

## Correction du TD 6 - SQL : calculs horizontaux et verticaux

### 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 des champs calculés (calculs horizontaux)*

Donner les requêtes SQL qui renvoient les informations suivantes :

1. pour chaque grand prix de la saison 2014, le nom du circuit, le nombre de tours et la longueur de la course (en km), dans l'ordre chronologique

```
SELECT cName, gLaps, gLaps * cLength AS 'longueur course'
FROM grandprix NATURAL JOIN circuit
WHERE YEAR(gDate) = 2014
ORDER BY gDate
```

2. pour chaque grand prix de la saison 2014 dont le résultat est connu (dans l'ordre chronologique), le nom du grand prix et le pilote vainqueur. Le pilote doit être indiqué au format suivant : l'initiale du prénom, suivi d'un point et d'un espace, et le nom (par exemple, pour Nico Rosberg il faut afficher « N. Rosberg »). Les colonnes doivent être renommées en « grand prix » et « vainqueur ».

```
SELECT gName AS 'grand prix',
       CONCAT(SUBSTRING(dFirstName,1,1), '. ', dLastName) AS vainqueur
FROM grandprix NATURAL JOIN standings
           NATURAL JOIN driver
WHERE sPos = 1
      AND YEAR(gDate) = 2014
ORDER BY gDate
```

3. les nom, prénom, date de naissance et âge des pilotes, du plus vieux au plus jeune. Le calcul d'âge peut être réalisé en calculant le nombre de jours séparant la date actuelle et la date de naissance (par la fonction DATEDIFF()), en divisant le résultat par 365.2425, et en gardant la partie entière. La date actuelle peut être écrite « en dur », ou récupérée en utilisant la fonction CURRENT\_DATE().

Version avec la date « en dur » (pour le mardi 17 octobre 2017) :

```
SELECT dFirstName, dLastName, dBirthdate,  
        FLOOR(DATEDIFF('2017-10-17', dBirthdate) / 365.2425) AS âge  
FROM driver  
ORDER BY FLOOR(DATEDIFF('2017-10-17', dBirthdate) / 365.2425) DESC
```

Une autre version, qui utilise CURRENT\_DATE() :

```
SELECT dFirstName, dLastName, dBirthdate,  
        FLOOR(DATEDIFF(CURRENT_DATE(), dBirthdate) / 365.2425) AS  
        âge  
FROM driver  
ORDER BY FLOOR(DATEDIFF(CURRENT_DATE(), dBirthdate) / 365.2425)  
        DESC
```

Plutôt que de réécrire la formule dans la clause ORDER BY, MySQL permet de réutiliser l'alias de colonne dans certains cas. Ici, on peut écrire :

```
SELECT dFirstName, dLastName, dBirthdate,  
        FLOOR(DATEDIFF(CURRENT_DATE(), dBirthdate) / 365.2425) AS  
        âge  
FROM driver  
ORDER BY âge DESC
```

Enfin, on peut aussi remarquer que l'âge décroît lorsque la date de naissance croît; on peut donc simplifier la clause ORDER BY en classant selon la date de naissance croissante, plutôt que l'âge décroissant. C'est d'ailleurs un meilleur choix car il permet de classer correctement les pilotes de même âge !

```
SELECT dFirstName, dLastName, dBirthdate,  
        FLOOR(DATEDIFF(CURRENT_DATE(), dBirthdate) / 365.2425) AS  
        âge  
FROM driver  
ORDER BY dBirthdate
```

4. même requête, mais pour les pilotes vivants uniquement

```
SELECT dFirstName, dLastName, dBirthdate,  
        FLOOR(DATEDIFF(CURRENT_DATE(), dBirthdate) / 365.2425) AS âge  
FROM driver  
WHERE dDeathdate IS NULL  
ORDER BY dBirthdate
```

5. les nom, prénom et durée de vie des pilotes décédés, par ordre alphabétique

```
SELECT dFirstName, dLastName,  
        FLOOR(DATEDIFF(dDeathDate, dBirthdate) / 365.2425) AS 'durée de  
        vie'  
FROM driver  
WHERE dDeathdate IS NOT NULL  
ORDER BY dLastName
```

Que se passe-t-il si on calcule la durée de vie pour tous les pilotes (même ceux encore vivants) ?

Lorsque le calcul est impossible (e.g. une opérande vaut NULL), le résultat de ce calcul vaut NULL. Une durée de vie serait donc affichée pour les pilotes décédés, et NULL pour les pilotes vivants.

6. les nom, prénom et durée de vie de tous les pilotes, par ordre alphabétique ; utiliser pour cela la fonction COALESCE (calcul de la durée de vie lorsque c'est possible, de l'âge sinon)

Une première solution consiste à faire le calcul de la durée de vie et, si ce calcul renvoie NULL (car dDeathDate vaut NULL), le remplacer par le calcul d'âge.

```
SELECT dFirstName, dLastName,  
       COALESCE(FLOOR(DATEDIFF(dDeathDate, dBirthdate) / 365.2425),  
                FLOOR(DATEDIFF(CURRENT_DATE(), dBirthdate) / 365.2425))  
       AS 'durée de vie'  
FROM driver  
ORDER BY dLastName
```

On peut aussi remarquer que la différence entre les deux formules est la date de référence (date de décès ou date actuelle), et utiliser COALESCE pour obtenir la date actuelle lorsque la date de décès vaut NULL.

```
SELECT dFirstName, dLastName,  
       FLOOR(DATEDIFF(COALESCE(dDeathDate, CURRENT_DATE()), dBirthdate)  
              / 365.2425) AS 'durée de vie'  
FROM driver  
ORDER BY dLastName
```

---

## Exercice 2. *Requêtes avec calculs verticaux (fonctions d'agrégat)*

Donner les requêtes SQL qui renvoient les informations suivantes :

1. pour les grands prix de 2014, le nombre de tours moyen à parcourir sur un circuit, et le nombre de tours total à effectuer dans la saison.

```
SELECT AVG(gLaps) AS 'nb tours moyen', SUM(gLaps) AS 'nb tours total'  
FROM grandprix  
WHERE YEAR(gDate) = 2014
```

2. pour les grands prix de 2014, la plus petite distance d'un grand prix (en km).

```
SELECT MIN(gLaps * cLength) AS 'distance minimale'  
FROM grandprix NATURAL JOIN circuit  
WHERE YEAR(gDate) = 2014
```

3. Peut-on afficher en plus (de façon simple) le circuit correspondant ?

Non, car l'ajout de `cName` dans le `SELECT` (`SELECT cName, MIN(gLaps * cLength) AS 'distance minimale'`) n'est pas valide. Sous MySQL, cette écriture n'est pas refusée, mais affiche un nom de circuit *au hasard* (souvent le premier de la liste sur la machine, ici « *Albert Park* ») au lieu du circuit correspondant à la distance minimale (ici « *Circuit de Monaco* »). On pourra faire cela plus tard, en utilisant une sous-requête.

4. l'année de naissance du plus vieux et celle du plus jeune des pilotes de course la saison 2014.

```
SELECT MIN(YEAR(dBirthdate)) AS vieux, MAX(YEAR(dBirthdate)) AS jeune
FROM driver NATURAL JOIN racedriver
WHERE rSeason = 2014
```

5. le nombre de courses de la saison 2014

```
SELECT COUNT(*) AS 'nb courses 2014'
FROM grandprix
WHERE YEAR(gDate) = 2014
```

6. le nombre de pilotes nés avant 1950 encore vivants

```
SELECT COUNT(*) AS 'nb pilotes'
FROM driver
WHERE YEAR(dBirthdate) < 1950 AND dDeathdate IS NULL
```

7. le nombre de pilotes décédés (avec 2 types de COUNT)

```
SELECT COUNT(dDeathdate) as 'nb pilotes'
FROM driver
```

```
SELECT COUNT(*) as 'nb pilotes'
FROM driver
WHERE dDeathdate IS NOT NULL
```

8. le nombre de pilotes qui ont gagné (au moins) 1 grand prix en 2014

```
SELECT COUNT(DISTINCT driverID) AS 'nb vainqueurs'
FROM driver NATURAL JOIN standings
      NATURAL JOIN grandprix
WHERE sPos = 1 AND YEAR(gDate) = 2014
```

Mais attention à la requête ci-dessous, qui ne renvoie pas le bon résultat :

```
SELECT COUNT(DISTINCT driverID) AS 'nb vainqueurs'
FROM driver NATURAL JOIN standings
      NATURAL JOIN racedriver
WHERE sPos = 1 AND rSeason = 2014
```

Elle donne le nombre de pilotes de course engagés en 2014, qui ont eu au moins 1 victoire en F1 dans leur vie...