

# CASE STUDY FOR DATA CONCEPTS ON School Database Management System

Written By: VISHAL THAKUR(<https://github.com/VishalThakur10>)

Submitted To: DR. JUNAID QAZI (<https://www.linkedin.com/in/jqazi/>)

# Table of Contents

1.Introduction

2.Mission

3.Objectives

4.Database Design

5.Tables

6.ER Diagram

7.Datsbase Design

8.Appendix

## 1) INTRODUCTION

A school database management system is a software which helps schools to manage their daily activity like attendance tracking, timetable scheduling and fee collection.

The school management system saves their quality of time by using that database and they can keep their data store in one place and make it accessible when needed.

Moreover, it is a tool which helps schools to do their daily activity smoothly. Apart from this, it will help them to provide quality of education to students and can increase their learning experience.

## 2) Mission

Their main goal is to improve the quality of education by enabling effective administration and accurate data management.

Due to this technology, they have an aim to provide smart and paperless education, and both the students and teachers get benefits from it.

### 3) OBJECTIVES

The objective of the School Management System is to ensure smooth operations and better decision making.

Moreover, it will help management to manage the record of students such as exam, attendance and fees. This system would enhance accuracy and reduce paperwork by designing online processes.

Apart from this, it ensures data security and provides real-time updates.

It offers a cloud-based solution on multiple devices, which helps teachers, students and parents to stay connected from anywhere remotely.

### 4) DATABASE DESIGN

#### TABLES

#### 1) Students Table

Field Name	Data Type	Constraints
<b>Student_id</b>	INT(9)	PK
<b>First_name</b>	Varchar(50)	Not Null
<b>Last_name</b>	Varchar(50)	Not Null
<b>Dob</b>	Date	Not Null
<b>Contact</b>	Varchar(15)	Unique
<b>Address</b>	Text	
<b>Email</b>	Varchar	Not Null

## 2) Instructor Table:

Field Name	Data Type	Constraints
<b>Instructor_id</b>	INT	PK
<b>First_Name</b>	Varchar(50)	Not Null
<b>Last_Name</b>	Varchar(50)	Not Null
<b>Phone_Number</b>	Varchar(15)	Unique
<b>Email</b>	Varchar(50)	Unique
<b>Department_id</b>	INT	FK

## 3) Payment Mode Table

Field Name	Data Type	Constraints
<b>Payment_mode_id</b>	INT	PK
<b>Payment_mode_name</b>	Varchar(50)	Not Null
<b>Description</b>	Text	

## 4) Subject Table

Field Name	Data Type	Constraints
<b>Subject_id</b>	INT	PK
<b>Instructor_id</b>	INT	FK
<b>Subject_Name</b>	Varchar(50)	Not Null

## 5) ENTITY-RELATIONSHIP DIAGRAM (ERD)

The Student Database Management System is designed to streamline student administration in an educational institution. This ERD ensures that:

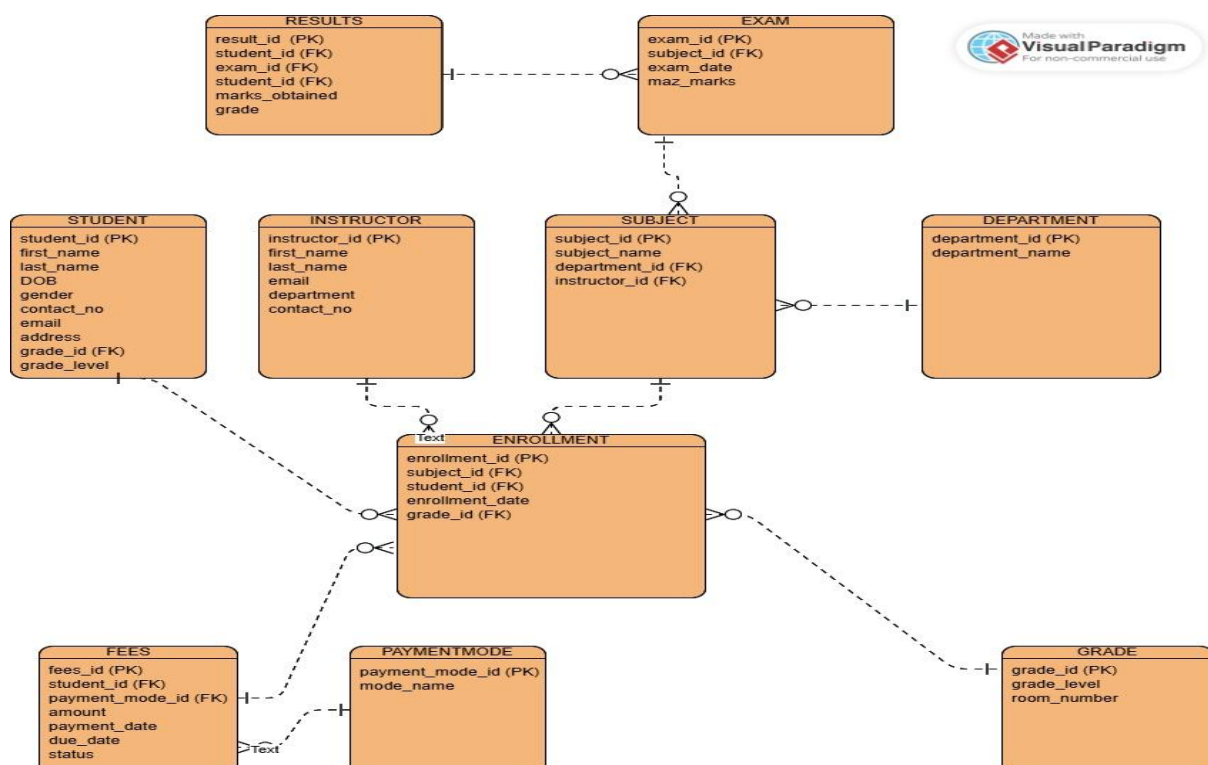
Students can enrol in subjects and be assigned grades.

Instructors can be linked to specific departments and subjects.

Exam results are systematically recorded and linked to students.

Fee payments are managed with different payment modes.

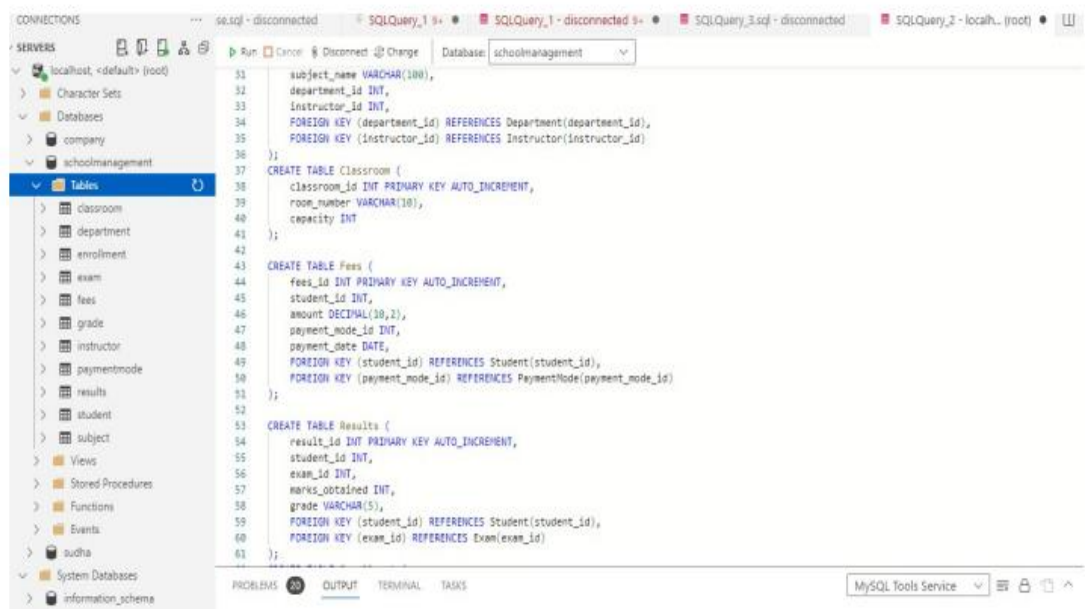
By implementing this database structure, the institution can efficiently manage academic and financial operations, ensuring accuracy, consistency, and scalability.



## Table Relationship

PRIMARY TABLE	PRIMARY KEY	RELATED TABLE	FOREIGN KEY	RELATIONSHIP TYPE	DESCRIPTION
Student	student_id	Enrollment	student_id	One-to-Many	A student can enroll in multiple subjects over time.
Grade	grade_id	Enrollment	grade_id	One-to-Many	Each enrollment belongs to a specific grade level.
Exam	exam_id	Subject	subject_id	Many-to-One	Each exam belongs to a subject.
Results	result_id	Student	student_id	Many-to-One	Each result is linked to a student.
Fees	fee_id	Student	student_id	Many-to-One	Each student has a fee record.

## 6) Database



## Database Design

## 7) APPENDIX

1) **Students Table:** This table stores details about the students who are enrolled in the school.

Results		Messages						
	student_id	first_name	last_name	dob	gender	address	contact_number	email
1	1	John	Doe	1998-05-15	Male	123 Elm Street	1234567890	john.doe@example.com
2	2	Alice	Johnson	1995-11-22	Female	789 Pine Road	2345678901	alice.johnson@example.com
3	3	Bob	Brown	1987-03-10	Male	654 Maple Lane	3456789012	bob.brown@example.com
4	4	Mary	Davis	2000-09-30	Female	321 Birch Boulevard	4567890123	mary.davis@example.com
5	5	Tom	Wilson	1992-01-14	Male	987 Cedar Drive	5678901234	tom.wilson@example.com
6	6	Emma	Moore	1984-06-28	Female	246 Oak Street	6789012345	emma.moore@example.com
7	7	James	Taylor	1997-04-09	Male	135 Pine Avenue	7890123456	james.taylor@example.com
8	8	Olivia	Martinez	2003-12-25	Female	753 Elm Road	8901234567	olivia.martinez@example.com
9	9	Sophia	Garcia	1999-08-12	Female	567 Birch Street	9012345678	sophia.garcia@example.com
10	10	William	Harris	1988-02-18	Male	432 Cedar Road	0123456789	william.harris@example.com

### Query to insert student table: -

```
CREATE TABLE Student (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    dob DATE,
    gender ENUM('Male', 'Female', 'Other'),
    address VARCHAR(100),
    contact_number VARCHAR(15),
    email VARCHAR(50)
);
```



2) **Instructor Table:** The instructor table contains information about the tutor working in the school.

	instructor_id	first_name	last_name	contact_number	email	department_id
1	1	Alice	Brown	1112223333	alice.brown@example.com	1
2	2	Bob	White	4445556666	bob.white@example.com	2
3	3	bobby	Brown	5552223333	bobby.brown@example.com	7
4	4	dua	lipa	1112456333	dua.lipa@example.com	5
5	5	nick	jonas	6662223333	nick.jonas@example.com	4
6	6	harry	potter	1116623333	harry.potter@example.com	3
7	7	mark	Brown	8882223333	mark.brown@example.com	6
8	8	goerge	doe	111278933	goerge.doe@example.com	1
9	9	julia	bloom	4442223333	julia.bloom@example.com	2
10	10	lily	Bloom	7712223333	lily.bloom@example.com	1

## Query to insert that table: -

```
CREATE TABLE Instructor (
    instructor_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    contact_number VARCHAR(15),
    email VARCHAR(100),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES
    Department(department_id)
);
```

3) **Payment Mode Table:** It manages payment transactions and statuses.

	mode_id ▼	mode_name ▼	description ▼
1	1	Credit Card	Payment made via credit card
2	2	Debit Card	Payment made via debit card
3	3	PayPal	Payment made through PayPal
4	4	Bank Transfer	Payment via direct bank transfer
5	5	Cash	Payment made in cash

### Query to insert payment mode table: -

```
CREATE TABLE PaymentMode
( PaymentModeID INT PRIMARY KEY AUTO_INCREMENT,
  ModeName VARCHAR(100) UNIQUE NOT NULL,
  Description TEXT );
```

4) **Subject Table:** It contains all the information about the subjects

	subject_id ▼	subject_name ▼	department_id ▼	instructor_id ▼
1	1	Database Management	1	1
2	2	Algebra	7	7
3	3	english grammar	2	2
4	4	french	4	4
5	5	biology	5	5
6	6	chemistry	5	5
7	7	history	6	6
8	8	political science	6	6
9	9	trigonometry	7	7
10	10	integration	7	7
11	11	economics	3	3
12	12	accounting	3	3
13	13	RDBMS	1	1

### Query to insert subject table: -

```
CREATE TABLE Subject (  
    subject_id INT PRIMARY KEY AUTO_INCREMENT,  
    subject_name VARCHAR(100),  
    department_id INT,  
    instructor_id INT,  
    FOREIGN KEY (department_id) REFERENCES  
Department(department_id),  
    FOREIGN KEY (instructor_id) REFERENCES Instructor(instructor_id)  
);
```

## Conclusion:

A School database management system helps institutions manage mandatory information about students, teachers, classes, exams, and payment. It saves time, reduces errors, and gives instant updates for better decisions and resource planning.

By using this system, the Institute can decline their manual work and improve data accuracy. Apart from this, it makes it easier to identify student performance and other records.