ABHINAV PATEL 500119461 R2142230047 B.TECH CSE BATCH 2 DBMS LAB

EXPERIMENT - 8

Title: Use of different SQL clauses and join

Objective: To understand the use of group by and having clause and execute the

SQL commands using JOIN

1. Consider the following schema:

Student (sid, sname, age)

Match (mid, mname, venue)

Play (sid, mid, day(date))

```
-- Create Database and Use It
          CREATE DATABASE sports_management;
          USE sports_management;
          -- Create Tables
      CREATE TABLE Student (
             sid INT PRIMARY KEY,
             sname VARCHAR(50),
             age INT
 10
 11
      mid VARCHAR(10) PRIMARY KEY,
 13
             mname VARCHAR(50),
 14
 15
             venue VARCHAR(50)
 16
 17
 18

    ○ CREATE TABLE Play (
 19
            sid INT,
            mid VARCHAR(10),
 20
 21
             day DATE,
 22
             PRIMARY KEY (sid, mid, day),
            FOREIGN KEY (sid) REFERENCES Student(sid),
 23
 24
             FOREIGN KEY (mid) REFERENCES `Match` (mid)
 25
 26
Output :
Action Output
                                                                                                                             Message

    1 22:25:14 CREATE DATABASE sports_management

                                                                                                                             1 row(s) affected
2 22:25:14 USE sports_management
    3 22:25:14 CREATE TABLE Student ( sid INT PRIMARY KEY, sname VARCHAR(50), age INT)
                                                                                                                            0 row(s) affected
4 22:25:14 CREATE TABLE 'Match' ( mid VARCHAR(10) PRIMARY KEY, mname VARCHAR(50), venue VARCHAR(50))
   5 22:25:14 CREATE TABLE Play ( sid INT, mid VARCHAR(10), day DATE, PRIMARY KEY (sid, mid, day), FOREIGN KEY (sid) REFERENCES Student(isi... 0 row(s) affected
```

2. Populate all the tables.

```
26
           Execute the selected portion of the script or everything, if there is no selection
27
          -- Populate the tables with sample data
28 •
         INSERT INTO Student (sid, sname, age) VALUES
29
         (1, 'Amit', 20),
         (2, 'Raj', 22),
30
         (3, 'Priya', 21),
31
         (4, 'Neha', 20);
33
         INSERT INTO `Match` (mid, mname, venue) VALUES
34 •
         ('B10', 'Cricket', 'Delhi'),
35
         ('B11', 'Football', 'Mumbai'),
37
         ('B12', 'Tennis', 'Delhi');
38
39 •
         INSERT INTO Play (sid, mid, day) VALUES
         (1, 'B10', '2024-10-10'),
40
41
         (2, 'B11', '2024-10-10'),
         (1, 'B11', '2024-10-11'),
42
         (3, 'B12', '2024-10-12'),
43
         (4, 'B10', '2024-10-13'),
45
         (1, 'B10', '2024-10-13');
46
         -- Query 1: Find all information of students who have played match number B10
47
48 •
       SELECT Student.*
49
         FROM Student
50
         JOIN Play ON Student.sid = Play.sid
otput ::
Action Output
                                                                                                                                            Message
   1 22:25:14 CREATE DATABASE sports_management
                                                                                                                                           1 row(s) affe
2 22:25:14 USE sports_management
                                                                                                                                           0 row(s) affe
  3 22:25:14 CREATE TABLE Student ( sid INT PRIMARY KEY, sname VARCHAR(50), age INT)
                                                                                                                                           0 row(s) affe
4 22:25:14 CREATE TABLE 'Match' ( mid VARCHAR(10) PRIMARY KEY, mname VARCHAR(50), venue VARCHAR(50))
                                                                                                                                           0 row(s) affe
) 5 22:25:14 CREATE TABLE Play ( sid INT, mid VARCHAR(10), day DATE, PRIMARY KEY (sid, mid, day), FOREIGN KEY (sid) REFERENCES Student(si... 0 row(s) affe
6 22:25:35 INSERT INTO Student (sid, sname, age) VALUES (1, 'Amit', 20), (2, 'Raj', 22), (3, 'Priya', 21), (4, 'Neha', 20)
7 22:25:35 INSERT INTO 'Match' (mid, mname, venue) VALUES ('B10', 'Cricket', 'Delhi'), ('B11', 'Football', 'Mumbai'), ('B12', 'Tennis', 'Delhi')
) 8 22:25:35 INSERT INTO Play (sid, mid, day) VALUES (1, 'B10', '2024-10-10'), (2, 'B11', '2024-10-10'), (1, 'B11', '2024-10-11'), (3, 'B12', '2024-10-12'), (4, 'B10', '2024-... 6 row(s) affe
```

3. Find all information of students who have played match number B10.



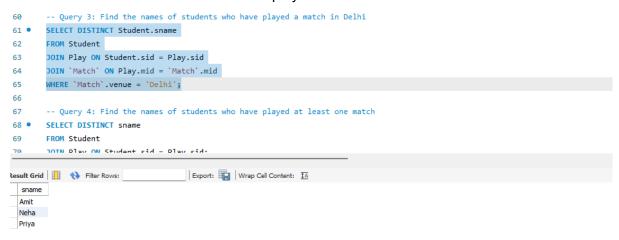


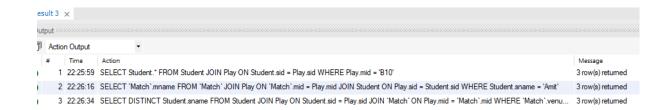
4. Find the name of matches played by Amit.

```
-- Query 2: Find the name of matches played by Amit
53
54 •
       SELECT `Match`.mname
       FROM 'Match'
55
       JOIN Play ON `Match`.mid = Play.mid
57
       JOIN Student ON Play.sid = Student.sid
       WHERE Student.sname = 'Amit';
58
59
       -- Query 3: Find the names of students who have played a match in Delhi
61 • SELECT DISTINCT Student.sname
       FROM Student
62
       JOIN Play ON Student.sid = Play.sid
       TOTM 'Match' ON Dlaw mid - 'Match' mid
                                       Export: Wrap Cell Content: 1A
tesult Grid H N Filter Rows:
  mname
 Cricket
 Cricket
 Football
```



5. Find the names of students who have played a match in Delhi.



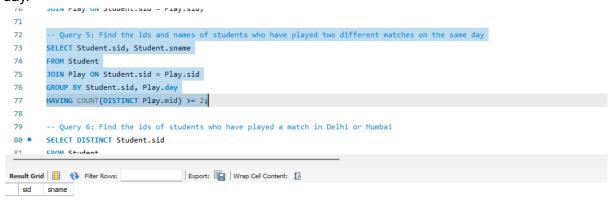


6. Find the names of students who have played at least one match.

```
67
        -- Query 4: Find the names of students who have played at least one match
       SELECT DISTINCT sname
68 •
        FROM Student
69
 70
        JOIN Play ON Student.sid = Play.sid;
 71
       -- Query 5: Find the ids and names of students who have played two different matches on the same day
 72
73
       SELECT Student.sid, Student.sname
       FROM Student
 75
       JOIN Play ON Student.sid = Play.sid
       CDOID RV Student eid Dlaw daw
                                   Export: Wrap Cell Content: 🔼
sname
 Amit
  Raj
  Priya
 Neha
```

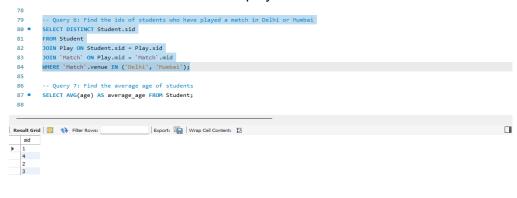


7. Find the ids and names of students who have played two different matches on the same day.



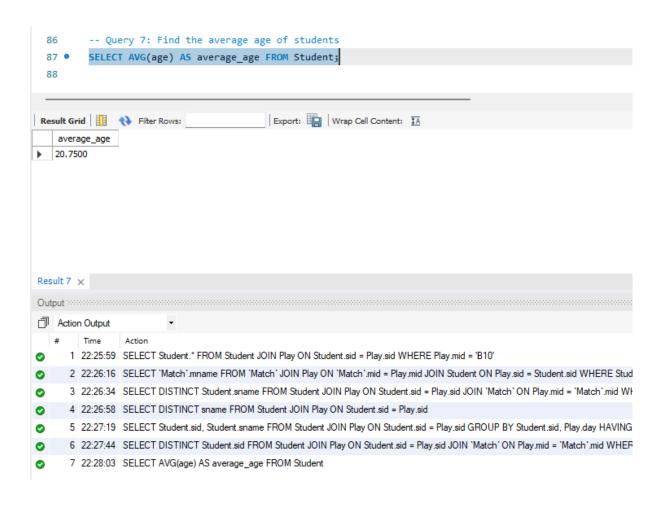


8. Find the ids of students who have played a match in Delhi or Mumbai.





9. Find the average age of students



EXPERIMENT - 9

Title: To understand the concepts of Views.

Objective: Students will be able to implement the concept of views.

1. Create table of table name: EMPLOYEES and add 6 rows

Column Name	Data Type	Width	Attributes
Employee_id	Character	10	PK
First_Name	Character	30	NN
Last_Name	Character	30	NN
DOB	Date		
Salary	Number	25	NN
Department_id	Character	10	

```
-- Creating the EMPLOYEES Table
CREATE TABLE EMPLOYEES (
  Employee_id CHAR(10) PRIMARY KEY,
  First_Name CHAR(30) NOT NULL,
  Last_Name CHAR(30) NOT NULL,
  DOB DATE,
  Salary DECIMAL(10, 2) NOT NULL, -- Using DECIMAL to handle salaries with two decimal places
  Department id CHAR(10)
· );
 -- Inserting values into the EMPLOYEES table
 INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES
 ('E001', 'John', 'Doe', '1990-01-15', 50000.00, 'D01'),
 ('E002', 'Jane', 'Smith', '1985-03-22', 60000.00, 'D02'),
 ('E003', 'Emily', 'Jones', '1992-06-30', 55000.00, 'D01'),
 ('E004', 'Michael', 'Brown', '1988-12-05', 70000.00, 'D03'),
 ('E005', 'Linda', 'Davis', '1995-05-15', 65000.00, 'D02'),
 ('E006', 'James', 'Wilson', '1980-09-25', 75000.00, 'D01');
```

- 2. Execute the following view related queries:
- 1) Create View of name emp_view and the column would be Employee_id, Last_Name, salary and department id only.
- 2) Insert values into view(remove the NOT NULL constraint and then insert values):
- 3) Modify, delete and drop operations are performed on view

```
-- Creating a View named emp_view
29 • CREATE VIEW emp_view AS
      SELECT Employee_id, Last_Name, Salary, Department_id
      FROM EMPLOYEES;
31
32
     -- You cannot directly insert into a view like this unless you are inserting into an updatable view that maps directly to a base table.
       -- Remove the insert into view since it will cause errors.
36
      -- If you need to modify the Salary column to allow NULL values, you'd do the following:
37 • ALTER TABLE EMPLOYEES MODIFY Salary DECIMAL(10, 2) NULL;
       -- Now, you can insert a row with a NULL salary
39
40 • INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES
      ('E007', 'Chris', 'Anderson', NULL, NULL, 'D01');
44
       -- Update operations on the View (affects the base table EMPLOYEES)
45 • UPDATE emp_view
      SET Salary = 80000.00
47
      WHERE Employee id = 'E001';
       -- Delete an employee from the view (and consequently from the EMPLOYEES table)
50 • DELETE FROM emp_view
      WHERE Employee_id = 'E003';
51
  53 •
             SELECT * FROM emp view;
```

```
53 • SELECT * FROM emp_view;
54 -- Dropping the emp_view
55 • DROP VIEW emp_view;
56
```

4) Create a view named salary_view. The view shows the employees in department 02 and their annual salary.

```
-- Create a View named salary_view to show annual salary for employees in Department D02

CREATE VIEW salary_view AS

SELECT Employee_id, Last_Name, Salary * 12 AS Annual_Salary

FROM EMPLOYEES

WHERE Department_id = 'D02';

-- View the salary_view

SELECT * FROM salary_view;

DROP database exp10;
```