ANP-C7971

STUDENTID:AF0401714

NAME: Sathwika Madam

# DATABASE DESIGN FOR SCHOOL

# MANAGEMENT SYSTEM

# SCHOOL MANAGEMENT SYSTEM

Creating a school management system (SMS) involves designing a comprehensive database to handle various aspects of school operations.

This school management system database design provides a robust framework to manage student data, track academic progress, and streamline school operations efficiently. By focusing on relationships, normalization, and access control, you can create a scalable and maintainable system.

A school management system (SMS) database design is structured to effectively manage various aspects of school operations. It includes key entities such as Students, Teachers, Parents, Classes, Subjects, Grades, and Attendance, each with specific attributes. The design establishes relationships, allowing for multiple enrollments per student and multiple Subjects per teacher. Data normalization is employed to reduce redundancy and ensure integrity, while constraints like primary and foreign keys maintain accurate data. Teachers can oversee their Subjects and grades, and Students can view their own information. Additionally, the system supports reporting for performance and attendance, alongside regular backups and recovery plans to safeguard against data loss, providing a comprehensive framework for school management.

Designing a database for school Management System includes different entities, attributes along with their relationships.

# ENTITIES:

* Students
* Teachers
* Classes
* Subjects
* Grades
* Attendance
* Parents

# ATTRIBUTES:

1. Students:

* Stu\_id
* Name
* DOB
* Ph\_no
* E\_mail
* Address
* Enroll\_date

1. Teachers:

* Teacher\_id
* Teacher\_name
* Teacher\_ph\_no
* Teacher\_e\_mail

1. Classes:
   * Class\_id
   * Class\_name
2. Subjects:
   * Sub\_id
   * Sub\_name
3. Grades:
   * Grade\_value
4. Attendance:
   * + Att\_date
     + Status
5. Parents:
   * Parent\_name
   * Parent\_ph\_no
   * Parent\_e\_mail

# ENTITY-RELATIONS:

* *Students to classes:*

Many-to-one relationship (A student can belong to one class, but a class can have many students).

* *Teacher to classes:*

Many-to-Many relationship (Many teachers can teach multiple classes, and multiple classes can have Many teachers).

* *Classes to Subjects:*

One-to-Many relationship (A class can have multiple subjects).

* *Students to Grades:*

Many-to-Many relationship (A student can have many grades, and a grade can be awarded to many students for different subjects).

* *Students to Attendance:*

Many-to-Many relationship (A student can have multiple attendance records).

* *Student to Parents:*

Many-to-One relationship (A parent can have multiple children enrolled in the school).

* *Students to Subjects:*

Many-to-Many relationship (A student can have many Subjects, and

Subjects can be hold by Many students).

M M

Sub\_Name

Sub\_id

b\_Id

subjects

have

Students

learn

classes

Teacher \_id

Teacher\_id

Parents

Sub\_id

Class\_id

Grades

Sub\_id

Enroll\_date

Stu\_id

Stu\_id

Address

Name

Ph\_no

DOB

Gender

M

Email

M M

Belongs

1

1

Awards

Teacher\_E-mail

M

ClassName

Teacher\_Ph\_no

Class\_id

teaches

M

Teacher\_Name

Teachers

M

M

G\_value

Stu\_id

have

M

Stu\_id

has

Parent\_Ph\_no

Parent\_name

M

Attendance

Class\_id

Parent\_e-mail

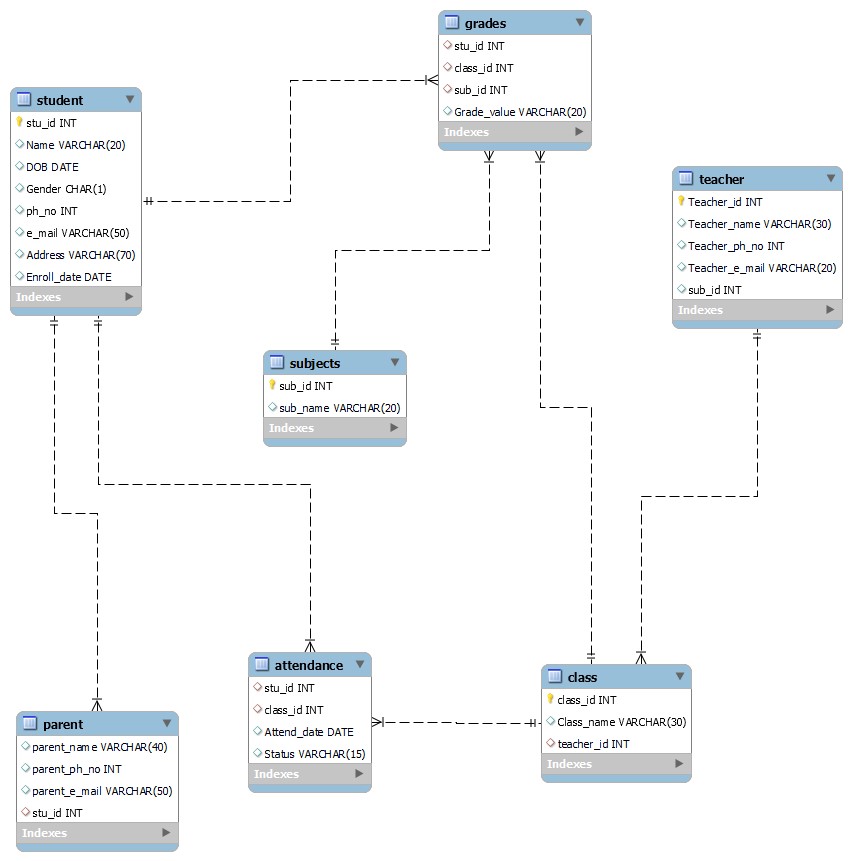
1

status

student\_id

Date

ER-Diagram



Reverse Engineering