

# Rapport : Gestion des données notariales avec MySQL

**Module** : Base de Données MySQL

**Nom**: Aya ouazzani akkar hammoudi

## SOMMAIRE :

- Réaliser les ateliers sous MySQL (création des tables, procédures, fonctions, triggers).
- des captures d'écran montrant l'exécution des requêtes

### 1)Création de la base et des tables :

#### Création de la base

The screenshot displays two successful MySQL command executions. The first command creates the database 'gestion\_notariale' with the character set 'utf8mb4' and the collation 'utf8mb4\_unicode\_ci'. The second command uses the 'gestion\_notariale' database. Both commands result in an empty output, indicating successful execution. Below each command block, there is an error message: 'Error: #1046 Aucune base n'a été sélectionnée'.

```
1 CREATE DATABASE IF NOT EXISTS gestion_notariale
2 CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
3
4 USE gestion_notariale;
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0035 seconde(s).)

CREATE DATABASE IF NOT EXISTS gestion\_notariale CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

⚠ Error: #1046 Aucune base n'a été sélectionnée

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

USE gestion\_notariale;

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

⚠ Error: #1046 Aucune base n'a été sélectionnée

#### Création des tables

```

5  -- Cabinet
6  CREATE TABLE IF NOT EXISTS cabinet (
7      id_cab VARCHAR(50) PRIMARY KEY,
8      nom VARCHAR(100) NOT NULL,
9      adresse VARCHAR(255),
10     tel VARCHAR(50),
11     ville VARCHAR(100)
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
13
14 -- Notaire
15 CREATE TABLE IF NOT EXISTS notaire (
16     numn VARCHAR(50) PRIMARY KEY,
17     nom VARCHAR(100) NOT NULL,
18     prenom VARCHAR(100) NOT NULL,
19     agc VARCHAR(100),
20     tel VARCHAR(50),
21     email VARCHAR(150),
22     id_cab VARCHAR(50),
23     FOREIGN KEY (id_cab) REFERENCES cabinet(id_cab)
24         ON DELETE SET NULL ON UPDATE CASCADE
25 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

26 -- Vendeur
27 CREATE TABLE IF NOT EXISTS vendeur (
28     ciny VARCHAR(50) PRIMARY KEY,
29     nom VARCHAR(100),
30     prenom VARCHAR(100),
31     adresse VARCHAR(255),
32     ddn DATE, -- تاريخ الازدياد
33     tel VARCHAR(50)
34 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
35
36 -- Terrain
37 CREATE TABLE IF NOT EXISTS terrain (
38     numter VARCHAR(50) PRIMARY KEY,
39     typeTerrain VARCHAR(50),
40     prixvente DECIMAL(15,2),
41     cinv VARCHAR(50),
42     titre_foncier VARCHAR(100),
43     adresse VARCHAR(255),
44     ville VARCHAR(100),
45     superficie DECIMAL(10,2),
46     FOREIGN KEY (cinv) REFERENCES vendeur(ciny)
47         ON DELETE SET NULL ON UPDATE CASCADE

```

```

44     superficie DECIMAL(10,2),
45     FOREIGN KEY (cinv) REFERENCES vendeur(ciny)
46     ON DELETE SET NULL ON UPDATE CASCADE
47 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
48
49 -- Dossier_terrain
50 CREATE TABLE IF NOT EXISTS dossier_terrain (
51     iddossier VARCHAR(50) PRIMARY KEY,
52     date_ouverture DATE NOT NULL,
53     date_cloture DATE,
54     numter VARCHAR(50),
55     numnotaire VARCHAR(50),
56     etat VARCHAR(20),
57     FOREIGN KEY (numter) REFERENCES terrain(numter)
58     ON DELETE RESTRICT ON UPDATE CASCADE,
59     FOREIGN KEY (numnotaire) REFERENCES notaire(numn)
60     ON DELETE RESTRICT ON UPDATE CASCADE
61 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0209 seconde(s).)

```
-- Cabinet CREATE TABLE IF NOT EXISTS cabinet ( id_cab VARCHAR(50) PRIMARY KEY, nom VARCHAR(100) NOT NULL, adresse VARCHAR(255), tel VARCHAR(50), ville VARCHAR(100) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0413 seconde(s).)

```
-- Notaire CREATE TABLE IF NOT EXISTS notaire ( numn VARCHAR(50) PRIMARY KEY, nom VARCHAR(100) NOT NULL, prenom VARCHAR(100) NOT NULL, agc VARCHAR(100), tel VARCHAR(50), email VARCHAR(150), id_cab VARCHAR(50), FOREIGN KEY (id_cab) REFERENCES cabinet(id_cab) ON DELETE SET NULL ON UPDATE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0088 seconde(s).)

```
-- 3Vendeur CREATE TABLE IF NOT EXISTS vendeur ( ciny VARCHAR(50) PRIMARY KEY, nom VARCHAR(100), prenom VARCHAR(100), adresse VARCHAR(255), ddn DATE, -- تاريخ الزيد tel VARCHAR(50) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0396 seconde(s).)

```
-- Terrain CREATE TABLE IF NOT EXISTS terrain ( numter VARCHAR(50) PRIMARY KEY, typeTerrain VARCHAR(50), prixvente DECIMAL(15,2), cinv VARCHAR(50), titre_foncier VARCHAR(100), adresse VARCHAR(255), ville VARCHAR(100), superficie DECIMAL(10,2), FOREIGN KEY (cinv) REFERENCES vendeur(ciny) ON DELETE SET NULL ON UPDATE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0395 seconde(s).)

```
-- Dossier_terrain CREATE TABLE IF NOT EXISTS dossier_terrain ( iddossier VARCHAR(50) PRIMARY KEY, date_ouverture DATE NOT NULL, date_cloture DATE, numter VARCHAR(50), numnotaire VARCHAR(50), etat VARCHAR(20), FOREIGN KEY (numter) REFERENCES terrain(numter) ON DELETE RESTRICT ON UPDATE CASCADE, FOREIGN KEY (numnotaire) REFERENCES notaire(numn) ON DELETE RESTRICT ON UPDATE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

La requête SQL a été exécutée avec succès.

[SHOW TABLES;](#)

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

Options supplémentaires

**Tables\_in\_gestion\_notariale**

cabinet

dossier\_terrain

notaire

terrain

vendeur

## 2) Insertion des données initiales :

```
1
2 -- Cabinets
3 INSERT INTO cabinet (id_cab, nom, adresse, tel, ville) VALUES
4 ('CAB01', 'Cabinet A', 'Rue 1', '0522000001', 'Casablanca'),
5 ('CAB02', 'Cabinet B', 'Rue 2', '0522000002', 'Rabat');
6
7 -- Notaires
8 INSERT INTO notaire (numn, nom, prenom, agc, tel, email, id_cab) VALUES
9 ('N001', 'El Amrani', 'Said', 'agc1', '0611223344', 'said@example.com', 'CAB01'),
10 ('N002', 'Benchebkroun', 'Amina', 'agc2', '0611223345', 'amina@example.com', 'CAB01'),
11 ('N003', 'Karim', 'Hassan', 'agc3', '0611223346', 'karim@example.com', 'CAB02');
12
13 -- Vendeurs
14 INSERT INTO vendeur (ciny, nom, prenom, adresse, ddn, tel) VALUES
15 ('CIN001', 'Ouazzani', 'Aya', 'Av 3', '1995-04-12', '0600000001');
16
17 -- Terrains
18 INSERT INTO terrain (numter, typeTerrain, prixvente, cinv, titre_foncier, adresse, ville, superficie) VALUES
19 ('T001', 'Terrain agricole', 50000, 'CIN001', 'TF001', 'Douar X', 'Casablanca', 1200.50),
20 ('T002', 'terrain en lotissement', 80000, 'CIN001', 'TF002', 'Quartier Y', 'Casa', 500.00);
21
22 -- Dossiers
23 INSERT INTO dossier_terrain (iddossier, date_ouverture, date_cloture, numter, numnotaire, etat) VALUES
24 ('D001', '2004-12-01', '2005-02-15', 'T001', 'N001', 'traité'),
25 ('D002', '2005-03-01', '2005-04-10', 'T002', 'N002', 'traité'),
26 ('D003', '2006-01-10', NULL, 'T001', 'N001', 'nontraité'),
27 ('D004', '2003-05-20', '2003-06-01', 'T002', 'N003', 'traité'),
28 ('D005', '2024-08-01', NULL, 'T002', 'N002', 'nontraité');
```

```
1 SELECT * FROM cabinet;
2 SELECT * FROM notaire;
3 SELECT * FROM terrain;
4 SELECT * FROM dossier_terrain;
```

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0008 seconde(s).)

`SELECT * FROM cabinet;`

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Expliquer SQL](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

			id_cab	nom	adresse	tel	ville
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	CAB01
				Cabinet A	Rue 1	0522000001	Casablanca
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	CAB02
				Cabinet B	Rue 2	0522000002	Rabat
<input type="checkbox"/>		<a href="#">Éditer</a>		<a href="#">Copier</a>		<a href="#">Supprimer</a>	CAB03
				Cabinet C	Rue 3	0522000003	Casablanca

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0004 seconde(s).)

`SELECT * FROM notaire;`

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Expliquer SQL](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

<div><div></div><div></div><div></div></div>			numn	nom	prenom	agc	tel	email	id_cab				
<input type="checkbox"/>		Éditer		Copier		Supprimer	N001	El Amrani	Said	agc1	0611223344	said@example.com	NULL
<input type="checkbox"/>		Éditer		Copier		Supprimer	N002	Bencheikroun	Amina	agc2	0611223345	amina@example.com	NULL
<input type="checkbox"/>		Éditer		Copier		Supprimer	N003	Karim	Hassan	agc3	0611223346	karim@example.com	NULL

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0004 seconde(s).)

`SELECT * FROM terrain;`

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Expliquer SQL](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

<div>←T→</div>		▼	numter	typeTerrain	prixvente	cinv	titre_foncier	adresse	ville	superfic
<input type="checkbox"/>	<div><div><div>✎ Éditer</div><div>📄 Copier</div><div>🗑 Supprimer</div></div></div>		T001	Terrain agricole	50000.00	CIN001	TF001	Douar X	Casablanca	
<input type="checkbox"/>	<div><div><div>✎ Éditer</div><div>📄 Copier</div><div>🗑 Supprimer</div></div></div>		T002	terrain en lotissement	80000.00	CIN001	TF002	Quartier Y	Casa	

☐ Tout cocher | Avec la sélection : Éditer Copier Supprimer Exporter



✓ Affichage des lignes 0 - 4 (total de 5, traitement en 0,0004 seconde(s).)

```
SELECT * FROM dossier_terrain;
```

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

		iddossier	date_ouverture	date_cloture	numter	numnotaire	etat
<input type="checkbox"/>	Éditer Copier Supprimer	D001	2004-12-01	2005-02-15	T001	N001	traité
<input type="checkbox"/>	Éditer Copier Supprimer	D002	2005-03-01	2005-04-10	T002	N002	traité
<input type="checkbox"/>	Éditer Copier Supprimer	D003	2006-01-10	NULL	T001	N001	nontraité
<input type="checkbox"/>	Éditer Copier Supprimer	D004	2003-05-20	2003-06-01	T002	N003	traité
<input type="checkbox"/>	Éditer Copier Supprimer	D005	2024-08-01	NULL	T002	N002	nontraité

↑ ☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

**Afficher tous les cabinets de la ville Casablanca qui ont clôturé des dossiers entre le 2 janvier et le 31 mai 2005.:**

```
1 SELECT DISTINCT c.id_cab, c.nom, c.ville
2 FROM cabinet c
3 JOIN notaire n ON c.id_cab = n.id_cab
4 JOIN dossier_terrain d ON n.numn = d.numnotaire
5 WHERE c.ville = 'Casablanca'
6 AND d.date_cloture BETWEEN '2005-01-02' AND '2005-05-31';
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0009 seconde(s).)

```
SELECT DISTINCT c.id_cab, c.nom, c.ville FROM cabinet c JOIN notaire n ON c.id_cab = n.id_cab JOIN dossier_terrain d ON n.numn = d.numnotaire WHERE c.ville = 'Casablanca' AND d.date_cloture BETWEEN '2005-01-02' AND '2005-05-31';
```

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

id\_cab nom ville

Opérations sur les résultats de la requête

Créer une vue

**4) Écrire une fonction qui retourne le nombre de notaires qui n'ont traité aucun dossier concernant un type de terrain passé en paramètre.:**

```

1 DELIMITER $$
2 CREATE FUNCTION nb_notaires_sans_dossier(typeT VARCHAR(50))
3 RETURNS INT
4 DETERMINISTIC
5 BEGIN
6     DECLARE nb INT;
7     SELECT COUNT(*) INTO nb
8     FROM notaire n
9     WHERE n.numn NOT IN (
10         SELECT DISTINCT d.numnotaire
11         FROM dossier_terrain d
12         JOIN terrain t ON d.numter = t.numter
13         WHERE t.typeTerrain = typeT
14     );
15     RETURN nb;
16 END $$
17 DELIMITER ;
18

```

SELECT \* SELECT INSERT UPDATE DELETE Effacer Format Récupérer la requête auto-sauvegardée

Console de requêtes SQL

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0174 seconde(s).)

```

CREATE FUNCTION nb_notaires_sans_dossier(typeT VARCHAR(50)) RETURNS INT DETERMINISTIC BEGIN DECLARE nb INT; SELECT COUNT(*) INTO nb FROM notaire n WHERE n.numn NOT IN ( SELECT DISTINCT d.numnotaire FROM dossier_terrain d JOIN terrain t ON d.numter = t.numter WHERE t.typeTerrain = typeT ); RETURN nb; END;

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## 5) Afficher les notaires (nom, prénom) qui ont traité plus de 200 dossiers.:

```

1 DELIMITER $$
2 CREATE PROCEDURE notaires_plus_200()
3 BEGIN
4     SELECT n.nom, n.prenom, COUNT(d.iddossier) AS nb_dossiers
5     FROM notaire n
6     JOIN dossier_terrain d ON n.numn = d.numnotaire
7     GROUP BY n.numn, n.nom, n.prenom
8     HAVING COUNT(d.iddossier) > 200;
9 END $$
10 DELIMITER ;
11
12 CALL notaires_plus_200();
13

```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0131 seconde(s).)

```

CREATE PROCEDURE notaires_plus_200() BEGIN SELECT n.nom, n.prenom, COUNT(d.iddossier) AS nb_dossiers FROM notaire n JOIN dossier_terrain d ON n.numn = d.numnotaire GROUP BY n.numn, n.nom, n.prenom HAVING COUNT(d.iddossier) > 200; END;

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0013 seconde(s).)

```

CALL notaires_plus_200();

```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

## 6. Empêcher la suppression de tout dossier sauf ceux datant de plus de 2 ans et non clôturés.

```
1 DELIMITER $$
2 CREATE TRIGGER verif_suppression_dossier
3 BEFORE DELETE ON dossier_terrain
4 FOR EACH ROW
5 BEGIN
6     DECLARE nb_jours INT;
7     SET nb_jours = DATEDIFF(CURDATE(), OLD.date_ouverture);
8
9     IF NOT (nb_jours > 730 AND OLD.etat = 'nontraité') THEN
10         SIGNAL SQLSTATE '45000'
11         SET MESSAGE_TEXT = 'Suppression interdite : dossier récent ou traité';
12     END IF;
13 END $$
14 DELIMITER ;
15
```

SELECT \* SELECT INSERT UPDATE DELETE Effacer Format Récupérer la requête auto-sauvegardée

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0104 seconde(s).)

```
CREATE TRIGGER verif_suppression_dossier BEFORE DELETE ON dossier_terrain FOR EACH ROW BEGIN DECLARE nb_jours INT; SET nb_jours = DATEDIFF(CURDATE(), OLD.date_ouverture); IF NOT (nb_jours > 730 AND OLD.etat = 'nontraité') THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Suppression interdite : dossier récent ou traité'; END IF; END;
```

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]