Name:Siddarth Sakthi M

Reg.no:230701314

Assignment:1

1. Relational Algebra Queries

a. "Find the names of suppliers who supply some red part."
Answer:

1π_sname (Suppliers ⋈ Supplies ⋈ σ_color='red' (Parts))

b. "Find the IDs of suppliers who supply some red or green part."
Answer:

1π_sid (Supplies ⋈ σ_color='red' ∪ σ_color='green' (Parts))

c. "Find the IDs of suppliers who supply some red part or are based at 21 George Street."
Answer:

1π_sid (σ_color='red' (Supplies ⋈ Parts)) ∪ π_sid (σ_address='21 George Street' (Suppliers))

d. "Find the names of suppliers who supply some red part or are based at 21 George Street."
Answer:

1π_sname (σ_color='red' (Supplies ⋈ Parts) ∪ σ_address='21 George Street' (Suppliers))

e. "Find the IDs of suppliers who supply some red part and some green part."
Answer:

1π_sid (σ_color='red' (Supplies ⋈ Parts)) ∩ π_sid (σ_color='green' (Supplies ⋈ Parts))

f. "Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid."
Answer:

1π_s1.sid, s2.sid (Supplies as s1 ⋈ Supplies as s2 | s1.cost > s2.cost)

g. "Find the IDs of suppliers who supply only red parts."
Answer:

1π_sid (Suppliers) - π_sid (Supplies ⋈ σ_color≠'red' (Parts))

h. "Find the IDs of suppliers who supply every part."
Answer:

1π_sid (Suppliers) - π_sid (Supplies ⋈ (Parts - Supplies))

2. Relational Database Queries

a) Get the complete details of all flights to New Delhi.
Answer:

1σ_dest='New Delhi' (flight)

b) Get the details about all flights from Chennai to New Delhi.
Answer:

1σ_src='Chennai' ∧ dest='New Delhi' (flight)

c) Find only the flight numbers for passenger with pid 123 for flights to Chennai before 06/11/2020.
Answer:

1π_fid (σ_pid=123 (booking) ⋈ σ_dest='Chennai' ∧ fdate<'2020-11-06' (flight))

d) Find the passenger names who have bookings on at least one flight.
Answer:

1π_pname (passenger ⋈ booking)

e) Find the passenger names for those who do not have any bookings in any flights.
Answer:

1π_pname (passenger) - π_pname (passenger ⋈ booking)

f) Find the agency names for agencies located in the same city as the passenger with passenger id 123.
Answer:

1π_aname (agency ⋈ σ_pid=123 (passenger))

g) Get the details of flights that are scheduled on both dates 01/12/2020 and 02/12/2020 at 16:00 hours.
Answer:

1σ_fdate='2020-12-01' ∧ time='16:00' (flight) ∩ σ_fdate='2020-12-02' ∧ time='16:00' (flight)

h) Get the details of flights that are scheduled on both dates 01/12/2020 and 02/12/2020 at 16:00 hours.
Answer:
(Same as g, no change needed.)

i) Find the agency names for agencies that do not have any bookings for passenger with id 123.
Answer:

1π_aname (agency) - π_aname (agency ⋈ booking where pid=123)

j) Find the details of all male passengers who are associated with the jet agency.
Answer:

π_pid, pname, pgender, pcity (σ_pgender='Male' (passenger) ⋈ (booking ⋈ σ_aname='jet' (agency)))

3. E-R Diagram for a Car-Insurance Company
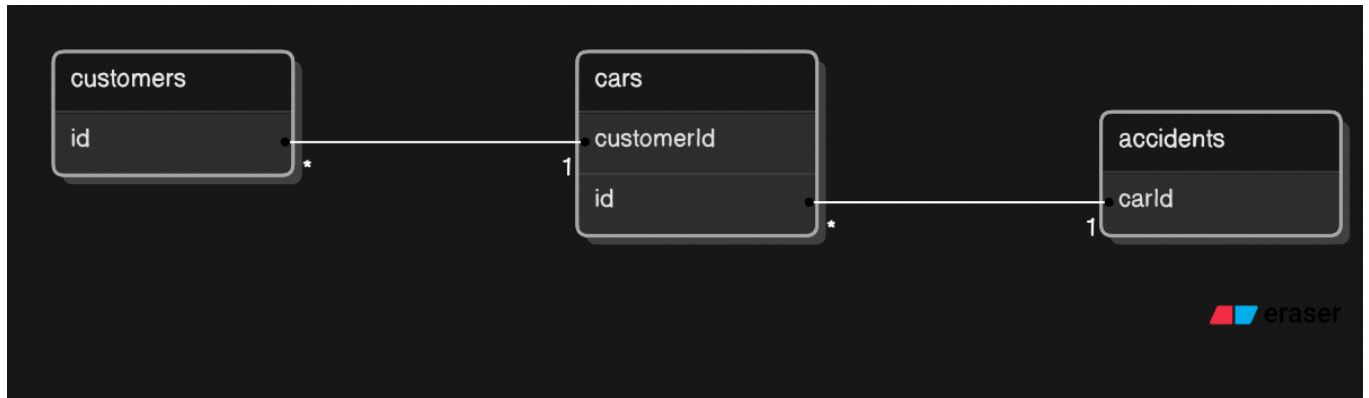
Entities:

- **Customer**
    - Attributes: Customer_ID (PK), Name, Address, Phone
- **Car**
    - Attributes: Car_ID (PK), Make, Model, Year, License_Plate
- **Accident**
    - Attributes: Accident_ID (PK), Date, Description, Severity

Relationships:

- **Owns**

    - Between Customer and Car (1 to Many): A customer can own one or more cars.

- **Involved_In**

    - Between Car and Accident (0 to Many): A car can be involved in zero or more accidents.

E-R Diagram Representation:

1[Customer] --- Owns ---< [Car] --- Involved_In ---< [Accident]

4. E-R Diagram for a Hospital

Entities:

- **Patient**

    - Attributes: Patient_ID (PK), Name, DOB, Address

- **Doctor**

    - Attributes: Doctor_ID (PK), Name, Specialty, Phone

- **Test**

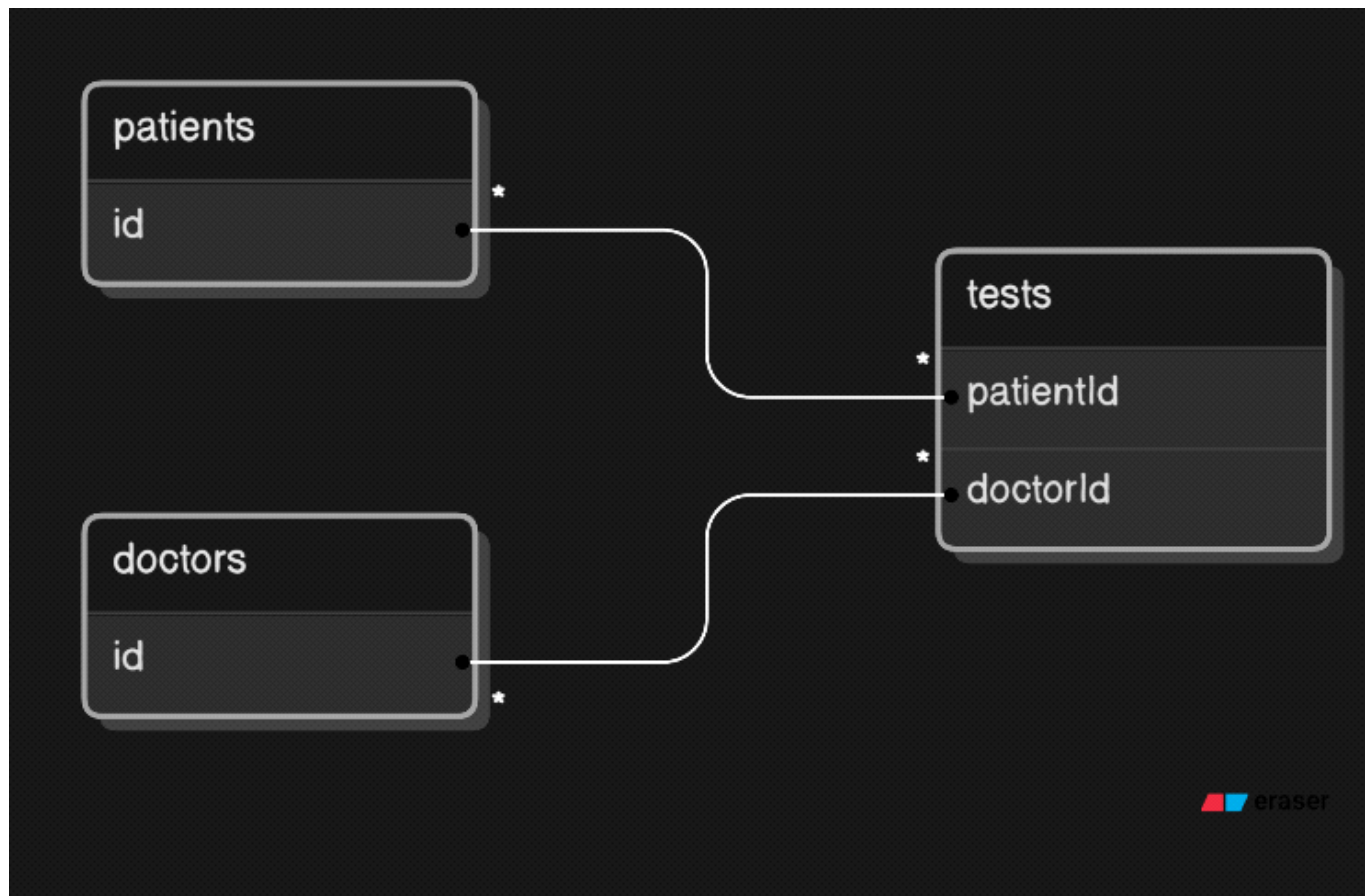   - Attributes: Test_ID (PK), Test_Name, Test_Date, Results

Relationships:

- **Undergoes**

    - Between Patient and Test (1 to Many): A patient can undergo multiple tests.

- **Conducts**

    - Between Doctor and Test (1 to Many): A doctor can conduct multiple tests.

E-R Diagram Representation:

1[Patient] --- Undergoes ---< [Test]

2[Doctor] --- Conducts ---< [Test]

5. E-R Diagram for a University Registrar's Office

Entities:

- **Course**
    - Attributes: Course_Number (PK), Title, Credits, Syllabus, Prerequisites
- **Course_Offering**
    - Attributes: Offering_ID (PK), Course_Number (FK), Year, Semester, Section_Number, Instructor_ID (FK), Timings, Classroom
- **Student**
    - Attributes: Student_ID (PK), Name, Program
- **Instructor**

- Attributes: Instructor_ID (PK), Name, Department, Title

- **Enrollment**

 - Attributes: Student_ID (FK), Offering_ID (FK), Grade

Relationships:

- **Teaches**

  - Between Instructor and Course_Offering (1 to Many): An instructor can teach multiple course offerings.

- **Enrolled_In**

  - Between Student and Course_Offering (Many to Many): A student can enroll in multiple course offerings, and a course offering can have multiple students.

E-R Diagram Representation:

1[Course] --- Offers ---< [Course_Offering] --- Teaches --- [Instructor]

2[Student] --- Enrolled_In ---< [Course_Offering]
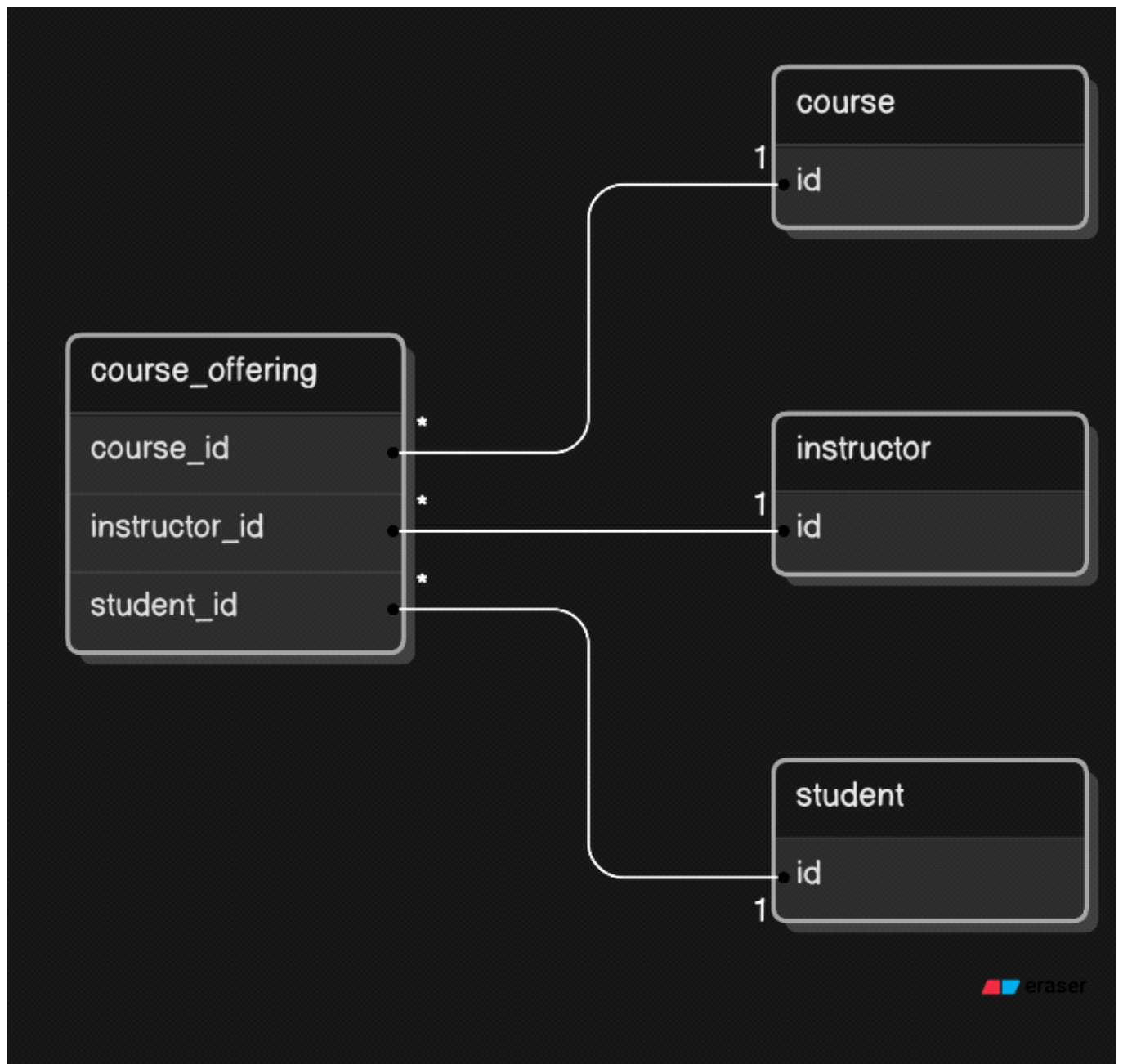
Assumptions:

- Each course can have multiple offerings in different semesters.

- Each student can enroll in multiple courses, and each course can have multiple students.

- Each course offering can have multiple instructors (team teaching).

6. E-R Diagrams for Student Marks in Exams

a. E-R Diagram Using Ternary Relationship

Entities:

- **Student**

    - Attributes: Student_ID (PK), Name

- **Course_Offering**

    - Attributes: Offering_ID (PK), Course_Number, Year, Semester

- **Exam**

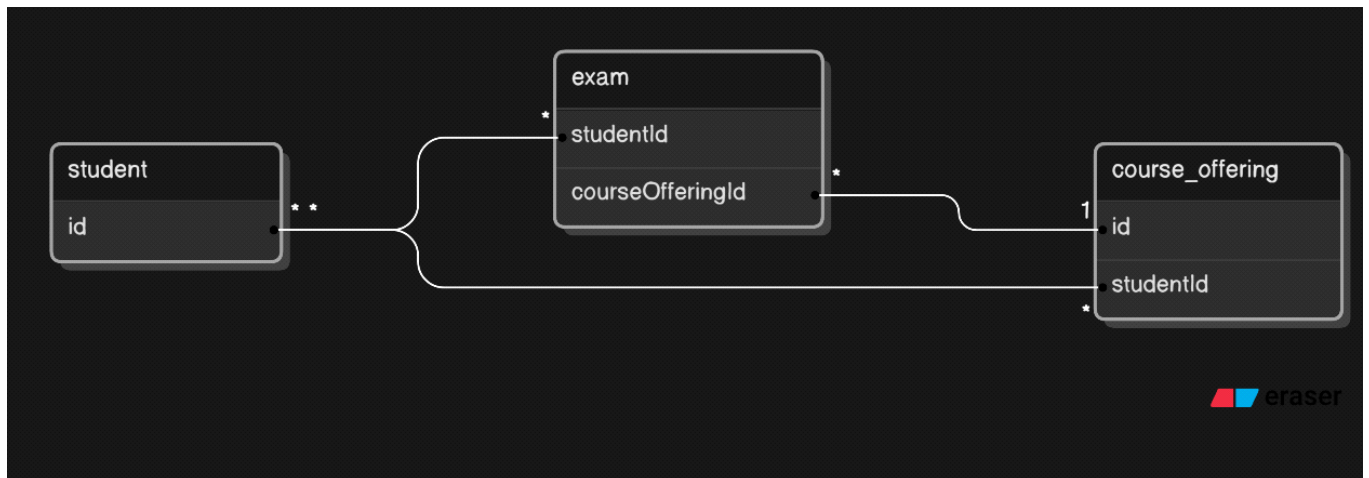   - Attributes: Exam_ID (PK), Exam_Name, Date

Relationships:

- **Takes**

    - A ternary relationship between Student, Course_Offering, and Exam.

    - Attributes: Marks

E-R Diagram Representation:

1[Student] --- Takes ---< [Exam]

2 | |

3   [Course_Offering]

b. E-R Diagram Using Binary Relationship

Entities:

- **Student**
    - Attributes: Student_ID (PK), Name
- **Course_Offering**
    - Attributes: Offering_ID (PK), Course_Number, Year, Semester
- **Marks**
    - Attributes: Student_ID (FK), Offering_ID (FK), Marks

Relationships:

- **Enrolled_In**
    - Between Student and Course_Offering (Many to Many).
- **Has_Marks**
    - Between Student and Marks (1 to Many): A student can have multiple marks for different exams in the same course offering.

E-R Diagram Representation:

1[Student] --- Enrolled_In ---< [Course_Offering]

2[Student] --- Has_Marks ---< [Marks]