**Semester Project**

**Web Technologies  
School Management System**

**Muhammad Ahmed – 221067**

**Noman Kiyani – 2211111**

**Muhammad Abdullah Khan - 221115**

**School Management System: Database Report**

**1. Introduction**

The **School Management System (SMS)** is an efficient software solution designed to streamline and automate various school administrative tasks. This includes managing students, staff, roles, classes, and sections. The system allows schools to maintain accurate records, manage courses and student assignments, and handle user permissions in an organized manner.

This report provides a detailed overview of the database design, its structure, and the relationships between various entities in the system.

**2. Database Overview**

The **School Management System Database** is designed to manage various elements critical to school operations, including users, classes, sections, students, and roles. The database is built on SQL Server, and it includes five core tables, along with primary and foreign key constraints to maintain data integrity.

**3. Database Structure**

**3.1. Tables and Their Structure**

**3.1.1. Classes Table**

The **Classes** table stores information about the classes in the school system.

* **Columns:**
  + Id: Unique identifier for the class.
  + Name: Name of the class (e.g., "Class 1", "Class 2").
* **Primary Key**: Id

**3.1.2. Roles Table**

The **Roles** table stores information about the roles assigned to users (e.g., Admin, Teacher, Student).

* **Columns:**
  + Id: Unique identifier for the role.
  + Name: Name of the role (e.g., "Admin", "Teacher", "Student").
* **Primary Key**: Id

**3.1.3. Sections Table**

The **Sections** table stores information about the sections within each class.

* **Columns:**
  + Id: Unique identifier for the section.
  + ClassId: Foreign key that links to the **Classes** table.
  + Name: Name of the section (e.g., "A", "B").
* **Primary Key**: Id
* **Foreign Key**: ClassId (references **Classes.Id**)

**3.1.4. Students Table**

The **Students** table stores information about students enrolled in the school.

* **Columns:**
  + Id: Unique identifier for the student.
  + ClassId: Foreign key that links to the **Classes** table.
  + SectionId: Foreign key that links to the **Sections** table.
  + UserId: Foreign key that links to the **Users** table (if available).
* **Primary Key**: Id
* **Foreign Keys**:
  + ClassId (references **Classes.Id**)
  + SectionId (references **Sections.Id**)
  + UserId (references **Users.Id**)

**3.1.5. Users Table**

The **Users** table stores user information for administrators, teachers, students, and other staff.

* **Columns:**
  + Id: Unique identifier for the user.
  + Email: Email address of the user.
  + Password: Hashed password for user authentication.
  + TempPassword: Temporary password for password resets.
  + FirstName: First name of the user.
  + LastName: Last name of the user.
  + Age: Age of the user.
  + Gender: Gender of the user.
  + RoleId: Foreign key linking the user to a specific role.
  + ImagePath: Path to the user’s image (optional).
  + CreatedAt: Date and time the user was created.
  + UpdatedAt: Date and time the user details were last updated.
  + DeletedAt: Date and time the user was deleted (if applicable).
  + IsDeleted: Flag indicating whether the user is deleted.
* **Primary Key**: Id
* **Foreign Key**: RoleId (references **Roles.Id**)

**4. Relationships Between Tables**

* **Users and Roles**: The **Users** table references the **Roles** table through the RoleId foreign key. This establishes the relationship between users and their respective roles (e.g., Admin, Teacher, Student).
* **Sections and Classes**: The **Sections** table has a foreign key relationship with the **Classes** table via the ClassId column. Each section belongs to a specific class.
* **Students and Sections**: The **Students** table is linked to the **Sections** table via the SectionId column. Each student is assigned to a specific section in a class.
* **Students and Users**: The **Students** table can reference the **Users** table through the UserId foreign key. This allows students to be associated with user accounts for login and authentication.

**5. Database Integrity**

**5.1. Primary Keys**

Each table has a primary key on the Id column. This ensures that each record in the table is unique and easily identifiable.

**5.2. Foreign Keys**

Foreign keys are used to establish relationships between the tables:

* The **Sections** table’s ClassId references the **Classes** table.
* The **Users** table’s RoleId references the **Roles** table.
* The **Students** table’s ClassId, SectionId, and UserId create relationships with the **Classes**, **Sections**, and **Users** tables respectively.

These foreign key constraints maintain referential integrity, ensuring that relationships between entities are consistently represented.

**5.3. Indexing**

Indexes are created on the foreign key columns (ClassId in **Sections** and RoleId in **Users**) to optimize query performance.

**6. Features of the School Management System**

**6.1. Role-based Access Control (RBAC)**

Users in the system can have different roles (Admin, Teacher, Student), and each role may have different permissions. The **Roles** table manages the assignment of roles to users.

**6.2. Class and Section Management**

The system allows administrators to create classes and assign students to sections. This is managed through the **Classes** and **Sections** tables.

**6.3. Student Management**

Students can be assigned to specific classes and sections, and their details are stored in the **Students** table.

**6.4. User Management**

Administrators can create, update, and delete user accounts (teachers, students, etc.) through the **Users** table. Temporary passwords can be assigned for password recovery.

**7. Conclusion**

The **School Management System** database is designed to facilitate smooth and organized management of school data. With a clear structure for classes, sections, roles, users, and students, the system ensures that each part of the school’s operations is linked and can be efficiently managed.

This relational database, backed by SQL Server, ensures data integrity and scalability, supporting the system's future growth.