### R2.06 - Exploitation d'une base de données Cours 2 - Jointures et fonctions de groupe

A. Ridard



#### A propos de ce document

- Pour naviguer dans le document, vous pouvez utiliser :
  - le menu (en haut à gauche)
  - l'icône en dessous du logo IUT
  - les différents liens
- Pour signaler une erreur, vous pouvez envoyer un message à l'adresse suivante : anthony.ridard@univ-ubs.fr

#### Plan du cours

- Compléments sur les jointures
  - Rappels sur les jointures (internes)
  - Les jointures externes

2 Les fonctions de groupe

- Compléments sur les jointures
- 2 Les fonctions de groupe

- 1 Compléments sur les jointures
  - Rappels sur les jointures (internes)
  - Les jointures externes

2 Les fonctions de groupe

Pour chaque pilote désigné par son nom, afficher le nom de sa compagnie.



#### Jointure relationnelle

```
SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil = idComp
```



NOMPILOTE	NOMCOMP
Ridard	Air France
Naert	Easy Jet
Godin	Ryanair
Fleurquin	Air France
Pham	American Airlines
Kamp	American Airlines

On peut aussi utiliser la clause JOIN ... ON condition respectant la norme SQL2.

```
Jointure SQL2
```

```
SELECT nomPilote, nomComp
FROM Pilote
JOIN Compagnie ON compPil = idComp
;
```



- Les jointures relationnelles laissent le soin au SGBD d'établir la meilleure stratégie d'accès pour optimiser les performances. Attention, si vous oubliez une condition de jointure, un produit cartésien est composé!
- Les jointures SQL2 sont plus verbales et donc préférées par certains (au détriment parfois de la performance).

La directive INNER devant JOIN est optionnelle et est appliquée par défaut.

```
Jointure SQL2

SELECT nomPilote, nomComp
FROM Pilote
INNER JOIN Compagnie ON compPil = idComp
```

Pour chaque pilote désigné par son nom, afficher le nom de sa compagnie et toutes ses qualifications.



#### Double jointure relationnelle

```
SELECT nomPilote, nomComp, unTypeAvion FROM Pilote, Compagnie, Qualification WHERE compPil = idComp AND idPilote = unPilote
```

#### RESULT

NOMPILOTE	NOMCOMP	UNTYPEAVION
Ridard	Air France	A 320
		A 350
Naert		A 320
Naert		B747
Godin	Ryanair	A 320
Fleurquin	Air France	A 320
Fleurquin	Air France	A 350
Fleurquin	Air France	B747
Pham	American Airlines	A 350
Pham	American Airlines	A 320
Kamp	American Airlines	A 350
Kamp	American Airlines	B747
	Ridard Ridard Naert Godin Fleurquin Fleurquin Plam Pham Pham	Ridard Air France Ridard Air France Ridard Air France Naert EasyJet Godin Ryanair Fleurquin Air France Fleurquin Air France Fleurquin Air France Pham American Airlines Kamp American Airlines

L'écriture de la double jointure SQL2 se "complique".

## Double jointure SQL2

```
SELECT nomPilote, nomComp, unTypeAvion
FROM Pilote

JOIN Compagnie ON compPil = idComp

JOIN Qualification ON idPilote = unPilote
```



La clause JOIN permet aussi de programmer :

- les produits cartésiens avec CROSS JOIN
- les jointures naturelles avec NATURAL JOIN
- les jointures externes...

- Compléments sur les jointures
  - Rappels sur les jointures (internes)
  - Les jointures externes

Les fonctions de groupe

Pour chaque pilote (éventuellement sans compagnie) désigné par son nom, afficher le nom de sa compagnie (si elle existe).



```
SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil = idComp
OR compPil IS NULL
ORDER BY nomPilote
```



NOMPILOTE	NOMCOMP
	Air France
Fleurquin	Air France
Godin	Ryanair
Kamp	American Airlines
Kerbellec	Ryanair
Kerbellec	Corsair International
Kerbellec	Air France
Kerbellec	EasyJet
Kerbellec	American Airlines
Naert	EasyJet
Pham	American Airlines
Ridard	Air France

On doit faire une jointure externe :



#### Jointure externe relationnelle

```
SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil = idComp(+)
```

#### RESULT

NOMPILOTE	NOMCOMP
Fleurquin	Air France
Ridard	Air France
Naert	EasyJet
Kamp	American Airlines
Pham	American Airlines
Godin	Ryanair
• Kerbellec	



#### Jointure externe

- Les jointures externes permettent d'extraire des enregistrements qui ne répondent pas aux critères de jointure.
- Lorsque deux tables sont en jointure externe, une table est dominante par rapport à l'autre qui est dite subordonnée.
- Ce sont les enregistrements de la table dominante qui sont retournés (même si les valeurs des colonnes des tables subordonnées ne satisfont pas aux conditions de jointures ou sont NULL)



La table subordonnée d'une jointure externe relationnelle est indiquée par (+)

```
SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil = idComp(+)
```

On peut aussi utiliser la directive LEFT ou RIGHT devant JOIN.



```
SELECT nomPilote, nomComp
FROM Pilote
LEFT JOIN Compagnie ON compPil = idComp
;
```



Le sens de la directive de jointure externe **LEFT** ou **RIGHT** de la clause **JOIN** désigne la table dominante.

La directive **OUTER** devant **JOIN** est optionnelle et est appliquée par défaut.

```
Jointure externe SQL2

SELECT nomPilote, nomComp
FROM Pilote
LEFT OUTER JOIN Compagnie ON compPil = idComp
```

Pour chaque compagnie (éventuellement sans pilote) désignée par son nom, afficher le nom de ses pilotes (s'il en existe).

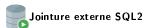
#### Jointure externe relationnelle

```
SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil(+)= idComp
ORDER BY nomComp
```

#### RESULT

NOMPILOTE	NOMCOMP
Ridard	Air France
Fleurquin	Air France
Pham	American Airlines
Kamp	American Airlines
	Corsair International
Naert	Easy Jet
Godin	Ryanair

On peut, là encore, utiliser la directive LEFT ou RIGHT devant JOIN.



```
SELECT nomPilote, nomComp
FROM Pilote
RIGHT JOIN Compagnie ON compPil = idComp
ORDER BY nomComp
```

Lorsque les deux tables sont « dominantes », on utilise FULL devant JOIN.

#### Jointure externe SQL2

```
SELECT nomPilote, nomComp
FROM Pilote
FULL JOIN Compagnie ON compPil = idComp
;
```

#### RESULT

NOMPILOTE	NOMCOMP
Fleurquin	Air France
Godin	Ryanair
Kamp	American Airlines
Kerbellec	
Naert	EasyJet
Pham	American Airlines
Ridard	Air France
•	Corsair International

Pour programmer une jointure **bilatérale** de manière relationnelle, il faut une union de jointures externes!

## Jointure externe relationnelle

```
SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil(+) = idComp

UNION

SELECT nomPilote, nomComp
FROM Pilote, Compagnie
WHERE compPil = idComp (+)
```

- Compléments sur les jointures
- 2 Les fonctions de groupe

Cette terminologie prendra tout son sens avec le regroupement...

#### Afficher le nombre de pilotes.

```
COUNT(*)
```

On peut nommer une colonne calculée.



```
NBPILOTES 7
```

Afficher le nombre de pilotes travaillant pour une compagnie.

```
Nombre de valeurs (NOT NULL)

| SELECT COUNT(compPil)
| FROM Pilote
```

```
COUNT (COMPPIL)
```

Afficher le nombre de compagnies employant (au moins) un pilote.



#### Nombre de valeurs distinctes

```
SELECT COUNT(DISTINCT compPil)
FROM Pilote
;
```

#### RESULT

```
COUNT(DISTINCTCOMPPIL)
```

Afficher le plus grand nombre d'heures de vol.

```
Maximum

| SELECT MAX(nbHVol)
| FROM Pilote
```

```
MAX(NBHVOL)
3000
```

Afficher le plus petit nombre d'heures de vol.

```
MIN (NBHVOL)
450
```

Afficher le nombre d'heures de vol total.

```
Somme

| SELECT SUM(nbHVol) | FROM Pilote | SUM(NBHVOL) | 10200
```

Afficher le nombre d'heures de vol moyen.

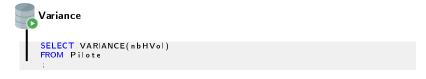
```
Moyenne

| SELECT AVG(nbHVol)
FROM Pilote
;
```

```
AVG(NBHVOL)

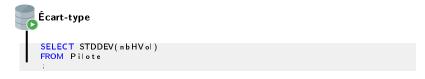
1457,14286
```

Afficher la variance du nombre d'heures de vol.



## VARIA

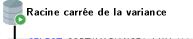
VARIANCE (NBHVOL) 1235357,14 Afficher l'écart-type du nombre d'heures de vol.



```
STDDEV(NBHVOL)

1111,46621
```

On peut aussi utiliser une fonction numérique.



```
SELECT SQRT(VARIANCE(nbHVol)) ecart_type FROM Pilote
```

```
ECART_TYPE
1111.46621
```

Afficher le(s) pilote(s) ayant le plus grand nombre d'heures de vol.



```
SELECT nomPilote
FROM Pilote
WHERE nbHVol = MAX(nbHVol)
```



```
/*
Rapport d'erreur —
Erreur SQL: ORA—00934: group function is not allowed here
00934. 00000 — "group function is not allowed here"
*/
```

Il faut utiliser une sous-interrogation.

# Argument du maximum SELECT nomPilote

```
FROM Pilote
WHERE nbHVol =

(
SELECT MAX(nbHVol)
FROM Pilote
)
:
```

```
NOMPILOTE

Fleurquin
Kamp
```

Afficher les pilotes ayant un nombre d'heures de vol supérieur à la moyenne.

```
SELECT nomPilote
FROM Pilote
WHERE nbHVol >

(
SELECT AVG(nbHVol)
FROM Pilote
);
```

```
NOMPILOTE

Ridard
Fleurquin
Kamp
```

Afficher les pilotes ayant un nombre d'heures de vol supérieur à la moyenne.

```
SELECT nomPilote
FROM Pilote
WHERE nbHVol >

(
SELECT AVG(nbHVol)
FROM Pilote
);
```

```
NOMPILOTE

Ridard
Fleurquin
Kamp
```

Pour afficher aussi le nombre d'heures de vol.



```
SELECT nomPilote, nbHVol
FROM Pilote
WHERE nbHVol >

(
SELECT AVG(nbHVol)
FROM Pilote
);
```

#### RESULT

I	
NOMPILOTE	NBHVOL
Ridard	1500
Fleurquin	3000
Kamp	3000



Comment peut-on afficher, en plus du nom et du nombre d'heures, la moyenne?

## Erreur

```
SELECT nomPilote, nbHVol, moyenne
FROM Pilote
WHERE nbHVol >

(
SELECT AVG(nbHVol) moyenne
FROM Pilote
);
```



```
/*

Erreur à la ligne de commande: 214 Colonne: 27

Rapport d'erreur —

Erreur SQL : ORA—00904: "MOYENNE": invalid identifier

00904. 00000 — "%s: invalid identifier"

*/
```

## Erreur

```
SELECT nomPilote, nbHVol, AVG(nbHVol) moyenne
FROM Pilote
WHERE nbHVol >

(SELECT AVG(nbHVol)
FROM Pilote
)
;
```



```
/*

Erreur à la ligne de commande: 247 Colonne: 8

Rapport d'erreur —

Erreur SQL : ORA—00937: not a single—group group function 00937. 00000 — "not a single—group group function":
```



```
SELECT nomPilote, nbHVol, moyenne
FROM Pilote,

(
SELECT AVG(nbHVol) moyenne
FROM Pilote
)
WHERE nbHVol > moyenne
```

#### RESULT

ı			
ı	NOMPILOTE	NBHVOL	MOYENNE
ı	Ridard	1500	1457,14286
ı	Fleurquin	3000	1457,14286
•	Kamp	3000	1457,14286



- Afficher le nombre de compagnies low cost (de deux manières différentes).
- Afficher le nombre d'avions appartenant à Air France.
- 3 Afficher le nombre de type d'avions appartenant à Ryanair.
- 4 Afficher le nombre total de passagers que peut transporter Air France.
- Afficher le type d'avion ayant la capacité maximale.
- 4 Afficher les compagnies (nom) ayant le type d'avion de capacité maximale.