

Correction du TD 7 - SQL : agrégats, sélection des agrégats

Base de données formula1

La base de données formula1 est composée des 7 relations suivantes :

- **circuit**(circuitID, cName, cCity, cCountry, cLength, cLapRec, #cDrivRec *réf. driverID*, cYearRec)
- **driver**(driverID, dFirstName, dLastName, dBirthdate, dDeathdate, dCountry, dGender)
- **grandprix**(gpID, gName, #circuitID, gDate, gLaps, gRank)
- **racedriver**(#teamID, #driverID, rSeason, rDriverNb)
- **standings**(#driverID, #gpID, sGrid, sPos, sRes, sPoints, sLaps)
- **team**(teamID, tName, tCountry, #twas *réf. teamID*)
- **tesdriver**(#teamID, #driverID, tSeason)

Cette base est utilisée dans les exercices qui suivent. Les requêtes peuvent être testées sur les sites :

- <http://pedago.uhb.fr/sql/> (à préférer à l'université),
- <http://bdur2m.free.fr/> (avec le login etudiant et le mot de passe rennes2).

Exercice 1. Requêtes avec agrégats (regroupements)

Donner les requêtes SQL qui renvoient les informations suivantes :

1. le nombre de pilotes de chaque nationalité (nationalité et nombre)

```
SELECT dCountry, COUNT(*) AS NbPilotes
FROM driver
GROUP BY dCountry
```

2. même chose, mais seulement pour les pilotes de course de la saison 2014, classés par nombre de pilote décroissant, puis par ordre alphabétique de pays

```
SELECT dCountry, COUNT(*) AS NbPilotes
FROM driver NATURAL JOIN racedriver
WHERE rSeason = 2014
GROUP BY dCountry
ORDER BY NbPilotes DESC, dCountry
```

Cette solution considère que les pilotes de course de la saison 2014 sont ceux engagés par une équipe en 2014.

Une autre possibilité consisterait à récupérer les pilotes qui ont participé à une course en 2014. Cependant, cette 2^e solution serait plus coûteuse. En effet, la table des résultats étant bien plus grande que celle des engagements, la jointure de la 2^e solution prendra plus de temps que celle de la 1^{re}. De plus, elles ne sont pas totalement équivalentes, car les 2 requêtes ne donnent la même réponse que lorsque les résultats de toutes les courses sont renseignés ! En effet, il suffit d'une course pour avoir les pilotes titulaires, qui en principe participent tous, mais il

faut attendre la fin de la saison pour avoir tous les pilotes, car certains pilotes remplaçant sont engagés pour 1 course ou 2 en cours de saison.

3. le nombre de points de chaque pilote en 2014 (nom, prenom, points), du meilleur au moins bon

```
SELECT dLastName, dFirstName, SUM(sPoints) AS totalPoints
FROM driver NATURAL JOIN standings
           NATURAL JOIN grandprix
WHERE YEAR(gDate) = 2014
GROUP BY driverID, dLastName, dFirstName
ORDER BY SUM(sPoints) DESC, dLastName
```

Bien que n'étant pas affiché, le champ de groupement driverID est dans le GROUP BY : comme on veut regrouper les données par pilote, le meilleur moyen est de grouper selon la clé de la table. Et comme le groupement se fait selon la clé de la table (driverID), les valeurs de dLastName et dFirstName sont identiques dans chaque agrégat (groupe). Ces 2 champs ne sont donc pas utiles au regroupement. Leur présence dans le GROUP BY est cependant nécessaire dans certains SGBD, qui n'autorisent dans la clause SELECT que les champs du regroupement et les champs calculés (version 1 ci-dessus). Avec MySQL, on peut les omettre (version 2 ci-dessous), et c'est ce qu'on fait en général !

```
SELECT dLastName, dFirstName, SUM(sPoints) AS totalPoints
FROM driver NATURAL JOIN standings
           NATURAL JOIN grandprix
WHERE YEAR(gDate) = 2014
GROUP BY driverID
ORDER BY SUM(sPoints) DESC, dLastName
```

La jointure sur la table grandprix sert juste à filtrer les résultats de la saison 2104. On peut *gagner* cette jointure en remarquant que l'identifiant du grand prix gpID présent dans la table standings est composé de l'année et du numéro d'ordre du grand prix. Par exemple, 201402 est le gpID du 2^e grand prix de la saison 2014. On peut alors simplifier la requête en :

```
SELECT dLastName, dFirstName, SUM(sPoints) AS totalPoints
FROM driver NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014'
GROUP BY driverID
ORDER BY SUM(sPoints) DESC, dLastName
```

4. pour chaque grand prix de 2014, le nombre de pilotes classés (afficher « classé(s) »), disqualifiés (DQ), n'ayant pas pris le départ (DNS) et qui ont abandonné (DNF). Ces informations sont disponibles dans la colonne des « incidents de course ». Afficher le nom du grand prix (par ordre chronologique), le type de résultat et le nombre de pilotes concernés.

```
SELECT gName, COALESCE(sInc, 'classé(s)'), COUNT(*)
FROM grandprix NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014'
GROUP BY gpID, sInc
ORDER BY gDate, sInc
```

5. pour chaque pilote (nom, prénom), le nombre total de grands prix courus dans sa carrière (éventuellement 0)

```
SELECT dLastName, dFirstName, COUNT(gpID)
FROM driver NATURAL LEFT JOIN standings
GROUP BY driverID
ORDER BY COUNT(gpID) DESC
```

Remarquez qu'il s'agit d'un des rares cas où on ne peut pas remplacer `COUNT(gpID)` par un `COUNT(*)` précédé d'un filtrage (`WHERE`) des valeurs nulles !

6. le nombre de points de chaque équipe en 2014 (nom, points), de la meilleure à la moins bonne

```
SELECT tName, SUM(sPoints) AS totalPoints
FROM team NATURAL JOIN racedriver
      NATURAL JOIN driver
      NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014' AND rSeason = 2014
GROUP BY teamID
ORDER BY SUM(sPoints) DESC, tName
```

7. [difficile / facultative] même chose, mais avec aussi le détail par pilote

```
SELECT tName, dLastName, SUM(sPoints) AS totalPoints
FROM team NATURAL JOIN racedriver
      NATURAL JOIN driver
      NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014' AND rSeason = 2014
GROUP BY tName, dLastName WITH ROLLUP
```

En règle générale, on groupe plutôt selon l'identifiant (`teamID` et `driverID` ici) pour éviter les problèmes d'homonymie. Pour comprendre pourquoi on a dû choisir `tName` et `dLastName` ici, il suffit de les remplacer par `teamID` et `driverID`, puis éventuellement de mettre les 4 dans le `GROUP BY...` De plus, avec `tName` et `dLastName`, on est certain de ne pas capturer d'homonymes dans cette requête. En effet, le premier critère est le nom d'équipe : il n'y a pas 2 équipes de même nom qui courent la même année. Et si on trouve souvent des noms de pilote identiques dans l'histoire (des pilotes de père en fils!), les chances d'avoir des pilotes homonymes dans la même équipe lors d'une même saison sont infimes.

Noter aussi qu'avec un `ROLLUP` on ne peut pas utiliser `ORDER BY`.

Exercice 2. Requêtes avec sélection sur les agrégats

Donner les requêtes SQL qui renvoient les informations suivantes :

1. les pays qui ont au moins 5 pilotes vivants (classés du pays qui a le plus de pilotes à celui qui en a le moins (on ne demande pas d'afficher combien il y en a))

```
SELECT dCountry
FROM driver
WHERE dDeathDate IS NULL
GROUP BY dCountry
HAVING COUNT(*) >= 5
ORDER BY COUNT(*) DESC
```

2. les pilotes (prénom, nom) qui ont fait au moins 5 podiums en 2014

```
SELECT dFirstName, dLastName
FROM driver NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = 2014
    AND sPos <= 3
GROUP BY driverID
HAVING COUNT(*) >= 5
```

3. les pilotes (prénom, nom) avec leur meilleur résultat en 2014, pour les pilotes qui ont fini au moins une fois « dans les points » (dans les 10 premiers); donner une version avec HAVING, et une version sans

```
SELECT dFirstName, dLastName, MIN(sPos) AS 'best result'
FROM driver NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014'
GROUP BY driverID
HAVING MIN(sPos) <= 10
```

```
SELECT dFirstName, dLastName, MIN(sPos) AS 'best result'
FROM driver NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014'
    AND sPos <= 10
GROUP BY driverID
```

4. le nom des équipes dont au moins 2 pilotes (différents) ont fini sur le podium en 2014

```
SELECT tName
FROM team NATURAL JOIN racedriver
    NATURAL JOIN standings
WHERE SUBSTRING(gpID,1,4) = '2014'
    AND rSeason = 2014
    AND sPos <= 3
GROUP BY teamID
HAVING COUNT(DISTINCT driverID) >= 2
```

Vous devez aussi pouvoir répondre aux questions :

- Pourquoi n’y a-t-il pas la table `driver` dans la jointure ?
- Pourquoi filtrer 2 fois sur la saison 2014 ?

5. les nationalités (de pilotes) pour lesquelles au moins 2 pilotes masculins sont décédés

```
SELECT dCountry
FROM driver
WHERE dGender = 'M' AND dDeathdate IS NOT NULL
GROUP BY dCountry
HAVING COUNT(*) >= 2
```

```
SELECT dCountry
FROM driver
WHERE dGender = 'M'
GROUP BY dCountry
HAVING COUNT(dDeathdate) >= 2
```