BASE DE DONNÉES – JOINTURES & AGRÉGATIONS

Jointures & Agrégations

Le concept de la jointure est de sélectionner des données se trouvant dans plusieurs tables distinctes au sein d'une base de données.

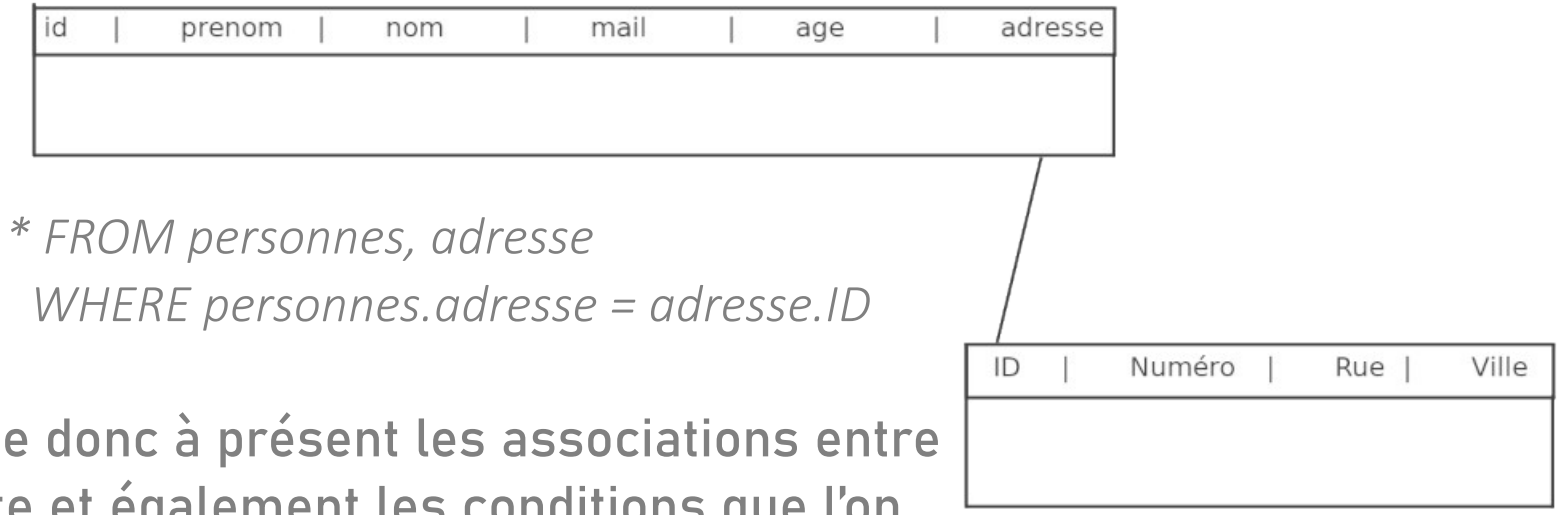
Ce qui va nous permettre de préciser les données sur lesquelles nous allons travailler aussi bien au niveau :

- d'un SELECT (lecture)
- d'un UPDATE (mise à jour)
- d'un DELETE (suppression)

Ces jointures ont lieu entre deux tables. On exprime la correspondance entre celles-ci par un critère d'égalité entre deux clés.

Jointures & Agrégations

Il va être possible de lire l'adresse correspondant à une personne de la manière suivante : *SELECT * FROM personnes, adresse*
WHERE personnes.adresse = adresse.ID



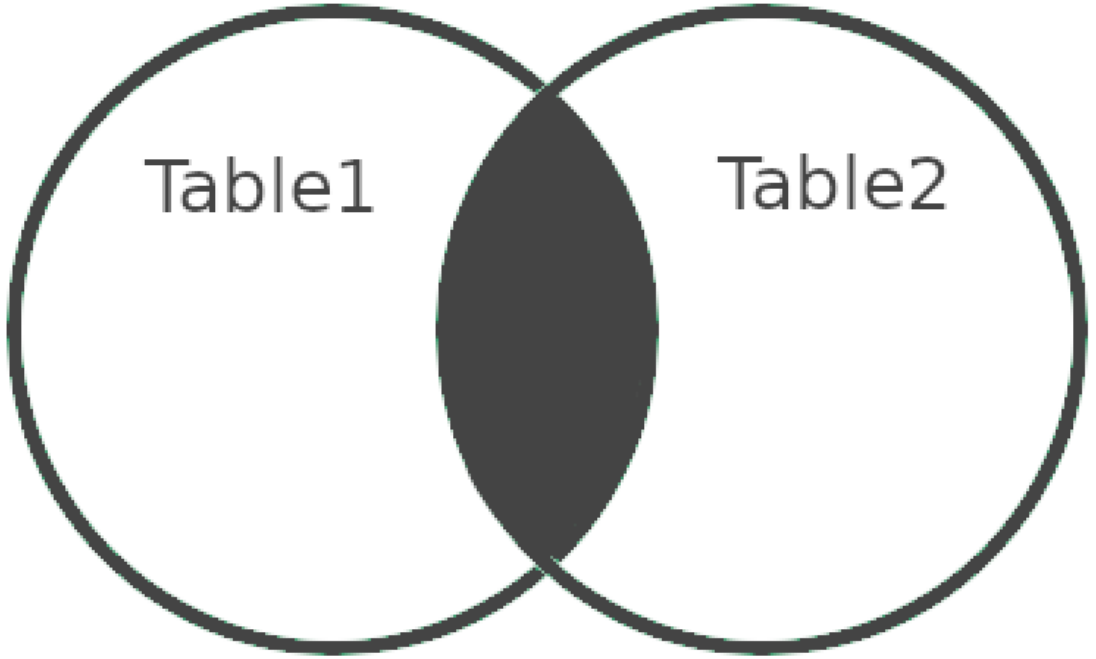
La clause WHERE mélange donc à présent les associations entre les tables que l'ont exploite et également les conditions que l'on va venir appliquer à notre sélection de données (l'ordre des conditions n'a pas d'importance).

*SELECT * FROM personnes, adresse WHERE personnes.adresse = adresse.ID*
AND personnes.nom = 'toto' ;

Jointures & Agrégations

Une jointure basique comme celle-ci consiste en la liaison de plusieurs tables entre elles.

Les enregistrements sont retournés lorsqu'il y a au moins une ligne dans chaque colonne qui correspond à la condition.



On pourra l'instancier tel que :

*SELECT * FROM table1 INNER JOIN table2*
ON table1.adresse = table2.id ;

(Il est possible de remplacer le ON par un WHERE)

Jointures & Agrégations

Exemple : Sur la base de la requête précédente, il est ainsi possible de créer un tableau joint contenant des informations provenant des deux tables incluent dans le requête.

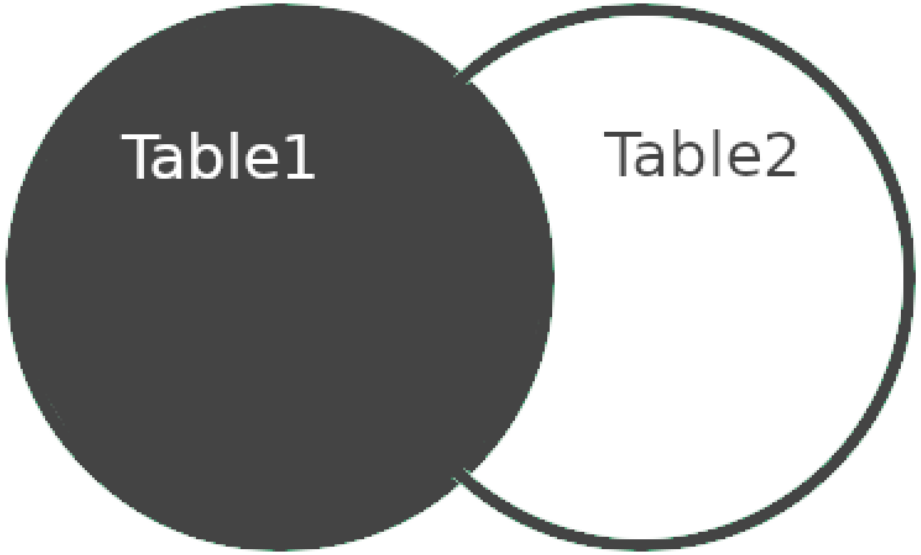
id	prenom	nom	age	numero	rue	ville
1	alex	robine	23	14	avenue du general de gaulle	paris
2	toto	toto	35	14	avenue du general de gaulle	paris
3	jean	dive	28	131	rue jean jaures	melun
4	paul	jean	29	99	boulevard jules ferry	etampes
5	robert	toto	40	145	place leon gambetta	paris
6	jean	marche	36	14	rue du pain	juvisy sur orge
7	jean	phil	18	19	rue louis pasteur	paris
8	leroy	jenkins	39	14	rue du pain	juvisy sur orge

Jointures & Agrégations

Si une adresse a été supprimée, il ne sera plus possible de visualiser les utilisateurs qui y sont rattachés dans la mesure ou notre requête ne retournera que les résultats ou la condition est vrai dans les deux tables.

Il est donc possible de procéder à une requête de « jointure gauche » qui nous permettra d'opérer notre jointure tout en conservant le contenu de la table de gauche même si il n'y a pas de correspondance avec la table de droite.

La commande LEFT JOIN (ou LEFT OUTER JOIN) permet d'arriver à ce résultat.



Jointures & Agrégations

Ainsi, on récupérera des valeurs NULL aux champs sans jointure possible.

```
MariaDB [joining]> SELECT eleves.id,prenom,nom,age,numero,rue,ville FROM eleves LEFT JOIN adresses ON eleves.adresse = adresse
s.id;
+----+----------+-----------+------+--------+-----------------------------+-----------------+
| id | prenom   | nom       | age  | numero | rue                         | ville           |
+----+----------+-----------+------+--------+-----------------------------+-----------------+
|  1 | alex     | robine    |   23 |     14 | avenue du general de gaulle | paris           |
|  2 | toto     | toto      |   35 |     14 | avenue du general de gaulle | paris           |
|  3 | jean     | dive      |   28 |    131 | rue jean jaures             | melun           |
|  4 | paul     | jean      |   29 |     99 | boulevard jules ferry       | etampes         |
|  5 | robert   | toto      |   40 |    145 | place leon gambetta         | paris           |
|  6 | jean     | marche    |   36 |     14 | rue du pain                 | juvisy sur orge |
|  7 | jean     | phil      |   18 |     19 | rue louis pasteur           | paris           |
|  8 | leroy    | jenkins   |   39 |     14 | rue du pain                 | juvisy sur orge |
|  9 | lee      | roy       |   21 |     19 | rue georges clemenceau      | troyes          |
| 10 | jean-luc | vlan      |   26 |     14 | rue du pain                 | juvisy sur orge |
| 11 | peter    | parker    |   37 |     14 | rue du pain                 | juvisy sur orge |
| 12 | bruce    | wayne     |   53 |     99 | boulevard jules ferry       | etampes         |
| 13 | jean     | parker    |   38 |     19 | rue louis pasteur           | paris           |
| 14 | martin   | martin    |   14 |     14 | avenue du general de gaulle | paris           |
| 15 | jean     | paul      |   59 |     14 | rue du pain                 | juvisy sur orge |
| 16 | marie    | gole      |   54 |     14 | avenue du general de gaulle | paris           |
| 17 | Sandrine | DE FRANCE |   29 |   NULL | NULL                        | NULL            |
+----+----------+-----------+------+--------+-----------------------------+-----------------+
17 rows in set (0.000 sec)
```

Jointures & Agrégations

Cette requête est particulièrement intéressante pour récupérer les informations de notre table eleves tout en récupérant les données associées même si il n'existe pas de correspondance avec notre table adresses.

Dans notre cas, cela nous permettra par exemple de lister les élèves n'ayant pas correctement renseigné leur dossier. En effet, il nous sera possible de filtrer par la suite sur la valeur NULL pour ressortir uniquement ceux qui n'ont pas de correspondance.

NULL n'étant pas une chaîne de caractère, il faudra utiliser l'opérateur IS NULL.

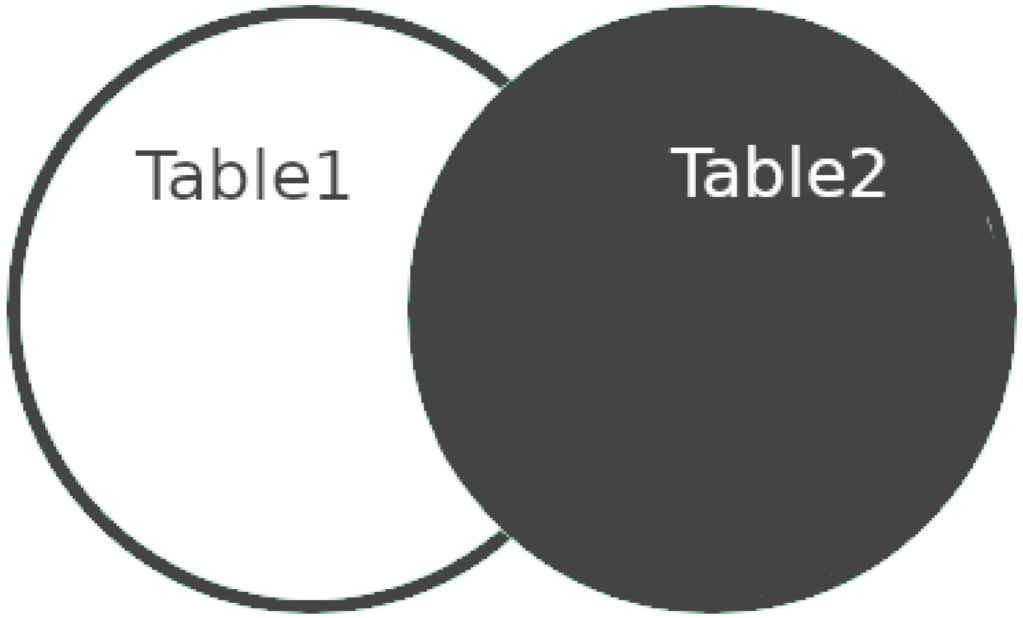
```
se = adresses.id WHERE adresses.id IS NULL;
+----+----------+-----------+------+------+--------+------+-------+
| id | prenom   | nom       | age  | id   | numero | rue  | ville |
+----+----------+-----------+------+------+--------+------+-------+
| 17 | Sandrine | DE FRANCE |   29 | NULL |   NULL | NULL | NULL  |
+----+----------+-----------+------+------+--------+------+-------+
1 row in set (0.000 sec)
```

Jointures & Agrégations

Étant donné qu'il est possible de ressortir l'ensemble des éléments avec une jointure par la gauche, nous sommes évidement en mesure d'opérer le même schéma avec une jointure par la droite.

Et de cette manière, cibler les adresses n'étant associées à aucune personne dans la base.

```
sse = adresses.id WHERE eleves.id IS NULL;
+------+--------+------+------+----+--------+--------------+------------+
| id   | prenom | nom  | age  | id | numero | rue          | ville      |
+------+--------+------+------+----+--------+--------------+------------+
| NULL | NULL   | NULL | NULL |  8 |     44 | rue du quart | antequatre |
+------+--------+------+------+----+--------+--------------+------------+
1 row in set (0.000 sec)
```

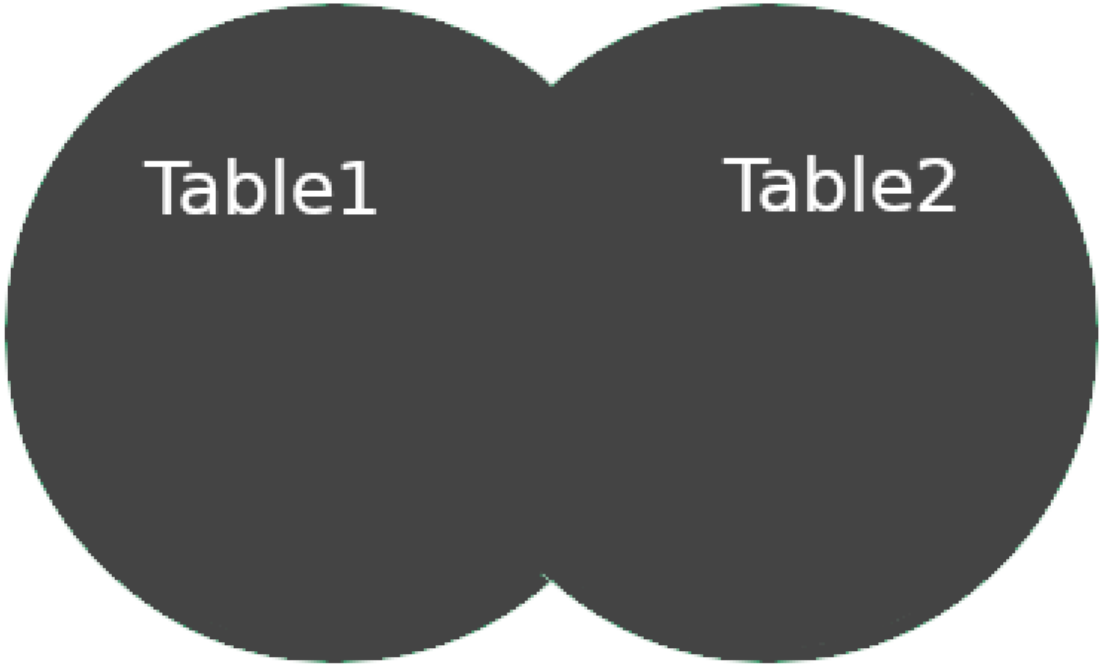


Jointures & Agrégations

En dernière instance, il est possible d'opérer une jointure complète des deux tables avec un FULL JOIN (FULL OUTER JOIN).

L'utilisation de cette commande permet de combiner les résultats des deux tables, les associer ensemble grâce à une condition et remplir avec des valeurs NULL si la condition n'est pas respectée.

*SELECT * FROM Table1 FULL JOIN Table2 ON*
Table1.id = Table2.id ;



Jointures & Agrégations

Au-delà des jointures, il va être possible d'agréger les données par les valeurs de leurs attributs. La commande GROUP BY est utilisée pour grouper plusieurs résultats et utiliser une fonction de totaux sur un ensemble de résultats.

Sur notre base de données d'exemple, il sera possible de grouper les élèves par adresse et obtenir le nombre de personnes habitant à chaque adresse.

SELECT numero,rue,ville,COUNT(RUE) AS 'Living Here' FROM eleves INNER JOIN adresses ON eleves.adresse = adresses.id GROUP BY numero,rue,ville;

numero	rue	ville	Living Here
14	avenue du general de gaulle	paris	4
99	boulevard jules ferry	etampes	2
145	place leon gambetta	paris	1
14	rue du pain	juvisy sur orge	5
19	rue georges clemenceau	troyes	1
131	rue jean jaures	melun	1
19	rue louis pasteur	paris	2

Jointures & Agrégations

Il existe un certain nombre d'autres fonctions d'agrégations permettant notamment la sortie de statistiques liées aux données présentes en base.

AVG => permet de calculer une valeur moyenne sur un ensemble d'enregistrements numériques non nuls.

SELECT AVG(age) FROM eleves ;

MAX => permet de sortir la valeur numérique la plus haute d'une colonne

SELECT MAX(age) FROM eleves ;

MIN => permet de sortir la valeur numérique la plus basse d'une colonne

SELECT MIN(age) FROM eleves ;

Jointures & Agrégations

COUNT => Renvoie le nombre de tuples

SUM => Calcule la somme numérique

STDDEV => Calcule l'écart type

VARIANCE => Calcule la variance des valeurs

On n'utilisera pas d'attribut SELECT en même temps qu'une fonction d'agrégation quelconque (sauf si on utilise GROUP BY simultanément).

Jointures & Agrégations

HAVING => L'instruction HAVING est très similaire à l'instruction WHERE à une différence près. Celle-ci permet de créer un filtre sur la base de fonctions d'agrégation (SUM, COUNT, AVG, …).

SELECT numero,rue,ville FROM eleves INNER JOIN adresses ON eleves.adresse = adresses.id_adresse GROUP BY numero,rue HAVING COUNT(rue) > 2;

numero	rue	ville
14	avenue du general de gaulle	paris
14	rue du pain	juvisy sur orge

Jointures & Agrégations

Il est également possible d'exécuter des requêtes en les imbriquant et ainsi exécuter une requête à l'intérieur d'une autre requête par exemple au sein d'un WHERE ou d'un HAVING.

SELECT CONCAT(nom, " ", prenom) FROM eleves WHERE adresse IN (SELECT id_adresse FROM adresses WHERE numero = 99);

```
+--------------------------+
| CONCAT(nom, " ", prenom) |
+--------------------------+
| jean paul                |
| wayne bruce              |
+--------------------------+
2 rows in set (0.001 sec)
```

TP – SQL QUERIES

FIN !

Merci pour votre participation !