LAB EXERCISE - 2

ORDER DATABASE

Consider the following schema for Order Database:

SALESMAN(<u>Salesman_id</u>, Name, City, Commission)

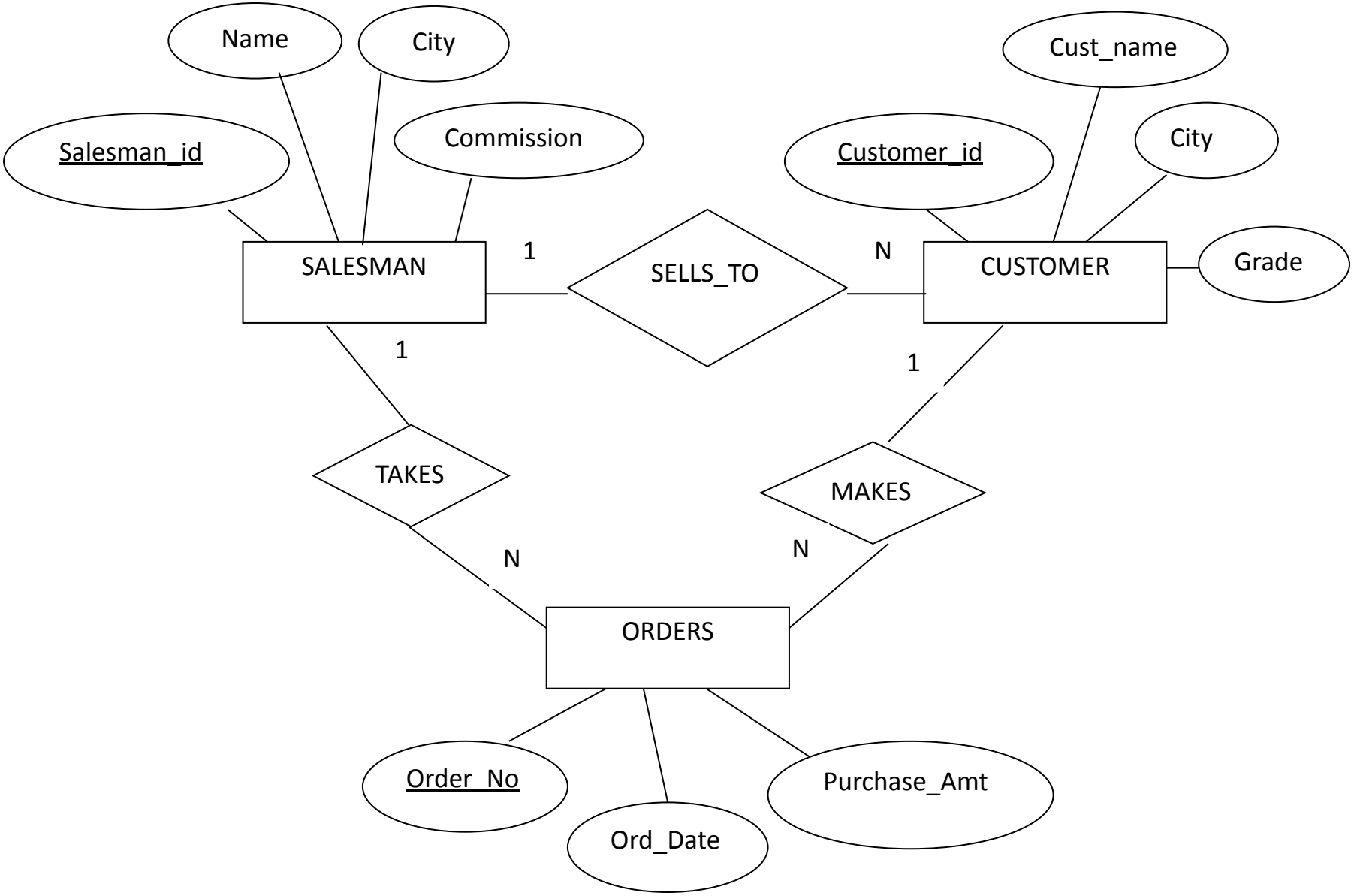
CUSTOMER(<u>Customer_id</u>, Cust_Name, City, Grade, Salesman_id)

ORDERS(<u>Ord_No</u>, 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 salesman who had more than one customer.
3. List all the salesman 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.

ER Diagram


Name
City
Salesman_id
Commission
SALESMAN
1
SELLS_TO
N
CUSTOMER
Cust_name
Customer_id
City
Grade
1
1
TAKES
MAKES
N
N
ORDERS
Order_No
Ord_Date
Purchase_Amt

Schema Diagram

SALESMAN

Salesman_id	Name	City	Commission

CUSTOMER

Customer_id	Cust_Name	City	Grade	Salesman_id

ORDERS

Order_No	Purchase_Amt	Ord_Date	Customer_id	Salesman_id

```
CREATE TABLE SALESMAN
 (
  SALESMAN_ID VARCHAR(20),
  NAME CHAR(15),
  CITY CHAR(15),
  COMMISSION INT,
  PRIMARY KEY(SALESMAN_ID)
 );
```

- ON **DELETE CASCADE** :
  - **SQL** Server deletes the rows in the child table that is corresponding to the row **deleted** from the parent table
- ON **DELETE** SET NULL :
  - **SQL** Server sets the rows in the child table to NULL if the corresponding rows in the parent table are **deleted**.

```
CREATE TABLE CUSTOMER
  (
   CUSTOMER_ID VARCHAR(20),
   CUST_NAME CHAR(15),
   CITY CHAR(15),
   GRADE FLOAT,
   SALESMAN_ID VARCHAR(20),
   PRIMARY KEY(CUSTOMER_ID),
   FOREIGN KEY(SALESMAN_ID)
   REFERENCES    SALESMAN(SALESMAN_ID)
   ON DELETE CASCADE
 );
```

```
CREATE TABLE CUSTOMER1
  (
   CUSTOMER_ID VARCHAR(20),
   CUST_NAME CHAR(15),
   CITY CHAR(15),
   GRADE FLOAT,
   SALESMAN_ID VARCHAR(20),
   PRIMARY KEY(CUSTOMER_ID),
   FOREIGN KEY(SALESMAN_ID)
   REFERENCES    SALESMAN(SALESMAN_ID)
   ON DELETE SET NULL
 );
```

```
CREATE TABLE ORDERS
 (
   ORDER_NO VARCHAR(15),
   PURCHSE_AMT INT,
   ORD_DATE DATE,
   CUSTOMER_ID VARCHAR(20),
   SALESMAN_ID VARCHAR(20),
   PRIMARY KEY(ORDER_NO),
   FOREIGN KEY(CUSTOMER_ID)
   REFERENCES CUSTOMER1(CUSTOMER_ID)
   ON DELETE CASCADE,
   FOREIGN KEY(SALESMAN_ID)
   REFERENCES SALESMAN(SALESMAN_ID)
   ON DELETE CASCADE
 );
```

- INSERT INTO SALESMAN VALUES ('&SALESMAN_ID','&NAME','&CITY','&COMMISSION');
- INSERT INTO CUSTOMER VALUES ('&CUSTOMER_ID','&CUST_NAME','&CITY','&GRADE','&SALESMAN_ID');
- INSERT INTO ORDERS VALUES ('&ORDER_NO','&PURCHASE_AMT','&ORD_DATE','&CUSTOMER_ID','&SALESMAN_ID');

- INSERT INTO SALESMAN VALUES ('1000', 'RAVI', 'BANGALORE', 12);
- INSERT INTO SALESMAN VALUES ('1001', 'SURAJ','DELHI', 20);
- INSERT INTO SALESMAN VALUES ('1002', 'PREM', 'LUCKNOW', 15);
- INSERT INTO SALESMAN VALUES ('1003', 'JOHN', 'BANGALORE', 20);
- INSERT INTO SALESMAN VALUES ('1004', 'RAJU', 'MYSORE', 18);

```
SQL> SELECT *FROM SALESMAN;
```

SALESMAN_ID	NAME	CITY	COMMISSION
1000	RAVI	BANGALORE	12
1001	SOORAJ	DELHI	20
1002	PREM	LUCKNOW	15
1003	JOHN	BANGALORE	20
1004	RAJU	MYSORE	18

- INSERT INTO CUSTOMER VALUES ('C1','SHERYL','BANGALORE',4.5,'1000');
- INSERT INTO CUSTOMER VALUES ('C2','DIYA','DELHI',5,'1000');
- INSERT INTO CUSTOMER VALUES ('C3','PRIYA','MUMBAI',5.5,'1001');
- INSERT INTO CUSTOMER VALUES ('C4','JACK','LUCKNOW',4,'1002');
- INSERT INTO CUSTOMER VALUES ('C5','JILL','BANGALORE',9,'1003');

```
SQL> SELECT *FROM CUSTOMER;
```

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
C1	SHERYL	BANGALORE	4.5	1000
C2	DIYA	DELHI	5	1000
C3	PRIYA	MUMBAI	5.5	1001
C4	JACK	LUCKNOW	4	1002
C5	JILL	BANGALORE	9	1003

- INSERT INTO ORDERS VALUES ('OR1',25000,'25-MAY-2017','C1','1000');
- INSERT INTO ORDERS VALUES ('OR2',15000,'25-MAY-2017','C2','1000');
- INSERT INTO ORDERS VALUES ('OR3',17000,'25-MAY-2017','C5','1003');
- INSERT INTO ORDERS VALUES ('OR4',30000,'17-FEB-2017','C4','1002');
- INSERT INTO ORDERS VALUES ('OR5',32000,'17-FEB-2017','C3','1001');
- INSERT INTO ORDERS VALUES ('OR6',14000,'05-JUN-2017','C1','1000');
- INSERT INTO ORDERS VALUES ('OR7',50000,'10-JUL-2017','C1','1000');

```
SQL> SELECT *FROM ORDERS;
```

ORDER_NO	PURCHSE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
OR1	25000	25-MAY-17	C1	1000
OR2	15000	25-MAY-17	C2	1000
OR3	17000	25-MAY-17	C5	1003
OR4	30000	17-FEB-17	C4	1002
OR5	32000	17-FEB-17	C3	1001
OR6	14000	05-JUN-17	C1	1000
OR7	50000	10-JUL-17	C1	1000

SALESMAN

SALESMAN_ID	NAME	CITY	COMMISSION
1000	RAVI	BANGALORE	12
1001	SOORAJ	DELHI	20
1002	PREM	LUCKNOW	15
1003	JOHN	BANGALORE	20
1004	RAJU	MYSORE	18

CUSTOMER

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
C1	SHERYL	BANGALORE	4.5	1000
C2	DIYA	DELHI	5	1000
C3	PRIYA	MUMBAI	5.5	1001
C4	JACK	LUCKNOW	4.5	1002
C5	JILL	BANGALORE	9	1003

ORDERS

ORDER_NO	PURCHSE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
OR1	25000	25-MAY-17	C1	1000
OR2	15000	25-MAY-17	C2	1000
OR3	17000	25-MAY-17	C5	1003
OR4	30000	17-FEB-17	C4	1002
OR5	32000	17-FEB-17	C3	1001
OR6	14000	05-JUN-17	C1	1000
OR7	50000	10-JUL-17	C1	1000

1. Count the customers with grades above Bangalore's average.

```
SQL>SELECT COUNT (CUSTOMER_ID)
  FROM CUSTOMER
  WHERE GRADE >
   ( SELECT AVG(GRADE)
  FROM CUSTOMER
  WHERE CITY='BANGALORE' );
```

```
COUNT(CUSTOMER_ID)
------------------
                 1
```

2. Find the name and numbers of all salesman who had more than one customer.

- **SQL**>SELECT SALESMAN_ID, NAME
  FROM SALESMAN
  WHERE SALESMAN_ID IN
  (SELECT SALESMAN_ID
  FROM CUSTOMER
  GROUP BY SALESMAN_ID
  HAVING COUNT(SALESMAN_ID) > 1);

```
SALESMAN_ID          NAME
-------------------- ----------
1000                 RAVI
```

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

- **SQL>** (SELECT A.SALESMAN_ID, A.NAME, A.CITY
  FROM SALESMAN A, CUSTOMER B
  WHERE A.SALESMAN_ID = B.SALESMAN_ID AND A.CITY=B.CITY)
  UNION
  (SELECT A.SALESMAN_ID, A.NAME, A.CITY
  FROM SALESMAN A, CUSTOMER B
  WHERE A.SALESMAN_ID = B.SALESMAN_ID AND A.CITY != B.CITY);

SALESMAN_ID	NAME	CITY
1000	RAVI	BANGALORE
1001	SURAJ	DELHI
1002	PREM	LUCKNOW
1003	JOHN	BANGALORE

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

SQL>

```
CREATE VIEW MAX_ORDERS AS
SELECT S.SALESMAN_ID, S.NAME, O.ORD_DATE
FROM SALESMAN S, ORDERS O
 WHERE S.SALESMAN_ID=O.SALESMAN_ID AND
PURCHASE_AMT =
(SELECT MAX(PURCHASE_AMT)
 FROM ORDERS O1
 WHERE O1.ORD_DATE = O.ORD_DATE);
```

View created.

```
SQL> SELECT *FROM MAX_ORDERS;

SALESMAN_ID          NAME                 ORD_DATE
-------------------- -------------------- ---------
1000                 RAVI                 25-MAY-17
1001                 SURAJ                17-FEB-17
1000                 RAVI                 05-JUN-17
1000                 RAVI                 10-JUL-17
```

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

DELETE FROM SALESMAN WHERE SALESMAN_ID=1000;

```
1 row deleted.

SQL> SELECT *FROM SALESMAN
  2  ;

SALESMAN_ID          NAME                 CITY                 COMMISSION
-------------------- -------------------- -------------------- ----------
1001                 SURAJ                DELHI                        20
1002                 PREM                 LUCKNOW                      15
1003                 JOHN                 BANGALORE                    20
1004                 RAJU                 MYSORE                       18
```