SQL Avancé
La division relationnelle

La division est un opérateur algébrique qui permet de comparer deux ensembles au sens de l'inclusion (division inexacte) ou de l'égalité (division exacte). Cet opérateur n'est pas implémenté dans le standard SQL ni dans aucun SGBD, il faut donc le simuler.

La division permet de répondre aux questions qui comportent en général les termes 'tous' ou 'toutes' comme par exemple "les employés qui sont affectés à **tous** les projets". L'idée est donc de comparer un ensemble, le dividende avec un ensemble référence, le diviseur.

Exemple :

EMPLOYES (idEmploye, nomEmploye, prenomEmploye)

E1	Nemard	Jean
E2	Bricot	Judas
E3	Palleja	Nathalie
E4	Deuf	John

PROJETS (idProjet, nomProjet, categorieProjet)

P1	x	Agile
P2	y	Cascade
P3	z	Agile

ETREAFFECTE (idEmploye#, idProjet#)

E1	P1
E1	P2
E1	P3
E2	P1
E2	P3
E3	P1
E3	P2
E4	P1
E4	P2

1 Division inexacte

La division inexacte consiste à vérifier que le diviseur est inclus dans le dividende c'est à dire rechercher dans l'ensemble dividende les lignes qui sont associées à toutes les lignes de l'ensemble diviseur.

Pour effectuer cette division, nous allons voir deux solutions, la première qui utilise les regroupements et la deuxième l'anti-jointure.

Requête 1 : l'identifiant des employés qui sont affectés sur tous les projets

DIVIDENDE : table ETREAFFECTE

idEmploye	idProjet
E1	P1
E1	P2
E1	P3
E2	P1
E2	P3
E3	P1
E3	P2
E4	P1
E4	P2

DIVISEUR : table PROJETS

idProjet
P1
P2
P3

idEmploye
E1

On cherche les idEmploye de la table ETREAFFECTE dont les idProjet sont associés à tous les idProjet de la table PROJETS.

1ère solution :

Dans cette solution, on cherche les employés qui sont affectés à un nombre de projets égal au nombre total de projets.

Diviseur : on compte le nombre de projets dans la table PROJETS

Dividende : on compte le nombre de projets par employé dans la table ETREAFFECTE

dividende

```
SELECT idEmploye
FROM EtreAffecte
GROUP BY idEmploye
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Projets);
```

diviseur

2ème solution :

Une solution pour vérifier que le diviseur est inclus dans le dividende est de programmer la condition diviseur - dividende = ∅.

Pour cette requête, il faut donc chercher les employés pour lesquels la différence (MINUS) entre les projets de la table PROJETS (diviseur) et les projets auxquels l'employé est affecté dans la table ETREAFFECTE (dividende) n'existe pas (NOT EXISTS).

```
SELECT idEmploye
FROM Employes e
WHERE NOT EXISTS (SELECT idProjet
                  FROM Projets
                 MINUS
                  SELECT idProjet
                  FROM EtreAffecte ea
                  WHERE ea.idEmploye = e.idEmploye);
```

diviseur

dividende

Requête 2 : l'identifiant des employés qui sont affectés sur tous les projets de la catégorie 'Agile'

DIVIDENDE : les employés de la table ETREAFFECTE qui sont affectés à des projets Agile'

idEmploye	idProjet	categorieProjet
E1	P1	Agile
~~E1~~	~~P2~~	~~Cascade~~
E1	P3	Agile
E2	P1	Agile
E2	P3	Agile
E3	P1	Agile
~~E3~~	~~P2~~	~~Cascade~~
E4	P1	Agile
~~E4~~	~~P2~~	~~Cascade~~

DIVISEUR : les projets 'Agile' de la table PROJETS

idProjet	categorieProjet
P1	Agile
~~P2~~	~~Cascade~~
P3	Agile

idEmploye
E1
E2

1ère solution :

On cherche les employés qui sont affectés à un nombre de projets 'Agile' égal au nombre total de projets 'Agile'.

```
SELECT idEmploye
FROM EtreAffecte ea
     JOIN Projets p ON ea.idProjet = p.idProjet
WHERE categorieProjet = 'Agile'
GROUP BY idEmploye
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Projets
                   WHERE categorieProjet = 'Agile');
```

2ème solution :

On cherche les employés pour lesquels la différence (MINUS) entre les projets 'Agile' de la table PROJETS (diviseur) et les projets auxquels l'employé est affecté dans la table ETREAFFECTE (dividende) n'existe pas (NOT EXISTS).

```
SELECT idEmploye
FROM Employes e
WHERE NOT EXISTS (SELECT idProjet
                  FROM Projets
                  WHERE categorieProjet = 'Agile'
                 MINUS
                  SELECT idProjet
                  FROM EtreAffecte ea
                  WHERE ea.idEmploye = e.idEmploye);
```

diviseur (A)

dividende (B)

Remarque : dans cette solution, il est inutile de sélectionner les projets 'Agile' dans la table ETREAFFECTE (dividende) car la différence sélectionne les projets de la catégorie 'Agile' (A) et retire les projets de B qui sont en commun avec A. Il reste donc les projets qui sont seulement dans A et pas dans B même si il y a des projets supplémentaires dans B.

Requête 3 : l'identifiant des employés qui sont affectés sur des projets de toutes les catégories.

DIVIDENDE : table ETREAFFECTE JOIN PROJETS

idEmploye	idProjet	categorieProjet
E1	P1	Agile
E1	P2	Cascade
E1	P3	Agile
E2	P1	Agile
E2	P3	Agile
E3	P1	Agile
E3	P2	Cascade
E4	P1	Agile
E4	P2	Cascade

DIVISEUR : les différentes catégories de projets de la table PROJETS

categorieProjet
Agile
Cascade
Agile

idEmploye
E1
E3
E4

1ère solution :

On cherche les employés qui sont affectés à un nombre de catégories **différentes** égal au nombre total de catégories **différentes** de projets.

```
SELECT idEmploye
FROM EtreAffecte ea
     JOIN Projets p ON ea.idProjet = p.idProjet
GROUP BY idEmploye
HAVING COUNT(DISTINCT categorieProjet) =
                                    (SELECT COUNT(DISTINCT categorieProjet)
                                     FROM Projets);
```

2ème solution :

Pour cette requête, il faut donc chercher les employés pour lesquels la différence (MINUS) entre les catégories de projets de la table PROJETS (diviseur) et les catégories de projets auxquels l'employé est affecté dans la table ETREAFFECTE (dividende) n'existe pas (NOT EXISTS).

```
SELECT idEmploye
FROM Employes e
WHERE NOT EXISTS (SELECT categorieProjet
                  FROM Projets
                 MINUS
                  SELECT categorieProjet
                  FROM EtreAffecte ea
                  JOIN Projets p ON ea.idProjet = p.idProjet
                  WHERE ea.idEmploye = e.idEmploye);
```

Remarque : dans cette solution, le `DISTINCT` est inutile car le `MINUS` fait un `DISTINCT`.

Requête 4 : l'identifiant des employés qui sont affectés sur tous les projets où est affecté l'employé 'E3'.

On cherche les employés dont tous les projets de 'E3' sont inclus dans leurs projets c'est à dire les employés qui sont affectés aux mêmes projets que 'E3' mais qui peuvent aussi être affectés à d'autres projets.

DIVIDENDE : les employés de table ETREAFFECTE qui sont affectés à des projets parmi ceux de 'E3'

idEmploye	idProjet
E1	P1
E1	P2
~~E1~~	~~P3~~
E2	P1
~~E2~~	~~P3~~
E3	P1
E3	P2
E4	P1
E4	P2

DIVISEUR : les projets de l'employé 'E3' dans la table ETREAFFECTE

idProjet
P1
P2

idEmploye
E1
E3
E4

1ère solution :

On cherche les employés qui sont affectés à un nombre de projets parmi les projets où est affecté 'E3' égal au nombre total de projets où est affecté 'E3'.

```
SELECT idEmploye
FROM EtreAffecte ea
WHERE idProjet IN (SELECT idProjet
                   FROM EtreAffecte
                   WHERE idEmploye = 'E3')
GROUP BY idEmploye
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM EtreAffecte
                   WHERE idEmploye = 'E3');
```

Remarque : il est indispensable dans le dividende de sélectionner pour chaque employé les projets auxquels il est affecté parmi les projets où est affecté de 'E3'. Si on ne rajoute pas cette condition, la requête retournera les employés qui sont affectés au même nombre de projets que 'E3'.

2ème solution :

Pour cette requête, il faut donc chercher les employés pour lesquels la différence (MINUS) entre les de projets de 'E3' de la table ETREAFFECTE (diviseur) et les projets auxquels l'employé est affecté dans la table ETREAFFECTE (dividende) n'existe pas (NOT EXISTS).

```
SELECT idEmploye
FROM Employes e
WHERE NOT EXISTS (SELECT idProjet
                  FROM EtreAffecte
                  WHERE idEmploye = 'E3'
                 MINUS
                  SELECT idProjet
                  FROM EtreAffecte ea
                  WHERE ea.idEmploye = e.idEmploye);
```

diviseur (A)

dividende (B)

Remarque : dans cette solution, il est inutile de sélectionner les projets de 'E3' dans l'ensemble B car la différence sélectionne les projets de 'E3' (A) et retire les projets de B qui sont en commun avec A. Il reste donc les projets qui sont seulement dans A et pas dans B même si il y a des projets supplémentaires dans B.

2 Division exacte

La division exacte consiste à vérifier que deux ensembles sont égaux. Une division exacte est donc une division qui n'a pas de reste contrairement à la division inexacte qui elle peut avoir un reste.

La division exacte permet de répondre aux questions qui comportent en général le terme 'exactement' comme par exemple "les employés qui sont affectés **exactement** aux mêmes projets que l'employé 'E3'". Pour répondre à cette question, il faut donc rechercher dans l'ensemble dividende les lignes qui correspondent exactement à toutes les lignes de l'ensemble diviseur.

La solution la plus simple pour vérifier que deux ensembles (A) et (B) sont égaux est de programmer les deux conditions (A) - (B) = ∅ **ET** (B) - (A) = ∅ (mais on pourrait réaliser cette requête autrement).

Requête 5 : l'identifiant des employés qui sont affectés sur tous les projets où est affecté l'employé 'E3' et uniquement à ceux-là (c'est-à-dire les employés qui sont affectés exactement aux mêmes projets que l'employée 'E3')

DIVIDENDE : les employés de table ETREAFFECTE qui sont affectés à des projets parmi ceux de 'E3'

idEmploye	idProjet
E1	P1
E1	P2
~~E1~~	~~P3~~
E2	P1
~~E2~~	~~P3~~
E3	P1
E3	P2
E4	P1
E4	P2

DIVISEUR : les projets de l'employé 'E3' dans la table ETREAFFECTE

idProjet
P1
P2

idEmploye
E3
E4

```
SELECT idEmploye
FROM Employes e
WHERE NOT EXISTS (SELECT idProjet                          } diviseur (A)
                  FROM EtreAffecte
                  WHERE idEmploye = 'E3'
                 MINUS
                  SELECT idProjet                          } dividende (B)
                  FROM EtreAffecte ea
                  WHERE ea.idEmploye = e.idEmploye)
AND NOT EXISTS   (SELECT idProjet                          } dividende (B)
                  FROM EtreAffecte ea
                  WHERE ea.idEmploye = e.idEmploye
                 MINUS
                  SELECT idProjet                          } diviseur (A)
                  FROM EtreAffecte
                  WHERE idEmploye = 'E3');
```

Remarque : le deuxième **NOT EXISTS** vérifie que la division n'a pas de reste.