Lycée Secondaire Mahmoud el Messaâdi, Ettadhamen ********** ********** ********* ENSEIGNANTE : Mervet Hakiri	NIVEAU : 4ème SI-1 -STI	
	Devoir de contrôle N°2	
	Date : 22 Novembre 2024	**Durée : 1H**

Nom & Prénom : ……..…… Note : ……../20

Exercice 1 (3 points)

Dans le contexte des bases de données et pour chacune des propositions suivantes, valider chacune de ces réponses, en mettant dans la case correspondante la lettre "**V**" si elle est juste ou la lettre "**F**" si elle est fausse.

1) Qu'est-ce qui caractérise une base de données ?

V	Elle est persistante
F	Elle est volumineuse
V	Elle est structurée

2) Suivant quel langage de données pouvons-nous décrire les tables et les relations entre elles ?

V	LDD
F	LMD
F	SGBD

3) À quelles clés s'applique la contrainte d'unicité dans une table ?

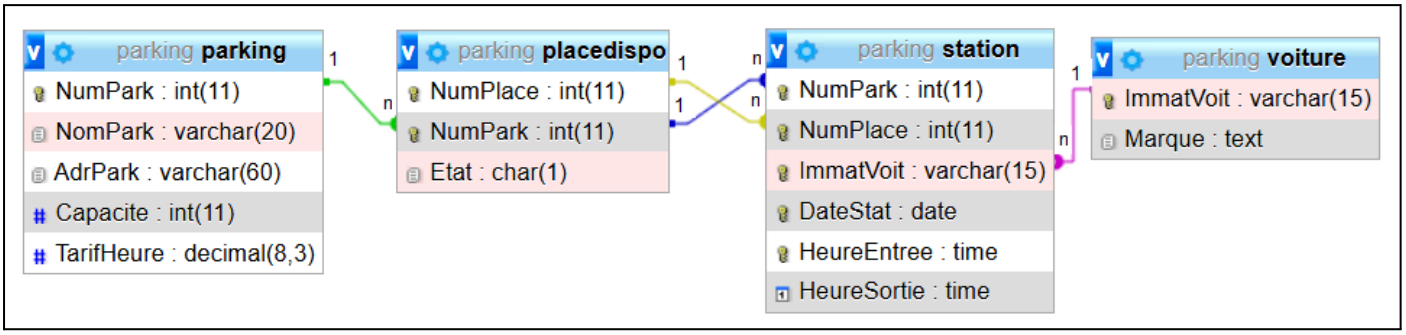
F	A sa clé primaire et aux clés étrangères
F	Aux clés étrangères
V	A sa clé primaire

4) Qu'est-ce qui représente un attribut dans une base de données relationnelle ?

V	Colonne
F	Ligne
V	Champ

Exercice 2 (6.5 points)

Soit la base de données intitulée "**Gestion_parkings**" permettant de gérer, en ligne, un ensemble de parkings de voitures situés au centre-ville. Elle est décrite par la représentation graphique simplifiée suivante :



N.B : La contrainte d'intégrité référentielle de suppression en cascade n'est pas activée entre toutes les tables.

1) Proposer une représentation textuelle de la base de données présentée ci-dessus.

parking (**NumPark**, NomPark, AdrPark, Capacite, TarifHeure)

placedispo (**NumPlace, NumPark#**, Etat)
voiture (**ImmatVoit,** Marque)
station (**NumPark#, NumPlace#, ImmatVoit#, DateStat, HeureEntree,** HeureSortie)

2) L'administrateur de la base de données a rempli les tables suivantes :

parking

NumPark	NomPark	AdrPark	Capacite	TarifHeure
110	El kasba	Rue du 2 mars 1934	1200	2
111	Park Avenue	Rue de Turquie	630	3,6
112	Lafayette Park	Rue de Palestine	917	4

placedispo

NumPlace	NumPark	Etat
12	110	O
13	110	N
12	112	O
25	111	O
99	110	N
125	112	N
4	111	O
25	112	O

voiture

ImmatVoit	Marque
4444 TU 144	Renault Clio N
5555 TU 129	Captur Renault TCe
8989 TU 195	Polo 7 G

station

NumPark	NumPlace	ImmatVoit	DateStat	HeureEntree	HeureSortie
110	99	4444 TU 144	2024-01-25	08 :00	13 :00
112	125	8989 TU 195	2024-02-14	10 :00	12 :00

Soient les requêtes suivantes :

R1: **INSERT INTO** placedispo (NumPlace, NumPark, Etat) **VALUES** (25, 111, 'N') ;

R2: **DELETE FROM** parking **WHERE** NumPark = 110 ;

R3: **INSERT INTO** placedispo **VALUES** (110, 122, 'O');

R4: **INSERT INTO** parking **VALUES** (113,'central park', 'Rue le caire', -500, 5);

En exécutant les requêtes, il apparaît qu'il y a des contraintes d'intégrités qui n'ont pas été respectées.
Remplir le tableau ci-après en expliquant les anomalies rencontrées et donner les noms des contraintes d'intégrité correspondantes.

Requêtes	Anomalie repérée (explication)	Contrainte d'intégrité non respectée
R1	Le couple (25,111) existe déjà dans la table « placedispo ». C'est une clé primaire qui doit être unique.	Contrainte de table
R2	La valeur 110 de la clé étrangère « NumPark » dans la table fille « placedispo » n'existe plus dans la table mère « parking » car la contrainte on delete cascade n'est pas activée.	Contrainte d'intégrité référentielle
R3	La valeur 122 de la clé étrangère « NumPark » n'existe pas dans la table mère « Parking ».	Contrainte d'intégrité référentielle
R4	La capacité ne doit pas être un entier négatif.	Contrainte de domaine

3) Après avoir traité toutes les anomalies, on désire écrire les requêtes SQL (ou rôles ou résultats) permettant d'assurer les tâches demandées ci-dessous :

N.B : respecter le format d'affichage.

Rôle	Requête	Résultat affiché
Afficher la plus grande capacité des parkings.	Select MAX(Capacite) As "Capacité maximale" From parking ;	**Capacité maximale** 1200
Afficher la liste des voitures de la marque « Renault ».	Select * From Voiture Where Marque like "%Renault%" ;	**ImmatVoit** / **Marque** 4444 TU 144 / Renault Clio N 5555 TU 129 / Captur Renault TCe

Afficher les parking (NumPark, NomPark) triés dans l'ordre décroissant des NomPark.	Select NumPark, NomPark From parking Order By NomPark desc ;	

NumPark	NomPark
111	Park Avenue
112	Lafayette Park
110	El Kasba

Exercice 3 (2 points)

Soit la base de données simplifiée intitulée "**PARCZOO**" permettant de gérer les opérations de vaccination des animaux du parc. Elle est décrite par le schéma suivant :

CAGE (NumCage, Superficie)
ANIMAL (IdAnimal, NomAnimal, Famille)
VACCIN (IdVaccin, Libelle)

On propose les contraintes suivantes :

Un animal appartient à une cage.
Une cage contient plusieurs animaux.
Un animal est vacciné plusieurs fois.
Un vaccin est accordé à plusieurs animaux et chaque vaccination est réalisée à une date précise avec une dose bien déterminée.

En tenant compte de ces contraintes, apporter les modifications nécessaires à la structure de cette base de données.

CAGE (**NumCage,** Superficie)
ANIMAL (**IdAnimal,** NomAnimal, Famille, NumCage#)
VACCIN (**IdVaccin,** Libelle)
VACCINATION (**IdAnimal#, IdVaccin#, DateVaccin,** Dose)

Exercice 4 (8.5 points)

Afin de gérer les victoires des courses de la **MOTOGP**, on se propose d'utiliser la base de données simplifiée intitulée "**GestionMOTOGP**" décrite par le schéma textuel suivant :

MARQUEMOTO (**CodeMarq**, LibMarq)
PILOTE (**IdPilote**, NomPilote, DateNais)
CIRCUIT (**CodeCircuit**, NomCir, PaysCir, NbVirage, LongCir)
VICTOIRE (**IdPilote#, CodeCircuit#,** DateCourse, TempsRealise, CodeMarq#)

Champ	Type	Description
CodeMarq	Chaine de 4 caractères	Code d'une marque du moto course
LibMarq	Chaine de 50 caractères	Marque d'une automobile, obligatoire
IdPilote	Chaine de 4 caractères	Identifiant d'un pilote
NomPilote	Chaine de 50 caractères	Nom et prénom d'un pilote
DateNais	Date	Date de naissance d'un pilote, obligatoire
CodeCircuit	Chaine de 4 caractères	Code d'un circuit
NomCir	Chaine de 50 caractères	Nom d'un circuit, obligatoire
PaysCir	Chaine de 50 caractères	Pays d'un circuit, obligatoire
Nbvirage	Entier	Nombre de tours d'un circuit, obligatoire, >0
LongCir	Entier	Longueur d'un circuit exprimée en mètre, obligatoire, >0
DateCourse	Date	Date d'une course, obligatoire
TempsRealise	Heure	Temps réalisé dans une course par le pilote vainqueur

Soit la description des colonnes des tables de la base de données "**GestionMOTOGP**" :

1) Créer, la base de données à nommer "**GestionMOTOGP**".

```
CREATE DATABASE GestionMOTOGP ;
```

2) Créer la table « Victoire ».

```
CREATE TABLE Victoire (IdPilote varchar (4), CodeCircuit varchar (4), DateCourse date NOT NULL,
TempsRealise time, CodeMarq varchar (4),
PRIMARY KEY (IdPilote, CodeCircuit),
```

```
FOREIGN KEY (IdPilote) REFERENCES Pilote (IdPilote) on delete cascade on update cascade,
FOREIGN KEY (CodeCircuit) REFERENCES Circuit (CodeCircuit) on delete cascade on update cascade,
FOREIGN KEY (CodeMarq) REFERENCES MarqueMoto (CodeMarq) on delete cascade on update
cascade ) ;
```

3) Ajouter une contrainte « C1 » sur le champ « NomCir » de la table « Circuit » pour qu'il soit unique.

```
ALTER TABLE Circuit
ADD CONSTRAINT C1 UNIQUE (NomCir) ;
```

4) Ajouter le champ « DateCourse » à la clé primaire de la table « Victoire ».

```
ALTER TABLE Victoire
DROP CONSTRAINT PRIMARY KEY,
ADD CONSTRAINT PRIMARY KEY (IdPilote, CodeCircuit, DateCourse) ;
```

5) Ajouter le champ « Genre » à la table « Pilote » de type caractère, qui ne peut contenir que « M » ou « F » et ayant « M » comme valeur par défaut.

```
ALTER TABLE Pilote
ADD Genre char (1) DEFAULT "M" CHECK (Genre IN ("M", "F")) ;
```

6) Par erreur lors de la création de la table « Pilote », l'administrateur a saisi 5 comme taille du champ « NomPilote », modifier la taille du ce dernier en 50 caractères.

```
ALTER TABLE Pilote
MODIFY NomPilote varchar (50) ;
```

7) Remplir la table « Pilote » en se basant sur les données présentées ci-dessous. (Utiliser une seule requête)

```
INSERT INTO Pilote VALUES (' P001', 'Francesco Bagnaia', '1979-02-16', 'M'),
                          (' P002', 'Alexandra Rins', '', 'F'),
                          (' P003', 'Jorge Martin', '1993-02-17', 'M') ;
```

PILOTE			
IdPilote	**NomPilote**	**DateNais**	**Genre**
P001	Francesco Bagnaia	1979-02-16	M
P002	Alexandra Rins		F
P003	Jorge Martin	1993-02-17	M

8) Lors de l'insertion de l'enregistrement du pilote « P001 », l'administrateur a fait une faute de saisie au niveau de sa date de naissance en lui diminuant 2 mois. Ecrire la requête qui permet de corriger cette faute.

```
UPDATE Pilote
SET DateNais = ADDDATE (DateNais, interval 2 month)
WHERE IdPilote LIKE "P001" ;      ou bien WHERE IdPilote = "P001" ;
```

9) Créer les requêtes suivantes :

R1 : Afficher le nombre des victoires par pilote.

```
SELECT IdPilote, COUNT(*     ……     CodeCircuit) "Nombre victoire"
FROM Victoire
GROUP BY IdPilote ;
```

R2 : Afficher le nom du pilote qui n'a pas une date de naissance.

```
SELECT NomPilote
FROM Pilote
WHERE DateNais = '' ;
```

R3 : Afficher la liste des victoires réalisées pendant le mois de janvier 2024.

```
SELECT *
```

```
FROM Victoire
WHERE MONTH(DateCourse) = 1 AND YEAR(DateCourse) = 2024 ;
```

R4 : Afficher la moyenne des nombres de virages.

```
SELECT AVG(NbVirage) AS "Moyenne virages"
FROM Circuit ;
```

R5 : Afficher les marques des motos (CodeMarq, LibMarq) triées dans l'ordre alphabétique des LibMarq.

```
SELECT CodeMarq, LibMarq
FROM MarqueMOTO
ORDER BY LibMarq ASC ;      ou bien ORDER BY 2 ASC ;
```

R6 : Afficher les circuits (NomCir, PaysCir) situés dans les pays dont le nom contient la lettre « i » comme avant dernier caractère.

```
SELECT NomCir, PaysCir
FROM Circuit
WHERE PaysCir LIKE '%i_' ;
```

Bon travail