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))

```sql
-- 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
);

CREATE TABLE `Match` (
    mid VARCHAR(10) PRIMARY KEY,
    mname VARCHAR(50),
    venue VARCHAR(50)
);

CREATE TABLE Play (
    sid INT,
    mid VARCHAR(10),
    day DATE,
    PRIMARY KEY (sid, mid, day),
    FOREIGN KEY (sid) REFERENCES Student(sid),
    FOREIGN KEY (mid) REFERENCES `Match`(mid)
);

```

Output

Action Output

#	Time	Action	Message
1	22:25:14	CREATE DATABASE sports_management	1 row(s) affected
2	22:25:14	USE sports_management	0 row(s) affected
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))	0 row(s) affected
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) affected

2. Populate all the tables.

```
26
27  -- Populate the tables with sample data
28  INSERT INTO Student (sid, sname, age) VALUES
29  (1, 'Amit', 20),
30  (2, 'Raj', 22),
31  (3, 'Priya', 21),
32  (4, 'Neha', 20);
33
34  INSERT INTO `Match` (mid, mname, venue) VALUES
35  ('B10', 'Cricket', 'Delhi'),
36  ('B11', 'Football', 'Mumbai'),
37  ('B12', 'Tennis', 'Delhi');
38
39  INSERT INTO Play (sid, mid, day) VALUES
40  (1, 'B10', '2024-10-10'),
41  (2, 'B11', '2024-10-10'),
42  (1, 'B11', '2024-10-11'),
43  (3, 'B12', '2024-10-12'),
44  (4, 'B10', '2024-10-13'),
45  (1, 'B10', '2024-10-13');
46
47  -- Query 1: Find all information of students who have played match number B10
48  SELECT Student.*
49  FROM Student
50  JOIN Play ON Student.sid = Play.sid
```

Execute the selected portion of the script or everything, if there is no selection

Output

Action Output

#	Time	Action	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)	4 row(s) affe
7	22:25:35	INSERT INTO `Match` (mid, mname, venue) VALUES ('B10', 'Cricket', 'Delhi'), ('B11', 'Football', 'Mumbai'), ('B12', 'Tennis', 'Delhi')	3 row(s) affe
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.

```
46
47      -- Query 1: Find all information of students who have played match number B10
48      SELECT Student.*
49      FROM Student
50      JOIN Play ON Student.sid = Play.sid
51      WHERE Play.mid = 'B10';
52
53      -- Query 2: Find the name of matches played by Amit
54      SELECT `Match`.mname
55      FROM `Match`
```

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

sid	sname	age
1	Amit	20
1	Amit	20
4	Neha	20

Result 1

Output

Action Output

#	Time	Action
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'

4. Find the name of matches played by Amit.

```
52
53      -- Query 2: Find the name of matches played by Amit
54      SELECT `Match`.mname
55      FROM `Match`
56      JOIN Play ON `Match`.mid = Play.mid
57      JOIN Student ON Play.sid = Student.sid
58      WHERE Student.sname = 'Amit';
59
60      -- Query 3: Find the names of students who have played a match in Delhi
61      SELECT DISTINCT Student.sname
62      FROM Student
63      JOIN Play ON Student.sid = Play.sid
64      JOIN `Match` ON Play.mid = `Match`.mid
```

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

mname
Cricket
Cricket
Football

esult 2 ×

utput

Action Output

#	Time	Action
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'
2	22:26:16	SELECT `Match`.mname FROM `Match` JOIN Play ON `Match`.mid = Play.mid JOIN Student ON Play.sid = Student.sid WHERE Student.snam

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

```
60      -- Query 3: Find the names of students who have played a match in Delhi
61      SELECT DISTINCT Student.sname
62      FROM Student
63      JOIN Play ON Student.sid = Play.sid
64      JOIN `Match` ON Play.mid = `Match`.mid
65      WHERE `Match`.venue = 'Delhi';
66
67      -- Query 4: Find the names of students who have played at least one match
68      SELECT DISTINCT sname
69      FROM Student
70      JOIN Play ON Student.sid = Play.sid;
```

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

sname
Amit
Neha
Priya

esult 3

lutput

Action Output

#	Time	Action	Message
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'	3 row(s) returned
2	22:26:16	SELECT `Match`.mname FROM `Match` JOIN Play ON `Match`.mid = Play.mid JOIN Student ON Play.sid = Student.sid WHERE Student.sname = 'Amit'	3 row(s) returned
3	22:26:34	SELECT DISTINCT Student.sname FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WHERE `Match`.venu...	3 row(s) returned

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
68      SELECT DISTINCT sname
69      FROM Student
70      JOIN Play ON Student.sid = Play.sid;
71
72      -- Query 5: Find the ids and names of students who have played two different matches on the same day
73      SELECT Student.sid, Student.sname
74      FROM Student
75      JOIN Play ON Student.sid = Play.sid
```

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

sname
Amit
Raj
Priya
Neha

Result 4

Output

Action Output

#	Time	Action	Message
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'	3 row(s) returned
2	22:26:16	SELECT `Match`.mname FROM `Match` JOIN Play ON `Match`.mid = Play.mid JOIN Student ON Play.sid = Student.sid WHERE Student.sname = 'Amit'	3 row(s) returned
3	22:26:34	SELECT DISTINCT Student.sname FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WHERE `Match`.venu...	3 row(s) returned
4	22:26:58	SELECT DISTINCT sname FROM Student JOIN Play ON Student.sid = Play.sid	4 row(s) returned

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

```
71
72  -- Query 5: Find the ids and names of students who have played two different matches on the same day
73  SELECT Student.sid, Student.sname
74  FROM Student
75  JOIN Play ON Student.sid = Play.sid
76  GROUP BY Student.sid, Play.day
77  HAVING COUNT(DISTINCT Play.mid) >= 2;
78
79  -- Query 6: Find the ids of students who have played a match in Delhi or Mumbai
80  SELECT DISTINCT Student.sid
```

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

sid	sname

Result 5

Output

Action Output

#	Time	Action	Message
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'	3 row(s) returned
2	22:26:16	SELECT `Match`.mname FROM `Match` JOIN Play ON `Match`.mid = Play.mid JOIN Student ON Play.sid = Student.sid WHERE Student.sname = 'Amit'	3 row(s) returned
3	22:26:34	SELECT DISTINCT Student.sname FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WHERE `Match`.venu...	3 row(s) returned
4	22:26:58	SELECT DISTINCT sname FROM Student JOIN Play ON Student.sid = Play.sid	4 row(s) returned
5	22:27:19	SELECT Student.sid, Student.sname FROM Student JOIN Play ON Student.sid = Play.sid GROUP BY Student.sid, Play.day HAVING COUNT(DISTINCT ...	0 row(s) returned

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

```
78
79  -- Query 6: Find the ids of students who have played a match in Delhi or Mumbai
80  SELECT DISTINCT Student.sid
81  FROM Student
82  JOIN Play ON Student.sid = Play.sid
83  JOIN `Match` ON Play.mid = `Match`.mid
84  WHERE `Match`.venue IN ('Delhi', 'Mumbai');
85
86  -- Query 7: Find the average age of students
87  SELECT AVG(age) AS average_age FROM Student;
88
```

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

sid
1
4
2
3

Result 6

Output

Action Output

#	Time	Action	Message
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'	3 row(s) returned
2	22:26:16	SELECT `Match`.mname FROM `Match` JOIN Play ON `Match`.mid = Play.mid JOIN Student ON Play.sid = Student.sid WHERE Student.sname = 'Amit'	3 row(s) returned
3	22:26:34	SELECT DISTINCT Student.sname FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WHERE `Match`.venu...	3 row(s) returned
4	22:26:58	SELECT DISTINCT sname FROM Student JOIN Play ON Student.sid = Play.sid	4 row(s) returned
5	22:27:19	SELECT Student.sid, Student.sname FROM Student JOIN Play ON Student.sid = Play.sid GROUP BY Student.sid, Play.day HAVING COUNT(DISTINCT ...	0 row(s) returned
6	22:27:44	SELECT DISTINCT Student.sid FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WHERE `Match`.venue IN...	4 row(s) returned

9. Find the average age of students

```
86      -- Query 7: Find the average age of students
87 •    SELECT AVG(age) AS average_age FROM Student;
88
```

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

average_age
20.7500

Result 7

Output

Action Output

#	Time	Action
1	22:25:59	SELECT Student.* FROM Student JOIN Play ON Student.sid = Play.sid WHERE Play.mid = 'B10'
2	22:26:16	SELECT `Match`.mname FROM `Match` JOIN Play ON `Match`.mid = Play.mid JOIN Student ON Play.sid = Student.sid WHERE Stud
3	22:26:34	SELECT DISTINCT Student.sname FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WH
4	22:26:58	SELECT DISTINCT sname FROM Student JOIN Play ON Student.sid = Play.sid
5	22:27:19	SELECT Student.sid, Student.sname FROM Student JOIN Play ON Student.sid = Play.sid GROUP BY Student.sid, Play.day HAVING
6	22:27:44	SELECT DISTINCT Student.sid FROM Student JOIN Play ON Student.sid = Play.sid JOIN `Match` ON Play.mid = `Match`.mid WHER
7	22:28:03	SELECT AVG(age) AS average_age FROM Student

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	

```sql
-- 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

```sql
-- Creating a View named emp_view
CREATE VIEW emp_view AS
SELECT Employee_id, Last_Name, Salary, Department_id
FROM EMPLOYEES;

-- 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.

-- If you need to modify the Salary column to allow NULL values, you'd do the following:
ALTER TABLE EMPLOYEES MODIFY Salary DECIMAL(10, 2) NULL;

-- Now, you can insert a row with a NULL salary
INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES
('E007', 'Chris', 'Anderson', NULL, NULL, 'D01');


-- Update operations on the View (affects the base table EMPLOYEES)
UPDATE emp_view
SET Salary = 80000.00
WHERE Employee_id = 'E001';

-- Delete an employee from the view (and consequently from the EMPLOYEES table)
DELETE FROM emp_view
WHERE Employee_id = 'E003';

```

.

```sql
SELECT * FROM emp_view;
-- Dropping the emp_view
DROP VIEW emp_view;

```

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

```
57      -- Create a View named salary_view to show annual salary for employees in Department D02
58  •   CREATE VIEW salary_view AS
59      SELECT Employee_id, Last_Name, Salary * 12 AS Annual_Salary
60      FROM EMPLOYEES
61      WHERE Department_id = 'D02';
62
63      -- View the salary_view
64  •   SELECT * FROM salary_view;
65  •    DROP database exp10;
```