
Correction du TD 12 - SQL : vues

Base de données formula1

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

- **circuit**(circuitID, cName, cCity, cCountry, cLength, cLapRec, #cDrivRec, 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)
- **tesdriver**(#teamID, #driverID, tSeason)

Cette base est utilisée pour tous les exercices qui suivent.

Les requêtes de ce TD modifiant la base de données, elle ne pourront pas être testées sur l'interface habituelle, qui utilise une BD commune et ne permet que les requêtes de consultation.

Exercice 1. *Manipulation préparatoire*

Pour ce TD, comme pour le précédent, vous devez travailler sur une copie personnelle de la base de données « formula 1 », via l'interface PHMyAdmin. Votre base ayant été modifiée lors du TD précédent, il faut la réinstaller.

- connectez-vous à votre interface de gestion PhPMyAdmin, à l'adresse <http://pedago.uhb.fr/BasesMySQL/>,
- choisissez la base de données utilisée lors du TD11,
- supprimez toutes les tables de cette BD, via l'interface PhPMyAdmin,
- récupérez un dump de la base « formula 1 » dans l'espace-cours sur cursus,
- importez la base « formula 1 » dans votre BD, via l'interface PhPMyAdmin.

Exercice 2. *Vues simples*

Dans cet exercice, les vues sont utilisées pour restreindre le contenu de certaines tables.

Question 2.1 : *Création d'une vue sur les résultats de la saison 2013*

Donner et exécuter la requête SQL qui permet de créer la vue `v_standings2013` correspondant à la table `standings` restreinte aux résultats de la saison 2013. Attention, utiliser seulement la table `standings` dans la requête de définition ; ne pas faire de jointure !

```
CREATE VIEW v_standings2013 AS
SELECT *
FROM standings
WHERE gpID BETWEEN 201300 AND 201399
```

Vérifier la création de la vue dans l'interface PHPMyAdmin.

Exécuter ensuite une requête qui compte le nombre de lignes de la table `standings` et une autre qui compte le nombre de lignes de la vue `v_standings2013`.

```
SELECT COUNT(*)
FROM standings

SELECT COUNT(*)
FROM v_standings2013
```

Question 2.2 : Utilisation de la vue

En utilisant la vue `v_standings2013`, donner la requête SQL qui affiche, pour chaque pilote de la saison 2013 (prénom, nom), le nombre de grands prix courus, et la somme des points obtenus, du meilleur au moins bon.

```
SELECT dFirstName, dLastName, COUNT(*) AS 'nb_gp', SUM(sPoints) AS 'nb_pts'
FROM driver NATURAL JOIN v_standings2013
GROUP BY driverID
ORDER BY SUM(sPoints) DESC
```

Question 2.3 : Création d'une vue sur les résultats de la saison 2014

Comme précédemment, créer la vue `v_standings2014` correspondant à la table `standings` restreinte aux résultats de la saison 2014, mais en ajoutant la directive `WITH CHECK OPTION`.

```
CREATE VIEW v_standings2014 AS
SELECT *
FROM standings
WHERE gpID BETWEEN 201400 AND 201499
WITH CHECK OPTION
```

Question 2.4 : Ajout via une vue simple

Via la vue `v_standings2014`, ajouter les 3 lignes de résultat suivantes (podium du grand prix d'abu dhabi 2014).

```
(9, 201419, 2, 1, NULL, 25, 55),
(16, 201419, 4, 2, NULL, 18, 55),
(3, 201419, 3, 3, NULL, 15, 55)
```

```
INSERT INTO v_standings2014 VALUES
(9, 201419, 2, 1, NULL, 25, 55),
(16, 201419, 4, 2, NULL, 18, 55),
(3, 201419, 3, 3, NULL, 15, 55)
```

Puis vérifier leur présence dans la table `standings`.

```
SELECT *
FROM standings
WHERE gpID = 201419
```

Via la vue `v_standings2013` (attention, 2013!), ajouter les 3 lignes de résultat suivantes (places 4 à 6 du grand prix d'abu dhabi 2014).

```
(19, 201419, 20, 4, NULL, 12, 55),  
(4, 201419, 6, 5, NULL, 10, 55),  
(10, 201419, 12, 6, NULL, 8, 55)
```

```
INSERT INTO v_standings2013 VALUES  
(19, 201419, 20, 4, NULL, 12, 55),  
(4, 201419, 6, 5, NULL, 10, 55),  
(10, 201419, 12, 6, NULL, 8, 55)
```

Puis vérifier leur présence dans la table `standings`. N'est-il pas anormal de pouvoir ajouter des résultats de 2014 via la vue des résultats 2013 ?

Corriger la définition de la vue `v_standings2013` par la requête suivante :

```
CREATE OR REPLACE VIEW v_standings2013 AS  
SELECT *  
FROM standings  
WHERE gpID BETWEEN 201300 AND 201399  
WITH CHECK OPTION
```

Essayez ensuite d'ajouter les lignes suivantes (autres pilotes dans les points) via la vue corrigée `v_standings2013`.

```
(17, 201419, 11, 7, NULL, 6, 55),  
(23, 201419, 19, 8, NULL, 4, 55),  
(1, 201419, 8, 9, NULL, 2, 55),  
(18, 201419, 7, 10, NULL, 1, 55)
```

```
INSERT INTO v_standings2013 VALUES  
(17, 201419, 11, 7, NULL, 6, 55),  
(23, 201419, 19, 8, NULL, 4, 55),  
(1, 201419, 8, 9, NULL, 2, 55),  
(18, 201419, 7, 10, NULL, 1, 55)
```

Vérifier que l'ajout est refusé par MySQL avec le message : `CHECK OPTION failed`.

Question 2.5 : *Suppression via une vue simple*

Vérifier le nombre de lignes de la table `standings`.

Via la vue `v_standings2014`, essayez maintenant de supprimer les résultats de Heikki Kovalainen (en utilisant son identifiant pilote 76) lors de la saison 2013.

```
DELETE FROM v_standings2014  
WHERE driverID = 76  
AND gpID BETWEEN 201300 AND 201399
```

Vérifier qu'aucune ligne n'a été supprimée de la table `standings`, et expliquer pourquoi.

La vue ne contient pas de résultats pour ce pilote en 2013 : aucun résultat ne peut donc être supprimé (à cause de ma clause `WITH CHECK OPTION`).

Faites la même suppression, mais via la vue `v_standings2013`, et vérifier que la suppression s'est bien répercutée dans la table `standings`.

```
DELETE FROM v_standings2013  
WHERE driverID = 76
```

Exercice 3. Vues complexes

Question 3.1 : Vue des pilotes de 2014

Créer la vue `v_driver2014`, avec toutes les colonnes `driverID`, `dFirstName`, `dLastName`, `dBirthdate` de la table `driver`, ainsi que les colonnes `teamID` et `tName` de la table `team`, et qui contient tous les pilotes de course de la saison 2014.

```
CREATE VIEW v_driver2014 AS
SELECT driverID, dFirstName, dLastName, dBirthdate, teamID, tName
FROM driver NATURAL JOIN racedriver
      NATURAL JOIN team
WHERE rSeason = 2014
```

Afficher le contenu de la vue, et noter le nombre de lignes qu'elle contient.

```
SELECT *
FROM v_driver2014

SELECT COUNT(*)
FROM v_driver2014
```

Question 3.2 : Vérifier qu'une vue est reconstruite à chaque utilisation

Le pilote Will Stevens a couru seulement le grand prix d'Abu Dhabi en 2014. Comme les résultats s'arrêtent avant ce grand prix, il n'est pas référencé dans la table `racedriver` comme un pilote de course de 2014. Lier ce pilote à son équipe en 2014, et définir son numéro de voiture, par la requête suivante :

```
INSERT INTO racedriver (teamID, driverID, rSeason, rDriverNb) VALUES
(11, (SELECT driverID FROM driver WHERE (dFirstName, dLastName) = ('Will',
'Stevens'))), 2014, 46)
```

Afficher à nouveau le contenu de la vue `v_driver2014`, et noter le nombre de lignes qu'elle contient. Cela montre que, contrairement aux vues matérialisées, les vues sont recalculées à chaque utilisation.

Question 3.3 : Résultats de 2014

En utilisant les vues `v_driver2014` et `v_standings2014`, créer la vue `v_res2014`, qui contient les identifiant, prénom et nom des pilotes de 2014, avec pour chacun l'identifiant et le nom de son équipe, le nombre de courses courues, et le total des points marqués.

```
CREATE VIEW v_res2014 AS
SELECT driverID, dFirstName, dLastName, teamID, tName,
      COUNT(*) AS nb_gp, SUM(sPoints) AS nb_pts
FROM v_driver2014 NATURAL JOIN v_standings2014
GROUP BY driverID, teamID
```

Noter que le groupement selon `teamID` n'est pas indispensable dans la requête d'interrogation `SELECT` car en 2014 chaque pilote n'est engagé que dans une seule équipe, mais qu'il est nécessaire pour la définition de la vue car le SGBD vérifie que les colonnes du `SELECT` sont en dépendance fonctionnelle des colonnes de groupement !

Afficher ces résultats (le contenu de la vue `v_res2014`), classés du meilleur pilote au moins bon.

```
SELECT *
FROM v_res2014
ORDER BY nb_pts DESC
```

Question 3.4 : Vérifier qu'une vue est reconstruite à chaque utilisation

Noter que certains pilotes ont couru 15 courses, d'autres 14 seulement. Il s'agit des résultats partiels du grand prix d'Abu Dhabi rajoutés précédemment. Supprimer ces résultats via la vue v_standings2014 (les résultats du grand prix d'identifiant 201419).

```
DELETE FROM v_standings2014
WHERE gpID = 201419
```

Consultez à nouveau le contenu de la vue v_res2014, classé du meilleur pilote au moins bon.

```
SELECT *
FROM v_res2014
ORDER BY nb_pts DESC
```

Question 3.5 : Même requête sans les vues

Donner la requête SQL qui affiche la même chose, mais sans utiliser les vues.

```
SELECT driverID, dFirstName, dLastName, teamID, tName,
       COUNT(*) AS nb_gp, SUM(sPoints) AS nb_pts
FROM driver NATURAL JOIN racedriver
       NATURAL JOIN team
       NATURAL JOIN standings
WHERE rSeason = 2014
      AND gpID BETWEEN 201400 AND 201499
GROUP BY driverID, teamID
ORDER BY nb_pts DESC
```

Question 3.6 : Simplification de requête grâce aux vues

En utilisant la vue v_res2014, puis sans utiliser de vues, donner des requêtes SQL qui affichent le nombre moyen de points marqués par grand prix pour chaque pilote (prénom, nom) de la saison 2014, par moyenne décroissante.

```
SELECT dFirstName, dLastName, nb_pts/nb_gp AS moy_pts_par_gp
FROM v_res2014
ORDER BY nb_pts/nb_gp DESC
```

Pour obtenir la même chose sans utiliser la vue, on peut dans un premier temps remplacer l'appel à la vue par sa requête de définition :

```
SELECT dFirstName, dLastName, nb_pts/nb_gp AS moy_pts_par_gp
FROM (SELECT dFirstName, dLastName,
            COUNT(*) AS nb_gp, SUM(sPoints) AS nb_pts
      FROM driver NATURAL JOIN standings
      WHERE gpID BETWEEN 201400 AND 201499
      GROUP BY driverID) AS res2014
ORDER BY nb_pts/nb_gp DESC
```

Mais il est préférable de simplifier la requête obtenue !

```
SELECT dFirstName, dLastName, moy_pts_par_gp
FROM (SELECT dFirstName, dLastName,
             AVG(sPoints) AS moy_pts_par_gp
       FROM driver NATURAL JOIN standings
       WHERE gpId BETWEEN 201400 AND 201499
       GROUP BY driverID) AS res2014
ORDER BY moy_pts_par_gp DESC
```