

# Les jointures

## Diagramme fort utile sur les jointures

On définit donc quel est notre table "de gauche" et notre table "de droite". Ensuite dépendant des informations que l'on veut obtenir on va utiliser différents types de jointures.

- Pour avoir seulement les tuples ayant une certaine valeur commune dans les 2 tables

```
SELECT b.id_bien, b.localite, c.prenom, c.nom
FROM Bien b
JOIN Client c ON b.id_client = c.id_client
```

- Pour avoir toute la table de gauche + les tuples communs aux deux tables

```
SELECT c.nom, c.prenom, o.montant, o.statut
FROM Candidat c
LEFT OUTER JOIN Offre o ON o.id_candidat = c.id_candidat
```

- Pour avoir **uniquement** les tuples unique à la table de gauche, et non présent sur la table de droite

```
SELECT c.nom, c.prenom, o.montant, o.statut
FROM Candidat c
LEFT OUTER JOIN Offre o ON o.id_candidat = c.id_candidat
WHERE o.id_candidat IS NULL
```

- Pour avoir toutes les données des deux tables

```
SELECT c.nom, c.prenom, o.montant, o.statut
FROM Candidat c
FULL OUTER JOIN Offre o ON c.id_candidat = o.id_candidat
```

- Pour avoir seulement les tuples unique à chaque table, mais pas ceux qui sont communs

```
SELECT c.nom, c.prenom, o.montant, o.statut
FROM Candidat c
FULL OUTER JOIN Offre o ON c.id_candidat = o.id_candidat
WHERE c.id_candidat IS NULL OR o.id_candidat IS NULL
```

- Pour tous les `RIGHT OUTER JOIN` c'est la même chose que dit plus tot, mais en inversant comme vu dans le schéma plus haut

# Quand il y a plusieurs joins (soit plus de 2 tables)

Après un JOIN, les deux tables liées se comporte comme une nouvelle table. Soit du point de vue du JOIN suivant, le résultat du premier JOIN se comporte comme si c'était la table 'A' sur le schéma.

Voici un exemple provenant de l'examen formatif de décembre : [Énoncé de la question](#)

Donc si on représente la situation avec des diagrammes ça nous donne ceci :

Diagramme de la situation

Et une fois représenté en SQL ça nous donne :

```
-- On prends les valeurs que l'on veut dans le select
SELECT v.nom AS nom_vendeur, v.prenom AS prenom_vendeur, m.texte AS message,
TO_CHAR(m.date_message, 'dd-month-yyyy') AS date_message
-- On va partir de la table Message (mais on pourrait aussi partir de vendeur)
FROM Message m
-- On va regrouper Message et Alerter
LEFT OUTER JOIN Alerter a ON a.id_message = m.id_message
-- On va ajouter Vendeur à notre (Message + Alerter)
FULL OUTER JOIN Vendeur v ON a.id_vendeur = v.id_vendeur
-- On va éliminer tout les tuples qui sont reliés en supprimant tous les liens de Alerter
WHERE a.id_message IS NULL;
```

---

Revision #1

Created 26 April 2023 19:01:32 by SnowCode

Updated 26 April 2023 19:01:32 by SnowCode