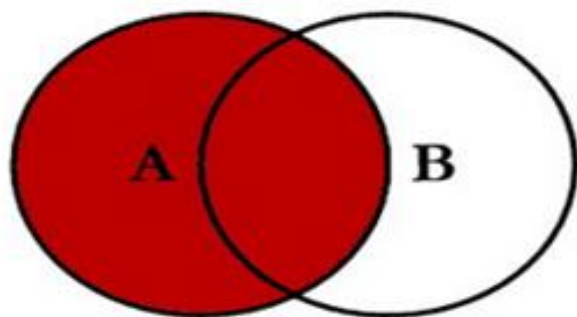
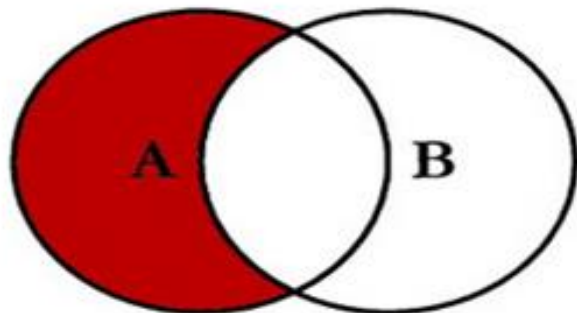


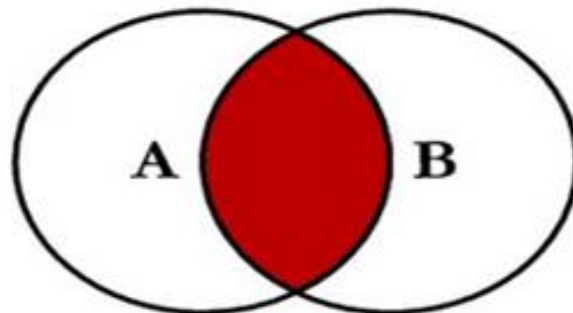
SQL JOINS



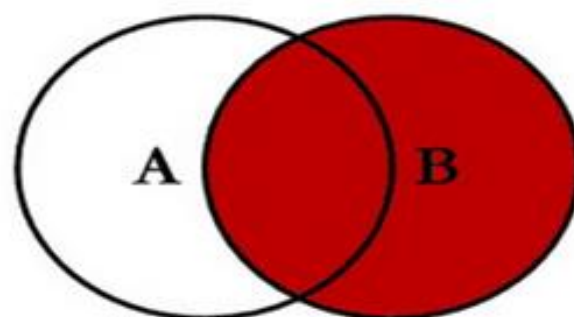
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



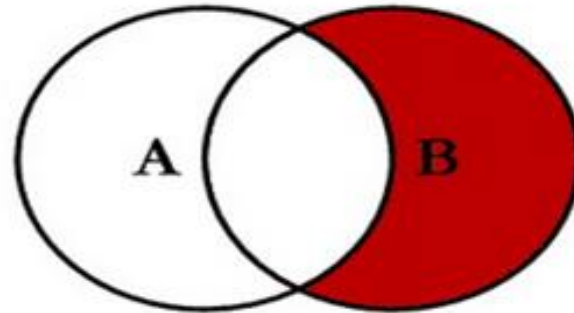
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



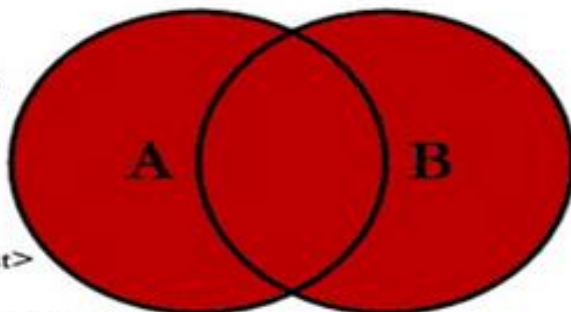
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



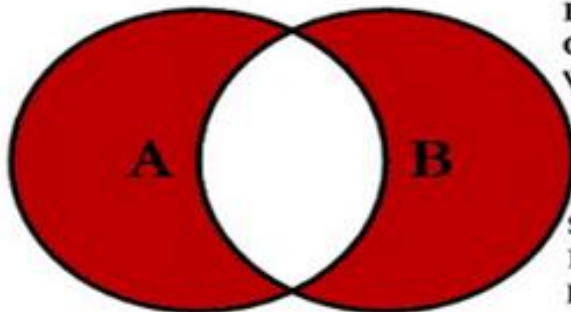
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

Différences entre INNER, LEFT, RIGHT et OUTER JOIN en SQL

Nous prendrons le cas exemple d'une table **A** , sélectionnée par un FROM à laquelle est joint la table **B**. Nous parlerons de correspondances dans le cas où une entrée de **A** possède une entrée associée dans **B**.

Pour comprendre le fonctionnement algorithmique d'une jointure, vous devez avoir en tête que **A** et **B** sont jointes par la valeur partagée d'un de leurs champs respectifs, c'est à dire un champs de **A** à la même domaine de valeur qu'un champs de **B**, permettant d'établir un lien, une correspondance entre les deux.

Mais il se peut qu'il existe des entrées de A qui ne soient liées à aucune entrée de B et qu'il existe des entrées de B qui ne soient pas liées à aucune entrée de A, ainsi certaines entrées peuvent théoriquement n'avoir aucune correspondance.

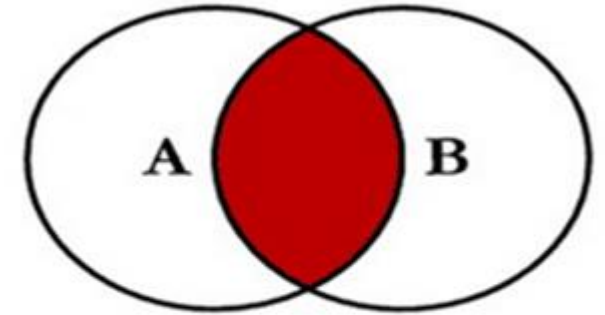
Si on tente de faire une jointure sur l'autre table alors qu'elle ne possède aucune entrée correspondante, toutes les valeurs vaudront NULL. Et si elle en possède plusieurs, il y a aura autant de résultat que de données correspondantes, même si les données de la première table doivent être dupliquées.

INNER JOIN

Le mot INNER correspond à l'intersection entre les deux tables, cette jointure rend donc la correspondance nécessaire pour obtenir des résultats.

Cette jointure permet de récupérer tous les résultats depuis la table A uniquement si elle possède une correspondance dans B.

```
SELECT * FROM A INNER JOIN B ON A.id = B.main_id
```



```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```

LEFT JOIN

- LEFT JOIN veut bien dire ici une jointure par la gauche, la table de gauche étant la table principale, cette jointure se base donc sur la table A pour établir sa jointure.

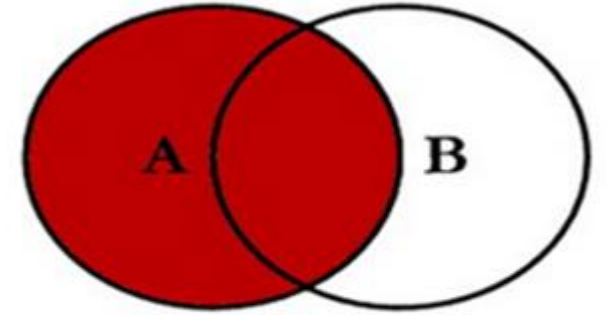
On sélectionne donc toutes les entrées de A et on les joint à la table B qu'il y ait une correspondance ou non. C'est implicitement un LEFT OUTER JOIN.

SELECT * FROM A LEFT JOIN B ON A.id = B.main_id

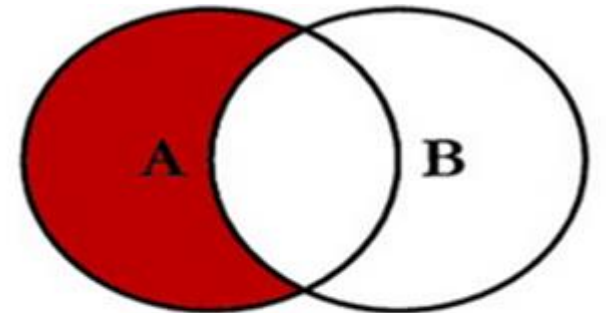
Ou

SELECT * FROM A LEFT JOIN B ON A.id = B.main_id

Where B.Key is NULL



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```

RIGHT JOIN

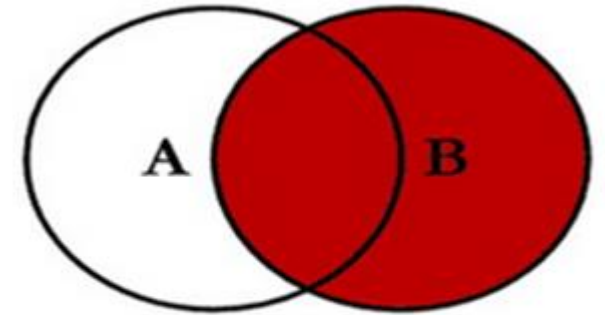
- Vous pouvez maintenant déduire ce qu'est une RIGHT JOIN, la table de droite étant la table jointe. On sélectionne donc toutes les entrées de A et on les joint à la table B qu'il y ait une correspondance ou non. C'est implicitement un RIGHT OUTER JOIN..

SELECT * FROM A RIGHT JOIN B ON A.id = B.main_id

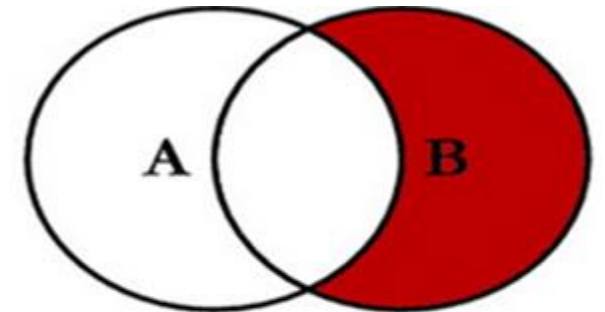
Ou

SELECT * FROM A RIGHT JOIN B ON A.id = B.main_id

Where A.Key is NULL



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```

FULL OUTER JOIN

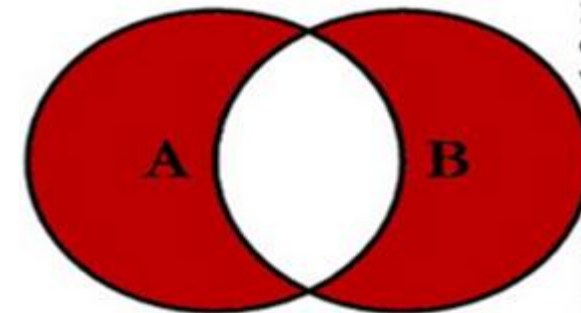
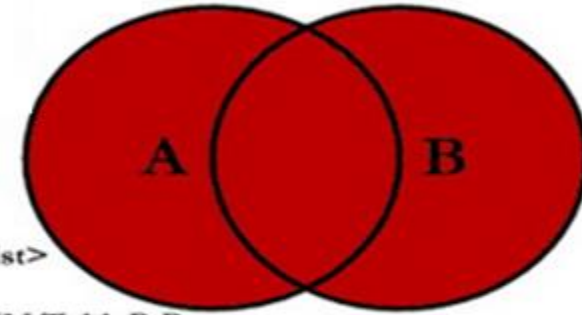
- Vous pouvez maintenant déduire ce qu'est une RIGHT JOIN, la table de droite étant la table jointe. On sélectionne donc toutes les entrées de A et on les joint à la table B qu'il y ait une correspondance ou non. C'est implicitement un RIGHT OUTER JOIN..

SELECT * FROM A FULL OUTER JOIN B ON A.id = B.main_id

OU

SELECT * FROM A FULL OUTER JOIN B ON A.id = B.main_id WHERE A.key is Null or B.Key is null

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

CAS D'USAGE

- **En pratique, on utilise généralement les LEFT Join et les INNER Join mais il est fréquent de les voir mal utilisés, parfois inversés, il est important de savoir au préalable si on souhaite récupérer toutes les entrées de table A ou seulement celles ayant une correspondance avec la table B**