

Une *base de données* regroupe un ensemble de *tables*, lesquelles peuvent être représentées sous la forme de tableaux bi-dimensionnels.

TABLE Avions :

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

Une base de données comprend généralement plus d'une table, et les systèmes de gestions de base de données permettent de *croiser* les données de plusieurs tables.

TABLE Compagnies :

Code	Compagnie	Modèle	Nombre
AF	Air France	777-200	25
AF	Air France	A321-200	15
AF	Air France	A330-200	15
AF	Air France	A340-300	12
AC	Air Canada	777-200	6
AC	Air Canada	A321-200	10
AC	Air Canada	CRJ-100	51

## Attributs et domaines

Chaque colonne d'une table correspond à un *attribut*.

L'ensemble des valeurs possibles pour un attribut est appelé *domaine* de l'attribut (chaîne de caractères, valeur numérique, booléen, etc)

TABLE Avions :

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Attributs manquants

Certaines « cases » d'une table peuvent être vides.

(En pratique, elles contiennent une valeur spéciale, NULL, ne faisant pas partie du domaine de l'attribut en question).

TABLE Avions :

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2		1574

Les lignes de la tables sont généralement appelées *enregistrements*.

TABLE Avions :

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Introduction

Les bases de données

## Objectifs

Effectuer des requêtes en langage SQL

Création et modification de tables

Algèbre relationnelle

Architecture des réseaux

## Objectifs d'une base de données

Les bases de données permettent de stocker une grande quantité de données de différentes natures, et de les mettre en relation afin de répondre à des questions variées et complexes :

Quel est le bi-moteur encore en service avec la plus grande capacité ?

Combien de modèles d'Airbus contient la flotte d'Air France ?

Quels sont les fabricants d'avions qui proposent un modèle permettant d'effectuer Paris - Pékin sans escale ?

Quelle est la dix-septième compagnie dont le nom ne commence pas par « A » ayant le plus d'avions pouvant transporter entre quatre-vingt-sept et cent onze passagers ?

Et globalement, toutes ces questions qui vous empêchent de dormir la nuit.

## Introduction

### Effectuer des requêtes en langage SQL

Principe

Projection, sélection

Organisation des résultats

Aggrégation

Sous-requêtes

Jointures

Opérations ensemblistes

### Création et modification de tables

### Algèbre relationnelle

Introduction

Effectuer des requêtes en langage SQL

**Principe**

Projection, sélection

Organisation des résultats

Aggrégation

Sous-requêtes

Jointures

Opérations ensemblistes

Création et modification de tables

Algèbre relationnelle

## Principe des requêtes

Le langage SQL permet de transcrire ces questions sous la forme de *requêtes* qui pourront être comprises par la base de donnée.

Le résultat d'une requête **doit impérativement avoir la forme d'une table** (qui peut-être réduite à une seule case).

Quels sont les fabricants d'avions ont fourni des appareils biréacteurs actuellement en service dans la flotte d'Air France ?

```
SELECT DISTINCT A.Fabricant  
FROM Avions AS A JOIN Compagnies AS C  
ON A.Modèle = C.Modèle  
WHERE A.Moteurs = 2 AND C.Compagnie = "Air France"
```

Introduction

Effectuer des requêtes en langage SQL

Principe

**Projection, sélection**

Organisation des résultats

Aggrégation

Sous-requêtes

Jointures

Opérations ensemblistes

Création et modification de tables

Algèbre relationnelle

On peut sélectionner certains attributs en effectuant une *projection* grâce au mot-clé SELECT :

SELECT Modèle, Moteurs FROM Avions

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Suppression des doublons

Le mot-clé DISTINCT, placé derrière le mot-clé SELECT, permet d'éliminer les doublons dans le résultat :

SELECT DISTINCT Fabricant, Moteurs FROM Avions

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

Il est possible de créer temporairement des attributs supplémentaires en utilisant des expressions :

SELECT Modèle, Places/Moteurs FROM Avions

Modèle	Fabricant	Moteurs	Places	Autonomie	Pl./Mot.
737-900	Boeing	2	215	5926	107.5
777-200	Boeing	2	440	14307	220
A321-200	Airbus	2	220	5926	110
A330-200	Airbus	2	380	13890	190
A340-300	Airbus	4	440	13705	110
CRJ-100	Bombardier	2	50	1574	25

## Sélection sur critère booléen

On peut également effectuer des *sélections*, c'est-à-dire ne conserver que les enregistrements vérifiant une expression booléenne grâce au mot-clé WHERE :

```
SELECT Modèle, Fabricant, Moteurs, Places, Autonomie FROM Avions  
WHERE Places > 250
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Sélection sur critère booléen

Pour obtenir la totalité des attributs, on peut utiliser « \* » :

```
SELECT * FROM Avions
```

```
WHERE Places > 250
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Expressions booléennes utiles

Pour savoir si un attribut figure parmi un ensemble de valeurs, on dispose du mot-clé IN :

```
SELECT * FROM table WHERE nom IN (val1, val2, val3)
```

Pour identifier les « cases vides » (attribut non présent dans un enregistrement), on utilisera IS NULL (ou IS NOT NULL) :

```
SELECT * FROM table WHERE nom IS NOT NULL
```

Pour faire des recherches sur une partie d'une chaîne, on peut utiliser LIKE, le caractère % pouvant remplacer n'importe quelle suite de caractères :

```
SELECT * FROM table WHERE nom LIKE "% Dupont"
```

## Opérateurs booléens

Les opérateurs booléens OR, AND et NOT permettent de combiner les expressions booléennes :

```
SELECT * FROM Avions
```

```
WHERE Places > 250 AND Moteurs = 2
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

Il est bien entendu possible de combiner sélection et projection :

```
SELECT Modèle, Moteurs FROM Avions
```

```
WHERE Places > 250 AND Moteurs = 2
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

Introduction

Effectuer des requêtes en langage SQL

Principe

Projection, sélection

**Organisation des résultats**

Aggrégation

Sous-requêtes

Jointures

Opérations ensemblistes

Création et modification de tables

Algèbre relationnelle

## Tri des résultats

Pour ordonner les résultats, on peut utiliser le mot-clé ORDER BY :

(on peut utiliser une expression, et les attributs n'ont pas besoin d'être sélectionnés)

```
SELECT Modèle, Moteurs FROM Avions
```

```
ORDER BY Places
```

Modèle	Fabricant	Moteurs	Places	Autonomie
CRJ-100	Bombardier	2	50	1574
737-900	Boeing	2	215	5926
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
777-200	Boeing	2	440	14307
A340-300	Airbus	4	440	13705

Le mot-clé DESC permet d'obtenir les résultats dans l'ordre décroissant :

```
SELECT Modèle, Moteurs FROM Avions
```

```
ORDER BY Places DESC
```

Modèle	Fabricant	Moteurs	Places	Autonomie
777-200	Boeing	2	440	14307
A340-300	Airbus	4	440	13705
A330-200	Airbus	2	380	13890
A321-200	Airbus	2	220	5926
737-900	Boeing	2	215	5926
CRJ-100	Bombardier	2	50	1574

## Limitation du nombre de résultats

On peut ne conserver que les  $n$  premiers résultats avec le mot-clé LIMIT :

```
SELECT Modèle, Moteurs FROM Avions
```

```
ORDER BY Places
```

```
LIMIT 3
```

Modèle	Fabricant	Moteurs	Places	Autonomie
CRJ-100	Bombardier	2	50	1574
737-900	Boeing	2	215	5926
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
777-200	Boeing	2	440	14307
A340-300	Airbus	4	440	13705

On peut ignorer les  $n$  premiers résultats avec le mot-clé OFFSET :

```
SELECT Modèle, Moteurs FROM Avions
```

```
ORDER BY Places
```

```
LIMIT 3 OFFSET 2
```

Modèle	Fabricant	Moteurs	Places	Autonomie
CRJ-100	Bombardier	2	50	1574
737-900	Boeing	2	215	5926
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
777-200	Boeing	2	440	14307
A340-300	Airbus	4	440	13705

Introduction

Effectuer des requêtes en langage SQL

Principe

Projection, sélection

Organisation des résultats

**Aggrégation**

Sous-requêtes

Jointures

Opérations ensemblistes

Création et modification de tables

Algèbre relationnelle

La fonction COUNT permet d'effectuer des décomptes dans le résultat de la requête :

```
SELECT COUNT(*) FROM Avions
```

Modèle	Fabricant	Moteurs	Places	Autonomie	COUNT
737-900	Boeing	2	215	5926	6
777-200	Boeing	2	440	14307	
A321-200	Airbus	2	220	5926	
A330-200	Airbus	2	380	13890	
A340-300	Airbus	4	440	13705	
CRJ-100	Bombardier	2	50	1574	

Les « cases » vides (contenant NULL) ne sont pas comptées.

COUNT(Moteurs) compte les enregistrements pour lesquels l'attribut Moteur n'est pas vide.

COUNT(Moteurs, Places) compte les enregistrements pour lesquels les attributs Moteur et Places ne sont pas *tous les deux* vides.

COUNT(\*) compte tous les enregistrements.

## Compter les valeurs uniques

Il est possible de dénombrer les différentes valeurs prises par l'attribut, grâce à DISTINCT :

```
SELECT COUNT(DISTINCT Fabricant) FROM Avions
```

Modèle	Fabricant	Moteurs	Places	Autonomie	COUNT
737-900	Boeing	2	215	5926	3
777-200	Boeing	2	440	14307	
A321-200	Airbus	2	220	5926	
A330-200	Airbus	2	380	13890	
A340-300	Airbus	4	440	13705	
CRJ-100	Bombardier	2	50	1574	

Si l'on combine une sélection et un décompte, seuls sont comptés les enregistrements respectant la condition :

```
SELECT COUNT(*) FROM Avions  
WHERE Moteurs = 2
```

Modèle	Fabricant	Moteurs	Places	Autonomie	COUNT
737-900	Boeing	2	215	5926	5
777-200	Boeing	2	440	14307	
A321-200	Airbus	2	220	5926	
A330-200	Airbus	2	380	13890	
A340-300	Airbus	4	440	13705	
CRJ-100	Bombardier	2	50	1574	

Il est également possible de déterminer la plus grande valeur d'un attribut grâce au mot-clé MAX :

```
SELECT MAX(Places) FROM Avions  
WHERE Autonomie < 10000
```

Modèle	Fabricant	Moteurs	Places	Autonomie	MAX(Pl.)
737-900	Boeing	2	215	5926	220
777-200	Boeing	2	440	14307	
A321-200	Airbus	2	220	5926	
A330-200	Airbus	2	380	13890	
A340-300	Airbus	4	440	13705	
CRJ-100	Bombardier	2	50	1574	

Diverses fonctions sont disponibles :

- MAX : retourne le plus grand élément
- MIN : retourne le plus petit élément
- SUM : retourne la somme des éléments
- AVG : retourne la moyenne des éléments
- VARIANCE : retourne l'écart-type des éléments

...

La disponibilité de ces fonctions peut varier d'un système de gestion de bases de données à l'autre.

## Mélanger attributs et fonctions

Le résultat devant être une table, mélanger attributs et fonctions peut donner une erreur, ou un résultat partiellement « aléatoire ».

La requête suivante est admise avec SQLite, mais pas avec Oracle SQL :

SELECT Modèle, MAX(Places) FROM Avions

Modèle	Fabricant	Moteurs	Places	Autonomie	MAX(Pl.)
737-900	Boeing	2	215	5926	440
777-200	Boeing	2	440	14307	
A321-200	Airbus	2	220	5926	
A330-200	Airbus	2	380	13890	
A340-300	Airbus	4	440	13705	
CRJ-100	Bombardier	2	50	1574	

Il est possible de regrouper les valeurs identiques d'un attribut, grâce au mot-clé GROUP BY :

```
SELECT Fabricant FROM Avions
```

```
GROUP BY Fabricant
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200		2	440	14307
A321-200	Airbus	2	220	5926
A330-200		2	380	13890
A340-300		4	440	13705
CRJ-100	Bombardier	2	50	1574

Le résultat doit rester une table, aussi sélectionner des attributs non-grouvés ne fonctionne pas toujours :

```
SELECT Modèle, Fabricant FROM Avions
```

```
GROUP BY Fabricant
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200		2	440	14307
A321-200	Airbus	2	220	5926
A330-200		2	380	13890
A340-300		4	440	13705
CRJ-100	Bombardier	2	50	1574

## COUNT, MIN, MAX... et groupes

Les fonctions telles que COUNT ou MAX agissent sur les groupes :

```
SELECT Fabricant, COUNT(*) FROM Avions
```

GROUP BY Fabricant

Modèle	Fabricant	Moteurs	Places	Autonomie	COUNT
737-900	Boeing	2	215	5926	2
777-200		2	440	14307	
A321-200	Airbus	2	220	5926	3
A330-200		2	380	13890	
A340-300		4	440	13705	
CRJ-100	Bombardier	2	50	1574	1

## Groupes sur plusieurs critères

Il est possible de regrouper les enregistrements sur plusieurs attributs :

```
SELECT Fabricant, MIN(Places) FROM Avions
```

```
GROUP BY Fabricant, Moteurs
```

Modèle	Fabricant	Moteurs	Places	Autonomie	MIN(Pl.)
737-900	Boeing	2	215	5926	215
777-200			440	14307	
A321-200	Airbus	2	220	5926	220
A330-200			380	13890	
A340-300	Airbus	4	440	13705	440
CRJ-100	Bombardier	2	50	1574	50

## Filtrer les groupes

De la même façon que l'on peut sélectionner des enregistrements, on peut filtrer les groupes grâce au mot-clé HAVING :

```
SELECT Fabricant, MIN(Places) FROM Avions
```

```
GROUP BY Fabricant, Moteurs
```

```
HAVING MIN(Places) > 215
```

Modèle	Fabricant	Moteurs	Places	Autonomie	MIN(Pl.)
737-900	Boeing	2	215	5926	215
777-200			440	14307	
A321-200	Airbus	2	220	5926	220
A330-200			380	13890	
A340-300	Airbus	4	440	13705	440
CRJ-100	Bombardier	2	50	1574	50

## Double filtrage

WHERE et HAVING peuvent être utilisés dans une même requête (WHERE agit avant le regroupement, HAVING après) :

```
SELECT Fabricant, MIN(Places) FROM Avions
```

```
WHERE Autonomie > 10000
```

```
GROUP BY Fabricant, Moteurs
```

```
HAVING MIN(Places) > 215
```

Modèle	Fabricant	Moteurs	Places	Autonomie	MIN(Pl.)
737-900	Boeing	2	215	5926	440
777-200			440	14307	
A321-200	Airbus	2	220	5926	380
A330-200			380	13890	
A340-300	Airbus	4	440	13705	440
CRJ-100	Bombardier	2	50	1574	50

Il convient de conserver les éléments de la requête dans un ordre logique, correspondant à la chronologie des opérations :

- WHERE avant GROUP BY
- HAVING après GROUP BY
- ORDER BY après les opérations de sélection
- LIMIT / OFFSET en dernier

La projection (SELECT) est la dernière opération effectuée, elle n'est placée en tête que par commodité pour la lecture.

## Exemple de requête combinant tous les éléments

```
SELECT Fabricant, MIN(Places) FROM Avions
```

```
WHERE Autonomie > 10000
```

```
GROUP BY Fabricant, Moteurs
```

```
HAVING MIN(Places) > 215
```

```
ORDER BY MIN(Places) DESC
```

```
LIMIT 1 OFFSET 2
```

Modèle	Fabricant	Moteurs	Places	Autonomie	MIN(Pl.)
737-900	Boeing	2	215	5926	440
777-200			440	14307	
A321-200	Airbus	2	220	5926	380
A330-200			380	13890	
A340-300	Airbus	4	440	13705	440
CRJ-100	Bombardier	2	50	1574	50

Introduction

Effectuer des requêtes en langage SQL

Principe

Projection, sélection

Organisation des résultats

Aggrégation

**Sous-requêtes**

Jointures

Opérations ensemblistes

Création et modification de tables

Algèbre relationnelle

## Sous-requêtes

Le résultat d'une requête peut servir dans une autre requête. On place la sous-requête entre parenthèses.

Pour une sous-requête renvoyant une valeur, dans une expression :

```
SELECT * FROM Avions
```

```
WHERE Places > (SELECT AVG(Places) FROM Avions) :
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Sous-requêtes

Pour une sous-requête renvoyant une « liste » (table réduite à un seul attribut), dans une expression booléenne, avec le mot-clé IN :

```
SELECT Modèles FROM Avions
```

```
WHERE Fabricant IN
```

```
(SELECT DISTINCT Fabricant FROM Avions
```

```
WHERE Places > 400)
```

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Sous-requêtes

Enfin, une sous-requête renvoyant une « table » peut servir de table :

```
SELECT AVG(TMin) FROM
    (SELECT Fabricant, MIN(Places) AS TMin
     FROM Avions GROUP BY Fabricant)
```

Modèle	Fabricant	Mot.	Places	Auton.	TMin	AVG(TM.)
737-900	Boeing	2	215	5926	215	161.33
777-200		2	440	14307		
A321-200	Airbus	2	220	5926	220	
A330-200		2	380	13890		
A340-300		4	440	13705		
CRJ-100	Bombardier	2	50	1574	50	

On remarquera le AS pour nommer une colonne de la sous-requête.

Introduction

Effectuer des requêtes en langage SQL

Principe

Projection, sélection

Organisation des résultats

Aggrégation

Sous-requêtes

**Jointures**

Opérations ensemblistes

Création et modification de tables

Algèbre relationnelle

## Combiner les tables

Le plein intérêt des systèmes de gestion de bases de données ne se révèle que lorsque l'on croise les informations de plusieurs tables.

Ils permettent la création de tables temporaires regroupant les informations de plusieurs tables.

Il existe plusieurs façons de regrouper les tables, la plus simple étant un produit cartésien :

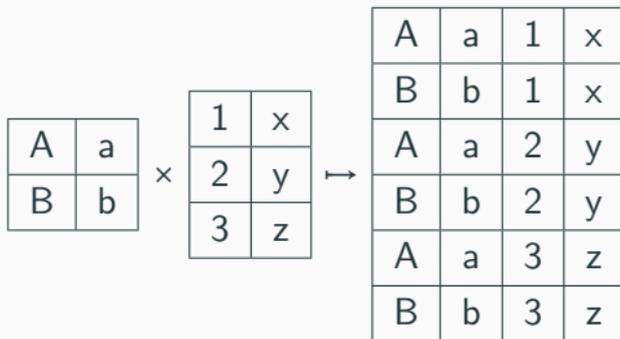


TABLE Avions :

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

TABLE Compagnies :

Code	Compagnie	Modèle	Nombre
AF	Air France	777-200	25
AF	Air France	A321-200	15
AF	Air France	A330-200	15
AF	Air France	A340-300	12
AC	Air Canada	777-200	6
AC	Air Canada	A321-200	10
AC	Air Canada	CRJ-100	51

# Produit cartésien de deux tables

TABLE Avions, Compagnies :

Avions					Compagnies			
Modèle	Fabricant	Moteurs	Places	Autonomie	Code	Compagnie	Modèle	Nombre
737-900	Boeing	2	215	5926	AF	Air France	777-200	25
777-200	Boeing	2	440	14307	AF	Air France	777-200	25
A321-200	Airbus	2	220	5926	AF	Air France	777-200	25
A330-200	Airbus	2	380	13890	AF	Air France	777-200	25
A340-300	Airbus	4	440	13705	AF	Air France	777-200	25
CRJ-100	Bombardier	2	50	1574	AF	Air France	777-200	25
737-900	Boeing	2	215	5926	AF	Air France	A321-200	15
777-200	Boeing	2	440	14307	AF	Air France	A321-200	15
A321-200	Airbus	2	220	5926	AF	Air France	A321-200	15
A330-200	Airbus	2	380	13890	AF	Air France	A321-200	15
A340-300	Airbus	4	440	13705	AF	Air France	A321-200	15
CRJ-100	Bombardier	2	50	1574	AF	Air France	A321-200	15
737-900	Boeing	2	215	5926	AF	Air France	A330-200	15
777-200	Boeing	2	440	14307	AF	Air France	A330-200	15
A321-200	Airbus	2	220	5926	AF	Air France	A330-200	15
A330-200	Airbus	2	380	13890	AF	Air France	A330-200	15
A340-300	Airbus	4	440	13705	AF	Air France	A330-200	15
CRJ-100	Bombardier	2	50	1574	AF	Air France	A330-200	15
...	...	...	...	...	...	...	...	...

(et 24 autres)

Puisque les tables peuvent avoir des attributs portant le même nom, on fait précéder le nom de l'attribut par le nom de la table pour les distinguer (Avions.Modèle, Compagnie.Code, etc.)

La table obtenue par le produit cartésien de deux tables n'a fréquemment guère de sens.

En effet, Si un même attribut est présent dans les deux tables (ici Avions.Modèle et Compagnie.Modèle), il est plus logique de ne s'intéresser qu'aux lignes où les deux attributs sont les mêmes.

# Rendre cohérent le produit cartésien

SELECT \* FROM Avions, Compagnies

WHERE Avions.Modèle = Compagnie.Modèle

Avions					Compagnies			
Modèle	Fabricant	Moteurs	Places	Autonomie	Code	Compagnie	Modèle	Nombre
737-900	Boeing	2	215	5926	AF	Air France	777-200	25
777-200	Boeing	2	440	14307	AF	Air France	777-200	25
A321-200	Airbus	2	220	5926	AF	Air France	777-200	25
A330-200	Airbus	2	380	13890	AF	Air France	777-200	25
A340-300	Airbus	4	440	13705	AF	Air France	777-200	25
CRJ-100	Bombardier	2	50	1574	AF	Air France	777-200	25
737-900	Boeing	2	215	5926	AF	Air France	A321-200	15
777-200	Boeing	2	440	14307	AF	Air France	A321-200	15
A321-200	Airbus	2	220	5926	AF	Air France	A321-200	15
A330-200	Airbus	2	380	13890	AF	Air France	A321-200	15
A340-300	Airbus	4	440	13705	AF	Air France	A321-200	15
CRJ-100	Bombardier	2	50	1574	AF	Air France	A321-200	15
737-900	Boeing	2	215	5926	AF	Air France	A330-200	15
777-200	Boeing	2	440	14307	AF	Air France	A330-200	15
A321-200	Airbus	2	220	5926	AF	Air France	A330-200	15
A330-200	Airbus	2	380	13890	AF	Air France	A330-200	15
A340-300	Airbus	4	440	13705	AF	Air France	A330-200	15
CRJ-100	Bombardier	2	50	1574	AF	Air France	A330-200	15
...	...	...	...	...	...	...	...	...

(et 24 autres)

On préférera effectuer une *jointure interne* des tables grâce au mot-clé JOIN (ou INNER JOIN)

```
SELECT * FROM Avions
```

```
JOIN Compagnies ON Avions.Modèle = Compagnie.Modèle
```

Avions					Compagnies			
Modèle	Fabricant	Moteurs	Places	Autonomie	Code	Compagnie	Modèle	Nombre
777-200	Boeing	2	440	14307	AF	Air France	777-200	25
A321-200	Airbus	2	220	5926	AF	Air France	A321-200	15
A330-200	Airbus	2	380	13890	AF	Air France	A330-200	15
A340-300	Airbus	4	440	13705	AF	Air France	A340-300	12
777-200	Boeing	2	440	14307	AC	Air Canada	777-200	6
A321-200	Airbus	2	220	5926	AC	Air Canada	A321-200	10
CRJ-100	Bombardier	2	50	1574	AC	Air Canada	CRJ-100	51

Une jointure interne donne le même résultat qu'un filtrage sur un produit cartésien, mais généralement plus rapidement.

On peut joindre plus de deux tables :

```
SELECT * FROM table1 JOIN table2 ON table1.x = table2.y  
                JOIN table3 ON table2.z = table3.t
```

Il existe d'autres jointures, notamment pour les cas où les éléments d'une table n'ont pas de correspondance dans l'autre table.

Le produit cartésien est un type de jointure (CROSS JOIN)

On peut utiliser AS pour définir des alias pour les noms des tables :

```
SELECT * FROM Avions AS a
```

```
JOIN Compagnies AS c ON a.Modèle = c.Modèle
```

Avions / a					Compagnies / c			
Modèle	Fabricant	Moteurs	Places	Autonomie	Code	Compagnie	Modèle	Nombre
777-200	Boeing	2	440	14307	AF	Air France	777-200	25
A321-200	Airbus	2	220	5926	AF	Air France	A321-200	15
A330-200	Airbus	2	380	13890	AF	Air France	A330-200	15
A340-300	Airbus	4	440	13705	AF	Air France	A340-300	12
777-200	Boeing	2	440	14307	AC	Air Canada	777-200	6
A321-200	Airbus	2	220	5926	AC	Air Canada	A321-200	10
CRJ-100	Bombardier	2	50	1574	AC	Air Canada	CRJ-100	51

Requis si l'on joint plusieurs fois la même table.

Introduction

Effectuer des requêtes en langage SQL

Principe

Projection, sélection

Organisation des résultats

Aggrégation

Sous-requêtes

Jointures

**Opérations ensemblistes**

Création et modification de tables

Algèbre relationnelle

## Combiner les requêtes

SELECT \* FROM Avions WHERE Places > 250

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

SELECT \* FROM Avions WHERE Fabricant = Airbus

Modèle	Fabricant	Moteurs	Places	Autonomie
737-900	Boeing	2	215	5926
777-200	Boeing	2	440	14307
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
CRJ-100	Bombardier	2	50	1574

## Union de requêtes $(Req_1 \cup Req_2)$

777-200	Boeing	2	440	14307
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705

SELECT \* FROM Avions WHERE Places > 250

UNION

SELECT \* FROM Avions WHERE Fabricant = Airbus

777-200	Boeing	2	440	14307
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
A321-200	Airbus	2	220	5926

Il doit y avoir le même nombre de colonnes, et les colonnes doivent être de même type, mais pas nécessairement contenir les mêmes attributs.

Les lignes apparaissant dans les deux requêtes ne sont retournées qu'une seule fois (utiliser UNION ALL sinon).

## Intersection de requêtes $(Req_1 \cap Req_2)$

777-200	Boeing	2	440	14307
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705

SELECT \* FROM Avions WHERE Places > 250

INTERSECT

SELECT \* FROM Avions WHERE Fabricant = Airbus

A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705

## Différence de requêtes (Req<sub>1</sub> – Req<sub>2</sub>)

777-200	Boeing	2	440	14307
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705
A321-200	Airbus	2	220	5926
A330-200	Airbus	2	380	13890
A340-300	Airbus	4	440	13705

SELECT \* FROM Avions WHERE Places > 250

MINUS (ou EXCEPT)

SELECT \* FROM Avions WHERE Fabricant = Airbus

777-200	Boeing	2	440	14307
---------	--------	---	-----	-------