (pour l'instant, seuls les modules qui ont des évaluations vont s'afficher. Nous verrons par la suite comment faire pour afficher toutes les évaluations (left join)) ?
- 8) UNIVERSITE : Quelle est la note maximale pour l'évaluation 1 ?
- 9) UNIVERSITE : Quelle est la moins bonne des notes obtenue par Tryphon TOURNESOL ?
- 10) LIBRAIRIE : Quelle est le prix moyen des livres ?
- 11) LIBRAIRIE : Quelle est le prix moyen des livres édités chez Larousse ?
- 12) LIBRAIRIE : Si on achète 1 exemplaire de chacun des livres, combien est-ce que ça va nous coûter ?
- 13) LIBRAIRIE : Si on achète 1 exemplaire de chacun des livres, combien est-ce que ça va nous coûter, sachant que l'on obtient alors une remise de 20% sur l'ensemble ?
- 14) LIBRAIRIE : Quelle est la valeur des livres en stock ?

1.5) Quelques exercices piégés ou difficiles sur les jointures

Les exercices de ce paragraphe correspondent à la base de données **GRH**.

- 1) Sélectionner tous les employés qui sont en CDI (attention au piège : vous aurez besoin des [fonctions sur les dates et heures](#))
- 2) Sélectionner tous les employés qui sont en CDD (attention aux pièges !)
- 3) Déterminer le nom, le prénom et la fonction de chaque employé
- 4) Déterminer les type de contrat des secrétaires
- 5) Quel est le numéro de téléphone du responsable informatique ?
- 6) Qui sont les subordonnés d'Aladdin ?
- 7) Qui est le chef de Bruce Lee ?
- 8) Quelle est la fonction du chef de Jennifer ?
- 9) Déterminer le nom, le prénom et la fonction du responsable hiérarchique de chaque employé (Le PDG est son propre chef).
- 10) Déterminer quel est le salaire le plus bas actuellement donné dans l'entreprise. (attention aux dates !)
- 11) Déterminer le nombre de contrats qu'a eu chaque employé
- 12) Compter le nombre de personnes en CDD travaillant actuellement dans l'entreprise. (Il faut en trouver 2 !)
- 13) Quels employés sont arrivés dans l'entreprise au mois de février ? (Quelle que soit l'année, on veut que le mois d'arrivée soit le mois de février)

1.6) Exercices sur les fonctions de MySQL

Utilisez pour ces questions les fonctions disponibles dans la rubrique [12. Fonctions à utiliser dans les clauses SELECT et WHERE](#).

Dans la base de données LIBRAIRIE :

- 1) Afficher les données de la table Auteur au format csv (Des points-virgules séparent les champs) :

xxx
2;Christie;Agatha
3;Pagnol;Marcel

- 2) Donnez le nom de chaque auteur, ainsi que la version **cryptée en md5** de ce nom (exercice idiot, mais cette fonction sera utile pour des mots de passe par exemple...).

- 3) Sélectionner le titre de chaque livre, ainsi que le nombre de caractères qui constituent ce livre

- 4) Sélectionner les noms des auteurs, en remplaçant les « a » par des « @ »

- 5) Dites quel utilisateur est actuellement connecté à la base (session en cours)

- 6) Donnez la date et l'heure actuelles du serveur.

- 7) Sélectionner la date et l'heure du serveur, ainsi que la même date/heure + 1 semaine. Exemple de résultat :

xxx	xxx
2006-02-01 12 :05 :32	2006-02-08 12 :05 :32

Dans la base de données GRH :

- 8) Quelle est la date d'arrivée de Georges WASHINGTON ? Donner cette date au format français, c'est-à-dire jj/mm/aaaa

- 9) Lister les employés par ordre aléatoire.

- 10) Lister les employés sous la forme suivante (1 seule colonne, et le caractère séparateur entre le nom et le prénom est un espace) :

nom complet
NOM Prénom

- 11) Lister les fonctions, en les mettant en lettres majuscules.

- 12) Donner la moyenne des salaires, arrondie à 1 chiffre après la virgule.

1.7) Exercices sur les UPDATE, DELETE et INSERT

- 1) HOTEL : Ajouter la télévision dans toutes les chambres qui n'en ont pas.
- 2) HOTEL : Mettre 2 lits dans toutes les chambres qui n'en ont qu'un.
- 3) HOTEL : Annuler toutes les réservations.
- 4) HOTEL : Supprimer toutes les réservations d'Homer Simpson.
- 5) LIBRAIRIE : Mettre à 10 € tous les livres qui ont un prix inférieur à 8 €.
- 6) LIBRAIRIE : Doubler la quantité des romans.
- 7) LIBRAIRIE : Supprimer tous les livres édités par Gallimard.
- 8) LIBRAIRIE : Supprimer les livres qui contiennent 2 fois la lettre e.
- 9) LIBRAIRIE : insérer un nouveau livre avec une commande INSERT.
- 10) LIBRAIRIE : insérer dans une seule commande INSERT 2 nouveaux éditeurs.
- 11) LIBRAIRIE : créer une nouvelle table à l'aide de ces requêtes :

```
DROP TABLE IF EXISTS auteur_copie ;
CREATE TABLE auteur_copie (
  cle int(11) NOT NULL auto_increment,
  nom varchar(50) default NULL,
  prenom varchar(50) default NULL,
  PRIMARY KEY (cle)
) ;
```

- 11.1) Copier en une seule commande SQL les enregistrements de la table auteur vers la table auteur_copie. (la réponse est en bas de page)
- 11.2) Réinitialiser la table auteur_copie à l'aide des commandes DROP TABLE et CREATE TABLE ci-dessus. Ensuite, insérer dans cette table uniquement les auteurs dont le nom de famille commence par un S.
- 11.3) Réinitialiser la table auteur_copie à l'aide des commandes DROP TABLE et CREATE TABLE ci-dessus. Ensuite, insérer dans cette table les auteurs qui ont écrit au moins 1 livre > 22 €.

- 12) HOTEL : Insérer dans la table historique le nombre de chambres réservées pour chaque nuit.

Pour réinitialiser la table, utilisez cette commande :

```
DELETE FROM historique ;
```

- 13) HOTEL : Insérer dans la table historique le nombre de chambres avec télévision réservées pour chaque nuit

- 14) GRH : Insérer dans la table historique tous les employés qui ont un contrat terminé (sauf informations de salaire et de type de contrat)

Pour réinitialiser la table, utilisez cette commande :

```
DELETE FROM historique ;
```

- 15) GRH : Insérer dans la table historique tous les employés qui ne sont plus dans l'entreprise (sauf informations de salaire et de type de contrat)

- 16) GRH : Insérer dans la table historique tous les employés qui ne sont plus dans l'entreprise, ainsi que leur dernier salaire et type de contrat (! Sous-requête)

- 17) PIZZERIA : augmenter de 1 € le prix des pizzas qui contiennent des champignons

- 18) PIZZERIA : Tripler le prix des boissons

- 19) UNIVERSITE : Augmenter de 1 point toutes les notes inférieures à la moyenne

- 20) HOTEL : Ajouter 1 lit et 1 télévision dans toutes les chambres réservées par Homer Simpson

- 21) LIBRAIRIE : Doubler la quantité et le prix des livres dont la quantité est supérieure à 15

22) LIBRAIRIE : Supprimer tous les livres dont le nom du livre, le nom de l'auteur ou le nom de l'éditeur commence par un L

1.8) Exercices sur les jointures externes

1) LIBRAIRIE : Sélectionner tous les auteurs, en donnant également la liste de leurs livres quand ils en ont écrit.

2) LIBRAIRIE : Sélectionner tous les livres, en donnant également, quand c'est précisé, l'auteur. Auparavant, vous aurez besoin d'exécuter ces requêtes, pour observer quelque chose :

```
DELETE FROM auteur WHERE cle = 8 ;  
UPDATE livre SET cle_auteur = NULL WHERE cle_auteur = 8 ;
```

3) HOTEL : Sélectionner toutes les chambres, et lister les noms de ceux qui les ont réservées.

4) HOTEL : Compter le nombre de réservations par chambre. Celles qui n'ont jamais été réservées doivent afficher 0.

5) HEBERGEUR : pour chaque extension de domaine, donner la liste des pays des clients, en laissant apparaître toutes les extensions.

6) HEBERGEUR : Lister les offres d'hébergement, en donnant également les clients associés.

2) Corrections

2.1) Exercices simples sur la projection

1)
`SELECT nom FROM acteur ;`

2)
`SELECT nom FROM film ;`

3)
`SELECT
 nom, prenom
FROM client
ORDER BY nom ;`

4)
`SELECT
 numero, lit
FROM chambre
ORDER BY numero ;`

5)
`SELECT
 nom, prenom
FROM auteur
ORDER BY nom ;`

```
6)
SELECT
    titre, prix
FROM livre
ORDER BY titre ;
```

```
7)
SELECT
    nom, prenom, adresse, code_postal, ville, pays
FROM etudiant
ORDER BY nom, prenom ;
```

```
8)
SELECT
    nom, prenom, date_de_naissance
FROM etudiant
ORDER BY date_de_naissance ;
```

2.2) Exercices simples sur la restriction

```
1)
SELECT
    nom, annee
FROM film
WHERE
    annee < 2000
ORDER BY nom ;
```

```
2)
SELECT
    nom
FROM acteur
WHERE
    nom LIKE 'v%'
ORDER BY nom ;
```

```
3)
SELECT
    nom
FROM role
WHERE
    nom LIKE '%de%'
ORDER BY nom ;
```

```
4)
SELECT
    *
FROM reservation
WHERE
    date_reservation < '2005-01-01' ;
```

```
5)
SELECT
    numero
FROM chambre
WHERE
    television > 0
ORDER BY numero ;
```

```
6)
SELECT
    titre
FROM livre
WHERE
    quantite < 5
ORDER BY titre ;
```

```
7)
SELECT
    titre
FROM livre
WHERE
    quantite < 3 AND
    prix > 20
ORDER BY titre ;
```

```
8)
SELECT
    titre
FROM livre
WHERE
    titre LIKE 't%' OR
    titre LIKE '%t'
ORDER BY titre ;
```

```
9)
SELECT
    nom, prenom
FROM etudiant
WHERE
    date_de_naissance > '1974-12-31'
ORDER BY nom, prenom ;
```

```
10)
SELECT
    nom, description
FROM module
WHERE
    description IS NULL
ORDER BY nom ;
```

2.3) Premiers exercices avec jointures

```
1)
SELECT
    titre
FROM
    auteur, livre
WHERE
    auteur.nom = 'BLAKE' AND
    auteur.prenom = 'Albert' AND
    auteur.cle = livre.cle_auteur ;

2)
SELECT
    titre
FROM
    genre, livre
WHERE
    genre.nom = 'Roman' AND
    genre.cle = livre.cle_genre ;

3)
SELECT
    titre
FROM
    editeur, livre
WHERE
    editeur.nom = 'Larousse' AND
    editeur.cle = livre.cle_editeur ;

4)
SELECT
    titre, auteur.nom, auteur.prenom, genre.nom, editeur.nom
FROM
    editeur, genre, auteur, livre
WHERE
    editeur.cle = livre.cle_editeur AND
    auteur.cle = livre.cle_auteur AND
    genre.cle = livre.cle_genre ;

5)
SELECT DISTINCT
    genre.nom
FROM
    genre, auteur, livre
WHERE
    auteur.nom = 'Proust' AND
    auteur.prenom = 'Marcel' AND
    auteur.cle = livre.cle_auteur AND
    genre.cle = livre.cle_genre ;
```

Le DISTINCT sert à ne pas afficher 2 fois la même ligne.

```
6)
SELECT DISTINCT
    film.nom
FROM
    jouer, acteur, film
WHERE
    acteur.nom = 'Christian Clavier' AND
    acteur.numActeur = jouer.numActeur AND
    film.numFilm = jouer.numFilm ;
```

...

```
10,5)
SELECT DISTINCT
    enseignant.nom, enseignant.prenom
FROM enseignant, module, semestre
WHERE
    semestre.nom = "S1" AND
    semestre.num_semestre = module.num_semestre AND
    module.num_enseignant = enseignant.num_enseignant
ORDER BY enseignant.nom, prenom ;
```

2.4) Premiers exercices avec fonctions de calcul

```
1)
SELECT
    COUNT(*)
FROM
    module ;
```

```
2)
SELECT
    COUNT(*)
FROM
    module
WHERE
    nom LIKE 'p%' ;
```

```
3)
SELECT
    COUNT(*)
FROM
    module, semestre
WHERE
    semestre.nom = 'S1' AND
    module.num_semestre = semestre.num_semestre ;
```

```
4)
SELECT
    COUNT(*)
FROM
    module, inscription, etudiant
WHERE
    module.nom LIKE 'equitation' AND
    inscription.num_module = module.num_module AND
    etudiant.num_etudiant = inscription.num_etudiant ;
```

```
5)
SELECT
    module.nom, COUNT(etudiant.num_etudiant)
FROM
    module, inscription, etudiant
WHERE
    inscription.num_module = module.num_module AND
    etudiant.num_etudiant = inscription.num_etudiant
GROUP BY
    module.nom ;
```

...

```
7)
SELECT
    module.nom, evaluation.date, evaluation.type_evaluation,
    AVG(note.note)
FROM
    note, evaluation, module
WHERE
    note.num_evaluation = evaluation.num_evaluation AND
    evaluation.num_module = module.num_module
GROUP BY
    module.cle, evaluation.cle ;
```

...

```
12)
SELECT
    SUM(prix)
FROM
    livre ;
```

```
13)
SELECT
    SUM(prix) * 0.8
FROM
    livre ;
```

```
14)
SELECT
    SUM(prix * quantite)
FROM
    livre ;
```

2.5) Exercices sur les fonctions de MySQL

1)
1^{ère} solution :
SELECT
 concat(cle, ";", nom, ";", prenom)
FROM
 auteur ;

2^{ème} solution :
SELECT
 concat_ws(";", cle, nom, prenom)
FROM
 auteur ;

2)
SELECT
 nom, md5(nom)
FROM
 auteur ;

3)
SELECT
 titre, length(titre)
FROM
 livre ;

4)
SELECT
 nom, replace(nom, "a", "@")
FROM
 auteur ;

5)
SELECT
 CURRENT_USER() ;

6)
SELECT
 NOW() ;

La fonction NOW() donne la date et l'heure.
La fonction CURDATE() donne la date uniquement.
La fonction CURTIME() donne l'heure uniquement.

7)
1^{ère} solution :
SELECT
 now(), date_add(now(), interval 7 day) ;

2^{ème} solution (fonctionne à partir de MySQL 5, mais pas avec MySQL 4
inclus dans EasyPHP) :
SELECT
 now(), date_add(now(), interval 1 week) ;

8)

1^{ère} étape : faire la requête normale, avec la date au format standard (américain) :

```
SELECT
    date_debut
FROM
    personnel, contrat
WHERE
    nom = "washington" AND
    prenom = "george" AND
    personnel.id_personnel = contrat.id_personnel ;
```

-> résultat = 2005-03-01

2^{ème} étape : transformer cette date pour qu'elle s'affiche au format jj/mm/aaaa :

```
SELECT
    date_format(date_debut, "%d/%m/%Y")
FROM
    personnel, contrat
WHERE
    nom = "washington" AND
    prenom = "george" AND
    personnel.id_personnel = contrat.id_personnel ;
```

-> résultat = 01/03/2005

Certains format sont prédéfinis, et sont accessible avec GET_FORMAT. On aurait pu l'utiliser ici :

```
SELECT
    date_format(date_debut, GET_FORMAT(DATE, 'EUR'))
FROM
    personnel, contrat
WHERE
    nom = "washington" AND
    prenom = "george" AND
    personnel.id_personnel = contrat.id_personnel ;
```

-> resultat = 01.03.2005

9) Voir la [section sur la fonction RAND\(\)](#)

10) et 11) Il faut utiliser une [fonction de chaîne de caractères](#)

12)
1^{ère} étape : Déterminer la moyenne des salaires, sans s'occuper de l'arrondi :
SELECT
 avg(salaire_brut)
FROM
 contrat ;

2^{ème} étape : On demande d'arrondir avec la fonction round() :
SELECT
 round(avg(salaire_brut) **,** 2)
FROM
 contrat ;

2.6) Exercices sur les UPDATE, DELETE et INSERT

6)
UPDATE
 livre INNER JOIN genre ON livre.cle_genre = genre.cle
SET
 quantite = quantite * 100
WHERE
 genre.nom = 'roman' ;

7) DELETE livre
FROM
 livre INNER JOIN editeur ON livre.cle_editeur = editeur.cle
WHERE
 editeur.nom = 'Larousse' ;

...

11.1)
INSERT INTO auteur_copie SELECT * FROM auteur ;

11.2)
INSERT INTO auteur_copie SELECT * FROM auteur WHERE auteur.nom LIKE 's%';

11.3)
INSERT INTO
 auteur_copie
SELECT auteur.*
FROM
 auteur, livre
WHERE
 livre.prix > 22 AND
 auteur.cle = livre.cle_auteur ;

```
12)
INSERT INTO historique (nbchambre, nuit)
SELECT count(*), date_reservation
FROM reservation
GROUP BY date_reservation ;
```

2.7) Exercices sur les jointures externes

```
1)
SELECT
    nom, prenom, titre
FROM
    auteur LEFT JOIN livre ON auteur.cle = cle_auteur
ORDER BY nom, prenom, titre ;

2)
SELECT
    titre, nom, prenom
FROM
    livre LEFT JOIN auteur ON auteur.cle = cle_auteur
ORDER BY titre, nom, prenom ;

3)
SELECT
    numero, nom, prenom
FROM
    (chambre LEFT JOIN reservation ON chambre.cle =
reservation.cle_chambre)
    LEFT JOIN client ON reservation.cle_client = client.cle ;

4)
SELECT
    numero, count(reservation.cle) as nb_reservation
FROM
    chambre LEFT JOIN reservation ON chambre.cle =
reservation.cle_chambre
GROUP BY numero ;

5)
SELECT
    extension, pays.nom
FROM
    (
        (
            (
                offre_domaine LEFT JOIN domaine USING(offre_domaine)
            ) LEFT JOIN client_domaine USING(domaine)
        ) LEFT JOIN client USING(client)
    ) LEFT JOIN pays USING(pays)
ORDER BY extension ;
```

```
6)
SELECT
    offre_hebergement.nom as offre, client.nom, prenom, hebergement.nom
as nom_compte
FROM
    (
        (
            offre_hebergement LEFT JOIN hebergement USING(offre_hebergement)
        ) LEFT JOIN client_hebergement USING(hebergement)
    ) LEFT JOIN client USING(client)
ORDER BY offre_hebergement.nom, client.nom, client.prenom ;
```