

# *Notions et requêtes SQL*

Le SQL est pensé pour manipuler des tables dans une BDD. Ses instructions se réfèrent donc à des tables.

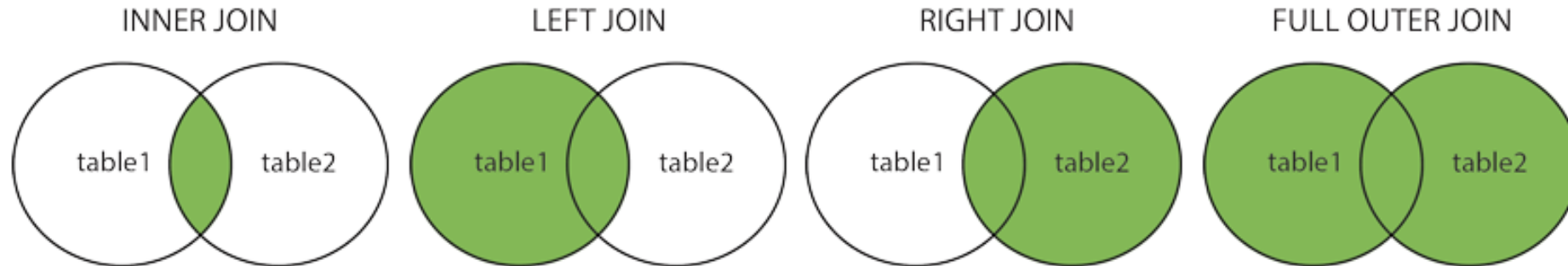
- **CREATE** : Créer une BDD ou une table
- **SELECT** : Sélectionner des données selon des critères.
- **DELETE** : Supprimer des données.
- **UPDATE** : Modifier des données
- **JOIN** : Fusionner des tables pour en faire une autre

Le SQL n'est plus le seul langage utilisé même si son ancienneté fait qu'il reste une référence.

## Association de tables (jointures)

Diagramme de Ven ou Diagramme logique

- **(INNER) JOIN** : renvoie les enregistrements dont les valeurs correspondent dans les deux tables
- **LEFT (OUTER) JOIN** : renvoie tous les enregistrements de la table de gauche et les enregistrements correspondants de la table de droite
- **RIGHT (OUTER) JOIN** : renvoie tous les enregistrements de la table de droite et les enregistrements correspondants de la table de gauche
- **FULL (OUTER) JOIN** : Renvoie tous les enregistrements lorsqu'il y a une correspondance dans la table de gauche ou de droite



## Association de tables (jointures)

```
CREATE TABLE IF NOT EXISTS `nom de la table` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB ou MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

### Explication :

Les mots clés (mot réservé pour MySQL) sont toujours écrits en majuscules par convention pour les différencier des autres mots.

```
CREATE TABLE => création d'une table  
IF NOT EXISTS => SI elle n'existe pas  
A_id => nom du champ  
Int => type de donnée  
NOT NULL => Ce champ ne peut pas être null  
AUTO_INCREMENT => incrémentation gérée par le SGBDR  
PRIMARY KEY => clé primaire  
ENGINE => moteur de stockage  
CHARSET => Encodage des caractères  
COLLATE => Spécification du type d'encodage
```

# Requêtes SQL

Pour sélectionner (afficher) des données en **SQL** on utilise le mot clé **SELECT** suivi du nom des champs que l'on veut afficher.

Ex : SELECT nom du champ

Ensuite on précise quelle table on veut sélectionner avec le mot clé **FROM**

Ex : **SELECT** a\_reference **FROM** g\_article

Cela affiche toutes les références de la table g\_article

Si l'on veut afficher par exemple que les référence ayant comme désignation « filtre à air », il faut mettre une condition on utilisant le mot clé **WHERE**.

Ex : **SELECT** a\_ reference **FROM** g\_article **WHERE** a\_désignation = « filtre a air »

a_reference
Référence du code barre
7MJCVIS
12PR3B8
WYQ2JCH
6XJ1PSZ
4QCSJLR
CQA40EG

# Requêtes SQL

On veut maintenant unir les deux tables article et voiture pour afficher le champ (colonne) type de la table g\_voiture, on va donc utiliser une **jointure**.

Dans le module3 page 22 nous avons vu qu'il existe plusieurs façons d'unir des tables.

Dans notre exemple nous voulons afficher que les enregistrements qui correspondent dans les deux tables.

Nous allons donc utiliser la jointure **INNER JOIN** suivi du nom de la table à unir.

Il faut aussi préciser comment ces tables sont unis avec le mot clé ON suivi de la primaire et secondaire (foreign key)

On utilise le mot clé SELECT pour afficher le résultat

```
SELECT a_reference , v_type
FROM g_article
INNER JOIN g_voiture
ON a_id = v_fk_article_voiture_id
WHERE a_designation = 'filtre a air'
```

a_reference Référence du code barre	v_type
7MJCVIS	500
12PR3B8	YARIS
WYQ2JCH	CLIO
6XJ1PSZ	CLIO
4QCSJLR	308
CQA40EG	500
H6MQS3D	500
EX4FM82	YARIS
4DQZN37	C3
7PDY250	CLIO
G3NL1S6	308
Z1T5ON8	C3
6J2FAG8	A180

Une condition sur un VARCHAR est à proscrire.

Pour insérer des données dans une table on utilise le mot clé **INSERT INTO** suivi :

1. du nom de la table
2. Des champs à modifier
3. Le mot clé **VALUES** suivi des valeurs à insérer

Dans l'exemple ci-dessous on insère une ligne (enregistrement).

**On met la valeur de la clé primaire à NULL car c'est le SGBD qui gère auto-incrémentation**

## INSERT INTO

```
INSERT INTO `g_article` (`a_id`, `a_reference`, `a_designation`, `a_marque`, `a_quantite`)
VALUES (NULL, '646464646', 'filtre à air', 'VALEO', '10');
```

On peut aussi ne pas mentionner l'id dans la requête le résultat sera le même

```
INSERT INTO `g_article` (`a_reference`, `a_designation`, `a_marque`, `a_quantite`)
VALUES ('646464646', 'filtre à air', 'VALEO', '10');
```

On peut voir dans le résultat que l'id a été incrémenté  
de 54 à 55

	a_id	a_reference	a_designation	a_marque	a_quantite
r	54	646464646	filtre à air	VALEO	10
r	55	646464646	filtre à air	VALEO	10

Pour mettre à jour des données dans une table on utilise le mot clé **UPDATE** suivi :

1. du nom de la table
2. Du mot clé **SET** qui désigne le ou les champs à modifier
3. Le mot clé **WHERE** (non obligatoire mais souvent utile) pour insérer une condition
4. Suivi de la condition

Si la clause **WHERE** n'est pas indiqué c'est toutes les quantités de la table qui sont modifiées

## UPDATE

```
UPDATE `g_article`  
SET `a_quantite` = '25'  
WHERE `g_article`.`a_id` = 55;
```

Dans cette requête on à modifier la quantité de 10 à 25 avec la condition que id = 55

## Résultat

a_id	a_reference	a_designation	a_marque	a_quantite
54	646464646	filtre à air	VALEO	10
55	646464646	filtre à air	VALEO	25



# Requêtes SQL

Pour un enregistrement (ligne) dans une table on utilise le mot clé **DELETE** suivi :

1. du nom de la table
2. Le mot clé **WHERE** (non obligatoire, mais très conseillé) pour insérer une condition
3. Suivi de la condition

Si la clause **WHERE** n'est pas indiqué c'est toute la table qui est vidée.

a_id	a_reference	a_désignation	a_marque	a_quantité
55	646464646	filtre à air	VALEO	25
56	646464645	Filtre à GO	MECAFILTER	11

## DELETE

```
DELETE FROM `g_article` WHERE a_id=56
```

On peut voir dans la figure du résultat que la ligne id=56 n'existe plus.

## Résultat

a_id	a_reference	a_désignation	a_marque	a_quantité
55	646464646	filtre à air	VALEO	25
57	46464gghlld	Filtre à GO	BOSCH	22