CREATE DATABASE INVENTORY;

USE INVENTORY;

------TABLE1----SALES--MAN-------

CREATE TABLE SALESMAN

(

SALESMAN\_ID INT NOT NULL CONSTRAINT PK\_SALES\_ID

PRIMARY KEY,

NAME VARCHAR(30) NULL,

CITY VARCHAR(20) NULL,

COMMISSION FLOAT NULL

);

------TABLE2----CUSTOMER-------

CREATE TABLE CUSTOMER

(

CUSTOMER\_ID INT NOT NULL CONSTRAINT PK\_CUSTOMER\_ID PRIMARY KEY,

CUST\_NAME VARCHAR(30) NULL,

CITY VARCHAR(20) NULL,

GRADE NUMERIC(3) NOT NULL CONSTRAINT

CK\_CUSTOMER\_GRADE

CHECK(GRADE=100 OR GRADE=200 OR GRADE=300),

SALESMAN\_ID INT NULL CONSTRAINT FK\_CUSTOMER\_SALESMAN\_ID REFERENCES SALESMAN(SALESMAN\_ID)

);

------TABLE3----ORDERS-------

CREATE TABLE ORDERS

(

ORD\_NO INT NOT NULL CONSTRAINT PK\_ORD\_NO PRIMARY KEY,

PURCH\_AMT FLOAT NOT NULL,

ORD\_DATE DATE NULL,

CUSTOMER\_ID INT NULL CONSTRAINT FK\_ORDERS\_CUSTOMER\_ID REFERENCES CUSTOMER(CUSTOMER\_ID),

SALESMAN\_ID INT NULL CONSTRAINT FK\_ORDERS\_SALESMAN\_ID REFERENCES SALESMAN(SALESMAN\_ID)

);

INSERTION OF VALUES;

-------INSERTION OF DATA IN SALESMAN TABLE-------------------

INSERT INTO SALESMAN

VALUES

(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)

-------INSERTION OF DATA IN CUSTOMER TABLE-------------------

INSERT INTO CUSTOMER

VALUES

(3002,'Nick Rimando','New York',100,5001),

(3007,'Brad Davis','New York',200,5001),

(3005,'Graham Zusi','California',200,5002),

(3008,'Julian Green','London',300,5002),

(3004,'Fabian Johnson','Paris',300,5006),

(3009,'Geoff Cameron','Berlin',100,5003),

(3003,'Jozy Altidor','Moscow',200,5007),

(3001,'Brad Guzan','London',100,5005)

-------INSERTION OF DATA IN ORDERS TABLE-------------------

INSERT INTO ORDERS

VALUES

(70009,270.65,'2012-09-10',3001,5005),

(70002,65.26,'2012-10-05',3002,5001),

(70004,110.50,'2012-08-17',3009,5003),

(70005,2400.60,'2012-07-27',3007,5001),

(70008,5760.00,'2012-09-10',3002,5001),

(70010,1983.43,'2012-10-10',3004,5006),

(70003,2480.40,'2012-08-17',3009,5003),

(70011,75.29,'2012-04-25',3003,5007),

(70013,3045.60,'2012-10-10',3002,5001),

(70001,150.50,'2012-10-05',3005,5002),

(70007,948.50,'2012-09-10',3005,5002),

(70012,250.45,'2012-06-27',3008,5002)

--------------------------X--------------------

---------QUERIES----------

1. Write a SQL statement to display all the information of all salesmen.

SELECT \* FROM SALESMAN;

2.Write a SQL statement to display a string "This is SQL Exercise, Practice and Solution".

SELECT 'THIS IS A SQL EXERCISE,PRACTICE AND SOLUTION';

3. Write a SQL statement to display specific columns like name and commission for all the salesmen.

SELECT NAME,COMMISSION FROM SALESMAN;

4. Write a query to display the columns in a specific order like order date,

salesman id, order number and purchase amount from for all the orders.

SELECT ORD\_DATE,SALESMAN\_ID,ORD\_NO,PURCH\_AMT

FROM ORDERS;

5. Write a query which will retrieve the value of salesman id of all salesmen, getting orders from

the customers in orders table without any repeats.

SELECT DISTINCT(SALESMAN\_ID) FROM ORDERS;

6. Write a SQL statement to display names and city of salesman, who belongs to the city of Paris

SELECT NAME,CITY

FROM SALESMAN

WHERE CITY='PARIS';

7.Write a SQL statement to display all the information for those customers with a grade of 200.

SELECT \*

FROM CUSTOMER

WHERE GRADE=200;

8.Write a SQL query to display the order number followed by order date and the purchase amount for

each order which will be delivered by the salesman who is holding the ID 5001.

SELECT ORD\_NO,ORD\_DATE,PURCH\_AMT

FROM ORDERS

WHERE SALESMAN\_ID=5001;

9.Show me the Result if there is a Increment in commssion by 20%,25%,30%?

SELECT \*,COMMISSION \* 120/100 AS "20% Inc",COMMISSION \* 125/100 AS "25% Inc",

COMMISSION \* 130/100 AS "30% Inc"

FROM SALESMAN;

10. GET SALESMAN ID,CUSTOMER NAME,CUSTOMER CITY,ORDER NUMBER,PURCHASE AMOUNT DETAIL FROM THE TABLES.

SELECT S.SALESMAN\_ID,C.CUST\_NAME,C.CITY,O.ORD\_NO,O.PURCH\_AMT

FROM SALESMAN AS "S" INNER JOIN CUSTOMER AS "C"

ON S.SALESMAN\_ID=C.SALESMAN\_ID

INNER JOIN ORDERS AS "O"

ON

C.CUSTOMER\_ID=O.CUSTOMER\_ID;

11.Update the purchase amount of order number 70003 to 600 and also for order number 70004 to 1300.

UPDATE ORDERS SET PURCH\_AMT=600 WHERE ORD\_NO=70003;

UPDATE ORDERS SET PURCH\_AMT=1300 WHERE ORD\_NO=70004;

12.WRITE THE STATEMENT AS CUSTOMER ID OF (CUSTOMER NAME) IS : CUSTOMER ID.

SELECT 'CUSTOMER ID OF '+ CUST\_NAME + ' IS ' +' : ' + CAST(CUSTOMER\_ID AS VARCHAR(30))

AS "CUSTOMER INFORMATION"

FROM CUSTOMER

13.DISPLAY ALL THE SALESMAN NAME IN UPPER CASE,LOWER CASE,LENGTH.

SELECT UPPER(NAME) AS "UPPER\_CASE",LOWER(NAME) AS "LOWER\_CASE",LEN(NAME) AS "LENGTH"

FROM SALESMAN;

15.Write a query to display all customers with a grade above 100.

SELECT \*

FROM CUSTOMER

WHERE GRADE > 100;

16. Write a query statement to display all customers in New York who have a grade value above 100.

SELECT \*

FROM CUSTOMER

WHERE CITY='NEW YORK' AND GRADE > 100;

17.Write a SQL statement to display all the customers, who are either belongs to the city New York

or not had a grade above 100.

SELECT \*

FROM CUSTOMER

WHERE CITY='NEW YORK' OR NOT GRADE > 100;

18.Write a SQL query to display those customers who are neither belongs to the city New York

nor grade value is more than 100.

SELECT \*

FROM CUSTOMER

WHERE CITY <> 'NEW YORK' AND NOT GRADE >100;

19.Write a SQL statement to display either those orders which are not issued on date

2012-09-10 and issued by the salesman whose ID is 5005 and below or those orders which

purchase amount is 1000.00 and below.

SELECT \*

FROM ORDERS

WHERE NOT ((ORD\_DATE = '2012-09-10' AND SALESMAN\_ID > 5005) OR PURCH\_AMT >1000);

20.Write a SQL statement to display salesman\_id, name, city and commission who gets the commission

within the range more than 0.10% and less than 0.12%.

SELECT SALESMAN\_ID,NAME,CITY,COMMISSION

FROM SALESMAN

WHERE COMMISSION > 0.10 AND COMMISSION < 0.12;

21.Write a SQL query to display all orders where purchase amount less than 200 or exclude

those orders which order date is on or greater than 10th Feb,2012

and customer id is below 3009

SELECT \*

FROM ORDERS

WHERE (PURCH\_AMT < 200) OR NOT(ORD\_DATE >= '2012-02-10' AND CUSTOMER\_ID < 3009);

22. Write a SQL statement to exclude the rows which satisfy

1) order dates are 2012-08-17 and purchase amount is below 1000

2) customer id is greater than 3005 and purchase amount is below 1000.

SELECT \*

FROM ORDERS

WHERE NOT ((ORD\_DATE = '2012-08-17' OR CUSTOMER\_ID > 3005) AND PURCH\_AMT < 1000);

23.Write a SQL statement to find those salesmen with all

information who come from the city either Paris or Rome

SELECT \*

FROM SALESMAN

WHERE CITY='PARIS' OR CITY='ROME';

24. Write a query to filter those salesmen with all information

who comes from any of the cities Paris and Rome.

SELECT \*

FROM SALESMAN

WHERE CITY IN('PARIS','ROME');

25.Write a query to produce a list of salesman\_id, name, city and commission

of each salesman who live in cities other than Paris and Rome.

SELECT \*

FROM SALESMAN

WHERE CITY NOT IN ('PARIS','ROME');

26.Write a SQL statement to find those salesmen with all information

who gets the commission within a range of 0.12 and 0.14.

SELECT \*

FROM SALESMAN

WHERE COMMISSION BETWEEN 0.12 AND 0.14;

27.Write a query to filter all those orders with all information which

purchase amount value is within the range 500 and 4000 except

those orders of purchase amount value 948.50 and 1983.43.

SELECT \*

FROM ORDERS

WHERE PURCH\_AMT BETWEEN 500 AND 4000

AND

NOT PURCH\_AMT IN (948.50,1983.43);

-------SECOND WAY------

SELECT \*

FROM ORDERS

WHERE PURCH\_AMT BETWEEN 500 AND 4000

AND

NOT PURCH\_AMT = 948.50 AND NOT PURCH\_AMT =1983.43;

28.Write a SQL statement to find those salesmen with all information whose name containing

the 1st character is 'N' and the 4th character is 'l'

and rests may be any character.

SELECT \*

FROM SALESMAN

WHERE NAME LIKE 'N\_\_L%';

29.Write a SQL statement to find the total purchase amount of all orders

SELECT SUM(PURCH\_AMT) AS "TOTAL OF PURCH"

FROM ORDERS;

30.. Write a SQL statement to find the average purchase amount of all orders.

SELECT AVG(PURCH\_AMT) AS "AVERAGE OF PURCH"

FROM ORDERS;

31.Write a SQL statement to find the number of salesmen

currently listing for all of their customers.

SELECT COUNT( DISTINCT SALESMAN\_ID)

FROM ORDERS;

32. Write a SQL statement to get the maximum and minimum purchase amount of all the orders.

SELECT MAX(PURCH\_AMT) AS "MAXIMUM",MIN (PURCH\_AMT) AS "MINIMUM"

FROM ORDERS;

33.Write a SQL statement which selects the highest grade

for each of the cities of the customers.

SELECT CITY,MAX(GRADE) AS "MAX GRADE"

FROM CUSTOMER

GROUP BY CITY;

34.Write a SQL statement to find the highest purchase amount ordered By the each customer with their ID and highest purchase amount.

SELECT CUSTOMER\_ID, MAX (PURCH\_AMT) AS "MAX"

FROM ORDERS

GROUP BY CUSTOMER\_ID

ORDER BY CUSTOMER\_ID DESC;

35.. Write a SQL statement to find the highest purchase amount ordered by the each

customer on a particular date with their ID, order date and highest purchase amount.

SELECT CUSTOMER\_ID,ORD\_DATE,MAX(PURCH\_AMT) AS "MAX"

FROM ORDERS

GROUP BY CUSTOMER\_ID,ORD\_DATE

ORDER BY CUSTOMER\_ID ASC;

36.Write a SQL statement to find the highest purchase amount

on a date '2012-08-17' for each salesman with their ID.

SELECT SALESMAN\_ID,MAX(PURCH\_AMT) AS "MAX"

FROM ORDERS

WHERE ORD\_DATE = '2012-08-17'

GROUP BY SALESMAN\_ID;

37.Write a SQL statement to find the highest purchase amount with their ID

and order date, for only those customers who have highest purchase

amount in a day is more than 2000.

SELECT CUSTOMER\_ID,ORD\_DATE,MAX(PURCH\_AMT) AS "MAX"

FROM ORDERS

WHERE PURCH\_AMT > 2000

GROUP BY CUSTOMER\_ID,ORD\_DATE;

38.Write a query to find those customers with their name and

those salesmen with their name and

city who lives in the same city.

SELECT CUSTOMER.CUST\_NAME,

SALESMAN.NAME,SALESMAN.CITY

FROM SALESMAN,CUSTOMER

WHERE SALESMAN.CITY = CUSTOMER.CITY;

-------DONE WITH INNER JOIN FOR MATCHED RECORD-----

SELECT S.NAME AS "SALESMAN NAME",S.CITY AS "SALESMAN CITY",

C.CUST\_NAME AS "CUSTOMER NAME",C.CITY AS "CUSTOMER CITY"

FROM

SALESMAN AS "S" INNER JOIN CUSTOMER AS "C"

ON

S.SA LESMAN\_ID = C.SALESMAN\_ID

WHERE S.CITY =C.CITY;

39. Write a SQL statement to display all those orders by the customers

not located in the same cities

where their salesmen live.

SELECT O.ORD\_NO,O.CUSTOMER\_ID,O.SALESMAN\_ID

FROM SALESMAN AS "S",ORDERS AS "O",CUSTOMER AS "C"

WHERE C.CITY <> S.CITY

AND O.CUSTOMER\_ID = C.CUSTOMER\_ID

AND O.SALESMAN\_ID = S.SALESMAN\_ID;