|  |  |
| --- | --- |
|  |  |
| Database Project School Management System |  |
|  |  |
|  | Group Members & Roll No  Saqlain Khan – 092  Taha Rasheed – 043  Kaif Tahir – 054 |
|  |  |

Table of Contents

[Database Project 1](#_Toc166862224)

[Chapter 1 4](#_Toc166862225)

[Introduction of School Mangement System 4](#_Toc166862226)

[Summary of database 4](#_Toc166862227)

[Each table has a specific role 4](#_Toc166862228)

[To ensure data correctness 5](#_Toc166862229)

[Data deletion and drop operations 5](#_Toc166862230)

[Chapter 2 Database Design and Table Structure 6](#_Toc166862231)

[- Confucius 7](#_Toc166862232)

# Chapter 1

## Introduction of School Mangement System

In today's educational world, managing school tasks efficiently is very important. A School Management System (SMS) is a tool that helps schools handle various tasks smoothly. This system includes different parts that help with managing student and staff information, keeping track of attendance, handling fees, and maintaining academic records.

The goal of the SMS is to make everyday tasks easier, reduce manual work, and provide quick access to important information. This allows teachers and school staff to focus more on teaching and student success. With features like safe data storage, easy access to information, and the ability to create detailed reports, the system supports better decision-making and helps schools run more smoothly.

## Summary of database

The School Management System includes a carefully designed database named schoolLab, which is set up to manage and store important school data. The database has several main tables: studentInfo, teacherInfo, deletedTeacherInfoLog, subjectInfo, TeacherAttendance, studentsFee, recordsOfFee, Result, employeeSalary, total, and passwordLogin.

Each table has a specific role: studentInfo holds detailed student records including personal, academic, and parental information. teacherInfo captures complete data on teachers such as personal details, job status, and contact information. Additionally, deletedTeacherInfoLog keeps a log of deleted teacher records for history. The subjectInfo table stores details about subjects offered and their respective classes. TeacherAttendance tracks teacher attendance, recording dates and attendance status, while studentsFee and recordsOfFee manage student fee details and payment statuses respectively. The Result table records student performance in various subjects, summarizing their academic results. The employeeSalary table tracks teacher salaries, and the total table aggregates important data such as total salary expenses, total number of students, and total fees collected. The passwordLogin table secures user credentials for system access.

To ensure data correctness, various checks like primary keys, foreign keys, unique constraints, and value validations are used across these tables. Sample data is inserted into key tables like studentInfo, teacherInfo, subjectInfo, studentsFee, and recordsOfFee to show and test the database functions. Teacher attendance records are started with a default status of 'P' (Present), and sample salary data is added into the employeeSalary table.

Data deletion and drop operations are clearly explained, including commands to remove records from all tables and to drop the passwordLogin table if needed. Specific deletion commands are provided to remove individual student records along with their related fee details.

The system also includes select statements to retrieve and view data from tables like total, employeeSalary, Result, studentInfo, teacherInfo, subjectInfo, TeacherAttendance, studentsFee, recordsOfFee, and passwordLogin.

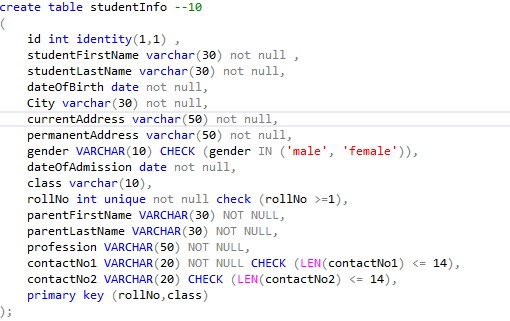
Finally, password management is addressed with an insertion command specifically for adding an admin password into the passwordLogin table, ensuring secure access to the system. This well-organized structure and operations plan ensures efficient, secure, and effective management of school data, helping with smooth administrative operations and improving the overall educational experience.

# Chapter 2

# Database Design and Table Structure

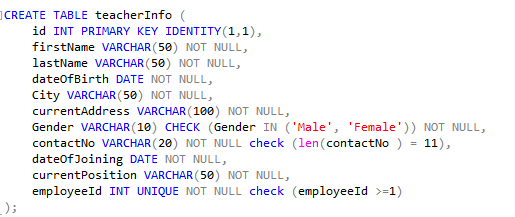
|  |  |  |
| --- | --- | --- |
|  |  |  |
| Our greatest glory is not in never failing, but in rising every time we fail. - Confucius |
|  |

## 1) Student Info



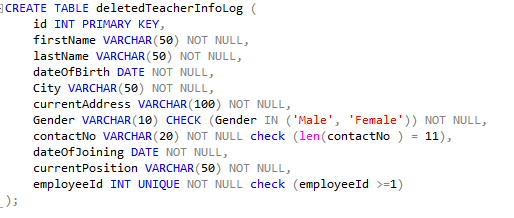
This table holds all information regarding student. We are generating id automatic and taking various inputs from the user such as student first name , last name etc. various checks and conditions are implemented so each record is error free. Primary key in this table is the combination of roll no and class because each student can easily be verified by their class and roll no

## 2) Teacher Info



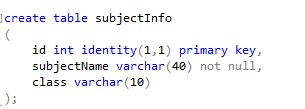
This table is responsible for storing all teacher records. Id is also automatic in this table but each employee has a unique employee code that will be assigned by the admin where the teacher joins the school. Primary key is also employeeId.

## 3) Deleted Teacher Info Log



This table works with the delete trigger of the table teacherInfo which we will explain further below, basically whenever a teacher record is deleted from the teacherInfo table the delete trigger will automatically insert the deleted teacher record in this table. If the admin wants to recover or want to see the teachers that have left the school , they can easily do it.

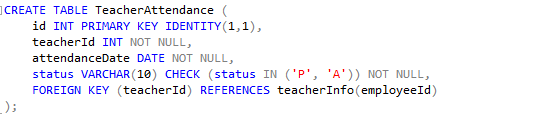
## 4) Subject Info



This table is responsible for holding all the subject names respective to their class. Id will be

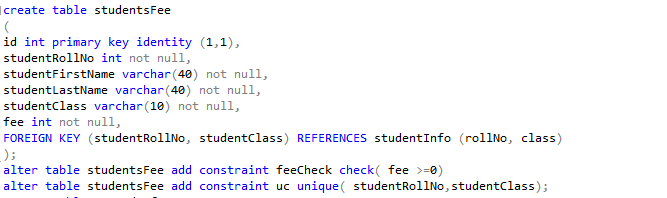
automatically generated. User will enter the class name and the subject name they want to add

## 5) Teacher Attendance



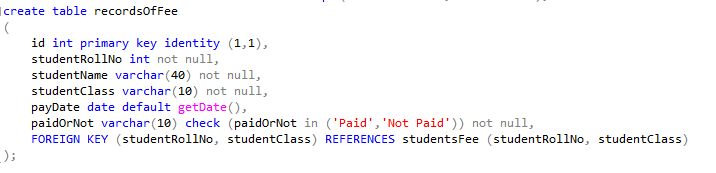
This table will have records of teacher attendance of each day. First the id will be auto generated. Then comes **teacherId**, this is used as a foreign key that references **employeeId** attribute in **teacherInfo** table. This is done because we don’t want attendance of that teacher that does not exist in our database right? That is why we have used foreign key to make sure that the user can only enter that **teacherId** that exists in the **teacherInfo** table. Attendance date will be picked automatically, and last comes status, option are P for Present and A for Absent. Necessary checks have been implemented to ensure data that is entered is correct.

## 6) Student Fee



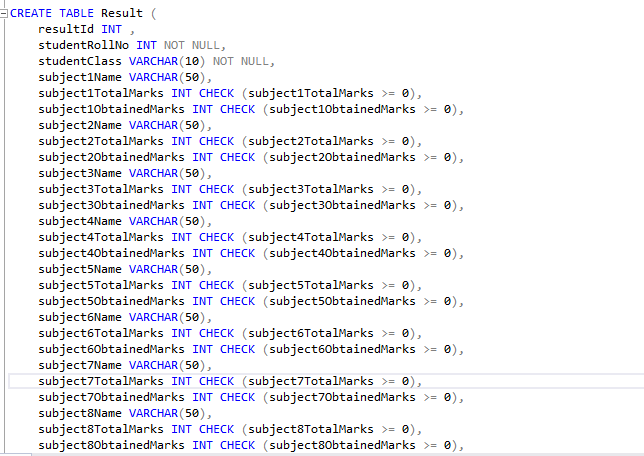
This table is responsible for storing the fee record of each student. Our way of doing this was simple, we will give the user option to enter student roll no and class, and **studentRollNo** and **studentClass** are foreign keys that are referencing **studentRollNo and StudentClass** in the original **studentInfo** table, to ensure that the roll no and class that the user have entered should exist in the original **studentInfo** table too.

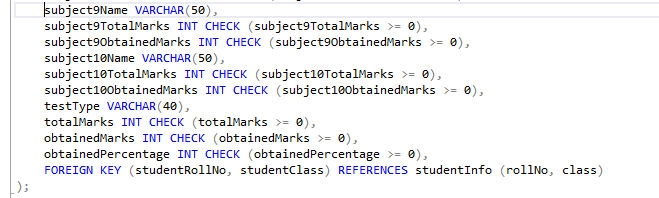
## 7) Records Of Fee



This table hold fee records of all the students that have paid the fee or not of the current month. Pay date will be auto picked as current day, then **studentRollNo** , **studentClass** are used as reference keys that are referencing the student roll no and student class in the original **studentFee** table and **studentFee** table is referencing **studentInfo** table, this chain of foreign keys ensures that all records that will be inserted into both tables will be correct and no wrong information can be entered.

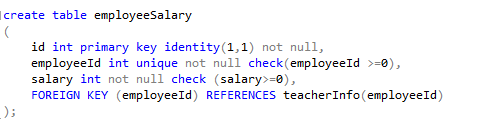
## 8) Results





This table is used to store the result of each student. First the admin will decide a resultId , each resultId will be associated with result of students of one class. For example: if user wants to store the 2nd term result of class 10 , they will enter sample data such as resultId = 1, and each student of that class will be linked to that resultId, if admin want to see the result of 2nd term result of some student of that class , they will enter the resultId and enter the roll no and class of that student and the result will show , we have given user options to enter each subject name , its obtained marks and its total marks , and the total marks and total obtained marks and obtained percentage will be calculated automatically using a trigger , so the user don’t have to the difficult task. **studentRollNo**, **studentClass** are foreign keys that are referencing attributes in **studentInfo** to ensure no wrong record should be entered in the result table

## 9) Employee Salary



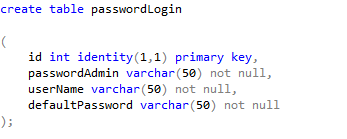
This table holds each employee salary. **employeeId** is used as a foreign key that is referencing **employeeId** in **teacherInfo** table. To ensure the user is entering the employeeId of that employee that exists in our record.

## 10) Total

## 

This table holds **total salary** that the admin have to pay to the employees , **total students** that are in the school and **total Fee** to collect each month from students. Each time a new record is inserted in the **studentInfo** , **studentsFee, employeeSalary** table this record is updated automatically using a trigger.

## 11) Password Login



This table holds the user name and password to login to the database , if the user forgot the user name or password , they will have to enter the default password to reset the user name and password