

R2.06 - Exploitation d'une base de données Cours 4 - Division

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

1 En algèbre relationnelle

- Division
- Division exacte

2 En SQL

- Division
- Division exacte

- 1 En algèbre relationnelle
- 2 En SQL

1 En algèbre relationnelle

- Division
- Division exacte

2 En SQL

- Division
- Division exacte

Définition (division)

Le résultat de la division de $R_D(x,y)$ par $R_d(y)$ est la relation $R_Q(x)$ comportant le plus grand ensemble possible de tuples qui concaténés à ceux de $R_d(y)$ donnent un tuple de $R_D(x,y)$. Autrement dit, c'est la plus grande^a relation $R_Q(x)$ vérifiant $R_Q(x) \times R_d(y) \subset R_D(x,y)$.

a. Au sens de l'inclusion qui est une relation d'ordre comme vous le savez.



C'est (évidemment) à comparer avec la division euclidienne $D = Q \times d + r$ où le quotient Q est le plus grand entier vérifiant $Q \times d \leq D$

Considérons $R_D(x,y)$ et $R_d(y)$ les deux relations suivantes :

x	y
A	1
B	1
B	2
C	1
C	2
C	3

 et

y
1
2

Alors la relation $R_Q(x)$ vaut :

x
B
C

Le "reste" est alors :

x	y
A	1
C	3

1 En algèbre relationnelle

- Division
- Division exacte

2 En SQL

- Division
- Division exacte

Définition (division exacte)

Le résultat de la division **exacte** de $R_D(x,y)$ par $R_d(y)$ est la plus grande relation $R_{Q^*}(x)$ vérifiant $R_{Q^*}(x) \times R_d(y) \subset R_D(x,y)$ de sorte que le "reste" (projeté sur x) ne contienne aucun élément de $R_{Q^*}(x)$.

Reprenons $R_D(x,y)$ et $R_d(y)$ les deux relations suivantes :

x	y
A	1
B	1
B	2
C	1
C	2
C	3

 et

y
1
2

Alors la relation $R_{Q^*}(x)$ vaut :

x
B

Le "reste" est alors :

x	y
A	1
C	1
C	2
C	3

- 1 En algèbre relationnelle
- 2 En SQL

1 En algèbre relationnelle

- Division
- Division exacte

2 En SQL

- **Division**
- Division exacte

Afficher les noms des pilotes possédant une qualification **pour tous les avions**.



```
SELECT nomPilote
FROM Pilote
WHERE NOT EXISTS
(
  SELECT idTypeAvion
  FROM TypeAvion
  MINUS
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
)
```

RESULT

NOMPILOTE
Fleurquin



I Proposer une solution avec un regroupement.



Solution

```
SELECT nomPilote
FROM Pilote
     JOIN Qualification ON idPilote = unPilote
GROUP BY nomPilote
HAVING COUNT(*) >=
    (
        SELECT COUNT(*)
        FROM TypeAvion
    )
;
```



Afficher les noms des pilotes possédant une qualification **pour tous les avions de leur compagnie.**



Solution

```
SELECT nomPilote
FROM Pilote
WHERE NOT EXISTS
(
  SELECT leTypeAvion
  FROM Avion
  WHERE compAv = compPil
  MINUS
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
)
AND compPil IS NOT NULL — Pourquoi est-ce nécessaire ?
;
```

RESULT

NOMPILOTE

Naert
Godin
Fleurquin
Kamp

Un étudiant propose :

```
SELECT nomPilote
FROM Pilote
JOIN Qualification ON idPilote = unPilote
GROUP BY nomPilote
HAVING COUNT(*) >=
(
SELECT COUNT(DISTINCT leTypeAvion)
FROM Avion
WHERE compAv = compPil
)
;
```

Cela provoque l'erreur suivante :

```
/*
Rapport d'erreur -
Erreur SQL : ORA-00979: not a GROUP BY expression
00979. 00000 - "not a GROUP BY expression"
*/
```



1 Corriger cette erreur.



La solution suivante est-elle alors valable ?

```
SELECT nomPilote , compPil
FROM Pilote
      JOIN Qualification ON idPilote = unPilote
GROUP BY nomPilote , compPil
HAVING COUNT(*) >=
(
  SELECT COUNT(DISTINCT leTypeAvion)
  FROM Avion
  WHERE compAv = compPil
)
```

RESULT

NOMPILOTE	COMPPIL
Fleurquin	1
Pham	4
Godin	5
Naert	3
Kamp	4



Pour implémenter une division à l'aide d'un regroupement, la table par laquelle on divise doit contenir toutes les valeurs possibles.
Autrement dit, le domaine du "y" de la table "**D**ividende" doit être inclus dans celui du "y" de la table "**d**iviseur".



Il faut restreindre la table Dividende

```
SELECT nomPilote , compPil
FROM Pilote
      JOIN Qualification ON idPilote = unPilote
      JOIN Avion ON compAv = compPil
WHERE unTypeAvion = leTypeAvion
GROUP BY nomPilote , compPil
HAVING COUNT(DISTINCT unTypeAvion) >= — cela ne peut pas être >
(
  SELECT COUNT(DISTINCT leTypeAvion)
  FROM Avion
  WHERE compAv = compPil
)
;
```



Le regroupement peut être plus simple, mais ce n'est pas le cas en général!!!

1 En algèbre relationnelle

- Division
- Division exacte

2 En SQL

- Division
- Division exacte



Afficher les noms des pilotes possédant une qualification **pour exactement tous les avions de leur compagnie.**



Solution

```
SELECT nomPilote
FROM Pilote
WHERE NOT EXISTS
(
  SELECT leTypeAvion
  FROM Avion
  WHERE compAv = compPil
  MINUS
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
)
AND NOT EXISTS
(
  SELECT unTypeAvion
  FROM Qualification
  WHERE unPilote = idPilote
  MINUS
  SELECT leTypeAvion
  FROM Avion
  WHERE compAv = compPil
)
AND compPil IS NOT NULL
;
```


RESULT

NOMPILOTE

Godin
Fleurquin
Kamp



Avec un regroupement, c'est bien plus délicat !

```
SELECT nomPilote, compPil
FROM Pilote
      JOIN Qualification ON idPilote = unPilote
      JOIN Avion ON compAv = compPil
WHERE unTypeAvion = leTypeAvion
GROUP BY nomPilote, compPil
HAVING COUNT(DISTINCT leTypeAvion) >=
(
  SELECT COUNT(DISTINCT leTypeAvion)
  FROM Avion
  WHERE compAv = compPil
)
INTERSECT — on exclue les pilotes ayant trop de qualifications
SELECT nomPilote, compPil
FROM Pilote
      JOIN Qualification ON idPilote = unPilote
GROUP BY nomPilote, compPil
HAVING COUNT(*) =
(
  SELECT COUNT(DISTINCT leTypeAvion)
  FROM Avion
  WHERE compAv = compPil
)
;
```



Afficher, d'abord avec **NOT EXISTS** (et **MINUS**), puis avec un regroupement :

- 1 les noms des compagnies possédant **tous les types d'Airbus**^a.
- 2 les noms des compagnies possédant **exactement tous les types d'Airbus**.

a. Le type d'un Airbus commence (évidemment) par la lettre "A"