

Devoir de synthèse N°3 (Nouveau régime)		
Commissariat régional de l'éducation de Tunis1	Section : Sciences de l'Informatique	
	Épreuve : Systèmes & Technologies de l'Informatique	
Niveau : 4 <sup>ème</sup> année	Date : Mardi 17 mai 2022	Durée : 3 H
Nom & Prénom : .....	Classe : .....	Note : ..... /20

*N.B. : Le sujet comporte 6 pages à remettre à la fin de l'épreuve.*

### EXERCICE 1 (4.5 points = 6 \* 0.75)

Dans un contexte de base de données et de développement Web, valider chacune des propositions suivantes en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** dans le cas contraire.

1) En **Javascript**, la fonction **Ch.substring(Deb,Fin)** permet :

- ☐ de copier une sous chaîne de la chaîne **Ch** à partir de la position **Deb** à la position **Fin**.
- ☐ d'extraire une sous chaîne de la chaîne **Ch** à partir de la position **Deb** à la position **Fin**.
- ☐ d'effacer une sous chaîne de la chaîne **Ch** à partir de la position **Deb** à la position **Fin**.

2) En **Javascript**, la fonction **IsNaN(X)** retourne la valeur :

- ☐ **False** si la variable **X** est une chaîne.
- ☐ **True** si la variable **X** est numérique.
- ☐ **True** si la variable **X** est alphanumérique.

3) Le lien entre plusieurs tables d'une base de données est appelé opération de :

- ☐ Sélection.
- ☐ Jointure.
- ☐ Projection.

4) L'utilisation des bases de données permet d'assurer :

- ☐ la mise en forme des données.
- ☐ la non redondance des données.
- ☐ l'intégrité des données.

5) Pour ajouter des données à une base de données, on peut utiliser :

- ☐ une requête.
- ☐ un formulaire dans un site Web dynamique.
- ☐ une page Web statique.

6) En **SQL**, lors de la création d'une contrainte, la clause "**ON DELETE CASCADE**" permet de :

- ☐ respecter l'intégrité référentielle entre la table mère et la table fille.
- ☐ supprimer en cascade les colonnes correspondantes.
- ☐ supprimer en cascade les lignes correspondantes.

## **EXERCICE 2 (6 points = 1.5 + 1 + 0.25 + 0.5 + 0.5 + 0.75 + 0.5 + 1)**

Un **Sniffer** est un programme qui permet de capturer toutes les informations envoyées sur un réseau et de récupérer les données de ses utilisateurs.

Une société se propose de gérer les flux de données transférées par ses employés via son réseau en se basant sur la technique du **Sniffer**.

Pour ce faire, l'administrateur du réseau utilise la base de données "**Gestion\_Transfert**" implémentée et décrite par les requêtes **SQL** suivantes :

```
CREATE TABLE Service (
```

```
    NumService INT NOT NULL PRIMARY KEY ,  
    NomService VARCHAR(50) );
```

```
CREATE TABLE Employe (
```

```
    Matricule VARCHAR(10) NOT NULL,  
    Nomprenom VARCHAR(50) ,  
    Taille_Max DECIMAL(15,3) CHECK ( Taille_Max > 0 ),  
    NumService INT REFERENCES Service ( NumService ),  
    PRIMARY KEY ( Matricule) );
```

```
CREATE TABLE Transfert (
```

```
    NumTransfert INT PRIMARY KEY,  
    DateHeureTransfert DATETIME,  
    Duree INT ,  
    TailleDonnees DECIMAL(15,3) CHECK ( TailleDonnees > 0 ),  
    Matricule VARCHAR(10) REFERENCES Employe (Matricule) );
```

- 1) Donner la représentation textuelle de cette base de données.
- 2) Valider chacune des propositions suivantes en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** dans le cas contraire.

<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

- Un employé peut appartenir à plusieurs services.
- Dans un service, on peut trouver un ou plusieurs employé(s).
- Un employé peut faire plusieurs transferts de données.
- Un transfert de données peut être effectué par plusieurs employés.

- 3) Lors de l'insertion des données, l'administrateur de la base de données a détecté des anomalies lors de la saisie. En effet, le champ "**Duree**" accepte des valeurs **négatives** ou **nulles**.

- a) Donner le nom de la contrainte d'intégrité non respectée.
- b) Ecrire la requête **SQL** permettant de prendre en compte cette contrainte.

- 4) L'administrateur de la base de données utilise une requête pour supprimer un employé qui a transféré des données, un message d'erreur de violation de la contrainte d'intégrité référentielle s'affiche lui interdisant cette suppression.

Ecrire la requête **SQL** permettant de modifier la structure de la base de données afin de permettre à l'administrateur la suppression des employés.

- 5) Afin de distinguer les différents employés par leurs fonctions, on se propose de créer une nouvelle table nommée "**Fonction**" ayant la description suivante :

Champ	Type et taille	Contrainte
CodeFonct	Numérique	Clé primaire
LibFonct	Chaîne de 20 caractères	Obligatoire

- a) Ecrire la requête **SQL** permettant de créer cette table.
- b) On veut appliquer l'intégrité référentielle entre les tables "**Employe**" et "**Fonction**" en respectant les règles de gestion suivantes :

**R1** : Un employé possède une seule fonction.

**R2** : Plusieurs employés peuvent avoir la même fonction.

Ecrire la requête **SQL** permettant d'assurer cette règle d'intégrité.

- c) Le champ "**Taille\_Max**" de la table "**Employe**" ne dépend pas de l'employé mais dépend de sa fonction.

Ecrire les requêtes **SQL** permettant de prendre en charge cette règle de gestion.

### EXERCICE 3 (9.5 points = 4.75 + 4.75)

La société "**Streaming\_HD**" a développé une plateforme autour d'une base de données permettant de gérer ses streamers publiant des contenus en ligne (en direct ou en différé).

Ci-dessous, la représentation textuelle simplifiée de cette base de données.

**Pays** (IdPays , NomPays)

**Streamer** (IdStream, Pseudo, Email, IdPays#)

**Chaine** (IdChaine, NomChaine, DesChaine, URL, IdStream#)

**TypeContenu** (IdType, NomType)

**Contenu** (CodeCont, Titre, DateCreation, Duree, NbreVues, Mode, IdType#, IdChaine#)

#### Description des champs des tables

Champ	Description
IdPays	Identifiant du pays d'un streamer
NomPays	Nom du pays d'un streamer
IdStream	Identifiant d'un streamer
Pseudo	Pseudo d'un streamer
Email	Adresse Email d'un streamer
IdChaine	Identifiant d'une chaîne du streamer
NomChaine	Nom d'une chaîne du streamer
DesChaine	Description d'une chaîne du streamer
URL	Adresse URL d'une chaîne du streamer

Champ	Description
IdType	Identifiant du type d'un contenu
NomType	Nom du type d'un contenu
CodeCont	Code d'un contenu
Titre	Titre d'un contenu
DateCreation	Date de création d'un contenu
Duree	Durée d'un contenu
NbreVues	Nombre de vues d'un contenu
Mode	Mode de transfert d'un contenu ( <b>D</b> : Direct ou <b>I</b> : Différé)

#### Partie A : Gestion de données (4.75 points = 5 \* 0.75 + 2 \* 0.5)

1) Pour chacune des propositions ci-dessous, écrire la requête **SQL** correspondante permettant d'afficher :

a) la liste des chaînes ( **NomChaine**, **DesChaine**, **URL**, **Pseudo** ).

b) la liste des contenus ( **Titre**, **DateCreation**, **Duree**, **NbreVues** ) publiés par le streamer dont le pseudo est "**zizo**".

c) la liste des chaînes ( **IdChaine**, **NomChaine**, **DesChaine** ) ayant des contenus de type "**Gaming**" tout en évitant d'afficher la même chaîne plusieurs fois.

d) le pseudo et le nombre total de vues pour chaque streamer.

e) la liste des contenus ( **Titre**, **DateCreation**, **Duree**, **NbreVues** ) ayant le plus grand nombre de vues.

2) Donner le rôle de chacune des requêtes suivantes :

a) **SELECT** Pseudo, Email

**FROM** Streamer

**WHERE** IdStream **NOT IN** ( **SELECT** IdStream **FROM** Chaîne ) );

b) **SELECT** NomChaîne, Count(CodeCont) nb

**FROM** Contenu CO, Chaîne CH

**WHERE** CO.IdChaîne = CH.IdChaîne

**GROUP BY** NomChaîne

**HAVING** nb > 100 ;

## Partie B : Développement Web (4.75 points = 2.5 + 2.25)

On se propose de créer les pages "Streamer.html" et "Affiche.php" permettant d'afficher la liste des chaînes d'un streamer donné, ainsi que le fichier "Contrôle.js" qui contient les fonctions écrites en **JavaScript** nécessaires pour le contrôle du formulaire.

La page "Streamer.html" contient le formulaire suivant :

### Recherche des chaînes d'un streamer

Nom du Streamer :

Soit le code **HTML** incomplet de cette page :

N° ligne	Code HTML
1	<html>
2	<head>
3	<meta http-equiv="Content-Type" content="text/html">
4	<title>Streamer</title>
5	.....
6	</head>
7	<body>
8	<h1 align="center"> Recherche des chaînes d'un streamer </h1>
9	<form method= .....action=..... name="f" onsubmit=.....>
10	<p>Nom du Streamer :<input type="text" name="nom" size="20"></p>
11	<p align="center">
12	<input type="....." value="Afficher" name="B1">
13	<input type="....." value="Annuler" name="B2">
14	</p>
15	</form>
16	</body>
17	</html>

En cliquant sur le bouton "**Afficher**", le passage à la page "**Affiche.php**" se fait lorsque la fonction "**Verif( )**" du fichier "**Controle.js**" retourne **vrai** c'est-à-dire la zone de texte "**Nom du Streamer**" est non vide et ne contient que des chiffres et des lettres.

1) Compléter le code **HTML** de la page "**Streamer.html**" décrite précédemment en ajoutant :

- a) la balise **HTML** manquante de la ligne **5** permettant de lier cette page au fichier "**Controle.js**".
- b) les options adéquates dans la balise **HTML** de la ligne **9**.
- c) le type adéquat de chaque bouton dans les balises **HTML** des lignes **12** et **13**.

2) Afin de créer la page "**Affiche.php**", valider chacune des propositions suivantes en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** dans le cas contraire.

a) En **php**, l'instruction qui permet de récupérer le contenu de la variable "**nom**" d'un formulaire est :

☐ \$nom=\$\_POST["nom"];      ☐ \$nom=\$POST["nom"];      ☐ \$nom=\$\_GET["nom"];

b) En **php**, la requête qui permet de chercher, dans la base de données, un streamer dont le nom saisi dans le formulaire est :

☐ \$req="select \* from streamer where (Pseudo= '\$nom') ";

☐ \$req="select \* from chaine where (Pseudo= '\$nom') ";

☐ \$req="select \* from streamer where (IdStream = '\$nom') ";

c) En **php**, la requête qui permet de chercher, dans la base de données, la liste des chaînes d'un streamer dont le nom saisi dans le formulaire est :

☐ \$req="select ch.\* from chaine ch, streamer s where (ch.IdStream= s.IdStream) and (s. IdStream= '\$nom') ";

☐ \$req="select ch.\* from chaine ch , streamer s where (ch.IdStream= s.IdStream) and (Pseudo= '\$nom') ";

☐ \$req="select \* from chaine where (IdStream= select IdStream from streamer where (Pseudo= '\$nom' ) ) ";