

Projet Soutenance

Partie SQL

```
-- Script SQL : student_management (compatible MySQL / MariaDB)
-- Encodage et moteur recommandés : utf8mb4, InnoDB
```

-- 1) Création de la base

```
CREATE DATABASE IF NOT EXISTS student_management
CHARACTER SET = 'utf8mb4'
COLLATE = 'utf8mb4_general_ci';
USE student_management;
```

-- 2) Table : admins (administrateurs)

```
CREATE TABLE IF NOT EXISTS admins (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
email VARCHAR(255) NOT NULL UNIQUE,
password VARCHAR(255) NOT NULL, -- stocker HASH généré par PHP password_hash()
full_name VARCHAR(255) NULL,
created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
last_login TIMESTAMP NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- IMPORTANT: pour des raisons de sécurité, générez le hash PHP avec password_hash()

```
-- Exemple (PHP) : $hash = password_hash('TonMotDePasseAdmin', PASSWORD_DEFAULT);
-- Puis insérez la valeur dans la colonne password.
```

-- 3) Table : classes

```
CREATE TABLE IF NOT EXISTS classes (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
level VARCHAR(50) NULL,
description TEXT NULL,
created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- 4) Table : teachers (professeurs)

```
CREATE TABLE IF NOT EXISTS teachers (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
first_name VARCHAR(100) NOT NULL,
last_name VARCHAR(100) NOT NULL,
```

```
phone VARCHAR(30) NULL,  
email VARCHAR(255) NULL,  
subject VARCHAR(100) NULL,  
created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- 5) Table : students (étudiants)

```
CREATE TABLE IF NOT EXISTS students (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(100) NOT NULL,  
    last_name VARCHAR(100) NOT NULL,  
    date_of_birth DATE NULL,  
    gender ENUM('M','F','Other') DEFAULT NULL,  
    class_id INT UNSIGNED NULL,  
    phone VARCHAR(30) NULL,  
    email VARCHAR(255) NULL,  
    address TEXT NULL,  
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    CONSTRAINT fk_students_class FOREIGN KEY (class_id)  
        REFERENCES classes(id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Index utile pour les recherches par classe

```
CREATE INDEX idx_students_class_id ON students(class_id);
```

-- 6) Table : results (notes / évaluations)

```
CREATE TABLE IF NOT EXISTS results (  
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    student_id INT UNSIGNED NOT NULL,  
    subject VARCHAR(150) NOT NULL,  
    grade DECIMAL(5,2) NOT NULL,  
    term VARCHAR(50) NULL,  
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    CONSTRAINT fk_results_student FOREIGN KEY (student_id)  
        REFERENCES students(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE INDEX idx_results_student ON results(student_id);
```

-- 7) Table optionnelle : class_teacher (relation N-N entre classes et professeurs)

```
CREATE TABLE IF NOT EXISTS class_teacher (
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    class_id INT UNSIGNED NOT NULL,
    teacher_id INT UNSIGNED NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT fk_ct_class FOREIGN KEY (class_id)
        REFERENCES classes(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT fk_ct_teacher FOREIGN KEY (teacher_id)
        REFERENCES teachers(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    UNIQUE KEY uq_class_teacher (class_id, teacher_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- 8) Vue pratique pour le tableau de bord (comptages)

```
DROP VIEW IF EXISTS vw_dashboard_stats;
CREATE VIEW vw_dashboard_stats AS
SELECT
    (SELECT COUNT(*) FROM students) AS total_students,
    (SELECT COUNT(*) FROM classes) AS total_classes,
    (SELECT COUNT(*) FROM teachers) AS total_teachers,
    (SELECT COUNT(*) FROM results) AS total_results;
```

-- 9) Exemples de données (insérer pour tests)

-- Note : remplacez le hash de l'admin par un password_hash PHP avant production

```
INSERT INTO admins (email, password, full_name)
```

```
VALUES
```

```
    ('admin@example.com', SHA2('ChangeMe123',256), 'Administrateur Principal'); -- SHA2 utilisé  
    uniquement comme placeholder
```

```
INSERT INTO classes (name, level, description) VALUES
```

```
    ('L1 Dev', 'Licence 1', 'Licence 1 - Développement'),  
    ('L2 Réseau', 'Licence 2', 'Licence 2 - Réseaux'),  
    ('L3 Sécurité', 'Licence 3', 'Licence 3 - Sécurité informatique');
```

```
INSERT INTO teachers (first_name, last_name, phone, email, subject) VALUES
```

```
('Jean','Kabongo','06554433','jkabongo@example.com','Réseaux'),  
('Sarah','Mbemba','07882190','smbemba@example.com','Développement'),  
('Paul','Ngoya','06667788','pngoya@example.com','Sécurité');
```

```
INSERT INTO students  
(first_name,last_name,date_of_birth,gender,class_id,phone,email,address) VALUES  
('Roger','Moundou','2001-04-06','M', 3, '06123456','roger.m@example.com','Libreville'),  
('Jean','Kabongo','2002-12-11','M', 2, '06554433','jean.k@example.com','Brazzaville'),  
('Sarah','Mbemba','2003-08-19','F', 1, '07882190','sarah.m@example.com','Pointe-Noire'),  
('Aline','Kouumba','2004-02-03','F', 1, '07001122','aline.k@example.com','Owando'),  
('Marc','Bongo','2001-09-30','M', 2, '06998877','marc.b@example.com','Lékoni');
```

```
INSERT INTO results (student_id, subject, grade, term) VALUES  
(1, 'Mathématiques', 14.50, 'Semestre 1'),  
(1, 'Réseaux', 12.00, 'Semestre 1'),  
(2, 'Réseaux', 15.00, 'Semestre 1'),  
(3, 'Développement', 13.25, 'Semestre 1');
```

```
-- 10) Index et optimisations supplémentaires (exemples)  
ALTER TABLE students ADD INDEX idx_students_name (last_name, first_name);  
ALTER TABLE results ADD INDEX idx_results_subject (subject);  
  
-- 11) Privileges (optionnel - si vous gérez via SQL)  
-- CREATE USER 'app_user'@'localhost' IDENTIFIED BY 'strong_password';  
-- GRANT SELECT, INSERT, UPDATE, DELETE ON student_management.* TO  
'app_user'@'localhost';  
-- FLUSH PRIVILEGES;  
  
-- 12) Remarques de sécurité et bonnes pratiques  
-- - Ne pas utiliser SHA2 pour les mots de passe en production ; utilisez PHP password_hash()  
et password_verify().  
-- - Limiter les privilèges de la connexion applicative (principe du moindre privilège).  
-- - Sauvegarder la base régulièrement (mysqldump).  
-- - En production, activer SSL/TLS entre l'appli et la base si distante.  
  
-- Fin du script
```