



School of Computer Science, UPES, Dehradun.

A

LABORATORY FILE

On

DATABASE MANAGEMENT
SYSTEM (DBMS) LAB

B.TECH. -III Semester

Submitted by:

Name: Mayank Negi

SAP ID: 500120279

Roll No: R2142230039

Batch: 2

Experiment 09

Use of different SQL clauses and join

Aim:

To design and implement a relational database for managing student and sports match data, enabling analysis of student participation in various matches.

Problem Statement:

The task involves creating a database to store information about students and the matches they play in, facilitating queries to extract insights on student participation, match venues, and student demographics.

Theory:

Relational databases organize data into structured tables, allowing for efficient querying and manipulation. SQL (Structured Query Language) is utilized for data definition and manipulation. Key concepts include:

- Tables: Structures to hold data in rows and columns.
- Primary Keys: Unique identifiers for records in a table, ensuring data integrity.
- Foreign Keys: References to primary keys in other tables, establishing relationships.
- JOIN Operations: Methods to combine rows from two or more tables based on related columns.
- Aggregate Functions: Functions such as `AVG()`, `COUNT()`, and `DISTINCT` to perform calculations and return summary information.

Commands Used:

1. Database and Table Creation
2. Data Insertion
3. Data Retrieval Queries

Results:

```
1      -- Ayush Vashishth
2      -- 500119331
3
4      ● CREATE DATABASE exp9;
5      ● USE exp9;
6
7      -- Creating Tables:
8      ● ○ CREATE TABLE Student (
9          sid INT PRIMARY KEY,
10         sname VARCHAR(50),
11         age INT
12     );
13
14      ● ○ CREATE TABLE Matchh (
15         mid VARCHAR(10) PRIMARY KEY,
16         mname VARCHAR(50),
17         venue VARCHAR(50)
18     );
19      ● ○ CREATE TABLE Play (
20         sid INT,
21         mid VARCHAR(10),
22         day DATE,
23         PRIMARY KEY (sid, mid),
24         FOREIGN KEY (sid) REFERENCES Student(sid),
25         FOREIGN KEY (mid) REFERENCES Matchh(mid)
26     );
27
```

```

28      -- Populating Tables:
29  •   INSERT INTO Student (sid, sname, age) VALUES
30      (1, 'Amit', 20),
31      (2, 'Ravi', 22),
32      (3, 'Suresh', 19),
33      (4, 'Priya', 21);
34  •   INSERT INTO Matchh (mid, mname, venue) VALUES
35      ('B10', 'Football', 'Delhi'),
36      ('B11', 'Cricket', 'Mumbai'),
37      ('B12', 'Basketball', 'Delhi'),
38      ('B13', 'Hockey', 'Chennai');
39  •   INSERT INTO Play (sid, mid, day) VALUES
40      (1, 'B10', '2024-09-01'),
41      (2, 'B11', '2024-09-01'),
42      (1, 'B12', '2024-09-02'),
43      (3, 'B10', '2024-09-03'),
44      (4, 'B11', '2024-09-04');
45
46      -- Find all information of students who have played matchh number B10.
47  •   SELECT s.*
48      FROM Student s
49      JOIN Play p ON s.sid = p.sid
50      WHERE p.mid = 'B10';
51      -- Find the name of matches played by Amit.
52  •   SELECT m.mname
53      FROM Student s
54      JOIN Play p ON s.sid = p.sid
55      JOIN Matchh m ON p.mid = m.mid
56      WHERE s.sname = 'Amit';
57      -- Find the names of students who have played a match in Delhi.
58  •   SELECT DISTINCT s.sname
59      FROM Student s
60      JOIN Play p ON s.sid = p.sid
61      JOIN Matchh m ON p.mid = m.mid
62      WHERE m.venue = 'Delhi';
63      -- Find the names of students who have played at least one match.
64  •   SELECT DISTINCT s.sname
65      FROM Student s
66      JOIN Play p ON s.sid = p.sid;
67      -- Find the ids and names of students who have played two different matches on the same day.
68  •   SELECT s.sid, s.sname
69      FROM Student s
70      JOIN Play p1 ON s.sid = p1.sid
71      JOIN Play p2 ON s.sid = p2.sid AND p1.mid != p2.mid AND
72      p1.day = p2.day;
73      -- Find the ids of students who have played a match in Delhi or Mumbai.
74  •   SELECT DISTINCT s.sid
75      FROM Student s
76      JOIN Play p ON s.sid = p.sid
77      JOIN Matchh m ON p.mid = m.mid
78      WHERE m.venue IN ('Delhi', 'Mumbai');
79      -- Find the average age of students.
80  •   SELECT AVG(age) AS average_age
81      FROM Student;

```

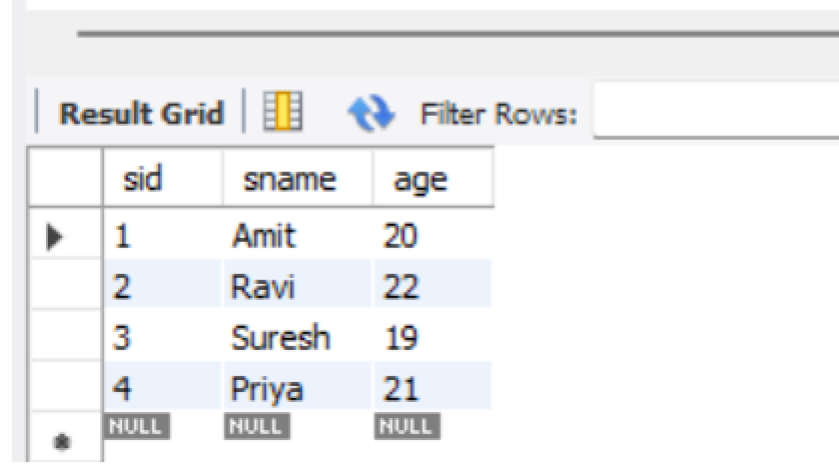
```
1 • SELECT * FROM exp9.matchh;
```

Result Grid			
Filter Rows:			
	mid	mname	venue
▶	B 10	Football	Delhi
	B 11	Cricket	Mumbai
	B 12	Basketball	Delhi
	B 13	Hockey	Chennai
●	NULL	NULL	NULL

```
1 • SELECT * FROM exp9.play;
```

Result Grid			
Filter Rows:			
	sid	mid	day
▶	1	B 10	2024-09-01
	1	B 12	2024-09-02
	2	B 11	2024-09-01
	3	B 10	2024-09-03
	4	B 11	2024-09-04
●	NULL	NULL	NULL

```
1 • SELECT * FROM exp9.student;
```



The screenshot shows a database interface with a 'Result Grid' tab. Above the grid is a 'Filter Rows:' input field. The grid contains a table with four columns: 'sid', 'sname', and 'age'. There are four data rows and one row with NULL values. The first row is highlighted with a blue background.

	sid	sname	age
▶	1	Amit	20
	2	Ravi	22
	3	Suresh	19
	4	Priya	21
•	NULL	NULL	NULL

Conclusion:

The SQL code effectively establishes a relational database to manage student and match data. Various queries demonstrate the ability to analyze student participation in matches based on specific criteria, such as match location and student demographics. The design supports efficient data retrieval and can be expanded for additional analyses, such as tracking performance or more detailed demographic studies.

Future enhancements could involve adding more attributes to students and matches, such as performance statistics or additional match details, to provide deeper insights into student engagement in sports.