

# Exercices SQL



*Johnny Piette*

---

## Table des matières

- [Introduction](#)
- [Copier les DB exercices](#)
- [Exercice n° 1 - Importation des fichiers de base de données](#)
- [Exercice n° 2 - Notre premier SELECT \\*](#)
- [Exercice n° 3 - SELECT ch1, ch2, etc.](#)
- [Exercice n° 4 - Notre premier WHERE](#)
- [Exercice n° 5 - WHERE AND](#)
- [Exercice n° 6 - WHERE IS NULL IS NOT NULL](#)
- [Exercice n° 7 - WHERE AND](#)
- [Exercice n° 8 - WHERE sur une DATE](#)
- [Exercice n° 9 - WHERE sur des DATES](#)
- [Exercice n° 10 - WHERE avec LIKE '%quelquechose' = se termine par 'quelquechose'](#)
- [Exercice n° 11 - WHERE avec LIKE 'quelquechose%' = commence par 'quelquechose'](#)

- Exercice n° 12 - WHERE avec LIKE '%quelquechose%' = contient 'quelquechose'
- Exercice n° 13 - WHERE AND OR
- Exercice n° 14 - ORDER BY
- Exercice n° 15 - ORDER BY
- Exercice n° 16 - AVG
- Exercice n° 17 - MIN, MAX, SUM
- Exercice n° 18 - GROUP BY
- Exercice n° 19 - GROUP BY
- Exercice n° 20 - GROUP BY
- Exercice n° 21 - GROUP BY
- Exercice n° 22 - INNER JOIN
- Exercice n° 23 - INNER JOIN
- Exercice n° 24 - INNER JOIN
- Exercice n° 25 - INNER JOIN
- Exercice n° 26 - CREATE DATABASE / CREATE TABLE
- Exercice n° 27 - CREATE DATABASE / CREATE TABLE
- Exercice n° 28 - INSERT INTO
- Exercice n° 29 - INSERT INTO
- Exercice n° 30 - UPDATE
- Exercice n° 31 - UPDATE
- Exercice n° 32 - UPDATE
- Exercice n° 33 - DELETE
- Exercice n° 34

## Introduction

Toute commande dans mysql doit se terminer par un point virgule ;

Si vous ne le faites pas mysql ne sait pas que vous avez terminé la commande en cours.

Pour savoir dans quelle base de données vous êtes actuellement, on va customiser notre prompt.

Dans la console mysql, tapez:

```
prompt <\d>
```

# Copier les DB exercices

Copier les fichiers SQL se trouvant via cette adresse: [Base de données des exercices](#)

Décompressez ce fichier DB.zip et mettez le répertoire DB dans votre dépôt Eqla-Exercices.

Demandez-moi le mot de passe pour décompresser le fichier.

## Exercice n°1 - Importation des fichiers de base de données

1. Allez dans le répertoire d'exercices SQL

2. Connectez-vous au SGBD MySQL:

```
mysql -u root -p
```

3. Entrez votre mot de passe

4. Importez le script qui va générer la base de données BlindCode:

```
source BlindCode.sql
```

5. Importez le script qui va générer la base de données Localites (Ca va prendre plus temps: attendez):

```
source Localites.sql
```

6. Importez le script qui va générer la base de données Ventes:

```
source Ventes.sql
```

7. Importez le script qui va générer la base de données Ventes:

```
source Pays.sql
```

8. Importez le script qui va générer la base de données BlindCode2:

```
source BlindCode2.sql
```

9. Importez le script qui va générer la base de données Exercices:

```
source Exercices.sql
```

10. Pour avoir la liste des bases de données présentes: `SHOW DATABASES;`
11. Vous devriez voir dans la liste des bases de données: BlindCode, BlindCode2, Exercices, Localites, Ventes, Pays
12. Si vous avez des erreurs appelez-moi.

## Exercice n°2 - Notre premier SELECT \*

1. Connectez-vous au SGBD MySQL: **mysql -u root -p**
2. Entrez votre mot de passe.
3. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
4. Ecrivez l'instruction qui affichez tous les enregistrements de la table Eleve: tous les champs à l'aide du caractère \*
5. Ecrivez l'instruction qui affichez tous les enregistrements de la table formation: tous les champs à l'aide du caractère \*

## Exercice n°3 - SELECT ch1, ch2, etc.

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez tous les enregistrements de la table Eleve avec les champs suivants: Nom, Prenom, Email
6. Affichez tous les enregistrements de la table formation avec les champs: Nom, Lieux

## Exercice n°4 - Notre premier WHERE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez tous les élèves de sexe Féminin.

6. Affichez tous les élèves de sexe Masculin.
7. Affichez Le nom, le prénom, la localite des élèves qui habitent sur le CP 6890.

## Exercice n°5 - WHERE AND

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez tous les élèves de Sexe Masculin ET ayant comme CP 1348

## Exercice n°6 - WHERE IS NULL IS NOT NULL

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez tous les élèves qui n'ont pas de numéro de téléphone: On utilisera IS NULL
6. A l'inverse, affichez les élèves qui ont un numéro de téléphone: On utilisera IS NOT NULL: On peut faire autrement, rappelez-le moi 😊

## Exercice n°7 - WHERE AND

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les élèves dont le code postal est dans les 4000

## Exercice n°8 - WHERE sur une DATE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe

4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les élèves nés après 1980: Le format de la date est année/Mois/Jour
6. Affichez les élèves nés avant 1980: Le format de la date est année/Mois/Jour

## Exercice n°9 - WHERE sur des DATES

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les élèves nés entre 1970 ET 1980.
6. Affichez les élèves qui ne sont pas nés entre 1970 ET 1980.

## Exercice n°10 - WHERE avec LIKE

**'%quelquechose' = se termine par 'quelquechose'**

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les élèves qui ont une adresse de type [gmail.com](#)
6. Affichez les élèves qui n'ont pas une adresse de type [gmail.com](#)

## Exercice n°11 - WHERE avec LIKE

**'quelquechose%' = commence par 'quelquechose'**

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les élèves qui n'ont pas une adresse de type [gmail.com](#)

## Exercice n°12 - WHERE avec LIKE

**'%quelquechose%' = contient 'quelquechose'**

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les élèves dont le nom commence par 'be'

## Exercice n°13 - WHERE AND OR

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB Localite, tapez: **use Localite;**
5. Affichez toutes les communes sur le code Postal 4280
6. Affichez toutes les communes sur le code Postal 4280 et dont la commune commence par 'b' ou par 'a'
7. Affichez les CP et les noms des communes de la Province de Liège: les CP sont compris entre 4000 et 4999

## Exercice n°14 - ORDER BY

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez les champs: Prenom, Nom, Sexe, Naissance et triez dans l'ordre alphabétique sur le nom et puis le prenom.
6. Ensuite, triez dans l'ordre inverse le nom.
7. Triez les élèves du plus âgé au plus jeune.
8. Ensuite, triez dans l'ordre inverse.

## Exercice n° 15 - ORDER BY

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. On va ici créer une colonne dynamiquement dans le résultat qu'on nommera Age. On utilisera le AS comme vu au cours dans la Exercice n° "champs calculés"  
Je vous demande d'afficher le prenom, nom et l'age des eleves.  
Pour cela, je vous rappelle que la fonction YEAR() permet de récupérer l'année d'une date et que CURDATE() donne la date du jour. Pour avoir l'age approximatif de l'élève on pourrait donc faire Année en cours(2022)-Année de naissance(ex:1974) AS Age
6. Reprenez la requête précédente et affichez les élèves du plus jeune au plus vieux.
7. Ensuite, du plus vieux au plus jeune.
8. Réécrivez vos deux requêtes pour ne traiter que des élèves de BlindCode4Data.

## Exercice n° 16 - AVG

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Reprenez la requête de la Exercice n° 15.
6. Modifiez-la pour que l'on ait l'age moyen des élèves.

## Exercice n° 17 - MIN, MAX, SUM

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB Ventes, tapez: **use ventes;**
5. Affichez le prix maximum des produits
6. Affichez le prix minimum des produits



7. Affichez le prix total de tous les produits en tenant du stock de chaque produit. ☺

## Exercice n° 18 - GROUP BY

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe
4. Si vous n'êtes pas dans la DB Ventes, tapez: **use ventes;**
5. Affichez pour chaque catégorie, son stock. (Indice utilisez aussi SUM)

## Exercice n° 19 - GROUP BY

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez pour chaque sexe, le nombre total. (Indice COUNT)
6. Affichez pour chaque nationalite, le nombre total.

## Exercice n° 20 - GROUP BY

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Pays, tapez: **use Pays;**
5. Faites un DESC Pays, pour voir sa structure.
6. Affichez pour chaque continent, combien il a de pays. On n'utilisera qu'une seule table: Pays. Donc pour le continent à afficher on prendra le champ continent. (On fera mieux plus tard.)

## Exercice n° 21 - GROUP BY

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**

3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB BlindCode2, tapez: **use BlindCode2;**
5. Expliquez ce que fait cette commande SQL:

```
SELECT YEAR(DateInscription) As 'Annee Inscription', COUNT(DateInscription)
FROM eleve_formation
group by YEAR(DateInscription);
```

6. Pourquoi avoir utilisé la Fonction **YEAR** ? Pour vous aider, comparez sans le **YEAR**:

```
SELECT DateInscription As 'Annee Inscription', COUNT(DateInscription)
FROM eleve_formation
group by DateInscription;
```

## Exercice n° 22 - INNER JOIN

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Pays, tapez: **use Pays;**
5. Affichez chaque pays et le nom du continent dont il fait partie.
6. Faites un tri ascendant sur le nom du pays.
7. Je vous conseille d'utiliser le mot clef **AS** après le nom du pays et après le nom du continent. Sinon vous allez avoir deux colonnes Name pour le pays et le continent. Ce qui n'est pas très pratique...

## Exercice n° 23 - INNER JOIN

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Pays, tapez: **use Pays;**
5. Affichez le nom COMPLET du pays et le nom COMPLET du continent (pas l'acronyme) dont il fait partie.

## Exercice n°24 - INNER JOIN

Ici, on va travailler sur trois tables. La table *eleve*, la table *formation* et la table *eleve\_formation*.

Analyser la description des tables pour comprendre comment les lier entre elles.

Pour connaître la description d'une table, on utilise la commande: **DESC nomdelatable;**

No stress, si vous n'y arrivez pas. On va le faire ensemble de toute manière 😊

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB BlindCode2, tapez: **use BlindCode2;**
5. On va afficher le nom et prénom de l'élève, le nom de la classe et la date d'inscription.

## Exercice n°25 - INNER JOIN

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB BlindCode, tapez: **use BlindCode;**
5. Affichez le nom des différentes classes/formations et le nombre d'élèves dans chacune d'elle. La colonne du nombre d'élèves s'appellera NBElevés.
6. Faites un tri descendant sur le nombre d'élèves.

## Exercice n°26 - CREATE DATABASE / CREATE TABLE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Créez une base de données nommée: Exercice26
5. Utilisez cette nouvelle de données avec la commande USE Exercice26;
6. Créer une Table Equipe. Cette table aura les champs suivants:
  - IdEquipe (clef primaire): Entier, NOT NULL, non signé, auto-incrémenté.

- NomClub: chaîne de max 30 caractères, NOT NULL.
- Localité: chaîne de max 30 caractères, NOT NULL.
- Division: TINYINT non signé, NOT NULL.

## Exercice n°27 - CREATE DATABASE / CREATE TABLE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Trouvez les commandes sql suivantes:
  - Si la base de données Exercice27 existe la supprimer.
  - Créez une base de données nommée: Exercice27
  - Utilisez cette nouvelle base de données avec la commande USE Exercice27;
5. Recopiez tout le CREATE TABLE Equipe de l'exercice26.
6. Créer la Table Joueur. Cette table aura les champs suivants:
  - IdJoueur (clef primaire): Entier, NOT NULL, non signé, auto-incrémenté.
  - Nom: chaîne de max 30 caractères, NOT NULL.
  - Prenom: Chaîne de max 30 caractères, NOT NULL.
  - DateNaissance: Date, NOT NULL.
  - IdEquipe (clef étrangère): entier NOT NULL, non signé, Référence IdEquipe de la table Equipe.

## Exercice n°28 - INSERT INTO

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **use Exercice27;**
5. Ajouter l'équipe suivante:
  - NomClub: Royal Club d'Andenne
  - Localité: Andenne
  - Division: 2

6. Ajouter l'équipe suivante:
  - NomClub: Standard
  - Localité: Liège
  - Division: 3
7. Afficher toutes les équipes encodées à l'aide d'un **SELECT**.

## Exercice n° 29 - INSERT INTO

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **use Exercice27.sql;**
5. Ajouter le joueur suivant:
  - Nom: votre nom
  - Prenom: votre prenom
  - DateNaissance: votre date de naissance. (attention au format de la date dans Mysql ☺ )
  - IdEquipe: l'IdEquipe correspondant à Standard.
6. Ajouter le joueur suivant:
  - Nom: Colin
  - Prenom: Pierre
  - DateNaissance: 22/07/1984
  - IdEquipe: l'IdEquipe correspondant à Andenne.
7. Ajouter le joueur suivant:
  - Nom: Jacques
  - Prenom: Gabriel
  - DateNaissance: 22/08/2014
  - IdEquipe: l'IdEquipe correspondant à Standard.
8. Ajouter le joueur suivant:
  - Nom: Dupont
  - Prenom: Philip
  - DateNaissance: 13/12/1991
  - IdEquipe: l'IdEquipe correspondant à Standard.
9. Affichez tous les champs des enregistrements de la table joueur.
10. Idem mais en plus affichez le nom du club (cfr **INNER JOIN**)

## Exercice n°30 - UPDATE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Mettre à jour le joueur où les champs sont les suivants:
  - Nom: Dupont
  - Prenom: Philip
6. Mettez la date de naissance à 25/12/1991 (attention au format de la date dans Mysql ☺ )
7. Idéalement il serait plus propre de faire un WHERE sur l'id du joueur et non sur le nom et prénom.

## Exercice n°31 - UPDATE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Mettre à jour le joueur dont l'idjoueur est = 3.
6. Changez-lui sa date de naissance en '1974/03/07' et son prénom deviendra 'Philipo';

## Exercice n°32 - UPDATE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Suite à l'arrêt du club d'Andenne, les joueurs de ce club feront dérénavant partie du Standard.
6. Mettez à jour la table joueur en tenant compte de ce changement de club.

## Exercice n°33 - DELETE

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Copiez les insertions suivantes:

```
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Irène', 'Link', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Andréanne', 'Kempstone', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mà', 'Keech', '1985-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Lèi', 'Blindmann', '2000-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Intéressant', 'Antcliff', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mélinda', 'Astlett', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Bénédicte', 'Kelbie', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Cinéma', 'Hugonneau', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Lén', 'Harteley', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mélodie', 'Yanin', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mélia', 'Joontjes', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Noémie', 'Cowherd', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Méryl', 'Olorenshaw', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Léana', 'Kilbourn', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mélina', 'Rodrigo', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Méryl', 'Fairley', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Lóng', 'Mehew', '1976-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Maëlys', 'Masser', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Gösta', 'Elgar', '1977-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Dorothee', 'Collie', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mailly', 'Purtell', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Gwenaëlle', 'Espinete', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mailis', 'Kennefick', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Thérèse', 'Leads', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Amélie', 'Lehrian', '2000-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Céline', 'Czajkowski', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Mélinda', 'Pedican', '1974-10-10', 1);
insert into Joueur (Prenom, Nom, DateNaissance, IdEquipe) values ('Lauréna', 'McGiffin', '1974-10-10', 1);
```

6. Supprimer les enregistrements qui respectent la condition suivante: Les personnes

nées avant le 1er janvier 1991 et nées après le 31 décembre 1979.

7. Supprimer tous les enregistrements qui respectent la condition suivante: Dont le nom de famille commence par M ou le prénom se termine par e.
8. Supprimer tous les enregistrements dont l'IdJoueur est supérieur à 3.

## Exercice n°34 - FUNCTIONS

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Créez une fonction qui s'appellera hello\_world.
6. Elle retournera la phrase 'Hello Word from SQL !'

## Exercice n°35 - FUNCTIONS avec paramètres

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Créez une fonction qui s'appellera hello.
6. Elle recevra deux paramètres de type VARCHAR(20).
7. Le premier se nommera first\_name, le second last\_name.
8. La fonction retournera par exemple 'Hello Johnny Piette !'

## Exercice n°35 - FUNCTIONS avec paramètres

1. Allez dans le répertoire d'exercices SQL
2. Connectez-vous au SGBD MySQL: **mysql -u root -p**
3. Entrez votre mot de passe.
4. Si vous n'êtes pas dans la DB Exercices, tapez: **USE Exercice27;**
5. Créez une fonction qui s'appellera remove\_tvac.
6. Elle aura un paramètre nommé prix\_tvac.
7. Elle aura un second paramètre nommé taux\_tva.



8. La fonction retournera le prix sans tva.
9. Soit le `prix_tvac = 80`

```
prix_htva = prix_tvac/(taux_tva+1)
prix_htva = 80/1.21
prix_htva = 66,1157
```

10. Votre fonction retournera donc 66,1157.

---

© 2023 [Johnny Piette](#).



Ce travail est licencié sous [Creative Commons Attribution 4.0 International License](#).

*Vous pouvez copier, modifier, distribuer et représenter ce travail, même à des fins commerciales, à condition de donner le crédit approprié, fournir un lien vers la licence, et indiquer si des modifications ont été effectuées.*