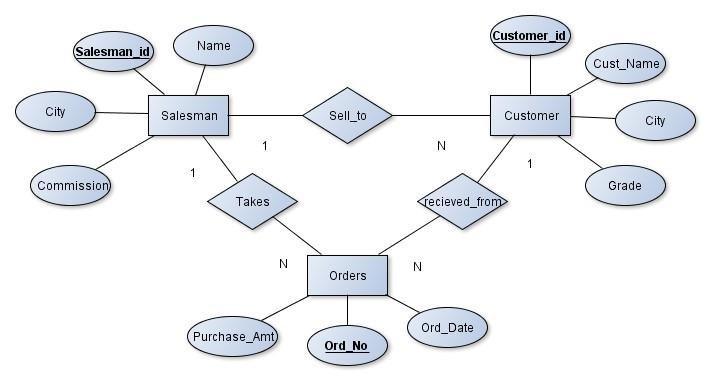
# 2.Consider the following schema for Order Database:



**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**

# Count the customers with grades above Bangalore’s average.

1. **Find the name and numbers of all salesmen who had more than one customer.**

# List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)

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

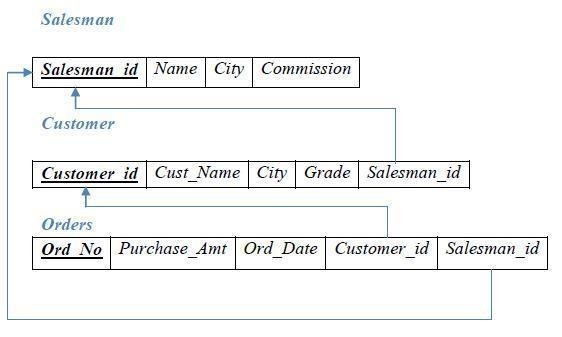
# Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

**Solution:**

**Entity-Relationship Diagram**

**Schema Diagram**

Table Creation



CREATE TABLE SALESMAN (SALESMAN\_ID NUMBER (4), NAME VARCHAR2 (20),

CITY VARCHAR2 (20),

COMMISSION VARCHAR2 (20), PRIMARY KEY(SALESMAN\_ID));

CREATE TABLE CUSTOMER1 (CUSTOMER\_ID NUMBER (4), CUST\_NAME VARCHAR2 (20),

CITY VARCHAR2 (20),

GRADE NUMBER (3),

SALESMAN\_ID NUMBER (4), PRIMARY KEY (CUSTOMER\_ID),

FOREIGN KEY(SALESMAN\_ID) REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE SET NULL);

CREATE TABLE ORDERS (ORD\_NO NUMBER (5), PURCHASE\_AMT NUMBER (10, 2),

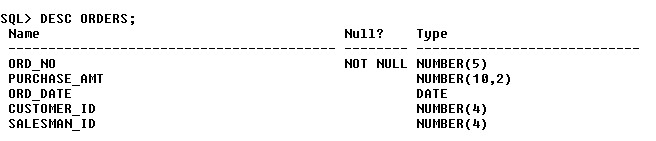
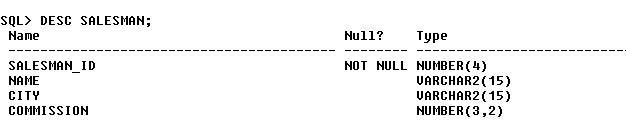
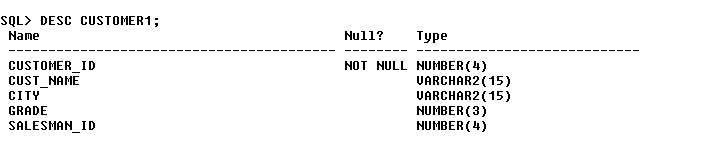
ORD\_DATE DATE, CUSTOMER\_ID NUMBER (4),

SALESMAN\_ID NUMBER (4), PRIMARY KEY (ORD\_NO),

CUSTOMER\_ID REFERENCES CUSTOMER1 (CUSTOMER\_ID) ON DELETE CASCADE, SALESMAN\_ID REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE CASCADE);

**Table Descriptions**

DESC SALESMAN;



DESC CUSTOMER1;

DESC ORDERS;

**Insertion of Values to Tables**

INSERT INTO SALESMAN VALUES (1000, ‘JOHN’,’BANGALORE’,’25 %’); INSERT INTO SALESMAN VALUES (2000, ‘RAVI’,’BANGALORE’,’20 %’); INSERT INTO SALESMAN VALUES (3000, ‘KUMAR’,’MYSORE’,’15 %’); INSERT INTO SALESMAN VALUES (4000, ‘SMITH’,’DELHI’,’30 %’); INSERT INTO SALESMAN VALUES (5000, ‘HARSHA’,’HYDRABAD’,’15 %’);

INSERT INTO CUSTOMER1 VALUES (10, ‘PREETHI’,’BANGALORE’, 100, 1000);

INSERT INTO CUSTOMER1 VALUES (11, ‘VIVEK’,’MANGALORE’, 300, 1000);

INSERT INTO CUSTOMER1 VALUES (12, ‘BHASKAR’,’CHENNAI’, 400, 2000);

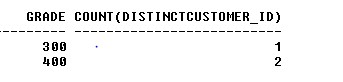
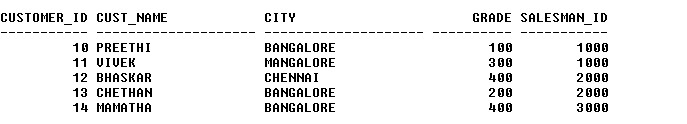
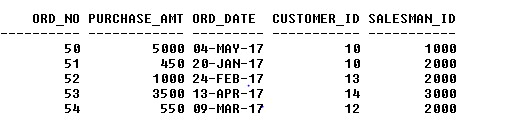
INSERT INTO CUSTOMER1 VALUES (13, ‘CHETHAN’,’BANGALORE’, 200, 2000);

INSERT INTO CUSTOMER1 VALUES (14, ‘MAMATHA’,’BANGALORE’, 400, 3000);

INSERT INTO ORDERS VALUES (50, 5000, ‘04-MAY-17’, 10, 1000);

INSERT INTO ORDERS VALUES (51, 450, ‘20-JAN-17’, 10, 2000);

INSERT INTO ORDERS VALUES (52, 1000, ‘24-FEB-17’, 13, 2000);



INSERT INTO ORDERS VALUES (53, 3500, ‘13-APR-17’, 14, 3000);

INSERT INTO ORDERS VALUES (54, 550, ‘09-MAR-17’, 12, 2000); SELECT \* FROM SALESMAN;

SELECT \* FROM CUSTOMER1;

SELECT \* FROM ORDERS;

**Queries:**

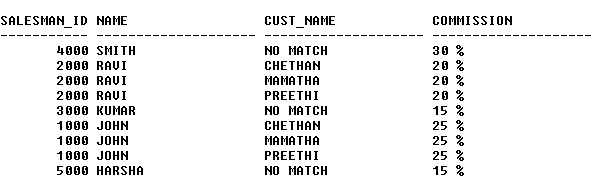
* 1. **Count the customers with grades above Bangalore’s average.** SELECT GRADE, COUNT (DISTINCT CUSTOMER\_ID) FROM CUSTOMER1

GROUP BY GRADE

HAVING GRADE > (SELECT AVG(GRADE) FROM CUSTOMER1

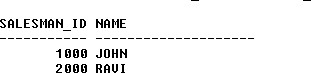
WHERE CITY='BANGALORE');

# Find the name and numbers of all salesmen who had more than one customer.



SELECT SALESMAN\_ID, NAME FROM SALESMAN A

WHERE 1 < (SELECT COUNT (\*) FROM CUSTOMER1



WHERE SALESMAN\_ID=A.SALESMAN\_ID);

# List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)

SELECT SALESMAN.SALESMAN\_ID, NAME, CUST\_NAME, COMMISSION FROM SALESMAN, CUSTOMER1

WHERE SALESMAN.CITY = CUSTOMER1.CITY UNION

SELECT SALESMAN\_ID, NAME, 'NO MATCH', COMMISSION FROM SALESMAN

WHERE NOT CITY = ANY (SELECT CITY

FROM CUSTOMER1) ORDER BY 2 DESC;

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

CREATE VIEW ELITSALESMAN AS

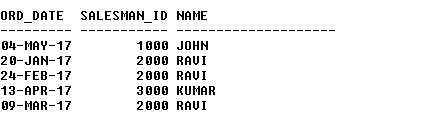
SELECT B.ORD\_DATE, A.SALESMAN\_ID, A.NAME FROM SALESMAN A, ORDERS B

WHERE A.SALESMAN\_ID = B.SALESMAN\_ID

AND B.PURCHASE\_AMT=(SELECT MAX (PURCHASE\_AMT)

FROM ORDERS C

WHERE C.ORD\_DATE = B.ORD\_DATE);



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

Use ON DELETE CASCADE at the end of foreign key definitions while creating child table orders and then execute the following:

Use ON DELETE SET NULL at the end of foreign key definitions while creating child table customers and then executes the following:

DELETE FROM SALESMAN WHERE SALESMAN\_ID=1000;

