

Nom : ..... Prénom : .....

Note :  
..... / 20

### EXERCICE N°1 : 5 points

Dans un contexte de base et pour chacune des affirmations suivantes, mettre une croix (X) dans la case correspondante à la réponse correcte.

**Important :** sachant qu'il existe **une seule proposition correcte** pour chaque affirmation, toute réponse comportant **plus d'une croix** est considérée **erronée**

1) **Les valeurs d'une clé primaire :**

- Peuvent être redondantes.
- Peuvent être vides.
- Sont obligatoirement uniques.
- Sont obligatoirement de type entier.

2) **Dans une base de données relationnelle, un attribut est :**

- Une Table
- Une ligne
- Une colonne

3) **Pour filtrer des données dans une requête de sélection, on utilise la clause :**

- ORDER BY
- WHERE
- GROUP BY
- LIKE

4) **En langage SQL, le tri des données doit être effectué à l'aide d'une requête :**

- De sélection
- De modification
- D'ajout
- De suppression

5) **En langage SQL, pour assurer l'intégrité des données sur une table fille suite à la modification de la valeur de la clé primaire d'une table mère, on utilise la clause :**

- ON INSERT CASCADE
- CHECK
- ON UPDATE CASCADE
- ON DELETE CASCADE

6) **Une colonne clé étrangère :**

- Peut contenir des doublons
- Peut-être une partie d'une clé primaire d'une autre table
- Peut contenir des valeurs 'NULL'

7) En langage SQL, la clause **DISTINCT** appliquée à une colonne, permet :

- D'éliminer les doublons dans les résultats d'une requête de sélection
- De supprimer des données d'une table
- De rechercher les valeurs différentes dans une colonne

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

- Jointure
- Projection
- Sélection

9) En langage SQL, la clause **DEFAULT** :

- Peut être appliqué à une colonne comme clé primaire
- Permet d'attribuer une valeur par défaut comme valeur initiale à une colonne
- Est appliquée lorsque la colonne est obligatoire.

10) En langage SQL, la modification de la structure d'une table est effectuée avec la commande :

- UPDATE
- MODIFY
- ALTER TABLE

## EXERCICE N°2 : 4 points

Soient les requêtes en SQL suivantes permettant la création des tables **ESPECES** et **CATEGORIE** d'une base de données relative à la gestion des espèces d'animaux en voie de disparition.

```
CREATE TABLE ESPECE
( IdEspece VARCHAR(8),
NomEspece VARCHAR(25) NOT NULL,
NombreRestant INT CHECK (NombreRestant >= 0),
AgeMoyen INT,
PRIMARY KEY (IdEspece)
);
```

```
CREATE TABLE CATEGORIE
( IdCategorie VARCHAR(4),
NomCategorie VARCHAR(30),
Description VARCHAR(10),
PRIMARY KEY (IdCategorie)
);
```

Questions :

1) Déduire la représentation textuelle de la table **ESPECE** (1pt).

.....  
 .....

2) Utiliser la liste des types de contraintes ci-dessous pour compléter le tableau par le type de la contrainte correspondante à chaque proposition (0,75pt).

Types de contraintes : de table/de domaine/d'intégrité référentielle

Proposition	Type de contrainte
PRIMARY KEY (IdEspece)	contrainte .....
NomEspece VARCHAR(25) NOT NULL	contrainte .....
NombreRestant INT CHECK (NombreRestant >= 0)	contrainte .....

3) Il s'est avéré que la taille définie du champ Description de la table CATEGORIE ne suffit pas pour contenir les données nécessaires. Ecrire en SQL la requête permettant de modifier cette taille à 100 caractères (1.pt).

4) Soit la règle de gestion suivante : "Une espèces appartient à une seule catégorie.  
Ecrire en SQL la ou les requêtes permettant de prendre en compte cette règle (1.25pts).

### Exercice n° 3 : 10 points

Ci-après, on donne la représentation textuelle simplifiée d'une base de données concernant un cycle de formation destiné à des étudiants. Il regroupe un ensemble de matières. On considère que chaque enseignant n'enseigne qu'une seule matière et qu'à la fin du cycle de formation, une note par matière, est attribuée à chaque étudiant. D'autre part, les étudiants peuvent ne pas suivre les mêmes matières.

**ETUDIANT** (CodeEt, NomEt, DatenEt)

**MATIERE** (CodeMat, NomMat, CoefMat)

**NOTE** (CodeEt#, CodeEns#, Note)

**ENSEIGNANT** (CodeEns, NomEns, GradeEns, CodeMat#)

Avec :

Intitulé	Libellé
CodeEt	Code de l'étudiant
NomEt	Nom de l'étudiant
DatenEt	Date de naissance de l'étudiant
CodeMat	Code de la matière
NomMat	Nom de la matière
CoefMat	Coefficient de la matière
Note	Note obtenue par l'étudiant dans une matière
CodeEns	Code de l'enseignant
NomEns	Nom de l'enseignant
GradeEns	Grade de l'enseignant (Grd1, Grd2, ...)

**Questions :** Ecrire les requêtes SQL permettant de :

1) Insérer l'étudiant ayant les informations suivantes (0.5 pt)

CodeEt='Et125' , NomEt='Rayen' et DatenEt='22/07/2002'

2) Modifier le coefficient de la matière "Programmation" de 2 à 3 (0.75 pt)

3) Afficher les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom (0.75 pt)

4) **Afficher les noms et les grades des enseignants de la matière dont le nom est 'STI' (1 pt)**

.....

.....

.....

5) **Afficher la liste distincte formée des noms et des coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3' (1 pt)**

.....

.....

.....

6) **Afficher la liste des matières (Nom et Coefficient) par ordre décroissant de Nom qui sont suivies par l'étudiant de code 'Et125' (1.5 pts)**

.....

.....

.....

.....

.....

7) **Afficher la meilleure note obtenue dans la matière dont le nom est 'Réseaux' (1.25 pt)**

.....

.....

.....

.....

8) **Afficher le nombre d'enseignants de la matière dont le nom est 'STI' (1 pt)**

.....

.....

.....

9) **Afficher pour chaque étudiant, le nombre de matières suivies qui sont supérieur à 5 (1.75 pt)**

.....

.....

.....

.....

.....

10) **Supprimer l'étudiant de code 'Et125' (0.5 pt)**

.....

.....