

Jointures

produit cartésien

```
T1 CROSS JOIN T2
```

```
SELECT * FROM T1 CROSS JOIN T2;  
<=> SELECT * FROM T1,T2;
```

jointure naturelle interne

```
T1 NATURAL JOIN T2
```

(déjà vu) ATTENTION: jointure sur tous les attributs communs de `T1` et `T2`
Vérifiez que l'on ne joint pas sur trop d'attributs.

1. jointures générale interne

```
T1 [INNER] JOIN T2 ON expression_boulienne  
T1 [INNER] JOIN T2 USING  
(att1,att2,...attrn)
```

Avec ON est suivis des condition de jointure:

```
SELECT * FROM T1 JOIN T2 ON exp; <=> SELECT * FROM T1 CROSS JOIN T2 WHERE
```

1.1. USING

idem que `ON`

```
T1.attr1 == T2.attr1 AND T1.attr2 == T2.attr2 AND ... AND T1.attrn == T2.attrn
```

Exemple

les couples de produits différents ayant la même description

```
SELECT desc_produit, p1.id_produit, p2.id_produit  
FROM produit AS p1 JOIN produit AS p2 USING (desc_produit)  
WHERE p1.id_produit != p2.id_produit;
```

```
SELECT p1.desc_produit, p1.id_produit, p2.id_produit  
FROM produit p1 JOIN produit p2  
ON p1.desc_produit = p2.desc_produit  
AND p1.id_produit != p2.id_produit
```

2. Jointure externe

On a deux tables

S et R

A	B	-	B	C
a	x	-	x	A
b	x	-	t	B
a	y	-	t	C

```
SELECT * FROM S NATURAL R;
```

A	B	C
a	x	A
b	x	A

On a "perdu" des ligne de **S** et **R**

```
SELECT *  
FROM S NATURAL FULL OUTER JOIN R;
```

B	A	C
x	a	A
x	b	A
y	a	NULL
t	NULL	B
t	NULL	B

3. JOINTURE Á GAUCHE

```
SELECT *  
FROM S NATURAL LEFT OUTER JOIN R;
```

B	A	C
x	a	A
x	b	A
y	a	NULL

4. JOINTURE À DROITE

```
SELECT *  
FROM S NATURAL RIGHT OUTER JOIN R;
```

B	A	C
x	a	A
x	b	A
t	NULL	B
T	NULL	C

Syntaxe:

```
T1 {LEFT | RIGHT | FULL} [OUTER] JOIN T2 ON expression
```

```
T1 _____  
JOIN T2 USING (attr1,attr2,...)
```

ATTENTION

```
SELECT *  
FROM T1 FULL OUTER JOIN T2  
ON T1.attr1 = T2.attr1 AND T1.attr2 = T2.attr2;
```

```
SELECT *  
FROM T1 FULL OUTER JOIN T2  
ON T1.attr1 = T2.attr1  
WHERE T1.attr2 = T2.attr2;
```

avec `T1 FULL OUTER JOIN T2 ON expression`

Le calcul `T1 X T2` auquel on ajoute : les lignes de T1 complété par des NULL
les lignes de T2 complété par des NULL

Puis sélection des lignes où expression est soit `TRUE` soit `NULL`

Exemple

Les couples(produit, magasin) où produit n'est pas dans le magasin

```
SELECT id_produit, id_magasin  
FROM (produit, CROSS JOIN magasin)  
LEFT OUTER JOIN stock USING (id_produit, id_magasin)  
WHERE quantite IS NULL;
```

id_produit	id_magasin	quantite		
100	1	10		
100	2	NULL		
100	3	1		
101	1	NULL		
101	2	NULL		
101	3	5		
...

Expression conditionnel 1-> CASE Syntaxe

```
CASE WHEN cond1 then result1
      WHEN cond2 then _____2
      [_____]
      [ELSE resultat]
END
```

retourne une valeur: retourne le premier resultI correspondant a une condI égale a TRUE Sinon retourne resultat (si ELSE) retourne NULL si pas de ELSE

Exemple On veut pour chaque couple (id_magasin,id_produit) la quantité en stock en mettant 0 si pas en stock

```
SELECT id_produit,id_magasin
      CASE WHEN quantite IS NULL THEN 0
      ELSE quantite END AS quantite
FROM (produit CROSS JOIN magasin) LEFT OUTER JOIN stock ON (id_produit,
```

On veut la moyenne des quantité en stock par produit

1ere Essai

```
SELECT id_produit,AVG(quantite)
FROM stock
GROUP BY id_produit;
```

NON FONCTIONNELLE

2eme Essai

```
WITH stock_bis AS
(SELECT id_produit,id_magasin
CASE
    WHEN quantite IS NULL THEN 0
    ELSE quantite
END as quantite
FROM (magasin CROSS JOIN produit)
LEFT OUTER JOIN stock USING(id_magasin,id_produit)
)

SELECT id_produit, AVG(quantite)
FROM stock_bis
GROUP BY id_produit
```