

## Jour 9 - Hello SQL

Le SQL ? .... “Oh, c'est bon, y'a pas de quoi vermifuger un abri-bus...”

### Job 00

---

Dans votre répertoire github “**runtrack2**”, créez un dossier “**jour09**”. Dans ce dossier, pour chaque étape, créez un dossier “**jobXX**” où XX est le numéro du job. Pour chacun des jobs, le code doit être présent dans un fichier exporté depuis **PhpMyAdmin** et nommé **jobXX.sql**.

Lorsque vous avez installé **WAMP (ou XAMP ou MAMP)**, vous avez installé **Apache** et **php** mais aussi un **serveur sql qui s'appelle MySQL** ou plus précisément **MariaDB**.

Pour administrer ce serveur, une application est également intégrée à WAMP : **PhpMyAdmin** (Vous pouvez y accéder également via votre navigateur ici : **<http://localhost/phpmyadmin>**, vous pouvez ajouter ce site à vos favoris).

Les codes d'accès par défaut sont :

- username : “root”,
- password : “ ” (root si vous êtes sous MAC OS)

# Job 01

---

A l'aide de phpmyadmin, créez une **base de données** nommée "jour09".  
Créez une **table** nommée "etudiants" ayant comme champs :

- **id**, int, clé primaire et Auto Incrément
- **prenom**, varchar de taille 255
- **nom**, varchar de taille 255
- **naissance**, date
- **sexe**, varchar de taille 25
- **email**, varchar de taille 255

Créez une **table** nommée "etage" ayant comme champs :

- **id**, int, clé primaire et Auto Incrément
- **nom**, varchar de taille 255
- **numero**, int
- **superficie**, int

Créez une **table** nommée "salles" ayant comme champs :

- **id**, int, clé primaire et Auto Incrément
- **nom**, varchar de taille 255
- **id\_etage**, int
- **capacite**, int

Exportez votre **base de données** en utilisant la méthode d'exportation rapide, renommez là "job 01.sql" et ajoutez la à votre répertoire de rendu.

## Job 02

---

Ajoutez maintenant des informations dans votre base de données.

Créez les **champs** et les étudiants suivants dans une table **"etudiants"**:  
**prenom, nom, naissance, sexe, email :**

- Cyril, Zimmermann, 1989-01-02, Homme, cyril@laplateforme.io
- Jessica, Soriano, 1995-09-08, Femme, jessica@laplateforme.io
- Roxan, Roumégas, 2016-09-08, Homme, roxan@laplateforme.io
- Pascal, Assens, 1999-12-31, Homme, pascal@laplateforme.io
- Terry, Cristinelli, 2005-02-01, Homme, terry@laplateforme.io
- Ruben, Habib, 1993-05-26, Homme, ruben.habib@laplateforme.io
- Toto, Dupont, 2019-11-07, Homme, toto@laplateforme.io

Créez les **champs** et les étages suivants dans une table **"etages"**:  
**id, nom, numero, superficie :**

- 1, RDC, 0, 500
- 2, R+1, 1, 500

Créez les **champs** et les salles suivantes dans une table **"salles"**:  
**id, nom, etage, capacite :**

- 1, Lounge, 1, 100
- 2, Studio Son, 1, 5
- 3, Broadcasting, 2, 50
- 4, Bocal Peda, 2, 4
- 5, Coworking, 2, 80
- 6, Studio Video, 2, 5

Exportez votre **base de données** en utilisant la méthode d'exportation rapide, renommez là **"job 02.sql"** et ajoutez la à votre répertoire de rendu.

## Job 03

---

Maintenant que vous avez créé une base de données, des tables et que vous y avez ajouté des données, vous allez pouvoir les manipuler.

Ecrivez dans le fichier **"job03.sql"** une **requête** permettant de sélectionner l'ensemble des **champs** de la table **etudiants**.

## Job 04

---

Ecrivez dans le fichier **"job04.sql"** une **requête** permettant de sélectionner le **nom** et la **capacite** de la **table "salles"**.

## Job 05

---

Ecrivez dans le fichier **"job05.sql"** une **requête** permettant de sélectionner le **prenom**, le **nom** et la **date de naissance** des **étudiants** de sexe **féminin**.

## Job 06

---

Ecrivez dans le fichier **"job06.sql"** une requête permettant de sélectionner l'ensemble des informations des **étudiants** dont **prenom** commence par un **"T"**.

## Job 07

---

Ecrivez dans le fichier **"job07.sql"** une **requête** permettant de sélectionner l'ensemble des informations des **étudiants** qui ont **plus de 18 ans**.

## Job 08

---

Ecrivez dans le fichier **"job08.sql"** une **requête** permettant de compter le **nombre d'étudiants**.

## Job 09

---

Ecrivez dans le fichier **"job09.sql"** une **requête** permettant de sélectionner l'ensemble des informations des **étudiants** qui ont **moins de 18 ans**.

## Job 10

---

Ecrivez dans le fichier **"job10.sql"** une **requête** permettant de calculer la **superficie** de l'ensemble des **étages**.

## Job 11

---

Ecrivez dans le fichier **"job11.sql"** une **requête** permettant de sélectionner la **somme** des **capacités** des **salles**.

## Job 12

---

Ecrivez dans le fichier **"job12.sql"** une **requête** permettant de sélectionner l'ensemble des **salles** en les **triant** par capacité décroissante.

## Job 13

---

Ecrivez dans le fichier **"job13.sql"** une **requête** permettant de sélectionner la **capacité moyenne des salles**.

## Job 14

---

Ecrivez dans le fichier **"job14.sql"** une **requête** permettant de sélectionner le **prenom**, le **nom** et la **date de naissance** des **étudiants** qui sont nés **entre 1998 et 2018**.

## Job 15

---

Ecrivez dans le fichier **"job15.sql"** une **requête** permettant de récupérer le **nom** des **salles** et le **nom** de leur **étage**.

## Job 16

---

Ecrivez dans le fichier **"job16.sql"** une **requête** permettant de récupérer le **nom** de l'**étage** ayant la **salle** avec la plus grande **capacité** (et afficher aussi le nom de cette salle ainsi que sa capacité).

Dans ce résultat, la colonne **"nom"** de la salle doit être renommée en **"Biggest Room"**

## Rendu

---

Le projet est à rendre sur <https://github.com/prenom-nom/runtrack2>.

Dossiers "jourXX" -> "jobXX". Fichiers nommés jobXX.sql

Pensez à donner les droits sur le répertoire à **deephoughtlaplateforme** !

## Compétences visées

---

- Concevoir des requêtes SQL
- Utiliser PhpMyAdmin

## Base de connaissances

---

- [phpMyAdmin – Wikipédia](#)  
Qu'est ce que phpMyAdmin ?
- [SQL SELECT](#)  
Cours et tutoriel SQL : la commande SELECT
- [Mettez en place une base de données avec phpMyAdmin - Concevez votre site web avec PHP et MySQL - OpenClassrooms](#)
- [MEMENTO SQL :](#)

## SQL CHEAT SHEET <http://www.sqltutorial.org>



### QUERYING DATA FROM A TABLE

**SELECT c1, c2 FROM t;**

Query data in columns c1, c2 from a table

**SELECT \* FROM t;**

Query all rows and columns from a table

**SELECT c1, c2 FROM t**

**WHERE condition;**

Query data and filter rows with a condition

**SELECT DISTINCT c1 FROM t**

**WHERE condition;**

Query distinct rows from a table

**SELECT c1, c2 FROM t**

**ORDER BY c1 ASC [DESC];**

Sort the result set in ascending or descending order

**SELECT c1, c2 FROM t**

**ORDER BY c1**

**LIMIT n OFFSET offset;**

Skip *offset* of rows and return the next *n* rows

**SELECT c1, aggregate(c2)**

**FROM t**

**GROUP BY c1;**

Group rows using an aggregate function

**SELECT c1, aggregate(c2)**

**FROM t**

**GROUP BY c1**

**HAVING condition;**

Filter groups using HAVING clause

### QUERYING FROM MULTIPLE TABLES

**SELECT c1, c2**

**FROM t1**

**INNER JOIN t2 ON condition;**

Inner join t1 and t2

**SELECT c1, c2**

**FROM t1**

**LEFT JOIN t2 ON condition;**

Left join t1 and t2

**SELECT c1, c2**

**FROM t1**

**RIGHT JOIN t2 ON condition;**

Right join t1 and t2

**SELECT c1, c2**

**FROM t1**

**FULL OUTER JOIN t2 ON condition;**

Perform full outer join

**SELECT c1, c2**

**FROM t1**

**CROSS JOIN t2;**

Produce a Cartesian product of rows in tables

**SELECT c1, c2**

**FROM t1, t2;**

Another way to perform cross join

**SELECT c1, c2**

**FROM t1 A**

**INNER JOIN t2 B ON condition;**

Join t1 to itself using INNER JOIN clause

### USING SQL OPERATORS

**SELECT c1, c2 FROM t1**

**UNION [ALL]**

**SELECT c1, c2 FROM t2;**

Combine rows from two queries

**SELECT c1, c2 FROM t1**

**INTERSECT**

**SELECT c1, c2 FROM t2;**

Return the intersection of two queries

**SELECT c1, c2 FROM t1**

**MINUS**

**SELECT c1, c2 FROM t2;**

Subtract a result set from another result set

**SELECT c1, c2 FROM t1**

**WHERE c1 [NOT] LIKE pattern;**

Query rows using pattern matching %, \_

**SELECT c1, c2 FROM t**

**WHERE c1 [NOT] IN value\_list;**

Query rows in a list

**SELECT c1, c2 FROM t**

**WHERE c1 BETWEEN low AND high;**

Query rows between two values

**SELECT c1, c2 FROM t**

**WHERE c1 IS [NOT] NULL;**

Check if values in a table is NULL or not

**Astuce : Vous pouvez tester vos requêtes dans PhpMyAdmin ici :**

