PROJECT NAME:

School Management System

Name : Gauri . Aniket .Arekar

INTRODUCTION

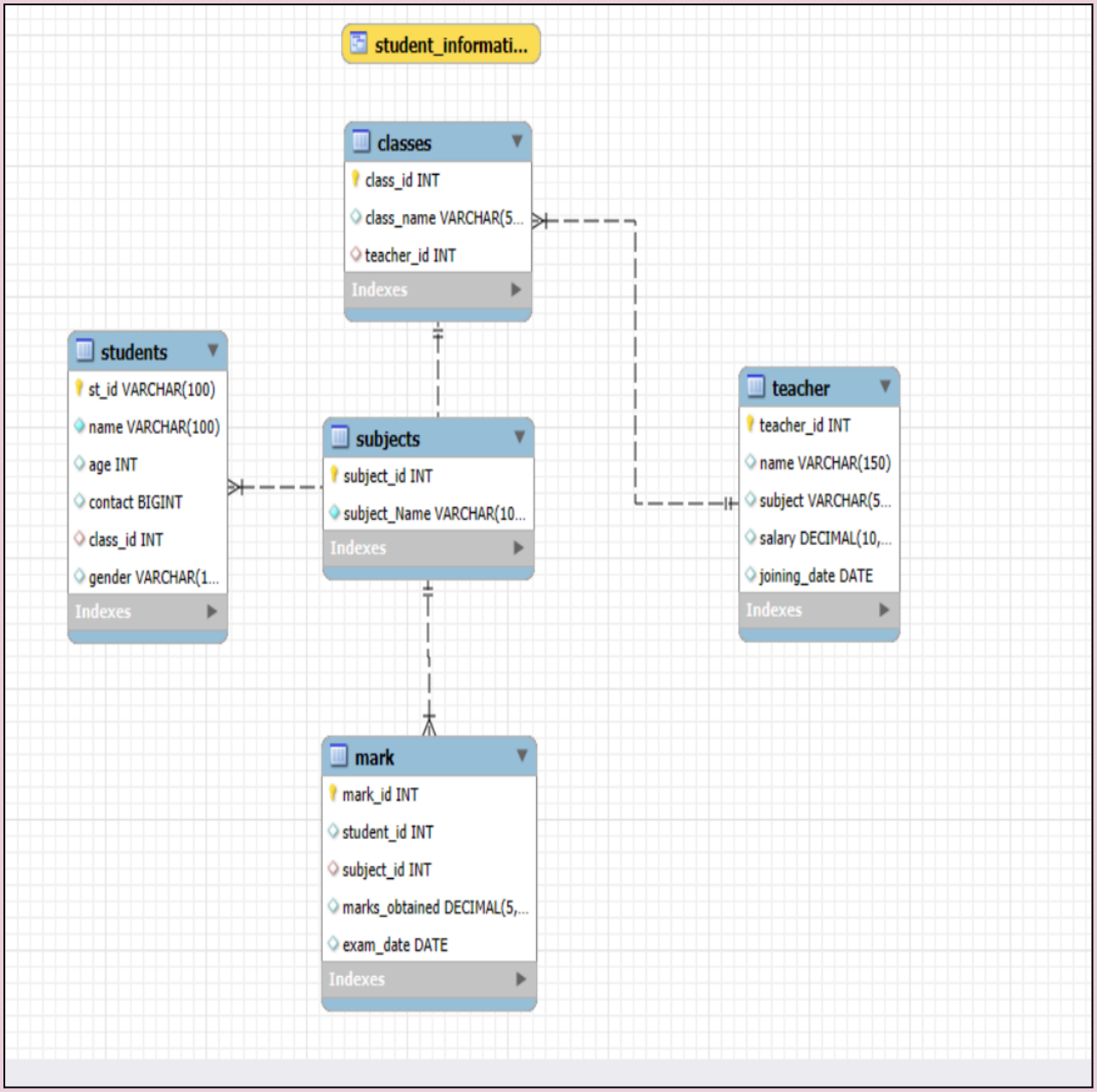
The School Management System is designed to efficiently track and manage critical academic and administrative data related to students, teachers, classes, subjects, and academic performance. This system is an SQL-based project that simulates the day-to-day operations of a real-world school environment, with a focus on maintaining structured, reliable, and accessible data for educational institutions.

The primary goal of this system is to provide school administrators, teachers, and academic staff with quick access to essential student and class information, enabling informed decision-making and enhanced academic planning. By organizing student records, teacher assignments, class schedules, and subject distributions, the system ensures that educational data is consistently managed and readily available.

From a technical perspective, the project includes the creation of relational tables, views, joins, and SQL queries that support efficient data retrieval and analysis. The system uses SQL to perform various data operations, including filtering, grouping, sorting, and aggregation. Advanced SQL techniques such as subqueries, inner and outer joins, and views are employed to generate comprehensive reports and insights into the school’s operations.

This project is particularly relevant to my Statistics background, as it allows for the analysis of academic performance, student demographics, subject-wise results, and teacher workload. Statistical concepts are naturally integrated into the system's query logic, enabling data-driven evaluation of academic trends, pass/fail rates, attendance patterns, and resource utilization. The project not only strengthens my SQL and database design skills but also illustrates how statistical analysis can enhance decision-making in educational administration and policy.

ER DIAGRAM


student_informati...
classes
class_id INT
class_name VARCHAR(5...
teacher_id INT
Indexes
students
st_id VARCHAR(100)
name VARCHAR(100)
age INT
contact BIGINT
class_id INT
gender VARCHAR(1...
Indexes
subjects
subject_id INT
subject_Name VARCHAR(10...
Indexes
teacher
teacher_id INT
name VARCHAR(150)
subject VARCHAR(5...
salary DECIMAL(10,...
joining_date DATE
Indexes
mark
mark_id INT
student_id INT
subject_id INT
marks_obtained DECIMAL(5,...
exam_date DATE
Indexes

Databases :

create database school_db;

use school_db;

Show databases;

Result Grid | Filter Rows:

Database
college
company
information_schema
joins_sql
mysql
performance_schema
school_db
sys

Tables in School_db :

show tables;

Result Grid | Filter Row

Tables_in_school_db
classes
mark
marks
student
student_information
students
subjects
teacher
teachers

teacher

1.DATA DEFINITION LANGUAGE (DDL):

1.Creating Tables :

A)Students

CREATE TABLE students (student_id INT PRIMARY KEY,name VARCHAR(100) NOT NULL,age INT,contact bigint unique,class_id INT);
desc students;

Result Grid | Filter Rows: | Export: | Wra

Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
age	int	YES		NULL	
contact	bigint	YES	UNI	NULL	
class_id	int	YES		NULL	

B)Teacher

CREATE TABLE teacher(teacher_id INT PRIMARY KEY,name VARCHAR(100),subject VARCHAR(50),salary DECIMAL(10, 2),joining_date DATE);
desc teacher;

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
teacher_id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
subject	varchar(50)	YES		NULL	
salary	decimal(10,2)	YES		NULL	
joining_date	date	YES		NULL	

C)Classes

CREATE TABLE classes (class_id INT PRIMARY KEY,class_name VARCHAR(50),teacher_id INT,FOREIGN KEY (teacher_id) REFERENCES teachers(teacher_id));
desc classes;

Result Grid | Filter Rows: | Export: | Wr

Field	Type	Null	Key	Default	Extra
class_id	int	NO	PRI	NULL	
class_name	varchar(50)	YES		NULL	
teacher_id	int	YES	MUL	NULL	

D) Non_teaching_staff

CREATE TABLE non_teaching_staff (staff_id INT PRIMARY KEY,name VARCHAR(100) NOT NULL,salary DECIMAL(10, 2),joining_date DATE);

desc non_teaching_staff;

Field	Type	Null	Key	Default	Extra
staff_id	int	NO	PRI	NULL	
nam staff_id	varchar(100)	NO		NULL	
salary	decimal(10,2)	YES		NULL	
joining_date	date	YES		NULL	

E)Subject

CREATE TABLE subjects (subject_id INT PRIMARY KEY,name VARCHAR(100) NOT NULL);

desc subjects;

Field	Type	Null	Key	Default	Extra
subject_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	

F)Marks

CREATE TABLE marks (mark_id INT PRIMARY KEY,student_id INT,subject_id INT,marks_obtained DECIMAL(5,2),exam_date DATE,FOREIGN KEY (student_id) REFERENCES student(student_id),

FOREIGN KEY (subject_id) REFERENCES subjects(subject_id));

desc marks;

Result Grid | Filter Rows: | Export: | Wrap C

Field	Type	Null	Key	Default	Extra
mark_id	int	NO	PRI	NULL	
student_id	int	YES	MUL	NULL	
subject_id	int	YES	MUL	NULL	
marks_obtained	decimal(5,2)	YES		NULL	
exam_date	date	YES		NULL	

2.Alter table :

Alter Table : Add column

ALTER TABLE students ADD gender VARCHAR(10);

desc students;

Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
age	int	YES		NULL	
contact	bigint	YES	UNI	NULL	
class_id	int	YES		NULL	
gender	varchar(10)	YES		NULL	

Alter Table : Modify Column

ALTER TABLE teacher MODIFY name VARCHAR(150);

desc teacher;

Result Grid | Filter Rows: | Export: | Wr

Field	Type	Null	Key	Default	Extra
teacher_id	int	NO	PRI	NULL	
name	varchar(150)	YES		NULL	
subject	varchar(50)	YES		NULL	
salary	decimal(10,2)	YES		NULL	
joining_date	date	YES		NULL	

Alter Table : Rename column

ALTER TABLE students CHANGE student_id st_id VARCHAR(100);

desc students;

Result Grid | Filter Rows: | Export: | W

Field	Type	Null	Key	Default	Extra
st_id	varchar(100)	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
age	int	YES		NULL	
contact	bigint	YES	UNI	NULL	
class_id	int	YES		NULL	
gender	varchar(10)	YES		NULL	

Alter Table : Drop column

ALTER TABLE non_teaching_staff DROP COLUMN salary;

desc non_teaching_staff;

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
staff_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
joining_date	date	YES		NULL	

Alter Table : Rename table

RENAME TABLE non_teaching_staff TO support_staff;

desc support_staff;

Field	Type	Null	Key	Default	Extra
staff_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
joining_date	date	YES		NULL	

3.Truncate table :

TRUNCATE TABLE support_staff;

Result Grid | Filter Rows:

staff_id	name	joining_date
NULL	NULL	NULL

4.Drop Table :

Drop table support_staff;

2.DATA MANIPULATION LANGUAGE (DML):

1. Insert into table :

INSERT INTO students (st_id, name, age, contact, class_id, gender) VALUES
(1, 'Anjali Sharma', 15, 9876543211, 101, 'Female'),
select * from students;

Result Grid | Filter Rows: | Edit:

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	15	9876543211	101	Female
10	Aman Yadav	14	9876543220	103	Male
2	Vikram Singh	17	9876543212	102	Male
3	Priya Mehta	16	9876543213	101	Female
4	Arjun Verma	15	9876543214	103	Male
5	Sonal Gupta	14	9876543215	103	Female
6	Rohan Joshi	16	9876543216	102	Male
7	Neha Reddy	17	9876543217	101	Female
8	Karan Kapoor	15	9876543218	104	Male
9	Meera Nair	16	9876543219	104	Female
NULL	NULL	NULL	NULL	NULL	NULL

2. Update into Table :

Update contact and Age in Student table

UPDATE students SET contact = 9876500000, age = 17 WHERE st_id = 1;
select * from students;

Result Grid | Filter Rows: | Edit:

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	17	9876500000	101	Female
10	Aman Yadav	14	9876543220	103	Male
2	Vikram Singh	17	9876543212	102	Male
3	Priya Mehta	16	9876543213	101	Female
4	Arjun Verma	15	9876543214	103	Male
5	Sonal Gupta	14	9876543215	103	Female
6	Rohan Joshi	16	9876543216	102	Male
7	Neha Reddy	17	9876543217	101	Female
8	Karan Kapoor	15	9876543218	104	Male
9	Meera Nair	16	9876543219	104	Female
NULL	NULL	NULL	NULL	NULL	NULL

3)Delete from table :

Delete record having subject_id 3.

DELETE FROM subjects WHERE subject_id= 3;

select * from subjects;

Result Grid | Filter Rows:

subject_id	name
1	Mathematics
2	Science
4	History
5	Geography
6	Computer Science
NULL	NULL

3.DATA QUERY LANGUAGE (DQL) :

1.Select Query:

a) Select Query for entire data

select * from students;

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	17	9876500000	101	Female
10	Aman Yadav	14	9876543220	103	Male
2	Vikram Singh	17	9876543212	102	Male
3	Priya Mehta	16	9876543213	101	Female
4	Arjun Verma	15	9876543214	103	Male
5	Sonal Gupta	14	9876543215	103	Female
6	Rohan Joshi	16	9876543216	102	Male
7	Neha Reddy	17	9876543217	101	Female
8	Karan Kapoor	15	9876543218	104	Male
9	Meera Nair	16	9876543219	104	Female
NULL	NULL	NULL	NULL	NULL	NULL

b)Select specific data i.e st_id and name

select st_id,name from students;

st_id	name
1	Anjali Sharma
10	Aman Yadav
2	Vikram Singh
3	Priya Mehta
4	Arjun Verma
5	Sonal Gupta
6	Rohan Joshi
7	Neha Reddy
8	Karan Kapoor
9	Meera Nair
NULL	NULL

c)Select query with changing Column name

select name as fullname from students;

Result Grid

fullname
Anjali Sharma
Aman Yadav
Vikram Singh
Priya Mehta
Arjun Verma
Sonal Gupta
Rohan Joshi
Neha Reddy
Karan Kapoor
Meera Nair

2.Order by

a)List of Teachers in ascending order by salary.

Select*from teacher order by salary;

Result Grid | Filter Rows: | Edit:

teacher_id	name	subject	salary	joining_date
4	Rajiv Nair	History	44000.00	2017-09-23
1	Sunita Desai	Mathematics	45000.00	2018-06-15
3	Anita Mehra	English	46000.00	2020-01-10
2	Ravi Patel	Science	47000.00	2019-07-01
5	Megha Reddy	Computer Science	48000.00	2021-03-05
NULL	NULL	NULL	NULL	NULL

b)List of subject in descending order by subject_id

Select*from subjects order by subject_id desc;

Result Grid | Filter Rows:

subject_id	name
6	Computer Science
5	Geography
4	History
2	Science
1	Mathematics
NULL	NULL

3.Limit Query

Display top 5 student with highest marks

Select * from marks order by marks_obtained desc limit 5;

Result Grid | Filter Rows: | Export:

student_id	marks_obtained	total_marks	exam_date
10	95	100	2025-05-02
3	92	100	2025-05-01
7	91	100	2025-05-02
6	88	100	2025-05-02
1	85	100	2025-05-01

4.Distinct Query

Display Unique Gender from Students.

Select distinct gender from students;

gender
Female
Male

5.Where Clause:

1)With Comparison Operator

SELECT * FROM students WHERE age > 15;

Result Grid | Filter Rows: | Edit:

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	17	9876500000	101	Female
2	Vikram Singh	17	9876543212	102	Male
3	Priya Mehta	16	9876543213	101	Female
6	Rohan Joshi	16	9876543216	102	Male
7	Neha Reddy	17	9876543217	101	Female
9	Meera Nair	16	9876543219	104	Female
NULL	NULL	NULL	NULL	NULL	NULL

2)Logical Operator

Using AND Operator

SELECT * FROM students WHERE age < 16 and gender='Female';

st_id	name	age	contact	class_id	gender
5	Sonal Gupta	14	9876543215	103	Female
NULL	NULL	NULL	NULL	NULL	NULL

Using AND/OR Operator

Find teacher who teach english or mathematics and salary >45000

SELECT * FROM teacher WHERE (subject = 'English' OR subject = 'Mathematics') AND salary > 45000;

teacher_id	name	subject	salary	joining_date
3	Anita Mehra	English	46000.00	2020-01-10
NULL	NULL	NULL	NULL	NULL

Using NOT Operator

Find Teachers who do NOT teach 'Mathematics'

SELECT * FROM teacher WHERE NOT subject = 'Mathematics';

Result Grid | Filter Rows: | Edit: | Ex

teacher_id	name	subject	salary	joining_date
2	Ravi Patel	Science	47000.00	2019-07-01
3	Anita Mehra	English	46000.00	2020-01-10
4	Rajiv Nair	History	44000.00	2017-09-23
5	Megha Reddy	Computer Science	48000.00	2021-03-05
NULL	NULL	NULL	NULL	NULL

Using Not null

Find teachers where subject is NOT NULL

SELECT * FROM teacher WHERE subject IS NOT NULL;

Result Grid | Filter Rows: | Edit: | Exp

teacher_id	name	subject	salary	joining_date
1	Sunita Desai	Mathematics	45000.00	2018-06-15
2	Ravi Patel	Science	47000.00	2019-07-01
3	Anita Mehra	English	46000.00	2020-01-10
4	Rajiv Nair	History	44000.00	2017-09-23
5	Megha Reddy	Computer Science	48000.00	2021-03-05
NULL	NULL	NULL	NULL	NULL

Using BETWEEN operator

Find students who scored between 70 and 90

SELECT * FROM marks WHERE marks_obtained BETWEEN 70 AND 90;

Result Grid | Filter Rows: | Export:

student_id	marks_obtained	total_marks	exam_date
1	85	100	2025-05-01
2	78	100	2025-05-01
5	74	100	2025-05-01
6	88	100	2025-05-02
9	80	100	2025-05-02

Using IN operator

Get students in class 101, 103, or 104

SELECT * FROM students WHERE class_id IN (101, 103, 104);

Result Grid | Filter Rows: | Edit:

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	17	9876500000	101	Female
10	Aman Yadav	14	9876543220	103	Male
3	Priya Mehta	16	9876543213	101	Female
4	Arjun Verma	15	9876543214	103	Male
5	Sonal Gupta	14	9876543215	103	Female
7	Neha Reddy	17	9876543217	101	Female
8	Karan Kapoor	15	9876543218	104	Male
9	Meera Nair	16	9876543219	104	Female
NULL	NULL	NULL	NULL	NULL	NULL

Using ANY operator

Get students whose age is greater than any age of students in class 101

SELECT * FROM students WHERE age > ANY (SELECT age FROM students WHERE class_id = 101);

Result Grid | Filter Rows: | Edit:

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	17	9876500000	101	Female
2	Vikram Singh	17	9876543212	102	Male
7	Neha Reddy	17	9876543217	101	Female
NULL	NULL	NULL	NULL	NULL	NULL

Using ALL operator

Get students younger than all students in class 104

SELECT * FROM students WHERE age < ALL (SELECT age FROM students WHERE class_id = 104);

Result Grid | Filter Rows: | Edit:

st_id	name	age	contact	class_id	gender
10	Aman Yadav	14	9876543220	103	Male
5	Sonal Gupta	14	9876543215	103	Female
NULL	NULL	NULL	NULL	NULL	NULL

6.Aggregate Functions:

Count Function:

Find the total number of Students.

Select count(*) as total_number_students from students;

Result Grid | Filter Ro

total_number_students
10

Average Function with round function

Find the average marks of all students in two decimal places.

Select round(avg(marks_obtained) , 2)as average_students_marks from marks;

Result Grid | Filter Rows:

average_students_marks
80.80

Sum Function :

Display total Salary Paid to teachers

select sum(salary) as total_teacher_salary from teacher;

Result Grid | Filter Rows:

total_teacher_salary
230000.00

Max, Min Function:

Find the highest and lowest salary of a teacher.

Select max(salary) as highest_salary,min(salary) as lowest_salary from teacher;

Result Grid | Filter Rows:

highest_salary	lowest_salary
48000.00	44000.00

7.Group by clause :

Count of Students per Class

SELECT class_id, COUNT(*) AS student_count FROM students GROUP BY class_id;

Result Grid | Filter Rows:

class_id	student_count
101	3
103	3
102	2
104	2

Find the Count of Male and Female Students per Class

SELECT class_id, gender, COUNT(*) AS count FROM students GROUP BY class_id, gender;

Result Grid | Filter Rows:

class_id	gender	count
101	Female	3
103	Male	2
102	Male	2
103	Female	1
104	Male	1
104	Female	1

Find the Average Age per Class

Average Age per Class

SELECT class_id, AVG(age) AS average_age FROM students GROUP BY class_id;

Result Grid | Filter Row

class_id	average_age
101	16.6667
103	14.3333
102	16.5000
104	15.5000

8.Having Clause:

Find the Classes with more than 2 students

SELECT class_id, COUNT(*) AS student_count FROM students GROUP BY class_id HAVING COUNT(*) > 2;

Result Grid | Filter Rows:

class_id	student_count
101	3
103	3

9.Like Operator :

Find teachers whose names start with 'S'

SELECT * FROM teacher WHERE name LIKE 'S%';

Result Grid | Filter Rows: | Edit:

teacher_id	name	subject	salary	joining_date
1	Sunita Desai	Mathematics	45000.00	2018-06-15
NULL	NULL	NULL	NULL	NULL

Find teachers whose name ends with 'dy'

SELECT * FROM teacher WHERE name LIKE '%dy';

Result Grid | Filter Rows: | Edit: | Expo

teacher_id	name	subject	salary	joining_date
5	Megha Reddy	Computer Science	48000.00	2021-03-05
NULL	NULL	NULL	NULL	NULL

10.Union :

List all teacher IDs (from teacher) and teacher IDs assigned to classes (from classes)

SELECT teacher_id FROM teacher UNION SELECT teacher_id FROM classes;

Result Grid

teacher_id
1
2
3
4
5

11.Joins :

Join teacher with subjects on subject name

SELECT t.teacher_id, t.name AS teacher_name, t.subject, s.subject_id
FROM teacher t INNER JOIN subjects s ON t.subject = s.name;

Result Grid | Filter Rows: | Export:

teacher_id	teacher_name	subject	subject_id
1	Sunita Desai	Mathematics	1
2	Ravi Patel	Science	2
4	Rajiv Nair	History	4
5	Megha Reddy	Computer Science	6

Write a query to get all classes along with their teachers — including classes that might not have a matching teacher

SELECT c.class_id,c.class_name,c.teacher_id,t.name AS teacher_name,t.subject
FROM classes c
LEFT JOIN teacher t ON c.teacher_id = t.teacher_id;

Result Grid | Filter Rows: | Export: | Wrap Cell

class_id	class_name	teacher_id	teacher_name	subject
101	Class 10-A	1	Sunita Desai	Mathematics
102	Class 10-B	2	Ravi Patel	Science
103	Class 9-A	3	Anita Mehra	English
104	Class 9-B	4	Rajiv Nair	History
105	Class 8-A	5	Megha Reddy	Computer Science
106	Class 8-B	2	Ravi Patel	Science
107	Class 7-A	1	Sunita Desai	Mathematics
108	Class 7-B	3	Anita Mehra	English
109	Class 6-A	4	Rajiv Nair	History
110	Class 6-B	5	Megha Reddy	Computer Science

12.Subqueries:

1.Single row Subqueries:

Find the name of the student who scored the highest mark

SELECT name FROM students WHERE st_id = (SELECT student_id FROM marks ORDER BY marks_obtained DESC LIMIT 1);

Result Grid

name
Aman Yadav

2.Multiple row subquery:

List the subjects in which marks have been given (from `marks` table)

SELECT name FROM subjects WHERE subject_id IN (SELECT DISTINCT subject_id FROM marks);

Result Grid

name
Mathematics
Science
History
Geography
Computer Science

3.Multiple column subquery :

Find student names and class_ids where students scored more than 85 in any subject

SELECT name, class_id FROM students WHERE st_id IN (SELECT student_id FROM marks WHERE marks_obtained > 85);

Result Grid Filter R

name	class_id
Aman Yadav	103
Priya Mehta	101
Rohan Joshi	102
Neha Reddy	101

13.VIEW:

create view : Student information

SELECT st_id,name,age,contact,class_id,gender
FROM students;
SELECT * FROM Student_information;

Result Grid Filter Rows: Export:

st_id	name	age	contact	class_id	gender
1	Anjali Sharma	17	9876500000	101	Female
10	Aman Yadav	14	9876543220	103	Male
2	Vikram Singh	17	9876543212	102	Male
3	Priya Mehta	16	9876543213	101	Female
4	Arjun Verma	15	9876543214	103	Male
5	Sonal Gupta	14	9876543215	103	Female
6	Rohan Joshi	16	9876543216	102	Male
7	Neha Reddy	17	9876543217	101	Female
8	Karan Kapoor	15	9876543218	104	Male
9	Meera Nair	16	9876543219	104	Female