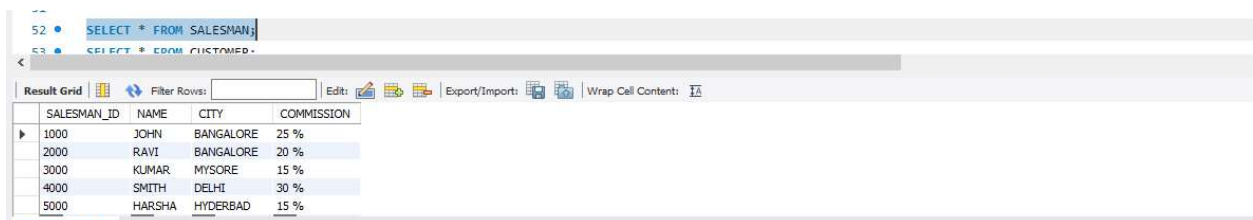**USN: 1BM19CS168**

NAME:SWETHA PATIL

```
CREATE DATABASE ORDER_;
USE ORDER_;

CREATE TABLE SALESMAN
(SALESMAN_ID INT(4),
NAME VARCHAR(20),
CITY VARCHAR(20),
COMMISSION VARCHAR(20),
PRIMARY KEY (SALESMAN_ID));

CREATE TABLE CUSTOMER
(CUSTOMER_ID INT,
CUST_NAME VARCHAR(20),
CITY VARCHAR(20),
GRADE INT,
SALESMAN_ID INT,
PRIMARY KEY (CUSTOMER_ID),
foreign key(SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE SET NULL);
desc customer;


CREATE TABLE ORDERS
(ORD_NO INT(5),
PURCHASE_AMT FLOAT,
ORD_DATE DATE,
CUSTOMER_ID INT,
SALESMAN_ID INT,
PRIMARY KEY (ORD_NO),
foreign key(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID) ON DELETE CASCADE,
foreign key(SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE CASCADE);
desc orders;

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','HYDERBAD','15 %');
SELECT * FROM SALESMAN;
```

INSERT INTO CUSTOMER VALUES (10, 'PREETHI','BANGALORE', 100, 1000);
INSERT INTO CUSTOMER VALUES (11, 'VIVEK','MANGALORE', 300, 1000);
INSERT INTO CUSTOMER VALUES (12, 'BHASKAR','CHENNAI', 400, 2000);
INSERT INTO CUSTOMER VALUES (13, 'CHETHAN','BANGALORE', 200, 2000);
INSERT INTO CUSTOMER VALUES (14, 'MAMATHA','BANGALORE', 400, 3000);
SELECT * FROM CUSTOMER;

INSERT INTO ORDERS VALUES (50, 5000, '2017-05-04', 10, 1000);
INSERT INTO ORDERS VALUES (51, 450, '2017-01-20', 10, 2000);
INSERT INTO ORDERS VALUES (52, 1000, '2017-02-24', 13, 2000);
INSERT INTO ORDERS VALUES (53, 3500, '2017-04-13', 14, 3000);
INSERT INTO ORDERS VALUES (54, 550, '2017-03-09', 12, 2000);
SELECT * FROM ORDERS;

-- 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');

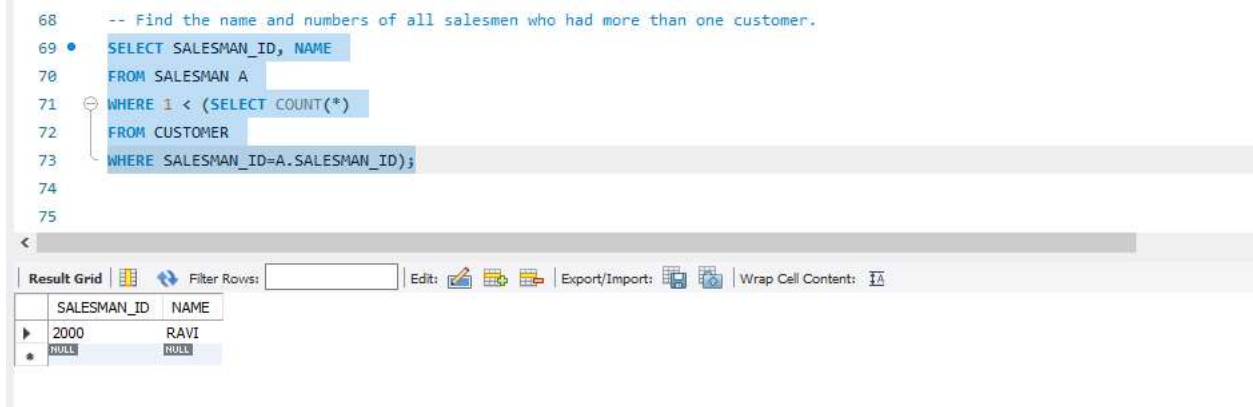-- Find the name and numbers of all salesmen who had more than one customer.
SELECT SALESMAN_ID, NAME
FROM SALESMAN A
WHERE 1 < (SELECT COUNT(*)
FROM CUSTOMER
WHERE SALESMAN_ID=A.SALESMAN_ID);

```
68      -- Find the name and numbers of all salesmen who had more than one customer.
69 •    SELECT SALESMAN_ID, NAME
70      FROM SALESMAN A
71 ⊝    WHERE 1 < (SELECT COUNT(*)
72      FROM CUSTOMER
73      WHERE SALESMAN_ID=A.SALESMAN_ID);
74
75
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

SALESMAN_ID	NAME
2000	RAVI
NULL	NULL

-- 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;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

SALESMAN_ID	NAME	CUST_NAME	COMMISSION
4000	SMITH	NO MATCH	30 %
2000	RAVI	PREETHI	20 %
2000	RAVI	CHETHAN	20 %
2000	RAVI	MAMATHA	20 %
3000	KUMAR	NO MATCH	15 %
1000	JOHN	PREETHI	25 %
1000	JOHN	CHETHAN	25 %
1000	JOHN	MAMATHA	25 %
5000	HARSHA	NO MATCH	15 %

Result 19 ⌄

-- 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);
select * from elitsalesman;

```
87  •   CREATE VIEW ELITSALESMAN AS
88      SELECT B.ORD_DATE, A.SALESMAN_ID, A.NAME
89      FROM SALESMAN A, ORDERS B
90      WHERE A.SALESMAN_ID = B.SALESMAN_ID
91  ⊖   AND B.PURCHASE_AMT=(SELECT MAX(PURCHASE_AMT)
92      FROM ORDERS C
93      WHERE C.ORD_DATE = B.ORD_DATE);
94  •   select * from elitsalesman;
95
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: TA

ORD_DATE	SALESMAN_ID	NAME
2017-05-04	1000	JOHN
2017-01-20	2000	RAVI
2017-02-24	2000	RAVI
2017-04-13	3000	KUMAR
2017-03-09	2000	RAVI

-- 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; -- deleted 1st row.

```
99      -- Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.
100 •   DELETE FROM SALESMAN
101     WHERE SALESMAN_ID=1000;
102 •   select * from salesman;  -- deleted 1st row.
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: TA

SALESMAN_ID	NAME	CITY	COMMISSION
2000	RAVI	BANGALORE	20 %
3000	KUMAR	MYSORE	15 %
4000	SMITH	DELHI	30 %
5000	HARSHA	HYDERBAD	15 %
NULL	NULL	NULL	NULL