**TEST -3**

**SUBMITTED BY – PADMNI BHARADWAJ**

ORDERS DATABASE

**1. Creating the tables by specifying appropriate Primary and Foreign keys**

SQL> CREATE TABLE SALESMAN(S\_ID CHAR(3),NAME VARCHAR(20),CITY VARCHAR(20),COMM NUMBER(5),CONSTRAINT PK\_SALES PRIMARY KEY(S\_ID));

SQL> CREATE TABLE CUSTOMER(C\_ID CHAR(3),C\_NAME VARCHAR(20),CITY VARCHAR(20),GRADE CHAR(2),S\_ID CHAR(3),CONSTRAINT PK\_CUST1 PRIMARY KEY(C\_ID), CONSTRAINT FK\_CUST FOREIGN KEY(S\_ID) REFERENCES SALESMAN(S\_ID));

SQL>CREATE TABLE ORDERS(O\_NO CHAR(3),PUR\_AMT NUMBER(5),O\_DATE DATE,C\_ID CHAR(3),S\_ID CHAR(3),

CONSTRAINT PK\_ORD PRIMARY KEY(O\_NO),

CONSTRAINT FK\_C FOREIGN KEY(C\_ID) REFERENCES CUSTOMER(C\_ID),

CONSTRAINT FK\_S FOREIGN KEY(S\_ID) REFERENCES SALESMAN(S\_ID));

**2. Count the customers with grades above Bangalore’s average.**

SQL> SELECT \* FROM CUSTOMER;

C\_I C\_NAME CITY GR S\_I

--- -------------------- -------------------- -- ---

C1 ABHI BANGALORE A S4

C2 BHARTI BANGALORE C S3

C3 MADHU KERALA B S3

C4 SWATHI UP D S2

C5 HARINI BANGALORE B S1

SQL> SELECT COUNT(GRADE) FROM CUSTOMER

2

SQL> SELECT COUNT(GRADE) FROM CUSTOMER

2 WHERE CITY='BANGALORE' AND

3 GRADE IN ('A','B','C');

COUNT(GRADE)

------------

3

**3. Find the name and numbers of all salesman who had more than one customer.**

SQL> SELECT \* FROM SALESMAN;

S\_I NAME CITY COMM

--- -------------------- -------------------- ----------

S1 PADMNI BANGALORE 3000

S2 JOSEPH DELHI 2000

S3 VIBHA DELHI 4000

S4 BHAVANA BANGALORE 5000

SQL> SELECT \* FROM CUSTOMER;

C\_I C\_NAME CITY GR S\_I

--- -------------------- -------------------- -- ---

C1 ABHI BANGALORE A S4

C2 BHARTI BANGALORE C S3

C3 MADHU KERALA B S3

C4 SWATHI UP D S2

C5 HARINI BANGALORE B S1

SQL> SELECT S\_ID,NAME FROM SALESMAN,CUSTOMER

2

SQL>

SQL> SELECT S\_ID,NAME FROM SALESMAN A WHERE 1<(SELECT COUNT(\*) FROM CUSTOMER WHERE S\_ID=A.S\_ID);

S\_I NAME

--- --------------------

S3 VIBHA

**4. List all the salesman and indicate those who have and don’t have customers in their**

cities (Use UNION operation.)

SQL> SELECT SALESMAN.S\_ID,NAME,C\_NAME,COMM FROM SALESMAN,CUSTOMER

2 WHERE SALESMAN.CITY=CUSTOMER.CITY

3 UNION

4 SELECT S\_ID,NAME,'NOT MATCHING',COMM FROM SALESMAN

5 WHERE NOT CITY = ANY(SELECT CITY FROM CUSTOMER)

6 ORDER BY 2 DESC;

S\_I NAME C\_NAME COMM

--- -------------------- -------------------- ----------

S3 VIBHA NOT MATCHING 4000

S1 PADMNI ABHI 3000

S1 PADMNI BHARTI 3000

S1 PADMNI HARINI 3000

S2 JOSEPH NOT MATCHING 2000

S4 BHAVANA ABHI 5000

S4 BHAVANA BHARTI 5000

S4 BHAVANA HARINI 5000

8 rows selected.

**5. Create a view that finds the salesman who has the customer with the highest order**

**of a day.**

SQL> CREATE VIEW SALES AS SELECT A.O\_DATE,B.S\_ID,B.NAME FROM ORDERS A,SALESMAN B

2 WHERE B.S\_ID=B.S\_ID AND

3 A.PUR\_AMT=(SELECT MAX(PUR\_AMT) FROM ORDERS C WHERE C.O\_DATE=A.O\_DATE);

View created.

SQL> SELECT \* FROM SALES;

O\_DATE S\_I NAME

--------- --- --------------------

12-SEP-17 S1 PADMNI

12-SEP-17 S2 JOSEPH

12-SEP-17 S3 VIBHA

12-SEP-17 S4 BHAVANA

11-DEC-16 S1 PADMNI

11-DEC-16 S2 JOSEPH

11-DEC-16 S3 VIBHA

11-DEC-16 S4 BHAVANA

03-AUG-17 S1 PADMNI

03-AUG-17 S2 JOSEPH

03-AUG-17 S3 VIBHA

O\_DATE S\_I NAME

--------- --- --------------------

03-AUG-17 S4 BHAVANA

12 rows selected.

**6. Demonstrate the DELETE operation by removing salesman with id 1000. All his**

**Order must also be deleted.**

SQL> DELETE FROM ORDERS

2 WHERE S\_ID=1000;

1 row deleted.

SQL> SELECT \* FROM CUSTOMER;

C\_I C\_NAME CITY GR S\_I

--- -------------------- -------------------- -- ---

C1 ABHI BANGALORE A S4

C2 BHARTI BANGALORE C S3

C3 MADHU KERALA B S3

C4 SWATHI UP D 1000

C5 HARINI BANGALORE B S1

SQL> UPDATE CUSTOMER

2 SET S\_ID=NULL

3 WHERE S\_ID=1000;

1 row updated.

SQL> DELETE FROM SALESMAN

2 WHERE S\_ID=1000;

1 row deleted.

LIBRARY DATABASE

**1. Creating the tables by specifying appropriate Primary and Foreign keys**

CREATE TABLE BOOK1(B\_ID CHAR(2),TITLE VARCHAR2(20),PNAME VARCHAR2(10),P\_YEAR NUMBER(4),CONSTRAINT PK\_B PRIMARY KEY(B\_ID));

SQL> CREATE TABLE BK\_AUTH (B\_ID char(3),A\_NAME VARCHAR2(20),

CONSTRAINT PK\_B\_A PRIMARY KEY(B\_ID,A\_NAME),

CONSTRAINT FK\_B FOREIGN KEY(B\_ID) REFERENCES BOOK1(B\_ID));

Table created.

SQL> CREATE TABLE PUBLISHER

2 (NAME VARCHAR2(20),ADDR VARCHAR(20),PH NUMBER(10),

3 CONSTRAINT PK\_pub PRIMARY KEY(NAME));

Table created.

SQL> CREATE TABLE LIB\_BRANCH(BR\_ID CHAR(3),BR\_NAME VARCHAR(20),ADDR VARCHAR(20),

CONSTRAINT PK\_BR PRIMARY KEY(BR\_ID));

Table created.

SQL> CREATE TABLE LIB\_BRANCH

(B\_ID CHAR(3),BR\_ID CHAR(3),CARD\_NO CHAR(3),D\_OUT DATE,

DUE\_DATE DATE,

CONSTRAINT PK1 PRIMARY KEY(B\_ID,BR\_ID,CARD\_NO),

CONSTRAINT FK2 FOREIGN KEY(B\_ID) REFERENCES BOOK1(B\_ID),

CONSTRAINT FK3 FOREIGN KEY(BR\_ID) REFERENCES LIBRARY\_BRANCH(B\_ID)

);

Table created.

SQL> CREATE TABLE BOOK\_LENDING

(B\_ID CHAR(3),BR\_ID CHAR(3),CARD\_NO CHAR(3)),D\_OUT DATE,

DUE\_DATE DATE,

CONSTRAINT PK\_BOOK\_LENDING PRIMARY KEY(B\_ID, BR\_ID, CARD\_NO),

CONSTRAINT FK\_B FOREIGN KEY(B\_ID) REFERENCES BOOK(B\_ID),

CONSTRAINT FK\_BR FOREIGN KEY(BR\_ID) REFERENCES LIBRARY\_BRANCH(BR\_ID));

Table created.