Program 6: Order Database

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)

```
CREATE DATABASE order_db;
CREATE TABLE SALESMAN
(SALESMAN_ID int,
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 (3),
SALESMAN_ID int,
PRIMARY KEY (CUSTOMER_ID),
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE SET NULL ON
UPDATE CASCADE);

CREATE TABLE ORDERS
(ORD_NO INT,
PURCHASE_AMT INT,
ORD_DATE DATE,
CUSTOMER_ID INT,
SALESMAN_ID INT,
PRIMARY KEY (ORD_NO),
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER (CUSTOMER_ID) ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE CASCADE ON
UPDATE CASCADE);
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0444 seconds.)

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

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0425 seconds.)

```
CREATE TABLE CUSTOMER (CUSTOMER_ID INT, CUST_NAME VARCHAR (20), CITY VARCHAR (20), GRADE INT (3), SALESMAN_ID int, PRIMARY KEY (CUSTOMER_ID), FOREIGN KEY
(SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE SET NULL ON UPDATE CASCADE)
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0559 seconds.)

```
CREATE TABLE ORDERS (ORD_NO INT, PURCHASE_AMT INT, ORD_DATE DATE, CUSTOMER_ID INT, SALESMAN_ID INT, PRIMARY KEY (ORD_NO), FOREIGN KEY (CUSTOMER_ID) REFERENCES
CUSTOMER (CUSTOMER_ID) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE CASCADE ON UPDATE CASCADE)
```

[Edit inline] [Edit] [Create PHP code]

```
INSERT INTO SALESMAN VALUES (1, 'JASON','BANGALORE','25 %'),
 (2, 'SHYAM','BANGALORE','20 %'),
 (3, 'KUMARA','MYSORE','15 %'),
 (4, 'SMITHA','DELHI','30 %'),
 (5, 'HARISH','HYDERABAD','15%');

 INSERT INTO CUSTOMER VALUES (10, 'PREETHI','BANGALORE', 100, 1),
(11, 'VIVEK','MANGALORE', 300, 1),
(12, 'BHASKAR','CHENNAI', 400, 2),
(13, 'CHETAN','BANGALORE', 200, 2),
(14, 'MAMATHA','BANGALORE', 400, 3),
(15,'ANJALI','UDUPI',600,2);

INSERT INTO ORDERS VALUES (50, 5000, '2017-04-04', 10, 1),
(51, 450, '20-01-17', 10, 2),
(52,1000,'24-02-17',13,2),
(53,3500,'13-04-17',14,3),
(54, 550, '09-03-17', 12, 2);
```

Write SQL queries to

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

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

COUNT(DISTINCT CUSTOMER_ID)
4

2. Find the name and numbers of all salesmen 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;
```

+ Options

←T→				SALESMAN_ID	NAME
☐	Edit	Copy	Delete	1	JASON
☐	Edit	Copy	Delete	2	SHYAM

3. List all salesmen 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;
```

SALESMAN_ID	NAME	CUST_NAME	COMMISSION
1	JASON	PREETHI	25 %
1	JASON	CHETAN	25 %
1	JASON	MAMATHA	25 %
2	SHYAM	PREETHI	20 %
2	SHYAM	CHETAN	20 %
2	SHYAM	MAMATHA	20 %
3	KUMARA	NO MATCH	15 %
4	SMITHA	NO MATCH	30 %
5	HARISH	NO MATCH	15%

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

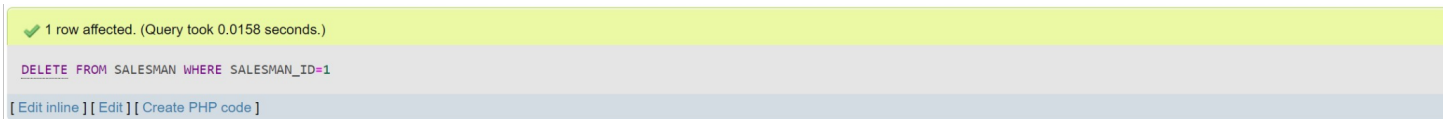
```
CREATE VIEW V_SALESMAN AS
SELECT O.ORD_DATE, S.SALESMAN_ID, S.NAME FROM SALESMAN S,ORDERS O
WHERE S.SALESMAN_ID = O.SALESMAN_ID
AND O.PURCHASE_AMT= (SELECT MAX(PURCHASE_AMT) FROM ORDERS C
                    WHERE C.ORD_DATE=O.ORD_DATE);
```

+ Options

				ORD_DATE	SALESMAN_ID	NAME
☐	Edit	Copy	Delete	2017-04-04	1	JASON
☐	Edit	Copy	Delete	2020-01-17	2	SHYAM
☐	Edit	Copy	Delete	2024-02-17	2	SHYAM
☐	Edit	Copy	Delete	2013-04-17	3	KUMARA
☐	Edit	Copy	Delete	2009-03-17	2	SHYAM

5. Demonstrate the DELETE operation by removing salesman with id 1. All his orders must also be deleted.

```
DELETE FROM SALESMAN
WHERE SALESMAN_ID=1;
```



1 row affected. (Query took 0.0158 seconds.)

```
DELETE FROM SALESMAN WHERE SALESMAN_ID=1
```

[Edit inline] [Edit] [Create PHP code]