# Remise à niveau SQL

Table des matières

[Remise à niveau SQL 1](#_Toc36196161)

[I. Intro 2](#_Toc36196162)

[1. Persistance des données 2](#_Toc36196163)

[2. CRUD : Create Read Update Delete 2](#_Toc36196164)

[3. SQL : Structured Query Language 2](#_Toc36196165)

[4. SGBDR : Système de gestion de bases de données relationnelles 2](#_Toc36196166)

[5. PHPMyAdmin 2](#_Toc36196167)

[II. Structure base de données 2](#_Toc36196168)

[III. Sélectionner des données : SELECT 2](#_Toc36196169)

[IV. ORDER BY : trier les résultats 3](#_Toc36196170)

[V. Exercice : 4](#_Toc36196171)

[VI. Requête d’insertion : INSERT INTO 5](#_Toc36196172)

[VII. Requête de mise à jour : UPDATE 5](#_Toc36196173)

[VIII. Requête de suppression : DELETE FROM 5](#_Toc36196174)

[IX. Fonctions 6](#_Toc36196175)

[1. Fonctions scalaires 6](#_Toc36196176)

[2. Fonctions d’agrégation 6](#_Toc36196177)

[X. Exercices 6](#_Toc36196178)

[XI. Sous requêtes 7](#_Toc36196179)

[XII. Les jointures 7](#_Toc36196180)

[XIII. Alias : 8](#_Toc36196181)

[1. Alias de table 8](#_Toc36196182)

[2. Alias de colonne 8](#_Toc36196183)

[XIV Création de clé étrangère OneToMany/ManyToOne 8](#_Toc36196184)

[XV. Création de clé étrangère ManyToMany 9](#_Toc36196185)

[XVI. Exercices : 9](#_Toc36196186)

[XVII. Pour aller plus loin 10](#_Toc36196187)

## Intro

### Persistance des données

### CRUD : Create Read Update Delete

### SQL : Structured Query Language

### SGBDR : Système de gestion de bases de données relationnelles

* + MySQL / MariaDB
  + SQL Server
  + Oracle
  + PostgreSQL

### PHPMyAdmin

Site web développé en PHP, offrant une interface graphique pour interagir avec une base de données MySQL/MariaDB.

## Structure base de données

* Base
  + Table1
    - Champs1 : type
    - Champs2 : type
  + Table2
    - Champs1 : type
    - Champs2 : type
* Requêtes SQL pour la structure

Créer une base de données : CREATE DATABASE nom\_de\_la\_base

* Créer une table :

CREATE TABLE (id INT NOT NULL AUTO\_INCREMENT, nom\_champ1 VARCHAR(50) NOT NULL, PRIMARY KEY (id))

CREATE TABLE `player` (

`id` int(11) NOT NULL,

`name` varchar(40) NOT NULL,

`birthdate` date DEFAULT NULL,

`email` varchar(150) NOT NULL,

`points` smallint(6) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

* Modifier une table (ajouter un champ, modifier un champ, supprimer un champ, ajouter une clé primaire/étrangère :

ALTER TABLE

* Supprimer (une base, une table) :

DROP DATABASE nom\_base et DROP TABLE nom\_table

* Requêtes pour les données

## Sélectionner des données : SELECT

* Sélectionner tous les enregistrements dans une table

SELECT \* FROM player

SELECT name, email FROM player

* Sélectionner les enregistrements correspondant à une condition

SELECT \* FROM player WHERE birthday >= ‘2000-01-01’

* Sélectionner les enregistrements correspondant à plusieurs conditions

SELECT \* FROM player WHERE birthday >= ‘2015-01-01’ AND points >= 50

SELECT \* FROM player WHERE birthday >= ‘2015-01-01’ AND points >= 50 OR name=‘toto’

SELECT \*

FROM player

WHERE birthday >= ‘2015-01-01’

AND (points >= 50 OR name=‘toto’)

* Sélectionner les enregistrements dont un champ respecte un « pattern »

%  : 0, 1 ou une infinité de caractères

\_  : un et un seul caractère

// adresse mail termine par mail.fr

SELECT \* FROM player WHERE email LIKE ‘%@mail.fr’

// nom commence par la lettre t

SELECT \* FROM player WHERE nom LIKE ‘t%’

// nom commence par la lettre t suivi d’un seul autre caractère

SELECT \* FROM player WHERE nom LIKE ‘t\_’

* Sélectionner les enregistrements dont la valeur d’un champ se trouve dans une liste prédéfinie

SELECT \* FROM player WHERE points = 45 OR points = 65 OR points = 85

SELECT \* FROM player WHERE points IN (45, 65, 85)

SELECT \* FROM player WHERE name IN (‘toto’, ‘gérard’, ‘abdel’)

* Sélectionner les enregistrements dont la valeur d’un champ se situe entre deux valeurs

SELECT \* FROM player WHERE birthday >= ‘2010-01-01’ AND birthday <= ‘2020-12-31’

SELECT \* FROM player WHERE birthday BETWEEN ‘2010-01-01’ AND ‘2020-12-31’

## ORDER BY : trier les résultats

On peut trier suivant un ou plusieurs champs (clause en fin de requête).

ASC : du plus petit au plus grand,

DESC : du plus grand au plus petit

SELECT \* FROM player ORDER BY birthday ASC

SELECT \* FROM player ORDER BY birthday ASC, name DESC

## Exercice :

Ecrire les requêtes SQL pour récupérer les joueurs dans la table player :

* les enregistrements dont l’id est supérieur ou égal à 3

SELECT \* FROM player WHERE id >= 3

* les enregistrements dont le nom est Dupond et l’adresse mail est [dupond@mail.fr](mailto:dupond@mail.fr)

SELECT \* FROM player WHERE name = 'Dupond' AND email=‘dupond@mail.fr’

* les enregistrements triés par date de naissance

SELECT \* FROM player ORDER BY birthday ASC

* les enregistrements dont la date de naissance est au moins 2005, triés par nom

SELECT \* FROM player WHERE birthday >= ‘2005-01-01’ ORDER BY name

SELECT \* FROM player WHERE YEAR(birthday) >= 2005 ORDER BY name

* les enregistrements dont l’adresse email termine soit par sfr.fr ou hotmail.fr

SELECT \* FROM player WHERE email LIKE ‘%@sfr.fr’ OR email LIKE ‘%hotmail.fr’

* les enregistrements dont le nom commence par « s » ou le nombre de points est compris entre 50 et 100

SELECT \* FROM player WHERE name LIKE ‘s%’ OR points >= 50 AND points <= 100

SELECT \* FROM player WHERE name LIKE ‘s%’ OR points BETWEEN 50 AND 100

## Regrouper des enregistrements : GROUP BY

// Sélectionner le nombre de joueurs par points

SELECT points, COUNT(\*) FROM player GROUP BY points

*// Sélectionner le nombre de points par équipe*

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points), team

FROM player

GROUP BY team

On peut cumuler plusieurs groupements :

*// Sélectionner le nombre de points par équipe par ville*

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points), team, zipcode

FROM player

GROUP BY team, zipcode

* Conditionner un GROUP BY : HAVING

*// Sélectionner le nombre de points par équipe en excluant les équipes qui moins de 60*

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points), team

FROM player

GROUP BY team

HAVING [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points)>= 60

Là où le WHERE sert à filtrer les enregistrements, le HAVING sert à filtrer les groupements.

*// Sélectionner le nombre de points par équipe en excluant les équipes qui n’ont pas au moins 60, et en ne comptant pas le nombre de points des players qui sont nés après 2019*

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points), team

FROM player

WHERE YEAR(birthday) < 2019

GROUP BY team

HAVING [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points)>= 60

## Requête d’insertion : INSERT INTO

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO player (id, name, email, birthday, points)

[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) (NULL, 'Bulbizar', 'bubul@mail.fr', NULL, '13');

Si un champ peut être null, il n’y a pas obligation à le préciser dans une requête INSERT INTO.

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO player (id, name, email, points)

[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) (NULL, 'Bulbizar', 'bubul@mail.fr', '13');

On peut ne préciser aucun, dans ce cas, il faut indiquer une valeur pour chaque champ dans VALUES, dans l’ordre des champs de la table.

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO player

[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) (NULL, 'Bulbizar', 'bubul@mail.fr', NULL, '13');

On peut réaliser plusieurs insertions dans la même requête :

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO player (id, name, email, birthday, points)

[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) (NULL, 'Bulbizar', 'bubul@mail.fr', NULL, '13'), (NULL, 'Pikachu', pika@mail.fr', NULL, '45');

## Requête de mise à jour : UPDATE

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) player [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) points = '29' WHERE id = 6;

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) player [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) points = points + 10 WHERE birthday = '2015-07-01';

La clause WHERE fonctionne exactement comme dans la requête de sélection. Ici elle sert à préciser quels sont les enregistrements à modifier.

Sans clause WHERE, la modification affectera tous les enregistrements de la table :

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) player [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) points = '29';

Pour modifier plusieurs champs en même temps, il suffit de les séparer par des virgules :

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) player [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) name = 'toto', email = 'new@mail.fr' WHERE id = 6;

## Requête de suppression : DELETE FROM

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) FROM player WHERE id = 6;

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) FROM player WHERE YEAR(birthday) = 2005;

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) FROM player;

## Fonctions

### Fonctions scalaires

Une fonction scalaire, c’est une fonction qui va appliquer un traitement sur un champ de chaque enregistrement.

* + UPPER / LOWER : majuscule / minuscule

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) UPPER(name) FROM player

* + REVERSE : inverser les caractères
  + CONCAT

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) CONCAT(name, '/', email) FROM player

* + TRUNCATE : troncature

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) TRUNCATE(price, 0) FROM product

* + ROUND : arrondi

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) ROUND(price, 1) FROM product

* + YEAR, MONTH, DAY, HOUR, MINUT, SECOND

### Fonctions d’agrégation

C’est une fonction qui s’applique sur un ensemble de valeur : elles regroupent les lignes.

* COUNT
* SUM
* AVG
* MIN / MAX
* GROUP\_CONCAT : lorsqu’un GROUP BY est présent dans la requête, la fonction permet d’obtenir toutes les valeurs qui ont été groupées, séparées par défaut par une virgule.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [SUM](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_sum)(points), team, [GROUP\_CONCAT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_group_concat)(name SEPARATOR ',') FROM player GROUP BY team