

# Quelques exemples de requêtes SQL sur la base de données *aviation*

Note : divers exercices sont proposés. Leurs corrections se trouvent à la fin du document.

## Le mot-clé COUNT

---

Combien de modèles d'avion différents la compagnie Air France exploite-t-elle ?

On va travailler avec la table « flottes\_passagers ». On sélectionne les enregistrements où l'attribut vaut AF :

```
SELECT * FROM flottes_passagers WHERE ID_Compagnie='AF'
```

Mais avec cette requête, on obtient les enregistrements eux-mêmes (i.e. tous les attributs) alors que seul le nombre des enregistrements nous intéresse ! Pour compter ces enregistrements, il suffit alors d'utiliser la fonction COUNT comme suit :

```
SELECT COUNT(*) FROM flottes_passagers WHERE ID_Compagnie='AF'
```

On obtient alors simplement 2 qui est bien la valeur attendue !

Supposons maintenant que l'on se demande :

Combien de modèles d'avion différents les compagnies Air France et Emirates exploitent-elles ?

On peut « froidement » écrire :

```
SELECT COUNT(*)  
FROM flottes_passagers  
WHERE ID_Compagnie='AF' OR ID_Compagnie='EK'
```

Cette requête nous renvoie la valeur 8 qui n'est pas satisfaisante puisque, en regardant attentivement la table « flottes\_passagers », on constate que les avions exploités par Emirates, à savoir les modèles AL2 et AL6, sont également exploités par Air France. Le résultat correct est donc 6 au lieu de 8.

Pour l'obtenir, nous utilisons COUNT de la façon suivante :

```
SELECT COUNT(DISTINCT ID_AL)  
FROM flottes_passagers  
WHERE ID_Compagnie='AF' OR ID_Compagnie='EK'
```

L'utilisation du mot-clé DISTINCT appliqué à l'attribut ID\_AL permet, dans le résultat de la requête, de ne comptabiliser qu'une seule fois les lignes (il y en a deux) où l'attribut « ID\_Compagnie » vaut AF et une seule fois également celles (il y en a deux également) où l'attribut « ID\_Compagnie » vaut EK.

### **Exercice N°1**

Pour chaque famille de moteurs, on souhaite connaître le nombre de moteurs dont la poussée est comprise, strictement, entre 300 et 500 kN.

### **Exercice N2**

Pour chaque famille de moteurs, on souhaite connaître la poussée moyenne (en kN). On affichera le nom du fabricant, celui de la famille et, bien sûr, la poussée moyenne demandée (on utilisera la fonction AVG).

## **Le mot-clé GROUP BY**

---

Combien de modèles d'avion différents les compagnies référencées dans la base de données *aviation* exploitent-elles ?

On continue de travailler avec la table « flottes\_passagers ».

Cette fois, on souhaiterait que la fonction COUNT s'applique à Air France seule, à Emirates seule, à Quantas seule, ... En d'autres termes, à chacune des compagnies aériennes présentes dans la table. Pour cela, il suffit d'utiliser GROUP BY de la façon suivante :

```
SELECT ID_Compagnie, COUNT(*)
FROM flottes_passagers
GROUP BY ID_Compagnie
```

Si on souhaite trier les lignes obtenues suivant la deuxième colonne (le nombre de modèles d'avion différents exploités par chacune des compagnies), il suffit de rajouter un ORDER approprié à la fin de la requête précédente :

```
SELECT ID_Compagnie, COUNT(*)
FROM flottes_passagers
GROUP BY ID_Compagnie
ORDER BY COUNT(*) DESC
```

De façon similaire, on peut afficher, pour chaque modèle d'avion, le nombre de compagnies aériennes qui l'exploitent :

```
SELECT ID_AL, COUNT(*)
FROM flottes_passagers
GROUP BY ID_AL
ORDER BY COUNT(*) DESC
```

## Le mot-clé JOIN

---

Dans l'avant-dernière requête, par exemple, on peut ne pas être satisfait d'obtenir un tableau où figure, en première colonne, l'ID standard (code international) de la compagnie aérienne. Si l'on n'est pas trop habitué à ces codes, on préférera le nom complet de la compagnie. Ce nom se trouve dans la table « compagnies\_aériennes ». Comment le récupérer ?

On peut répondre de diverses façons...

Une solution consiste à construire une requête SELECT « travaillant » sur les deux tables concernées : « flottes\_passagers » et « compagnies\_aériennes ».

Dans la première, nous allons comptabiliser les divers modèles d'avion exploitées par chacune des compagnies aériennes et dans la seconde, les noms complets de ces compagnies.

```
SELECT Nom, COUNT(*)
FROM flottes_passagers AS fp, compagnies_aériennes AS ca
WHERE fp.ID_Compagnie=ca.ID_Compagnie
GROUP BY fp.ID_Compagnie
ORDER BY COUNT(*) DESC
```

Dans cette requête, il est indispensable de préciser que l'on ne retient que le(s) attribut(s) (ici Nom et les résultats de COUNT) correspondant à des enregistrement des deux tables pour lesquels les ID des compagnies sont identiques (d'où : WHERE fp.ID\_Compagnie=ca.ID\_Compagnie).

On note également que, pour simplifier l'écriture du GROUP BY, on a utilisé des alias : la table « flottes\_passagers » est désignée par « fp » et la table « compagnies\_aériennes » est désignée par « ca ». Une fois les alias définis grâce au mot-clé AS, on doit utiliser le(s) mot(s)-clé(s) (y compris avant le mot-clé AS si on a besoin de se référer à une table aliasée).

Une autre solution au problème posé consiste à effectuer une jointure des deux tables considérées. Au plan mathématique, la jointure s'apparente au produit cartésien et permet, à partir de deux tables d'en construire une troisième. N'hésitez pas, par exemple, à tester la requête suivante :

```
SELECT * FROM flottes_passagers JOIN compagnies_aériennes
```

On peut considérer que la jointure crée ainsi une nouvelle table dont les enregistrements sont obtenus en « rapprochant » les enregistrements de la table « flottes\_passagers » à ceux de la table « compagnies\_aériennes ».

Bien évidemment, la plupart des enregistrement ainsi obtenus n'ont aucun sens !

Comme précédemment, on va fournir un critère de sorte que les « rapprochements » effectués par la jointure aient un sens. Ce critère est introduit par le mot-clé ON :

```
SELECT Nom, COUNT(*)
FROM flottes_passagers AS fp JOIN compagnies_aériennes AS ca
ON fp.ID_Compagnie=ca.ID_Compagnie
GROUP BY fp.ID_Compagnie
ORDER BY COUNT(*) DESC, Nom
```

Remarque : à la fin de la requête, on a ajouté un tri sur l'attribut « Nom » pour trier les noms des compagnies possédant le même nombre de modèles d'avion différents.

### **Exercice N°3**

On souhaite connaître le nombre de Boeing B777-300ER exploités par les compagnies concernées. La requête devra renvoyer le nom en clair de la compagnie et le nombre correspondant.

On peut effectuer des jointures sur plus de deux tables.

Si, par exemple, on souhaite effectuer une jointure sur les tables T<sub>1</sub> et T<sub>2</sub>, puis sur la table T<sub>3</sub>, on écrira, formellement :

```
FROM (T1 JOIN T2) JOIN T3
```

Supposons par exemple que l'on souhaite connaître le nombre de Boeing 777-300ER et d'Airbus A380-841 pour chacune des compagnies exploitant ces avions.

On peut par exemple utiliser la requête suivante :

```
SELECT Nom, ID_ALFull, Nombre
FROM (avions_ligne AS al JOIN flottes_passagers AS fp
      ON al.ID_AL=fp.ID_AL)
JOIN compagnies_aériennes AS ca
ON ca.ID_Compagnie=fp.ID_Compagnie
WHERE al.ID_AL='AL2' OR al.ID_AL='AL5'
ORDER BY Nom, Nombre DESC
```

## Correction des exercices

---

### Exercice N°1

On travaille avec la table moteurs et on raisonne sur les attributs Famille et Poussée.

```
SELECT Famille, COUNT(*)
FROM moteurs
WHERE Poussée>'140' AND Poussée<'380'
GROUP BY Famille
ORDER BY COUNT(*) DESC
```

Remarque : notez que les contraintes introduites par WHERE suivent le FROM... Attention donc à l'ordre !

### Exercice N°2

On travaille bien sûr avec la table « moteurs ». La moyenne sera calculée grâce à la fonction AVG qui reçoit comme argument l'attribut « Poussée ». On doit effectuer un regroupement par famille car c'est sur ces groupes de moteurs que la fonction AVG va « travailler » :

```
SELECT Fabricant, Famille, AVG(Poussée)
FROM moteurs
GROUP BY Famille
ORDER BY AVG(Poussée) DESC
```

### Exercice N3

Un WHERE est ici indispensable pour ne conserver que les valeurs (attribut « Nombre ») relatives à l'avion considéré (B777-300ER) :

```
SELECT Nom, Nombre
FROM flottes_passagers AS fp JOIN compagnies_aériennes AS ca
ON fp.ID_Compagnie=ca.ID_Compagnie
WHERE ID_AL='AL2'
ORDER BY Nom
```