

## Abréviations

<b>BDD</b>	Base de données
<b>SGBD</b>	Système de gestion de bases de données
<b>SQL</b>	Structured Query Language ( <i>en français, langage de requête structurée</i> )

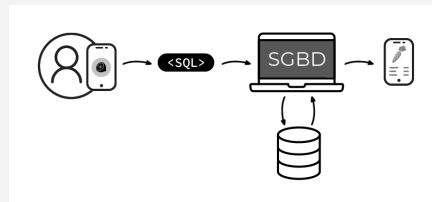
## Définitions

<b>SGBD</b>	Logiciel qui va manipuler les données d'une base. C'est ce logiciel qui commande les interactions avec la base pour y récupérer, ajouter, modifier ou supprimer des données.
<b>SQL</b>	Langage informatique qui permet d'interagir avec les bases de données.

## Les différents SGBD

<b>MySQL</b>	Le plus connu des SGBD. C'est le plus utilisé, car il <b>était</b> open-source avant son rachat par Oracle Corporation. Connu pour être notamment utilisé par les sites WordPress.
<b>MariaDB</b>	"Copie" open-source de MySQL qui suit les mêmes règles de langage que MySQL.
<b>Oracle Database</b>	Très cher, mais utile pour traiter un très gros volume de données. Ce sont presque exclusivement les grandes entreprises qui l'utilisent. Oracle tend à se faire rattraper par les SGBD open-source type MariaDB ou PostgreSQL. Il est en réelle perte de vitesse sur le marché.
<b>PostgreSQL</b>	Grand SGBD open-source disponible sur le marché. C'est le SGBD qui suit le plus les recommandations du SQL, ainsi que le plus rapide (ces dernières années). Il est notamment utilisé par Instagram ou par Spotify.
<b>SQLite</b>	Il stocke toute la base de données dans un seul et unique fichier. Peu propice à l'utilisation sur un grand nombre de données. Recommandé pour développer une base de données "en local".

## Quel est le lien entre le SGBD et le langage SQL ?



Un utilisateur arrive sur Foodly. Il scanne un aliment présent dans son supermarché pour connaître ses caractéristiques nutritionnelles. Que va faire l'application ?

**L'application va traduire cette recherche en SQL** et l'envoyer **au SGBD**, qui va **récupérer l'aliment** en question dans le stockage de la base de données, pour ensuite **redonner à l'application**. L'utilisateur retrouvera ainsi son aliment avec toutes ses caractéristiques.



By **Piotezaza** (Piotezaza)  
[cheatography.com/piotezaza/](https://cheatography.com/piotezaza/)

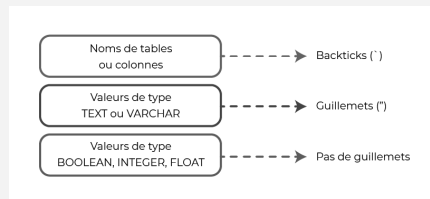
Published 24th January, 2024.  
Last updated 24th January, 2024.  
Page 1 of 6.

Sponsored by **ApolloPad.com**  
Everyone has a novel in them. Finish  
Yours!  
<https://apollopad.com>

### Commandes terminal MySQL

<code>mysql -u root -p</code>	Accéder à MySQL.
<code>mysql -u root -p nom_de_la_base_de_donnees &lt; nom_du_fichier.sql</code>	Permet de "charger" une base de données en une seule commande, en chargeant le fichier.
<code>use nom_de_la_base_de_donnees;</code> <code>source nom_du_fichier.sql;</code>	Permet de "charger" les données d'une bdd en chargeant le fichier.
<code>exit;</code>	Retourner dans le terminal en mode "normal".

### Rédaction des valeurs selon leur type



### Effectuer des opérations

<code>SELECT COUNT(*)</code> <code>FROM utilisateur</code> <code>WHERE email LIKE "%gm ail.co m";</code>	<b>COUNT (*)</b> permet de compter le nombre de lignes répondant au filtre parmi <b>toutes les colonnes</b> .
<code>SELECT COUNT(nom)</code> <code>FROM utilisateur</code> <code>WHERE email LIKE "%gm ail.co m";</code>	<b>COUNT (nom)</b> permet de compter le nombre de lignes répondant au filtre <b>dans la colonne nom</b> .
<code>SELECT COUNT( DISTINCT nom)</code> <code>FROM utilisateur</code> <code>WHERE email LIKE "%gm ail.co m";</code>	<b>COUNT ( DISTINCT nom)</b> permet d'éviter les doublons.
<code>SELECT COUNT( DISTINCT nom) AS " Alias"</code>	<b>AS</b> permet de créer un <b>alias</b> à la colonne précédemment citée.
<code>SELECT MAX(sucre) AS "taux de sucre maximum"</code> <code>FROM aliment;</code>	Nous donne le <b>maximum</b> de la colonne sur la sélection.
<code>MIN</code>	Nous donne le <b>minimum</b> de la colonne sur la sélection.
<code>AVG</code>	Nous donne la <b>moyenne</b> de la colonne sur la sélection.
<code>SUM</code>	Nous donne la <b>somme</b> de la colonne sur la sélection.
<code>SELECT ROUND( AVG (calories)) AS " calories moyennes des aliments &gt; 30g"</code> <code>FROM aliment</code> <code>WHERE calories &gt; 30;</code>	<b>ROUND( AVG (calories))</b> permet d'arrondir la valeur retournée.
<code>UPPER</code>	Transformer le texte en majuscules.
<code>NOW</code>	Retourne la date actuelle.



## Filtres SQL

<b>WHERE colonne = valeur</b>	WHERE peut s'exécuter avec SELECT, mais aussi avec n'importe quelle autre commande : vous pouvez l'utiliser avec UPDATE ou DELETE pour ne mettre à jour ou supprimer qu'un objet spécifique.
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égal à
<=	Inférieur ou égal à
<b>LIKE = " tex te"</b>	Permet de sélectionner les objets dont le texte d'une colonne répond à un modèle spécifique.
" %gm ail.co m"	Texte se <b>terminant</b> par "gmail.com".
" gma il.c om %"	Texte <b>commençant</b> par "gmail.com".
" %gm ail.co m%"	Texte <b>comprenant</b> "gmail.com" au début ou à la fin.
<b>AND</b>	Permet d'ajouter un filtre supplémentaire avec la condition <b>ET</b>
<b>OR</b>	Permet d'ajouter un filtre supplémentaire avec la condition <b>OU</b>

Le caractère % à un rôle très spécifique. Il va permettre de faire correspondre des schémas spécifiques, on parle parfois de pattern.

## Types et options de champs

<b>PRIMARY KEY</b> (option)	Champ spécial <b>obligatoire</b> dans toutes les tables. Indique à MySQL que ce champ sera l'identifiant permettant <b>d'identifier</b> les objets.
<b>NOT NULL</b> (option)	Ce champ ne peut <b>pas être nul</b> .
<b>AUTO_INCREMENT</b> (option)	Ce champ sera créé par MySQL <b>automatiquement</b> , pas besoin de s'en soucier ! MySQL va utiliser l'id précédent et y <b>ajouter +1 lors de l'ajout</b> d'un nouvel objet.
<b>UNIQUE</b> (option)	Ce champ ne peut <b>pas avoir la même valeur en double</b> .
<b>DEFAULT value</b> (option)	Ce champ sert à indiquer une valeur <b>par défaut</b> . Utile pour ne pas avoir à spécifier une valeur tout le temps.
<b>INTEGER</b> (type)	Champ numérique sous forme de nombre entier.
<b>VARCHAR (100)</b> (type)	Champ sous forme de <b>texte</b> , limité à <b>100</b> caractères
<b>FLOAT</b> (type)	Ce champ contiendra des chiffres <b>décimaux</b> .
<b>BOOLEAN</b> (type)	Ce champ ne peut stocker <b>que</b> les valeurs <b>true</b> (vrai) ou <b>false</b> (faux).

Une option dans un champ est un attribut *optionnel* qui va **modifier le comportement** de ce champ. Le type lui est obligatoire !



## Commandes SQL

<code>CREATE DATABASE nomdel abase;</code>	Création d'une nouvelle base de données.
<code>USE nomdel abase;</code>	Utiliser une base de données.
<code>SHOW DATABASES;</code>	Montre toutes les bases de données.
<code>CREATE TABLE utilis ateur ( id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY, nom VARCHAR(100), prenom VARCHAR(100), email VARCHAR(255) NOT NULL UNIQUE );</code>	Création d'une table.
<code>SHOW TABLES;</code>	Affiche les tables existantes.
<code>SHOW COLUMNS FROM nomdel atable;</code>	Affiche le schéma d'une table.
<code>FROM nomdel atable</code>	Spécifie la table de la base de donnée qui nous intéresse
<code>INSERT INTO `nomde lat able` (`nom`, `prenom`, `email` ) VALUES ( 'Doe', 'John', 'john.doe@gmail.com'), ( 'Smith', 'Jane', 'jane@hotmail.com');</code>	Permet d'insérer des données dans une table.
<code>SELECT * FROM `nomde lat able`;</code>	<code>SELECT</code> indique à MySQL que nous souhaitons récupérer de la donnée ; * indique que l'on souhaite récupérer <b>toutes</b> les colonnes (ou champs) présents dans cette table ; <code>FROM</code> permet à MySQL de comprendre depuis quelle table nous souhaitons récupérer de la donnée.
<code>SELECT `nom`, `prenom`, `email` FROM `nomde lat able`;</code>	Indiquer les colonnes à afficher permet de choisir les champs que MySQL va montrer.
<code>UPDATE `nomdelatable` SET `email` = 'iloverammstein@gmail.com' WHERE `id` = '1';</code>	<code>UPDATE table</code> mettre à jour de la donnée en indiquant la table dans laquelle se trouve(nt) le ou les objets que l'on veut modifier. <code>SET colonne = valeur</code> sert à indiquer quelles sont la ou les colonnes à modifier, et quelles sont la ou les valeurs qu'elles doivent désormais prendre.
<code>DELETE FROM `nomdelatable` WHERE `id` = '1';</code>	Supprime une ligne de table en fonction du filtre.
<code>DROP TABLE `nomde lat able`;</code>	Supprime <b>toutes</b> les données d'une table et la table elle-même.
<code>DROP DATABASE `nomde lab ase`;</code>	Supprime entièrement et de façon <b>irréversible</b> la base de données.



## Commandes SQL (cont)

```
ORDER BY DATABASE `nom` ASC;
```

Ce mot clé vous permet d'ordonner une colonne par ordre croissant (*ascending* en anglais, d'où le mot clé SQL **ASC**), ou décroissant (*descending* en anglais, soit le mot clé **DESC**).

```
CREATE VIEW utilis_ate_urs_gm_ail_vw AS
( SELECT *
  FROM utilisateur
  WHERE email LIKE "%gmail.com"
);
```

Créé une **vue**. MySQL a un système de "vues" qui permet de créer **tables temporaires** à partir d'une commande SQL. Il est donc possible de "sauvegarder" une commande SQL.

```
SELECT * FROM utilis_ate_urs_gm_ail_vw;
```

Utilisation d'une vue. À noter que **\_vw** permet de faire la distinction des "vraies" tables.

```
SELECT *
FROM utilisateur
JOIN langue
ON utilis_ate_urs_gm_ail_vw.langue_id = langue.id;
```

La table `langue` a été jointe grâce à **JOIN langue**. Pour pouvoir faire cette jointure, il faut préciser à MySQL la correspondance entre la table `langue` et la table `utilisateur`. Ici, cette correspondance est effectuée via la clé `langue_id` pour la table `utilisateur` et la table `langue`. Cela se fait grâce à `ON `utilisateur`.`langue_id` = `langue`.`id``.

```
SELECT *
FROM utilisateur
JOIN utilis_ate_urs_aliment
ON (utilisateur.id = utilisateur_aliment.utilisateur_id)
JOIN aliment
ON (aliment.id = utilis_ate_urs_aliment.aliment_id);
```

- MySQL sélectionne tous les utilisateurs avec `SELECT * FROM utilisateur`
- On joint la table `utilis_ate_urs_aliment` avec `JOIN utilis_ate_urs_aliment`
- On relie en considérant que l'id de l'utilisateur est stocké en tant que `utilisateur_id` dans la table `utilis_ate_urs_aliment` `ON (utilisateur.id = utilis_ate_urs_aliment.utilisateur_id)`
- À ce `JOIN`, on lie de la donnée de la table `aliment`, soit un `nom` avec `JOIN aliment`
- On précise que l'id de l'aliment est stocké dans `utilis_ate_urs_aliment` en tant que `aliment_id` avec `ON (aliment.id = utilis_ate_urs_aliment.aliment_id)`

```
ALTER TABLE nomdel_atable ADD colonne FLOAT;
```

Modification de la table en **ajoutant** une colonne de type `FLOAT`.

```
ALTER TABLE nomdel_atable DROP colonne;
```

Modification de la table en **supprimant** une colonne.

```
ALTER TABLE nomdel_atable MODIFY colonne;
```

Modification de la table en **changeant le type** de la colonne.

```
ALTER TABLE nomdel_atable CHANGE truc colonne;
```

Modification de la table en **changeant le nom** de la colonne.



### Commandes SQL (cont)

```
ALTER TABLE aliment
```

```
ADD FOREIGN KEY (famille_id)
```

```
REFERENCES famille (id)
```

```
ON DELETE CASCADE;
```

FOREIGN KEY (famille\_id) indique que la colonne "famille\_id" est une **clé étrangère**

REFERENCES famille (id) on indique ensuite ce à quoi cette clé fait référence

ON DELETE permet de savoir quel comportement avoir en cas de suppression de la référence

Options :

- RESTRICT ou NO ACTION empêche la suppression sous conditions

- SET NULL supprime la référence et remplace les objets par NULL

- CASCADE supprime tous les objets reliés

```
ALTER TABLE aliment
```

```
ADD FOREIGN KEY (famille_id)
```

```
REFERENCES famille (id)
```

```
ON DELETE CASCADE ON UPDATE CASCADE;
```

Options de ON UPDATE :

- RESTRICT ou NO ACTION empêche la mise à jour sous conditions

- SET NULL met à jour la référence et remplace les objets par NULL

- CASCADE met à jour tous les objets reliés



By **Piotezaza** (Piotezaza)  
[cheatography.com/piotezaza/](https://cheatography.com/piotezaza/)

Published 24th January, 2024.  
Last updated 24th January, 2024.  
Page 6 of 6.

Sponsored by **ApolloPad.com**  
Everyone has a novel in them. Finish  
Yours!  
<https://apollopad.com>