Name: Nazia Zarin
Roll: 05-002-08

Assignment 1:

You have to develop a Human Resource Management database for a company. It requires that you have to add some information of departments, employees and projects including company. As mentioned, every department has many employees and each employee works for a department and each department is led by only one manager who is also an employee. Initially a new department need not have any employee. Here, though an employee belongs a department but they can work for different projects at the same time. Each employee can work for different job positions like Developer, Programmer and front- end designer.

ER diagram for Human Resource Management

Step 1: Identify the Attributes and Primary key for each Entity

1. Employees : employee_id(primary key), Name, Address, Gender, Dob, hire_date, salary, position_id(foreign key), dept_id.
2. Departments : dept_id(primary key), Name, num_of_employee, hod, Location.
3. Projects : Project_id(primary key), name, num_of_employee, location
4. Dependents : D_no, Gender, relationship.
5. Positions :position_id (Primary Key), title, description

Step 2: Identify the Relationship

Entities have some relationships with each other. Relationships define how entities are associated with each other.

Let's Establishing Relationships between them are:

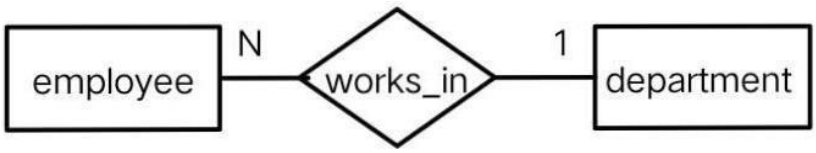
1. Employee works in Departments
2. Employee(manager) manages Department
3. Employee works on project
4. Department controls Projects
5. Employee has dependents
6. Employee holds position

Step 3: Identify the Cardinality Ratio and Participation

1. Employee-Department

Relationship: Many-to-One

Many employee works in one Department but one employee can not work in many departments.



2. Employee(manager)-Department

Relationship: one-to-one

Employee works under the manager of the Department and the manager records the date of joining of employee in the department.

3. Employee-Project

Relationship: Many-to-Many

Many employee works on several projects and the number of hours worked by the employee on a single project is recorded.

4. Department-project

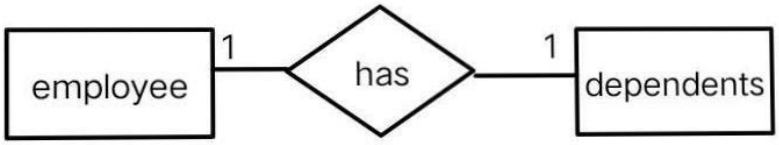
Relationship: one-to-Many

One department has many projects but one project can not come under many departments.

5. Employee-Dependent

Relationship: one-to-one

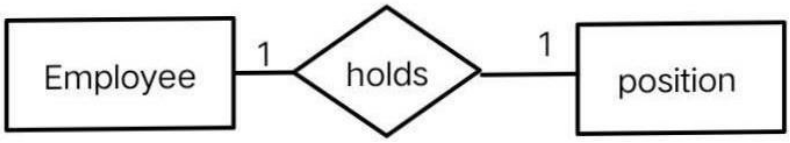
Each Employee has dependents. Each dependent is dependent of only one employee.



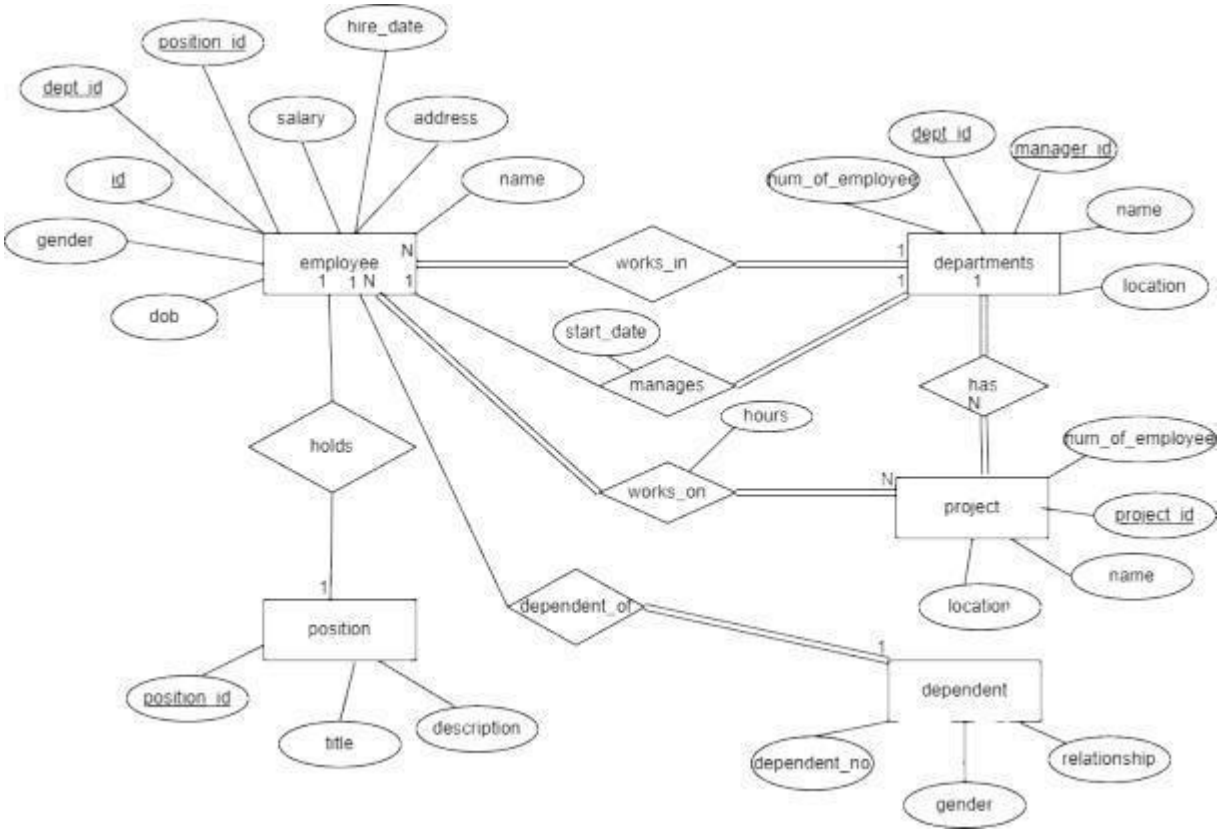
6. Employee-Position

Relationship: one-to-One

An Employee can have one Position at a time and a Position is held by one Employee.



Step 4: ER Diagram Representation



ERD to table:

1. Employee(id, Name, Address, Gender, Dob, hire_date, salary, position_id, dept_id.
2. Departments : dept_id, Name, num_of_employee, manager_id, Location.
3. Projects : Project_id, name, num_of_employee, location
4. Dependents : D_no, Gender, employee_id
5. Positions :position_id, title, description
6. works_on (id, hours, start_date, employee_id, Project_id.

SQL code

```
CREATE DATABASE human_resourse_management;

CREATE TABLE employee (

   id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    name VARCHAR(20),
    address VARCHAR(20),
    dob DATE,
    gender CHAR(10), hire_date
    DATE,
    salary DECIMAL(10, 2),
    position_id INT,
    FOREIGN KEY (position_id) REFERENCES position(position_id), dept_id
    INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);
CREATE TABLE departments (
    dept_id INT PRIMARY KEY
    AUTO_INCREMENT, name VARCHAR(20),
    location VARCHAR(30),
num_of_employee INT,
    manager_id INT,
    FOREIGN KEY (manager_id) REFERENCES employee(employee_id)
);

CREATE TABLE project (
    project_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    location VARCHAR(30),
num_of_employee INT,
);
```

```
CREATE TABLE position (
   position_id INT PRIMARY KEY AUTO_INCREMENT
   , title VARCHAR(100),
   description TEXT
);
CREATE TABLE works_on (
id INT PRIMARY KEY
   AUTO_INCREMENT, hours
   DECIMAL(5, 2),
   start_date DATE,
employee_id INT,
   FOREIGN KEY (employee_id) REFERENCES employee(employee_id),
Project_id INT,
   FOREIGN KEY (project_id) REFERENCES project(project_id)
);

CREATE TABLE dependent (
   dependent_id INT PRIMARY KEY
   AUTO_INCREMENT, name VARCHAR(20),
   gender VARCHAR(10),
   employee_id INT,
   FOREIGN KEY (employee_id) REFERENCES employee(employee_id)
);
```

Question -2: ER diagram for University Management System

ER diagram for University Management System

Step 1: Identify the Attributes and Primary key for each Entity

1. Students: Student_ID (Primary Key), Name, Date_of_Birth, Age, Gender, Address, Phone_Number, Email

2. Courses: Course_ID (Primary Key), Course_Name, Credits, Duration

3. Faculty: Faculty_ID (Primary Key), Name, Date_of_Birth, Gender, Address, Phone_Number, Email, Hire_Date, Salary, Department, Designation.

4. Department: Dept_ID (Primary Key), Dept_Name, Location

5. Hostel: Hostel_id (primary key), Hostel_name, No_of_seats

6. Subjects: Sub_id(primary key), Sub_name

7. Exams: Exam_code(primary key), Room, Date, Time

Step 2: Identify the Relationship

Entities have some relationships with each other. Relationships define how entities are associated with each other.

Let's Establishing Relationships between them are:

1. Faculty teaches student
2. Faculty takes subjects
3. Faculty belongs to department
4. Students enrolls courses
5. Student living in hostel
6. Course handles Department
7. Department conducts exams

Step 3: Identify the Cardinality Ratio and Participation

1. Student— Faculty

Relationship: Many-to-One

A faculty members teaches many students , but each student is taught by one faculty members.

2. Faculty— Subject

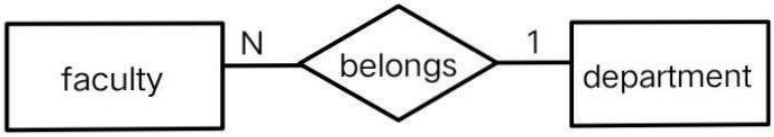
Relationship: One-to-many

A faculty members takes many subjects , but each subject is taught by one faculty members.

3. Faculty— Department

Relationship: Many-to-One

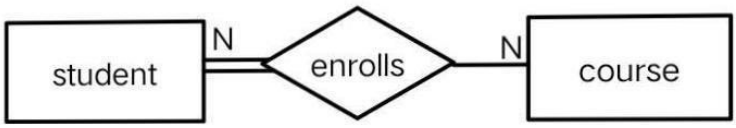
A department can have multiple faculty members, but a faculty member belongs to only one department.



4. Student — Course

Relationship: Many-to-Many

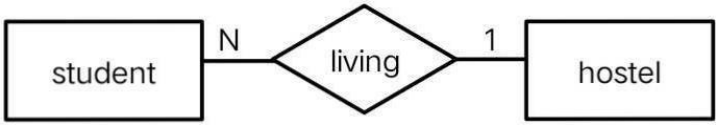
A student can enroll in many courses, and each course enrolled by many students.



5. Student — Hostel

Relationship: Many-to-One

A hostel accommodates many students, but single student is living in one hostel.



6. Course — Department

Relationship: Many-to-One

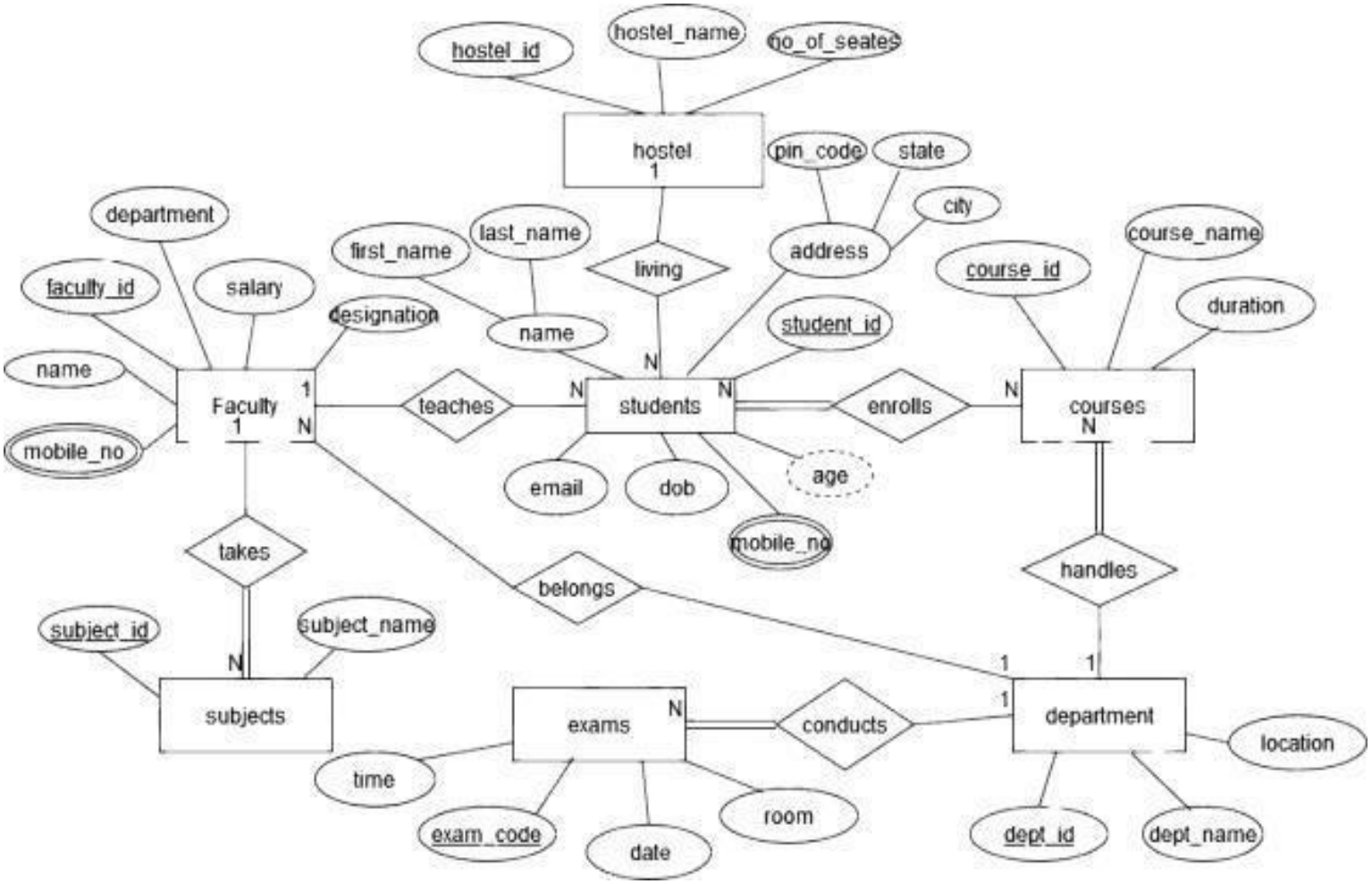
A department handles multiple courses, but each course is handled by one department.

7. Department — Exams

Relationship: one-to-Many

A department conducts multiple exams, but each exam is conducted by one department.

Step 4: ER Diagram Representation



ERD to table :

1. Students: Student_ID, first_name, last_name, Date_of_Birth, Age, Gender, Address, Phone_Number, Email

2. Courses: Course_ID, Course_Name, Credits, Duration, dept_id

3. Faculty: Faculty_ID, Name, Date_of_Birth, Gender, Address, Phone_Number, Email, Hire_Date, Salary, dept_id, Designation.

4. Department: Dept_ID , Dept_Name, Location

5. Hostel: Hostel_id, Hostel_name, No_of_seats

6. Subjects: Sub_id, Sub_name, course_id

7. Exams: Exam_code, Room, Date, Time, subject_id.

8. Address : city, state, pin_code

SQL code:

```
CREATE DATABASE university_management_system ;


CREATE TABLE Student (
   student_id INT PRIMARY KEY,
   first_name VARCHAR(255),
   last_name VARCHAR(255),
   city VARCHAR(100),
   state VARCHAR(100),
   pin_code VARCHAR(10),
   mobile_no VARCHAR(20),
   email VARCHAR(100),
   dob DATE,
   age INT,
   hostel_id INT,
   dept_id INT,
   FOREIGN KEY (hostel_id) REFERENCES Hostel(hostel_id),
   FOREIGN KEY (dept_id) REFERENCES Department(dept_id),
   FOREIGN KEY (street, city, state) REFERENCES Address(street, city, state)
);
```

```
CREATE TABLE Address
   ( state
   VARCHAR(255), city
   VARCHAR(100),
   Pin_code VARCHAR(10),
   PRIMARY KEY (pin_code, city, state)
);

CREATE TABLE Faculty (
   faculty_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(30),
   mobile_no VARCHAR(20),
   salary DECIMAL(10, 2),
   designation VARCHAR(255),
   dept_id INT,
   FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);
CREATE TABLE Course (
   course_id INT PRIMARY KEY AUTO_INCREMENT,
   course_name VARCHAR(30),
   duration INT,
   dept_id INT,
   FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);
CREATE TABLE Department (
   dept_id INT PRIMARY KEY AUTO_INCREMENT,
   dept_name VARCHAR(30),
```

```
    location VARCHAR(30)
);
CREATE TABLE Subject (
    subject_id INT PRIMARY KEY AUTO_INCREMENT,
    subject_name VARCHAR(30), course_id
    INT,
    FOREIGN KEY (course_id) REFERENCES Course(course_id)
);
CREATE TABLE Exam (
    exam_code INT PRIMARY
    KEY, date DATE,
    time TIME,
    room VARCHAR(50),
    subject_id INT,
    FOREIGN KEY (subject_id) REFERENCES Subject(subject_id)
);

CREATE TABLE Hostel (
    hostel_id INT PRIMARY KEY AUTO_INCREMENT,
    hostel_name VARCHAR(30),
    no_of_seats INT
);
```

Question 3: Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.

ER diagram for car insurance company

Step 1: Identify the Attributes and Primary key for each Entity

1. Customer: C_id(primary key), name, address, phone_no
2. Car: car_id(primary key), color, model, year, engine_no
3. Insurance policy : policy_id(primary key), name, start_date, end_date
4. Premium installments : premium_no, amount, received_date, due_date
5. Accident : accident_id(primary key), car_id(foreign key), date, place, report_no, damage_cost

Step 2: Identify the Relationship

Entities have some relationships with each other. Relationships define how entities are associated with each other.

Let's Establishing Relationships between them are:

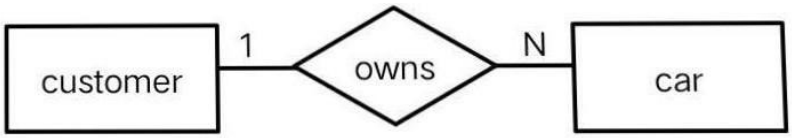
1. Customer owns Car
2. Car insured by Insurance policy
3. Car associated with Accident
4. Insurance policy involves premium installments

Step 3: Identify the Cardinality Ratio and Participation

1. Customer-car

Relationship: One-to-many

One customer can own multiple cars but one car can't own by multiple customers.



2. Car-insurance_policy

Relationship: One-to-One

One car can insured by only one insurance policy.

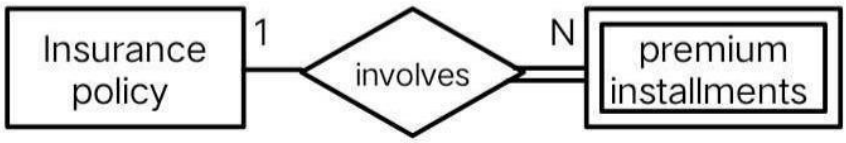
3. Car-Accident

Relationship: one-to-Many

One car can meet with multiple accidents.

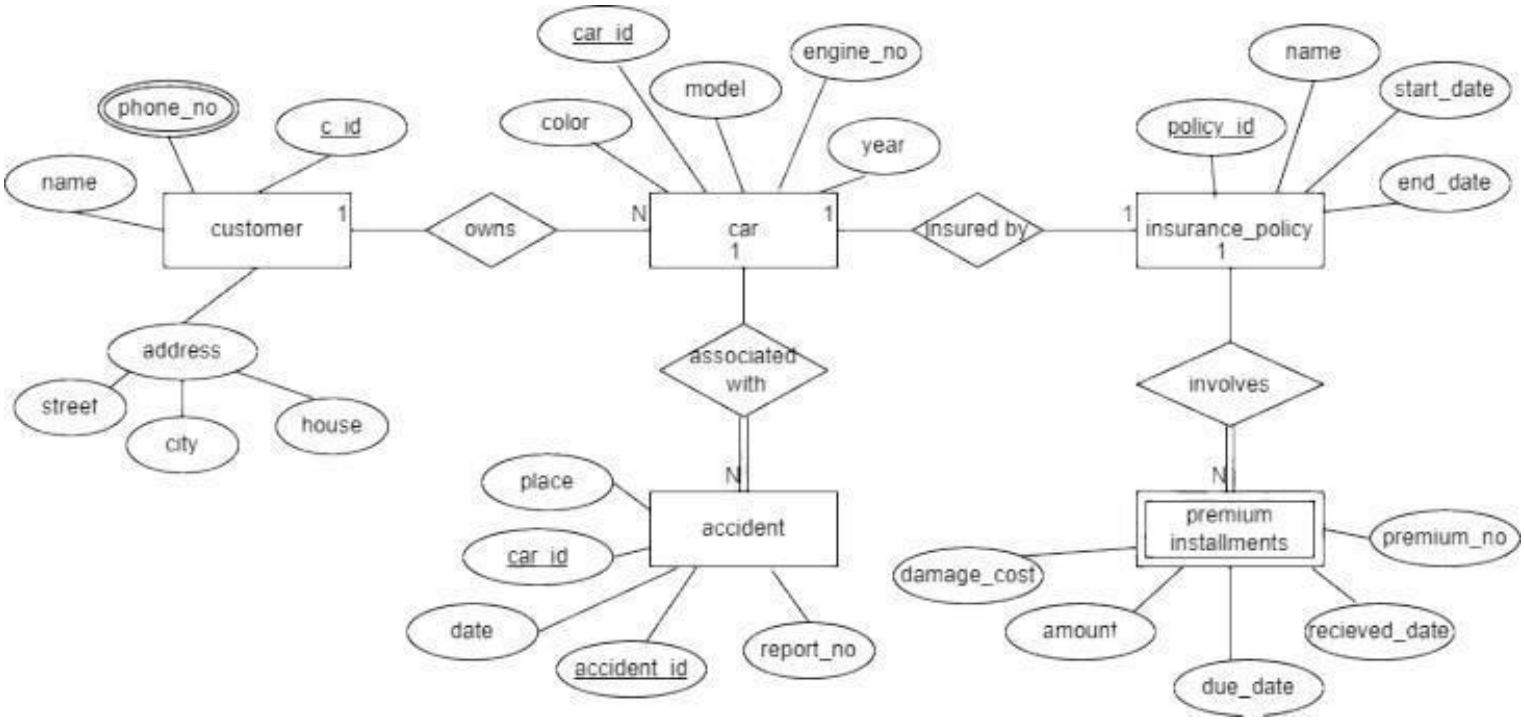
4. Insurance_policy-premium installments

Relationship: one-to-Many



One insurance policy has multiple premium installments but one premium installments has one Insurance policy.

Step 4: ER Diagram Representation



ERD to table :

1. Customer: C_id, name, address, phone_no
2. Car: car_id, color, model, year, engine_no
3. Insurance policy : policy_id, name, start_date, end_date
4. Premium installments : premium_no, amount, received_date, due_date, policy_id
5. Accident : accident_id, car_id, date, place, report_no, damage_cost

SQL code

CREATE DATABASE Car_Insurance_Company;

CREATE TABLE customer (

C_id INT PRIMARY KEY AUTO_INCREMENT,

```
 phone_no VARCHAR(20),
 name VARCHAR(20),
 address VARCHAR(200),
FOREIGN KEY (street, city, house) REFERENCES Address(street, city, house)
);
CREATE TABLE address
( street VARCHAR(100),
city VARCHAR(50),
house VARCHAR(50),
PRIMARY KEY (street, city, house)
);
CREATE TABLE car (
   car_id INT PRIMARY KEY
   AUTO_INCREMENT, model VARCHAR(50),
   year YEAR,
   engine_no VARCHAR(50),
  color VARCHAR(50)
);
CREATE TABLE insurance_policy (
   policy_id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(20),
   start_date DATE,
   end_date DATE
);
CREATE TABLE premium_installments (
   premium_no INT,
```

```
    policy_id INT,
    amount DECIMAL(10, 2),
   damage_cost DECIMAL(10, 2),
    received_date DATE,
    due_date DATE,
    PRIMARY KEY (premium_no, policy_id),
    FOREIGN KEY (policy_id) REFERENCES insurance_policy(policy_id)
);
CREATE TABLE accident (
    accident_id INT PRIMARY KEY
    AUTO_INCREMENT, date DATE,
    report_no VARCHAR(50),
    place VARCHAR(50),
    car_id INT,
    FOREIGN KEY (car_id) REFERENCES car(car_id)
);
```