

**Corrigé : Base de Données**  
**Section : Sciences de l'informatique**  
**Session de contrôle 2017**

---

**Exercice 1 (3 points= 0.25 point \*12)**

1) L'extraction d'un sous ensemble de colonnes d'une table correspond à une :

- |          |            |
|----------|------------|
| <b>F</b> | jointure   |
| <b>V</b> | projection |
| <b>F</b> | sélection  |

2) En SQL, pour automatiser la modification de la valeur d'une clé étrangère dans une table fille à la suite de la modification de la valeur de la clé primaire de la table mère correspondante, on ajoute la clause **ON UPDATE CASCADE** à la définition de la :

- |          |                                   |
|----------|-----------------------------------|
| <b>F</b> | clé primaire dans la table mère   |
| <b>V</b> | clé étrangère dans la table fille |
| <b>F</b> | clé primaire dans la table fille  |

3) L'intégrité d'une base de données consiste à :

- |          |  |
|----------|--|
| <b>V</b> | garantir la cohérence des données                          |
| <b>V</b> | définir un ensemble de règles (valeur, référence, unicité) |
| <b>F</b> | définir les données d'une façon unique                     |

4) En SQL, la clause **GROUP BY** permet :

- |          |   |
|----------|---|
| <b>V</b> | de grouper, selon un critère, des lignes d'une ou de plusieurs tables               |
| <b>F</b> | de grouper, selon un critère, des colonnes d'une ou de plusieurs tables             |
| <b>V</b> | d'effectuer des calculs sur un groupe de résultats à l'aide des fonctions d'agrégat |

**Exercice 2 (9 points)**

Soit la base de données simplifiée "**Gestion\_Rallye**" permettant à une ligue d'amateurs de rallye moto de gérer les différentes étapes d'un rallye.

Cette base de données est décrite par la représentation textuelle suivante :

**PILOTE** (NumPil, NomPil, PrePil, DateNaisPil)

**VILLE** (CodeVil, NomVil)

**ETAPE** (NumEtp, CodeVilDep#, CodeVilArr#, DisEtp, DateEtp, CodeType#)

**TYPE\_ETAPE** (CodeType, DesType)

**TEMPS** (NumEtp#, NumPil#, TempsReal)

1) Suite aux travaux d'entretien entamés sur l'itinéraire de l'étape numéro **5** ayant comme désignation du type de l'étape "**Asphalte**", la ligue décide de changer cet itinéraire par un autre de type "**Forêt**". Pour prendre en charge cette modification, écrire les requêtes SQL permettant:

a) de supprimer la contrainte de domaine nommée **CD**.

```
ALTER TABLE TYPE_ETAPE
DROP CONSTRAINT CD;
```

b) d'ajouter à la table **TYPE\_ETAPE** une nouvelle contrainte de domaine sur la colonne **DesType** permettant d'accepter seulement les désignations suivantes : "**Montagne**", "**Asphalte**", "**Désert**" et "**Forêt**".

```
ALTER TABLE TYPE_ETAPE
ADD CONSTRAINT CHECK (DesType in ('Montagne','Asphalte','Désert','Forêt'));
```

c) d'insérer dans la table **TYPE\_ETAPE** la ligne suivante :

CodeType	DesType
T4	Forêt

```
INSERT INTO TYPE_ETAPE (CodeType, DesType)
VALUES ('T4', 'Forêt');
```

d) de mettre à jour la colonne **CodeType** à « **T4** » pour l'étape numéro **5**.

```
UPDATE ETAPE
SET CodeType='T4'
WHERE NumEtp=5;
```

2) Afin de permettre aux amateurs de suivre de près le déroulement du rallye, la ligue se propose de mettre à leur disposition un ensemble d'informations.

Ecrire les requêtes SQL permettant d'afficher :

a) la liste des étapes programmées pour la journée du « **01/06/2017** ».

```
SELECT *
FROM ETAPE
WHERE DateEtp='20170601';
```

b) La liste des pilotes (**nom** et **prénom**) participants dans l'étape numéro **7** triée par ordre croissant selon leurs temps réalisés.

```
SELECT NomPil, PrePil
FROM PILOTE P, TEMPS T
WHERE P.NumPil = T.NumPil
AND NumEtp = 7
ORDER BY TempsReal;
```

- c) Le total des distances parcourues dans toutes les étapes qui sont programmées dans la montagne.

```
SELECT SUM(DisEtp)
FROM ETAPE E, TYPE_ETAPE T
WHERE E.CodeType = T.CodeType
AND DesType = 'Montagne' ;
```

- d) pour chaque étape du rallye, le meilleur temps réalisé.

```
SELECT NumEtp, MIN(TempsReal)
FROM TEMPS
GROUP BY NumEtp ;
```

- 3) Pour éviter le changement des résultats des étapes du rallye, l'administrateur de la base de données décide de retirer de l'utilisateur **User1** le droit de modification sur la table **TEMPS**.

Sachant que l'utilisateur **User1** est déjà créé et bénéficie de tous les droits, écrire une requête SQL permettant d'effectuer cette tâche.

```
REVOKE UPDATE ON TEMPS
FROM User1 ;
```

### Exercice 3 (8 points)

- 1) Liste des colonnes :

Nom de la colonne	Description	Type
<b>CodZon</b>	Code de la zone	Numérique ou texte
<b>IntZon</b>	Intitulé de la zone	Texte
<b>DigZon</b>	Digicode de la zone	Texte
<b>NumNiv</b>	Numéro du niveau de sécurité	Numérique ou texte
<b>DesNiv</b>	Description du niveau de sécurité	Texte
<b>NumLoc</b>	Numéro du local	Numérique ou texte
<b>DesLoc</b>	Description du local	Texte
<b>MatEmp</b>	Matricule de l'employé	Numérique ou texte
<b>NomEmp</b>	Nom de l'employé	Texte
<b>PreEmp</b>	Prénom de l'employé	Texte
<b>DatnEmp</b>	Date de naissance de l'employé	Date
<b>AdrEmp</b>	Adresse de l'employé	Texte
<b>TelEmp</b>	Numéro de téléphone de l'employé	Numérique ou texte
<b>HeuEnt</b>	Heure d'entrée autorisée à une zone	Date
<b>HeuSor</b>	Heure de sortie autorisée d'une zone	Date

- 2) Représentation textuelle :

**ZONE** (CodZon, IntZon, DigZon, NumNiv#)

**NIVEAU\_SECURITE** (NumNiv, DesNiv)

**LOCAL** (NumLoc, DesLoc, CodZon#)

**EMPLOYE** (MatEmp, NomEmp, PreEmp, DatnEmp, AdrEmp, TelEmp)

**ACCES** (MatEmp#, CodZon#, HeuEnt, HeuSor)