Création et modification de la structure d'une base de données

Soit la description textuelle de la base de données d'une agence de location de voitures:

Client (NCIN_Cli, Nom_Cli, Prenom_Cli, Adresse_Cli, Ville)

Vehicule (Immat_vehicule, Marque, Model, Annee_Acq)

Location (Immat_Vehicule#, NCIN#, Date_Loc, Duree_Loc, Cout_Loc)

Créer les tables "**Client**", "**Vehicule**" et "**Location**" en spécifiant les types de données et les contraintes d'intégrités énoncées dans les tableaux ci-dessous:

Table Client		
Nom de colonne	***Type de données***	***Contraintes***
NCIN_Cli	Texte(8)	Clé primaire
Nom_Cli	Texte(20)	Obligatoire
Prenom_Cli	Texte(20)	Obligatoire
Adresse_Cli	Texte(50)	
Ville	Texte(20)	

Table Vehicule		
Nom de colonne	***Type de données***	***Contraintes***
Immat_vehicule	Texte(10)	Clé primaire
Marque	Texte(20)	Obligatoire et prend l'un des valeurs suivantes {"Peugeot", "Renault", "Fiat", "Opel", "Volkswagen"," Citroën"}
Model	Texte(20)	Obligatoire
Annee_Acq	année	obligatoire

Table Location		
Nom de colonne	***Type de données***	***Contraintes***
Immat_vehicule	Texte(10)	Clé primaire
NCIN_Cli	Texte(8)	
Date_Loc	Date	Par défaut la date d'aujourd'hui <=date d'aujourd'hui
Duree_Loc	Numérique	>0
Cout_Loc	Numérique	>0

III. Création d'une BD en mode commande

Le mode commande consiste à créer les différentes structures de la base de données à l'aide de commandes du langage SQL.

Ce langage est composé de trois familles de commandes:

- Langage de définition de données (LDD)
  - création de tables : CREATE TABLE
  - modification de tables: ALTER TABLE
  - suppression de tables: DROP TABLE
- Langage de manipulation de données (LMD /DML)
  - insertion de lignes: INSERT
  - mise à jour des lignes: UPDATE
  - suppression de ligne: DELETE
  - Langage de requêtes : SELECT ……. FROM ……. WHERE …….
- Langage de contrôle de données: GRANT, REVOKE

1) Création de la BD:

N.B: *Le SGBD utilisé pour exécuter les différentes commandes SQL vu tout le long de ce chapitre et les chapitres suivants est MySQL version 5.0.45*

Activité9

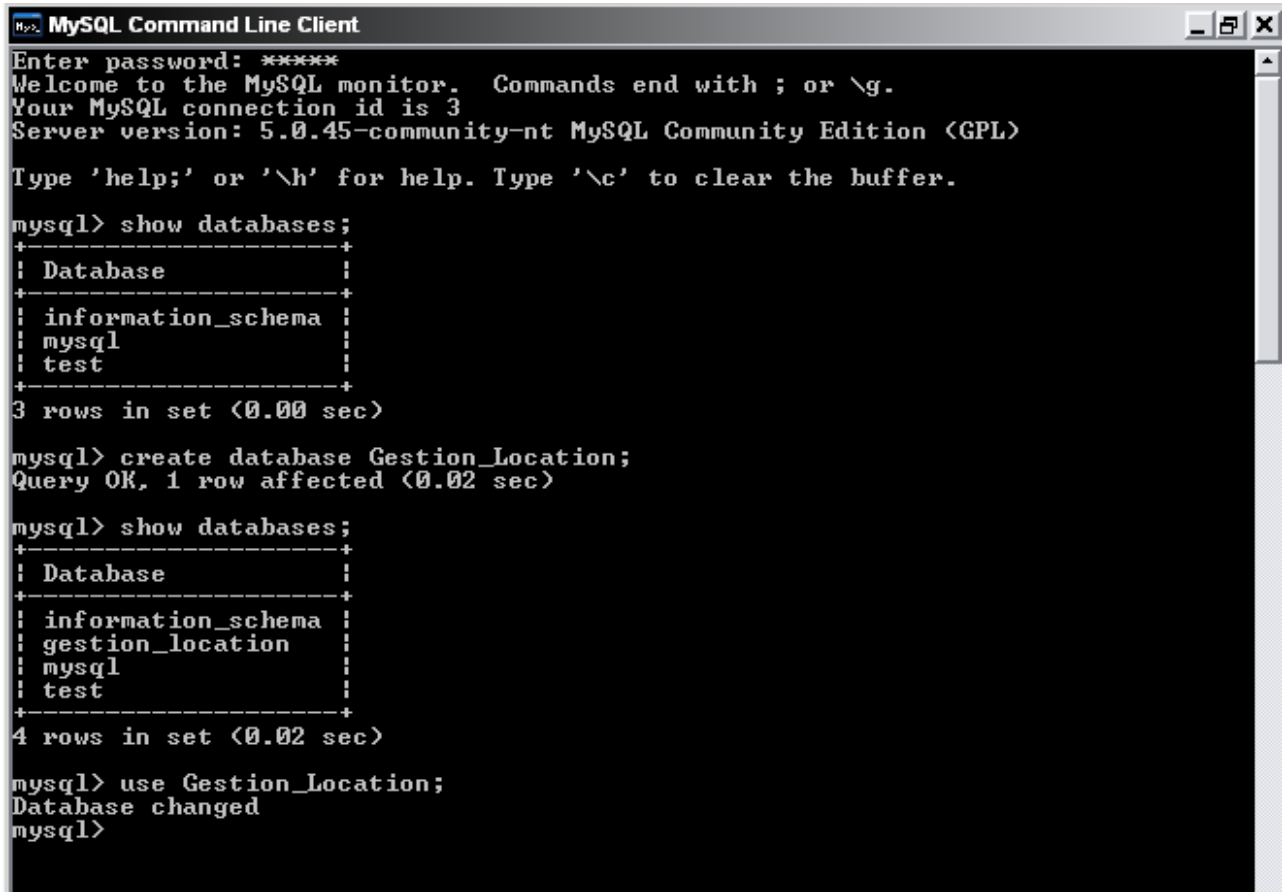
Lancer MySQL
Créer la base de données "***Gestion_Location***"

Les étapes à suivre:

1. *Taper la commande "* **`CREATE DATABASE Gestion_Location`**;"
2. *Pour utiliser la base déjà créée, taper la commande "***`USE Gestion_Location;"`**

Remarque:

- Les commandes **`"CREATE DATABASE <nom_de_la_base>"`** et **`"USE <nom_BD>"`** sont des commandes SQL propres au SGBD MySQL et non des commandes normalisées comme celle qu'on va les voir à la suite de ce cours.
- Il s'agit de même de trois commandes:
  - **`"SHOW DATABASES`**" pour afficher les noms des bases de données existantes
  - **`"SHOW TABLES"`** pour afficher les tables d'une base de données déjà utilisée
  - **`"DESCRIBE <nom_table>"`** affiche pour la table "nom_table" ses colonnes ainsi que leurs types et contraintes de domaine s'il existe.



2) Création d'une table

La commande SQL qui permet de créer une table en donnant son nom, ses attributs et ses contraintes est:

```
CREATE TABLE nom_table (
        Nom_Colonne1        type_colonne1        [ [CONSTRAINT] contrainte-col]*
        …
        [CONSTRAINT] contrainte-table);
```

Légende :

[ ]:option ***** : applicable autant de fois que souhaité **mots en capitale** : mots-clé.

Noter Bien :

- **Contrainte-col : contrainte sur une colonne**
  - **NOT NULL** : obligatoire
  - **UNIQUE** : pas deux lignes avec la même valeur
  - **PRIMARY KEY** : identifiant de la table (UNIQUE + NOT NULL)
  - **REFERENCES** nom_table (nom-col)
  - **CHECK (condition):** Condition que chaque ligne de la table doit vérifier
  - **AUTO_INCREMENT :** s'applique pour une colonne de type entier (identique au type Numéro_Auto en Ms. Access)
    Default : Permet d'ajouter une valeur par défaut.

❖ Contrainte-table : contraintes sur une table

- **PRIMARY KEY** (nom-col1, [nom-col2,…])
- **FOREIGN KEY** (nom-col)* **REFERENCES** nom_table [(nom-col)*]
- **CHECK** (condition)

❖ Les principaux types de données sous MySQL

✓ Les champs numériques :

TINYINT	Entier très petit compris entre -128 et 127 Si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 255
INT	Entier standard compris entre –2 147 483 648 et 2 147 483 647. Si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 4 294 967 295
NUMERIC (n)	Décimal de double précision
DECIMAL (entier,décimal)	Réel, définissez la longueur de chacune des deux parties.

✓ Les champs chaines de caractères :

CHAR (n)	Chaîne de n caractère, taille fixe	
VARCHAR(M)	Chaîne de caractères variable. M peut être compris entre 1 et 255.	255 caractères maximum
BLOB, TEXT	Zone de texte standard Objet d'une longueur maximale de 65535 caractères, TEXT aura un contenu de type ASCII (casse insensible) et BLOB aura un contenu de type binaire (casse sensible).	65 535 caractères. maximum

✓ Les champs de type date et heure :

DATE	Date (ex: 2000-08-24)
TIME	Heure (ex: 23:44:05)
DATETIME	Date et heure (ex: 2000-08-24 23:44:05)
YEAR	Année (ex: 2000)

Remarque:

❶ Qu'est ce qui se passe quand on détruit une clé primaire qui est référencée par un tuple (foreign key) d'une autre table?

Pour supprimer aussi la ligne de la table fille qui correspond à la clé primaire de la table mère, il faut utiliser l'action "**ON DELETE CASCADE**" lors de la définition de clé étrangère.

❷ Lors de la création des tables en SQL, on doit toujours commencer par les tables mères.

Activité 10

Créer les trois tables **Client**, **Vehicule** et **location**
Les commandes SQL permettant la création de 3 tables sont:

❖ Table Client

```
CREATE TABLE Client(
NCIN_Cli VARCHAR(8) PRIMARY KEY,
Nom_Cli VARCHAR(20) NOT NULL,
Prenom_Cli VARCHAR(20) NOT NULL,
Adresse_Cli VARCHAR(50),
Ville VARCHAR(20));
```

❖ Table Vehicule

```
CREATE TABLE Vehicule(
immat_Vehicule VARCHAR(10)  primary key,
marque VARCHAR(20) NOT NULL CHECK(marque
in('Peugeot','Renault','Fiat','Opel','Volkswagen','Citroën')),
model VARCHAR(20) NOT NULL,
date_acq YEAR NOT NULL);
```

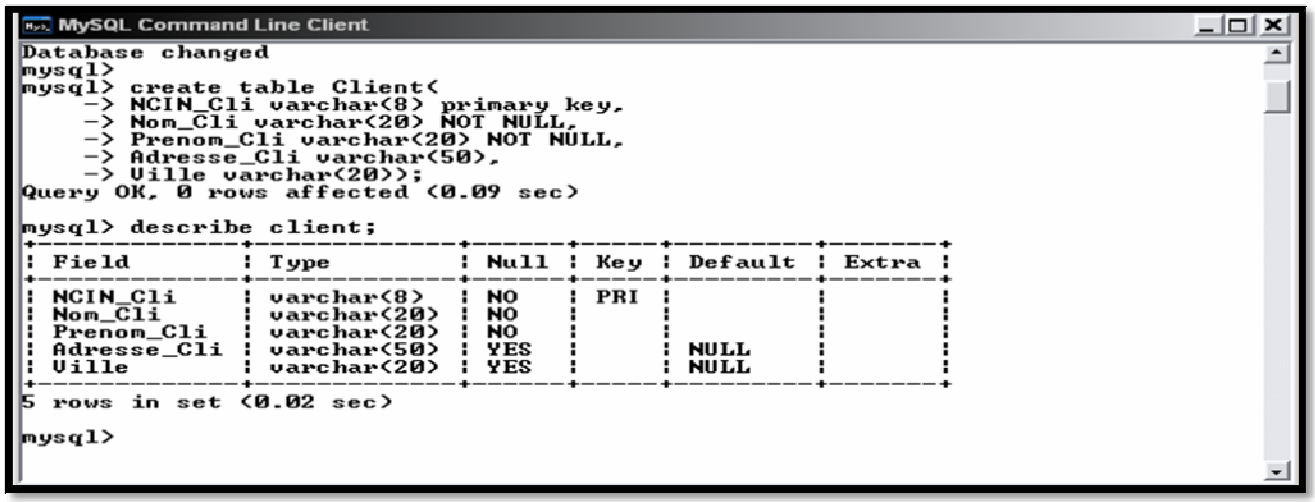
❖ Table Location

```
CREATE TABLE Location(
immat_Vehicule VARCHAR(10) REFERENCES vehicule(immat_Vehicule)ON DELETE
CASCADE,
NCIN_Cli VARCHAR(8) REFERENCES Client(NCIN_Cli)ON DELETE CASCADE,
date_loc TIMESTAMP DEFAULT CURRENT_TIMESTAMP CHECK(date_loc<=NOW()),
duree_loc INT(3) CHECK(duree_loc>0),
cout_loc DECIMAL(8,3) CHECK(cout_loc>0),
CONSTRAINT PK_Location PRIMARY KEY(immat_Vehicule,NCIN_Cli));
```

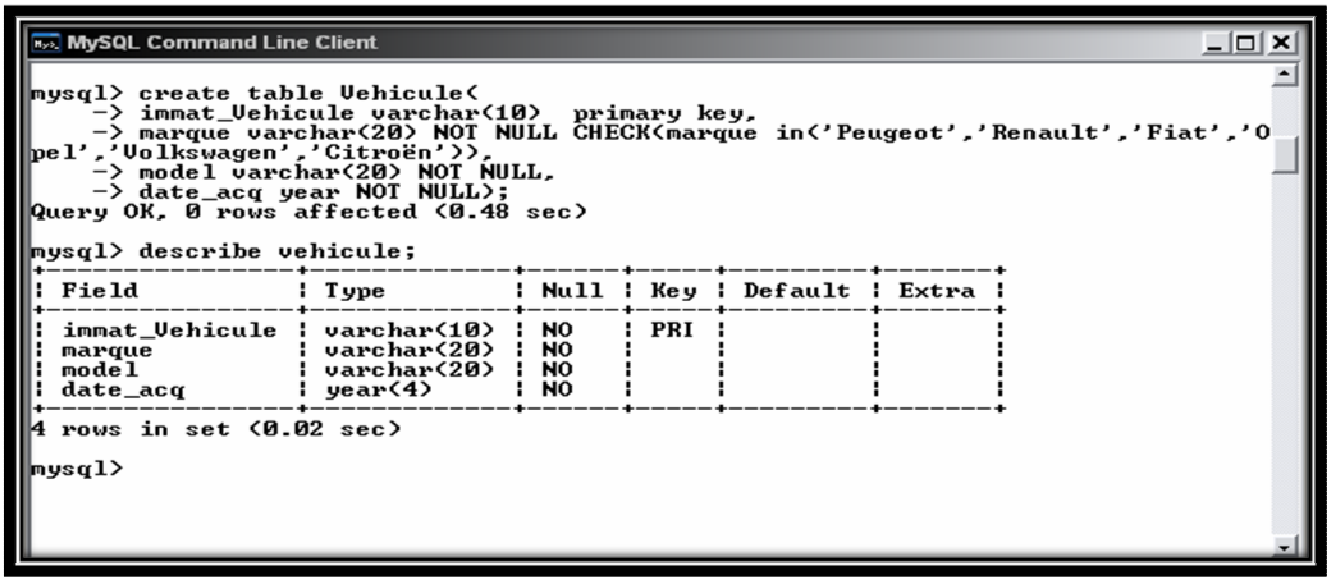
NB: Pour attribuer la date d'aujourd'hui comme valeur par défaut, il faut utiliser le type "timestamp" sinon ça ne marche pas avec le type date.

Illustrations graphiques de la création de différentes tables:

❖ Table Client



❖ Table Vehicule



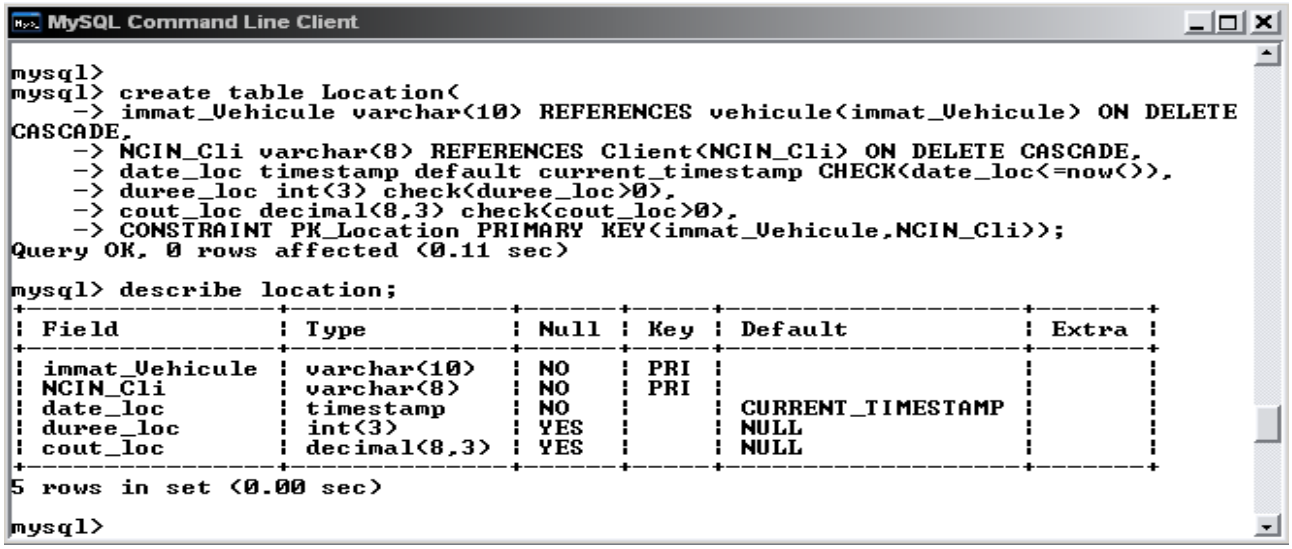
```
MySQL Command Line Client

mysql> create table Vehicule(
    -> immat_Vehicule varchar(10)  primary key,
    -> marque varchar(20) NOT NULL CHECK(marque in('Peugeot','Renault','Fiat','O
pel','Volkswagen','Citroën')),
    -> model varchar(20) NOT NULL,
    -> date_acq year NOT NULL);
Query OK, 0 rows affected (0.48 sec)

mysql> describe vehicule;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| immat_Vehicule | varchar(10) | NO   | PRI |         |       |
| marque         | varchar(20) | NO   |     |         |       |
| model          | varchar(20) | NO   |     |         |       |
| date_acq       | year(4)     | NO   |     |         |       |
+----------------+-------------+------+-----+---------+-------+
4 rows in set (0.02 sec)

mysql>
```

❖ **Table Location**



```
MySQL Command Line Client

mysql>
mysql> create table Location(
    -> immat_Vehicule varchar(10) REFERENCES vehicule(immat_Vehicule) ON DELETE
CASCADE,
    -> NCIN_Cli varchar(8) REFERENCES Client(NCIN_Cli) ON DELETE CASCADE,
    -> date_loc timestamp default current_timestamp CHECK(date_loc<=now()),
    -> duree_loc int(3) check(duree_loc>0),
    -> cout_loc decimal(8,3) check(cout_loc>0),
    -> CONSTRAINT PK_Location PRIMARY KEY(immat_Vehicule,NCIN_Cli));
Query OK, 0 rows affected (0.11 sec)

mysql> describe location;
+----------------+--------------+------+-----+---------------------+-------+
| Field          | Type         | Null | Key | Default             | Extra |
+----------------+--------------+------+-----+---------------------+-------+
| immat_Vehicule | varchar(10)  | NO   | PRI |                     |       |
| NCIN_Cli       | varchar(8)   | NO   | PRI |                     |       |
| date_loc       | timestamp    | NO   |     | CURRENT_TIMESTAMP   |       |
| duree_loc      | int(3)       | YES  |     | NULL                |       |
| cout_loc       | decimal(8,3) | YES  |     | NULL                |       |
+----------------+--------------+------+-----+---------------------+-------+
5 rows in set (0.00 sec)

mysql>
```

3) Modification de la structure d'une table

❖ **Syntaxe de modification de la structure d'une table :**

ALTER TABLE nom-table
(ADD [COLUMN] (nom-col type-col [DEFAULT valeur] [contrainte-col])*,
|ADD [CONSTRAINT] (contrainte_table)*,
| **MODIFY** [COLUMN] nom-col [type-col] [DEFAULT valeur] [contrainte-col],
| **MODIFY** [CONSTRAINT] nom_contrainte
| **DROP** COLUMN nom-col ,
| **DROP** {PRIMARY KEY
| CONSTRAINT nom_contrainte }
|**RENAME TO** <nouveau_nom_table>
);

❖ **Syntaxe de suppression d'une table :**

```
DROP TABLE nom_table;
```

a) Ajout de colonnes à une table

Ajouter à la table **Client** un champ nommé **Tel_Cli** de type texte et de taille égale à 10.

```
ALTER TABLE Client
ADD COLUMN Tel_Cli varchar(10);
```

NB: *l'utilisation de () après le mot clé "COLUMN" est facultatif quand il s'agit de l'ajout d'une seule colonne.*

b) Modification d'une colonne de la table

Modifier le type de donnée de la colonne "**Tel_Ci**" de la table "**Client**" en numérique de 8 chiffres.

```
ALTER TABLE Client
MODIFY COLUMN Tel_Cli INT(8);
```

c) Suppression des colonnes d'une table

Supprimer la colonne "**Adresse_Cli**" dans la table "**Client**".

```
ALTER TABLE Client
DROP COLUMN Adresse_Cli;
```

d) Ajout d'une contrainte

Vérifier que les valeurs de "**Annee_Acq**" dans la table "**Vehicule**" est entre 1990 et 2008.

```
ALTER TABLE Vehicule
ADD CONSTRAINT CHECK Annee_Acq BETWEEN 1990 AND 2008;
```

Remarque: *Si on veut attribuer la ville "tunis" comme valeur par défaut à tous les clients, il ne s'agit pas d'ajouter une contrainte mais de modifier la colonne* ***"Ville"***

```
ALTER TABLE Client
MODIFY COLUMN Ville VARCHAR(20) DEFAULT 'tunis';
```

e) Modification de la clé primaire d'une table

Modifier la clé primaire actuelle de la table Location par la clé primaire composée suivante: **Immat_Vehicule, NCIN, Date_Loc**

```
ALTER TABLE Location
DROP PRIMARY KEY
ADD PRIMARY KEY (immat_Vehicule,NCIN_Cli,date_loc);
```

f) Suppression d'une contrainte

Supprimer la clé primaire de la table "**Client**"

```
ALTER TABLE Client
DROP PRIMARY KEY;
```

g) Activer/désactiver une contrainte

Désactiver la contrainte de clé primaire de la table "**client**"

```
ALTER TABLE Client
DISABLE KEYS;
```

Réactiver cette même contrainte

```
ALTER TABLE Client
ENABLE KEYS;
```

h) Renommer une table

Renommer la table "**Vehicule**" en "**Voiture**"

```
ALTER TABLE Vehicule
RENAME TO Voiture;
```

i) Suppression d'une table

Supprimer la table "**Client**"

```
DROP TABLE Client;
```

j) Suppression de la base de données

Supprimer la base de données "**Gestion_Location**"

```
DROP DATABASE Gestion_Location;
```