KIET GROUP OF INSTITUTIONS

DEPARTMENT OF COMPUTER APPLICATIONS

LAB ASSIGNMENT 6

DBMS Lab (KCA – 252)

Assignments on Join

Questions from 1 to 14 refer the sample tables Salesman, Customer, Order.

Sample table: salesman

Salesman_id	Name	City	Commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

```
CREATE TABLE SALESMAN
(
 SALESMAN_ID INT(8) PRIMARY KEY,
 NAME VARCHAR(20),
 CITY VARCHAR(20),
 COMMISSION DOUBLE );

INSERT INTO SALESMAN VALUES(5001,'JAMES HOOG','NEW YORK',0.15);
 INSERT INTO SALESMAN VALUES(5002,'NAIL KNITE','PARIS',0.13);
 INSERT INTO SALESMAN VALUES(5005,'PIT ALEX','LONDON',0.11);
 INSERT INTO SALESMAN VALUES(5006,'MC LYON','PARIS',0.14);
 INSERT INTO SALESMAN VALUES(5007,'PAUL ADAM','ROME',0.13);
 INSERT INTO SALESMAN VALUES(5003,'LAUSON HEN','SAN JOSE',0.12);
```

Sample table: customer

Customer_id	Cust_name	City	Grade
3002	Nick Rimando	New York	100
3007	Brad Davis	New York	200
3005	Graham Zusi	California	200
3008	Julian Green	London	300
3004	Fabian Johnson	Paris	300
3009	Geoff Cameron	Berlin	100
3003	Jozy Altidor	Moscow	200
3001	Brad Guzan	London	100

```
 CREATE TABLE CUSTOMER(
```

```
CUSTOMER_ID INT(8) PRIMARY KEY,
CUST_NAME VARCHAR(20),
CITY VARCHAR(20),
GRADE INT(8));

INSERT INTO CUSTOMER VALUES(3002,'NICK RIMANDO','NEW YORK',100);
INSERT INTO CUSTOMER VALUES(3007,'BRAD DAVIS','NEW YORK',200);
INSERT INTO CUSTOMER VALUES(3005,'GRAHAM ZUSI','CALIFORNIA',200);
INSERT INTO CUSTOMER VALUES(3008,'JULIAN GREEN ','LONDON',300);
INSERT INTO CUSTOMER VALUES(3004,'FABIAN JOHNSON','PARIS',300);
INSERT INTO CUSTOMER VALUES(3009,'GEOFF CAMERON','BERLIN',100);
INSERT INTO CUSTOMER VALUES(3003,'JOZY ALTIDOR','MOSCOW',200);
INSERT INTO CUSTOMER VALUES(3001,'BRAD GUZAN','LONDON',100);
```

***Sample table*: orders**

Ord_no	Purch_amt	Ord_date	Customer_id	Salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

```
CREATE TABLE ORDERS(
 ORD_NO INT(10),
 PURCH_AMT DOUBLE(7,2),
 ORD_DATE DATE,
 CUSTOMER_ID INT(8), FOREIGN KEY(CUSTOMER_ID)
REFERENCES CUSTOMER (CUSTOMER_ID),
 SALESMAN_ID INT(8), FOREIGN KEY (SALESMAN_ID)
REFERENCES SALESMAN(SALESMAN_ID));


INSERT INTO ORDERS VALUES(70001,150.5,'2012-10-
05',3005,5002);
 INSERT INTO ORDERS VALUES(70009,270.65,'2012-09-
10',3001,5005);
 INSERT INTO ORDERS VALUES(70002,65.26,'2012-10-
05',3002,5001);
```

INSERT INTO ORDERS VALUES(70004,110.5,'2012-08-17',3009,5003);
INSERT INTO ORDERS VALUES(70007,948.5,'2012-09-10',3005,5002);
INSERT INTO ORDERS VALUES(70005,2400.6,'2012-07-27',3007,5001);
INSERT INTO ORDERS VALUES(70008,5760,'2012-09-10',3002,5001);
INSERT INTO ORDERS VALUES(70010,1983.43,'2012-10-10',3009,5003);
INSERT INTO ORDERS VALUES(70003,2480.4,'2012-10-10',3004,5006);
INSERT INTO ORDERS VALUES(70012,250.45,'2012-06-27',3008,5002);
INSERT INTO ORDERS VALUES(70011,75.29,'2012-08-17',3003,5007);
INSERT INTO ORDERS VALUES(70013,3045.6,'2012-04-25',3002,5001);

1. Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

   SELECT SALESMAN.NAME, CUSTOMER.CUST_NAME,CITY FROM SALESMAN JOIN CUSTOMER USING(CITY) WHERE CUSTOMER.CITY=SALESMAN.CITY;

2. Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.
   SELECT ORDERS.ORD_NO, ORDERS.PURCH_AMT, CUSTOMER.CUST_NAME,CUSTOMER.CITY FROM CUSTOMER JOIN ORDERS USING(CUSTOMER_ID)WHERE ORDERS.PURCH_AMT BETWEEN 500 AND 2000;

3. Write a SQL statement to know which salesman are working for which customer.
   SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME,'HANDLE BY',SALESMAN.NAME AS SALESMAN_NAME FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING(SALESMAN_ID)) USING (CUSTOMER_ID);

4. Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.
   SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME ,'APPOINTED', SALESMAN.NAME AS SALESMAN_NAME FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID) WHERE SALESMAN.COMMISSION>0.12;

5. Write a SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in the same city where their customer lives, and gets a commission is above 12%.

   SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME ,'APPOINTED', SALESMAN.NAME AS SALESMAN_NAME FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID) WHERE CUSTOMER.CITY NOT IN (SALESMAN.CITY) AND SALESMAN.COMMISSION>0.12;

6. Write a SQL statement to find the details of a order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.

   SELECT DISTINCT ORDERS.ORD_NO,ORDERS.ORD_DATE,ORDERS.PURCH_AMT,CUSTOMER.CUST_NAME AS CUSTOMER_NAME , 'APPOINTED', SALESMAN.NAME AS SALESMAN_NAME,SALESMAN.COMMISSION FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID);

7. Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.

   SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME ,'APPOINTED', SALESMAN.NAME AS SALESMAN_NAME FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID) ORDER BY CUSTOMER.CUST_NAME ASC;

8. Write a SQL statement to make a list in ascending order for the customer who holds a grade less than 300 and works either through a salesman or by own.

   SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME ,'APPOINTED', SALESMAN.NAME AS SALESMAN_NAME FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID) WHERE CUSTOMER.GRADE<300 ORDER BY CUSTOMER.CUST_NAME ASC;

9. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.

   SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME ,CUSTOMER.CITY, ORDERS.ORD_NO,ORDERS.ORD_DATE,ORDERS.PURCH_AMT FROM CUSTOMER JOIN(SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID) ORDER BY ORDERS.ORD_DATE ASC;

10. Write a SQL statement to make a report with customer name, city, order number, order date, order amount salesman name and commission to find that either any of the existing customers have placed no order or placed one or more orders by their salesman or by own.

SELECT DISTINCT CUSTOMER.CUST_NAME AS CUSTOMER_NAME ,CUSTOMER.CITY,ORDERS.ORD_NO,ORDERS.ORD_DATE,ORDERS.PURCH_AMT, SALESMAN.NAME AS SALESMAN_NAME,SALESMAN.COMMISSION FROM CUSTOMER JOIN (SALESMAN JOIN ORDERS USING (SALESMAN_ID))USING (CUSTOMER_ID);