

# PHP – SQL

An SQL query walks into a bar and says : “Can I join you ?”

## Introduction

---

L’objectif de ce sujet de RunTrack est de vous remettre en jambes concernant les requêtes **SQL** avec **PHP**. Pour cela, nous allons faire quelques exercices afin de revoir comment nous pouvons **insérer, lire, mettre à jour et supprimer des données**.

Importez la base de données [lp\\_official.sql](#) en **donnant le nom “lp\_official”** à votre base de donnée. Une fois ceci fait, vous êtes prêt à faire des requêtes et devenir un professionnel des formulaires PHP et un administrateur de bases de données chevronné.

Dans le dossier **runtrack-b2-php**, créez un dossier **jour-02**. Dans ce dossier, créez un dossier **job-XX** pour chacun des jobs **où XX correspond au numéro de chaque job**.

Pensez à faire des commits réguliers et à **respecter les consignes dans le sujet**.



Pour la bonne correction de vos projets, merci de mettre vos **déclarations de fonction au début** de chacun de vos fichiers. Pour chacun des jobs, **n'affichez que ce qui vous est demandé et respectez les signatures des fonctions.**

Vos requêtes doivent être faites en utilisant **PDO uniquement.**

Bon courage !

## Job 1

---

Dans le dossier **job-01**, faites un fichier **index.php**. À l'intérieur de ce fichier, faites une fonction **find\_all\_students()**. Cette fonction devra **retourner un tableau avec toutes les lignes et toutes les colonnes de la table student sous forme de tableau associatif.**

Dans la suite de votre fichier PHP, faites une **structure HTML basique** et **générer un tableau HTML à l'aide du retour de la fonction** pour afficher tous les étudiants.

```
function find_all_students() : array {  
    /**  
     * Your code here  
     */  
}
```



## Job 2

---

Dans le dossier **job-02**, faites un fichier `index.php`. À l'intérieur de ce fichier, faites une fonction **`find_one_student()`**. Cette fonction devra **retourner un tableau avec toutes les colonnes d'une ligne de la table student en fonction d'un email**.

Dans la suite de votre page `index.php`, faites un **formulaire avec pour méthode get, un input de type text avec comme attribut name input-email-student et un bouton submit**. Cet input doit permettre de **récupérer toutes les informations de l'étudiant correspondant**.

```
function find_one_student(string $email) : array {  
    /**  
     * Your code here  
     */  
}
```

## Job 3

---

Dans le dossier **job-03**, faites un fichier `index.php`. À l'intérieur de ce fichier, faites une fonction **`insert_student()`**. Cette fonction devra **insérer un nouvel étudiant en base de donnée**.



Dans la suite de votre page `index.php`, faites un formulaire avec pour méthode `post`. Ce formulaire aura les input suivant :

- input type `email`, name = `input-email`
- input type `text`, name = `input-fullname`
- `select` avec deux option (`male`, `female`), name = `input-gender`
- input type `date`, name = `input-birthdate`
- input type `number`, name = `input-grade_id`

```
function insert_student(string $email, string $fullname, string $gender, DateTime $birthdate, int $gradeId): void {  
    /**  
     * Your code here  
     */  
}
```

## Job 4

---

Dans le dossier **job-04**, faites un fichier `index.php`. À l'intérieur de ce fichier, faites une fonction **`find_all_students_grades()`**. Cette fonction devra récupérer les emails, nom complets et nom de promotions des étudiants sous forme de tableau associatif avec la forme [`"email" ⇒ ...`, `"fullname" ⇒ ...`, `"grade_name" ⇒ ...`].

Dans la suite de votre fichier PHP, faites une **structure HTML basique** et **générez un tableau avec le retour de la fonction** pour afficher toutes les lignes récupérées.



```
function find_all_students_grades() : array {  
    /**  
     * Your code here  
     */  
}
```

## Job 5

Dans le dossier **job-05**, faites un fichier `index.php`. À l'intérieur de ce fichier, faites une fonction **`find_full_rooms()`**. Cette fonction devra retourner les noms et la capacité des salles. Ajoutez une colonne pour indiquer si une salle est pleine avec les étudiants présents dedans. La fonction devra retourner un tableau avec la structure suivante :

```
$rooms = [  
    ['name' => '...', 'capacity' => '...', 'is_full' => '...'],  
    ['name' => '...', 'capacity' => '...', 'is_full' => '...'],  
    ['name' => '...', 'capacity' => '...', 'is_full' => '...'],  
    ['name' => '...', 'capacity' => '...', 'is_full' => '...'],  
    /** ... */  
]
```

Dans la suite de votre fichier PHP, faites une structure HTML basique et générez un tableau avec le retour de la fonction pour afficher toutes les lignes récupérées.

## Job 6

Dans le dossier **job-06**, faites un fichier `index.php`. À l'intérieur de ce fichier, faites une fonction **`find_ordered_students()`**. Cette fonction devra retourner un tableau avec les noms des promotions et tous les étudiants associés avec



toutes leurs informations, les promotions triées par taille d'étudiants. Le tableau devra avoir la forme suivante :

```
$grades = [  
    "Bachelor Prépar" => [  
        ["fullname" => "...", "birthdate" => "...", "email" => "..."],  
        ["fullname" => "...", "birthdate" => "...", "email" => "..."],  
        ["fullname" => "...", "birthdate" => "...", "email" => "..."]  
    ],  
    "Bachelor 2 Logiciel" => [  
        ["fullname" => "...", "birthdate" => "...", "email" => "..."],  
        ["fullname" => "...", "birthdate" => "...", "email" => "..."],  
        ["fullname" => "...", "birthdate" => "...", "email" => "..."]  
    ]  
];
```

## Compétences visées

---

- Algorithmie
- PHP
- SQL



## Rendu

---

Le projet est à rendre sur  
<https://github.com/prenom-nom/runtrack-b2-php>.

C'est le deuxième jour de RunTrack, donc normalement, vous n'avez que le dossier jour-02 à push et le dossier jour-01 est déjà présent. Faites attention à bien **respecter la signature de chacune de vos fonctions** et les **noms de fichiers et de dossiers à la lettre !**

## Base de connaissances

---

→ [PHP](#)

→ [SQL](#)