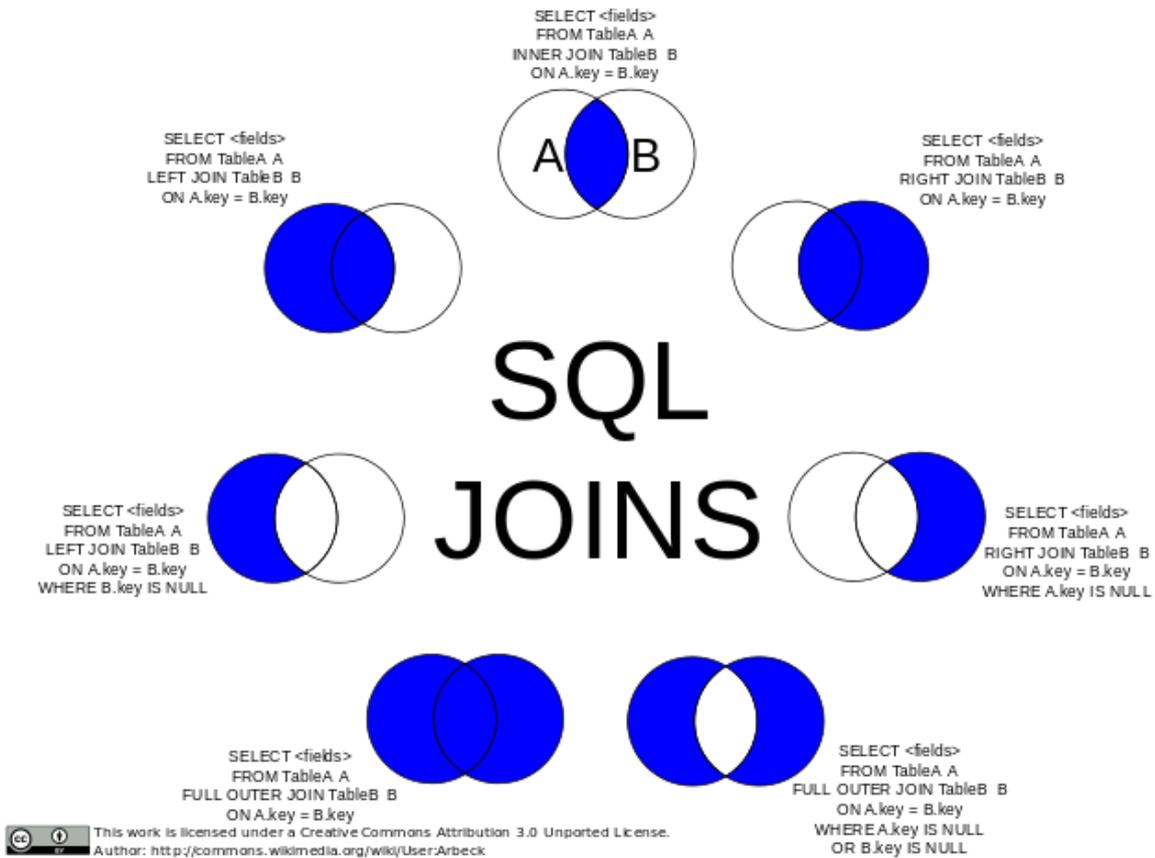


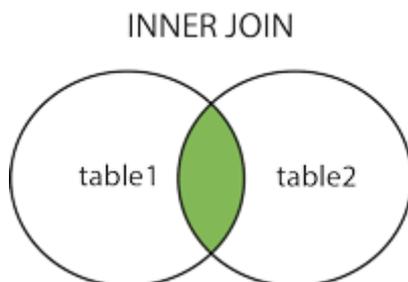
# SGDB -SQL mise en relations de plusieurs tables

Pour mettre en relation deux ou plusieurs tables on effectue une opération de jointure pour obtenir un résultat bien précis. Il existe plusieurs types de jointures pas toujours disponible dans chaque version des langages SQL



## 1. Jointure interne INNER JOIN (SQLite -MySQL )

Si on considère deux Tables T1 et T2. La jointure interne de ces deux tables sur un attribut donné « key » est une table formée en ne prenant que les enregistrements de T1 et T2 qui ont une valeur de « key » commune.



```
SELECT * ( ou attribut1,attribut2,...de table1)
FROM table1
INNER JOIN table2 ON table1.id = table2.fk_id
```

# SGDB -SQL mise en relations de plusieurs tables

EX

Table T1

Id_users	nom	prenom	Id_town
1	Valjean	Jean	1
2	Lambert	Cosette	3
3	Duprès	Louis	1

Table T2

Id_towns	metropole
1	LYON
2	PARIS

```
SELECT *  
FROM users  
INNER JOIN villes ON users.id_town = villes.id_towns
```

Il existe une seconde manière d'écrire cette relation de correspondance :

```
SELECT *  
FROM users  
JOIN villes  
WHERE users.id_town = villes.id_towns
```

Résultats :

<u>id</u>	<u>users</u>	<u>nom</u>	<u>prenom</u>	<u>id</u>	<u>town</u>	<u>id</u>	<u>towns</u>	<u>metropole</u>
5		Valjean	Jean	1		1		LYON
7		Duprès	Louis	1		1		LYON

Si on modifie

```
SELECT nom  
FROM utilisateurs  
INNER JOIN villes ON utilisateurs.id_town = villes.id_towns
```

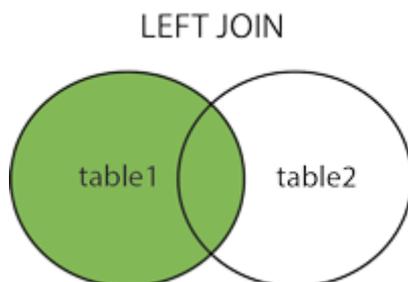
nom  
Valjean  
Duprès

## 2. Jointure GAUCHE LEFT JOIN (SQLite -MySQL )

La jointure gauche externe des tables T1 et T2 est une table réunissant :

Les enregistrements de T1 et T2 qui possèdent un attribut commun « key » idem jointure interne

Les enregistrements de T1 (table T1 à gauche) pour lesquels l'attribut « key » de T1 ne correspond avec aucun attribut « key » de T2



```
SELECT * ( attribut1,attribut2, ...de table1)  
FROM table1  
LEFT JOIN table2 ON table1.id = table2.fk_id
```

# SGDB -SQL mise en relations de plusieurs tables

EX :

```
SELECT *  
FROM users  
LEFT JOIN villes ON users.id_town = villes.id_towns
```

Résultat ( DBRowner ) :

	id_users	nom	prenom	id_town	id_towns	metropole
1	1	Valjean	Jean	1	1	PARIS
2	2	Lambert	Cosette	3	NULL	NULL
3	3	Duprès	Louis	1	1	PARIS

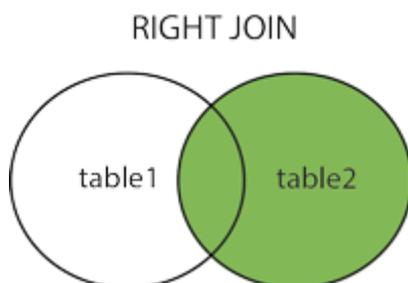
On récupère donc tous les éléments de la table de gauche plus ceux de la table de droite avec des « NULL » qui correspondent à l'absence de ville pour l'identifiant ville 3.

## 3. Jointure DROITE RIGHT JOIN (SQLite -MySQL )

La jointure droite externe des tables T1 et T2 est une table réunissant :

Les enregistrements de T1 et T2 qui possèdent un attribut commun « key » idem jointure interne

Les enregistrements de T2 (table T2 à droite) pour lesquels l'attribut « key » de T2 ne correspond avec aucun attribut « key » de T1



```
SELECT *  
FROM table1  
RIGHT JOIN table2 ON table1.id = table2.fk_id
```

EX :

```
SELECT *  
FROM utilisateurs  
RIGHT JOIN villes ON users.id_town = villes.id_towns
```

Résultat ( PhPMysql ) :

<u>id_users</u>	<u>nom</u>	<u>prenom</u>	<u>id_town</u>	<u>id_towns</u>	<u>metropole</u>
5	Valjean	Jean	1	1	LYON
7	Duprès	Louis	1	1	LYON
NULL	NULL	NULL	NULL	2	PARIS

Les NULL sont les ligne pour lesquels aucun élément de la table à gauche correspond avec l'attribut commun de la table de droite.

# SGDB -SQL mise en relations de plusieurs tables

## 4. Alias

Dans le langage SQL il est possible d'utiliser des **alias** pour renommer temporairement une colonne ou une table dans une requête. Cette astuce est particulièrement utile pour faciliter la lecture des requêtes en particulier quand les noms des colonnes/attributs sont à ralonge.

Alias sur une colonne

<pre>SELECT colonne1 AS c1, colonne2 FROM `table`</pre>	ou	<pre>SELECT colonne1 c1, colonne2 FROM `table`</pre>
---	----	--

Alias sur une table

<pre>SELECT * FROM `nom_table` AS t1</pre>	ou	<pre>SELECT * FROM `table` t1</pre>
--	----	-------------------------------------

Ex sur notre BDD même si ici l'intérêt est limité.

```
1 SELECT metropole AS ville
2 FROM villes AS v
3 INNER JOIN users AS u ON u.id_town = v.id_towns
4 WHERE u.nom= "Duprès"
```

Réponse

	ville
1	PARIS