

4ème Partie : Les Jointures

Objectifs de cette partie :

- i. Equijointure et jointure interne, Jointure naturelle
- ii. Jointure externe
- iii. Non équijointure
- iv. Produit cartésien

Les Jointures. Définition.

- Une jointure permet **d'extraire** des données de **plusieurs tables** à la fois.
- La condition de jointure peut être exprimée dans la clause « WHERE » ou « ON ».
- **Précédez** le nom de la colonne par le nom de la table lorsque celui-ci figure dans plusieurs tables

Types de Jointures.

- Equijointure ou jointure interne
- Jointure naturelle
- Jointure externe
- Non équijointure
- Produit cartésien

Equijointure. Définition

- Une jointure **interne** ou **équijointure** est une jointure avec une condition de jointure contenant **un opérateur d'égalité**.
- C'est la plus répandue, elle combine les lignes qui ont des valeurs **équivalentes** pour les colonnes de la jointure.
- Généralement dans ce type de jointure, ce sont les **Primary Key** et **Foreign Key** qui sont utilisées.

Equijointure. Syntaxe

(1/2)

```
SELECT table1.column, table2.column  
  
FROM   table1  
  
INNER JOIN table2  
  
ON (table1.column_name = table2.column_name);
```

Equijointure. Syntaxe

(2/2)

```
SELECT      T1.Colonne1, ...T1.ColonneN,  
            T2.Colonne1, ...T2.ColonneN  
FROM table1 T1 INNER JOIN table2 T2  
ON T1.C1=T2.C1  
WHERE <Condition(s)>
```

- ✓ **T1** : Alias de la table : table1
- ✓ **T2** : Alias de la table : table2
- ✓ **C1** : colonne permettant la jointure entre les 2 tables
(primary key + foreign key)

Équivalent à :

```
SELECT      T1.Colonne1, ...T1.ColonneN,  
            T2.Colonne1, ...T2.ColonneN  
FROM T1 ,T2  
WHERE T1.C1=T2.C1 AND <Condition(s)>
```

Equijointure. Exemple

Exemple1:

```
SELECT nomE, e.numdept, nomdept  
FROM emp e INNER JOIN dept d  
ON e.numdept=d.numdept;
```

Exemple2:

```
SELECT nomE, e.numdept, nomdept  
FROM emp e INNER JOIN dept d  
ON e.numdept=d.numdept  
WHERE e.numdept=11;
```

Jointure naturelle.

Définition

- La jointure naturelle est une **équijointure**.
- La clause **NATURAL JOIN** est basée sur toutes les colonnes des deux tables portant **le même nom**.
- Elle sélectionne les lignes des deux tables dont les valeurs sont **identiques** dans toutes les colonnes qui correspondent.
- Si les colonnes portant le même nom présentent des types de données **différents**, une erreur est renvoyée.

Jointure naturelle. Syntaxe

```
SELECT colonne1, colonne2...  
FROM Table1  
NATURAL JOIN Table2  
WHERE Condition(s) ;
```



Conditions autres que la condition
de jointure

Jointure naturelle. Exemple

Exemple 1:

```
SELECT nomE, numdept,nomdept  
FROM emp NATURAL JOIN dept;
```

Exemple 2:

```
SELECT nomE, numdept,nomdept  
FROM emp NATURAL JOIN dept  
WHERE numdept=11;
```

Non équijointure.

Définition

- Il s'agit là d'utiliser n'importe quelle condition de jointure entre deux tables, exceptée la stricte égalité. Ce peuvent être les conditions suivantes :

>	supérieur
>=	supérieur ou égal
<	inférieur
<=	inférieur ou égal
<>	différent de
IN	dans un ensemble
LIKE	correspondance partielle
BETWEEN ... AND ...	entre deux valeurs
EXISTS	dans une table

Jointure externe.

Définition

- JOINTURES EXTERNES = **OUTER JOINS**.
- Une jointure externe **élargie** le résultat d'une jointure interne (INNER JOINS) et permet d'extraire des enregistrements qui ne répondent pas aux critères de jointure.
- Une jointure externe renvoie **toutes les lignes** qui satisfont la condition de jointure et retourne également une partie de ces lignes de la table pour laquelle aucune des lignes de l'autre ne satisfait pas la condition de jointure.

Jointure externe. Types de jointures externes

```
SELECT t1.column, t2.column..  
FROM table1 t1  
  
LEFT | RIGHT | FULL OUTER JOIN table2 t2 ON  
(t1.column_name = t2.column_name);
```

- Le sens de la jointure externe **LEFT** ou **RIGHT** de la clause **OUTER JOIN** désigne la table **dominante**.
- **FULL = INNER + LEFT + RIGHT**

Jointure externe. Gauche

Exemple:

```
SELECT nomE, e.numdept, nomdept  
FROM dept d LEFT JOIN emp e  
ON e.numdept=d.numdept
```

NOME	NUMDEPT	NOMDEPT
afia tarek	11	informatique
benflen salah	11	informatique
bensalah salma	11	informatique
hamid ali	11	informatique
benafia ahmed	11	informatique
benflen ali	21	télécommunication
benafia tarek	21	télécommunication
hena salma	31	électromécanique
-	-	genie logiciel

Jointure externe. Droite

Exemple:

```
SELECT nomE, e.numdept, nomdept  
FROM dept d RIGHT JOIN emp e  
ON e.numdept=d.numdept
```

NOME	NUMDEPT	NOMDEPT
test test	-	-
benafia ahmed	11	informatique
hamid ali	11	informatique
bensalah salma	11	informatique
afia tarek	11	informatique
benflen salah	11	informatique
benflen ali	21	télécommunication
benaf salma	31	électromécanique
benafia tarek	21	télécommunication

Jointure externe. complète

Exemple:

```
SELECT nomE, e.numdept, nomdept  
FROM dept d FULL JOIN emp e  
ON e.numdept=d.numdept
```

NOME	NUMDEPT	NOMDEPT
afia terek	11	informatique
benflen salah	11	informatique
bensalah salma	11	informatique
hamid ali	11	informatique
benafia ahmed	11	informatique
benflen ali	21	télécommunication
benafia terek	21	télécommunication
bena salma	31	électromécanique
-	-	genie logiciel
test test	-	-

Produit Cartésien. Définition

On obtient un produit cartésien lorsque :

- Une condition de jointure est omise
- Une condition de jointure est incorrecte
- A chaque ligne de la table 1 sont jointes toutes les lignes de la table 2.
- Le nombre de lignes renvoyés est égal $n1 * n2$ où
 - ✓ $n1$ est le nombre de lignes de la table 1
 - ✓ $n2$ est le nombre de lignes de la table 2

Produit Cartésien. Syntaxe

```
SELECT « nom_colonne1 », « nom_colonne2 »...  
FROM table1 t1  
CROSS JOIN table2 t2
```

Exemple:

```
Select * from employees CROSS JOIN departments
```

→ On va avoir **107*27** lignes = 2889 lignes

Produit Cartésien. Exemple

SELECT * From emp e

INNER JOIN dept d

ON e.numdept=e.numdept

▪ OU

SELECT * From emp, dept

→ On va avoir **9*4** lignes

Jointure. Récapitulatif

Jointure interne	<pre>SELECT FROM <table gauche> [INNER]JOIN <table droite> ON <condition de jointure></pre>
Jointure externe	<pre>SELECT FROM <table gauche> LEFT RIGHT FULL [OUTER]JOIN <table droite> ON <condition de jointure></pre>
Jointure naturelle	<pre>SELECT FROM <table gauche> NATURAL JOIN <table droite> (USING <noms de colonnes>]</pre>
Jointure croisée	<pre>SELECT FROM <table gauche> CROSS JOIN <table droite></pre>

5ème Partie : Les Opérateurs ensemblistes

- i. UNION
- ii. UNION ALL
- iii. INTERSECT
- iv. MINUS

Les opérateurs ensemblistes.

OPERATEUR	DESCRIPTION
INTERSECT	Ramène toutes les lignes communes aux deux requêtes
UNION	Toutes les lignes distinctes ramenées par les deux requêtes
UNION ALL	Toutes les lignes ramenées par les deux requêtes y compris les doublons
MINUS	Toutes les lignes ramenées par la première requête sauf les lignes ramenées par la seconde requête

Union(ALL). Exemple

- Créer une table d'archivage qui contient les informations des employés qui ont été embauchés après l'année 2009.
- Ecrire une requête qui va afficher tous les employés:

```
SELECT * from emp
```

```
Union (ALL)
```

```
SELECT * from emp2
```

Intersect/Minus. Exemple

Ecrire les requêtes permettant de :

- Afficher les employés qui **ont été archivés**.
- Afficher les employés qui **n'ont pas été archivés**.

```
SELECT * from emp
```

Intersect/minus

```
SELECT * from emp2;
```


Exercice d'application.

- Ecrire une requête qui permet d'afficher ce résultat à partir de la table 'dept' :

NOMDEPT	NUMDEPT
électromécanique	31
genie logiciel	41
informatique	11
télécommunication	21
TOTAL DES DEPARTEMENTS	4