Report

Schema

Schemas	Foreign Keys
college(name, location)	-
participant(pid, name)	-
volunteer(roll)	-
event(eid, date, ename, type)	-
student(roll, name, dept)	-
role(rid, rname, description)	-
student_from(college_name, pid)	college_name -> college_name, pid -> participant_pid
participant_has(pid, eid)	pid -> participant_pid, eid -> event_eid
volunteer_has(roll, eid)	roll -> volunteer_roll, eid -> event_eid
manage(roll, eid)	roll -> student_roll, eid -> event_eid
student_has(roll, rid)	roll -> student_roll, rid -> role_rid

Underlined attributes are Primary Keys.

Table & Attribute Definitions

- college (
  name varchar(511) primary key,
  location varchar(2047) not null
)
- participant (
  pid varchar(20) primary key,
  name varchar(255) not null
)
- volunteer (
  roll varchar(20) primary key
)
- event (
  eid varchar(20) primary key,
  date DATE not null,
  ename varchar(255) not null,
  type varchar(255)
)
- student (
  roll varchar(20) primary key,
  name varchar(255) not null,
  dept varchar(255) not null
)

- role (
  rid varchar(20) primary key,
  rname varchar(255) not null,
  description varchar(255)
)

- student_from (
  college_name varchar(255),
  pid varchar(20),
  foreign key (college_name) references college(name),
  foreign key (pid) references participant(pid),
  primary key (college_name, pid)
)

- participant_has (
  pid varchar(20),
  eid varchar(20),
  foreign key (pid) references participant(pid),
  foreign key (eid) references event(eid),
  primary key (pid, eid)
)

- volunteer_has (
  roll varchar(20),
  eid varchar(20),
  foreign key (roll) references volunteer(roll),
  foreign key (eid) references event(eid),
  primary key (roll, eid)
)

- manage (
  roll varchar(20),
  eid varchar(20),
  foreign key (roll) references student(roll),
  foreign key (eid) references event(eid),
  primary key (roll, eid)
)

- student_has (
  roll varchar(20),
  rid varchar(20),
  foreign key (roll) references student(roll),
  foreign key (rid) references role(rid),
  primary key (roll, rid)
)

Primary keys are implicitly not null.

Commands

Create Tables & Insert Data

-
  ```
  – Create college table
  CREATE TABLE college (
  name VARCHAR(511) PRIMARY KEY,
  location VARCHAR(2047) NOT NULL
  );
  ```

  ```
  – Insert sample data
  INSERT INTO college (name, location) VALUES
  (’IITB’, ’Mumbai’),
  (’MIT’, ’Cambridge’),
  (’Stanford’, ’Palo Alto’),
  (’Harvard’, ’Cambridge’),
  (’Caltech’, ’Pasadena’);
  ```

-
  ```
  – Create participant table
  CREATE TABLE participant (
  pid VARCHAR(20) PRIMARY KEY,
  name VARCHAR(255) NOT NULL
  );
  ```

  ```
  – Insert sample data
  INSERT INTO participant (pid, name) VALUES
  (’P001’, ’John Doe’),
  (’P002’, ’Jane Smith’),
  (’P003’, ’Michael Johnson’),
  (’P004’, ’Emily Brown’),
  (’P005’, ’Robert Davis’);
  ```

-
  ```
  – Create volunteer table
  CREATE TABLE volunteer (
  roll VARCHAR(20) PRIMARY KEY
  );
  ```

  ```
  – Insert sample data
  INSERT INTO volunteer (roll) VALUES
  (’V001’),
  (’V002’),
  (’V003’),
  (’V004’),
  (’V005’);
  ```

-
  ```
  – Create event table
  CREATE TABLE event (
  eid VARCHAR(20) PRIMARY KEY,
  date DATE NOT NULL,
  ename VARCHAR(255) NOT NULL,
  type VARCHAR(255)
  );
  ```

  ```
  – Insert sample data
  INSERT INTO event (eid, date, ename, type) VALUES
  (’E001’, ’2024-02-01’, ’Megaevent’, ’Conference’),
  (’E002’, ’2024-02-15’, ’Tech Symposium’, ’Symposium’),
  (’E003’, ’2024-03-05’, ’Sports Fest’, ’Sports’),
  (’E004’, ’2024-04-10’, ’Cultural Night’, ’Cultural’),
  (’E005’, ’2024-05-20’, ’Workshop Series’, ’Workshop’);
  ```

-
  ```
  – Create student table
  CREATE TABLE student (
  roll VARCHAR(20) PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  dept VARCHAR(255) NOT NULL
  );

  – Insert sample data
  INSERT INTO student (roll, name, dept) VALUES
  (’S001’, ’Alice Johnson’, ’CSE’),
  (’S002’, ’Bob Smith’, ’ECE’),
  (’S003’, ’Charlie Brown’, ’Mechanical’),
  (’S004’, ’Diana Davis’, ’Chemical’),
  (’S005’, ’Edward White’, ’Civil’);
  ```

-
  ```
  – Create role table
  CREATE TABLE role (
  rid VARCHAR(20) PRIMARY KEY,
  rname VARCHAR(255) NOT NULL,
  description VARCHAR(255)
  );

  – Insert sample data
  INSERT INTO role (rid, rname, description) VALUES
  (’R001’, ’Secretary’, ’Manages administrative tasks’),
  (’R002’, ’Treasurer’, ’Handles financial matters’),
  (’R003’, ’President’, ’Leads and oversees the organization’),
  (’R004’, ’Event Coordinator’, ’Plans and coordinates events’),
  (’R005’, ’Public Relations Officer’, ’Manages public relations and communications’);
  ```

-
  ```
  – Create student_from table
  CREATE TABLE student_from (
  college_name VARCHAR(255),
  pid VARCHAR(20),
  PRIMARY KEY (college_name, pid),
  FOREIGN KEY (college_name) REFERENCES college(name),
  FOREIGN KEY (pid) REFERENCES participant(pid)
  );

  – Insert sample data
  INSERT INTO student_from (college_name, pid) VALUES
  (’IITB’, ’P001’),
  (’MIT’, ’P002’),
  (’Stanford’, ’P003’),
  (’Harvard’, ’P004’),
  (’Caltech’, ’P005’);
  ```

-
  ```
  – Create participant_has table
  CREATE TABLE participant_has (
  pid VARCHAR(20),
  eid VARCHAR(20),
  PRIMARY KEY (pid, eid),
  FOREIGN KEY (pid) REFERENCES participant(pid),
  FOREIGN KEY (eid) REFERENCES event(eid)
  );

  – Insert sample data
  INSERT INTO participant_has (pid, eid) VALUES
  (’P001’, ’E001’),
  (’P002’, ’E002’),
  (’P003’, ’E003’),
  (’P004’, ’E004’),
  (’P005’, ’E005’);
  ```

-
  ```
  – Create volunteer_has table
  CREATE TABLE volunteer_has (
  roll VARCHAR(20),
  eid VARCHAR(20),
  PRIMARY KEY (roll, eid),
  FOREIGN KEY (roll) REFERENCES volunteer(roll),
  FOREIGN KEY (eid) REFERENCES event(eid)
  );

  – Insert sample data
  INSERT INTO volunteer_has (roll, eid) VALUES
  (’V001’, ’E001’),
  (’V002’, ’E002’),
  (’V003’, ’E003’),
  (’V004’, ’E004’),
  (’V005’, ’E005’);
  ```

-
  ```
  – Create manage table
  CREATE TABLE manage (
  roll VARCHAR(20),
  eid VARCHAR(20),
  PRIMARY KEY (roll, eid),
  FOREIGN KEY (roll) REFERENCES student(roll),
  FOREIGN KEY (eid) REFERENCES event(eid)
  );

  – Insert sample data
  INSERT INTO manage (roll, eid) VALUES
  (’S001’, ’E001’),
  (’S002’, ’E002’),
  (’S003’, ’E003’),
  (’S004’, ’E004’),
  (’S005’, ’E005’);
  ```

-
  ```
  – Create student_has table
  CREATE TABLE student_has (
  roll VARCHAR(20),
  rid VARCHAR(20),
  PRIMARY KEY (roll, rid),
  FOREIGN KEY (roll) REFERENCES student(roll),
  FOREIGN KEY (rid) REFERENCES role(rid)
  );

  – Insert sample data
  INSERT INTO student_has (roll, rid) VALUES
  (’S001’, ’R001’),
  (’S002’, ’R002’),
  (’S003’, ’R003’),
  (’S004’, ’R004’),
  (’S005’, ’R005’);
  ```

Queries & Responses

- – Query 1

  ```
  SELECT Student.Roll, Student.Name
  FROM Student
  JOIN MANAGE ON Student.Roll = MANAGE.Roll
  JOIN Event ON MANAGE.EID = Event.EID
  WHERE Event.EName = 'Megaevent';
  ```

  – Relational Algebra

  $\pi_{Student.Roll,Student.Name}(\sigma_{Event.EName='Megaevent'}(Student \bowtie MANAGE \bowtie Event))$

  – Response 1

roll	name
S001	Alice Johnson

- – Query 2

  ```
  SELECT Student.Roll, Student.Name
  FROM Student
  JOIN MANAGE ON Student.Roll = MANAGE.Roll
  JOIN Event ON MANAGE.EID = Event.EID
  JOIN Role ON MANAGE.Roll = Role.RID
  WHERE Event.EName = 'Megaevent' AND Role.Rname = 'Secretary';
  ```

  – Relational Algebra

  $\pi_{Student.Roll,Student.Name}(\sigma_{Event.EName='Megaevent' \wedge Role.Rname='Secretary'}(Student \bowtie MANAGE \bowtie Event \bowtie Role))$

  – Response 2

roll	name

  (0 rows)

- – Query 3

  ```
  SELECT Participant.Name
  FROM Participant
  JOIN Student_FROM ON Participant.PID = Student_FROM.PID
  JOIN College ON Student_FROM.College_Name = College.Name
  JOIN Participant_HAS ON Participant.PID = Participant_HAS.PID
  JOIN Event ON Participant_HAS.EID = Event.EID
  WHERE College.Name = 'IIT Bombay' AND Event.EName = 'Megaevent';
  ```

  – Relational Algebra

  $\pi_{Participant.Name}(\sigma_{College.Name='IITBombay' \wedge Event.EName='Megaevent'}(Participant \bowtie Student\_FROM \bowtie College \bowtie Participant\_HAS \bowtie Event))$

  – Response 3

name

  (0 rows)

- – Query 4

  ```
  SELECT DISTINCT College.Name
  FROM College
  JOIN Student_FROM ON College.Name = Student_FROM.College_Name
  JOIN Participant_HAS ON Student_FROM.PID = Participant_HAS.PID
  JOIN Event ON Participant_HAS.EID = Event.EID
  WHERE Event.EName = 'Megaevent';
  ```

  – Relational Algebra

  $\pi_{College.Name}(\sigma_{Event.EName='Megaevent'}(College \bowtie Student\_FROM \bowtie Participant\_HAS \bowtie Event))$

  – Response 4

name
IITB

- – Query 5

  ```
  SELECT DISTINCT Event.EName
  FROM Event
  JOIN MANAGE ON Event.EID = MANAGE.EID
  JOIN Student ON MANAGE.Roll = Student.Roll
  JOIN Role ON MANAGE.Roll = Role.RID
  WHERE Role.Rname = 'Secretary';
  ```

  – Relational Algebra

  $\pi_{Event.EName}(\sigma_{Role.Rname='Secretary'}(Event \bowtie MANAGE \bowtie Student \bowtie Role))$

  – Response 5

ename

  (0 rows)

- – Query 6

  ```
  SELECT DISTINCT Student.Name
  FROM Student
  JOIN Student_HAS ON Student.Roll = Student_HAS.Roll
  JOIN Role ON Student_HAS.RID = Role.RID
  JOIN Volunteer ON Student.Roll = Volunteer.Roll
  JOIN Volunteer_HAS ON Volunteer.Roll = Volunteer_HAS.Roll
  JOIN Event ON Volunteer_HAS.EID = Event.EID
  WHERE Student.Dept = 'CSE' AND Event.EName = 'Megaevent';
  ```

  – Relational Algebra

  $\pi_{Student.Name}(\sigma_{Student.Dept='CSE' \wedge Event.EName='Megaevent'}(Student \bowtie Student\_HAS \bowtie Role \bowtie Volunteer \bowtie Volunteer\_HAS \bowtie Event))$

  – Response 6

name

  (0 rows)

- – Query 7

  ```
  SELECT DISTINCT Event.EName
  FROM Event
  JOIN Volunteer_HAS ON Event.EID = Volunteer_HAS.EID
  JOIN Volunteer ON Volunteer_HAS.Roll = Volunteer.Roll
  JOIN Student ON Volunteer.Roll = Student.Roll
  WHERE Student.Dept = 'CSE';
  ```

  – Relational Algebra

  $\pi_{Event.EName}(\sigma_{Student.Dept='CSE'}(Event \bowtie Volunteer\_HAS \bowtie Volunteer \bowtie Student))$

  – Response 7

ename

  (0 rows)

- – Query 8

  ```
  SELECT College.Name
  FROM College
  JOIN Student_FROM ON College.Name = Student_FROM.College_Name
  JOIN Participant_HAS ON Student_FROM.PID = Participant_HAS.PID
  JOIN Event ON Participant_HAS.EID = Event.EID
  WHERE Event.EName = 'Megaevent'
  GROUP BY College.Name
  ORDER BY COUNT(Participant_HAS.PID) DESC
  LIMIT 1;
  ```

  – Relational Algebra

  $\pi_{College.Name}(\sigma_{Event.EName='Megaevent'}(College \bowtie Student\_FROM \bowtie Participant\_HAS \bowtie Event))$

  – Response 8

name
IITB

- – Query 9

  ```
  SELECT College.Name
  FROM College
  JOIN Student_FROM ON College.Name = Student_FROM.College_Name
  JOIN Participant_HAS ON Student_FROM.PID = Participant_HAS.PID
  GROUP BY College.Name
  ORDER BY COUNT(Participant_HAS.PID) DESC
  LIMIT 1;
  ```

  – Relational Algebra

  $\pi_{College.Name}(College \bowtie Student\_FROM \bowtie Participant\_HAS)$

  – Response 9

name
Harvard

- – Query 10

  ```
  SELECT Student.Dept
  FROM Student
  JOIN Student_HAS ON Student.Roll = Student_HAS.Roll
  JOIN Volunteer ON Student.Roll = Volunteer.Roll
  JOIN Volunteer_HAS ON Volunteer.Roll = Volunteer_HAS.Roll
  JOIN Event ON Volunteer_HAS.EID = Event.EID
  WHERE Event.EID IN (
  SELECT DISTINCT Participant_HAS.EID
  FROM Participant_HAS
  JOIN Student_FROM ON Participant_HAS.PID = Student_FROM.PID
  WHERE Student_FROM.College_Name = 'IITB'
  )
  GROUP BY Student.Dept
  ORDER BY COUNT(Volunteer_HAS.Roll) DESC
  LIMIT 1;
  ```

  – Relational Algebra

  $\pi_{Student.Dept}(\sigma_{Event.EID \in (\pi_{Participant\_HAS.EID}(\sigma_{Student\_FROM.College\_Name='IITB'}(Participant\_HAS \bowtie Student\_FROM)))}(Student \bowtie Student\_HAS \bowtie Volunteer \bowtie Volunteer\_HAS \bowtie Event))$

  – Response 10

dept

  (0 rows)

Note: Some of the Queries do not have equivalent Relational Algebra expressions. In these cases, the closest possible expression has been provided.