NAME SWADHIN SATYAPRAKASH MAJHI
ROLL NO:-21CS10067
DBMS ASSIGNMENT 2

TABLE CREATION

TABLE CREATION TOTAL(6 tables + 1 for many to many relation)

```
-- STUDENT Table: Stores information about students, including their role
and the event they are associated with.
CREATE TABLE STUDENT (
    ROLL CHAR(9) PRIMARY KEY,          -- Unique roll number for each
student
    NAME VARCHAR(50) NOT NULL,         -- Name of the student
    DEPT VARCHAR(25) NOT NULL,         -- Department of the student
    RID INTEGER,                       -- Role ID, foreign key to ROLE table
    EID INTEGER,                       -- Event ID, foreign key to EVENT
table
    FOREIGN KEY (RID) REFERENCES ROLE(RID),
    FOREIGN KEY (EID) REFERENCES EVENT(EID)
);

-- ROLE Table: Defines various roles that students can have.
CREATE TABLE ROLE (
    RID INTEGER PRIMARY KEY,           -- Unique role ID
    RNAME VARCHAR(25),                 -- Name of the role
    RDESC TEXT                         -- Description of the role
);

-- EVENT Table: Details of different events.
CREATE TABLE EVENT (
    EID INTEGER PRIMARY KEY,           -- Unique event ID
    DATE DATE,                         -- Date of the event
    ENAME VARCHAR(20),                 -- Name of the event
    TYPE VARCHAR(20)                   -- Type of the event
);

-- COLLEGE Table: Information about colleges, ensuring each name-location
pair is unique.
CREATE TABLE COLLEGE (
    CID INTEGER PRIMARY KEY,           -- Unique college ID
    NAME VARCHAR(50),                  -- Name of the college
    LOCATION TEXT,                     -- Location of the college
    UNIQUE (NAME, LOCATION)            -- Ensuring uniqueness for
```

```
name-location pair
);

-- PARTICIPANT Table: Details of participants in events, linked to their
colleges and events.
CREATE TABLE PARTICIPANT (
    PID INTEGER PRIMARY KEY,           -- Unique participant ID
    PNAME VARCHAR(50) NOT NULL,        -- Name of the participant
    CID INTEGER,                       -- College ID, foreign key to COLLEGE
table
    EID INTEGER,                       -- Event ID, foreign key to EVENT
table
    FOREIGN KEY (CID) REFERENCES COLLEGE(CID),
    FOREIGN KEY (EID) REFERENCES EVENT(EID)
);

-- VOLUNTEER Table: Information about student volunteers, linking them to
specific events and participants.
CREATE TABLE VOLUNTEER (
    ROLL CHAR(10) PRIMARY KEY,         -- Roll number of the volunteer
(student)
    EID INTEGER,                       -- Event ID, foreign key to EVENT
table
    PID INTEGER,                       -- Participant ID, foreign key to
PARTICIPANT table
    FOREIGN KEY (EID) REFERENCES EVENT(EID),
    FOREIGN KEY (PID) REFERENCES PARTICIPANT(PID)
);

-- HAS_EVENT_PARTICIPANT Table: Many-to-many relationship table between
events and participants.
CREATE TABLE HAS_EVENT_PARTICIPANT (
    EID INTEGER,                       -- Event ID, foreign key to EVENT
table
    PID INTEGER,                       -- Participant ID, foreign key to
PARTICIPANT table
    FOREIGN KEY (EID) REFERENCES EVENT(EID),
    FOREIGN KEY (PID) REFERENCES PARTICIPANT(PID)
);
```

INSERT DATA INTO TABLE

```
INSERT INTO EVENT (EID, DATE, ENAME, TYPE)
VALUES (1, '2024-01-10', 'Megaevent', 'Conference'),
    (2, '2024-02-15', 'TechTalk', 'Seminar'),
    (3, '2024-03-20', 'Workshop2024', 'Workshop'),
```

```
    (4, '2024-04-25', 'CulturalFest', 'Festival'),
    (5, '2024-05-30', 'SportsMeet', 'Sports');
INSERT INTO COLLEGE (CID, NAME, LOCATION)
VALUES (1, 'IITB', 'Mumbai'),
    (2, 'IITD', 'Delhi'),
    (3, 'IITK', 'Kanpur'),
    (4, 'IITM', 'Madras'),
    (5, 'IISC', 'Bangalore');
INSERT INTO ROLE (RID, RNAME, RDESC)
VALUES (1, 'Secretary', 'Manages administrative tasks'),
    (2, 'Coordinator', 'Coordinates events'),
    (3, 'Treasurer', 'Manages finances'),
    (4, 'Volunteer', 'Assists in various tasks'),
    (5, 'Speaker', 'Delivers talks or presentations');

INSERT INTO STUDENT (ROLL, NAME, DEPT, RID, EID)
VALUES ('001CSE', 'John Doe', 'CSE', 1, 1),
    ('002ECE', 'Alice Smith', 'ECE', 2, 1),
    ('003ME', 'Bob Johnson', 'ME', 4, 2),
    ('004EE', 'Emma Brown', 'EE', 3, 3),
    ('005CS', 'David Wilson', 'CS', 2, 4);

INSERT INTO PARTICIPANT (PID, PNAME, CID, EID)
VALUES (1, 'Rachel Green', 1, 1),
    (2, 'Monica Geller', 2, 2),
    (3, 'Phoebe Buffay', 3, 1),
    (4, 'Ross Geller', 4, 3),
    (5, 'Chandler Bing', 5, 1);

INSERT INTO VOLUNTEER (ROLL, EID, PID)
VALUES ('001CSE', 1, 1),
    ('002ECE', 1, 2),
    ('003ME', 2, 3),
    ('004EE', 3, 4),
    ('005CS', 1, 5);
INSERT INTO HAS_EVENT_PARTICIPANT (EID, PID)
VALUES (1, 1),
    (1, 2),
    (2, 3),
    (3, 4),
    (1, 5);
```

QUERY SEARCH

```
--Certainly! Let's structure the requirements into a more organized format.
I'll define the relational schema, including table definitions, attribute
definitions, and attribute data types. Then, I'll create SQL commands for
```

table creation and insertion of sample records. Finally, I'll provide examples of SQL queries for the specified conditions.

--4. SQL Queries:

--(i) Roll number and name of all the students who are managing the "Megaevent":

```
SELECT S.ROLL,
    S.NAME
FROM STUDENT S
    JOIN EVENT E ON S.EID = E.EID
WHERE E.ENAME = 'Megaevent';
```

OUTPUT:-

```
   roll    |    name
-----------+-------------
 001CSE    | John Doe
 002ECE    | Alice Smith
```

--(ii) Roll number and name of all the students who are managing "Megaevent" as a "Secretary":

```
SELECT S.ROLL,
    S.NAME
FROM STUDENT S
    JOIN ROLE R ON S.RID = R.RID
    JOIN EVENT E ON S.EID = E.EID
WHERE R.RNAME = 'Secretary'
    AND E.ENAME = 'Megaevent';
```

OUTPUT:-

```
   roll    |   name
-----------+----------
 001CSE    | John Doe
```

--(iii) Name of all the participants from the college "IITB" in "Megaevent":

```
SELECT P.PNAME
FROM PARTICIPANT P
    JOIN COLLEGE C ON P.CID = C.CID
    JOIN EVENT E ON P.EID = E.EID
WHERE C.NAME = 'IITB'
    AND E.ENAME = 'Megaevent';
```

```
OUTPUT:-
    pname
--------------
 Rachel Green


--(iv) Name of all the colleges who have at least one participant in
"Megaevent":
SELECT DISTINCT C.NAME
FROM COLLEGE C
    JOIN PARTICIPANT P ON C.CID = P.CID
    JOIN EVENT E ON P.EID = E.EID
WHERE E.ENAME = 'Megaevent';

OUTPUT:-
 name
------
 IISC
 IITB
 IITK

--(v) Name of all the events which are managed by a "Secretary":
SELECT DISTINCT E.ENAME
FROM EVENT E
    JOIN STUDENT S ON E.EID = S.EID
    JOIN ROLE R ON S.RID = R.RID
WHERE R.RNAME = 'Secretary';

OUTPUT:-
   ename
-----------
 Megaevent
(1 row)


--(vi) Name of all the "CSE" department student volunteers of "Megaevent":
SELECT S.NAME
FROM STUDENT S
    JOIN VOLUNTEER V ON S.ROLL = V.ROLL
    JOIN EVENT E ON V.EID = E.EID
WHERE S.DEPT = 'CSE'
    AND E.ENAME = 'Megaevent';

OUTPUT:-
   name
----------
 John Doe
```

```sql
--(vii) Name of all the events which have at least one volunteer from "CSE":
SELECT DISTINCT E.ENAME
FROM EVENT E
    JOIN VOLUNTEER V ON E.EID = V.EID
    JOIN STUDENT S ON V.ROLL = S.ROLL
WHERE S.DEPT = 'CSE';
```

```
OUTPUT:-
   ename
-----------
 Megaevent
(1 row)
```

```sql
--(viii) Name of the college with the largest number of participants in
"Megaevent":
SELECT C.NAME
FROM COLLEGE C
    JOIN PARTICIPANT P ON C.CID = P.CID
    JOIN EVENT E ON P.EID = E.EID
WHERE E.ENAME = 'Megaevent'
GROUP BY C.NAME
ORDER BY COUNT(*) DESC
LIMIT 1;
```

```
OUTPUT:-
 name
------
 IISC
```

```sql
--(ix) Name of the college with the largest number of participants overall:
SELECT C.NAME
FROM COLLEGE C
    JOIN PARTICIPANT P ON C.CID = P.CID
GROUP BY C.NAME
ORDER BY COUNT(*) DESC
LIMIT 1;
```

```
OUTPUT:-
 name
------
 IITD
```

```sql
--(x) Name of the department with the largest number of volunteers in all
the events which have at least one participant from "IITB":
SELECT S.DEPT
FROM STUDENT S
    JOIN VOLUNTEER V ON S.ROLL = V.ROLL
    JOIN EVENT E ON V.EID = E.EID
    JOIN PARTICIPANT P ON E.EID = P.EID
    JOIN COLLEGE C ON P.CID = C.CID
WHERE C.NAME = 'IITB'
GROUP BY S.DEPT
ORDER BY COUNT(DISTINCT V.ROLL) DESC
LIMIT 1;
```

OUTPUT:-

```
 dept
------
 CS
(1 row)
```