

# Bases de Données Relationnelles

Langage d'interrogation de Données (LID)

L2

Rafael Angarita  
Maitre de Conférences  
[rangarit@parisnanterre.fr](mailto:rangarit@parisnanterre.fr)

# SELECT

- Permet d'interroger une BDD en :
  - sélectionnant certaines colonnes d'une table : **projection**
  - sélectionnant certaines lignes d'une table en fonction de leur contenu : **sélection**
  - combinant des informations venant de plusieurs tables : **jointure, union, intersection, différence et division**
  - combinant entre elles ces différentes opérations
- Une **requête** est une combinaison d'opérations portant sur des tables et dont le résultat est lui-même une table dont l'existence est éphémère (le temps de la requête)

# SELECT : La projection

- Syntaxe :
  - **SELECT** colonne1 **FROM** tableA → Récupère une colonne en particulier
  - **SELECT \* FROM** tableA ou **SELECT** tableA.\* **FROM** tableA → Récupère toutes les colonnes
  - **SELECT DISTINCT** colonne1 **FROM** tableA → Récupère une colonne en particulier sans doublons
    - **ALL** (par défaut) → toutes les lignes sont présentes dans le résultat
  - **SELECT** colonne1 **AS** C1, colonne2 **AS** C2 **FROM** tableA **AS** TA → Alias
  - **SELECT concat**(colonne1, ' ',colonne\_2) **AS** colonne **FROM** T\_CLIENT → Concatène les colonnes

# SELECT : Exemples de projection

Flights:

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

Aircrafts:

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

- **SELECT ALL** \* FROM Aircrafts

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

- **SELECT** Company FROM Aircrafts

Company
CSA
KLM
KLM

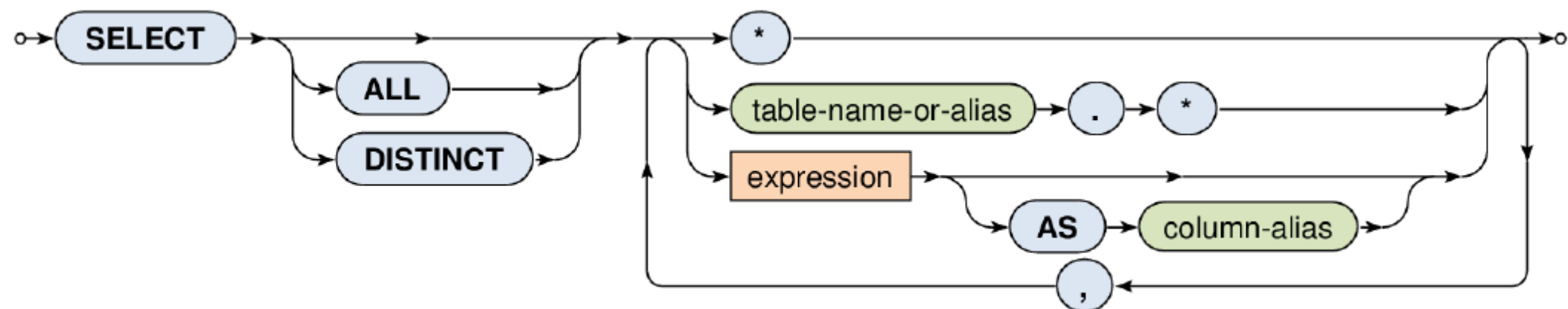
- **SELECT DISTINCT** Company FROM Aircrafts

Company
CSA
KLM

- **SELECT DISTINCT** Company **AS** Carrier FROM Aircrafts

Carrier
CSA
KLM

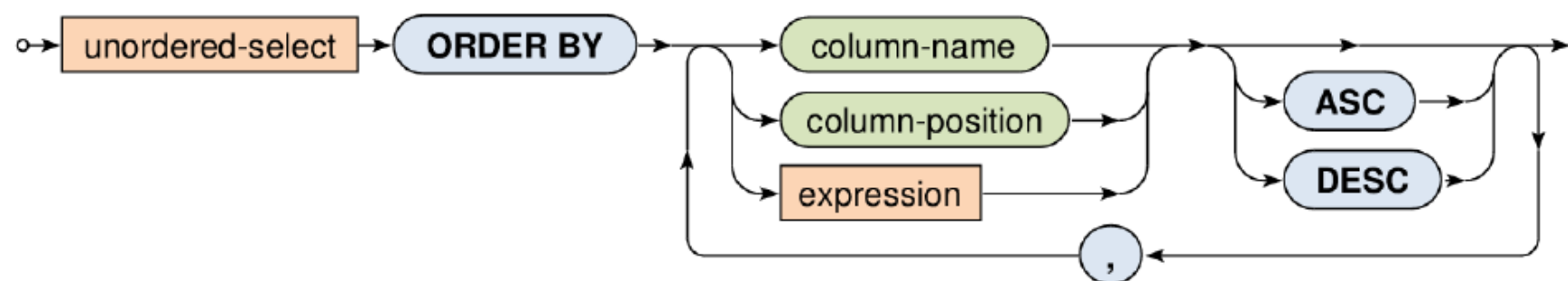
# SELECT : La projection (exemple)



# SELECT : ORDER BY

- Permet de trier des colonnes
  - Soit en précisant le nom littéral de la colonne
  - Soit en précisant son n° d'ordre dans l'énumération qui suit le mot clef SELECT
- ASC (par défaut) ou DESC
- **SELECT** colonne1, colonne2  
**FROM** tableA  
**ORDER BY** colonne2

**SELECT** colonne1,  
colonne2 **FROM** tableA  
**ORDER BY** 2



# SELECT : ORDER BY (exemple)

Flights:

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

Aircrafts:

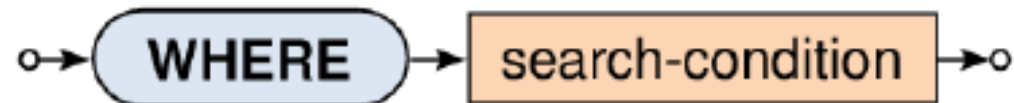
Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

- Retourner une liste ordonnée de toutes les destinations programmées

```
SELECT DISTINCT Destination  
FROM Flights  
ORDER BY Destination ASC
```

Destination
Amsterdam
London
Milano
New York
Rotterdam
Stuttgart
Toronto

# SELECT : La sélection WHERE



- Représente la condition de sélection que doit satisfaire une ligne afin d'apparaître dans un résultat
- Utilisation d'expressions qui peuvent être combinées grâce à **AND, OR et NOT**
- Exemples :
  - ... **WHERE** (Capacity > 200) **AND** (Aircraft **LIKE** 'Airbus%') ...
  - ... **WHERE** (Company **IN** ('KLM', 'Emirates')) ...
  - ... **WHERE NOT** (Passengers **BETWEEN** 100 AND 200) ...



# SELECT : WHERE (examples)

Flights:

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

Aircrafts:

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

1. Retourner tous les vols de KLM

2. Retourner tous les vols de KLM avec plus de 100 passagers

```
1. SELECT *  
   FROM Flights  
   WHERE Company='KLM'
```

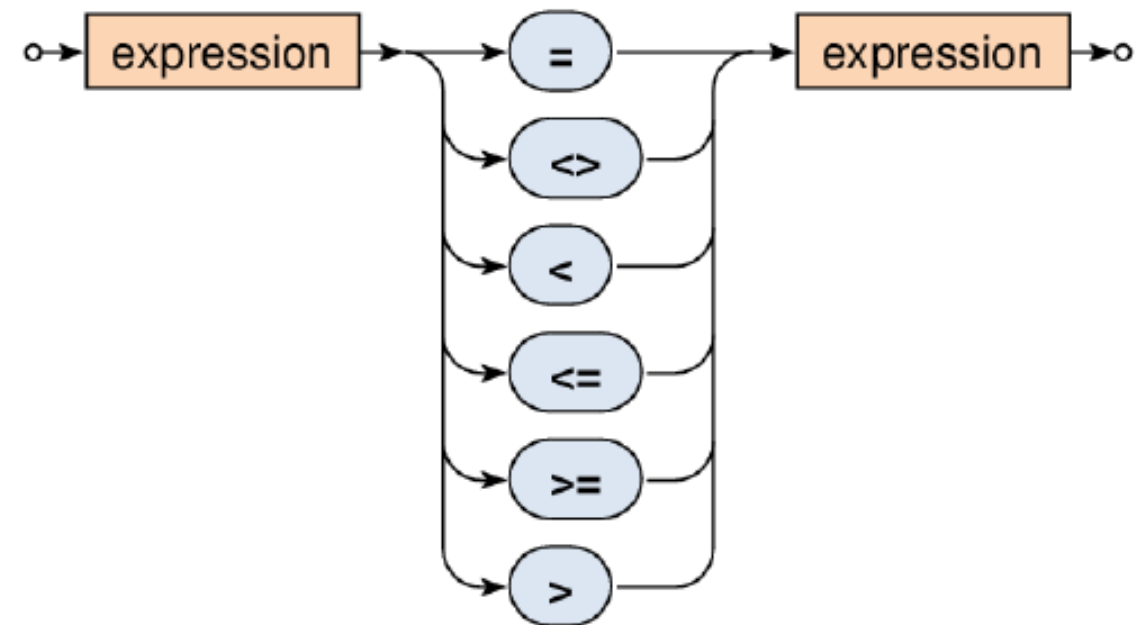
Flight	Company	Destination	Passengers
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

```
2. SELECT Destination, Passengers  
   FROM Flights  
   WHERE Company='KLM' AND Passengers>100
```

Destination	Passengers
Amsterdam	130

# SELECT : WHERE et Opérateurs (1)

- Opérateurs de **comparaison**

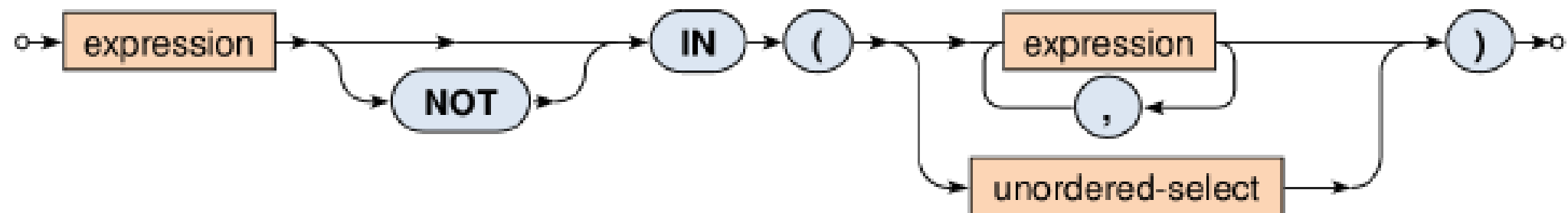


- Opérateur d'intervalle **BETWEEN**
  - value BETWEEN Min AND Max** équivalent à **(Min <= value) AND (value <= Max)**
  - Ex : ...WHERE** Passengers **BETWEEN** 100 **AND** 200



# SELECT : WHERE et Opérateurs (2)

- Opérateur **IN** : vérifie qu'une valeur existe (ou pas) dans un ensemble de données
  - Ex : Company **IN** ('KLM', 'Emirates')
  - Ex : Company **NOT IN** ('KLM', 'Emirates')



# SELECT : WHERE et Opérateurs (3)

- Opérateur **LIKE**
  - Permet de faire une comparaison partielle
  - Surtout employé avec les colonnes contenant des données de type alpha
  - utilise les jokers % et \_
    - % remplace n'importe quelle chaîne de caractères, y compris la chaîne vide
    - \_ remplace un et un seul caractère
- Exemples :
  - LIKE 'B%' : valeur qui commence par B
  - LIKE '%B' : valeur qui se termine par B
  - LIKE '\_B%' : valeur qui contient le 'B' en seconde position
  - LIKE 'B\_\_%' : valeur qui commence par B et qui a au moins 3 caractères de long
  - LIKE 'A%B' : valeur qui commence par A et se termine par B

# FROM

- Définit les tables sur lesquelles faire la requête
- Deux façons de faire :

- **Notation WHERE**

- Liste des tables séparées par une virgule
- Le **produit cartésien** de leurs lignes est supposé
- La condition de jointure est spécifiée dans la clause WHERE

```
SELECT ...  
FROM Table1, Table2  
WHERE Table1.xxx = Table2.yyy  
AND condition
```

- **Notation avec différents opérateurs JOIN**

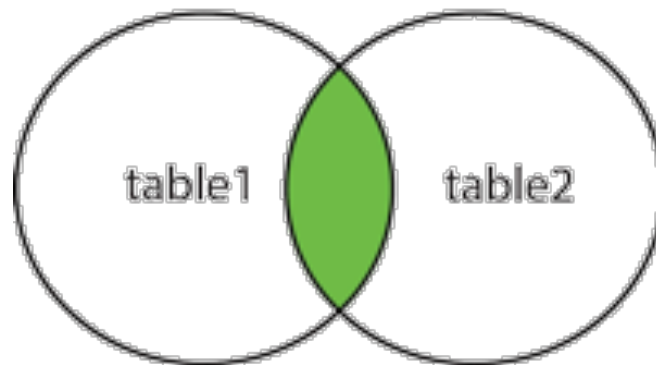
```
SELECT ...  
FROM Table1 JOIN Table2  
WHERE condition
```

# FROM : Jointures

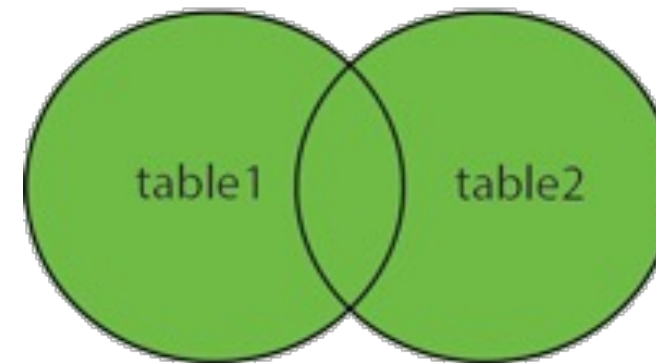
- **NATURAL JOIN**
- **INNER JOIN**
- **CROSS JOIN**
- **LEFT JOIN (ou LEFT OUTER JOIN)**
- **RIGHT JOIN (ou RIGHT OUTER JOIN)**
- **FULL JOIN (ou FULL OUTER JOIN)**
- **UNION JOIN**

# FROM : Jointures

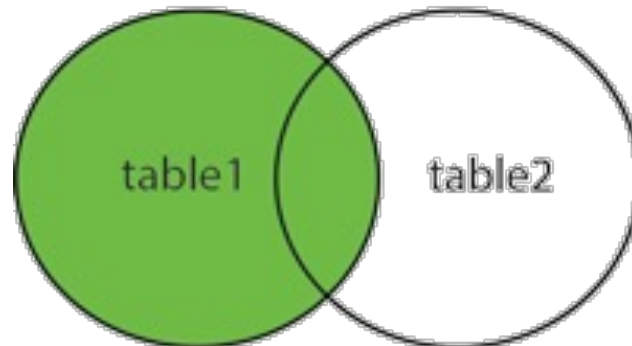
INNER JOIN



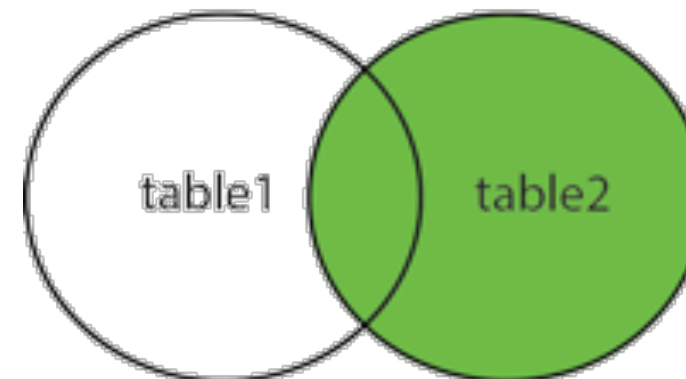
FULL OUTER JOIN



LEFT JOIN



RIGHT JOIN



Source : W3SCHOOLS

# FROM: (Cross) join



- Jointure croisée permettant de faire le produit cartésien de 2 tables. i.e., joindre chaque ligne d'une table avec chaque ligne d'une seconde table. Attention, le nombre de résultats est en général très élevé.

- `SELECT * FROM T1 CROSS JOIN T2`
- `SELECT * FROM T1 JOIN T2`

A	T1.*		A	T2.*		T1.A	T1.*	T2.A	T2.*
1	...		1	...		1	...	1	...
2	...		4	...		1	...	4	...
3	...					2	...	1	...
						2	...	4	...
						3	...	1	...
						3	...	4	...

## Notation Where :

```
SELECT *  
FROM T1, T2
```



# FROM : Natural join



- Jointure naturelle entre 2 tables s'il y a au moins **une colonne qui porte le même nom** entre les 2 tables SQL

- I.e., colonnes du même nom

- `SELECT * FROM T1 NATURAL JOIN T2`

A	T1.*		A	T2.*		A	T1.*	T2.*
1	...		1	...	➡	1	...	...
2	...		4	...				
3	...							

## Notation Where :

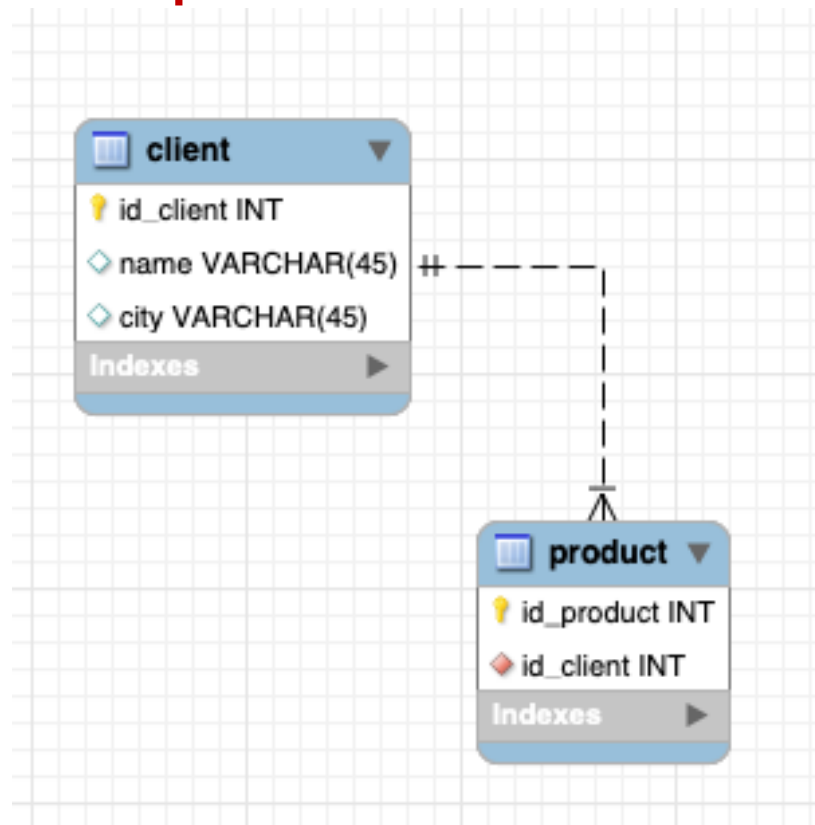
```
SELECT *  
FROM T1, T2  
WHERE T1.A = T2.A
```

# FROM : Natural join, est-ce une bonne pratique ?

La syntaxe est anti-pattern :

- Le but de la requête est moins évident
- les colonnes utilisées par l'application ne sont pas claires les colonnes utilisées peuvent changer "de façon inattendue »
- La syntaxe va à l'encontre de la règle de modularité, qui consiste à utiliser un typage strict chaque fois que possible. Explicit est meilleur.

# FROM : Natural join, est-ce une bonne pratique ?



```
INSERT INTO client VALUES(1,'Un client', 'Tokyo');
```

```
INSERT INTO product VALUES(1,1);
```

```
INSERT INTO product VALUES(2,1);
```

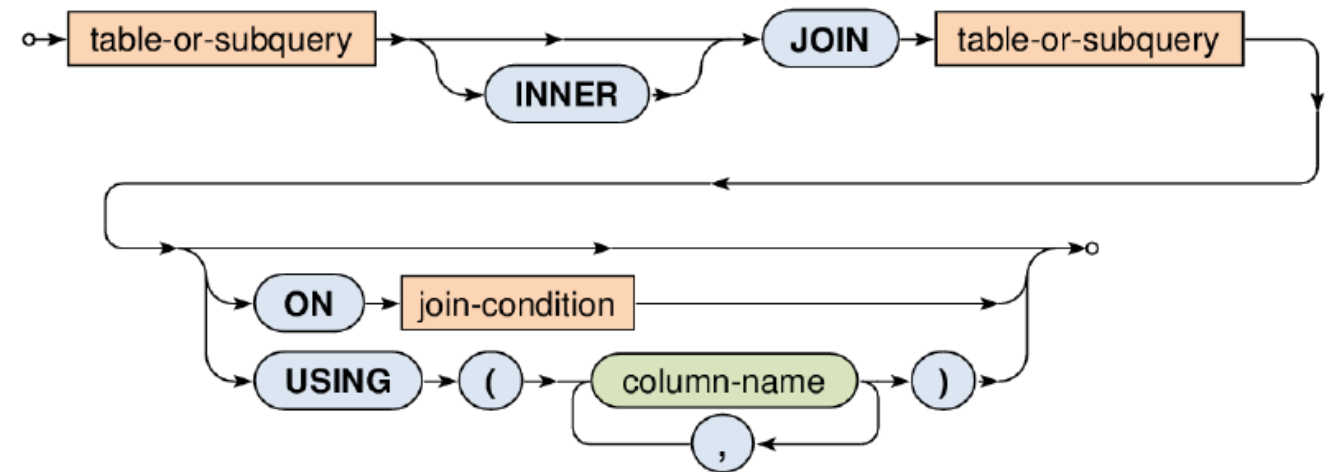
```
SELECT * FROM client NATURAL JOIN product;
```

On souhaite ajouter le nom aux produits :

```
ALTER TABLE product ADD COLUMN name VARCHAR(20) ;
```

Notre requête NATURAL JOIN ne marche plus !

# FROM: Inner join

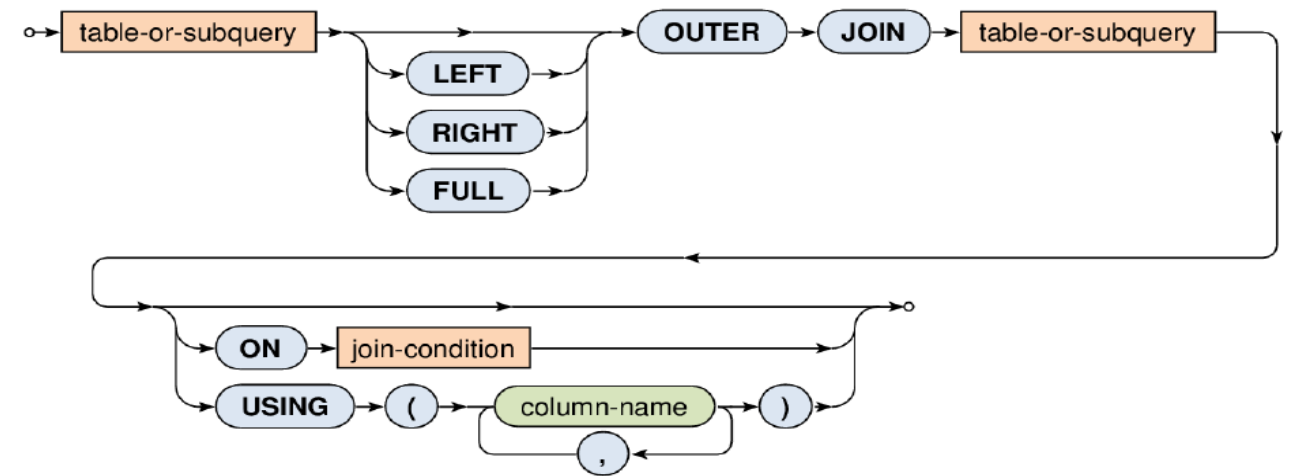


- Jointure interne pour retourner les lignes quand la condition qui suit le mot clé **ON** est vraie.
  - C'est l'une des jointures les plus communes.
  - SELECT \* FROM T1 **INNER JOIN** T2 **ON** (T1.A <= T2.A)
  - SELECT \* FROM T1 **JOIN** T2 **ON** (T1.A <= T2.A)

A	T1.*		A	T2.*		T1.A	T1.*	T2.A	T2.*
1	...		1	...	→	1	...	1	...
2	...		4	...		1	...	4	...
3	...					2	...	4	...
						3	...	4	...

**Where notation:**  
 SELECT \*  
 FROM T1, T2  
 WHERE T1.A <= T2.A

# FROM: Outer join



- Jointure externe pour retourner tous les lignes de la table de gauche (**LEFT OUTER JOIN**) ou droite (**RIGHT OUTER JOIN**) même si la condition n'est pas vraie dans l'autre table
  - **FULL** (par défaut): quand la condition est vraie dans au moins une des 2 tables
  - `SELECT * FROM T1 LEFT OUTER JOIN T2 ON (T1.A = T2.A)`

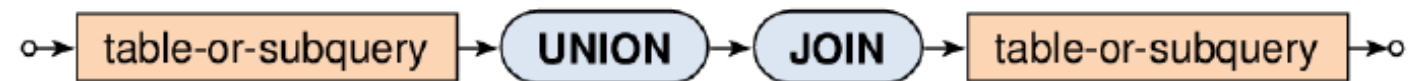
A	T1.*
1	...
2	...
3	...

A	T2.*
1	...
4	...



T1.A	T1.*	T2.A	T2.*
1	...	1	...
2	...	NULL	NULL
3	...	NULL	NULL

# FROM: Union join



- Les lignes des deux tables sont intégrées au sein d'une même tables, pas de combinaison de lignes
  - `SELECT * FROM T1 UNION JOIN T2`

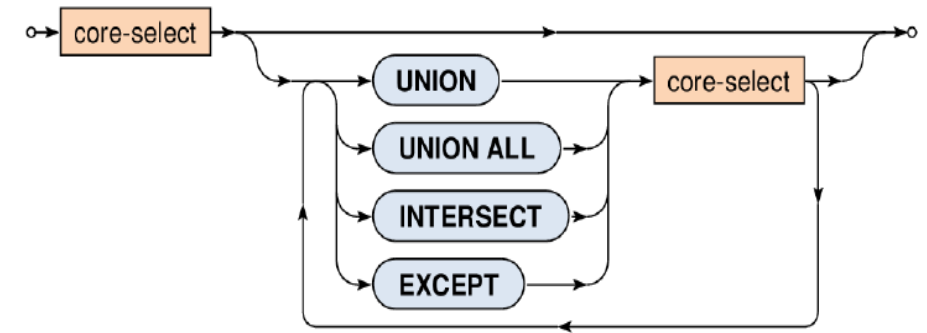
The diagram illustrates the result of a UNION JOIN operation. On the left, two input tables are shown. The first table has columns 'A' and 'T1.\*' with rows (1, ...), (2, ...), and (3, ...). The second table has columns 'A' and 'T2.\*' with rows (1, ...) and (4, ...). An arrow points to the resulting table on the right, which has four columns: 'T1.A', 'T1.\*', 'T2.A', and 'T2.\*'. The first three rows of the result table correspond to the rows of the first input table, with 'T2.A' and 'T2.\*' containing NULL values. The next two rows correspond to the rows of the second input table, with 'T1.A' and 'T1.\*' containing NULL values.

A	T1.*
1	...
2	...
3	...

A	T2.*
1	...
4	...

T1.A	T1.*	T2.A	T2.*
1	...	NULL	NULL
2	...	NULL	NULL
3	...	NULL	NULL
NULL	NULL	1	...
NULL	NULL	4	...

# Opérateurs ensemblistes



- **UNION** : Union de deux tables sans duplication
- **UNION ALL** : Union de deux tables avec duplication
- **INTERSECT** : Intersection de deux tables
- **EXCEPT** : Différence entre deux tables
- **Les deux opérandes doivent être compatibles**
  - Même nombre de colonnes
  - Ces colonnes doivent être du même type

# Opérateurs ensemblistes : Exemples

Flights:

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

Aircrafts:

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

Requête :

```
SELECT Company
FROM Flights
UNION
SELECT Company
FROM Aircrafts
```

Company
CSA
Lufthansa
Air Canada
KLM

Requête :

```
SELECT Company FROM
Flights
INTERSECT
SELECT Company FROM
Aircrafts
```

Company
CSA
KLM

Requête :

```
SELECT Company FROM
Flights
EXCEPT
SELECT Company FROM
Aircrafts
```

Company
Lufthansa
Air Canada



