
CONTRAINTES :

Durée : 1h30

Documents autorisés : tous les documents de cours et vos TP sont autorisés, la copie du voisin ne fait pas partie de ces catégories !

- L'accès à internet est exclusivement limité à l'accès à *phppgadmin* et à la plateforme AMETIC
Il est bien entendu que les personnes qui sont prises en train de frauder se verront attribuer la note forfaitaire de 0.
- Les téléphones portables sont éteints au fond de vos sacs, leur utilisation quelle qu'en soit la raison entrainera un 0
- Pas d'écouteur sur les oreilles, pas de clé USB insérée dans les ordinateurs

Ressources disponibles :

Annexe 1 : Modèle Physique des données de la base GESTION des grand prix automobile

Annexe 2 : définition de la vue *view_position_arr_pilote*

Barème indicatif :	1 les vues	: Q1 = 2 points ;	Q2 = 2 points
	2 les règles	: Q1 = 4 points ;	Q2 = 4 points
	3 les fonctions	: Q1 = 4 points ;	Q2 = 4 points

COMMENT ME RENDRE VOTRE DEVOIR :

Vous devez copier vos requêtes dans un fichier texte, que vous **nommerez à votre nom** (Ex exemple: *carmignac.txt*) en appliquant la numérotation suivante pour chaque requête :

N°Exo - N°requête

Exemple : 1. 6 pour la requête 6 de l'exercice 1

A la fin, ce fichier sera déposé à l'adresse suivante : http://laurent.carmignac.free.fr/depose_devoir/

BASE UTILISEE : GESTION DES GRAND PRIX AUTOMOBILE

1. Lancez la console d'administration du SGBD PostGres
<http://a-pedagoarles-postgresql.aix.univ-amu.fr/phppgadmin/>) et connectez-vous avec vos paramètres;
2. Récupérez le script de création des tables (*create_formule1.sql*)
3. Exécuter le script
4. Vos tables sont "normalement" créées et renseignées dans le schéma de données *CTRL2_formule1*

QUELQUES EXPLICATIONS SUR LA BDD : GESTION DES GRAND PRIX AUTOMOBILE

La base de données permet la gestion des grands prix de formule1, il se dispute 19 grand prix au cours de la saison. Chacun d'eux a un nom (Grand Prix de France, d'Italie, de San-Marin. . .) et se déroule sur un circuit à une date donnée. Habituellement, chaque circuit est utilisé une seule fois dans la saison, mais il se peut qu'un même circuit soit utilisé pour courir deux grands prix distincts. Chaque circuit a ses caractéristiques : longueur de la piste, pays où il est situé ; nombre de spectateurs que l'on peut y accueillir.

La longueur de la course, est déduite de la longueur de la piste et du nombre de tours à effectuer. Ce nombre de tours varie pour chaque grand prix.

Il y a un certain nombre de qualifiés par grand prix (*nbqualif* table *grandprix*), si un pilote n'est pas qualifié parce qu'il n'a pas réalisé un temps suffisant aux essais qualificatifs sa position sur la grille de départ (*positiongrille* dans la table *courir*) sera égale à 0, de fait il en sera de même pour la colonne *positionarrivee* qui sera elle aussi égale à 0.

S'il abandonne durant la course sa position à l'arrivée sera égale à 'A' (*positionarrivee* dans la table *courir*)

place	points
1	25
2	18
3	15
4	12
5	10
6	8
7	6
8	4
9	2
10	1

Barème des points

Les points sont calculés en fonction du nombre de tours réalisés (*nbtourseffectue* table *grandprix*), si la voiture de tête a fait deux tours au moins mais n'a pas fait la moitié des tours prévus (*nbtour* de la table *grandprix*), on attribue la moitié des points aux 10 premiers. Le barème est dans la table *bareme* qui n'apparaît pas dans le digramme de classe et dans le modèle physique des données puisque c'est une table de paramètre

TRAVAIL A FAIRE

REMARQUES IMPORTANTES

Respecter les noms que je vous donne pour les vues, règles et fonctions,
Écrire les requêtes qui permettent de tester les vues, règles et fonctions.

1. Les vues

1. Créez une vue nommée *pts_GP_Malaisie*, qui affiche le nom, le prénom, la position d'arrivée des pilotes et le nombre de points qu'ils ont acquis aux *Grand Prix de Malaisie*, on s'intéresse uniquement aux pilotes qualifiés (*positionarrivee* différent de 0) et à ceux qui n'ont pas abandonnés (*positionarrivee* différent de A)

Vous ferez attention au type de la colonne *positionarrivee* qui est un *varchar*, vous devez le convertir à la volée si vous voulez la comparez à une colonne qui est de type entier, par exemple pour le convertir en entier \Rightarrow *positionarrivee* ::int

Attention de traiter les positions d'arrivée hors barème, pas de valeur de points acquis vide (NULL) pensez à la fonction *coalesce*.

Vous nommerez les colonnes comme suit :

premier	nom	position_arrivee	pts_acquis
---------	-----	------------------	------------

2. Créez une vue nommée *nb_km_GP*, qui affiche le nom du grand prix et la distance totale à parcourir pour chaque grand prix. Vous nommerez les colonnes comme suit :

nomgp	longueurgp
-------	------------

2. Les règles

1. REGLE SUR UN UPDATE SUR LA VUE *VIEW_POSITION_ARR_PILOTE*

Vous disposez la vue *view_position_arr_pilote* qui donne la position d'arrivée des pilotes pour chaque grand prix, uniquement pour les pilotes qui sont qualifiés et qui n'ont pas abandonnés (voir la définition en ANNEXE 2)

- a. A l'aide de la requête SQL appropriée afficher le résultat de cette vue !!!
- b. Créez la règle *regle_update_view_position_arr_pilote* qui lors d'une modification (sur la vue *view_position_arr_pilote*) de la position d'arrivée et/ou du nom et/ou du prénom du pilote et en fonction de l'id du pilote et de l'id du grand prix, permet de mettre à jour la table *courir* et la table *pilote* en fonction de l'id du pilote et du grand prix.

Voici deux requêtes qui permettent de tester votre règle :

```
UPDATE view_position_arr_pilote
SET nom='DURAND',
    prenom='Carlos',
    position=20
WHERE id_pilote=18 AND id_gp=19 ;
```

```
UPDATE view_position_arr_pilote
SET position=1
WHERE id_pilote=18 AND id_gp=19
```

2. REGLE SUR UN INSERT SUR LA TABLE CIRCUIT

Créer la règle ***regle_insert_circuit*** qui si, lors de l'insertion d'un nouveau circuit, la valeur de la clé primaire (id_circuit) existe dans la table circuit alors il faudra détourner l'insertion et à la place mettre à jour le circuit qui correspond à l'id_circuit avec les données de l'insertion cad :

- nomcircuit,
- payscircuit,
- longpiste,
- nbspectateur

Attention lors de l'insertion vous devez spécifier un id pour le circuit.

Vous devez me fournir les requêtes qui permettent de tester la règle, une pour une insertion effective (cad sans déclencher la règle) et une avec déclenchement de la règle

3. Les fonctions

1. Ecrire la fonction ***tour_Effectue(identifiant du grand prix)*** qui renvoie vrai si le nombre de tours effectué (*nbtourseffectue* de la table *grandprix*) est supérieur ou égal à la moitié du nombre de tours à réaliser (*nbtour* de la table *grandprix*), elle renvoie FAUX le cas contraire

Paramètre : *identifiant du grand prix*;

Vous donnerez 2 jeux de test pour tester cette fonction, un qui renvoie FAUX et l'autre VRAI

2. Ecrire la fonction ***point_acquis(date, identifiant du pilote)*** qui renvoie le total des points acquis par un pilote à une date donnée, cette date permettra de connaître les grands qui se sont déroulés avant cette date et de fait calculer le cumul des points acquis à chacun des grands auxquelles a été qualifié le pilote, ATTENTION pour ce calcul on ne tiendra pas compte du nombre de tours effectué, on ne considérera que la position d'arrivée.

Par exemple si la date est le 2016-12-15 alors on ne prendra en compte que les grands prix qui se sont déroulés avant ou à cette date pour le calcul des points acquis par le pilote.

Si le pilote n'existe pas vous devez retourner un message d'erreur

Paramètre : *date* : date limite pour considérer les grand prix ;
Identifiant du pilote ;

Vous me donnerez 2 jeux de test pour tester cette fonction, avec les valeurs suivantes :

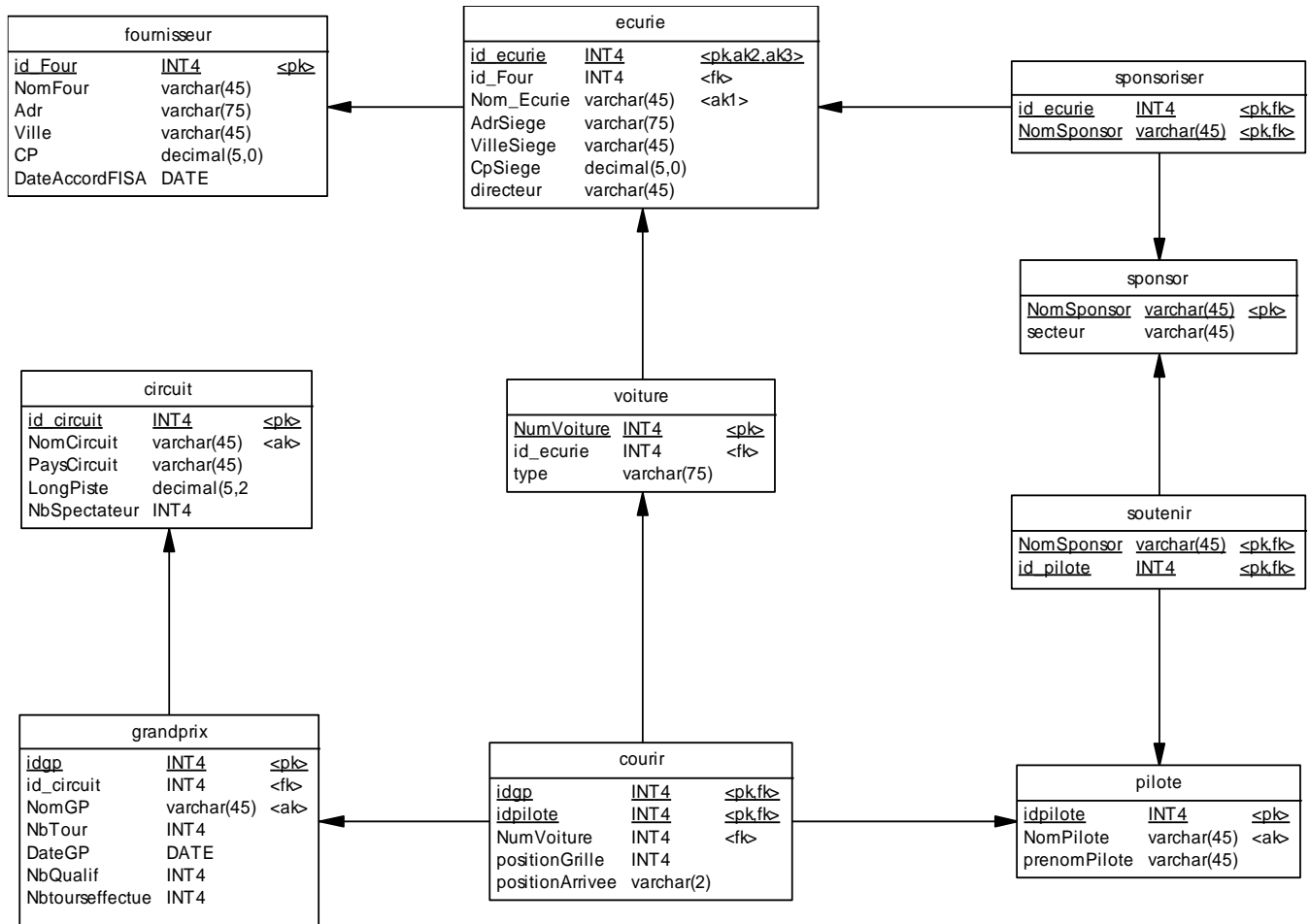
date = '2017-01-30' / *identifiant du pilote* = 1

date = '2016-12-15' / *identifiant du pilote* = 2

Attention la position d'arrivée (*positionarrivee*) est de type *varchar*, donc pour les calculs il faut la convertir en *integer* de cette manière : *positionarrivee :: int*

ET ATTENTION au valeur NULL, puisque au-delà de la 10^{ème} place il n'y a pas de point dans la table *bareme*, donc 0 point

ANNEXE 1 : MODELE PHYSIQUE DES DONNEES



ANNEXE 2 : DEFINITION DE LA VUE VIEW_POSITION_ARR_PILOTE

```
CREATE OR REPLACE VIEW view_position_arr_pilote (id_pilote,id_gp,
nom_grand_prix, prenom, nom, position)
```

AS

```
SELECT P.idpilote, G.idgp, nomgp, prenompilote, nompilote, positionarrivee
from pilote P JOIN courir C ON C.idpilote=P.idpilote
      JOIN grandprix G ON G.idgp=C.idgp
where positiongrille != 0 AND positionarrivee != 'A'
ORDER BY nomgp, positionarrivee::int
```