

---

## Correction du TD 11 - SQL : modification des données

---

### 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, vous devez travailler sur une copie personnelle de la base de données « formula 1 ».

- connectez-vous à votre interface de gestion PhPMyAdmin, à l'adresse <http://pedago.uhb.fr/BasesMySQL/>,
- choisissez la base de données utilisée lors du TD2,
- vérifiez qu'elle contient bien 7 tables (c'est le cas si vous avez terminé le TD2 !),
- si c'est bien le cas, passez à l'exercice suivant.

Sinon, vous devez recréer la base de données « formula 1 » de la façon suivante :

- supprimez les tables créées lors du TD2 s'il en reste dans votre 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. *Requêtes SQL d'ajout*<sup>1</sup>

Le calendrier<sup>2</sup> (prévisionnel) 2015 du championnat du monde de formula 1 est reproduit dans la Table 1 (ci-dessous).

Notez que *TBA* signifie « to be announced » et \* est commentée par « Provisional ».

---

1. Ce sujet a été écrit fin 2014... des informations n'étaient pas connues à ce moment-là, et n'ont pas été ajoutées pour conserver l'esprit du sujet, et la particularité de certains exercices !

2. Il est issu du site web officiel de la F1.

01	Australia (Melbourne)	13 - 15 Mar
02	Malaysia (Kuala Lumpur)	27 - 29 Mar
03	China (Shanghai)	10 - 12 Apr
04	Bahrain (Sakhir)	17 - 19 Apr
05	Korea (TBA) *	01 - 03 May
06	Spain (Catalunya)	08 - 10 May
07	Monaco (Monte Carlo)	22 - 24 May
08	Canada (Montréal)	05 - 07 Jun
09	Austria (Spielberg)	19 - 21 Jun
10	Great Britain (Silverstone)	03 - 05 Jul
11	Germany (TBA)	17 - 19 Jul
12	Hungary (Budapest)	24 - 26 Jul
13	Belgium (Spa-Francorchamps)	21 - 23 Aug
14	Italy (Monza)	04 - 06 Sep
15	Singapore (Singapore)	18 - 20 Sep
16	Japan (Suzuka)	25 - 27 Sep
17	Russia (Sochi)	09 - 11 Oct
18	United States (Austin)	23 - 25 Oct
19	Mexico (Mexico City)	30 Oct - 01 Nov
20	Brazil (São Paulo)	13 - 15 Nov
21	Abu Dhabi (Yas Marina)	27 - 29 Nov

TABLE 1 – Calendrier des grands prix de la saison 2015

### Question 2.1 : Ajout d'un grand prix

Donnez la requête SQL qui permet d'ajouter dans la base le grand prix d'Australie 2015.

On suppose connaître les informations ci-dessous <sup>3</sup> :

- l'identifiant est composé de l'année et du numéro d'ordre,
- le code du circuit est 1,
- le nom du grand prix est « ROLEX AUSTRALIAN GRAND PRIX » (comme en 2014),
- le nombre de tours à parcourir est 58 (comme en 2014),
- la course aura lieu le 15 mars à 17h (heure de 2014).

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201501, 'ROLEX AUSTRALIAN GRAND PRIX', 1, '2015-03-15 17:00:00', 58, 1)
```

### Question 2.2 : Ajout de plusieurs grands prix

Donnez la requête SQL (une seule) qui permet d'ajouter dans la base les grands prix de Malaisie, de Chine et de Bahrain 2015.

Les informations à ajouter, obtenues comme dans la question précédente sont :

gpID	gName	circuitID	gDate	gLaps	gRank
201502	'PETRONAS MALAYSIA GRAND PRIX'	2	'2015-03-29 16:00:00'	56	2
201503	'UBS CHINESE GRAND PRIX'	4	'2015-04-12 15:00:00'	56	3
201504	'GULF AIR BAHRAIN GRAND PRIX'	3	'2015-04-19 18:00:00'	57	4

3. En cas d'absence d'information officielle, on utilise les informations de 2014 pour cet exercice.

Notez l'inversion des grands prix de Chine et de Bahrain par rapport à 2014 ; l'identifiant du circuit n'est pas le numéro d'ordre !

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201502, 'PETRONAS MALAYSIA GRAND PRIX', 2, '2015-03-29 16:00:00', 56, 2),
(201503, 'UBS CHINESE GRAND PRIX', 4, '2015-04-12 15:00:00', 56, 3),
(201504, 'GULF AIR BAHRAIN GRAND PRIX', 3, '2015-04-19 18:00:00', 57, 4)
```

### Question 2.3 : Ajout d'un grand prix

On souhaite ajouter le grand prix de Corée. Si c'est possible, donner la requête qui permet de le faire, sinon expliquez le problème qui empêche l'ajout, et une solution qui permettrait de le faire tout de même.

Le circuit sur lequel aura lieu le grand prix de Corée n'est pas encore déterminé (*to be announced*). Il manque donc de nombreuses informations : identifiant du circuit, nombre de tours. De plus, il n'y a pas eu de grand prix de Corée en 2014 : on ne peut récupérer le nom *commercial* du grand prix, ni l'heure de l'épreuve.

En autorisant la valeur NULL dans `circuitID` et `gLaps` (à faire via PHPMyAdmin), en mettant un nom générique, et en donnant la date sans l'heure, il serait possible d'ajouter cette épreuve au calendrier des grands prix avec la requête suivante (qui n'était pas demandée).

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201505, 'KOREAN GRAND PRIX', NULL, '2015-03-03', NULL, 5)
```

Même question pour le grand prix du Mexique.

La situation est différente. Le circuit est connu, mais n'est pas déjà présent dans la BD car il n'a pas été utilisé lors des saisons enregistrées dans la BD. En effet, la page [http://fr.wikipedia.org/wiki/Grand\\_Prix\\_automobile\\_du\\_Mexique](http://fr.wikipedia.org/wiki/Grand_Prix_automobile_du_Mexique), nous apprend que le dernier grand prix du mexique s'est déroulé en 1992. Le problème ici est donc le respect de la contrainte d'intégrité référentielle : on ne peut pas faire référence à l'identifiant du circuit du mexique tant que celui-ci n'est pas dans la table `circuit`.

Pour pouvoir ajouter le grand prix du mexique, l'idéal est donc de commencer par ajouter le circuit de Mexico, puis le grand prix (ce qui n'était pas demandé dans le TD !) Pour l'ajout du circuit, on ne précise pas l'identifiant dans la requête : il sera ajouté automatiquement par la machine (par auto-incrément).

Remarque : cet identifiant sera 20 si aucun circuit n'est ajouté avant.

Les autres informations sur ce circuit sont :

- nom : Autodromo Hermanos Rodriguez,
- ville : Mexico,
- pays : Mexico,
- record : établi par Gerard Berger en 1992, avec un temps de 1:17.711.

Il nous faut donc l'identifiant du pilote Gerard Berger... mais celui-ci n'est pas dans la table des pilotes. Il faut donc commencer par l'ajouter.

```
INSERT INTO driver (dFirstName, dLastName, dBirthdate, dCountry) VALUES
('Gerard', 'Berger', '1959-08-27', 'Austria')
```

Notez qu'on n'a pas donné l'identifiant (défini par le SGBD par auto-incrément à 74), ni la date de décès et le genre (utilisation des valeurs par défaut NULL et M).

On peut maintenant ajouter le circuit de Mexico. L'identifiant de Gerard Berger peut-être donné directement (il faut aller vérifier sa valeur dans la BD), ou obtenu par sous-requête (mieux).

```
INSERT INTO circuit (cName, cCity, cCountry, cLength, cLapRec, cDrivRec,
cYearRec) VALUES
('Autodromo Hermanos Rodriguez', 'Mexico city', 'Mexico', 4.438, 77.7, (SELECT
driverID FROM driver WHERE (dFirstName, dLastName) = ('Gerard', 'Berger'))
, 1992)
```

On peut maintenant ajouter le grand prix du Mexique 2015.

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201519, 'GRAN PREMIO DE MÉXICO', (SELECT circuitID FROM circuit WHERE cCity =
'Mexico city'), '2015-11-01', 69, 19)
```

#### Question 2.4 : Ajout d'un grand prix avec sous-requête

On souhaite ajouter le grand prix d'Espagne, en utilisant uniquement les informations disponibles dans le tableau ci-dessus (6<sup>e</sup> gp de la saison, circuit de Catalogne, 10 mai). Pour les autres informations (nom, heure, nombre de tours, on suppose que c'est *comme en 2014*, mais au lieu de recopier les informations, on souhaite utiliser une (ou plusieurs) sous-requête(s) pour les obtenir, sachant que l'identifiant du grand prix d'Espagne 2014 est 201405.

Notez que le nom du circuit de Catalogne contient le mot Catalunya.

Donnez la requête SQL qui permet d'ajouter le grand prix d'Espagne.

Pour faciliter la lecture, les valeurs sont données 1 par ligne.

La requête ci-dessous génère une erreur : on ne peut faire une sous-requête sur grandprix dans une requête de modification de grandprix.

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201506,
(SELECT gName FROM grandprix WHERE gpID = 201405),
(SELECT circuitID FROM circuit WHERE cName LIKE '%Catalunya%'),
CONCAT('2015-05-10 ', (SELECT TIME(gDate) FROM grandprix WHERE gpID = 201405))
,
(SELECT gLaps FROM grandprix WHERE gpID = 201405),
6)
```

La solution consiste à *renommer* la table grandprix des sous-requêtes (par exemple en gp) :

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201506,
(SELECT gName FROM grandprix AS gp WHERE gpID = 201405),
(SELECT circuitID FROM circuit WHERE cName LIKE '%Catalunya%'),
CONCAT('2015-05-10 ', (SELECT TIME(gDate) FROM grandprix AS gp WHERE gpID =
201405)),
(SELECT gLaps FROM grandprix AS gp WHERE gpID = 201405),
6)
```

La portée du renommage étant limitée à la sous-requête (une requête principale ne *voit* pas les tables de ses sous-requêtes), on peut utiliser le même nom `gp` pour renommer `grandprix` dans les 3 sous-requêtes.

On pourrait envisager d'écrire une seule sous-requête, qui renvoie à la fois le nom du circuit, son identifiant, l'heure et le nombre de tour, au lieu des 4 sous-requêtes ci-dessus. C'est possible, mais pas avec la syntaxe vue en cours utilisant `VALUES` ! En effet, avec cette syntaxe, il faut le bon nombre d'informations dans la parenthèse. Et, la requête ci-dessus renvoie l'erreur « #1136 - Column count doesn't match value count at row 1 » due à une mauvaise imbrication des données...

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank) VALUES
(201506,
 (SELECT gName, circuitID, CONCAT('2015-05-10 ', TIME(gDate)), gLaps
  FROM circuit NATURAL JOIN grandprix AS gp WHERE gpID = 201405),
6)
```

La bonne façon de faire est la suivante : remplacer `VALUES` par une sous-requête qui renvoie *toutes* les valeurs de la ligne à ajouter. Noter que la sous-requête peut être parenthésée ou non.

```
INSERT INTO grandprix (gpID, gName, circuitID, gDate, gLaps, gRank)
SELECT 201506, gName, circuitID, CONCAT('2015-05-10 ', TIME(gDate)), gLaps, 6
FROM circuit NATURAL JOIN grandprix AS gp WHERE gpID = 201405
```

### Question 2.5 : Copie de sauvegarde

Pour garder une copie de votre base, créez un « dump » au format SQL (export de la BD au format SQL).

## Exercice 3. Requetes SQL de modification

### Question 3.1 : Modifier les valeurs de plusieurs champs d'une ligne

Le premier grand prix sur le circuit de Sochi a eu lieu ; il y a donc depuis un record du tour. Il a été établi par V. Bottas (en 2014 donc), avec un temps de 1:40.896.

Donnez la requête SQL qui permet de modifier cette information dans la BD.

```
UPDATE circuit SET
  cLapRec = 100.9,
  cDrivRec = (SELECT driverID FROM driver WHERE dLastName = 'Bottas'),
  cYearRec = 2014
WHERE cCity = 'Sochi'
```

L'erreur fréquente est l'oubli de la clause `WHERE` dans la requête précédente. La modification impacte alors toutes les lignes de la table `circuit`. Les circuits se retrouvent alors tous avec le même record du tour. S'il est possible de retrouver l'état précédent (réparer l'erreur) en modifiant les records, ou en supprimant la BD et en rechargeant le dump, il n'est pas possible d'*annuler* la modification ! La plus grande prudence s'impose donc lors de l'utilisation des requêtes de modification des données ! Dans le pire des cas, un dump « récent » de la base permet de retrouver une partie des données perdues ou abîmées...

---

### Question 3.2 : Modifier plusieurs lignes en fonction d'une condition

1. Récupérez sur cursus le fichier `abu_dhabi.sql` et importez-le dans votre BD.

Ce fichier contient les requêtes SQL permettant d'ajouter les résultats du dernier grand prix de la saison 2014 (Abu Dhabi). Cependant, les points indiqués sont les points *normaux* (25 pour le 1<sup>er</sup>, 18 pour le 2<sup>e</sup>...). Or, pour le dernier grand prix de la saison, les points attribués doivent être doublés<sup>4</sup>.

2. Donnez une requête SQL d'interrogation qui permet d'afficher, pour chaque place, les points attribués lors du grand prix d'Abu Dhabi 2014 (la requête doit afficher les valeurs que vous venez d'importer).

```
SELECT driverID, sPos, sPoints
FROM standings NATURAL JOIN grandprix
WHERE gName LIKE '%Abu Dhabi%' AND YEAR(gDate) = 2014
ORDER BY sPos
```

3. Donnez la requête SQL de modification qui permet de multiplier par 2 tous les points attribués lors du grand prix d'Abu Dhabi 2014.

```
UPDATE standings
SET sPoints = sPoints * 2
WHERE gpID = (SELECT gpID FROM grandprix WHERE gName LIKE '%Abu Dhabi%'
AND YEAR(gDate) = 2014)
```

On aurait pu utiliser directement `WHERE gpID = 201419` au lieu de la sous-requête.

4. Vérifiez à nouveau les points attribués (vérifiez aussi que les points des autres grands prix n'ont pas été modifiés).

---

## Exercice 4. Requetes SQL de suppression

### Question 4.1 : Supprimer une équipe

1. Notez le nombre de lignes de chacune des tables de la base (par des requêtes appropriées ; ne pas faire confiance à PHPMyAdmin).

Pour obtenir le nombre de lignes de la table `circuit` on peut exécuter la requête :

```
SELECT COUNT(*)
FROM circuit
```

et recommencer pour chaque table.

Comme on va le faire plusieurs fois, il sera plus pratique de le faire en une seule requête qui les regroupe toutes, avec des `UNION` (et en utilisant des noms de colonnes constants pour renseigner le nom des tables)

---

4. Cette règle a été instaurée en 2014, pour tenter de ménager le suspense sur l'attribution du championnat, et fut supprimée dès l'année suivante !

```

SELECT 'circuit' AS laTable, COUNT(*) AS nbLignes
FROM circuit
UNION
SELECT 'driver' AS laTable, COUNT(*) AS nbLignes
FROM driver
UNION
SELECT 'grandprix' AS laTable, COUNT(*) AS nbLignes
FROM grandprix
UNION
SELECT 'racedriver' AS laTable, COUNT(*) AS nbLignes
FROM racedriver
UNION
SELECT 'standings' AS laTable, COUNT(*) AS nbLignes
FROM standings
UNION
SELECT 'team' AS laTable, COUNT(*) AS nbLignes
FROM team
UNION
SELECT 'testdriver' AS laTable, COUNT(*) AS nbLignes
FROM testdriver

```

On obtient :

<b>laTable</b>	<b>nbLignes</b>
circuit	22
driver	83
grandprix	38
racedriver	47
standings	746
team	14
testdriver	39

2. Donnez la requête de suppression de l'équipe « Ferrari » et exécutez-là.

```

DELETE FROM team
WHERE tName = 'Ferrari'

```

3. Ensuite, notez à nouveau le nombre de lignes de chacune des tables.

On obtient :

<b>laTable</b>	<b>nbLignes</b>	<b>différence</b>
circuit	22	
driver	83	
grandprix	38	
racedriver	43	-4
standings	746	
team	13	-1
testdriver	33	-6

Expliquez les différences.

La requête de suppression a supprimé une ligne de la table team (l'équipe Ferrari), et pour respecter les contraintes d'intégrité référentielle, le SGBD a supprimé (en cascade) des lignes dans les tables des pilotes de course et d'essai.

4. Vérifier s'il y a toujours des résultats de course pour Fernando Alonso en 2014 (par la requête adéquate).

La requête ci-dessous permet de vérifier que tous les résultats de Fernando Alonso, pilote Ferrari en 2014, sont toujours dans la base de données.

Notez que la requête était inutile pour arriver à cette conclusion... en effet, le nombre de ligne des tables driver et standings n'a pas été modifié par la suppression de l'équipe Ferrari.

```
SELECT dFirstName, dLastName, gpID, sPos, sPoints
FROM standings NATURAL JOIN driver
WHERE (dFirstName, dLastName) = ('Fernando', 'Alonso')
AND gpID LIKE '2014%'
```

5. Donnez la requête qui affiche l'équipe de Fernando Alonso pour la saison 2014.

```
SELECT tName
FROM team NATURAL JOIN racedriver
NATURAL JOIN driver
WHERE (dFirstName, dLastName) = ('Fernando', 'Alonso')
AND rSeason = 2014
```

Sans surprise, cette requête renvoie une réponse vide.

6. On se retrouve donc avec un pilote qui a fait des courses mais n'appartient à aucune équipe... Comment pourrait-on corriger la modélisation pour empêcher cela ?

En réalité, les résultats d'une course ne sont pas ceux d'un pilote, mais du couple (pilote, équipe). Chaque résultat devrait donc être associé à un *racedriver* plutôt qu'à un *driver*.

#### Question 4.2 : Supprimer un pilote

Supprimez maintenant le pilote Sebastian Vettel.

```
DELETE FROM driver
WHERE (dFirstName, dLastName) = ('Sebastian', 'Vettel')
```

Notez à nouveau le nombre de lignes de chaque table. Expliquez ces nouvelles différences.

On obtient :

laTable	nbLignes	différence
circuit	16	-6
driver	82	-1
grandprix	28	-10
racedriver	41	-2
standings	525	-221
team	13	
testdriver	33	

La suppression du pilote, de ses 2 engagements (2013 et 2014) et de ses 34 résultats en grand prix (19 en 2013, 14 gp initiaux en 2014 + Abu Dhabi 2014) était facilement prévisible. Ce qui l'était moins



(encore que...) c'est la suppression des 6 circuits dont il était détenteur du record du tour. Notez que pour cette contrainte d'intégrité référentielle sur le record du tour, il aurait été plus judicieux de déclarer `ON DELETE SET NULL ON UPDATE CASCADE`.

La suppression de ces 6 circuits a entraîné (en cascade), la suppression des 10 grands prix (6 en 2013 et 4 en 2014) qui s'y sont déroulés, et des résultats associés à ces courses.

#### Question 4.3 : [long] Suppression sans la « cascade »

Supprimez complètement la base de données.

Récupérez sur cursus le dump de la base « sans cascades ». Dans cette version, les contraintes d'intégrité référentielles sont définies avec : `ON DELETE NO ACTION ON UPDATE NO ACTION` au lieu de `ON DELETE CASCADE ON UPDATE CASCADE`.

Restaurez la BD à l'aide de ce dump.

Donnez la suite de requêtes à exécuter pour pouvoir supprimer Sebastian Vettel, et exécutez-les. Ces requêtes ne doivent pas utiliser de résultats intermédiaires écrits « en dur » (mais peuvent utiliser des sous-requêtes).

Sans les cascades, il faut supprimer *à la main* les lignes contenant des clés étrangères, avant de pouvoir supprimer les clés primaires correspondantes. Pour la requête qui nous occupe, il faut donc supprimer (dans un ordre quelconque) :

- les résultats de Sebastian Vettel dans la table `standings`,
- les associations de Sebastian Vettel avec une équipe dans la table `racedriver`,

puis supprimer, dans un second temps :

- les résultats de tous les pilotes lors des grands prix courus sur des circuits dont Sebastian Vettel détient le record, dans la table `standings`,

puis supprimer, dans un troisième temps :

- les grands prix courus sur des circuits dont Sebastian Vettel détient le record, dans la table `grandprix`,

puis supprimer, dans un quatrième temps :

- les circuits dont Sebastian Vettel détient le record, dans la table `circuit`,

et enfin supprimer, dans un cinquième temps :

- Sebastian Vettel, dans la table `driver`,

ce qui donne la suite de requêtes :

```
DELETE FROM standings
WHERE driverID = (SELECT driverID
                  FROM driver
                  WHERE (dFirstName, dLastName) = ('Sebastian', 'Vettel'));

DELETE FROM racedriver
WHERE driverID = (SELECT driverID
                  FROM driver
                  WHERE (dFirstName, dLastName) = ('Sebastian', 'Vettel'));

DELETE FROM standings
WHERE gpID IN (SELECT gpID
               FROM grandprix NATURAL JOIN circuit
               JOIN driver ON cDrvRec = driverID
               WHERE (dFirstName, dLastName) = ('Sebastian', 'Vettel'));
```

```

DELETE FROM grandprix
WHERE circuitID IN (SELECT circuitID
                    FROM circuit JOIN driver ON cDrivRec = driverID
                    WHERE (dFirstName, dLastName)=('Sebastian', 'Vettel'));

DELETE FROM circuit
WHERE cDrivRec IN (SELECT driverID
                  FROM driver
                  WHERE (dFirstName, dLastName)=('Sebastian', 'Vettel'));

DELETE FROM driver
WHERE (dFirstName, dLastName)=('Sebastian', 'Vettel');

```

Pour finir, on peut noter que dans cette BD, il n’y a aucune raison de supprimer des informations. Ces informations sont publiques (e.g. pas de notion de droit à l’oubli ou de données à caractère privé dont la durée de conservation doit être limitée) et reflètent une réalité (e.g. aucune évaluation subjective soumise à évolution). Les contraintes d’intégrité référentielles devraient donc être définies avec ON DELETE RESTRICT ON UPDATE CASCADE.

#### Question 4.4 : *Suppression en fonction d’une condition*

Donnez la requête SQL qui supprime tous les résultats avec 0 point.

```

DELETE FROM standings
WHERE sPoints = 0

```

#### Question 4.5 : *Suppression en fonction d’une condition dans une autre table*

Donnez la requête SQL qui supprime tous les résultats des pilotes anglais.

```

DELETE FROM standings
WHERE driverID IN (SELECT driverID
                  FROM driver
                  WHERE dCountry = 'Great Britain')

```