

```
CREATE TABLE SALESMAN (SALESMAN_ID NUMBER (4), NAME VARCHAR2 (20), CITY  
VARCHAR2 (20), COMMISSION VARCHAR2 (20), PRIMARY KEY (SALESMAN_ID));
```

```
CREATE TABLE CUSTOMER (CUSTOMER_ID NUMBER (4), CUST_NAME VARCHAR2  
(20), CITY VARCHAR2 (20), GRADE NUMBER (3), PRIMARY KEY (CUSTOMER_ID),  
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, PRIMARY KEY (ORD_NO), CUSTOMER_ID REFERENCES CUSTOMER  
(CUSTOMER_ID) ON DELETE CASCADE, SALESMAN_ID REFERENCES SALESMAN  
(SALESMAN_ID) ON DELETE CASCADE);
```

Insertion of Values to Tables

```
INSERT INTO SALESMAN VALUES (1000, 'GANESH', 'BLR', '30 %');
```

Insert as many as required rows

```
INSERT INTO CUSTOMER VALUES (10, 'PREETHI', 'BANGALORE', 100, 1000);
```

Insert as many as required rows

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

Insert as many as required rows

Queries:

1. 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');
```

2. 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 CUSTOMER  
WHERE SALESMAN_ID=A.SALESMAN_ID);
```

3. 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, CUSTOMER  
WHERE SALESMAN.CITY = CUSTOMER.CITY  
UNION  
SELECT SALESMAN_ID, NAME, 'NO MATCH', COMMISSION  
FROM SALESMAN  
WHERE NOT CITY = ANY  
(SELECT CITY  
FROM CUSTOMER)  
ORDER BY 2 DESC;
```

4. 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);
```

5. 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;
```