

|   |                                      |  |
|---|--------------------------------------|--|
| <b>RÉPUBLIQUE TUNISIENNE</b><br><b>MINISTÈRE DE L'ÉDUCATION</b><br><br><b>EXAMEN DU</b><br><b>BACCALAURÉAT</b><br><br><b>SESSION 2019</b> | <b>Corrigé Session de contrôle</b>   |  |
|   | Épreuve :<br><b>Bases de données</b> | Section :<br><b>Sciences de l'informatique</b> |
|   | Durée : <b>2h</b>                    | Coefficient de l'épreuve : <b>1.5</b>          |

**Exercice 1 (2,5 Points = 4 \* 0.25pt + 3 \* 0.5 pt)**

A. Pour chacune des propositions ci-dessous, compléter la colonne « Valide » par la lettre **V** si la proposition est correcte, ou par la lettre **F** dans le cas contraire.

| Propositions  | Valide   |
|---|----------|
| Pour ajouter des données à une table, on utilise des états.   | <b>F</b> |
| La commande <b>DELETE</b> permet d'effacer la structure d'une table.  | <b>F</b> |
| Dans une base de données, deux utilisateurs différents, peuvent avoir le même mot de passe.                           | <b>V</b> |
| L'option <b>WITH GRANT OPTION</b> permet d'attribuer à des utilisateurs le droit de sauvegarde de la base de données. | <b>F</b> |

B. Compléter les propositions suivantes, par le nom de la contrainte appropriée, qui peut être : **de table**, **référentielle** ou **de domaine** :

1. La clause **ON DELETE CASCADE** est utilisée pour maintenir la contrainte d'intégrité référentielle en cas de suppression.
2. La clause **CHECK** est utilisée pour garantir la contrainte d'intégrité de domaine pour chaque valeur saisie d'une colonne.
3. La clause **PRIMARY KEY** est utilisée pour assurer la contrainte d'intégrité de table lors de sa création.

## Exercice 2 (8.5 points)

Soit la base de données intitulée "**Club\_Robotique**" permettant à un club d'amateurs de robotique de gérer leurs activités.

Cette base est décrite par la représentation textuelle simplifiée suivante :

**ROBOT** (IdRob, NomRob)

**FABRICANT** (IdFab, LibFab)

**COMPOSANT** (CodeComp, LibComp, PoidsComp, PrixComp, IdFab#)

**MEMBRE** (IdMemb, NomMemb, PreMemb)

**MONTAGE** (IdRob#, CodeComp#, IdMemb#, DateMont)

### Description des colonnes des tables

| Nom de la colonne | Description                             |
|-------------------|---|
| <b>IdRob</b>      | Identifiant du robot                    |
| <b>NomRob</b>     | Nom du robot                            |
| <b>IdFab</b>      | Identifiant du fabricant des composants |
| <b>LibFab</b>     | Libellé du fabricant des composants     |
| <b>CodeComp</b>   | Code du composant monté                 |
| <b>LibComp</b>    | Nom du composant monté                  |
| <b>PoidsComp</b>  | Poids du composant exprimé en grammes   |
| <b>PrixComp</b>   | Prix du composant exprimé en dinars     |
| <b>IdMemb</b>     | Identifiant du membre du club           |
| <b>NomMemb</b>    | Nom du membre du club                   |
| <b>PreMemb</b>    | Prénom du membre du club                |
| <b>DateMont</b>   | Date du montage du composant            |

1. Ecrire une requête SQL permettant de créer la table **MONTAGE** en tenant compte de la description suivante :

**1 pt**

| Nom de la colonne | Type                   |
|-------------------|------------------------|
| IdRob             | Chaîne de 5 caractères |
| CodeComp          | Chaîne de 5 caractères |
| IdMemb            | Chaîne de 5 caractères |
| DateMont          | Date                   |

**N.B :** La valeur de la colonne « **DateMont** » est obligatoire.

```

CREATE TABLE MONTAGE (
  IDRob varchar(5) REFERENCES ROBOT (IDRob),
  CodeComp varchar (5) REFERENCES COMPOSANT (CodeComp),
  IDMemb varchar (5) REFERENCES MEMBRE (IDMemb),
  DateMont Date Not Null,
  CONSTRAINT PK PRIMARY KEY (IDRob, CodeComp, IDMemb));

```

2. Pour visualiser les images capturées par les robots, nous avons besoin d'un nouveau composant «**Caméra vidéo**». Ecrire une requête SQL, permettant d'ajouter ce composant avec les caractéristiques suivantes : **0.75 pt**

| CodeComp | LibComp      | PoidsComp | PrixComp | IdFab |
|----------|--------------|-----------|----------|-------|
| C0015    | Caméra vidéo | 150       | 180.250  | F234  |

```

INSERT INTO COMPOSANT VALUES ('C0015', 'Caméra vidéo', 150, 180.250 , 'F234') ;

```

3. Pour encourager les activités du club, les fournisseurs accordent une réduction de **20%** sur le prix des composants dont le libellé commence par «**Diode LED** ». Ecrire une requête SQL permettant de faire cette mise à jour.

```

UPDATE COMPOSANT SET PrixComp =PrixComp*80%
WHERE LibComp Like 'Diode LED % ' ;

```

**0.75 pt**

4. Pour alléger les robots, le responsable du club décide dans le futur, de n'acheter que des composants dont le poids ne dépasse pas **200 g**. Ecrire une requête SQL permettant d'ajouter cette contrainte sous le nom **VP**.

```

ALTER TABLE COMPOSANT
ADD CONSTRAINT VP CHECK PoidsComp <= 200;

```

**0.75 pt**

5. Ecrire les requêtes **SQL** permettant d'afficher :

- a. la liste nominative des fabricants de composants, triée par ordre alphabétique croissant.

```

SELECT      LibFab
FROM        FABRICANT
ORDER BY   LibFab ;

```

**0.75 pt**

- b. les **nom** et **prénom** des membres qui ont participé dans le montage du robot identifié par '**R101**'.

```

SELECT      NomMemb, PreMemb
FROM        MEMBRE Mb, MONTAGE Mt
WHERE       Mt.IdMemb = Mb.IdMemb
AND         IdRob = 'R101';

```

**0.75 pt**



### Exercice 3 (9 points)

#### A. (6.5 pts)

##### 1. Liste des colonnes (1.5 pts = 6 \* 0.25 avec -0.25 pt par erreur)

| Nom de la colonne | Description                                   | Type         |
|-------------------|---|--------------|
| CodeVoie          | Identifiant de la voie urbaine                | Texte/Entier |
| NomVoie           | Nom de la voie urbaine                        | Texte        |
| IdObj             | Identifiant de l'objet intelligent            | Texte/Entier |
| DesObj            | Descriptif de l'objet intelligent             | Texte        |
| DesPan            | Descriptif de la puissance du panneau solaire | Texte        |
| DesCap            | Descriptif du capteur                         | Texte        |

##### 2. Liste des identifiants (1 pt = 4 \* 0.25)

| Sujet   | Nom de l'identifiant | Description                        | Type         |
|---------|----------------------|------------------------------------|--------------|
| VOIE    | CodeVoie             | Identifiant de la voie urbaine     | Texte/Entier |
| OBJET   | IdObj                | Identifiant de l'objet intelligent | Texte/Entier |
| PANNEAU | IdPan                | Identifiant du panneau solaire     | Texte/Entier |
| CAPTEUR | IdCap                | Identifiant du capteur             | Texte/Entier |

##### 3. Représentation textuelle (4 pts)

VOIE (CodeVoie, NomVoie)

PANNEAU (IdPan, DesPan, CodeVoie#)

OBJET (IdObj, DesObj, IdPan#)

CAPTEUR (IdCap, DesCap, IdObj#)

#### B. (2.5 pts)

VOIE (CodeVoie, NomVoie)

PANNEAU (IdPan, DesPan, CodeVoie#)

OBJET (IdObj, DesObj)

CAPTEUR (IdCap, DesCap, IdObj#)

ALIMENTER (IdPan#, IdObj#)