

JOINTURES

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.

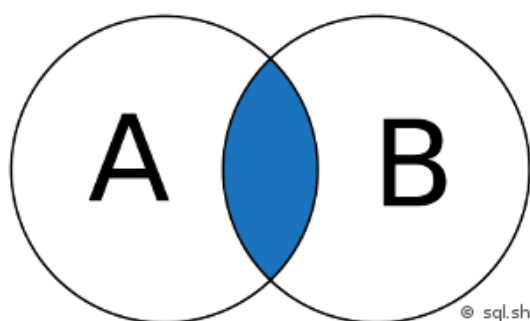
Types de jointures

Il y a plusieurs méthodes pour associer 2 tables ensemble. Voici la liste des différentes techniques qui sont utilisées :

- **INNER JOIN** : jointure interne pour retourner les enregistrements quand la condition est vraie 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 vérifiée 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.

SQL INNER JOIN

Dans le langage SQL la commande INNER JOIN, aussi appelée EQUIJOIN, est un type de jointures très communes pour lier plusieurs tables entre-elles. Cette commande retourne les enregistrements lorsqu'il y a au moins une ligne dans chaque colonne qui correspond à la condition.



Pour utiliser ce type de jointure il convient d'utiliser une requête SQL avec cette syntaxe :

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

La syntaxe ci-dessus stipule qu'il faut sélectionner les enregistrements des tables **table1** et **table2** lorsque les données de la colonne "id" de table1 est égal aux données de la colonne fk_id de table2.

Exemple

Imaginons une application qui possède une table **utilisateur** ainsi qu'une table **commande** qui contient toutes les commandes effectuées par les utilisateurs.

Table utilisateur :

Id	NOM	PRENOM	MAIL	VILLE
1	Aimée	Marechal	aime.marechal@example.com	Paris
2	Esmée	Lefort	esmee.lefort@example.com	Lyon
3	Marine	Prevost	m.prevost@example.com	Lille
4	Luc	Rolland	lucrolland@example.com	Marseille

Table commande :

Utilisateur_id	Date_achat	Num_facture	Prix_total
1	2013-01-23	A00103	203.14
1	2013-02-14	A00104	124.00
2	2013-02-17	A00105	149.45
2	2013-02-21	A00106	235.35
5	2013-03-02	A00107	47.58

Pour afficher toutes les commandes associées aux utilisateurs, il est possible d'utiliser la requête suivante :

```
SELECT id, prenom, nom, date_achat, num_facture, prix_total  
FROM utilisateur  
INNER JOIN commande ON utilisateur.id = commande.utilisateur_id
```

Résultats :

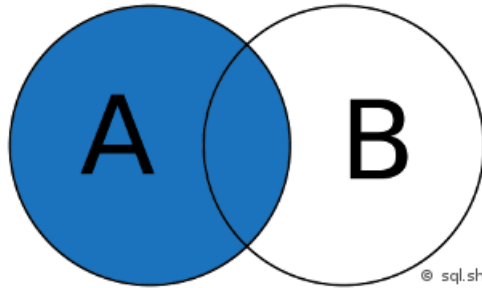
Id	prenom	nom	Date_achat	Num_facture	Prix_total
1	Aimée	Marechal	2013-01-23	A00103	203.14
1	Aimée	Marechal	2013-02-14	A00104	124.00
2	Esmée	Lefort	2013-02-17	A00105	149.45
2	Esmée	Lefort	2013-02-21	A00106	235.35

Le résultat de la requête montre parfaite la jointure entre les 2 tables. Les utilisateurs 3 et 4 ne sont pas affichés puisqu'il n'y a pas de commandes associés à ces utilisateurs.

Attention : il est important de noter que si un utilisateur à été supprimé, alors on ne verra pas ses commandes dans la liste puisque INNER JOIN retourne uniquement les résultats ou la condition est vrai dans les 2 tables.

SQL LEFT JOIN

Dans le langage SQL, la commande LEFT JOIN (aussi appelée LEFT OUTER JOIN) est un type de jointure entre 2 tables. Cela permet de lister tous les résultats de la table de gauche (left = gauche) même s'il n'y a pas de correspondance dans la deuxième tables.



Syntaxe

Pour lister les enregistrements de table1, même s'il n'y a pas de correspondance avec table2, il convient d'effectuer une requête SQL utilisant la syntaxe suivante.

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

Exemple

Dans l'exemple précédent, on peut alors lister tous les utilisateurs avec leurs commandes et afficher également les utilisateurs qui n'ont pas effectués d'achats. On utilise la requête suivante :

```
SELECT *  
FROM utilisateur  
LEFT JOIN commande ON utilisateur.id = commande.utilisateur_id
```

Résultats :

Id	Prenom	Nom	Date_achat	Num_facture	Prix_total
1	Aimée	Marechal	2013-01-23	A00103	203.14
1	Aimée	Marechal	2013-02-14	A00104	124.00
2	Esmée	Lefort	2013-02-17	A00105	149.45
2	Esmée	Lefort	2013-02-21	A00106	235.35
3	Marine	Prevost	NULL	NULL	NULL
4	Luc	Rolland	NULL	NULL	NULL

Les dernières lignes montrent des utilisateurs qui n'ont effectuée aucune commande. La ligne retourne la valeur NULL pour les colonnes concernant les achats qu'ils n'ont pas effectués.

Filtrer sur la valeur NULL

Attention, la valeur NULL n'est pas une chaîne de caractère. Pour filtrer sur ces caractères il faut utiliser la commande IS NULL. Par exemple, pour lister les utilisateurs qui n'ont pas effectués d'achats il est possible d'utiliser la requête suivante.

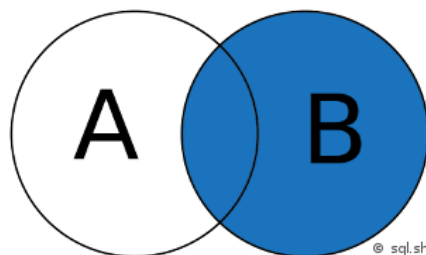
```
SELECT id, prenom, nom, utilisateur_id
FROM utilisateur
LEFT JOIN commande ON utilisateur.id = commande.utilisateur_id
WHERE utilisateur_id IS NULL
```

Résultats :

Id	Prenom	Nom	Utilisateur_id
3	Marine	Prevost	NULL
4	Luc	Rolland	NULL

SQL RIGHT JOIN

En SQL, la commande RIGHT JOIN (ou RIGHT OUTER JOIN) est un type de jointure entre 2 tables qui permet de retourner tous les enregistrements de la table de droite (right = droite) même s'il n'y a pas de correspondance avec la table de gauche. S'il y a un enregistrement de la table de droite qui ne trouve pas de correspondance dans la table de gauche, alors les colonnes de la table de gauche auront NULL pour valeur.



Syntaxe

L'utilisation de cette commande SQL s'effectue de la façon suivante :

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

Exemple

Prenons l'exemple d'une base de données qui contient des utilisateurs et un historique d'achat de ces utilisateurs. Cette 2 tables sont reliées entre grâce à la colonne **utilisateur_id** de la table des commandes. Cela permet de savoir à quel utilisateur est associé un achat.

Table utilisateur :

Id	NOM	PRENOM	MAIL	VILLE
1	Aimée	Marechal	aime.marechal@example.com	Paris
2	Esmée	Lefort	esmee.lefort@example.com	Lyon
3	Marine	Prevost	m.prevost@example.com	Lille
4	Luc	Rolland	lucrolland@example.com	Marseille

Table commande :

Utilisateur_id	Date_achat	Num_facture	Prix_total
1	2013-01-23	A00103	203.14
1	2013-02-14	A00104	124.00
2	2013-02-17	A00105	149.45
3	2013-02-21	A00106	235.35
5	2013-03-02	A00107	47.58

Pour afficher toutes les commandes avec le nom de l'utilisateur correspondant il est normalement d'habitude d'utiliser INNER JOIN en SQL. Malheureusement, si l'utilisateur a été supprimé de la table, alors ça ne retourne pas l'achat. L'utilisation de RIGHT JOIN permet de retourner tous les achats et d'afficher le nom de l'utilisateur s'il existe. Pour cela il convient d'utiliser cette requête :

```
SELECT id, prenom, nom, utilisateur_id, date_achat, num_facture
FROM utilisateur
RIGHT JOIN commande ON utilisateur.id = commande.utilisateur_id
```

Résultats :

Id	Prenom	Nom	Utilisateur_id	Date_achat	Num_facture
1	Aimée	Marechal	1	2013-01-23	A00103
1	Aimée	Marechal	1	2013-02-14	A00104
2	Esmée	Lefort	2	2013-02-17	A00105
3	Marine	Prevost	3	2013-02-21	A00106
NULL	NULL	NULL	5	2013-03-02	A00107