

Correction du TD 4 - SQL : jointures - partie 1

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. Une requête avec jointure, construite par étapes

On souhaite obtenir les informations suivantes : « Les noms et prénom des pilotes ayant réalisé un record du tour sur un circuit (avec nom et ville du circuit), ainsi que le temps et année du record, pour les circuits de plus de 5km, classés par nom du pilote, par année du record (du plus récent au plus ancien), puis selon la longueur du circuit (plus long d'abord). »

Pour vous permettre de voir les 4 étapes de construction du résultat (cf. cours), cette requête est décomposée en 4 questions successives. En utilisant la syntaxe SQL92 (jointures explicites), donner les requêtes SQL qui renvoient les informations suivantes :

1. les circuits avec les pilotes ayant réalisé les records du tour (toutes les colonnes)

```
SELECT *  
FROM driver JOIN circuit ON driverID = cDrivRec
```

2. même chose, mais seulement pour les circuits de plus de 5km

```
SELECT *  
FROM driver JOIN circuit ON driverID = cDrivRec  
WHERE cLength >= 5
```

3. même chose, mais en ordonnant les résultats selon le nom du pilote, l'année du record (du plus récent au plus ancien), puis selon la longueur du circuit (plus long d'abord)

```
SELECT *  
FROM driver JOIN circuit ON driverID = cDrivRec  
WHERE cLength >= 5  
ORDER BY dLastName, cYearRec DESC, cLength DESC
```

4. même chose, mais en affichant seulement les nom et prénom du pilote, nom, ville et longueur du circuit, et enfin année et temps du record

```
SELECT dLastName, dFirstName, cName, cCity, cLength, cYearRec, cLapRec
FROM driver JOIN circuit ON driverID = cDrivRec
WHERE cLength >= 5
ORDER BY dLastName, cYearRec DESC, cLength DESC
```

Exercice 2. *Requêtes simples avec une jointure*

En utilisant obligatoirement la syntaxe SQL92 (jointures explicites), donner les requêtes SQL qui renvoient les informations suivantes.

Pour chaque question, donner plusieurs versions : en utilisant JOIN...ON puis, lorsque c'est possible, en utilisant NATURAL JOIN et JOIN...USING. Lorsque ce n'est pas possible, indiquer pourquoi.

1. les grands prix de la saison 2014 (nom du circuit, date, longueur de la piste, nombre de tours) dans l'ordre chronologique.

En utilisant JOIN ON, il faut préfixer les attributs communs des 2 tables :

```
SELECT cName, DATE(gDate), cLength, gLaps
FROM grandprix JOIN circuit ON grandprix.circuitID = circuit.
    circuitID
WHERE YEAR(gDate) = 2014
ORDER BY gDate
```

Une ré-écriture « classique » plus condensée, mais pas forcément plus claire...

```
SELECT cName, DATE(gDate), cLength, gLaps
FROM grandprix g JOIN circuit c ON g.circuitID = c.circuitID
WHERE YEAR(gDate) = 2014
ORDER BY gDate
```

Avec une jointure naturelle (puisque'on fait une équijointure sur le seul attribut commun), version à préférer dans ce cas :

```
SELECT cName, DATE(gDate), cLength, gLaps
FROM grandprix NATURAL JOIN circuit
WHERE YEAR(gDate) = 2014
ORDER BY gDate
```

Avec JOIN USING, indique clairement l'attribut de jointure en simplifiant l'écriture par rapport à JOIN ON :

```
SELECT cName, DATE(gDate), cLength, gLaps
FROM grandprix JOIN circuit USING(circuitID)
WHERE YEAR(gDate) = 2014
ORDER BY gDate
```

2. Les pilotes de course (nom, prénom, no de voiture) de la saison 2014, classés selon le no de voiture

```
SELECT dLastName, dFirstName, rDriverNb
FROM driver JOIN racedriver ON driver.driverID = racedriver.driverID
WHERE rSeason = 2014
ORDER BY rDriverNb
```

```
SELECT dLastName, dFirstName, rDriverNb
FROM driver NATURAL JOIN racedriver
WHERE rSeason = 2014
ORDER BY rDriverNb
```

```
SELECT dLastName, dFirstName, rDriverNb
FROM driver JOIN racedriver USING(driverID)
WHERE rSeason = 2014
ORDER BY rDriverNb
```

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

```
SELECT DISTINCT dLastName, dFirstName
FROM driver JOIN circuit ON driverID = cDrvRec
ORDER BY dLastName
```

Il s'agit d'une jointure entre clé étrangère et clé primaire associée, et la projection ne conserve que des attributs issus de la table contenant la clé primaire ; la jointure est donc susceptible de générer des doublons dans cette requête. Ces doublons ne sont pas souhaitables : ne pas oublier DISTINCT.

La condition de jointure est une égalité entre des attributs de nom différent : on ne peut donc pas utiliser NATURAL JOIN ni JOIN USING.

Exercice 3. *Requêtes avec plusieurs jointures*

En utilisant obligatoirement la syntaxe SQL92 (jointures explicites), donner les requêtes SQL qui renvoient les informations suivantes.

Pour chaque question, donner plusieurs versions : en utilisant JOIN . . . ON puis, lorsque c'est possible, en utilisant NATURAL JOIN et JOIN . . . USING . Lorsque ce n'est pas possible, indiquer pourquoi.

1. les pilotes (nom, prénom, numéro de voiture), et leur équipe (nom) pour la saison 2014, classés par équipe, puis par numéro de voiture

```
SELECT dLastName, dFirstName, rDriverNb, tName
FROM driver JOIN racedriver ON driver.driverID = racedriver.driverID
      JOIN team ON racedriver.teamID = team.teamID
WHERE rSeason = 2014
ORDER BY tName, rDriverNb
```

```
SELECT dLastName, dFirstName, rDriverNb, tName
FROM driver NATURAL JOIN racedriver
      NATURAL JOIN team
WHERE rSeason = 2014
ORDER BY tName, rDriverNb
```

```

SELECT dLastName, dFirstName, rDriverNb, tName
FROM driver JOIN racedriver USING(driverID)
           JOIN team USING(teamID)
WHERE rSeason = 2014
ORDER BY tName, rDriverNb

```

2. pour chaque grand prix de 2014 (dont les résultats sont renseignés dans la BD) le pays du circuit, le nom du grand prix, le nom du pilote vainqueur et le nom de son équipe (par ordre chronologique des grands prix)

```

SELECT cCountry, gName, dLastName, tName
FROM driver JOIN racedriver ON driver.driverID = racedriver.driverID
           JOIN team ON racedriver.teamID = team.teamID
           JOIN standings ON driver.driverID = standings.driverID
           JOIN grandprix ON standings.gpID = grandprix.gpID
           JOIN circuit ON grandprix.circuitID = circuit.circuitID
WHERE rSeason = 2014
       AND sPos = 1
ORDER BY gDate

```

```

SELECT cCountry, gName, dLastName, tName
FROM driver NATURAL JOIN racedriver
           NATURAL JOIN team
           NATURAL JOIN standings
           NATURAL JOIN grandprix
           NATURAL JOIN circuit
WHERE rSeason = 2014
       AND sPos = 1
ORDER BY gDate

```

```

SELECT cCountry, gName, dLastName, tName
FROM driver JOIN racedriver USING(driverID)
           JOIN team USING(teamID)
           JOIN standings USING(driverID)
           JOIN grandprix USING(gpID)
           JOIN circuit USING(circuitID)
WHERE rSeason = 2014
       AND sPos = 1
ORDER BY gDate

```

3. pour chaque grand prix de 2014, le nom du grand prix, la ville du circuit, le temps du record du tour et les prénom et nom du pilote recordman, dans l'ordre alphabétique des villes

```

SELECT cName, cCity, cLapRec, dFirstName, dLastName
FROM grandprix JOIN circuit ON grandprix.circuitID = circuit.circuitID
           JOIN driver ON cDrivRec = driverID
ORDER BY cCity

```

La jointure entre circuit et driver étant effectuée sur des attributs de nom différent, il n'est pas possible d'utiliser NATURAL JOIN, ni JOIN USING.

```
SELECT cName, cCity, cLapRec, dFirstName, dLastName  
FROM grandprix NATURAL JOIN circuit  
                JOIN driver ON cDrvRec = driverID  
ORDER BY cCity
```

```
SELECT cName, cCity, cLapRec, dFirstName, dLastName  
FROM grandprix JOIN circuit USING(circuitID)  
                JOIN driver ON cDrvRec = driverID  
ORDER BY cCity
```