



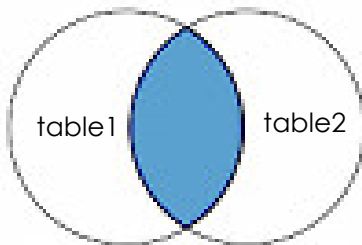
Les jointures

Les jointures

- La jointure va nous permettre de sélectionner des informations dans plusieurs tables ,grâce aux relations existant entre ces dernières.
- Il existe des types de jointure: jointure interne, jointure externe(jointure de gauche, jointure de droite...)

Jointure interne(INNER JOIN)

- Ce type de jointure va retourner les entrées des deux tables **si jamais une correspondance a été trouvée entre les deux tables.**



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

Cette syntaxe va 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.

Jointure interne(INNER JOIN)

○ Exemple 1:

si on veut sélectionner les informations des stagiaires et les information de la filières où sont inscrit, on écrira :

```
Select nomstg, prenomstg, datenaissance, nomfiliere  
FROM stagiaires INNER JOIN filiere ON  
Filiere.IdFil=stagiaires.IdFil
```

Jointure interne(INNER JOIN)

- **Exemple 2:**

On va ajouter une condition à la requête précédente:

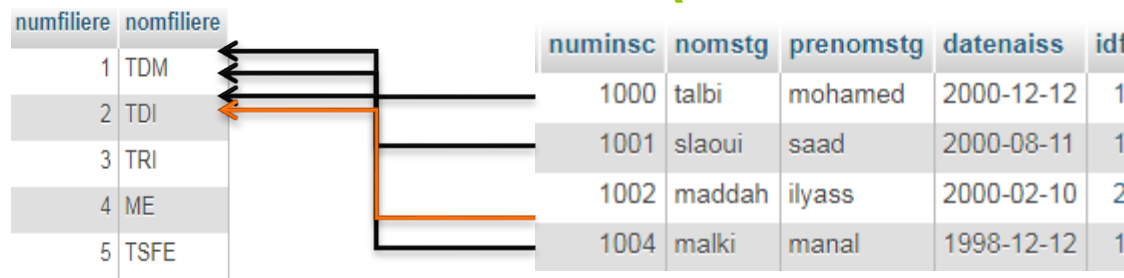
```
Select nomstg, prenomstg, datenaissance, nomfiliere FROM  
stagiaires INNER JOIN filiere ON  
Filiere.IdFil=Stagiaires.IdFil where year(datenaissance) =2000  
AND nomfiliere IN('TDM','TDI')
```

Rq:

- Si on a des colonnes qui se trouvent dans des tables différentes (comme les clés étrangères), lorsqu' on fait la sélection, on doit préciser d'où provient cette colonne, et cela en écrivant:

NomTable.NomChamp (exp: stagiaires.idf)

Jointure interne (INNER JOIN)



Select nomstg, prenomstg, datenaissance, nomfiliere FROM stagiaires **INNER JOIN** filiere ON Filieres.IdFil=Stagiaires.IdFil

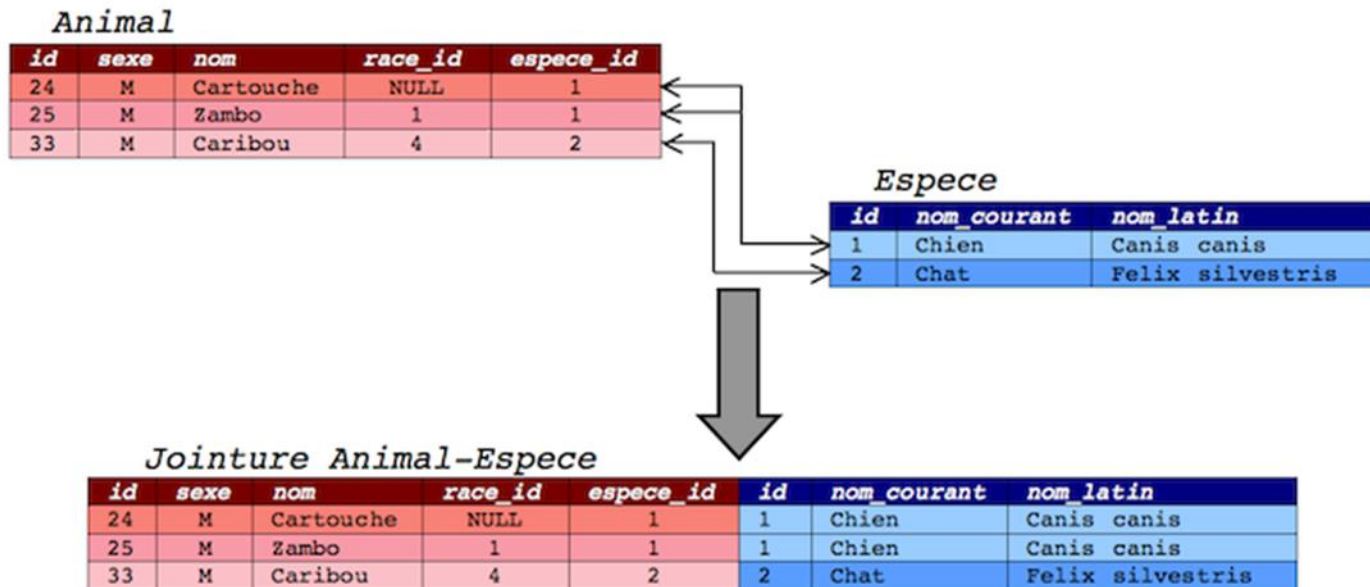
Cela va nous retourner juste les enregistrements dans lesquels on il y'a une correspondance entre les deux tables. C'est-à-dire que la filière **TRI, ME et TSFE** ne seront pas affichée.

nomstg	prenomstg	datenaiss	nomfiliere
talbi	mohamed	2000-12-12	TDM
slaoui	saad	2000-08-11	TDM
malki	manal	1998-12-12	TDM
maddah	ilyass	2000-02-10	TDI

Jointure interne (INNER JOIN)

• Autre exemple :

Select * from **Animal** **inner join** **Espec** on **Animal.espece_id=Espec.id**



Autre syntaxe utilisée pour la jointure interne

- On peut écrire la requête de l'exemple 2 de la manière suivante:

```
Select nomstg, prenomstg, datenaissance, nomfiliere  
FROM stagiaires , filiere where  
Filiere.IdFil=stagiaires.IdFil and year(datenaissance) =2000  
AND nomfiliere IN('TDM','TDI')
```


Le mot USING

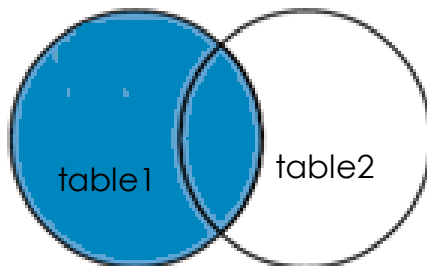
```
select * from produits inner join categories using (`code  
categorie`)
```

- Equivalent à:

```
select * from produits inner join categories on produits  
.`code categorie`=`categories`.`code categorie`
```

Jointure à gauche (LEFT JOIN ou LEFT OUTER JOIN)

- LEFT JOIN 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 table.



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

Jointure à gauche (LEFT JOIN)

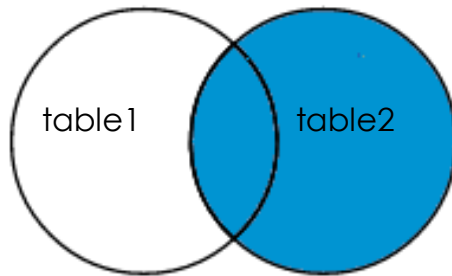
Exemple:

```
Select * FROM filieres LEFT JOIN stagiaires ON
Filiere.IdFil=Stagiaires.IdFil
```

numfiliere	nomfiliere	numinsc	nomstg	prenomstg	datenaiss	idf
1	TDM	1000	talbi	mohamed	2000-12-12	1
1	TDM	1001	slaoui	saad	2000-08-11	1
2	TDI	1002	maddah	ilyass	2000-02-10	2
1	TDM	1004	malki	manal	1998-12-12	1
3	TRI	NULL	NULL	NULL	NULL	NULL
4	ME	NULL	NULL	NULL	NULL	NULL
5	TSFE	NULL	NULL	NULL	NULL	NULL

Jointure à droite (RIGHT JOIN)


- RIGHT JOIN 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.



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

La différence entre INNER JOIN et LEFT JOIN

- LEFT JOIN renvoie la partie "**GAUCHE**" c'est à dire que toutes les lignes de la table gauche seront retournées, même s'il n'y a pas de ligne correspondante dans la table droite.
- Tandis que **INNER JOIN** renvoie toutes les lignes de deux tables qui remplissent une condition donnée.



Les sous requêtes

Les sous requêtes

- Une sous-requête est une requête **à l'intérieur** d'une autre requête. Avec le SQL, vous pouvez construire des requêtes imbriquées sur autant de niveaux que vous voulez.
- On peut utiliser les sous requêtes avec la commande de sélection, modification, insertion ou suppression.

Les sous requêtes

- Lorsqu'on fait une requête SELECT, le résultat est **sous forme de table**. Ces tables de résultats peuvent avoir :
 - ❖ plusieurs colonnes et plusieurs lignes ;
 - ❖ plusieurs colonnes, mais une seule ligne ;
 - ❖ plusieurs lignes, mais une seule colonne ;
 - ❖ ou encore une seule ligne et une seule colonne (c'est-à-dire juste **une** valeur).

Sous requête qui retourne une seule valeur

- Syntaxe:

```
SELECT * FROM table1 where ColonneX OPERATEUR  
(SELECT champy from table2 where.....)
```

- Où operateur est un opérateur de comparaison (=, >, <, >=, <=, !=)
- La sous requête c'est la requête interne entre parenthèses.

Sous requête qui retourne une seule valeur

- Exemple 1:

```
Select * from commandes where montant= (select min(montant) from commandes)
```

- Exemple 2:

```
Select * from employés where salaire > (select avg(salaire) from employés)
```

Sous requête qui retourne une seule valeur

- Exemple 3:

```
Select * from clients where `code client` = (select `code client` from commandes where `n commande`=10000)
```

- Cette requête est équivalente à :

```
Select clients.* from clients inner join commandes on commandes.`code client`=clients.`code client` where `n commande`=10000
```

Sous requête qui retourne une colonne de valeurs

- Syntaxe 1:

```
SELECT * from table1 where colonneX IN/NOT IN  
(SELECT colonneY from table2)
```

- **Rq:** puisque la sous requête retourne une colonne de valeurs, impossible de mettre
=

Sous requête qui retourne une colonne de valeurs

- Exemple 1:

```
Select produits.* from produits where `code categorie`  
IN (select `code categorie` from categories where  
`nom de categorie` IN('Boissons','Viandes') )
```

- Exemple 2:

```
Select * from clients where codecli NOT IN (select  
codecli from commandes where year(`date  
commande`) between 1990 and 1991)
```

Sous requête qui retourne une colonne: avec la clause ALL

- La commande **ALL**, permet de comparer une valeur **avec toutes** les valeurs d'une sous requête.
- Syntaxe:

```
SELECT * from table1 where colonneX >=ALL (select  
colonneY from table2)
```

Sous requête avec la clause ALL

- Exemple:

```
Select * from produits where prix >= all(select prix from  
produits)
```

On peut utiliser d'autres opérateurs de
comparaison (!=, =, >, <, >=, <=)