

# Base de données

## TP n°2

### Mise en situation

Le gérant de l'agence de location de voitures, du premier TP, vous appelle furieux. L'utilisation de la base de données `location_voitures` n'a pas résolu ses problèmes, pire elle en a ajouté d'autres :

- La redondance de données :
  - les informations des véhicules et celles des clients se répètent.
  - les noms de certains clients se répètent d'une façon différente empêchant ainsi les recherches.
- Il a un nouveau client appelé « Youssef » et il est incapable de l'ajouter dans la base de données car il n'a pas encore fait de location.
- Il ne peut pas supprimer le véhicule dont le matricule est « 207TU8780 », qui n'est plus disponible, car il ne veut pas supprimer les informations des clients qui l'ont loués.
- La table accepte l'insertion de locations ayant une `date_retour` vide, une `date_location` vide. Pire, encore, elle accepte que la `date_retour` soit antérieure à la `date_location`.
- Aussi, la colonne `jours` accepte n'importe quelle valeur hors elle doit correspondre exactement au nombre de jours depuis `date_location` et jusqu'à `date_retour`.

```
MariaDB [location_voitures]> SELECT * FROM locations;
```

matricule	vehicule	prix_unit	client	tel	date_location	date_retour	jours	prix_a_payer
205TU6551	Seat IBIZA	60	Mahmoud	56504673	2022-01-03	2022-01-05	2	120
207TU4480	Suzuki Swift-BVA	80	Meriem	31827055	2022-01-05	2022-01-07	2	160
204TU9333	Seat IBIZA	60	Mohammed	56542559	2022-01-21	2022-01-30	9	540
205TU6551	Seat IBIZA	60	Mouna	36774811	2022-01-25	2022-02-04	10	600
205TU9334	Clio 5	70	Meriem	31827055	2022-01-27	2022-02-01	2	120
204TU9333	Seat IBIZA	60	Meriem	59380983	2022-01-30	2022-02-01	2	120
207TU4480	Suzuki Swift-BVA	80	Mohamed	25848116	2022-02-04	2022-02-05	1	80
207TU8780	Clio 5	70	Mouhamed	97793055	2022-02-20	2022-02-25	5	350
NULL	NULL	NULL	Youssef	NULL	NULL	NULL	NULL	NULL
205TU5651	Seat IBIZA	60	Med	25848116	2022-02-01	2022-01-30	25	100

10 rows in set (0.000 sec)

Figure 1, Problèmes identifiés dans la base de données location de voitures

Le gérant de cette agence dit qu'il vient juste de remplacer Microsoft Excel par une base de données qui ne satisfait pas ses besoins.

**Essayez de le convaincre tout en lui proposant les corrections nécessaires.**

### Solution

Il est important de noter que les bases de données sont des outils puissants qui peuvent aider à gérer efficacement les données d'une entreprise lorsqu'elles sont utilisées correctement.

Pour résoudre les problèmes que identifiés, voici quelques suggestions de corrections que vous pourriez mettre en œuvre :

- Créer deux tables « clients » et « vehicules » et ce pour pouvoir :
  - ajouter de nouveaux clients qui n'ont pas encore fait de location sans avoir à ajouter des enregistrements de location fictifs.
  - supprimer les véhicules qui ne sont plus disponibles sans avoir à supprimer les données des clients.
- Utiliser des clés primaires dans les tables `vehicules` et `clients` pour identifier les véhicules et respectivement les clients dans ces tables
- Utiliser des contraintes de domaine pour empêcher l'insertion de locations avec des dates manquantes ou des dates de retour antérieures à la date de location
- Calculer le nombre de jours de location, à partir de `date_location` et `date_retour`, plutôt que de permettre aux utilisateurs de saisir cette valeur manuellement.
- Créer une table troisième table nommée `locations` qui permet d'enregistrer les locations de véhicules tout en évitant les redondances. Les tables `vehicules` et `clients` sont liées à cette table en utilisant des clés étrangères qui référencent les clés primaires des autres tables.

En mettant en place ces corrections, nous serons en mesure de résoudre tous les problèmes et d'améliorer l'efficacité de notre base de données de location de voitures.

## Partie Théorique

Faire une recherche sur les thèmes suivants :

- Clé primaire
- Relations
- Contraintes d'intégrité
- Schéma d'une base de données

Vous pourrez consulter le cours de base de données pour répondre à ces questions ou faire une recherche sur Internet.

## Partie Pratique

1. Créer une nouvelle base de données appelée `location_vehicules`. Puis, l'utiliser.

```
-- Création de la base de données
CREATE DATABASE location_vehicules;
-- Utiliser la base de données locatiion_vehicules
USE location_vehicules;
```

2. Le gérant de l'agence de location de voitures a besoin des informations suivantes :

Colonne	Description	Type de données	Taille	Oblig.	Val. par déf.	Val. auto.
matricule	Matricule de la voiture	Texte	10	✓		
vehicule	Marque de la voiture	Texte	32	✓		
prix_unit	Prix de location pour une journée	Numérique réel		✓	0.0	≥ 0.0
libre	Indique si le véhicule est disponible pour la locations	Texte	1	✓	"O"	"O" ou "N"
cin	Numéro de la carte d'identité du client	Texte	10	✓		
client	Nom du client	Texte	32	✓		
genre	Genre du client (Homme / Femme)	Texte	1	✓	"M"	"M" ou "F"
tel	Téléphone du client	Texte	16	✓		
num_location	Numéro de location	Entier		✓	Auto incrémenté	
date_location	Date de location du véhicule	Date		✓	Date du jour	
date_retour	Date de retour du véhicule	Date		✓		date_retour > date_location
montant_location	Prix total de location du véhicule	Numérique réel		✓	0.0	≥ 0.0

On demande de regrouper ces colonnes dans plusieurs entités distinctes.

## Solution

On peut dégager trois entités, correspondants aux objets du monde réel, à partir du tableau précédent :

- Entité **vehicules** qui rassemble les propriétés d'un véhicule.
- Entité **clients** qui rassemble les propriétés d'un client.
- Entité **locations** qui rassemble les propriétés d'une location effectuée par un certain client à une date bien déterminée.

Chacune des entités correspond à une table :

vehicules(matricule, vehicule, prix\_unit, libre)

clients(cin, client, genre, tel)

location(num\_location, **cin#**, **matricule#**, date\_location, date\_retour, montant\_location)

Les champs soulignés sont les clés primaires qui permettent d'identifier les n-uplets d'une façon unique.

Les champs **gras suivis par #** sont les clés étrangères qui permettent de relier la table **locations** aux autres tables, ainsi :

- Un client pourra effectuer un ou plusieurs locations.
- Un véhicule pourra être loué une ou plusieurs fois.

### 3. Créer la table **vehicules**.

Les enregistrements de la table sont identifiés par la colonne **matricule**.

```
CREATE TABLE vehicules (
    matricule VARCHAR(10) PRIMARY KEY NOT NULL,
    vehicule VARCHAR(32) NOT NULL,
    prix_unit DOUBLE NOT NULL DEFAULT 0.0,
    libre CHAR(1) DEFAULT '0' CHECK (libre = '0' OR libre = 'N')
);
```

### 4. Créer la table **clients**.

La table **clients** est identifiée par la colonne **cin**.

```
CREATE TABLE clients (
    cin VARCHAR(10) PRIMARY KEY,
    client VARCHAR(32) NOT NULL,
    genre CHAR(1) NOT NULL DEFAULT 'M' CHECK(genre IN ('M', 'F')),
    tel VARCHAR(16) NOT NULL
);
```

### 5. Créer la table **locations**.

La table est identifiée par le champ **num\_location**.

```
CREATE TABLE locations (
    num_location INT PRIMARY KEY AUTO_INCREMENT,
    matricule VARCHAR(10) NOT NULL,
    cin VARCHAR(10) NOT NULL,
    date_location DATE NOT NULL DEFAULT NOW(),
    date_retour DATE NOT NULL CHECK(date_retour > date_location),
    montant_location DOUBLE NOT NULL CHECK(montant_location >= 0.0),
    CONSTRAINT fk_location_1 FOREIGN KEY (matricule)
        REFERENCES vehicules (matricule)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT fk_location_2 FOREIGN KEY (cin)
        REFERENCES clients (cin)
        ON UPDATE CASCADE
        ON DELETE CASCADE);
```

6. Remplir la table **clients** par les données suivantes :

cin	client	genre	tel
02500463	Hajer	F	36079515
03972583	Amir	M	36704253
04202363	Aziz	M	70116709

```
INSERT INTO clients (cin, client, genre, tel)
VALUES ('03972583', 'Amir', 'M', '36704253');
```

7. Remplir la table **vehicules** par les données suivantes :

matricule	vehicule	prix_unit	libre
200TU4906	Clio 4	60	0
203TU4480	Suzuki Swift-BVA	80	0

#### Requête d'insertion

```
INSERT INTO vehicules (matricule, vehicule, prix_unit, libre)
VALUES ('203TU8041', 'Clio 4', '60', '0'),
('204TU9333', 'Seat IBIZA', '60', 'N'),
('205TU6551', 'Seat IBIZA', '60', 'O'),
('205TU9334', 'Clio 5', '70', 'N'),
('206TU5473', 'Suzuki Swift-BVA', '80', 'O'),
('206TU9804', 'Clio 4', '60', 'O'),
('207TU4480', 'Suzuki Swift-BVA', '80', 'O'),
('207TU8780', 'Clio 5', '70', 'O');
```

```
INSERT INTO vehicules (matricule, vehicule, prix_unit, libre)
VALUES ('200TU4906', 'Clio 4', 60, 'O');
```

8. Remplir la table **locations** par les données suivantes :

num_location	matricule	cin	date_location	date_retour	montant_location
1	203TU4480	03972583	2022-01-02	2022-01-03	80
2	206TU5473	04202363	2022-01-02	2022-01-03	80

On remarque qu'au lieu de saisir les informations des clients et des véhicules à chaque fois il nous suffit juste d'insérer le **cin** du client et le **matricule** du véhicule loué.

#### Requête d'insertion

```
INSERT INTO locations (num_location, matricule, cin, date_location, date_retour, montant_location)
VALUES ('3', '203TU8041', '04202363', '2022-01-05', '2022-01-06', '60'),
('4', '203TU8041', '03972583', '2022-01-06', '2022-01-09', '180'),
('5', '205TU6551', '04202363', '2022-01-07', '2022-01-08', '60'),
('6', '206TU9804', '04202363', '2022-01-08', '2022-01-26', '1080'),
('7', '207TU4480', '04202363', '2022-01-10', '2022-01-12', '160'),
('8', '200TU4906', '03972583', '2022-01-13', '2022-01-14', '60'),
('9', '204TU9333', '03972583', '2022-01-15', '2022-01-16', '60'),
('10', '207TU4480', '04202363', '2022-01-18', '2022-01-19', '80'),
('11', '206TU5473', '04202363', '2022-01-21', '2022-01-22', '80'),
('12', '207TU4480', '04202363', '2022-01-24', '2022-01-25', '80'),
('13', '205TU9334', '04202363', '2022-01-28', '2022-02-03', '420'),
('14', '204TU9333', '03972583', '2022-01-28', '2022-02-01', '240');
```

```
INSERT INTO locations (matricule, cin, date_location, date_retour, montant_location)
VALUES ('203TU4480', '03972583', '2022-01-02', '2022-01-03', 80);
```

9. Ajouter le nouveau client « Youssef » son **cin** est « 12569800 » et son **téléphone** est « 52349157 ».

```
INSERT INTO clients (cin, client, genre, tel)
VALUES ('12569800', 'Youssef', 'M', 52349157);
```

10. Le gérant a voulu insérer l'enregistrement suivant d'une nouvelle cliente (**client**: 'Assia', **cin**: '12650890', **tel**: '52349157') en utilisant la commande suivante :

```
INSERT INTO clients (client, cin)
VALUES ('Assia', '12650890');
```

Mais il obtient l'erreur suivante :

ERROR 1364 (HY000): Field 'tel' doesn't have a default value

### Quel est le problème ? Comment le corriger ?

Le champ **tel** est obligatoire le gérant doit l'indiquer pour corriger le problème.

11. Le gérant a corrigé sa commande en tapant :

```
INSERT INTO clients (client, cin, tel)
VALUES ('Assia', '12650890', '52349157');
```

Puis, pour vérifier qu'elle a été insérée :

```
SELECT * FROM clients WHERE cin = '12650890';
```

```
+-----+-----+-----+-----+
| cin    | client | genre | tel    |
+-----+-----+-----+-----+
| 12650890 | Assia  | M     | 52349157 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

### Pourquoi le genre de « Assia » est-il incorrect ? Comment le corriger ?

Comme la requête n'indique pas le **genre** de « Assia » le SGBD insère la valeur par défaut **M**.

Pour corriger ce problème il faut mettre à jour ce champ à l'aide de la commande :

```
UPDATE clients SET genre = 'F' WHERE cin = '12650890';
```

12. Afficher la liste des clientes de cette agence par ordre croissant de leurs noms.

```
SELECT * FROM clients WHERE genre = 'F' ORDER BY client;
```

13. Afficher la liste des clients qui disposent d'un numéro de téléphone Orange (commençant par 5) ou Tunisie Telecom (commençant par 7 ou 9).

```
SELECT * FROM clients WHERE LEFT(tel, 1) IN ('5', '7', '9');
```

14. Le client « Aziz » vient de changer son numéro de téléphone à 25025637. Mettre à jour les informations de « Aziz ».

```
UPDATE clients SET tel = '25025637' WHERE cin = '04202363';
```

15. Afficher la liste des véhicules 'Clio 5' libres.

```
SELECT * FROM vehicules WHERE vehicule = 'Clio 5' AND libre = '0';
```

16. Ce véhicule a été loué à « Youssef » pendant 5 jours du 01/02/2022 au 06/02/2022.

**Insérer cette location, et mettre à jour la liste des véhicules. Vérifier le bon déroulement de l'opération.**

```
-- Insérer cette location,  
INSERT INTO locations (matricule, cin, date_location, date_retour, montant_location)  
VALUES ('207TU8780', '12569800', '2022-02-01', '2022-02-06', 5 * 70.0);  
  
-- et mettre à jour la liste des véhicules.  
UPDATE vehicules SET libre = 'N' WHERE matricule = '207TU8780';  
  
-- Vérifier le bon déroulement de l'opération  
SELECT * FROM vehicules WHERE matricule = '207TU8780';  
SELECT * FROM locations WHERE matricule = '207TU8780';
```

17. Oubliant que le véhicule « Seat IBIZA » ayant comme matricule « 204TU9333 » n'est pas libre. Le gérant l'a loué à « Assia ». Il a tapé la commande suivante :

```
INSERT INTO locations (matricule, cin, date_location, date_retour, montant_location)  
VALUES ('204TU9333', '12650890', '2022-02-01', '2022-02-03', 2 * 60.0);
```

Maintenant, il veut annuler cette location en supprimant cet enregistrement. **Comment vas-t-il procéder ?**

Il doit trouver le dernier `num_location` inséré dans la table `locations` à l'aide de la commande :

```
SELECT MAX(num_location) FROM locations;
```

Puis, supprimer l'enregistrement incorrect à l'aide de la commande :

```
DELETE FROM locations WHERE num_location = 16;
```