LAB PROGRAM - 2

2. Consider the following schema for Order Database:

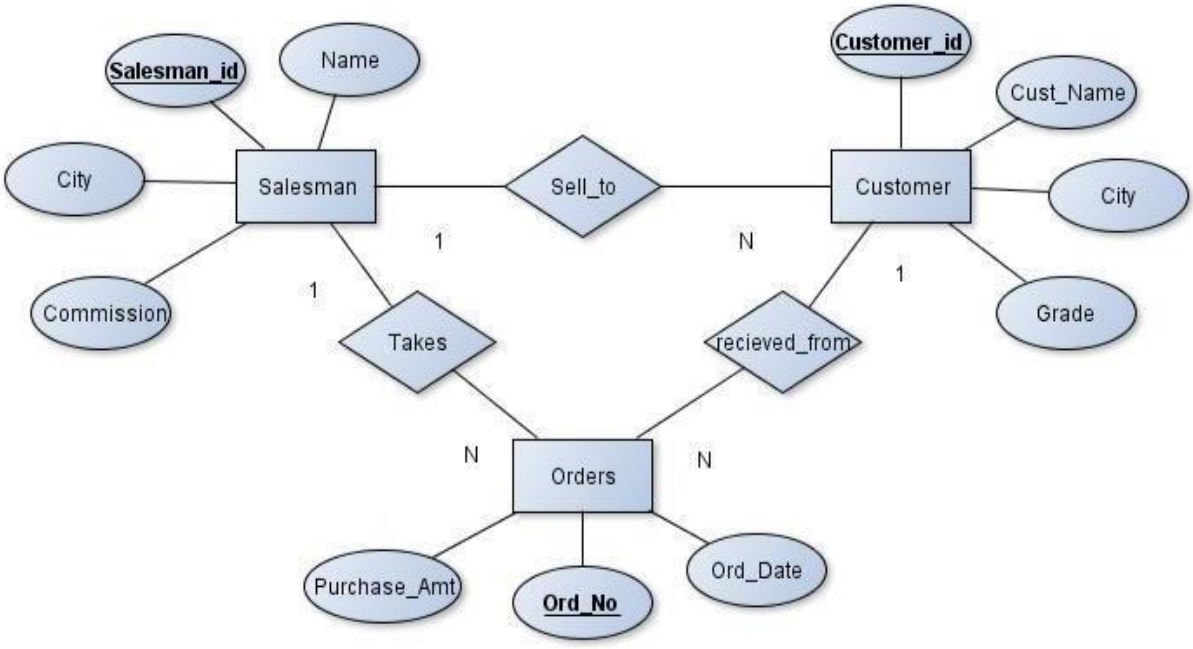
SALESMAN (Salesman_id, Name, City, Commission)
CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id)
ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

Write SQL queries to

1. **Count the customers with grades above Bangalore's average.**
2. **Find the name and numbers of all salesmen who had more than one customer.**
3. **List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)**
4. **Create a view that finds the salesman who has the customer with the highest order of a day.**
5. **Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.**

Solution:

Entity-Relationship Diagram



Schema Diagram

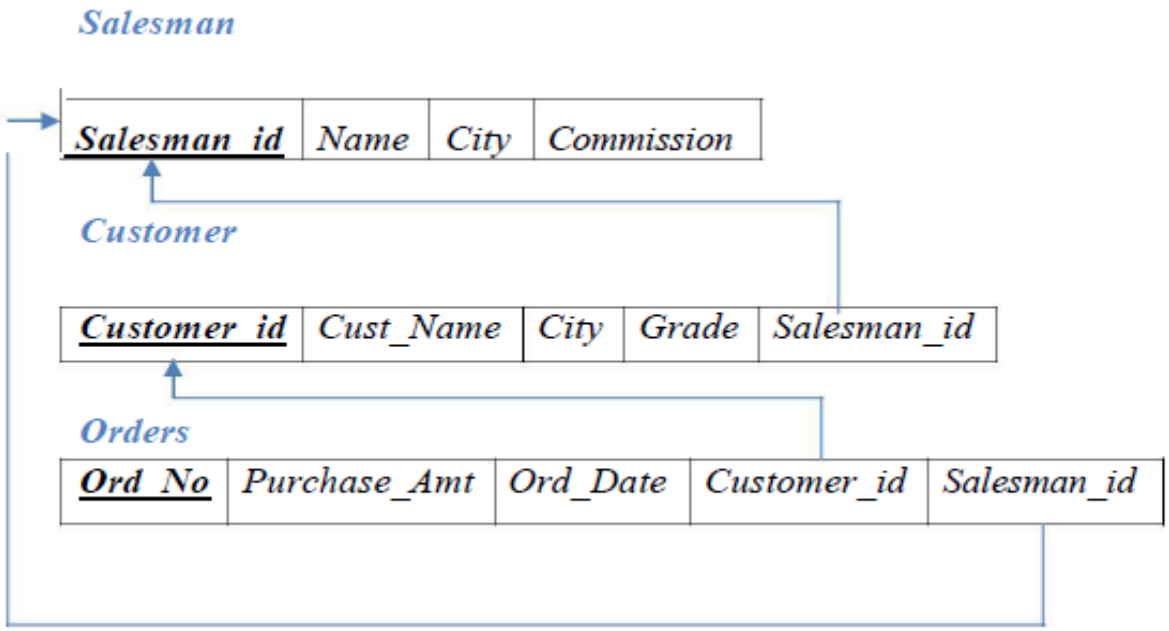


TABLE CREATION

Create Table SALESMAN with Primary Key as SALESMAN_ID

```
CREATE TABLE SALESMAN(
SALESMAN_ID INTEGER PRIMARY KEY,
NAME VARCHAR(20),
CITY VARCHAR(20),
COMMISSION VARCHAR(20));

DESC SALESMAN;
```

Create Table CUSTOMER with Primary Key as CUSTOMER_ID and Foreign Key SALESMAN_ID referring the SALESMAN table

```
CREATE TABLE CUSTOMER(
CUSTOMER_ID INTEGER PRIMARY KEY,
CUST_NAME VARCHAR(20),
CITY VARCHAR(20),
GRADE INTEGER,
SALESMAN_ID INTEGER,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE SET NULL);

DESC CUSTOMER;
```

Create Table ORDERS with Primary Key as ORDER_NO and Foreign Key CUSTOMER_ID and SALESMAN_ID referring the CUSTOMER and SALESMAN tables respectively

```
CREATE TABLE ORDERS(
ORDER_NO INTEGER PRIMARY KEY,
PURCHASE_AMOUNT DECIMAL(10,2),
ORDER_DATE DATE,
CUSTOMER_ID INTEGER,
SALESMAN_ID INTEGER,
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID)ON DELETE CASCADE,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE CASCADE);

DESC ORDERS;
```

INSERTING VALUES

Inserting records into SALESMAN table

```
INSERT INTO SALESMAN VALUES(1000,'RAHUL','BANGALORE','20%');
INSERT INTO SALESMAN VALUES(2000,'ANKITA','BANGALORE','25%');
INSERT INTO SALESMAN VALUES(3000,'SHARMA','MYSORE','30%');
INSERT INTO SALESMAN VALUES(4000,'ANJALI','DELHI','15%');
INSERT INTO SALESMAN VALUES(5000,'RAJ','HYDERABAD','15%');

SELECT * FROM SALESMAN;
```

Inserting records into CUSTOMER table

```
INSERT INTO CUSTOMER VALUES(1,'ADYA','BANGALORE',100,1000);
INSERT INTO CUSTOMER VALUES(2,'BANU','MANGALORE',300,1000);
INSERT INTO CUSTOMER VALUES(3,'CHETHAN','CHENNAI',400,2000);
INSERT INTO CUSTOMER VALUES(4,'DANISH','BANGALORE',200,2000);
INSERT INTO CUSTOMER VALUES(5,'ESHA','BANGALORE',400,3000);

SELECT * FROM CUSTOMER;
```

Inserting records into ORDERS table

```
INSERT INTO ORDERS VALUES(201,5000,'2020-06-02',1,1000);
INSERT INTO ORDERS VALUES(202,450,'2020-04-09',1,2000);
INSERT INTO ORDERS VALUES(203,1000,'2020-03-15',3,2000);
INSERT INTO ORDERS VALUES(204,3500,'2020-07-09',4,3000);
INSERT INTO ORDERS VALUES(205,550,'2020-05-05',2,2000);

SELECT * FROM ORDERS;
```

QUERIES

I) Count the customers with grades above Bangalore's average

```
SELECT GRADE,COUNT(DISTINCT CUSTOMER_ID)
FROM CUSTOMER
GROUP BY GRADE
HAVING GRADE>(SELECT AVG(GRADE)
FROM CUSTOMER
WHERE CITY='BANGALORE');
```

II) Find the name and numbers of all salesman who had more than one customer

```
SELECT SALESMAN_ID, NAME
FROM SALESMAN S
WHERE (SELECT COUNT(*)
FROM CUSTOMER C
WHERE C.SALESMAN_ID=S.SALESMAN_ID) > 1;
```

III) List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.)

```
SELECT S.SALESMAN_ID, S.NAME, C.CUST_NAME, S.COMMISSION
FROM SALESMAN S, CUSTOMER C
WHERE S.CITY=C.CITY
UNION
SELECT S.SALESMAN_ID,S.NAME,'NO MATCH',S.COMMISSION
FROM SALESMAN S
WHERE CITY NOT IN
```

```
(SELECT CITY
FROM CUSTOMER)
ORDER BY 1 ASC;
```

IV)Create a view that finds the salesman who has the customer with the highest order of a day.

```
CREATE VIEW V_SALESMAN AS
SELECT O.ORDER_DATE, S.SALESMAN_ID, S.NAME
FROM SALESMAN S,ORDERS O
WHERE S.SALESMAN_ID = O.SALESMAN_ID
AND O.PURCHASE_AMOUNT= (SELECT MAX(PURCHASE_AMOUNT)
FROM ORDERS C
WHERE C.ORDER_DATE=O.ORDER_DATE);

SELECT * FROM V_SALESMAN;
```

V) Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

```
DELETE FROM SALESMAN
WHERE SALESMAN_ID=1000;

SELECT * FROM SALESMAN;

SELECT * FROM ORDERS;
```