

JOINTURES en SQL

Sortes de jointures - Trucs - Avec agrégation



CLÉ ÉTRANGÈRE

...

Partie publique du site

Les maisons de production

Maisons de production

Les Productions Vic Pelletier

5 productions cinématographiques
500 minutes de vidéos offertes
depuis 1997

Le Théatre de la Maison croche

5 productions cinématographiques
700 minutes de vidéo offertes
depuis 2005

L'agence de communication Vidéo-toi

1 productions cinématographiques
3500 minutes de vidéo offertes
depuis 2012

index.php

Les productions Vic Pelletier

Maisons de production > Productions Vic Pelletier

Cette maison de production matanaise est blablablabla.....

Matane – depuis 1990 – 10 employés

Liste des films

La maison du pêcheur

55 minutes – Historique – Réalisation Jude Côté
Ce film raconte blablabla.....
.....

La maison du marin

75 minutes – Historique – Réalisation Lacroche
Ce film raconte blablabla.....
.....

producteur.php

La maison du pêcheur

Maisons de production > Productions Vic Pelletier > La maison du pêcheur

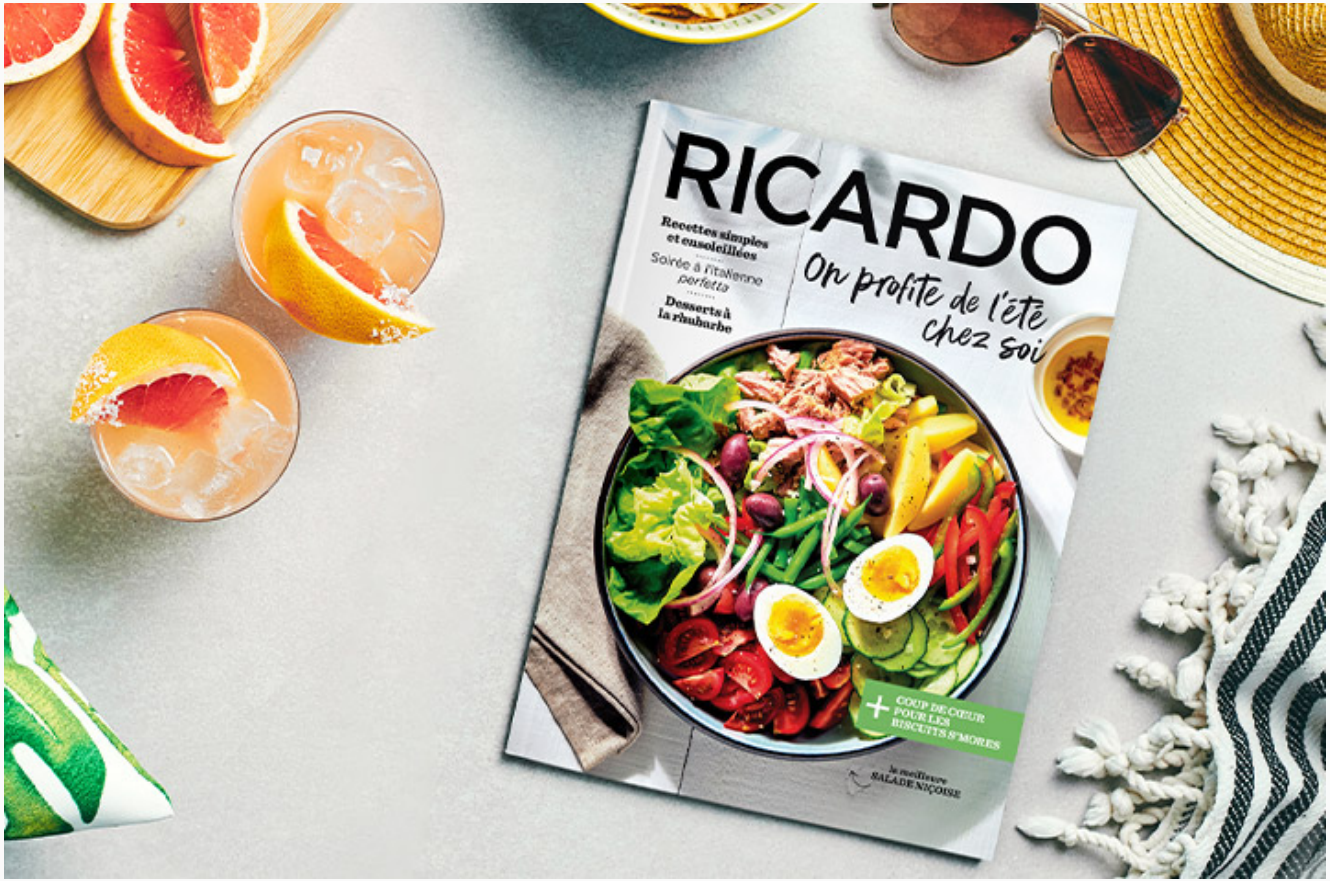


Classification : Général
Durée : 55 minutes
Réalisation : Jude Côté
Production : Vic Pelletier
Style : Historique
Budget : 1 million
Sortie : 3 octobre 2013

Ce film bla bla bla bla..... .....bla bla bla bla .....
.....bla bla bla bla .....bla bla bla bla
.....bla bla bla bla .....bla bla bla bla
.....bla bla bla bla .....bla bla bla bla .....
.....bla bla bla bla .....bla bla bla bla
.....bla bla bla bla .....bla bla bla bla
.....bla bla bla bla .....bla bla bla bla .....

film.php

Exemple de maisons de production qui ont chacune une liste de films



Exemple de parutions de revue ou magazines qui ont chacun plusieurs articles



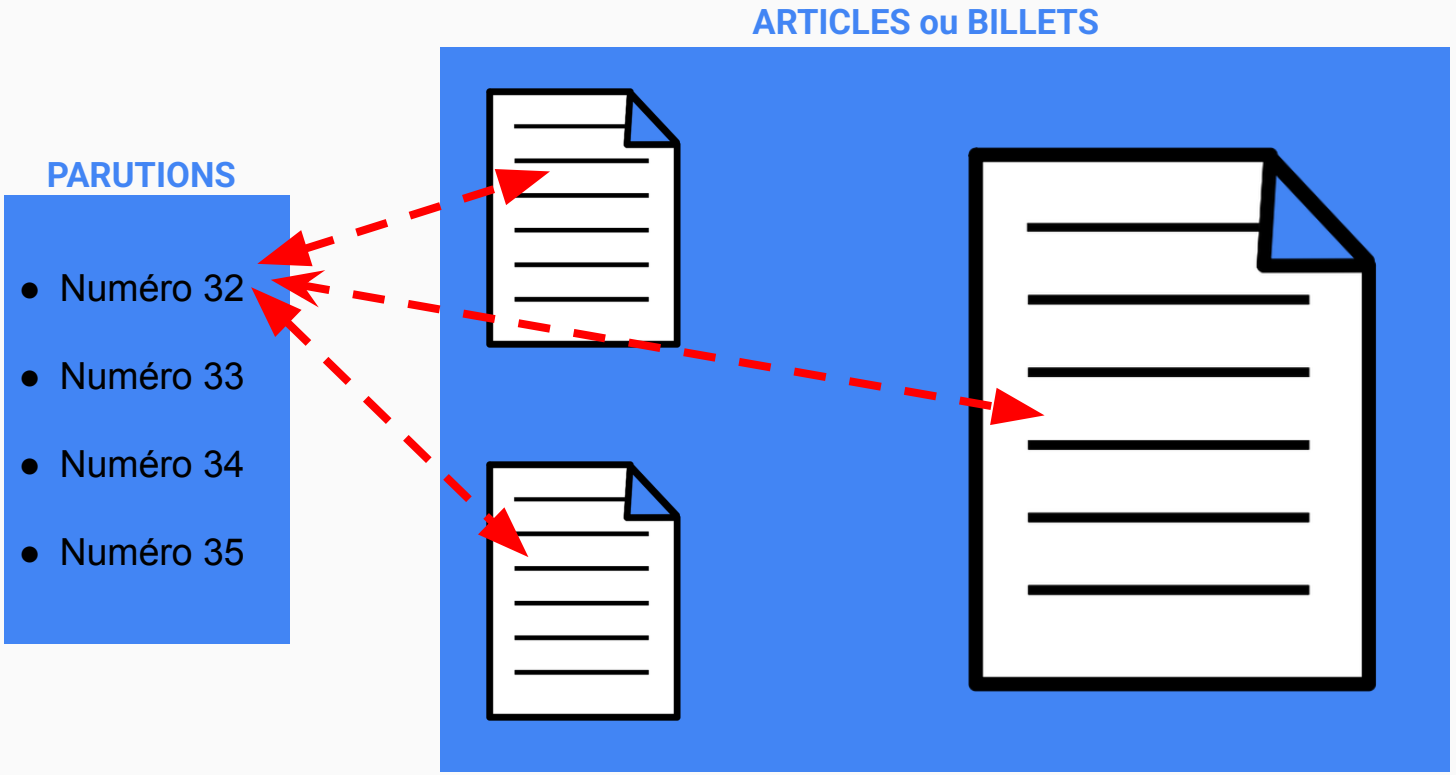




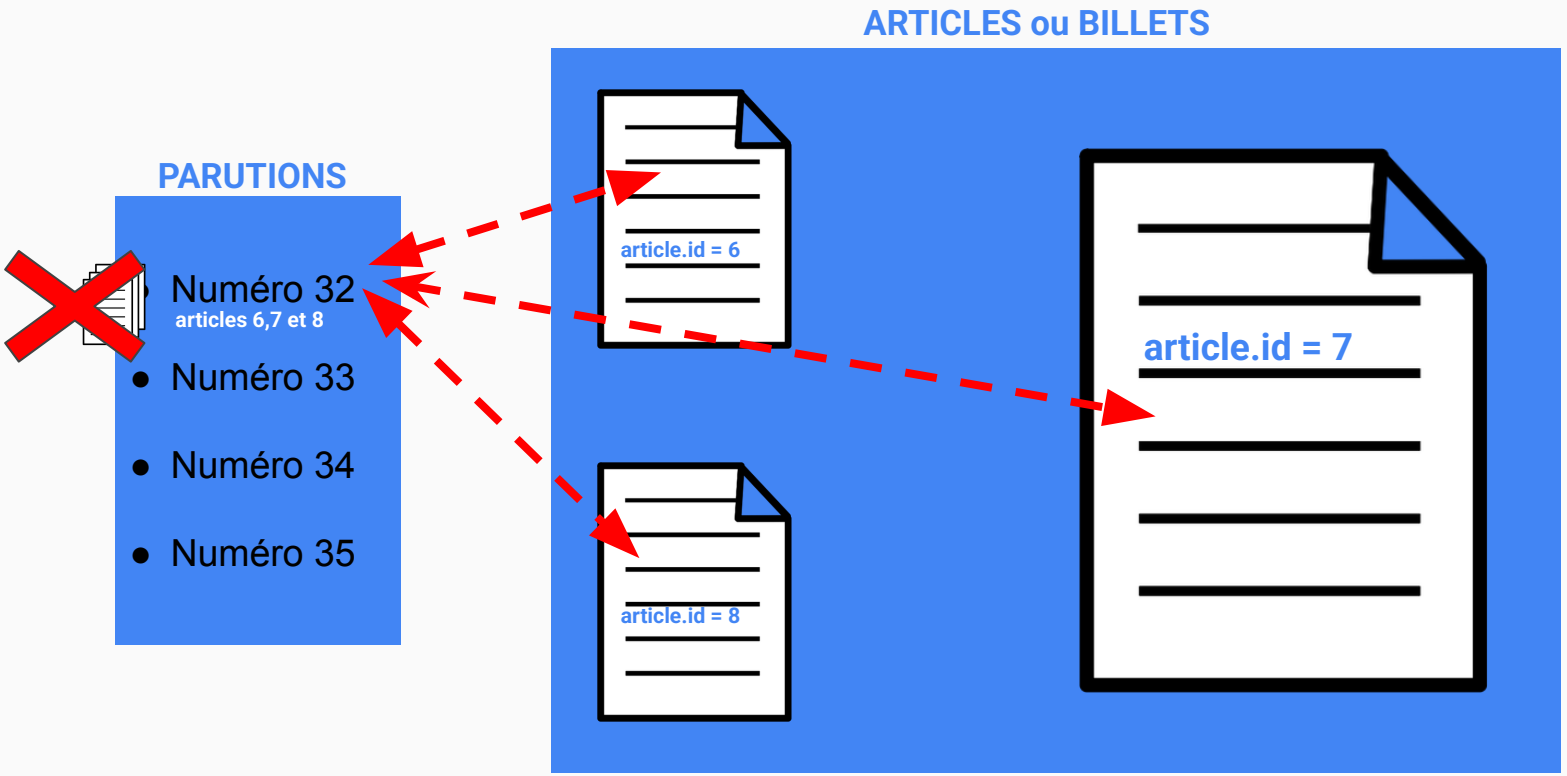




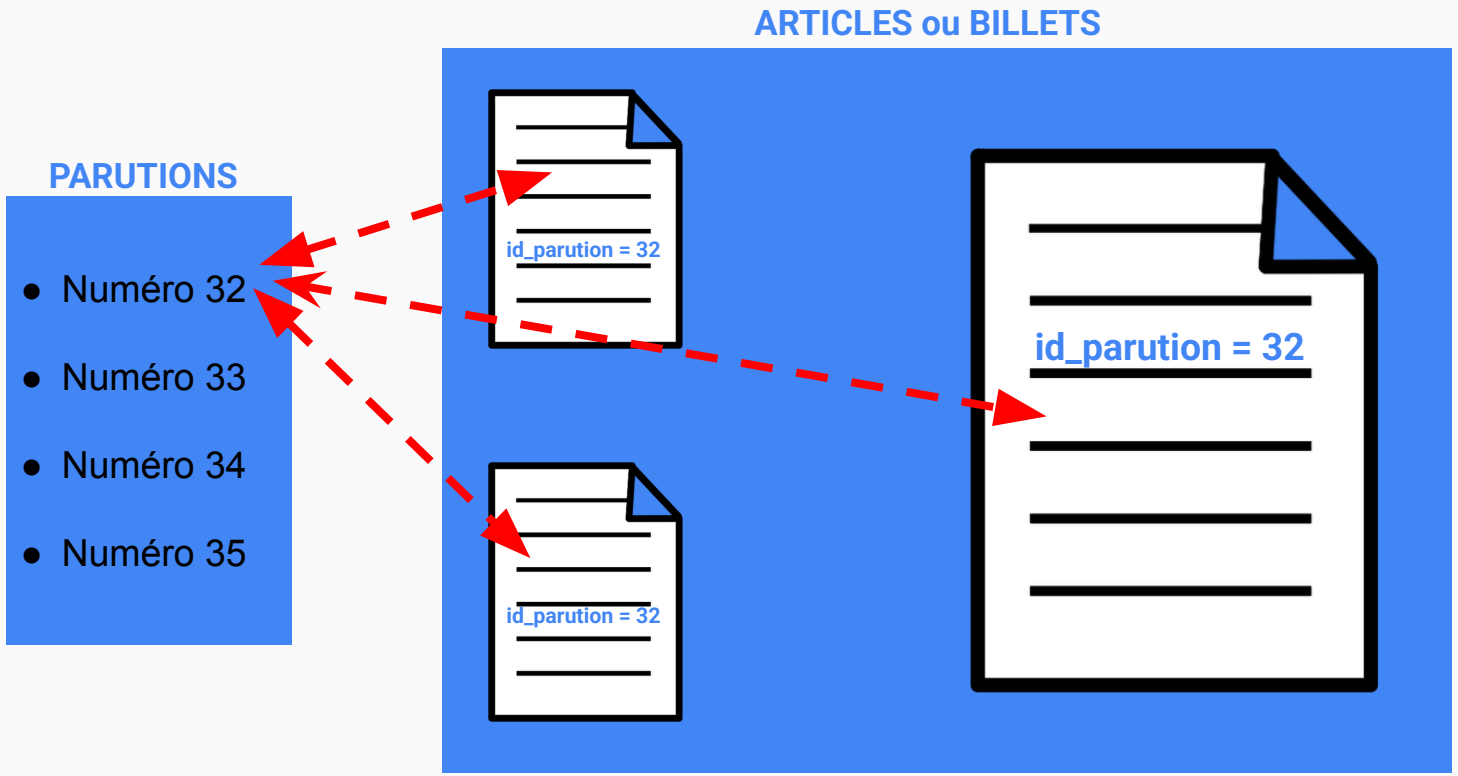
Exemple de parutions de revue ou magazines qui ont chacun plusieurs articles



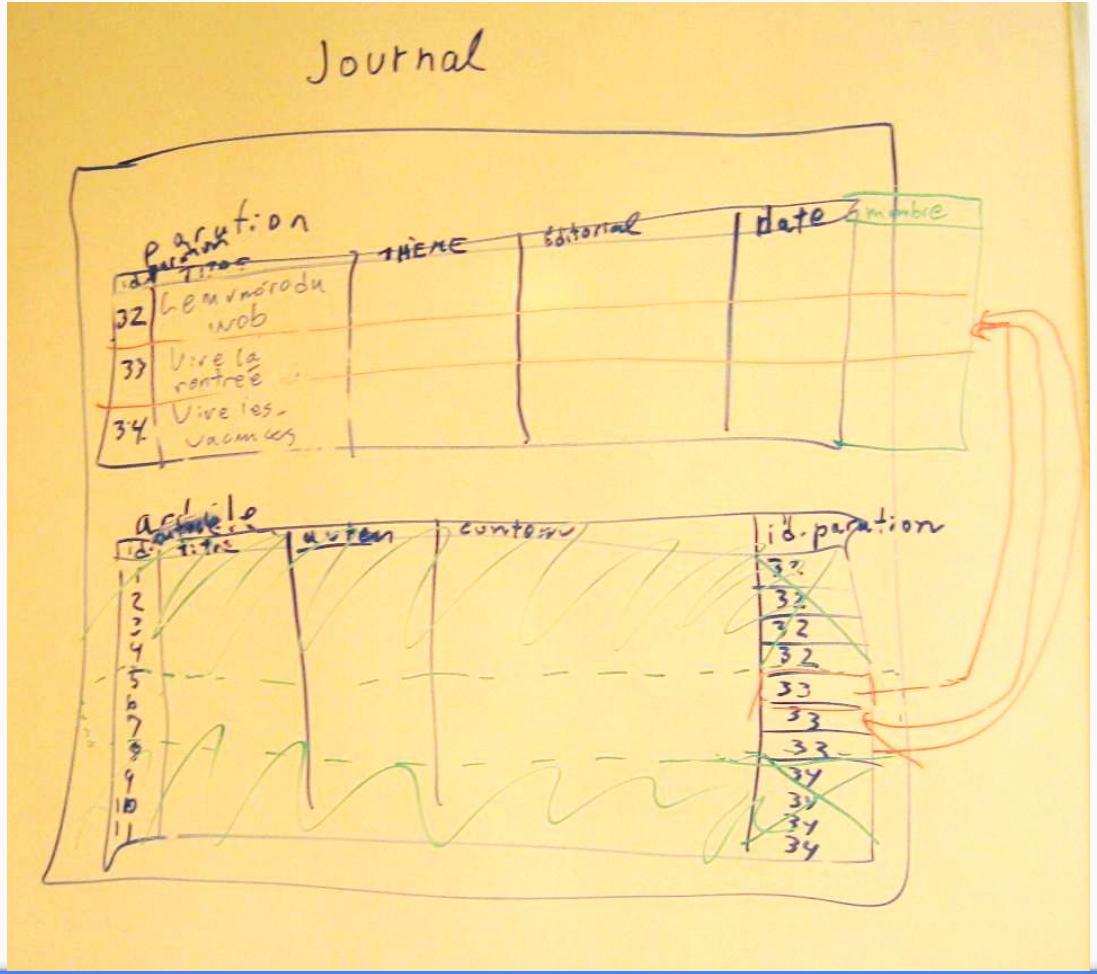
Dans un même numéro, il y a plusieurs articles ou billets



Pour connecter articles à leur parution, on NE va PAS lister les articles dans la parution



Pour connecter les articles avec leur parution, c'est du coté de l'article qu'on écrit l'id



Dans un même numéro, il y a plusieurs articles ou billets

Clé étrangère

- C'est quand on dit que
  - le champs id_parution de la table article
    - doit correspondre
  - à un id qui existe vraiment dans la table parution

⇛ *Autrement dit, si la parution 60 n'existe pas encore, je ne peux pas créer d'article qui aura comme valeur id_parution = 60.*

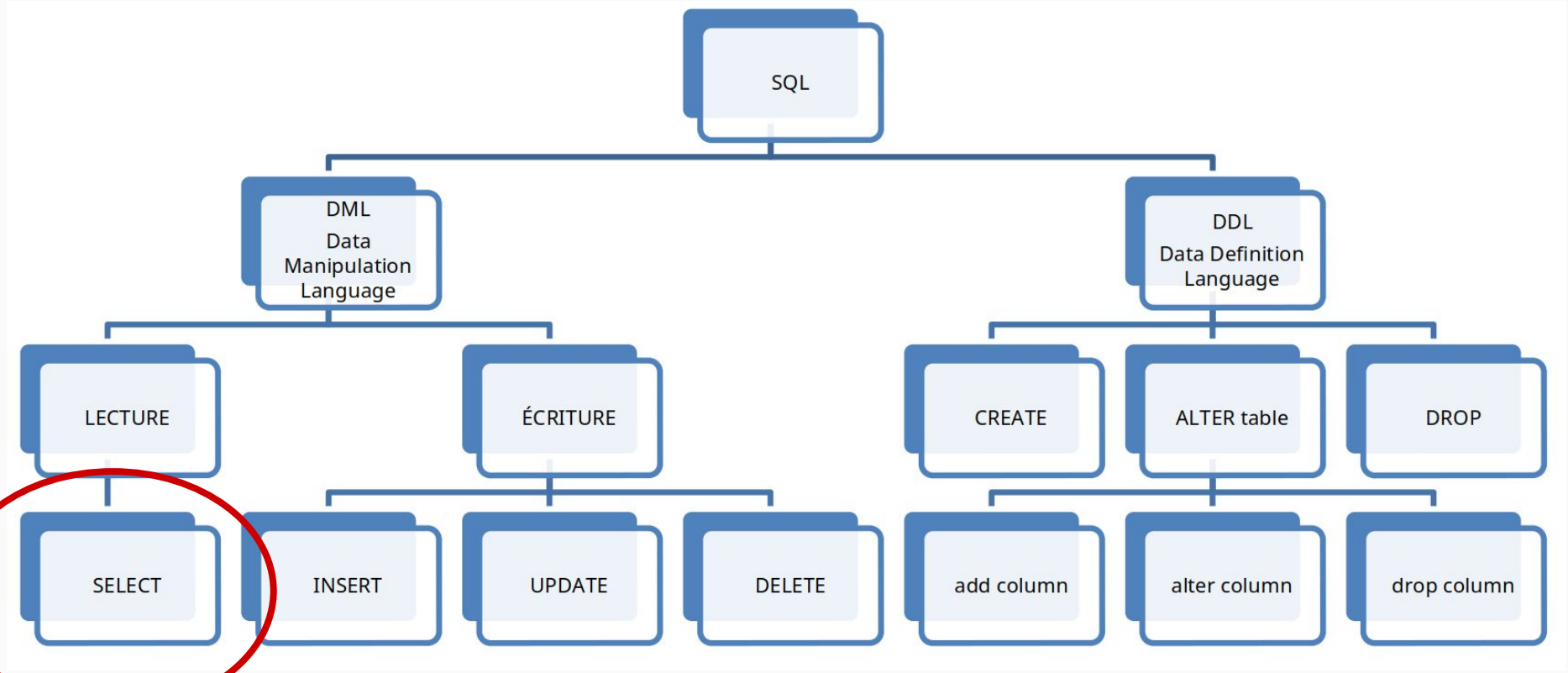
⇛ *Je ne peux pas non plus inscrire un 0 dans le champs.*



JOINTURE

...

La JOINTURE s'utilise pour lire des données avec un SELECT



Produit croisé

SELECT parution.*, article.*

FROM parution, article

(pas bon)

Produit croisé

```
SELECT parution.*, article.*

FROM parution, article

WHERE article.id_parution = 2
```

Produit croisé

```
SELECT parution.*, article.*

FROM parution, article

WHERE article.id_parution = 2 AND parution.id = 2
```

Jointure

```
SELECT article.*, parution.*

FROM parution

JOIN article ON article.id_parution = parution.id
```

Jointure

```
SELECT article.*, parution.*

FROM parution

JOIN article ON article.id_parution = parution.id

WHERE parution.id = 2
```

Jointure

```
SELECT article.titre AS titre_article, auteur AS auteur_article, parution.slogan
AS slogan_parution, parution.mois, parution.annee

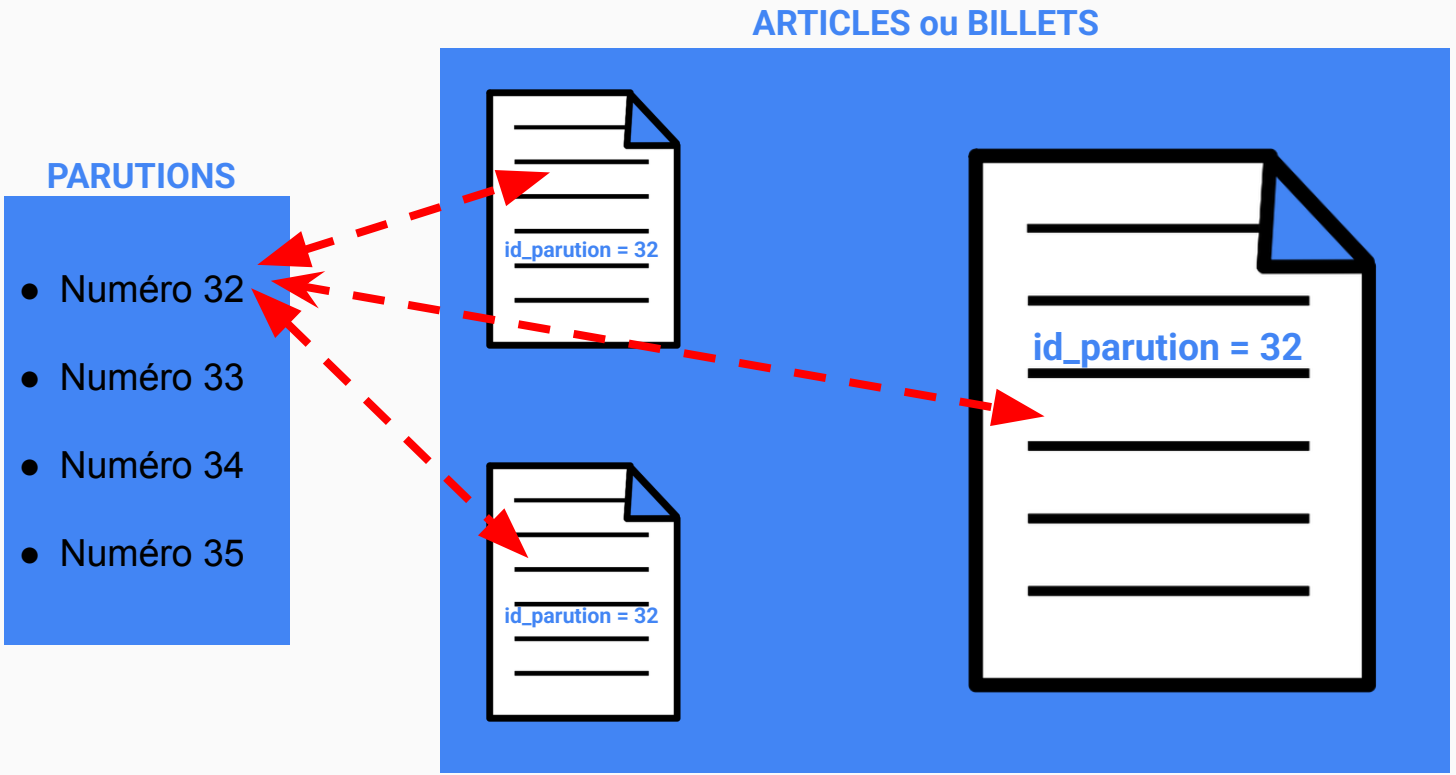
FROM parution

JOIN article ON article.id_parution = parution.id

WHERE parution.id = 2
```

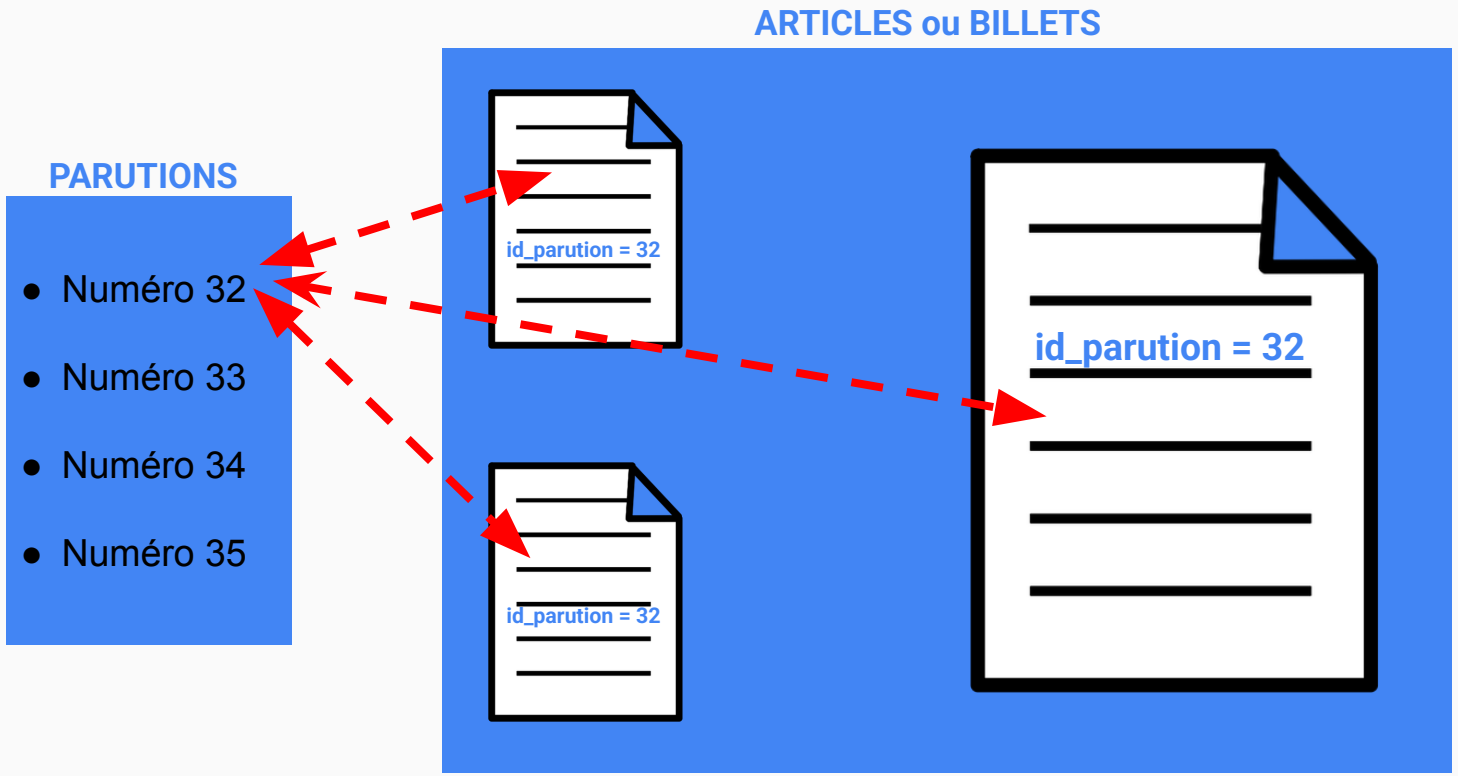
Table cours pour exercice

```
CREATE TABLE IF NOT EXISTS public.cours
(
    id serial not null,
    nom text COLLATE pg_catalog."default",
    description text COLLATE pg_catalog."default",
    ponderation text COLLATE pg_catalog."default",
    session integer,
    programme text COLLATE pg_catalog."default",
    CONSTRAINT cours_pkey PRIMARY KEY (id)
)
```




```
SELECT article.*, parution.*
FROM article
LEFT JOIN parution ON parution.id = article.id_parution
```

Pour lister les articles, avec pour chaque article le nom du numéro auquel il appartient




```
SELECT article.*, parution.titre as parution_titre
FROM article
LEFT JOIN parution ON parution.id = article.id_parution
WHERE article.id_parution = 32
```

Pour lister les articles d'un seul numéro, avec les détails du numéro

Autre exemple de jointure

SANS la JOINTURE

```
SELECT ville.nom as nom_ville, ville.fondation, province.nom as province_nom
FROM ville, province
WHERE ville.id_province = province.id
```

AVEC la JOINTURE

```
SELECT ville.nom as nom_ville, ville.fondation, province.nom as province_nom
FROM ville
LEFT JOIN province ON ville.id_province = province.id
```



JOINTURES et ENSEMBLES

...

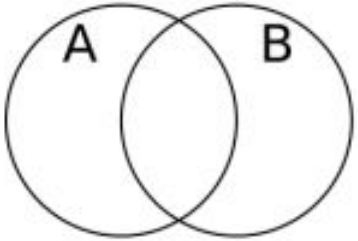
Types de jointures

- LEFT JOIN
- RIGHT JOIN

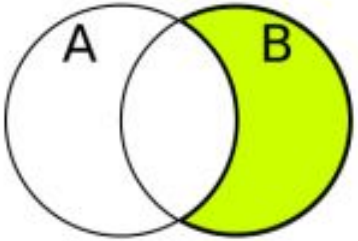
SQL JOINS

Arranged in a Karnaugh Map by Jason Charney (jrcharney@gmail.com)

No join (∅)

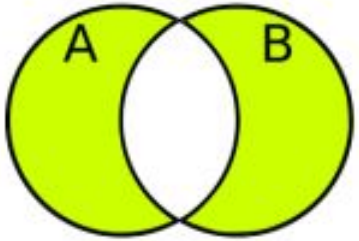


[Exclusive] Right Join (¬A)



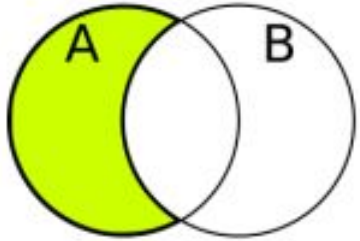

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

[Exclusive] Full Join (A⊕B)



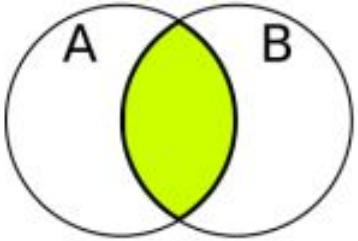

```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
WHERE B.key IS NULL
OR A.key IS NULL
```

[Exclusive] Left Join (¬B)



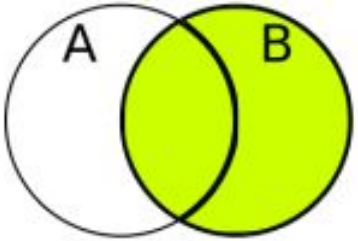

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

Inner Join (A∧B)



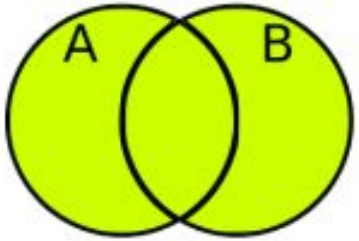

```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

[Inclusive] Right Join (B)



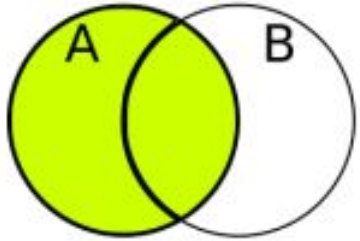

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
```

[Inclusive] Full Join (A∨B)




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
```

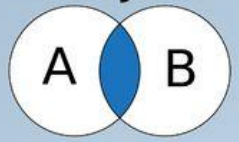
[Inclusive] Left Join (A)




```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
```

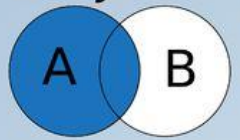
SQL JOINS

INNER JOIN



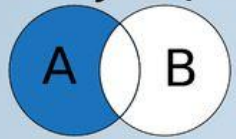

```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

LEFT JOIN



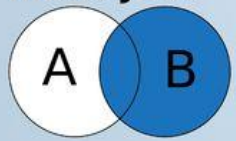

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
```

LEFT JOIN (sans l'intersection de B)



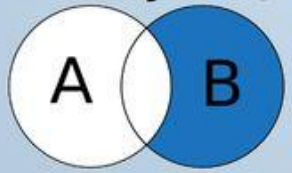

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

RIGHT JOIN



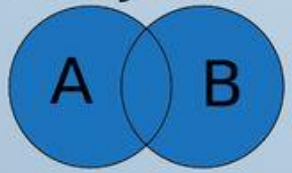

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
```

RIGHT JOIN (sans l'intersection de A)



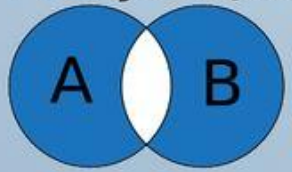

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

FULL JOIN




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
```

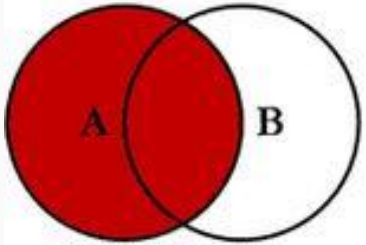
FULL JOIN (sans intersection)



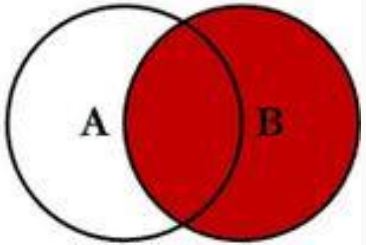

```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

sql.sh

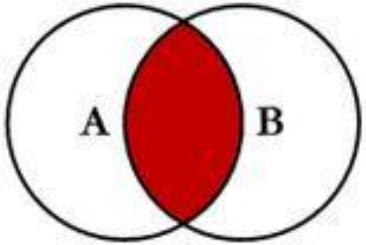
SQL JOINS



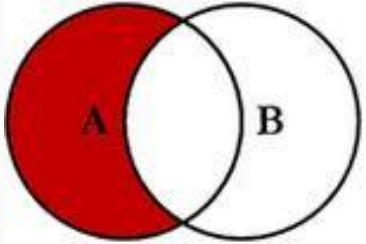

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



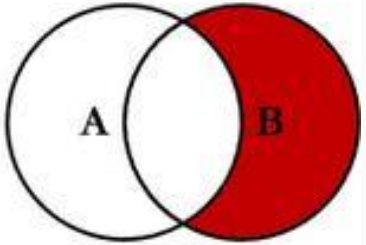

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



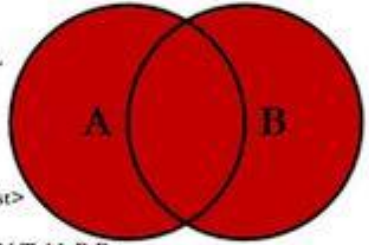

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



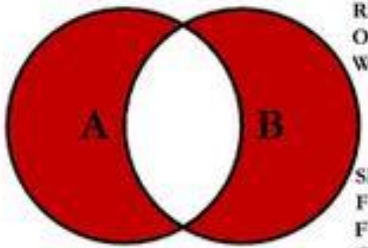

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

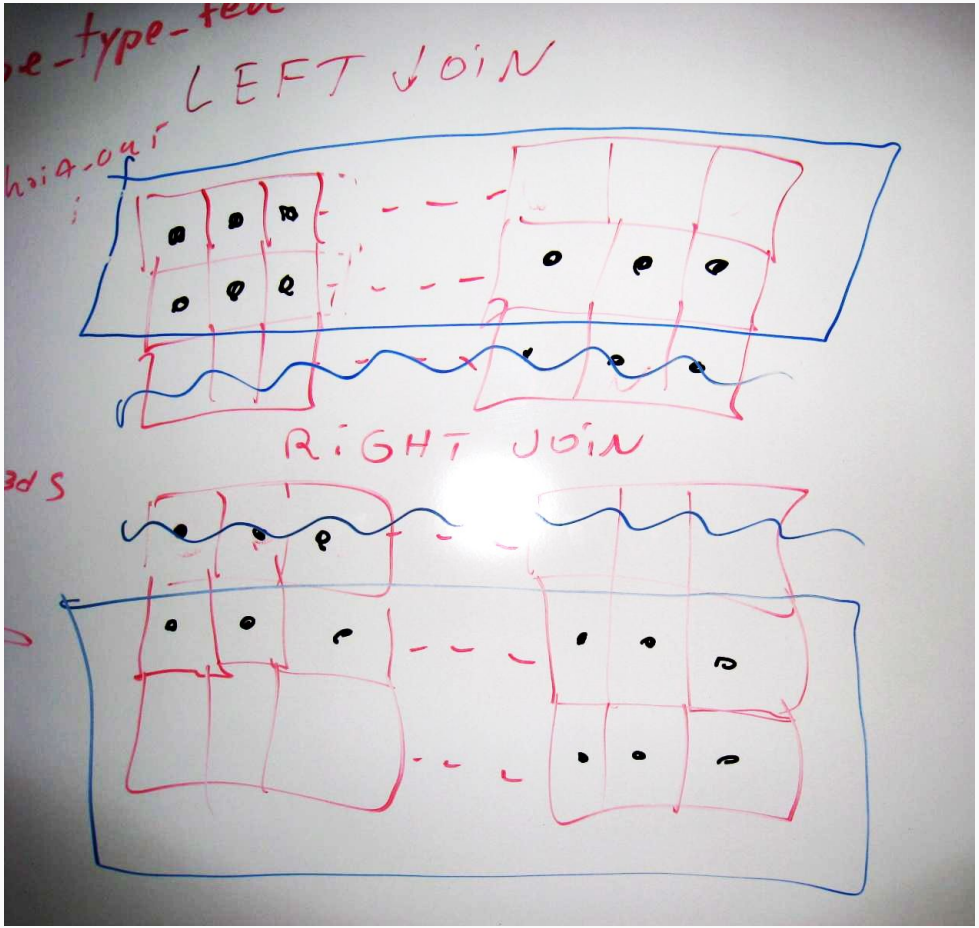



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008


LEFT JOIN
RIGHT JOIN