

## Correction du TD 10 - SQL : sous-requêtes corrélées, division

### 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. Sous-requêtes corrélées

Donner les requêtes SQL qui renvoient les informations suivantes :

1. les pilotes (nom, prénom) dans l'ordre alphabétique, avec leur meilleur résultat (place), et le circuit sur lequel il a été réalisé (nom, pays) pour la saison 2014; si ce meilleur résultat a été réalisé sur plusieurs grands prix (eg. plusieurs victoires), donner tous les circuits (1 par ligne) dans l'ordre chronologique des grands prix

```
SELECT dLastName, dFirstName, sPos, cName, cCountry
FROM driver NATURAL JOIN standings
      NATURAL JOIN grandprix
      NATURAL JOIN circuit
WHERE YEAR(gDate) = 2014
      AND sPos = (SELECT MIN(sPos)
                  FROM standings
                  WHERE gpID LIKE '2014%'
                  AND driverID = driver.driverID)
ORDER BY dLastName, gDate
```

Pour chaque ligne de la table standings (ie. chaque résultat obtenu par un pilote particulier), la sous-requête calcule la meilleure place obtenue par ce pilote; le WHERE conserve cette ligne seulement si elle correspond au meilleur résultat de ce pilote.

C'est la version demandée dans cet exercice sur les sous-requêtes corrélées mais, comme souvent, lorsqu'il existe une version sans corrélation celle-ci est plus efficace !

On pouvait faire plus efficace ici en calculant, par une sous-requête statique, la liste des identi-

fians de pilotes et leur meilleur résultat. La requête principale conserve alors une ligne résultat seulement si le doublet (idpilote, meilleure place) appartient à la liste. On obtient la requête ci-dessous :

```
SELECT dLastName, dFirstName, sPos, cName, cCountry
FROM driver NATURAL JOIN standings
           NATURAL JOIN grandprix
           NATURAL JOIN circuit
WHERE YEAR(gDate) = 2014
      AND (driverID, sPos) IN (SELECT driverID, MIN(sPos)
                               FROM standings
                               WHERE gpID LIKE '2014%'
                               GROUP BY driverID)
ORDER BY dLastName, gDate
```

2. pour chaque pilote (nom, prénom), le nombre d'années depuis sa dernière victoire (en supposant la BD plus complète qu'elle n'est réellement...)

```
SELECT dLastName, dFirstName, YEAR(CURRENT_DATE()) -
      (SELECT MAX(YEAR(gDate))
       FROM standings NATURAL JOIN grandprix
       WHERE sPos = 1
        AND driverID = driver.driverID)
      AS 'n''a plus gagné depuis (années)'
FROM driver
```

Il est aussi possible de se passer de la sous-requête corrélée. Pour cela il faut construire, à partir de la table standings, une table contenant l'année de la dernière victoire de chaque pilote, et la joindre sur la table des pilotes.

```
SELECT dLastName, dFirstName, YEAR(CURRENT_DATE()) - annee_vict
      AS 'n''a plus gagné depuis (années)'
FROM driver NATURAL LEFT JOIN
      (SELECT driverID, MAX(YEAR(gDate)) AS annee_vict
       FROM standings NATURAL JOIN grandprix
       WHERE sPos = 1
       GROUP BY driverID) AS vict_recente
```

3. les pilotes (nom, prénom) qui n'ont jamais été au delà de la 10<sup>e</sup> place sur la grille de départ d'un grand prix en 2014 (en utilisant EXISTS)

```
SELECT dLastName, dFirstName
FROM driver NATURAL JOIN racedriver
WHERE rSeason = 2014
      AND NOT EXISTS (SELECT *
                      FROM standings
                      WHERE gpID LIKE '2014%'
                        AND sGrid > 10
                        AND driverID = driver.driverID)
ORDER BY dLastName
```

Donne le résultat pour les pilotes *engagés*, ce qui n'assure pas qu'ils aient couru un grand prix. En effet, avec cette solution, un pilote qui n'a pas encore couru est un pilote qui n'a jamais fini au delà de la 10<sup>e</sup> place.

```
SELECT DISTINCT dLastName, dFirstName
FROM driver NATURAL JOIN standings
WHERE gpID LIKE '2014%'
      AND NOT EXISTS (SELECT *
                      FROM standings
                      WHERE gpID LIKE '2014%'
                        AND sGrid > 10
                        AND driverID = driver.driverID)
ORDER BY dLastName
```

Donne le résultat pour les pilotes qui ont couru au moins un grand prix (et qui sont donc dans la table `standings`). Avec cette solution un pilote qui n'a pas encore couru ne fait pas partie du résultat. Par contre, la solution est plus coûteuse : la table `standings` est bien plus grande que la table `racedriver` (16 à 19 fois plus grande car il y a un résultat par course), et dans la jointure, pour chaque pilote il y a 1 correspondance par course courue, au lieu d'une correspondance par saison courue pour la requête précédente, ce qui fait plusieurs résultats par pilote (au lieu d'un seul) pour 2014 et oblige à faire un `SELECT DISTINCT`.

---

## Exercice 2. *Division*

Donner les requêtes SQL qui renvoient les informations suivantes :

1. les pilotes (nom, prénom) qui ont fini les 10 premiers grands prix de 2014 dans les points (= dans les 10 premiers)

C'est-à-dire les pilotes pour lesquels il n'existe pas un grand prix (dans les 10 premiers de 2014) qu'ils n'aient pas fini dans les 10 premiers.

```
SELECT dLastName, dFirstName
FROM driver
WHERE NOT EXISTS
  (SELECT *
   FROM grandprix
   WHERE grank <= 10
     AND YEAR(gDate) = 2014
     AND NOT EXISTS
       (SELECT *
        FROM standings
        WHERE sPos <= 10
          AND driverID = driver.driverID
          AND gpID = grandprix.gpID))
```

### Remarques :

- Quelle est la différence entre cette requête et la précédente ?  
Dans la précédente, on regarde les résultats à l'arrivée, et on s'assure qu'il n'y en a pas au delà de la 10<sup>e</sup> place. Cependant, cela ne garantit pas que le pilote a participé à tous les grands prix concernés ! S'il a participé à un seul grand prix et fini 8<sup>e</sup>, il est capturé.  
Dans cette requête, il faut avoir couru tous les grands prix concernés, et les avoir tous terminés dans les 10 premiers.
- Dans la sous-requête intermédiaire, il faut restreindre les grands prix explorés aux 10 premiers de 2014. Par contre, il n'est pas nécessaire de le préciser dans la 2<sup>e</sup> sous-requête. En effet, cette 2<sup>e</sup> sous-requête corrélée n'explore que les pilotes de la requête principale pour les grands prix de la requête intermédiaire. Le filtre sur les 10 premiers grands prix de 2014 est donc bien appliqué !

On peut remarquer que cette requête explore tous les pilotes de la BD. Les sous-requêtes corrélées sont donc exécutées de (trop) nombreuses fois pour rien. . . En effet, pour les pilotes qui ne sont pas engagés en 2014, la 2<sup>e</sup> sous-requête ne renvoie rien (car pas de participation à un grand prix de 2014 donc pas de résultat dans les 10 premières places), et la sous-requête intermédiaire renvoie donc la liste complète des grands prix concernés (auxquels le pilote n'a pas fini dans les 10 premiers); la requête principale élimine donc ces pilotes pour lesquels *il existe des grands prix auxquels ils n'ont pas fini dans les 10 premiers*. Mais tout ce travail est inutile car on savait à l'avance que ces pilotes ne remplissaient pas les critères recherchés.

Pour améliorer l'efficacité de cette requête, il faut donc limiter la recherche aux pilotes de courses de 2014, les seuls qui ont une chance de répondre aux critères !

```
SELECT dLastName, dFirstName
FROM driver NATURAL JOIN racedriver
WHERE rseason = 2014
  AND NOT EXISTS
    (SELECT *
     FROM grandprix
     WHERE grank <= 10
        AND YEAR(gDate) = 2014
        AND NOT EXISTS
          (SELECT *
           FROM standings
           WHERE sPos <= 10
              AND driverID = driver.driverID
              AND gpID = grandprix.gpID))
```

2. les équipes (nom) qui ont placé (au moins) un pilote sur le podium de chaque course en 2014

C'est-à-dire équipes pour lesquels il n'existe pas de grand prix de 2014 (dont le résultat est connu) pour lesquels un de leurs pilotes n'est pas dans les 3 premiers.

On se limite aux courses de 2014 dont le résultat est connu (les 14 premières, donc celles dont l'identifiant est entre 201401 et 201414), car aucune équipe n'a pu finir dans les 3 premières places des courses dont le résultat n'est pas renseigné !

```
SELECT DISTINCT tname
FROM team NATURAL JOIN racedriver
WHERE rSeason = 2014
  AND NOT EXISTS
    (SELECT *
     FROM grandprix
     WHERE gpID BETWEEN 201401 AND 201414
      AND NOT EXISTS
        (SELECT *
         FROM standings NATURAL JOIN racedriver
         WHERE sPos <= 3
          AND teamID = team.teamID
          AND gpID = grandprix.gpID))
```

Dans la 2<sup>e</sup> sous-requête, on explore les résultats d'une équipe lors d'un grand prix, et on espère (au moins) 1 résultat dans les 3 premières places (parmi les 2 résultats de l'équipe puisque chaque équipe peut engager 2 pilotes à chaque course).