

Jointure en SQL

M1 Statistique

E. Claeys

ICUBE/IRMA
Université de Strasbourg

Base de données, 2018

Programme du cours

1 Les jointures

2 Types de jointures

3 Conclusion

Les jointures en SQL permettent d'associer plusieurs tables dans une même requête. Cela permet d'exploiter la puissance des bases de données relationnelles pour obtenir des résultats qui combinent les données de plusieurs tables de manière efficace.

En général, les jointures consistent à associer des lignes de 2 tables en associant l'égalité des valeurs d'une colonne d'une première table par rapport à la valeur d'une colonne d'une seconde table. Imaginons qu'une base de 2 données possède une table "utilisateur" et une autre table "adresse" qui contient les adresses de ces utilisateurs. Avec une jointure, il est possible d'obtenir les données de l'utilisateur et de son adresse en une seule requête.

On peut aussi imaginer qu'un site web possède une table pour les articles (titre, contenu, date de publication ...) et une autre pour les rédacteurs (nom, date d'inscription, date de naissance ...). Avec une jointure il est possible d'effectuer une seule recherche pour afficher un article et le nom du rédacteur. Cela évite d'avoir à afficher le nom du rédacteur dans la table "article".

Il y a plusieurs méthodes pour associer 2 tables ensemble. Voici quelques techniques parmi les utilisées :

- **INNER JOIN** : jointure interne pour retourner les enregistrements quand la condition est vrai dans les 2 tables. C'est l'une des jointures les plus communes.
- **LEFT JOIN** (ou **LEFT OUTER JOIN**) : jointure externe pour retourner tous les enregistrements de la table de gauche (**LEFT** = gauche) même si la condition n'est pas vérifié dans l'autre table.

- **RIGHT JOIN** (ou **RIGHT OUTER JOIN**) : jointure externe pour retourner tous les enregistrements de la table de droite (**RIGHT** = droite) même si la condition n'est pas vérifiée dans l'autre table.
- **FULL JOIN** (ou **FULL OUTER JOIN**) : jointure externe pour retourner les résultats quand la condition est vraie dans au moins une des 2 tables.
- **UNION JOIN** : jointure d'union
- **NATURAL JOIN** : jointure naturelle entre 2 tables s'il y a au moins une colonne qui porte le même nom entre les 2 tables SQL

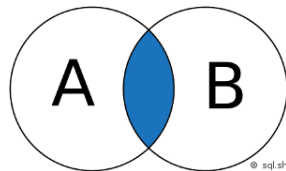
Exemple : INNER JOIN

```
SELECT *  
FROM A  
INNER JOIN B ON A.key = B.key
```



Exemple : INNER JOIN

```
SELECT *  
FROM A  
INNER JOIN B ON A.key = B.key
```



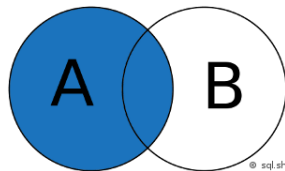
Exemple : LEFT JOIN

```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```



Exemple : LEFT JOIN

```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```



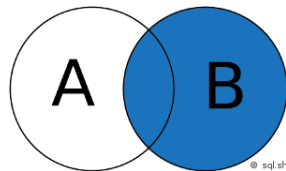
Exemple : RIGHT JOIN

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```



Exemple : RIGHT JOIN

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```



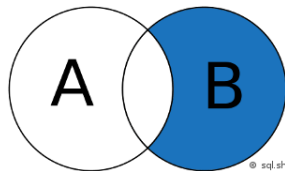
Exemple : RIGHT JOIN

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE A.key IS NULL
```



Exemple : RIGHT JOIN

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE A.key IS NULL
```



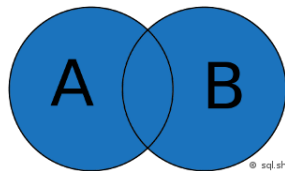
Exemple : FULL JOIN

```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key
```

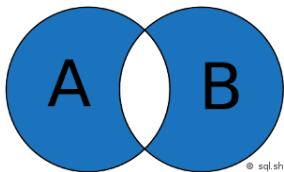


Exemple : FULL JOIN

```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key
```

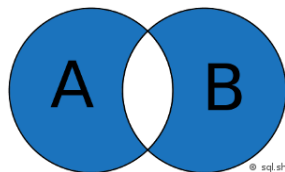


Exemple : FULL JOIN



Exemple : FULL JOIN

```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL
```



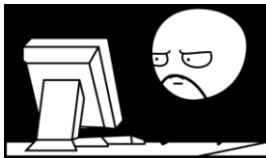
Pour conclure !

- Une jointure SQL permet de lier deux ou plusieurs tables entre elles. La première table écrite dans la requête SQL est toujours considérée comme la table de gauche, les autres sont considérées comme les tables de droites.
- `INNER JOIN` est une jointure interne, elle retourne les entrées qui ont une correspondance entre les deux tables en fonction de la valeur choisie après `ON`. Les autres jointures sont des jointures externes.

Pour conclure !

- LEFT JOIN et RIGHT JOIN vont retourner les entrées qui ont une correspondance entre les deux tables, mais également toute la table qui est désignée (gauche ou droite). Les entrées qui n'ont pas trouvé de correspondance auront certains de leurs champs mis à NULL.
- FULL JOIN retourne tout ! S'il n'y a pas de correspondance entre les deux tables pour une ligne, certains de ses champs seront mis à NULL. Il est possible d'écarter les correspondances du résultat en ajoutant une condition dans la requête.

TD n° 8 : Les jointures en MySQL (1h30/2h)



Pensez à sauvegarder vos commandes !