Q.ddl commands with integrity constraints

```
CREATE TABLE DEPARTMENT (
   Dname VARCHAR(50),
   Dnumber INT PRIMARY KEY,
   Mgr_ssn CHAR(9),
   Mgr_start_date DATE,
   FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
);

CREATE TABLE EMPLOYEE (
   Fname VARCHAR(20),
   Minit CHAR(1),
   Lname VARCHAR(20),
   Ssn CHAR(9) PRIMARY KEY,
   Bdate DATE,
   Address VARCHAR(100),
   Sex CHAR(1),
   Salary DECIMAL(10, 2),
   Super_ssn CHAR(9),
   Dno INT,
   FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
   FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
);

CREATE TABLE PROJECT (
   Pname VARCHAR(50),
   Pnumber INT PRIMARY KEY,
   Plocation VARCHAR(100),
   Dnum INT,
   FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber)
);

CREATE TABLE WORKS_ON (
   Essn CHAR(9),
   Pno INT,
   Hours DECIMAL(5, 2),
   PRIMARY KEY (Essn, Pno),
   FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
   FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)
);
```

Q . Ddl with integrity constraints

```
CREATE TABLE Company (
    Company_id INT PRIMARY KEY,
    Name VARCHAR(50),
    Address VARCHAR(100)
);

CREATE TABLE Customer (
    Customer_id INT PRIMARY KEY,
    Name VARCHAR(50),
    Address VARCHAR(100),
    Phone VARCHAR(15),
    Insurance_company VARCHAR(50)
);

CREATE TABLE Car (
    Car_Number VARCHAR(15) PRIMARY KEY,
    Car_Model VARCHAR(50),
    Owner_id INT,
    FOREIGN KEY (Owner_id) REFERENCES Customer(Customer_id)
);

CREATE TABLE Accidents (
    Accident_id INT PRIMARY KEY,
    Car_Number VARCHAR(15),
    Location VARCHAR(100),
    Date DATE,
    Time TIME,
    FOREIGN KEY (Car_Number) REFERENCES Car(Car_Number)
);
```

Q.create and insert

```
CREATE TABLE DEPARTMENT (
   Dname VARCHAR(50),
   Dnumber INT PRIMARY KEY,
   Mgr_ssn CHAR(9),
   Mgr_start_date DATE,
   FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
);

CREATE TABLE EMPLOYEE (
   Fname VARCHAR(20),
   Minit CHAR(1),
   Lname VARCHAR(20),
   Ssn CHAR(9) PRIMARY KEY,
   Bdate DATE,
   Address VARCHAR(100),
   Sex CHAR(1),
   Salary DECIMAL(10, 2),
   Super_ssn CHAR(9),
   Dno INT,
   FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
   FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
);
-- Insert
INSERT INTO EMPLOYEE VALUES (‘John’, ‘A’, ‘Smith’, 101, ‘1990-01-01’, ‘NY’, ‘M’, 50000,
NULL, 1);
INSERT INTO EMPLOYEE VALUES (‘Alice’, ‘B’, ‘Brown’, 102, ‘1985-03-15’, ‘CA’, ‘F’, 60000, 101,
2);
INSERT INTO EMPLOYEE VALUES (‘David’, ‘C’, ‘Clark’, 103, ‘1992-07-21’, ‘TX’, ‘M’, 55000,
101, 1);
```

```
INSERT INTO EMPLOYEE VALUES (‘Emily’, ‘D’, ‘Davis’, 104, ‘1991-10-11’, ‘FL’, ‘F’, 58000, 102, 2);
```

- Update salary

```
UPDATE EMPLOYEE SET Salary = Salary * 1.2;
```

Q.remove by department

```
CREATE TABLE PROJECT (
   Pname VARCHAR(50),
   Pnumber INT PRIMARY KEY,
   Plocation VARCHAR(100),
   Dnum INT,
   FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber)
);

CREATE TABLE WORKS_ON (
   Essn CHAR(9),
   Pno INT,
   Hours DECIMAL(5, 2),
   PRIMARY KEY (Essn, Pno),
   FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
   FOREIGN KEY(Pno) REFERENCES PROJECT (Pnumber)
);
```

- Insert

```
INSERT INTO PROJECT VALUES (‘AI Dev’, 201, ‘NY’, 1);
INSERT INTO PROJECT VALUES (‘WebApp’, 202, ‘CA’, 2);
INSERT INTO PROJECT VALUES (‘ML Tool’, 203, ‘TX’, 1);

INSERT INTO PROJECT VALUES (‘IoT System’, 204, ‘FL’, 2);


INSERT INTO WORKS_ON VALUES (101, 201, 20);

INSERT INTO WORKS_ON VALUES (102, 202, 25);

INSERT INTO WORKS_ON VALUES (103, 203, 15);

INSERT INTO WORKS_ON VALUES (104, 204, 18);
```

- Delete all projects from one department (example: Dnum = 1)

```
DELETE FROM PROJECT WHERE Dnum = 1;
```

Q. change the address of all the customers with the name beginning with letter “A”.

```
-- Insert

INSERT INTO Company VALUES (1, 'XYZ Insure', 'NY');

INSERT INTO Company VALUES (2, 'SafeDrive', 'CA');

INSERT INTO Company VALUES (3, 'InsurePro', 'TX');

INSERT INTO Company VALUES (4, 'TrustCover', 'FL');


INSERT INTO Customer VALUES (1, 'Alice Johnson', 'Old Address', '1234567890', 'XYZ Insure');

INSERT INTO Customer VALUES (2, 'Aaron Paul', 'Old Address', '0987654321', 'SafeDrive');

INSERT INTO Customer VALUES (3, 'Bob Stone', 'Some Address', '5555555555', 'TrustCover');

INSERT INTO Customer VALUES (4, 'Anna Lee', 'Another Address', '4444444444', 'InsurePro');


-- Update address

UPDATE Customer SET Address = 'New Address' WHERE Name LIKE 'A%';
```

Q. Create and insert four rows in the following relations. Write a query to delete all cars owned by a single owner.

```
-- Insert

INSERT INTO Car VALUES (‘ABC123’, ‘Honda Civic’, 1);

INSERT INTO Car VALUES (‘XYZ456’, ‘Toyota Camry’, 2);

INSERT INTO Car VALUES (‘LMN789’, ‘Ford Focus’, 2);

INSERT INTO Car VALUES (‘DEF321’, ‘Tesla Model 3’, 3);


INSERT INTO Accidents VALUES (301, ‘ABC123’, ‘NY’, ‘2024-01-01’, ’10:00:00’);

INSERT INTO Accidents VALUES (302, ‘XYZ456’, ‘CA’, ‘2024-02-01’, ’12:00:00’);

INSERT INTO Accidents VALUES (303, ‘LMN789’, ‘CA’, ‘2024-03-01’, ’14:00:00’);

INSERT INTO Accidents VALUES (304, ‘DEF321’, ‘TX’, ‘2024-04-01’, ’16:00:00’);
```

- Delete cars owned by a single owner

```
DELETE FROM Car WHERE Owner_id IN (

   SELECT Owner_id FROM Car GROUP BY Owner_id HAVING COUNT(*) = 1

);
```

Q) Create and insert four rows in the following relations. Write a query to find average salary of all employees

```
SELECT Dno, AVG(Salary) AS Avg_Salary

FROM EMPLOYEE

GROUP BY Dno;
```

Q.Create and insert four rows in the following relations. Write a query to count the number of project belonging to each department.

```
SELECT Dnum, COUNT(*) AS Project_Count

FROM PROJECT

GROUP BY Dnum;
```

Q) Create and insert four rows in the following relations. Write a query to find number of customers in each insurance

```
CREATE TABLE Company (

   Company_id INT,

   Name VARCHAR(50),

   Address VARCHAR(100)

);

CREATE TABLE Customer (

   Customer_id INT,

   Name VARCHAR(50),

   Address VARCHAR(100),

   Phone VARCHAR(15),

   Insurance_company INT

);

INSERT INTO Company VALUES (1, ‘LIC’, ‘Delhi’), (2, ‘HDFC’, ‘Mumbai’), (3, ‘ICICI’, ‘Chennai’), (4, ‘SBI Life’, ‘Bangalore’);

INSERT INTO Customer VALUES

(101, ‘Alice’, ‘Delhi’, ‘1234567890’, 1),

(102, ‘Bob’, ‘Mumbai’, ‘2345678901’, 2),

(103, ‘Charlie’, ‘Chennai’, ‘3456789012’, 1),

(104, ‘David’, ‘Kolkata’, ‘4567890123’, 3);
```

```
SELECT Company.Name, COUNT(*) AS Customer_Count
FROM Customer
JOIN Company ON Customer.Insurance_company = Company.Company_id
GROUP BY Company.Name;
```

Q. Create and insert four rows in the following relations. Write a query to arrange the accidents date wise.

```
CREATE TABLE Car (
   Car_Number INT,
   Car_Model VARCHAR(50),
   Owner_id INT
);
CREATE TABLE Accidents (
   Accident_id INT,
   Car_Number INT,
   Location VARCHAR(100),
   Date DATE,
   Time TIME
);
INSERT INTO Car VALUES (1001, 'Swift', 101), (1002, 'i20', 102), (1003, 'City', 103), (1004, 'Polo', 104);
INSERT INTO Accidents VALUES
(1, 1001, 'Delhi', '2024-04-10', '10:30:00'),
(2, 1002, 'Mumbai', '2024-03-15', '08:45:00'),
(3, 1003, 'Chennai', '2024-04-01', '12:00:00'),
(4, 1004, 'Kolkata', '2024-02-20', '16:15:00');
SELECT * FROM Accidents ORDER BY Date;
```

Q. Create and insert four rows in the following relations. Write a query to perform Equi Join.

```
SELECT *

FROM EMPLOYEE E, DEPARTMENT D

WHERE E.Dno = D.Dnumber;
```

Q. Create and insert four rows in the following relations. Write a query to perform Natural Join.

```
SELECT *

FROM EMPLOYEE

NATURAL JOIN DEPARTMENT;
```

Q.Create and insert four rows in the following relations. Write a query to perform Left Outer Join.

```
SELECT *

FROM EMPLOYEE E

LEFT OUTER JOIN DEPARTMENT D

ON E.Dno = D.Dnumber;
```

Q.Create and insert four rows in the following relations. Write a query to perform Right Outer Join.

```
SELECT *

FROM EMPLOYEE E

RIGHT OUTER JOIN DEPARTMENT D
```

```
ON E.Dno = D.Dnumber;
```

Q. Create and insert four rows in the following relations. Write a query to perform Inner Join.

```
SELECT *

FROM EMPLOYEE E

INNER JOIN DEPARTMENT D

ON E.Dno = D.Dnumber;
```

Q.Write a query to create a user and a table. Grant Select and insert privilege to this user and revoke select privilege from same user;

Q. Write a query to create a user and a table. Grant Update and delete privilege to this user and revoke delete privilege From same user;

```
CREATE USER user_test IDENTIFIED BY password123;

CREATE TABLE student (

   Student_id INT PRIMARY KEY,

   Name VARCHAR(50),

   Course VARCHAR(50)

);

GRANT SELECT, INSERT ON student TO user_test;

REVOKE SELECT ON student FROM user_test;

GRANT UPDATE, DELETE ON student TO user_test;

GRANT ALL PRIVILEGES ON student TO user_test;

REVOKE ALL PRIVILEGES ON student FROM user_test;

REVOKE UPDATE, DELETE ON student FROM user_test;
```

Q.Create and insert four rows in the following relations. Write a query to create a virtual table of join of these two Tables

```
CREATE TABLE Car (
    Car_Number INT PRIMARY KEY,
    Car_Model VARCHAR(50),
    Owner_id INT
);

CREATE TABLE Accidents (
    Accident_id INT PRIMARY KEY,
    Car_Number INT,
    Location VARCHAR(100),
    Date DATE,
    Time TIME,
    FOREIGN KEY (Car_Number) REFERENCES Car(Car_Number)
);
-- Insert into Car
INSERT INTO Car VALUES
(101, 'Maruti Swift', 1),
(102, 'Hyundai i20', 2),
(103, 'Honda City', 3),
(104, 'Volkswagen Polo', 4);

-- Insert into Accidents
INSERT INTO Accidents VALUES
(1, 101, 'Delhi', '2024-01-15', '10:30:00'),
(2, 102, 'Mumbai', '2024-03-22', '14:00:00'),
(3, 103, 'Chennai', '2024-02-10', '08:45:00'),
(4, 104, 'Bangalore', '2024-04-01', '18:20:00');
CREATE VIEW Car_Accident_View AS
SELECT
   C.Car_Number, C.Car_Model, C.Owner_id,
   A.Accident_id, A.Location, A.Date, A.Time
FROM
   Car C
JOIN
   Accidents A ON C.Car_Number = A.Car_Number;
SELECT * FROM Car_Accident_View;
```

Q) Create and insert four rows in the following relations. Write a query to create a virtual table of join of these two

Tables. Identify the number of accidents occurred to one owner.

Car(Car_Number, Car_Model, Owner_id)

Accidents (Accident_id, Car_Number, Location, date, time)

```
CREATE TABLE Car (
   Car_Number VARCHAR(10) PRIMARY KEY,
   Car_Model VARCHAR(20),
   Owner_id INT
);

CREATE TABLE Accidents (
   Accident_id INT PRIMARY KEY,
    Car_Number VARCHAR(10),
    Location VARCHAR(50),
    Date DATE,
    Time TIME,
    FOREIGN KEY (Car_Number) REFERENCES Car(Car_Number)
);
INSERT INTO Car VALUES (‘C101’, ‘Honda’, 1);
INSERT INTO Car VALUES (‘C102’, ‘Toyota’, 2);
INSERT INTO Car VALUES (‘C103’, ‘Ford’, 1);
INSERT INTO Car VALUES (‘C104’, ‘BMW’, 3);

INSERT INTO Accidents VALUES (1, ‘C101’, ‘Delhi’, ‘2023-01-01’, ’10:00:00’);
INSERT INTO Accidents VALUES (2, ‘C102’, ‘Mumbai’, ‘2023-02-15’, ’12:30:00’);
INSERT INTO Accidents VALUES (3, ‘C103’, ‘Pune’, ‘2023-03-05’, ’14:45:00’);
INSERT INTO Accidents VALUES (4, ‘C101’, ‘Delhi’, ‘2023-04-01’, ’16:20:00’);

SELECT C.Owner_id, COUNT(A.Accident_id) AS Number_of_Accidents
FROM Car C
JOIN Accidents A ON C.Car_Number = A.Car_Number
GROUP BY C.Owner_id;
```

Q) Create the following relations. Write queries to display actions performed by commit and rollback.

Company(Company_id, Name, Address)

Customer(Customer_id, Name, Address, phone, Insurance_company)

```
CREATE TABLE Company (
  Company_id INT PRIMARY KEY,
  Name VARCHAR(50),
  Address VARCHAR(100)
);
```

```
CREATE TABLE Customer (
  Customer_id INT PRIMARY KEY,
  Name VARCHAR(50),
  Phone VARCHAR(15),
  Insurance_company INT,
  FOREIGN KEY (Insurance_company) REFERENCES Company(Company_id)
);
```

- Start transaction

```
BEGIN;
```

- Insert a customer

```
INSERT INTO Customer VALUES (1, ‘John Doe’, ‘9876543210’, 101);
```

- Commit the transaction

```
COMMIT;
```

- Start another transaction

```
BEGIN;
```

- Insert a company

```
INSERT INTO Company VALUES (101, ‘LIC’, ‘New Delhi’);
```

- Rollback this insert

```
ROLLBACK;
```