

**NED UNIVERSITY OF ENGINEERING**

**&**

**TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE & IT**

**CT-261 Data Base Management Systems**

*Instructor: Sir Umer Farooq*

**PROJECT REPORT**

**UNIVERSITY MANAGEMENT SYSTEM**

**Members:**

|  |  |
| --- | --- |
| MUSAB UMAIR | CT-22095 |
| FAIZAN TAHIR | CT-22074 |
| SHAHZAIN ADIL | CT-22076 |
| NOFIL AHMED KHAN | CT-22301 |

**ORGANIZATION INTRODUCTION:**

At our institution, we prioritize excellence in education by providing a seamless and comprehensive learning experience. Our advanced system ensures personalized attention for every student, from admission to graduation. Our dedicated teachers are equipped with tools to manage classes, prepare tests, and track student progress meticulously. Our administrators oversee all operations, including library management and fee processing, ensuring a smooth and efficient academic environment. Join us in nurturing the leaders of tomorrow in an enriching and supportive educational setting. Welcome to a world of possibilities. Welcome to our institution.

**PROBLEM OF ORGANIZATION (Fact Finding)**

In our pursuit of educational excellence, we have identified a significant challenge in effective communication and data management across departments and stakeholders. This issue has led to:

1. **Fragmented Information Systems:** Disconnected modules result in inefficiencies and data discrepancies.
2. **Delayed Communication:** Students and staff experience delays in receiving important updates, causing confusion.
3. **Administrative Overload:** Manual data management overwhelms administrative staff, increasing errors.
4. **Limited Teacher-Student Interaction:** Feedback and communication between teachers and students need improvement.

**Proposed Solutions:**

1. **Integrated Information System:** A unified dashboard for real-time updates and data consolidation.
2. **Automated Notification System:** Timely SMS, email, and app notifications for important updates.
3. **Administrative Support Tools:** Advanced tools and training to reduce manual workload and errors.
4. **Enhanced Communication Platforms:** Improved platforms for better teacher-student interaction.

Addressing these challenges will streamline operations, improve communication, and enhance the educational experience for students and staff alike.

**OVERVIEW OF THE SYSTEM:**

Our institution's advanced management system streamlines academic and administrative processes, ensuring a seamless experience for all stakeholders.

**Key Features:**

**Student Management:** Tracks student journeys from admission to graduation, including personal information, attendance, test scores, and fee payments.

**Teacher Management:** Provides tools for managing classes, grading tests, and recording attendance, facilitating efficient communication with students.

**Administrative Operations:** Supports daily tasks like posting notices, managing library resources, and processing financial transactions.

**Library Management:** Manages book inventories, borrowing records, and returns, offering easy access to academic resources.

**Fee Management:** Handles financial transactions with transparency and efficiency.

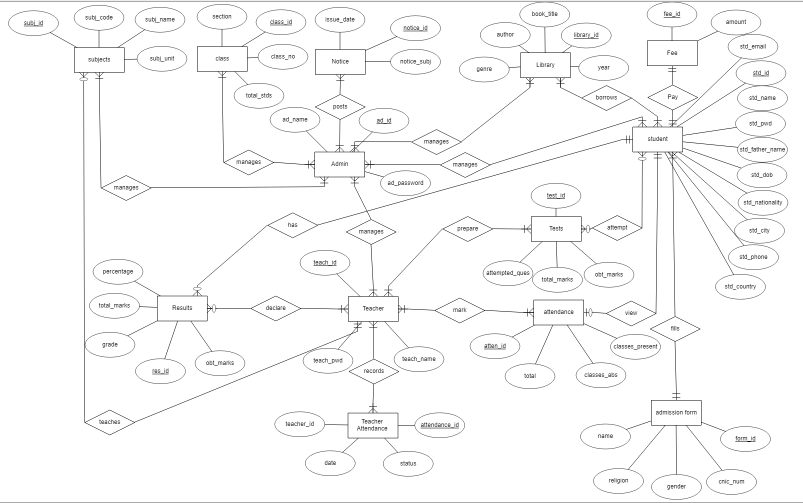
**Performance Tracking:** Monitors student performance, aiding in informed decision-making and personalized support.

**Communication and Notifications:** Ensures timely updates via SMS, email, and mobile app alerts.

**LIMITATIONS:**

* **Fragmented Information Systems:** Disconnected modules can lead to inefficiencies and data discrepancies.
* **Delayed Communication:** Important updates may not reach stakeholders promptly, causing confusion.
* **Administrative Overload:** High manual data management demands increase the risk of errors.
* **Limited Teacher-Student Interaction:** Feedback and communication channels between teachers and students need enhancement.
* **Scalability Issues:** The system may face challenges adapting to rapid increases in the number of users or data volume.
* **Integration Challenges:** Difficulty in integrating with external systems or newer technologies.

**LOGICAL ENTITY RELATIONSHIP DIAGRAM:**



**FOREIGN KEYS:**

* **subjects**

subj\_id (Foreign key to sub\_id in subject)

* **class**

class\_id (Foreign key to class\_id in subject)

* **Notice**

class\_id (Foreign key to class\_id in class)

notice\_id (Primary key)

* **Library**

library\_id (Foreign key to lib\_id in admin)

* **Fee**

fee\_id (Foreign key to fee\_id in student)

* **student**

std\_id (Primary key)

class\_id (Foreign key to class\_id in class)

* **Admin**

ad\_id (Primary key)

* **Tests**

test\_id (Foreign key to test\_id in teacher)

* **Teacher**

teach\_id (Foreign key to teach\_id in class)

teacher\_id (Foreign key to teacher\_id in teacher attendance)

* **Teacher Attendance**

attendance\_id (Foreign key to attendance\_id in teacher)

* **Results**

res\_id (Primary key)

teach\_id (Foreign key to teach\_id in teacher)

std\_id (Foreign key to std\_id in student)

* **admission form**

form\_id (Primary key)

std\_id (Foreign key to std\_id in student)

**Application of Entity and Referential Integrity:**

**Subjects and Class:**

* + Each subject and class must have a unique identifier (subj\_id and class\_id respectively), ensuring entity integrity.
  + subj\_id in subjects and class\_id in class must exist in their respective referenced tables, maintaining referential integrity.

**Notices:**

* + Each notice must have a unique notice\_id.
  + class\_id in Notice must exist in class, ensuring the class referred to in the notice exists.

**Library and Admin:**

* + library\_id must be unique.
  + library\_id in Library must correspond to an existing lib\_id in Admin.

**Fees:**

* + fee\_id must be unique.
  + fee\_id in Fee must exist in student, ensuring the fee record is associated with an existing student.

**Students:**

* + std\_id must be unique.
  + class\_id in student must exist in class, linking students to existing classes.

**Tests:**

* + test\_id must be unique.
  + test\_id in Tests must exist in teacher, ensuring tests are associated with existing teachers.

**Teachers:**

* + teach\_id and teacher\_id must be unique.
  + teach\_id in Teacher must exist in class, linking teachers to existing classes.

**Teacher Attendance:**

* + attendance\_id must be unique.
  + attendance\_id in Teacher Attendance must exist in teacher, ensuring attendance records are linked to existing teachers.

**Results:**

* + res\_id must be unique.
  + teach\_id in Results must exist in teacher, and std\_id must exist in student, ensuring results are associated with existing teachers and students.

**Admission Forms:**

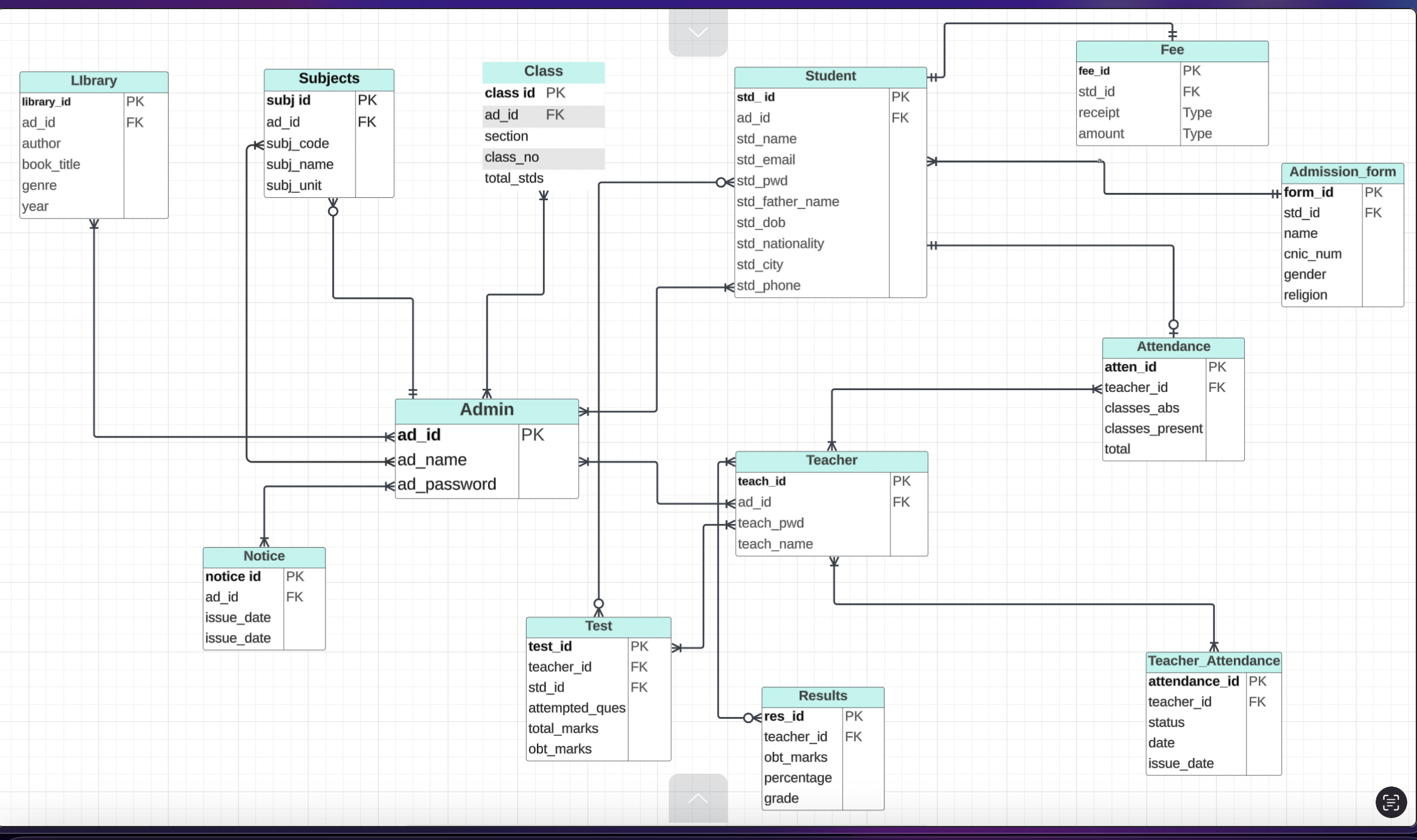
* + form\_id must be unique.
  + std\_id in admission form must exist in student, ensuring forms are associated with existing students.

By adhering to these entity and referential integrity rules, we maintain data consistency and reliability across our educational management system, ensuring accurate and trustworthy records.

**General Constraints for the Educational Management System Schema:**

* **Primary Key Constraints:**
  + Ensure each table has a unique primary key to maintain entity integrity.
  + Example: std\_id in student, teach\_id in teacher, class\_id in class.
* **Foreign Key Constraints:**
  + Enforce referential integrity by ensuring foreign keys reference valid primary keys in other tables.
  + Example: class\_id in student referencing class\_id in class.
* **Unique Constraints:**
  + Ensure certain fields that must be unique across records.
  + Example: std\_email in student, ad\_name in Admin.
* **Not Null Constraints:**
  + Ensure critical fields are not left empty.
  + Example: std\_name, std\_email, teach\_name, ad\_name.
* **Check Constraints:**
  + Validate the values of specific fields based on predefined conditions.
  + Example: std\_dob should be a valid date and ensure students are within an acceptable age range.
  + Example: percentage in Results should be between 0 and 100.
* **Default Constraints:**
  + Provide default values for certain fields to ensure consistency and reduce null entries.
  + Example: status in Teacher Attendance can have a default value of 'present'.
* **Domain Constraints:**
  + Limit the values of a field to a specific set of values.
  + Example: gender in admission form can be constrained to ('Male', 'Female', 'Other').
  + Example: grade in Results can be constrained to ('A', 'B', 'C', 'D', 'F').
* **Referential Actions:**
  + Define actions to take on foreign key updates or deletions.
  + Example: ON DELETE CASCADE for class\_id in student to automatically delete students when a class is deleted.
  + Example: ON UPDATE CASCADE for teach\_id in Results to update the foreign key in results when the teacher's ID changes.
* **Length Constraints:**
  + Restrict the length of certain fields to ensure data consistency.
  + Example: std\_name should not exceed 50 characters, std\_phone should be exactly 10 digits.
* **Value Constraints:**
  + Enforce specific value constraints on numeric fields.
  + Example: amount in Fee should be a positive number.

**PHYSICAL ENTITY RELATIONSHIP DIAGRAM:**



**SQL CODE:**

* **ADMIN**

Create table Admin (

ad\_id number not null constraint admin\_pk primary key,

ad\_password varchar2(20) not null,

ad\_name varchar2(45) not null

)

* **SUBJECTS**

create table Subjects (

subj\_id number not null,

subj\_name varchar2(30) not null,

subj\_unit varchar2(8) not null,

subj\_code varchar2(8) not null,

ad\_id number not null,

primary key (subj\_id),

foreign key(ad\_id)references Admin(ad\_id)

)

* **STUDENTS**

create table Students (

std\_id number not null,

std\_name varchar2(50) not null,

std\_dob varchar2(25) not null,

std\_pwd varchar2(50) not null,

std\_fathername varchar2(30) not null,

std\_nationality varchar2(30) not null,

std\_city varchar2(30) not null,

std\_phoneno varchar2(15) not null,

std\_email varchar2(30) not null,

ad\_id not null,

primary key(std\_id),

foreign key(ad\_id) references ADMIN(ad\_id)

)

* **CLASS**

create table Class (

class\_id number not null,

class\_roomno varchar2(4) not null,

class\_section varchar2(3) not null,

class\_totalstd number not null,

ad\_id number not null,

primary key(class\_id),

foreign key(ad\_id) references Admin(ad\_id)

)

* **TEACHERS**

CREATE table Teachers

(

teacher\_id Number not null,

teacher\_name varchar(30) not null,

teacher\_phoneno varchar(20)not null,

teacher\_pwd varchar(40) not null,

ad\_id number not null,

primary key (teacher\_id),

foreign key(ad\_id) references Admin(ad\_id)

)

* **RESULTS**

CREATE TABLE Results

(

total\_marks NUMBER NOT NULL,

obt\_marks NUMBER NOT NULL,

percentage NUMBER NOT NULL,

grade VARCHAR2(3) NOT NULL,

result\_id NUMBER NOT NULL,

teacher\_id NUMBER NOT NULL,

PRIMARY KEY (result\_id),

FOREIGN KEY (teacher\_id) REFERENCES Teachers(teacher\_id)

)

* **TEST**

create table test(

TEST\_NO number not null,

attempted\_ques VARCHAR2(3) NOT NULL,

total\_ques VARCHAR(5) NOT NULL,

total\_marks NUMBER NOT NULL,

obt\_marks NUMBER NOT NULL,

std\_id NUMBER NOT NULL,

teacher\_id NUMBER NOT NULL,

PRIMARY KEY (TEST\_NO),

FOREIGN KEY (std\_id) REFERENCES Students(std\_id),

FOREIGN KEY (teacher\_id) REFERENCES Teachers(teacher\_id)

)

* **FEE**

CREATE TABLE Fee

(

fee\_id NUMBER NOT NULL,

fee\_receipt VARCHAR2(15) NOT NULL,

fee\_amount NUMBER NOT NULL,

std\_id NUMBER NOT NULL,

PRIMARY KEY (fee\_id),

FOREIGN KEY (std\_id) REFERENCES Students(std\_id)

)

* **NOTICE**

CREATE TABLE Notice

(

issue\_date date NOT NULL,

notice\_id NUMBER NOT NULL,

notice\_subj VARCHAR2(100) NOT NULL,

ad\_id NUMBER NOT NULL,

PRIMARY KEY (notice\_id),

FOREIGN KEY (ad\_id) REFERENCES Admin(ad\_id)

)

* **ADMISSION\_FORM**

CREATE TABLE Admission\_Form

(

form\_id NUMBER NOT NULL,

name VARCHAR(40) NOT NULL,

cnic\_num VARCHAR2(15) NOT NULL,

religion VARCHAR2(20) NOT NULL,

gender VARCHAR2(12) NOT NULL,

std\_id NUMBER NOT NULL,

PRIMARY KEY (form\_id),

FOREIGN KEY (std\_id) REFERENCES Students(std\_id)

)

* **ATTENDANCE**

CREATE TABLE Attendance

(

classes\_present NUMBER NOT NULL,

classes\_abs NUMBER NOT NULL,

classes\_total NUMBER NOT NULL,

attendance\_id NUMBER NOT NULL,

teacher\_id NUMBER NOT NULL,

PRIMARY KEY (attendance\_id),

FOREIGN KEY (teacher\_id) REFERENCES Teachers(teacher\_id)

)

* **TEACHER\_ATTENDANCE**

CREATE TABLE Teacher\_attendance (

teachattendance\_id NUMBER NOT NULL,

teachattendance\_date DATE NOT NULL,

teachattendance\_status VARCHAR2(15) NOT NULL,

teacher\_id number not null,

PRIMARY KEY (teachattendance\_id),

FOREIGN KEY (teacher\_id) REFERENCES teachers(teacher\_id)

)

* **LIBRARY**

CREATE TABLE library (

book\_id number Not null,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

published\_year date NOT NULL,

genre VARCHAR(100),

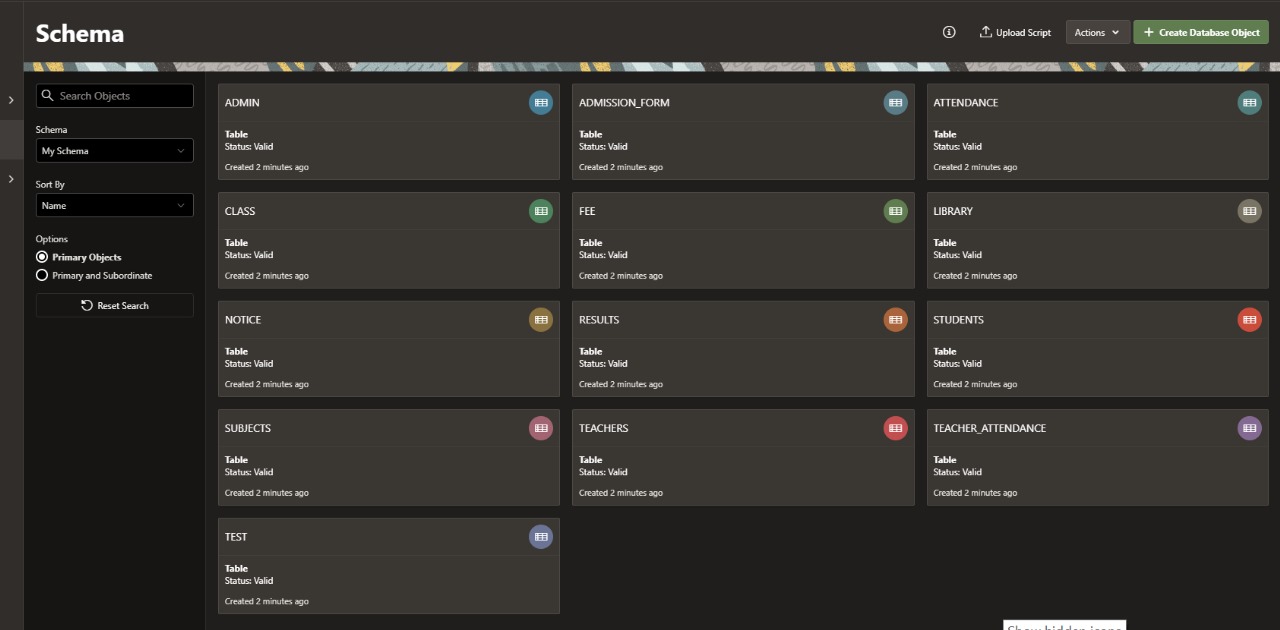
ad\_id number NOT NULL,

primary key(book\_id),

foreign key(ad\_id)REFERENCES Admin(ad\_id)

)

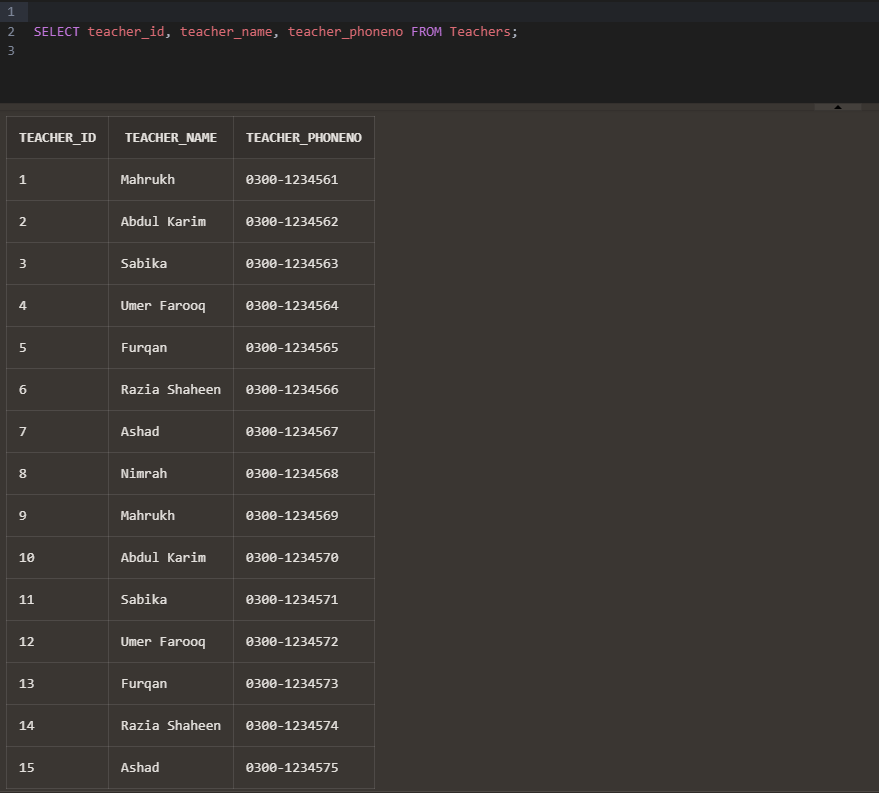
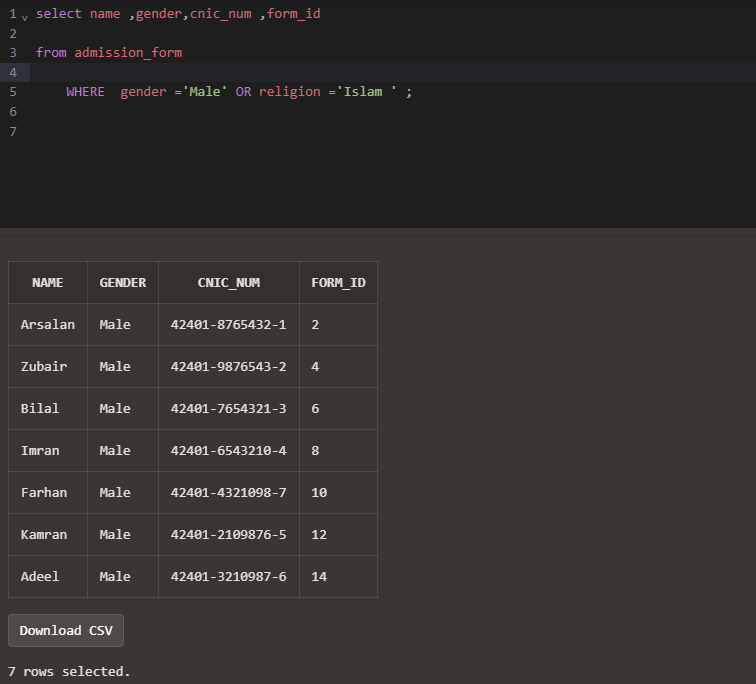
**SCHEMA OF ORGANIZATION:**

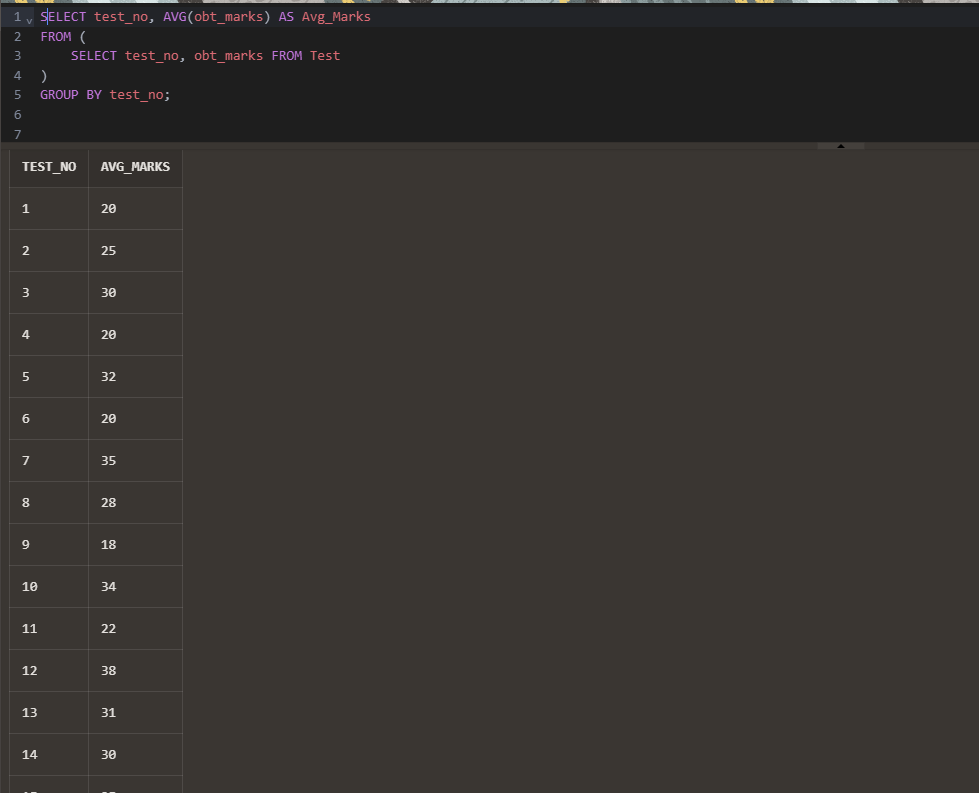


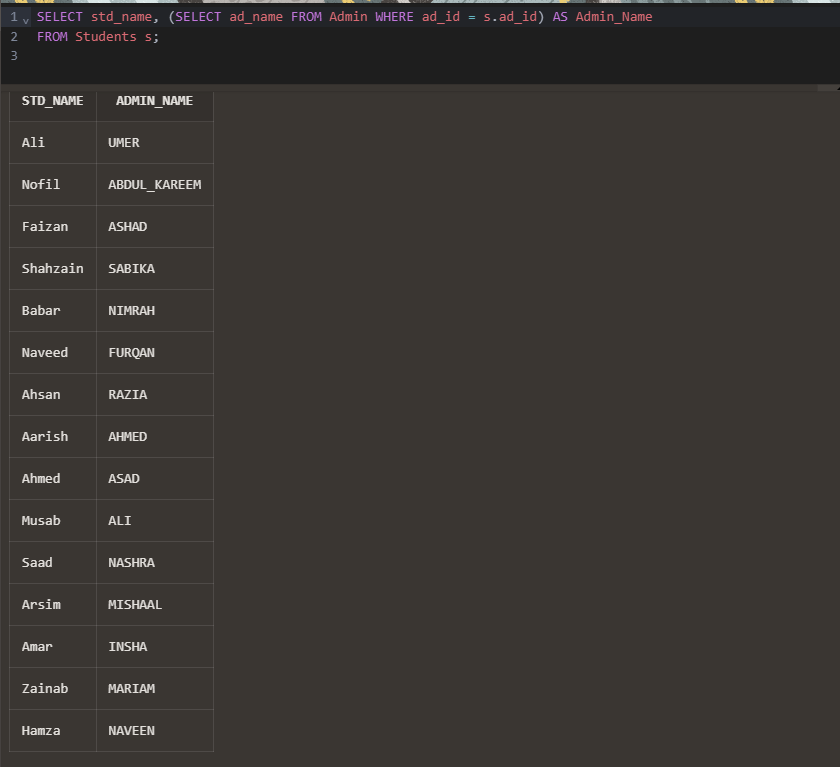
**USER PRIVELEGES:**

* Admin: Full access to all tables and operations as well as manage the results, view attendance, and issue notice.
* Students: Access to browse attendance, view results, and verify fees update.

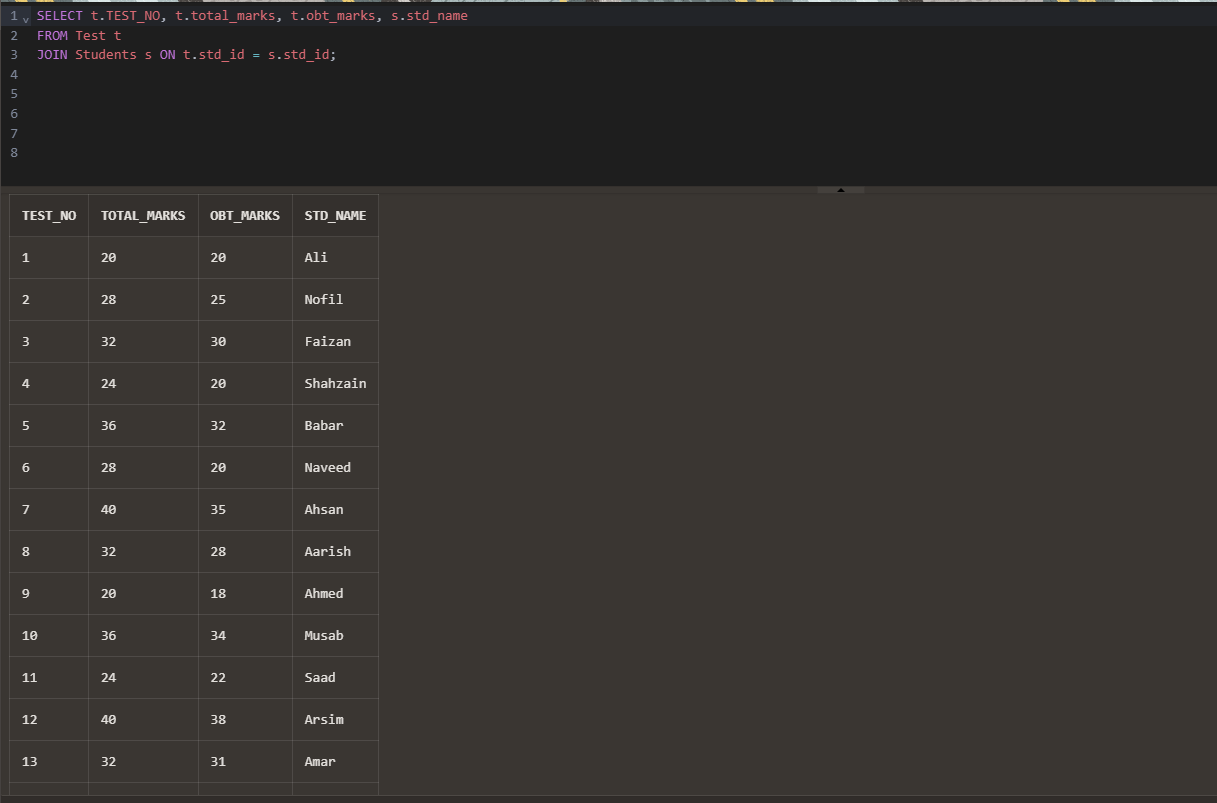
**QUERIES:**

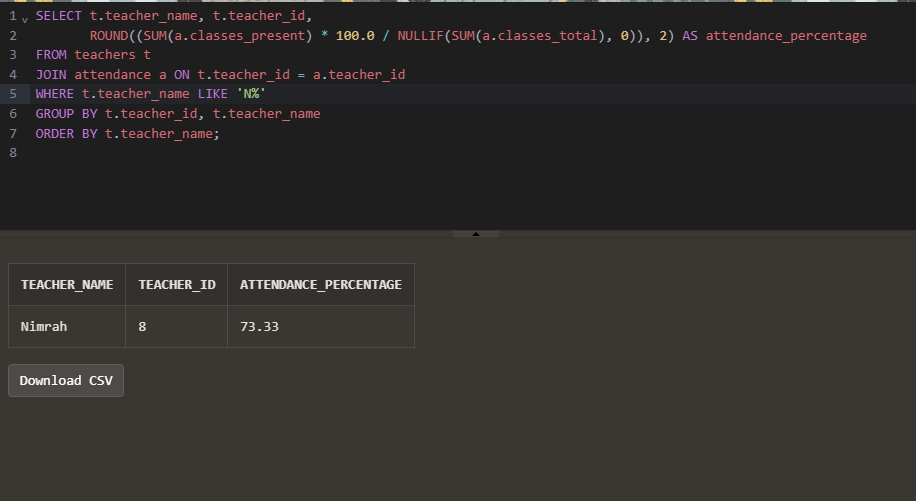
* **SELECT QUERIES**  
    
  
* **WHERE QUERIES**  
  
* **SUBQUERY**



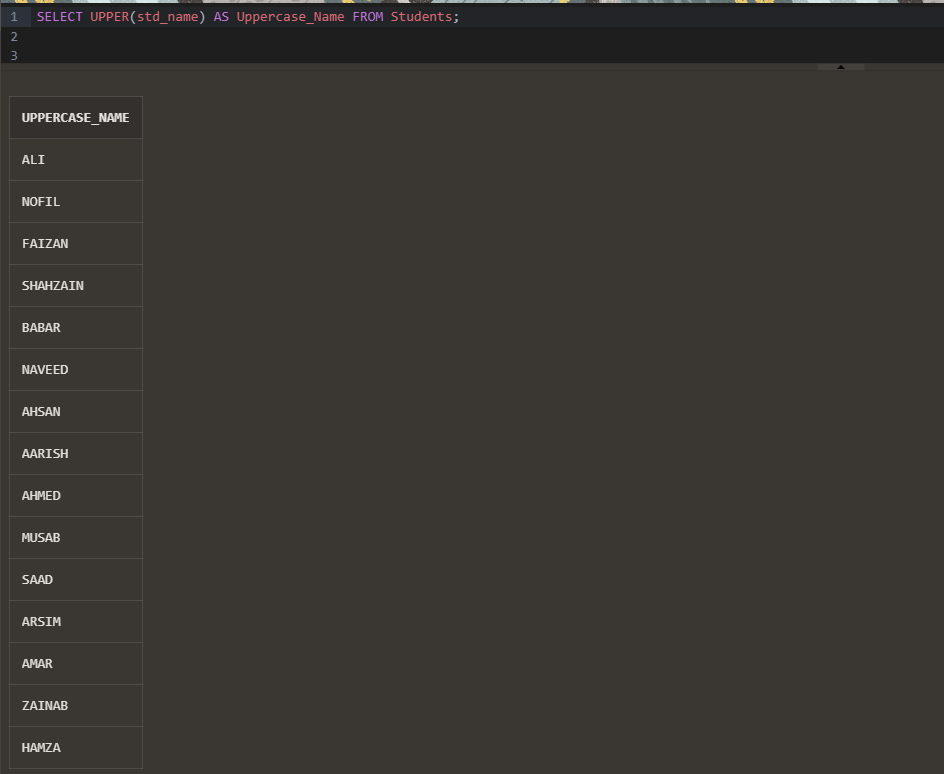


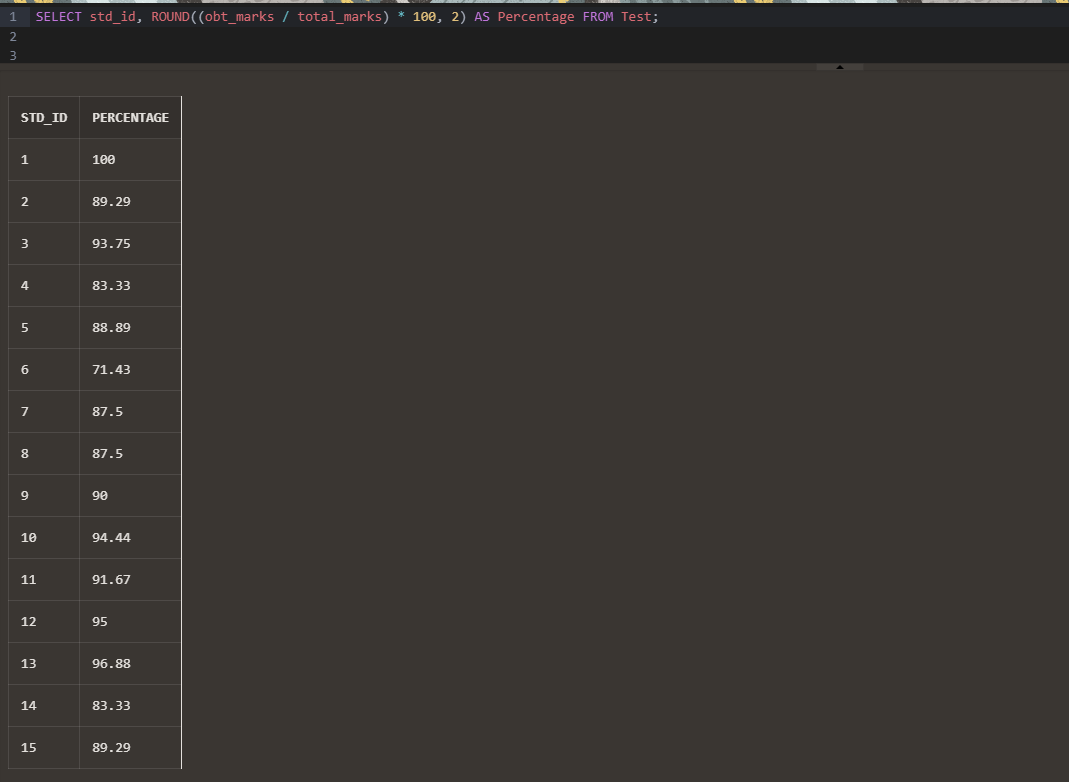
* **JOIN**



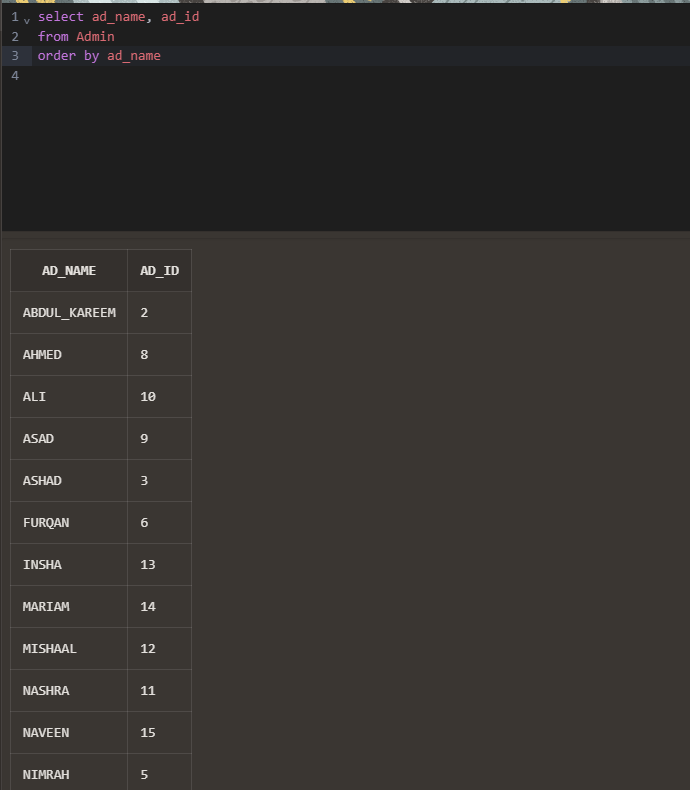


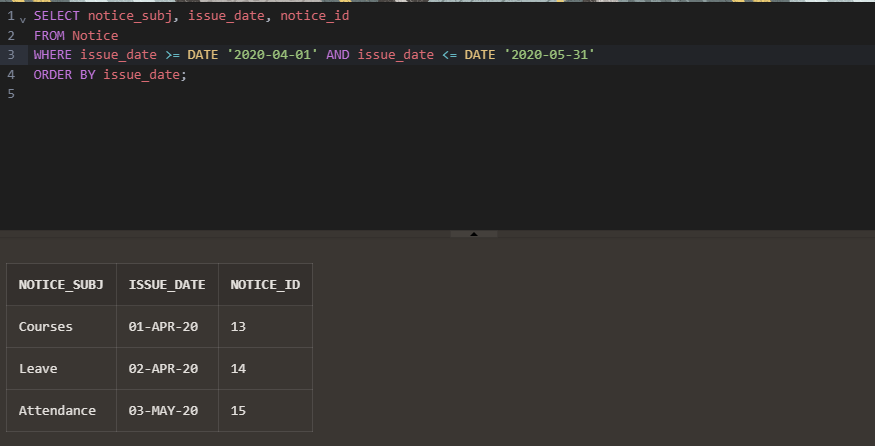
* **SINGLE-ROW**





* **FUNCTIONS**





**CONCLUSION:**

Our organization is committed to excellence in education and administration. By implementing a robust database system, we have streamlined various processes, ensuring efficient management of academic records, user privileges, and institutional data. The creation of distinct user roles—Admin, Teacher, and Student—reflects our dedication to providing tailored access and responsibilities, enhancing security and operational efficiency. This structured approach fosters a collaborative environment where administrators can oversee operations seamlessly, teachers can focus on delivering quality education, and students can easily access their academic information.

Through these efforts, we aim to create a supportive and dynamic educational ecosystem that empowers all stakeholders, promotes transparency, and upholds the highest standards of educational excellence.