Les jointures

Définition

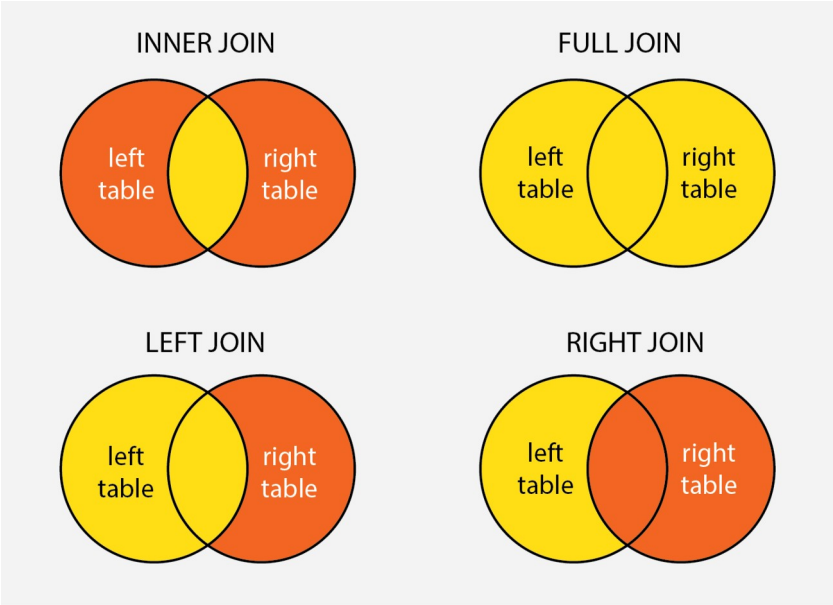
Une jointure est une opération permettant de combiner les enregistrements de deux ou plusieurs tables en fonction de leurs colonnes communes.

Plus précisément, une jointure permet de lier les enregistrements de deux tables en comparant les valeurs de colonnes spécifiques dans chacune des tables.

Il existe différents types de jointures, en SQL, notamment :

- la jointure interne ("INNER JOIN"),
- la jointure externe ("OUTER JOIN"),
- la jointure gauche ("LEFT JOIN"),
- la jointure droite ("RIGHT JOIN"),
- la jointure croisée ("CROSS JOIN").

Chacun de ces types de jointures est utilisé pour combiner les enregistrements de différentes manières en fonction des besoins spécifiques de l'utilisateur.



Nous nous intéressons dans ce cours uniquement uniquement aux jointures internes.

Requête avec jointure

Voici un exemple de requête utilisant la jointure interne (INNER JOIN) pour joindre deux tables "table1" et "table2" sur la colonne "id" :

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

Il est possible, aussi, de réaliser la jointure sans utiliser l'opérateur INNER JOIN en utilisant la syntaxe de jointure plus ancienne avec la clause WHERE.

```
SELECT * FROM table1, table2 WHERE table1.id = table2.id;
```

Il est préférable d'utiliser l'opérateur INNER JOIN plutôt que la méthode de jointure plus ancienne basée sur la clause WHERE :

1. Clarté du code : l'utilisation de INNER JOIN rend le code SQL plus compréhensible.
2. Performance : l'utilisation d'INNER JOIN est parfois plus performante car elle permet à l'optimiseur de requête du SGBD de mieux comprendre l'intention de la requête.
3. Compatibilité : l'opérateur INNER JOIN est standard SQL, ce qui signifie que les requêtes utilisant cet opérateur sont plus facilement portables entre différents SGBD.

Cependant, pour respecter les directives du curriculum, et dans la suite de ce cours, on optera pour l'ancienne méthode avec la clause WHERE.

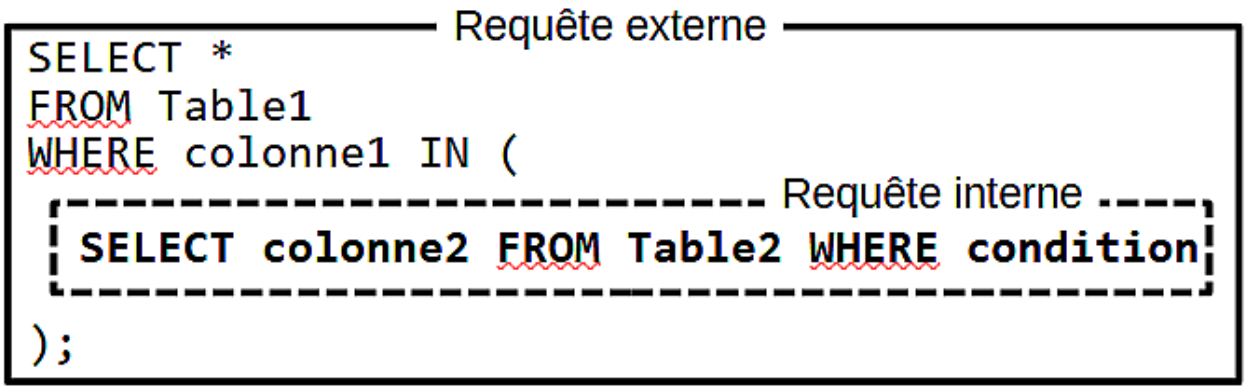
Requêtes imbriquées

Définition

Les requêtes imbriquées en SQL sont des requêtes qui sont incorporées dans d'autres requêtes. Elles sont utilisées pour récupérer des données plus complexes en combinant les résultats de plusieurs requêtes.

Requête imbriquée

Une requête imbriquée est écrite sous la forme :



La requête interne est exécutée en premier et les résultats sont ensuite utilisés dans la requête externe pour fournir une réponse plus précise et spécifique.

Exemple 1

La requête suivante utilise une sous-requête pour récupérer l'ID du département "Ventes" dans la table "Départements". La requête principale utilise, ensuite, cet ID pour sélectionner tous les employés du département "Ventes" dans la table "Employés".

```
SELECT * FROM Employés WHERE id_departement=(SELECT id FROM Départements WHERE nom='Ventes');
```

Exemple 2

La requête suivante utilise une sous-requête pour récupérer tous les IDs des clients ayant passé une commande d'un montant supérieur à 1000DT. La requête principale utilise ensuite ces IDs pour sélectionner tous les clients correspondants dans la table "Clients".

```
SELECT * FROM Clients WHERE id IN (SELECT id_client FROM Commandes WHERE montant > 1000);
```

Application

Les tables suivantes sont extraites d'une base de données d'un établissement scolaire.

Table **Eleves**

numel	nom	prenom	genre	datenaiss
E004	Lajmi	Isra	F	2004-11-29
E012	Hassen	Maha	F	2004-11-29
E011	Ben Saber	Sami	M	2004-11-29
E006	Ben Hmida	Adem	M	2005-09-10
E008	Jedidi	Ayhem	M	2006-02-18
E009	Gaaloul	Abrar	F	2006-02-24
E010	Garzoul	Raslène	M	2006-03-10
E007	Ben Massoud	Amal	F	2006-05-05
E003	Ben Moussa	Ahmed	M	2006-08-20
E005	Kassouma	Mohamed Amine	M	2006-10-15
E002	Lajmi	Eya	F	2006-12-18
E001	Mazzez	Adem	M	2007-04-27

Table **ClassesEleves**

numel	numcl	annee
E005	C003	2020
E010	C003	2020
E001	C001	2021
E002	C002	2021
E004	C002	2021
E003	C003	2021
E005	C003	2021
E006	C003	2021
E007	C003	2021
E008	C003	2021
E009	C003	2021
E010	C003	2021
E001	C004	2022
E002	C004	2022
E003	C004	2022
E004	C004	2022
E005	C004	2022
E006	C005	2022
E007	C005	2022
E008	C005	2022
E009	C005	2022
E010	C005	2022

Table **Classes**

numcl	libelle	niveau	section	numord
C001	1S1	1	S	1
C002	1S2	1	S	2
C006	2ECO1	2	ECO	1
C007	2ECO2	2	ECO	2
C003	2INFO1	2	INFO	1
C004	2INFO2	2	INFO	2
C005	3INFO1	3	INFO	1

Travail demandé

1. Dessiner le schéma graphique de cette base de données.
2. Écrire les requêtes SQL suivantes :
   a) Afficher la liste de tous les élèves (nom, prenom, libelle, annee) ordonnée par année, les plus récents en premier, puis par ordre alphabétique du libelle.
   b) Afficher la liste des élèves (nom, prenom, libelle, annee) qui font partie d'une classe de niveau 2 ordonnée par année, puis par libelle.
   c) Afficher la liste des élèves (nom, prenom) de la classe '2INFO2' pour l'année 2022.
   d) Afficher quelles sont les classes (libelle, annee) dans lesquelles a étudié l'élève 'Lejmi Eya'.
   e) Afficher les classes des élèves (nom, prenom, niveau, section) à la rentrée scolaire 2023 en supposant qu'ils vont réussir tous et qu'ils ne vont pas changer de section.
   f) Déterminer la date de naissance de l'élève le plus agé, puis afficher la liste des élèves (nom, prenom) les plus agés.
   g) Afficher la classe (numcl, libelle) dans laquelle étudie 'Adem Mazzez' en 2022, puis afficher la liste des élèves qui étudient avec lui en même classe (nom, prenom, genre).
   h) Afficher les élèves qui n'appartiennent à aucune classe.
   i) Afficher les classes qui ne comptent aucun élèves.

```sql
-- 2.a
SELECT nom, prenom, libelle, annee
FROM Eleves AS e, Classes AS c, ClassesEleves AS ce
WHERE ce.numel = e.numel AND
      ce.numcl = c.numcl
ORDER BY annee DESC, libelle;
```

```sql
-- 2.b
SELECT nom, prenom, libelle, annee
FROM Eleves AS e, Classes AS c, ClassesEleves AS ce
WHERE ce.numel = e.numel AND
      ce.numcl = c.numcl AND
    niveau = 2
ORDER BY annee, libelle;
```

```sql
-- 2.c
SELECT nom, prenom, libelle, annee
FROM Eleves AS e, Classes AS c, ClassesEleves AS ce
WHERE ce.numel = e.numel AND
      ce.numcl = c.numcl AND
    libelle = '2INFO2' AND
    annee = 2022;
```

```sql
-- 2.d
SELECT libelle, annee
FROM Eleves AS e, Classes AS c, ClassesEleves AS ce
WHERE ce.numel = e.numel AND
      ce.numcl = c.numcl AND
      nom = 'Lajmi' AND
      prenom = 'Eya'
ORDER BY annee;
```

```sql
-- 2.e
SELECT nom, prenom, CONCAT((niveau+1), section) AS niveau23
FROM Eleves AS e, Classes AS c, ClassesEleves AS ce
WHERE ce.numel = e.numel AND
      ce.numcl = c.numcl AND
      annee = 2022;
```

```sql
-- 2.f
SELECT MIN(datenaiss) AS datenaiss
FROM Eleves;

SELECT nom, prenom
FROM Eleves
WHERE datenaiss IN (SELECT MIN(datenaiss) FROM Eleves);
```

```sql
-- 2.g
SELECT ce.numcl, libelle
FROM Classes AS c, Eleves AS e, ClassesEleves AS ce
WHERE ce.numcl = c.numcl AND
    ce.numel = e.numel AND
    nom = 'Mazzez' AND
    prenom = 'Adem' AND
    annee = 2022;
```

```
SELECT nom, prenom, genre
FROM Eleves AS e, ClassesEleves AS ce
WHERE ce.numel = e.numel AND
    ce.numcl = (
        SELECT ce.numcl
        FROM Classes AS c, Eleves AS e, ClassesEleves AS ce
        WHERE ce.numcl = c.numcl AND
            ce.numel = e.numel AND
            nom = 'Mazzez' AND
            prenom = 'Adem' AND
            annee = 2022
    ) AND
    annee = 2022;
```

```
-- 2.h
SELECT *
FROM Eleves
WHERE numel NOT IN (
    SELECT DISTINCT numel FROM ClassesEleves
);
```

```
-- 2.i
SELECT *
FROM Classes
WHERE numcl NOT IN (
    SELECT DISTINCT numcl FROM ClassesEleves
);
```