

Corrigé d'Examen pratique Base de Données

Répondre aux questions suivantes en utilisant des commandes SQL:

1. Lister tous les étudiants avec leur filière et leur niveau
2. Compter le nombre total d'étudiants admis dans toutes les filières
3. Compter le nombre d'étudiants de niveau 2 dans la filière rss
4. Afficher le nombre d'étudiants redoublants par filière et par niveau
5. Lister tous les étudiants de niveau 3 dans les filières dwm et dsi
6. Afficher le nombre de redoublants par filière et par niveau comme suit :

Nom filière	Niveau	total_redoublants
Dev informatique	1	10
Dev informatique	2	20
Dev informatique	3	10
...	...	...

7. Ecrire une commande qui permet de transférer l'étudiant 23033 vers la filière dsi, niveau 1 avec statut admis
8. Vérifier que votre matricule existe dans la table étudiant. S'il existe, mettez à jour les autres colonnes avec vos données réelles, sinon ajouter une ligne dans étudiant en fournissant vos informations
9. Insérer 4 notes dans la table notes pour les matières de codes *DEVI* 10, *MAIL* 10, *SYRI* 10 et *DPR110* pour vous-même (préciser votre matricule). Les notes doivent être différentes
10. Calculer la moyenne des notes pour chaque matière
11. Afficher l'étudiant avec la meilleure moyenne générale
12. Afficher l'étudiant avec la plus mauvaise moyenne générale
13. Lister les étudiants dont la moyenne est supérieure à 15

Les données :

```
-- Création (ou utilisation) de la base de données
CREATE DATABASE IF NOT EXISTS University;
USE University;

-- Suppression des tables si elles existent déjà
DROP TABLE IF EXISTS etudiant;
DROP TABLE IF EXISTS filiere;

-- Création de la table filiere avec deux colonnes : code (clé primaire) et nom
```

```

CREATE TABLE filiere (code VARCHAR(10) PRIMARY KEY,
                        nom VARCHAR(30) NOT NULL
);

-- Création de la table etudiant avec une référence vers filiere par filiere_code
CREATE TABLE etudiant (matricule INT PRIMARY KEY,
                        nom VARCHAR(30) NOT NULL,
                        filiere_code VARCHAR(10) NOT NULL,
                        niveau INT NOT NULL,
                        est_admin TINYINT(1) DEFAULT 0,
                        FOREIGN KEY (filiere_code) REFERENCES filiere(code)
);

-- Insertion des filières avec leur code et leur nom
INSERT INTO filiere (code, nom) VALUES ('dsi', 'Dev informatique'),
                                         ('rss', 'Reaseaux S et Securite'),
                                         ('dwm', 'Dev Web et Mult');

INSERT INTO etudiant (matricule,nom, filiere_code, niveau, est_admin) VALUES
(24001,'Ahmed', 'dsi', 1, 1),
(24002,'Aicha', 'dsi', 1, 1),
(24003,'Oumar', 'dsi', 1, 1),
(24004,'Mariem', 'dsi', 1, 0),
(24005,'Sidi', 'dsi', 1, 0),
(24006,'Moussa', 'dsi', 1, 0),
(24007,'Aicha', 'dsi', 1, 0),
(24008,'Ahmed', 'dsi', 1, 0),
(24009,'Khaled', 'dsi', 1, 0),
(24010,'Ahmedou', 'dsi', 1, 0),
(23010,'Alioune', 'dsi', 2, 1),
(23011,'Aicha', 'dsi', 2, 1),
(23012,'Fatimetou', 'dsi', 2, 1),
(23020,'Mariyata', 'dsi', 2, 0),
(23021,'Aminata', 'dsi', 2, 0),
(23022,'Cheikh', 'dsi', 2, 0),
(22010,'Med', 'dsi', 3, 1),
(22011,'Abdallahi', 'dsi', 3, 1),
(22012,'Brahim', 'dsi', 3, 1),
(22020,'Housseyni', 'dsi', 3, 0),
(22021,'Hassen', 'dsi', 3, 0),

```

```

(22022,'Ammar', 'dsi', 3, 0);

INSERT INTO etudiant (matricule,nom, filiere_code, niveau, est_admin) VALUES
(24030,'Ahmedou', 'rss', 1, 1),
(24031,'Alia', 'rss', 1, 1),
(24032,'Oumar', 'rss', 1, 1),
(24033,'Mariem', 'rss', 1, 0),
(24034,'Sidi', 'rss', 1, 0),
(24035,'Demba', 'rss', 1, 0),
(23030,'Alia', 'rss', 2, 1),
(23031,'Ahmed', 'rss', 2, 1),
(23032,'Sidi', 'rss', 2, 1),
(23033,'Oumar', 'rss', 2, 0),
(23034,'Mariem', 'rss', 2, 0),
(23035,'Ali', 'rss', 2, 0),
(22040,'Cheikh', 'rss', 3, 1),
(22041,'Ahmed', 'rss', 3, 1),
(22042,'Saad', 'rss', 3, 1),
(22043,'Fati', 'rss', 3, 0),
(22044,'Bah', 'rss', 3, 0),
(22045,'Ali', 'rss', 3, 0);

INSERT INTO etudiant (matricule,nom, filiere_code, niveau, est_admin) VALUES
(24060,'Alia', 'dwm', 1, 1),
(24061,'Ahmed', 'dwm', 1, 1),
(24062,'Sidi', 'dwm', 1, 1),
(24063,'Oumar', 'dwm', 1, 0),
(24064,'Mariem', 'dwm', 1, 0),
(24065,'Ali', 'dwm', 1, 0),
(23063,'Oumar', 'dwm', 2, 0),
(23064,'Mariem', 'dwm', 2, 0),
(23065,'Ali', 'dwm', 2, 0);

```

Corrigé :

1.

```
SELECT MATRICULE, FILIERE_CODE, NIVEAU FROM etudiant ;
```

2.

```
SELECT filiere_code, COUNT(MATRICULE) FROM etudiant WHERE est_admin = 1 GROUP BY
filiere_code ;
```

3.

```
SELECT COUNT(matricule) FROM etudiant WHERE niveau = 2 AND filiere_code = "rss" ;
```

4.

```
SELECT filiere_code, niveau, COUNT(matricule) FROM etudiant WHERE est_admin = 0  
GROUP BY filiere_code, niveau ;
```

5.

```
SELECT * FROM etudiant WHERE niveau = 3 AND (filiere_code = "dsi" OR filiere_code =  
"dwm") ;
```

6.

```
SELECT filiere.nom, niveau, COUNT(matricule) FROM etudiant, filiere WHERE  
est_admin = 0 AND etudiant.filiere_code = filiere.code GROUP BY filiere.nom, niveau  
ORDER BY filiere.nom ASC ;
```

7.

```
UPDATE etudiant SET filiere_CODE = "dsi", NIVEAU = 1, EST_ADMIN = 1 WHERE  
MATRICULE = 23033 ;
```

8.

```
INSERT INTO etudiant VALUES (25181, "Cheikh", "dsi", 1, 1) ;
```

9.

```
INSERT INTO notes VALUES (25181, 'DEVI10', 14),  
                           (25181, 'MAIL10', 15),  
                           (25181, 'SYRI10', 16),  
                           (25181, 'DPR110', 13) ;
```

10.

```
SELECT NOTES.MATIERES, ETUDIANT.MATRICULE, AVG(NOTES.NOTE) FROM ETUDIANT, NOTES  
GROUP BY NOTES.MATIERES, ETUDIANT.MATRICULE ;
```

11.

```
SELECT NOTES.MATIERES, ETUDIANT.MATRICULE, AVG(NOTES.NOTE) FROM ETUDIANT, NOTES  
GROUP BY NOTES.MATIERES, ETUDIANT.MATRICULE HAVING AVG(NOTES.NOTE) = MAX(NOTES.NOTE) ;
```

12.

```
SELECT NOTES.MATIERES, ETUDIANT.MATRICULE, AVG(NOTES.NOTE) FROM ETUDIANT, NOTES  
GROUP BY NOTES.MATIERES, ETUDIANT.MATRICULE HAVING AVG(NOTES.NOTE) = MIN(NOTES.NOTE) ;
```

13.

```
SELECT NOTES.MATIERES, ETUDIANT.MATRICULE, AVG(NOTES.NOTE) FROM ETUDIANT, NOTES  
GROUP BY NOTES.MATIERES, ETUDIANT.MATRICULE HAVING AVG(NOTES.NOTE) > 15 ;
```