

## Correction du TD 8 - SQL : sous-requêtes simples

### 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, sInc, 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. Sous-requêtes et jointures

Donner les requêtes SQL qui renvoient les informations suivantes, tout d'abord sans sous-requête puis, lorsque c'est possible, avec une sous-requête à la place de la jointure (si ce n'est pas possible expliquer pourquoi) :

1. le nom des circuits sur lesquels il y a eu un grand prix en mars ou avril 2014

```
SELECT cName
FROM circuit NATURAL JOIN grandprix
WHERE gDate BETWEEN '2014-03-01' AND '2014-04-30'
```

```
SELECT cName
FROM circuit
WHERE circuitID IN
    (SELECT circuitID
     FROM grandprix
     WHERE gDate BETWEEN '2014-03-01' AND '2014-04-30')
```

2. même chose, mais en classant les circuits selon le nombre de tours lors du grand prix

```
SELECT cName
FROM circuit NATURAL JOIN grandprix
WHERE gDate BETWEEN '2014-03-01' AND '2014-04-30'
ORDER BY gLaps
```

On ne peut pas remplacer la jointure par une sous-requête car la requête affiche cName de la table circuit qui doit donc être dans la requête principale, et classer les réponses selon gLaps de la table grandprix qui doit donc aussi être dans la requête principale...

3. les noms et prénoms des pilotes qui ont actuellement un record du tour sur un circuit, par ordre alphabétique

```
SELECT DISTINCT dFirstname, dLastname
FROM driver JOIN circuit ON driverID = cDrivRec
ORDER BY dLastname, dFirstname
```

```
SELECT dFirstname, dLastname
FROM driver
WHERE driverID IN (SELECT DISTINCT cDrivRec
                  FROM circuit)
ORDER BY dLastname, dFirstname
```

Un même pilote peut détenir le record du tour de plusieurs circuits. Le SELECT DISTINCT permet de dédoubler la liste des identifiants des pilotes ayant un record, ce qui accélère la recherche d'appartenance dans la clause WHERE de la requête principale.

4. les pilotes (nom, prénom) nés la même année, et de même nationalité que Charles Pic

```
SELECT autre.dFirstname, autre.dLastname
FROM driver AS autre JOIN driver AS pic
      ON YEAR(autre.dBirthdate) = YEAR(pic.dBirthdate)
      AND autre.dCountry = pic.dCountry
WHERE (pic.dFirstname, pic.dLastname) = ('Charles', 'Pic')
```

Sans surprise, Charles Pic est né la même année et est de même nationalité que... Charles Pic ! Cela ne me semble pas faux dans cette requête. Cependant, si vous souhaitez répondre à la requête : « Quels sont les autres pilotes... », il faut ajouter dans la condition de jointure une troisième contrainte autre.driverID <> pic.driverID.  
L'écriture avec sous-requête est plus simple, car il n'est pas nécessaire de renommer les tables ni de préfixer les attributs :

```
SELECT dFirstname, dLastname
FROM driver
WHERE (YEAR(dBirthdate), dCountry) =
      (SELECT YEAR(dBirthdate), dCountry
       FROM driver
       WHERE (dFirstname, dLastname) = ('Charles', 'Pic'))
```

On peut écrire un « = » ici car Charles Pic a une seule nationalité et une seule année de naissance. Dans d'autres cas, on peut être amené à utiliser l'opérateur de listes « IN ».  
Comment pourrait-on éliminer Charles Pic des réponses ici ?

---

## Exercice 2. *Sous-requêtes dans la clause WHERE*

Donner les requêtes SQL qui renvoient les informations suivantes :

1. les pilotes de F1 (nom, prénom) qui n'ont pas pu rencontrer Ayrton Senna

Il s'agit des pilotes nés après la mort d'Ayrton Senna, et de ceux décédés avant sa naissance ; on utilise 2 sous-requêtes : une qui renvoie l'année de décès, et l'autre l'année de naissance d'Ayrton Senna.

```
SELECT dFirstname, dLastname
FROM driver
WHERE dBirthdate >
      (SELECT dDeathdate
       FROM driver
       WHERE (dFirstname, dLastname) = ('Ayrton', 'Senna'))
OR dDeathdate <
      (SELECT dBirthdate
       FROM driver
       WHERE (dFirstname, dLastname) = ('Ayrton', 'Senna'))
```

Remarque : on aurait pu écrire cette requête avec une auto-jointure... Mais comme la condition de jointure est une disjonction (un « OU ») d'inégalités sur des attributs qui ne sont pas indexés, la requête avec auto-jointure serait probablement moins efficace (en plus d'être moins facilement lisible).

2. le nom et le pays du circuit le plus long

```
SELECT cName, cCountry
FROM circuit
WHERE cLength = (SELECT MAX(cLength)
                 FROM circuit)
```

```
SELECT cName, cCountry
FROM circuit
WHERE cLength >= ALL (SELECT cLength
                     FROM circuit)
```

3. les pilotes de 2014 (nom, prénom) qui n'ont jamais terminé une course sur le podium en 2014

```
SELECT dFirstname, dLastname
FROM driver NATURAL JOIN racedriver
WHERE rSeason = 2014
      AND driverID NOT IN (SELECT DISTINCT driverID
                          FROM standings
                          WHERE sPos <= 3
                          AND gpID LIKE '2014%')
```

Remarque : une sous-requête n'est cependant pas nécessaire ici. Un pilote n'a pas terminé sur le podium si son meilleur résultat est plus loin que la 3<sup>e</sup> place.

```
SELECT dFirstname, dLastname
FROM driver NATURAL JOIN standings
WHERE gpID LIKE '2014%'
GROUP BY driverID
HAVING MIN(sPos) > 3
```

4. les grands prix (nom, pays) que Kobayashi n'a pas courus entre janvier et septembre 2014

```
SELECT gName, cCountry
FROM grandprix NATURAL JOIN circuit
WHERE gDate BETWEEN '2014-01-01' AND '2014-09-30'
AND gpID NOT IN (SELECT gpID
                  FROM driver NATURAL JOIN standings
                  NATURAL JOIN grandprix
                  WHERE gDate BETWEEN '2014-01-01' AND '2014-09-30'
                  AND dLastname = 'Kobayashi')
```

### Exercice 3. *Sous-requêtes dans la clause HAVING*

Donner les requêtes SQL qui renvoient les informations suivantes :

1. la nationalité la plus représentée parmi les pilotes de course en 2014

```
SELECT dCountry
FROM driver NATURAL JOIN racedriver
WHERE rSeason = 2014
GROUP BY dCountry
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
                        FROM driver NATURAL JOIN racedriver
                        WHERE rSeason = 2014
                        GROUP BY dCountry)
```

Remarque : on ne peut pas calculer en une seule fois MAX(COUNT(\*)) pour avoir le nombre de pilotes de la nationalité la plus représentée... pour cela, il faudrait passer par une table intermédiaire qui contient le nombre de pilotes de chaque nationalité, et calculer le MAX dans cette table ; cela suppose donc une sous-requête dans la clause FROM.

2. les pilotes (prénom, nom) qui ont terminé au moins 80 % des courses entre janvier et septembre 2014 (on considère la course terminée si le pilote n'a connu aucun incident de course)

```
SELECT dFirstname, dLastname
FROM driver NATURAL JOIN standings
      NATURAL JOIN grandprix
WHERE sInc IS NULL
AND gDate BETWEEN '2014-01-01' AND '2014-09-30'
GROUP BY driverID
HAVING COUNT(*) >= 0.8 *
      (SELECT COUNT(*)
       FROM grandprix
       WHERE gDate BETWEEN '2014-01-01' AND '2014-09-30')
```