

Database Management Systems Laboratory
Assignment 2: Database Design – SQL

Prasanna Paithankar | 21CS30065

PostgreSQL DB for University Festival Management System

Tables:

1. Event(EID, EName, Date, Type)
2. College(Name, Location)
3. Participant(PID, Name, CName)
 - CName creates one-to-one relation to College entity
4. Role(RID, Rname, Description)
5. Student(Roll, Name, Dept, RID)
 - RID creates many-to-one relation to Role entity
6. Student_Event(Roll, EID)
 - Many-to-many relationship table for Student MANAGES Event as each event is managed by multiple students and one student can be incharge of multiple events.
7. Event_Participant(EID, PID)
 - Many-to-many relationship table for Event HAS Participant as each event consists of multiple participants and one participant can register for multiple events.
8. Volunteer(Roll, EID)
 - Many-to-one relationship table for Event HAS Volunteer as each event is managed by multiple volunteers.

SQL Commands to create the above:

```
CREATE TABLE Event (  
    EID INTEGER PRIMARY KEY,  
    EName VARCHAR(255) NOT NULL,  
    Date DATE NOT NULL,  
    Type VARCHAR(255)  
);  
  
-- Create College table  
CREATE TABLE College (  
    Name VARCHAR(255) PRIMARY KEY,  
    Location VARCHAR(255) NOT NULL  
);  
  
-- Create Participant table  
CREATE TABLE Participant (  
    PID INTEGER PRIMARY KEY,  
    Name VARCHAR(255) NOT NULL,  
    CName VARCHAR(255),  
    FOREIGN KEY (CName) REFERENCES College(Name)  
);
```

```

-- Create Role table
CREATE TABLE Role (
    RID INTEGER PRIMARY KEY,
    Rname VARCHAR(255) NOT NULL,
    Description VARCHAR(255) NOT NULL
);

-- Create Student table
CREATE TABLE Student (
    Roll INTEGER PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Dept VARCHAR(255) NOT NULL,
    RID INTEGER,
    FOREIGN KEY (RID) REFERENCES Role(RID)
);

-- Create Volunteer table
CREATE TABLE Student_Event (
    Roll INTEGER,
    EID INTEGER,
    PRIMARY KEY (Roll, EID),
    FOREIGN KEY (Roll) REFERENCES Student(Roll),
    FOREIGN KEY (EID) REFERENCES Event(EID)
);

-- Create Event_Participant Relation table
CREATE TABLE Event_Participant (
    EID INTEGER,
    PID INTEGER,
    PRIMARY KEY (EID, PID),
    FOREIGN KEY (EID) REFERENCES Event(EID),
    FOREIGN KEY (PID) REFERENCES Participant(PID)
);

-- Create Volunteer table
CREATE TABLE Volunteer (
    Roll INTEGER PRIMARY KEY,
    EID INTEGER,
    FOREIGN KEY (Roll) REFERENCES Student(Roll),
    FOREIGN KEY (EID) REFERENCES Event(EID)
);

```

To populate data into the table we execute the following SQL commands in psql:

```

INSERT INTO Event
VALUES
(1, 'SmallEvent', '2024-01-01', 'Guest Lecture'),
(2, 'MediumEvent', '2024-01-02', 'Hackathon'),
(3, 'LargeEvent', '2024-01-03', 'Competition'),
(4, 'Megaevent', '2024-01-04', 'Hackathon'),
(5, 'Gigaevent', '2024-01-05', 'Competition'),

```

```
(6, 'Teraevent', '2024-01-06', 'Guest Lecture'),
(7, 'Petaevent', '2024-01-07', 'Hackathon'),
(8, 'Exaevent', '2024-01-08', 'Competition'),
(9, 'Zettaevent', '2024-01-09', 'Guest Lecture'),
(10, 'Yottaevent', '2024-01-10', 'Guest Lecture');
```

```
INSERT INTO College
VALUES
```

```
('IITKGP', 'Kharagpur'),
('IITB', 'Bombay'),
('IITM', 'Madras'),
('IITD', 'Delhi'),
('IITK', 'Kanpur'),
('IITG', 'Guwahati'),
('IITH', 'Hyderabad'),
('IITI', 'Indore'),
('IITR', 'Roorkee'),
('IITJ', 'Jodhpur');
```

```
INSERT INTO Participant
VALUES
```

```
(1, 'Siddharth', 'IITB'),
(2, 'Ram', 'IITM'),
(3, 'Prasanna', 'IITKGP'),
(4, 'Sam', 'IITD'),
(5, 'Raj', 'IITK'),
(6, 'Rahul', 'IITG'),
(7, 'Rohan', 'IITH'),
(8, 'Rajesh', 'IITI'),
(9, 'Rakesh', 'IITR'),
(10, 'Ramesh', 'IITJ'),
(11, 'Rajat', 'IITB'),
(12, 'Prabhu', 'IITM'),
(13, 'Pranav', 'IITKGP'),
(14, 'Samarth', 'IITD'),
(15, 'Sahil', 'IITK'),
(16, 'Alex', 'IITG'),
(17, 'Bob', 'IITB'),
(18, 'Charlie', 'IITI'),
(19, 'David', 'IITB'),
(20, 'Ethan', 'IITJ'),
(21, 'Felix', 'IITB'),
(22, 'Gabe', 'IITM'),
(23, 'Hannah', 'IITKGP'),
(24, 'Ivan', 'IITD'),
(25, 'Jack', 'IITKGP'),
(26, 'Karl', 'IITG'),
(27, 'Liam', 'IITB'),
(28, 'Mason', 'IITI'),
(29, 'Noah', 'IITR'),
(30, 'Oliver', 'IITJ');
```

```
INSERT INTO Role
```

```
VALUES
```

```
(1, 'Manager', 'Manages the event'),  
(2, 'Secretary', 'Manages the event'),  
(3, 'Designer', 'Designs the event'),  
(4, 'Logistics', 'Manages the logistics'),  
(5, 'Sponsor', 'Sponsorship of the event'),  
(6, 'Organizer', 'Organizes the event');
```

```
INSERT INTO Student
```

```
VALUES
```

```
(1, 'Ramunujan', 'ME', 2),  
(2, 'Prasanna', 'CSE', 1),  
(3, 'Sam', 'EE', 3),  
(4, 'Raj', 'ME', 4),  
(5, 'Rahul', 'CSE', 5),  
(6, 'Rohan', 'CSE', 6),  
(7, 'Rajesh', 'ME', 1),  
(8, 'Rakesh', 'CSE', 2),  
(9, 'Ramesh', 'EE', 3),  
(10, 'Rajat', 'ME', 4),  
(11, 'Prabhu', 'CSE', 5),  
(12, 'Pranav', 'EE', 6),  
(13, 'Samarth', 'ME', 1),  
(14, 'Sahil', 'CSE', 2),  
(15, 'Alex', 'EE', 3);
```

```
INSERT INTO Volunteer
```

```
VALUES
```

```
(1, 1),  
(2, 4),  
(3, 3),  
(4, 5),  
(5, 4),  
(6, 6),  
(7, 7),  
(8, 8);
```

```
INSERT INTO Student_Event
```

```
VALUES
```

```
(1, 1),  
(2, 4),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10),
```

```

(11, 1),
(12, 2),
(13, 3),
(14, 4),
(15, 5);

INSERT INTO Event_Participant
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(4, 11),
(2, 12),
(3, 13),
(4, 14),
(5, 15),
(6, 16),
(4, 17),
(8, 18),
(9, 19),
(10, 20),
(1, 21),
(2, 22),
(3, 23),
(4, 24),
(5, 25),
(6, 26),
(7, 27),
(8, 28),
(9, 29),
(10, 30);

```

We now execute the following queries:

(i) Roll number and name of all the students who are managing the “Megaevent”

```

SELECT S.Roll, S.Name
FROM Student S, Student_Event SE, Event E
WHERE S.Roll = SE.Roll AND SE.EID = E.EID AND E.ENAME = 'Megaevent';

```

Output:

roll	name
2	Prasanna
4	Raj
14	Sahil

(3 rows)

(ii) Roll number and name of all the students who are managing the “Megaevent” as a “Secretary”

```
SELECT S.Roll, S.Name
FROM Student S, Student_Event SE, Event E, Role R
WHERE S.Roll = SE.Roll AND SE.EID = E.EID AND E.ENAME = 'Megaevent' AND S.RID =
R.RID AND R.Rname = 'Secretary';
```

Output:

roll	name
14	Sahil

(1 row)

(iii) Name of all the participants from the college “IITB” in “Megaevent”

```
SELECT P.Name
FROM Participant P, Event_Participant VEP, Event E
WHERE P.PID = VEP.PID AND VEP.EID = E.EID AND E.ENAME = 'Megaevent' AND P.CName =
'IITB';
```

Output:

name
Rajat
Bob

(2 rows)

(iv) Name of all the colleges who have at least one participant in “Megaevent”

```
SELECT DISTINCT P.CName
FROM Participant P, Event_Participant VEP, Event E
WHERE P.PID = VEP.PID AND VEP.EID = E.EID AND E.ENAME = 'Megaevent';
```

Output:

cname
IITB
IITD

(2 rows)

(v) Name of all the events which is managed by a “Secretary”

```
SELECT E.ENAME
FROM Event E, Student S, Student_Event SE, Role R
WHERE E.EID = SE.EID AND SE.Roll = S.Roll AND S.RID = R.RID AND R.Rname =
'Secretary';
```

Output:

```
      ename
-----
SmallEvent
Exaevent
Megaevent
(3 rows)
```

(vi) Name of all the “CSE” department student volunteers of “Megaevent”

```
SELECT S.Name
FROM Student S, Volunteer V, Event E, Student_Event SE
WHERE S.Roll = V.Roll AND V.EID = E.EID AND E.ENAME = 'Megaevent' AND S.Dept =
'CSE' AND S.Roll = SE.Roll AND SE.EID = E.EID;
```

Output:

```
      name
-----
Prasanna
(1 row)
```

(vii) Name of all the events which has at least one volunteer from “CSE”

```
SELECT E.ENAME
FROM Event E, Student S, Volunteer V
WHERE E.EID = V.EID AND V.Roll = S.Roll AND S.Dept = 'CSE';
```

Output:

```
      ename
-----
Megaevent
Megaevent
Teraevent
Exaevent
(4 rows)
```

(viii) Name of the college with the largest number of participants in “Megaevent”

```
SELECT P.CName
FROM Participant P
JOIN Event_Participant VEP ON P.PID = VEP.PID
JOIN Event E ON VEP.EID = E.EID
WHERE E.ENAME = 'Megaevent'
GROUP BY P.CName
HAVING COUNT(*) = (
    SELECT MAX(cnt)
    FROM (
        SELECT COUNT(*) AS cnt
        FROM Participant P1
        JOIN Event_Participant VEP1 ON P1.PID = VEP1.PID
```

```

        JOIN Event E1 ON VEP1.EID = E1.EID
        WHERE E1.ENAME = 'Megaevent'
        GROUP BY P1.CName
    ) AS subquery
);

```

Output:

```

cname
-----
IITD
(1 row)

```

(ix) Name of the college with largest number of participants overall

```

SELECT P.CName
FROM Participant P
JOIN Event_Participant VEP ON P.PID = VEP.PID
GROUP BY P.CName
HAVING COUNT(*) = (
    SELECT MAX(cnt)
    FROM (
        SELECT COUNT(*) AS cnt
        FROM Participant P1
        JOIN Event_Participant VEP1 ON P1.PID = VEP1.PID
        GROUP BY P1.CName
    ) AS subquery
);

```

Output:

```

cname
-----
IITB
(1 row)

```

(x) Name of the department with the largest number of volunteers in all the events which has at least one participant from "IITB"

```

WITH DeptCounts AS (
    SELECT S.Dept, COUNT(*) AS deptCount
    FROM Student S
    JOIN Volunteer V ON S.Roll = V.Roll
    JOIN Student_Event SE ON S.Roll = SE.Roll
    JOIN Event E ON SE.EID = E.EID
    JOIN Event_Participant VEP ON E.EID = VEP.EID
    JOIN Participant P ON VEP.PID = P.PID
    WHERE P.CName = 'IITB'
    GROUP BY S.Dept
)
SELECT Dept
FROM DeptCounts
WHERE deptCount = (SELECT MAX(deptCount) FROM DeptCounts);

```


Output:

```
dept
-----
ME
(1 row)
```

Notation: Standard notational constructs, as in the course slides
