# 

**College Management System**

# Software Requirements Specifications

# Nashra Ghaffar (CT-032)

# Mehak Hussain (CT-030)

# M. Abdul Rafay (CT-012)

# Mahshid Fatima (CT-023)

# Prepared For Final Project DBMS

# Instructor: Dr. M. Umer Farooq

# Introduction:

A College Management System (CMS) is a System that manages the records of student regarding admission and examination part. A College Management System (CMS) is designed to help colleges for management of student. Extensive information is available at your fingertips through this System. Viewing student data, managing admission, subject management, scheduling exam, result and related issues are made simple and easy. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. College Management System allows the keeping of personnel data in a form that can be easily accessed and analyzed in a consistent way.

# Purpose:

The project aims at the following matters:

* + 1. Automation of admission and enrolment.
    2. Assistance in decision-making.
    3. To manage information of student, faculty and courses.
    4. Consistently update information of all the students.

# Functional Requirements

#### Log in Module (LM)

User (admin, student and teachers) shall be able to load the Login Module in the internet browser. The LM shall support the user to log into the system.

#### Normal Users Module (NUM)

Users who visit CMS but have not registered, are able to navigate through the website.

#### Administrator Module (AM)

. When administrator adds, updates or delete and entry, the AM module will send the request to the Server Module which will do the necessary changes to the DB.

#### Server Module (SM)

SM shall be between the various modules and the DB. SM shall receive all requests and format the pages accordingly to be displayed.

# Non-Functional Requirements

#### Performance

The College Management System shall be built upon the web development technique put on the web server online.

#### Availability

Immediate feedback of the systems activities shall be communicated to the user by clearing the system and giving space n speed to their hospitality.

#### Security

There shall be a strong security mechanism should be place in the server side of the system to keep unwanted users to hack or damage the system.

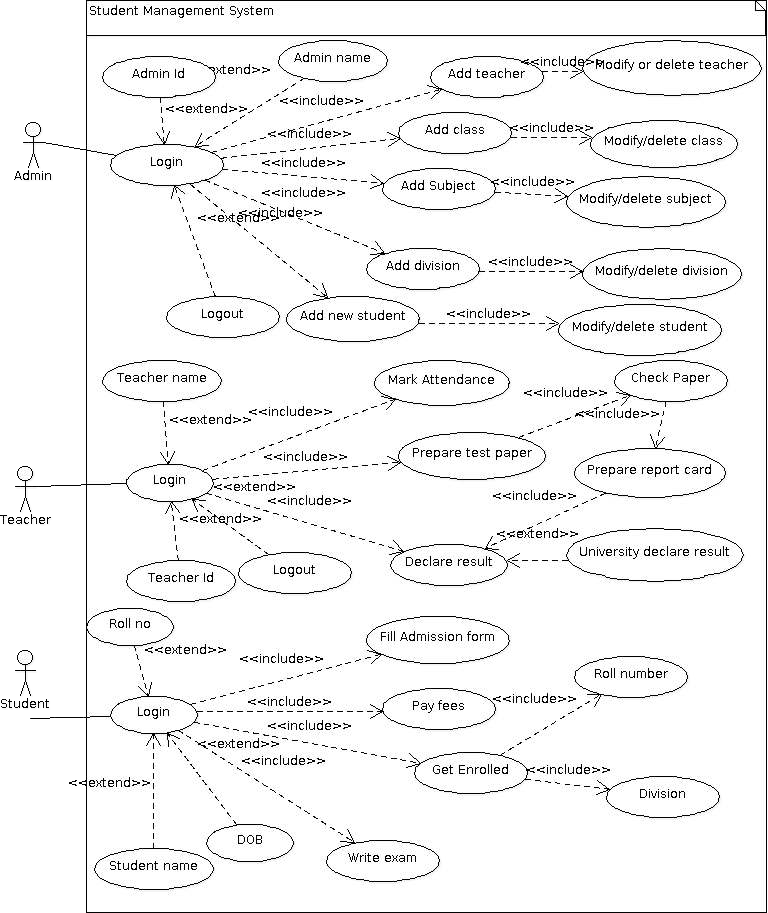
#### Maintainability

There shall be design documents describing maintenance of the software and database used to save the user details as well as the daily updated and modification done in system.

#### Portability

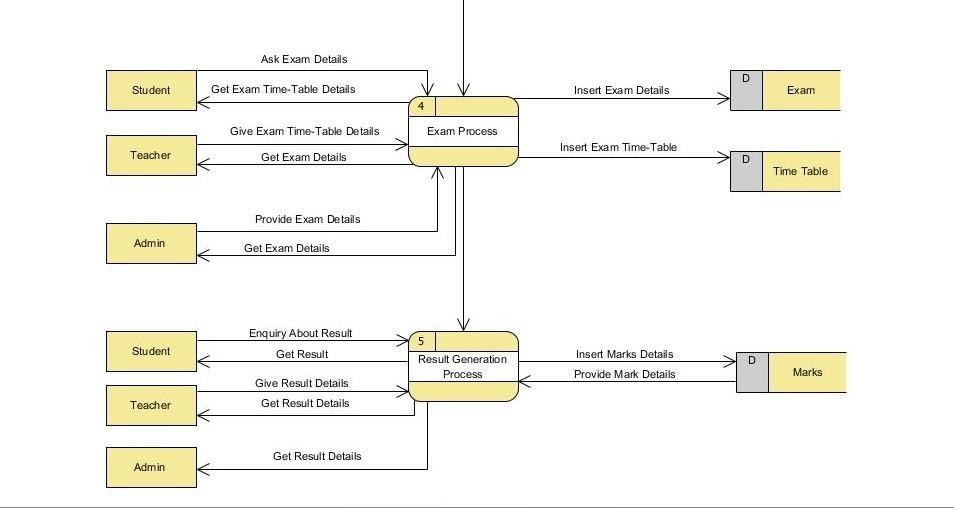
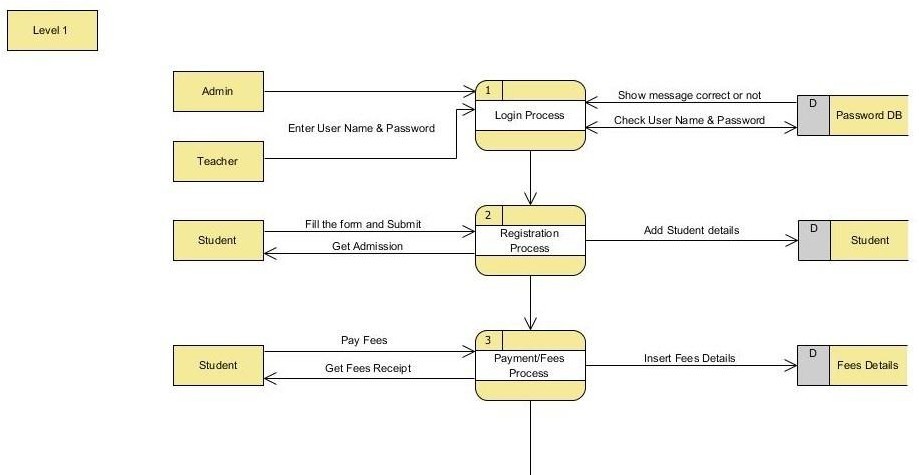
There is portability requirement as far as our system is concern because it is an online as well as offline (local server based) system so we can access it from anywhere through the internet connection.

# Use Case:

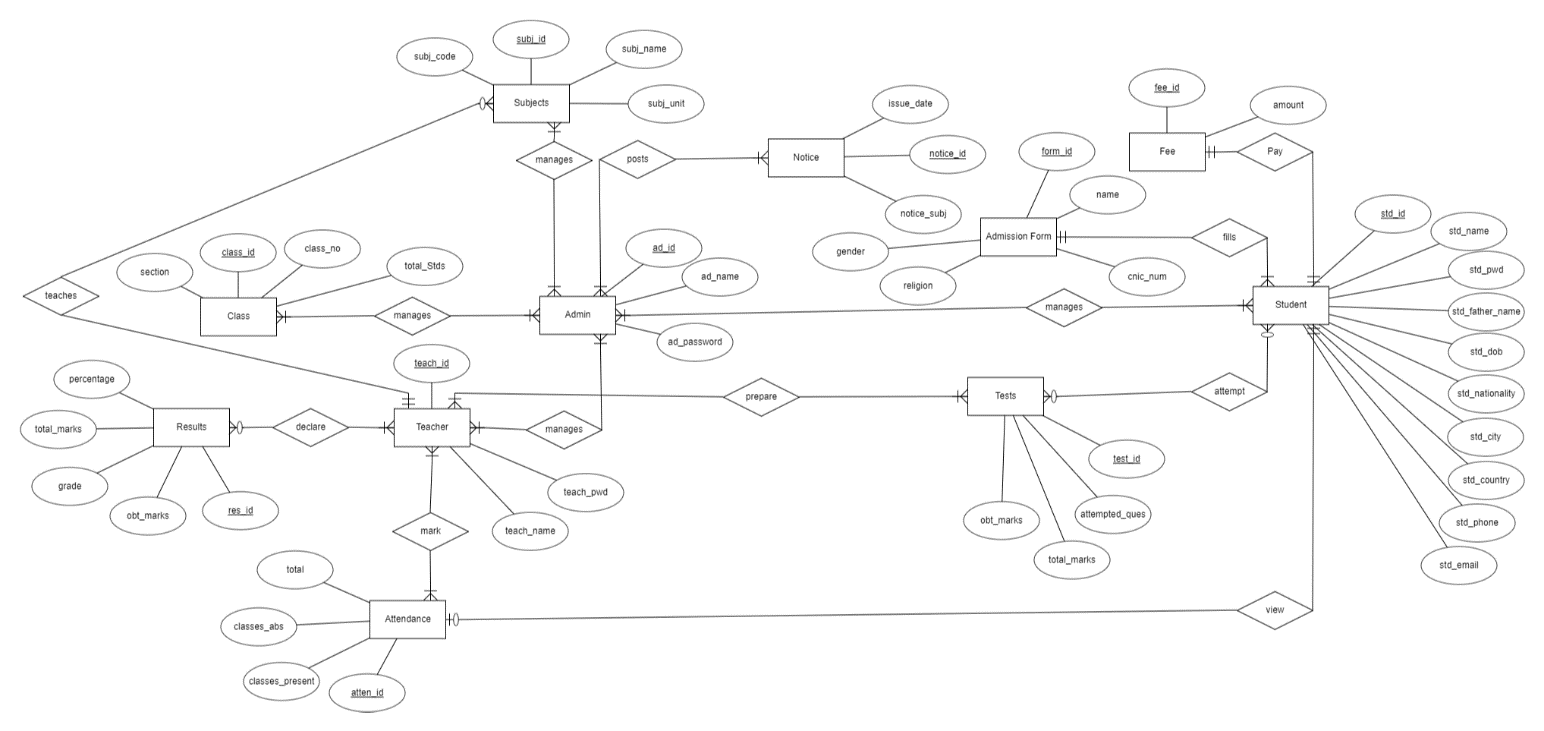


# Data Flow Diagram:

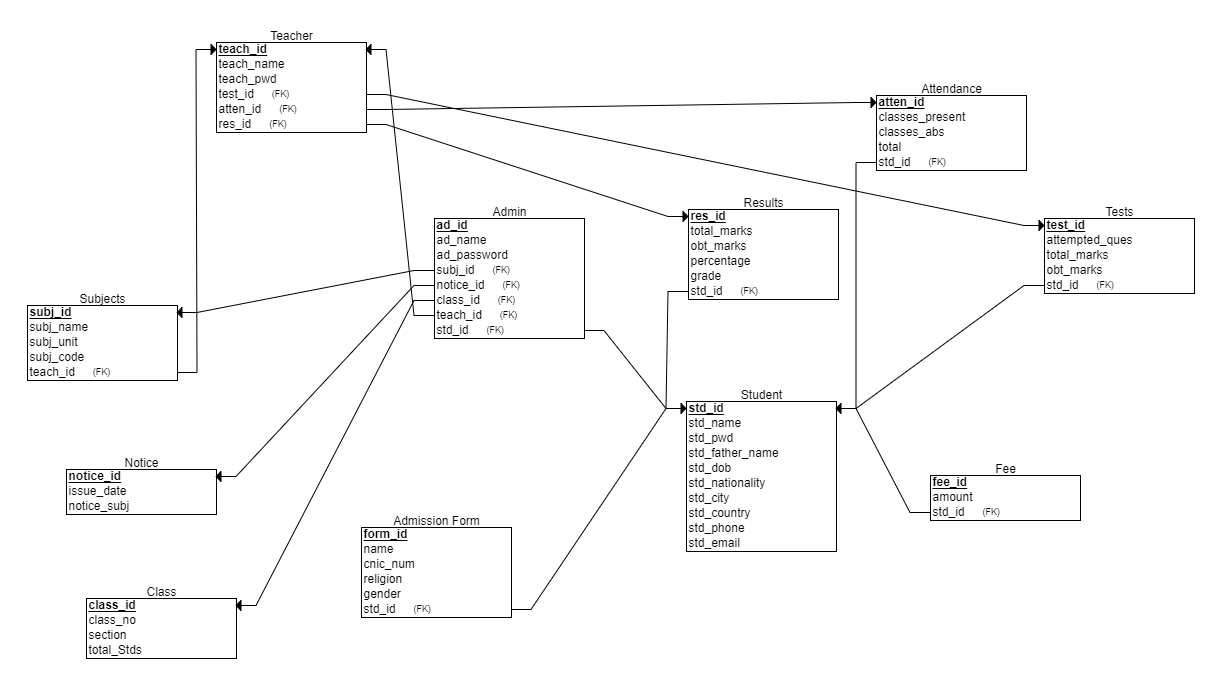
# 



1. **ER Diagram (Logical):**



1. **ER Diagram (Physical):**



1. **Foreign Keys:**

*“teach\_id”* from the table (Teacher) is treated as a foreign key in the tables (Subjects) and (Admin)

*“std\_id”* from the table (Student) is treated as a foreign key in the tables (Admin),(Attendance), (Tests) , (Fee),(Results) and (Admission Form)

1. **Integrity Rules:**

The application of entity and referential integrity among these relations are performed such as in every table the value of primary key cannot be NULL like in tables e.g. (Student, Class, Subjects etc) which satisfies the entity integrity between these relations. Furthermore the primary keys of the tables (Student, Admin and Teacher) are acting as a foreign key in the other following tables which itself explains Referential integrity between these relations.

1. **Recommended General constraints for this schema:**

In this schema, the suggested general constraints would be “NOT NULL” which references a column to NOT accept NULL values; for example name, id, password etc which cannot be NULL as these are the important aspects of the system to proceed further functionalities

Other general constraints used in our system is “Unique”. And else could be Domain and Key constraints.

1. **SQL Code (ERD Plus):**

CREATE TABLE Admin

(

ad\_id NUMBER NOT NULL,

ad\_name VARCHAR2(40) NOT NULL,

ad\_password VARCHAR2(15) INT NOT NULL,

PRIMARY KEY (ad\_id)

);

CREATE TABLE Subjects

(

subj\_id NUMBER NOT NULL,

subj\_name VARCHAR2(55) NOT NULL,

subj\_unit VARCHAR2(5) 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)

);

CREATE TABLE Student

(

std\_id NUMBER NOT NULL,

std\_name VARCHAR2(40) NOT NULL,

std\_pwd VARCHAR2(15) NOT NULL,

std\_father\_name VARCHAR2(55) NOT NULL,

std\_dob VARCHAR2(10) NOT NULL,

std\_nationality VARCHAR2(15) NOT NULL,

std\_city VARCHAR2(20) NOT NULL,

std\_country VARCHAR2(25) NOT NULL,

std\_phone VARCHAR2(12) NOT NULL,

std\_email VARCHAR2(30) NOT N…

[5:03 pm, 22/07/2022] Mahshid: PRIMARY KEY (teach\_id),

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

);

CREATE TABLE Class

(

class\_id NUMBER NOT NULL,

class\_no VARCHAR2(2) NOT NULL,

section VARCHAR2(2) NOT NULL,

total\_Stds VARCHAR2(3) NOT NULL,

ad\_id NUMBER NOT NULL,

PRIMARY KEY (class\_id),

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

);

CREATE TABLE Results

(

total\_marks NUMBER NOT NULL,

obt\_marks NUMBER NOT NULL,

percentage NUMBER NOT NULL,

grade VARCHAR2(3) NOT NULL,

res\_id NUMBER NOT NULL,

teach\_id NUMBER NOT NULL,

PRIMARY KEY (res\_id),

FOREIGN KEY (teach\_id) REFERENCES Teacher(teach\_id)

);

CREATE TABLE Attendance

(

classes\_present NUMBER NOT NULL,

classes\_abs NUMBER NOT NULL,

total NUMBER NOT NULL,

atten\_id NUMBER NOT NULL,

teach\_id NUMBER NOT NULL,

PRIMARY KEY (atten\_id),

FOREIGN KEY (teach\_id) REFERENCES Teacher(teach\_id)

);

CREATE TABLE Tests

(

test\_id NUMBER NOT NULL,

attempted\_ques VARCHAR2(2) NOT NULL,

total\_marks NUMBER NOT NULL,

obt\_marks NUMBER NOT NULL,

std\_id NUMBER NOT NULL,

teach\_id NUMBER NOT NULL,

PRIMARY KEY (test\_id),

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

FOREIGN KEY (teach\_id) REFERENCES Teacher(teach\_id)

);

CREATE TABLE Fee

(

fee\_id NUMBER NOT NULL,

fee\_receipt VARCHAR2(15) NOT NULL,

amount NUMBER NOT NULL,

std\_id NUMBER NOT NULL,

PRIMARY KEY (fee\_id),

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

);

CREATE TABLE Notice

(

issue\_date INT 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)

);

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,

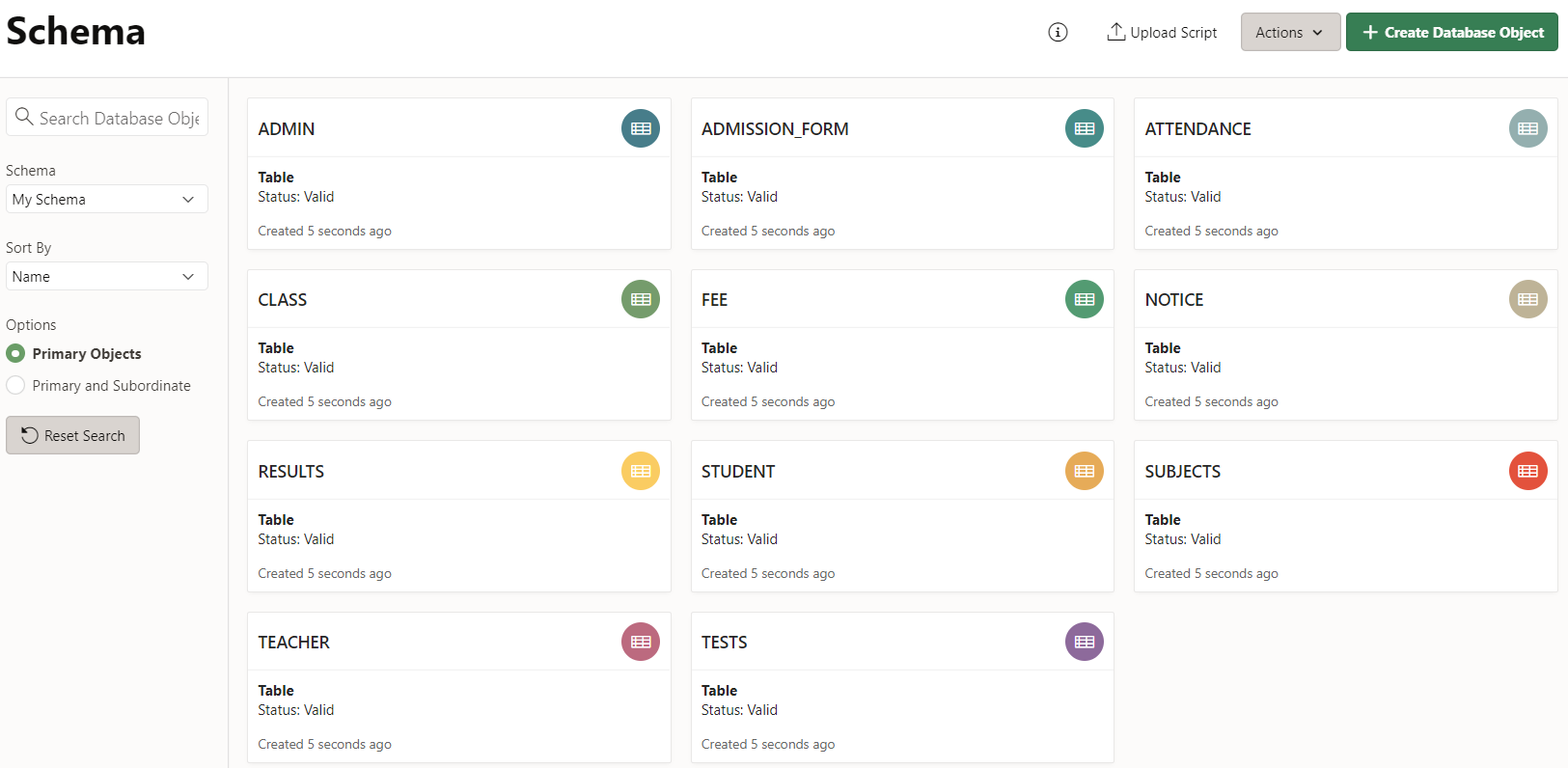
std\_id NUMBER NOT NULL,

PRIMARY KEY (form\_id),

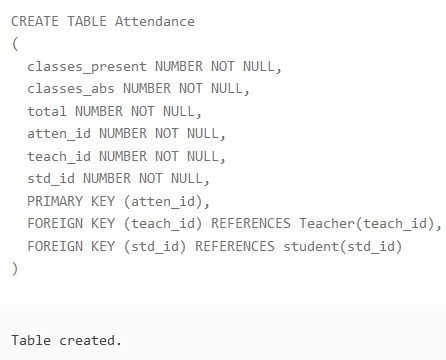
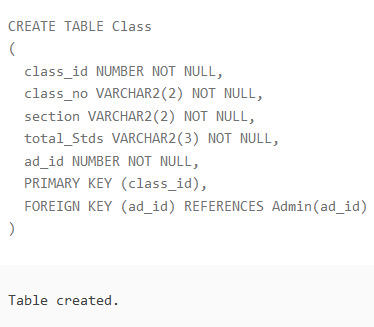
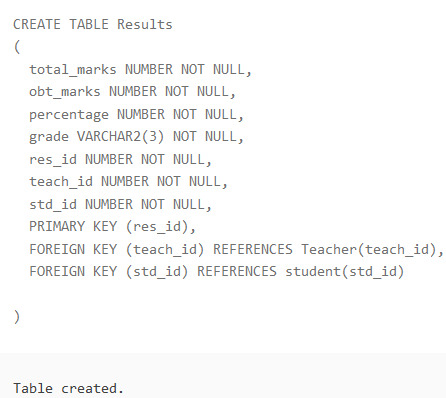
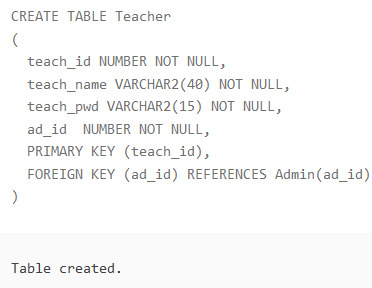
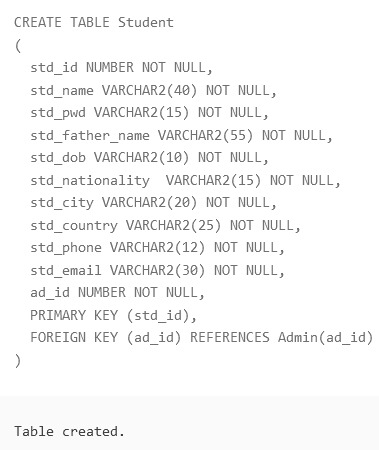
FOREIGN KEY (std\_id) REFERENCES Student(std\_id)

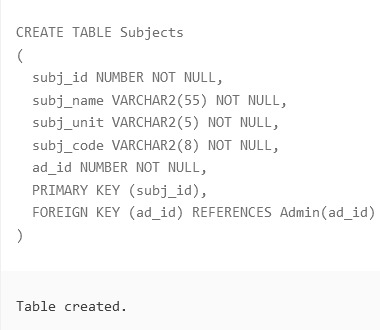
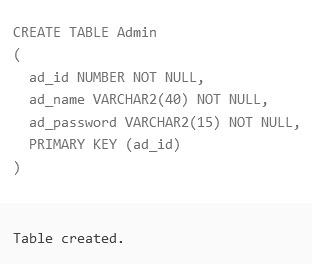
);

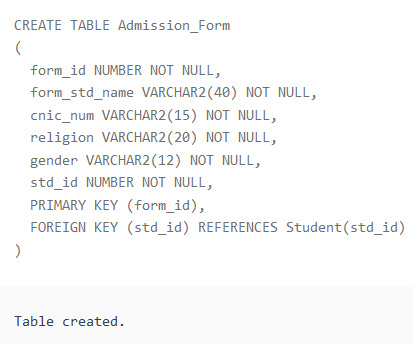
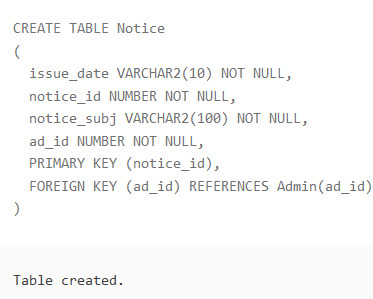
**ORACLE QUERIES:**

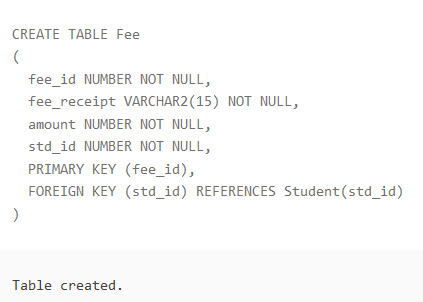
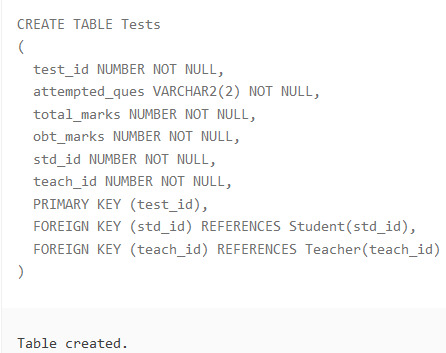


CREATE TABLES:

INSERTING RECORDS: