SRM Institute of Science & Technology, Delhi NCR Campus

Department of Computer Science & Engineering

Database Management Systems (18CSC303J)

Lab File

SRM Institute of Science & Technology, Delhi NCR Campus

Department of Computer Science & Engineering

LABORATORY FILE

Faculty Name	: Ms.Neetu Bansla	**Department**	: CSE
Course Name	: DBMS Lab	**Course Code**	: 18CSC303J
Year/Sem	: $3^{rd}/6^{th}$	**Academic Year**	: 2022-23

LIST OF EXPERIMENTS

Expt. No.	Title of experiment
1.	Creating tables and writing Queries in SQL.
2.	To implement various DML Operations on table
3.	To Implement the restrictions/constraints on the table
4.	To Alter the structure of the table
5.	To implement the concept of Joins
6.	To implement the concept of grouping of Data
7.	To implement the concept of Sub Queries
8.	To implement the concept of Indexes and views.
9.	To implement the basics of PL/SQL.
10.	To implement the concept of Cursor and Trigger

Content Beyond Syllabus	
1.	Write a program to implement REPORTS.
2.	Write a program to implement FORMS.

GUIDELINES FOR LABORTORY RECORD PREPARATION

While preparing the lab records, the student is required to adhere to the following guidelines:

Contents to be included in Lab Records:

1. Cover page
2. Index
3. Experiments
    - Aim
    - Source code
    - Input-Output

DBMS Lab (18CSC303J)

Student Name	ANANYA GUPTA
Registration No.	RA1911003030265
Section- Batch	B.TECH CSE I – BATCH 1

SRM Institute of Science & Technology, Ghaziabad
Department of Computer Science & Engineering

INDEX

Experiment No.	Experiment Name	Date of Conduction	Date of Submission	Faculty Signature
1	Creating tables and writing Queries in SQL.	24-01-2022	31-01-2022	
2	To implement various DML Operations on table.	31-01-2022	14-02-2022	
3	To Implement the restrictions/constraints on the table.	14-02-2022	21-02-2022	
4	To Alter the structure of the table.	21-02-2022	28-02-2022	
5	To implement the concept of Joins.	28-02-2022	07-03-2022	
6	To implement the concept of grouping of Data.	07-02-2022	24-03-2022	

Experiment No.1

Aim: Creating tables and writing Queries in SQL.

Q1. Create the following tables:

i) client_master

columnname	datatype	size
client_no	varchar2	6
name	varchar2	20
address1	varchar2	30
address2	varchar2	30
city	varchar2	15
state	varchar2	15
pincode	number	6
bal_due	number	10,2

ii) Product_master

Columnname	datatype
Product_no	varchar2
Description	varchar
Profit_percent	number
Unit_measure	varchar
Qty_on_hand	number
Reorder_lvl	number
Sell_price	number
Cost_price	number

Q2- Insert the following data into their respective tables:

Clientno	Name	city	pincode	state	bal.due
0001	Ivan	Bombay	400054	Maharashtra	15000
0002	Vandana	Madras	780001	Tamilnadu	0
0003	Pramada	Bombay	400057	Maharashtra	5000
0004	Basu	Bombay	400056	Maharashtra	0
0005	Ravi	Delhi	100001		2000
0006	Rukmini	Bombay	400050	Maharashtra	0

Data for Product Master:

Product No.	Desciption	Profit % Percent	Unit measured	Qty on hand	Reorder lvl	Sell price	Cost price
P00001	1.44floppies	5	piece	100	20	525	500
P03453	Monitors	6	piece	10	3	12000	11200

P06734	Mouse	5	piece	20	5	1050	500
P07865	1.22 floppies	5	piece	100	20	525	500
P07868	Keyboards	2	piece	10	3	3150	3050
P07885	CD Drive	2.5	piece	10	3	5250	5100
P07965	540 HDD	4	piece	10	3	8400	8000
P07975	1.44 Drive	5	piece	10	3	1050	1000
P08865	1.22 Drive	5	piece	2	3	1050	1000

ANSWER –

Run SQL Command Line

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Jan 24 07:41:06 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> desc client_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                          VARCHAR2(6)
 NAME                                               VARCHAR2(20)
 ADDRESS1                                           VARCHAR2(30)
 ADDRESS2                                           VARCHAR2(30)
 CITY                                               VARCHAR2(15)
 STATE                                              VARCHAR2(15)
 PINCODE                                            NUMBER(6)
 BAL_DUE                                            NUMBER(10,2)

SQL> desc product_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_NO                                         VARCHAR2(20)
 DESCRIPTION                                        VARCHAR2(30)
 PROFIT_PERCENT                                     NUMBER(20)
 UNIT_MEASURE                                       VARCHAR2(20)
 QTY_ON_HAND                                        NUMBER(20)
 SELL_PRICE                                         NUMBER(20)
 COST_PRICE                                         NUMBER(20)
```

Run SQL Command Line

```
SQL> INSERT INTO CLIENT_MASTER VALUES(0001,'IVAN',NULL,NULL,'BOMBAY','MAHARASHTRA',400054,15000);

1 row created.

SQL> INSERT INTO CLIENT_MASTER VALUES(0002,'VANDANA',NULL,NULL,'MADRAS','TAMILNADU',780001,0);

1 row created.

SQL> INSERT INTO CLIENT_MASTER VALUES(0003,'PRAMADA',NULL,NULL,'BOMBAY','MAHARASHTRA',400057,5000);

1 row created.

SQL> INSERT INTO CLIENT_MASTER VALUES(0004,'BASU',NULL,NULL,'BOMBAY','MAHARASHTRA',400056,0);

1 row created.

SQL> INSERT INTO CLIENT_MASTER VALUES(0005,'RAVI',NULL,NULL,'DELHI',NULL,100001,2000);

1 row created.

SQL> INSERT INTO CLIENT_MASTER VALUES(0006,'RUKMINI',NULL,NULL,'BOMBAY','MAHARASHTRA',400050,0);

1 row created.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT NAME                ADDRESS1
------ ------------------- ------------------------------
ADDRESS2                      CITY           STATE              PINCODE
------------------------------ -------------- -------------- ----------
   BAL_DUE
----------
1      IVAN
                              BOMBAY         MAHARASHTRA         400054
     15000

2      VANDANA
                              MADRAS         TAMILNADU           780001
        0
```

Run SQL Command Line

```
1      IVAN
                              BOMBAY          MAHARASHTRA          400054
     15000

2      VANDANA
                              MADRAS          TAMILNADU            780001
         0

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY            STATE             PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------

3      PRAMADA
                              BOMBAY          MAHARASHTRA          400057
      5000

4      BASU
                              BOMBAY          MAHARASHTRA          400056

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY            STATE             PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
         0

5      RAVI
                              DELHI                                100001
      2000

6      RUKMINI

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY            STATE             PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
                              BOMBAY          MAHARASHTRA          400050
         0


6 rows selected.

SQL>
```

Run SQL Command Line

```
SQL>  INSERT INTO PRODUCT_MASTER VALUES('P00001','1.44FLOPPIES',5,'PIECE',100,,525,500);
 INSERT INTO PRODUCT_MASTER VALUES('P00001','1.44FLOPPIES',5,'PIECE',100,,525,500)
                                                                      *
ERROR at line 1:
ORA-00936: missing expression


SQL> ED
Wrote file afiedt.buf

  1*  INSERT INTO PRODUCT_MASTER VALUES('P00001','1.44FLOPPIES',5,'PIECE',100,525,500)
SQL> /

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P03453','MONITORS',6,'PIECE',10,12000,11200);

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P06734','MOUSE',5,'PIECE',1050,500);
INSERT INTO PRODUCT_MASTER VALUES('P06734','MOUSE',5,'PIECE',1050,500)
            *
ERROR at line 1:
ORA-00947: not enough values


SQL> ED
Wrote file afiedt.buf

  1* INSERT INTO PRODUCT_MASTER VALUES('P06734','MOUSE',5,'PIECE',20,1050,500)
SQL> /

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P07865','1.22FLOPPIES',5,'PIECE',100,525,500);

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P07868','KEYBOARDS',2,'PIECE',10,3150,3050);

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P07885,'CD DRIVE',2.5,'PIECE',10,5250,5100);
ERROR:
ORA-01756: quoted string not properly terminated


SQL> ED
Wrote file afiedt.buf
```

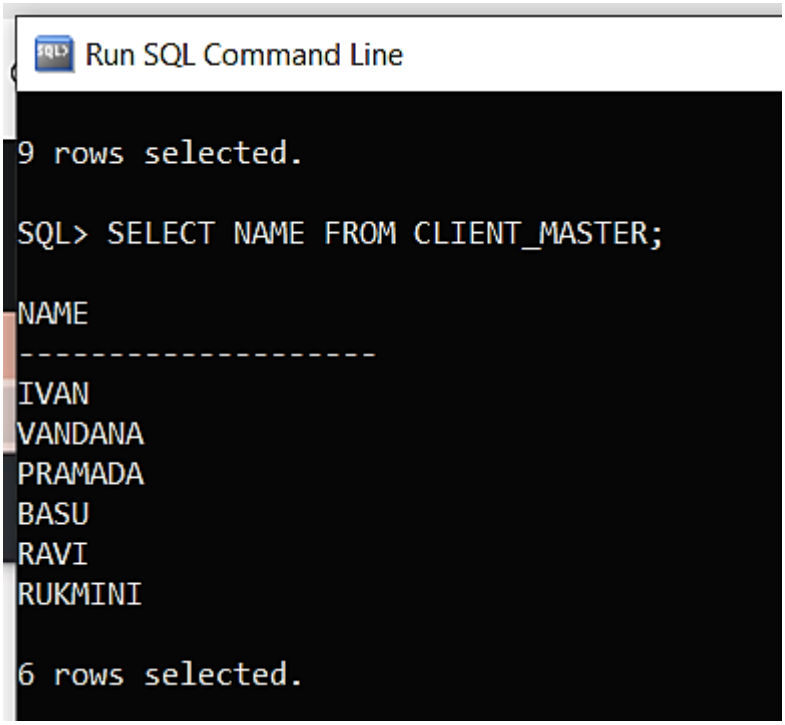
```
SQL> INSERT INTO PRODUCT_MASTER VALUES('P07885,'CD DRIVE',2.5,'PIECE',10,5250,5100);
ERROR:
ORA-01756: quoted string not properly terminated


SQL> ED
Wrote file afiedt.buf

  1* INSERT INTO PRODUCT_MASTER VALUES('P07885','CD DRIVE',2.5,'PIECE',10,5250,5100)
SQL> /

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P07965','540 HDD',4,'PIECE',10,8400,8000);

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P07975','1.44 DRIVE',5,'PIECE',10,1050,1000);

1 row created.

SQL> INSERT INTO PRODUCT_MASTER VALUES('P08865','1.22 DRIVE',5,'PIECE',2,1050,1000);

1 row created.
```

Run SQL Command Line

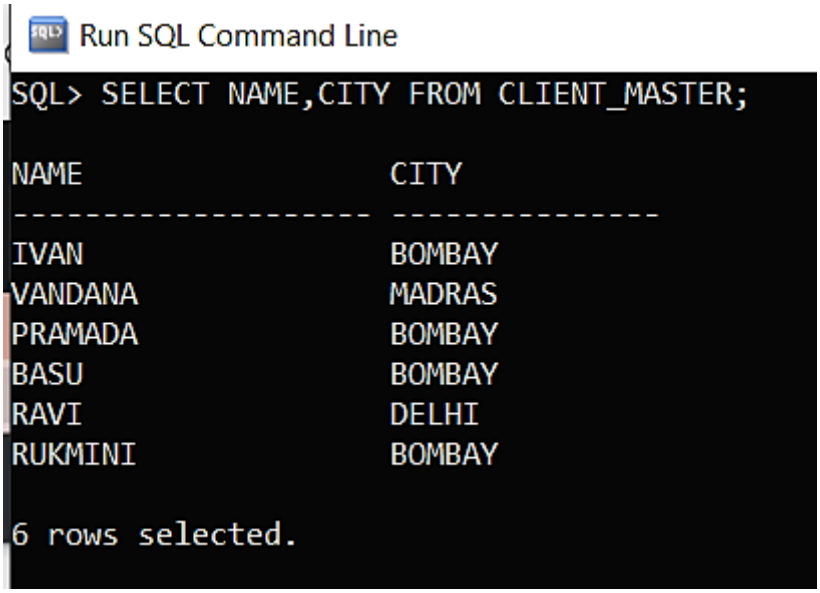
```
SQL> SELECT * FROM PRODUCT_MASTER;

PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P00001               1.44FLOPPIES                                5
PIECE                        100        525        500

P03453               MONITORS                                    6
PIECE                         10      12000      11200

P06734               MOUSE                                       5
PIECE                         20       1050        500


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P07865               1.22FLOPPIES                                5
PIECE                        100        525        500

P07868               KEYBOARDS                                   2
PIECE                         10       3150       3050

P07885               CD DRIVE                                    3
PIECE                         10       5250       5100


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P07965               540 HDD                                     4
PIECE                         10       8400       8000

P07975               1.44 DRIVE                                  5
PIECE                         10       1050       1000

P08865               1.22 DRIVE                                  5
PIECE                          2       1050       1000


9 rows selected.

SQL>
```

Q3:- On the basis of above two tables answer the following Queries:

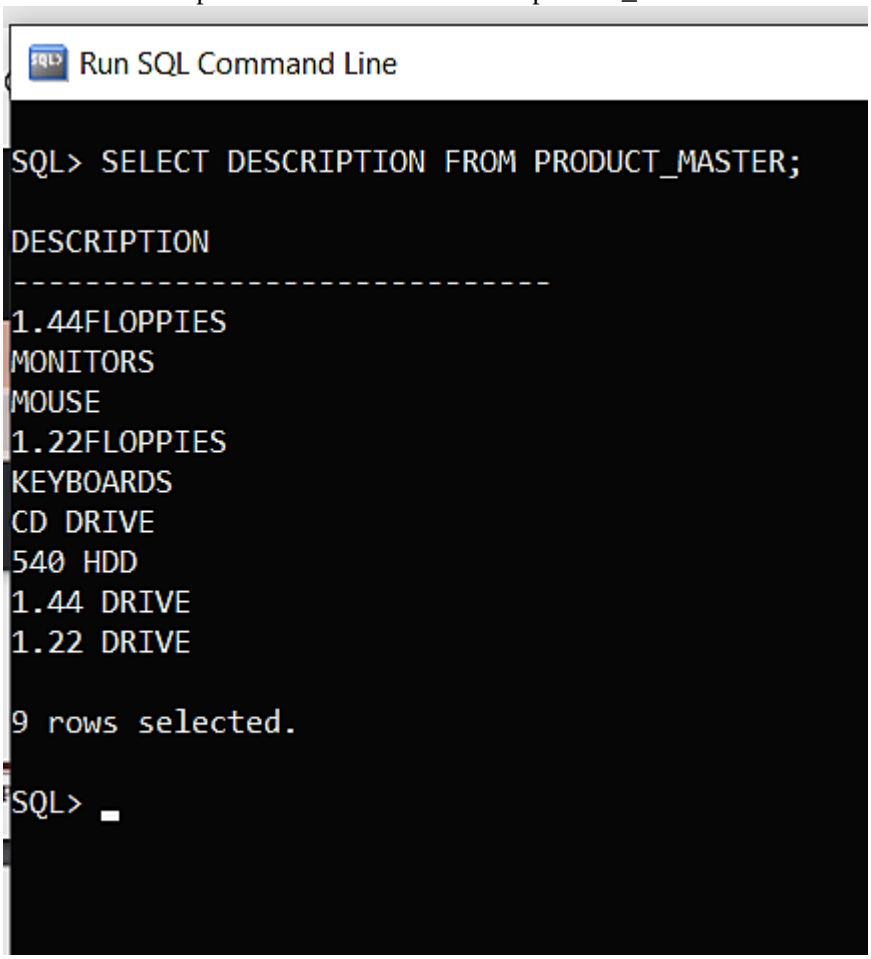
i) Find out the names of all the clients.



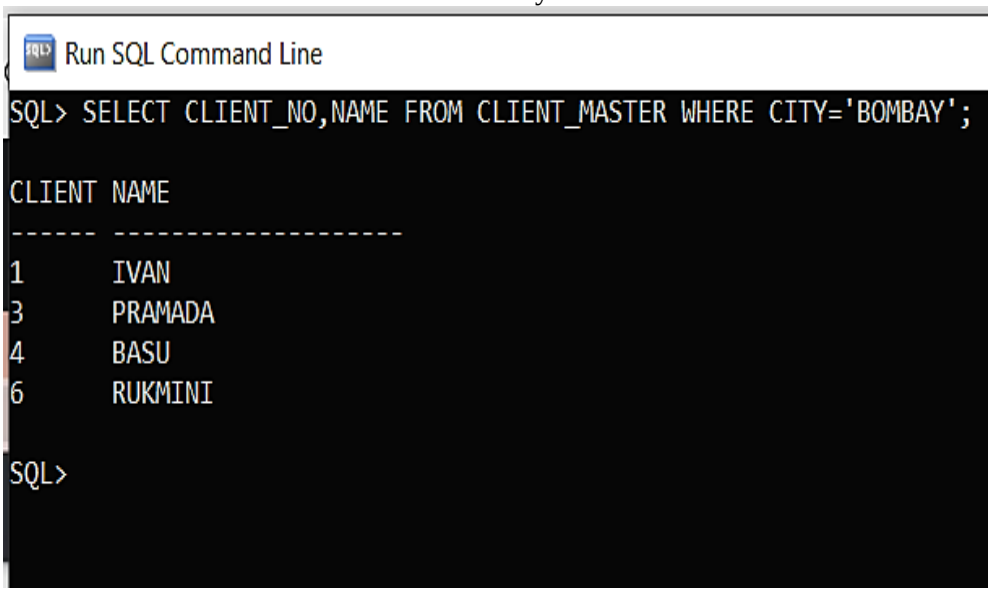
ii) Retrieve the list of names and cities of all the clients.



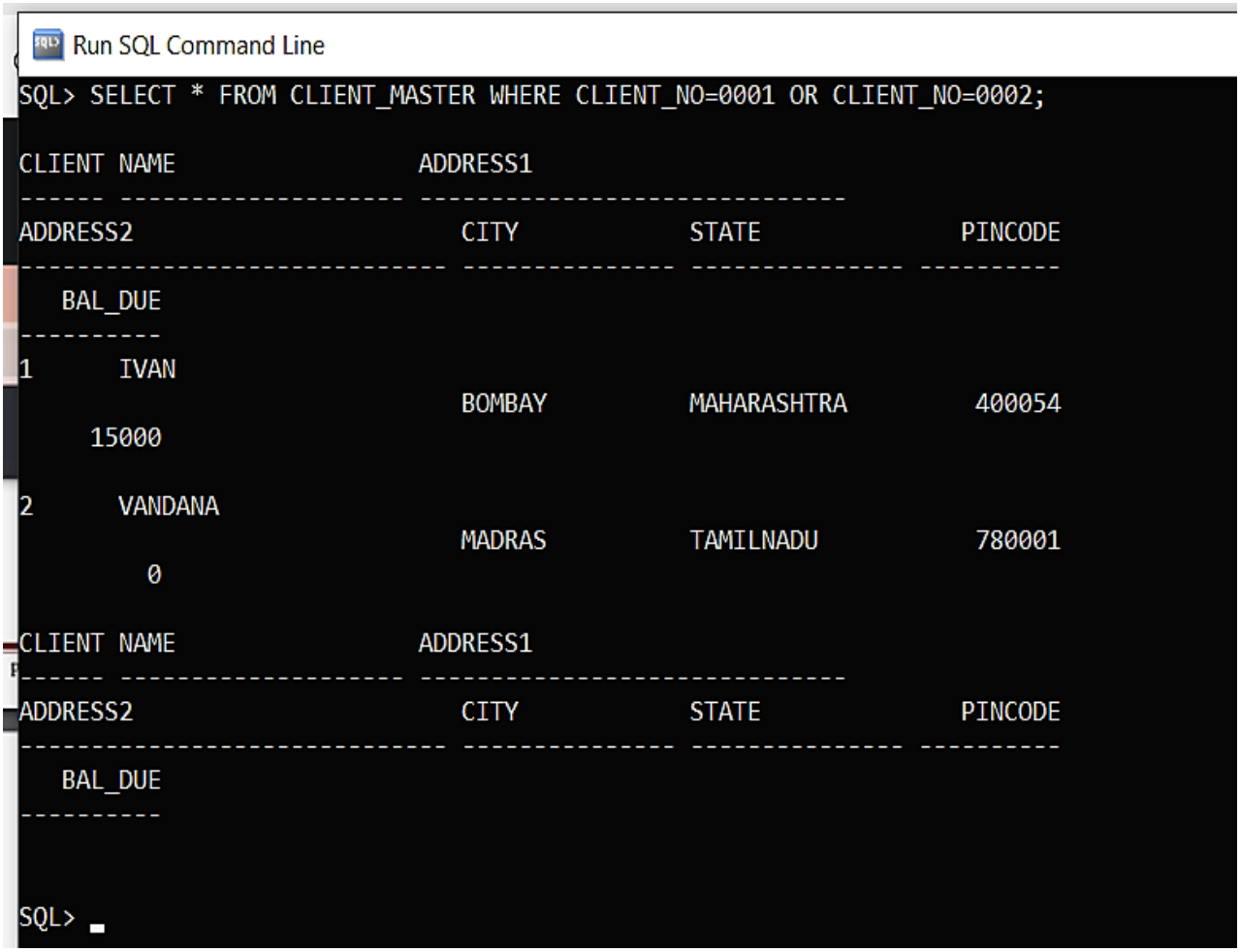
iii) List the various products available from the product_master table.



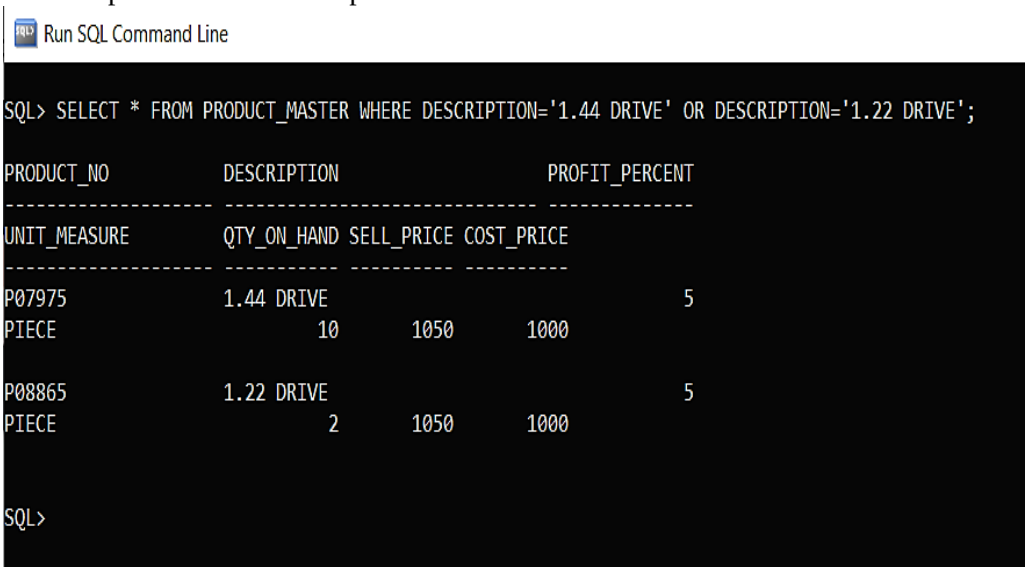
iv) List all the clients who are located in Bombay.



v) Display the information for client no 0001 and 0002.



vi) Find the products with description as '1.44 drive' and '1.22 Drive'.



vii) Find all the products whose sell price is greater than 5000.

Run SQL Command Line

```
SQL> SELECT * FROM PRODUCT_MASTER WHERE SELL_PRICE > 5000;

PRODUCT_NO          DESCRIPTION                    PROFIT_PERCENT
------------------- ------------------------------ --------------
UNIT_MEASURE        QTY_ON_HAND SELL_PRICE COST_PRICE
------------------- ----------- ---------- ----------
P03453              MONITORS                                    6
PIECE                       10      12000      11200

P07885              CD DRIVE                                    3
PIECE                       10       5250       5100

P07965              540 HDD                                     4
PIECE                       10       8400       8000


SQL>
```

viii) Find the list of all clients who stay in city ‘Bombay’ or city ‘Delhi’ or ‘Madras’.

Run SQL Command Line

```
SQL> SELECT * FROM CLIENT_MASTER WHERE CITY='BOMBAY' OR CITY='DELHI' OR CITY='MADRAS';

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                   CITY            STATE              PINCODE
-------------------------- --------------- --------------- ----------
   BAL_DUE
----------
1      IVAN
                           BOMBAY          MAHARASHTRA         400054
     15000

2      VANDANA
                           MADRAS          TAMILNADU           780001
         0

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                   CITY            STATE              PINCODE
-------------------------- --------------- --------------- ----------
   BAL_DUE
----------

3      PRAMADA
                           BOMBAY          MAHARASHTRA         400057
      5000

4      BASU
                           BOMBAY          MAHARASHTRA         400056

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                   CITY            STATE              PINCODE
-------------------------- --------------- --------------- ----------
   BAL_DUE
----------
         0

5      RAVI
                           DELHI                               100001
      2000

6      RUKMINI
```

```
---------- ---------- ---------- ----------
  BAL_DUE
----------
         0

5       RAVI
                            DELHI                          100001
       2000

6       RUKMINI

CLIENT NAME                ADDRESS1
------ ------------------- ----------------------------
ADDRESS2                   CITY           STATE           PINCODE
---------- ---------- ---------- ----------
  BAL_DUE
----------
                            BOMBAY         MAHARASHTRA     400050
         0


6 rows selected.

SQL>
```

ix) Find the product whose selling price is greater than 2000 and less than or equal to 5000.

```
Run SQL Command Line
SQL> SELECT DESCRIPTION FROM PRODUCT_MASTER WHERE SELL_PRICE > 2000 AND SELL_PRICE <=5000;

DESCRIPTION
---------------------------
KEYBOARDS

SQL>
```

x) List the name, city and state of clients not in the state of 'Maharashtra'.

```
Run SQL Command Line
SQL> SELECT NAME,CITY,STATE FROM CLIENT_MASTER WHERE STATE!='MAHARASHTRA';

NAME                CITY           STATE
------------------- -------------- ---------------
VANDANA             MADRAS         TAMILNADU

SQL>
```

Experiment No. 2

Aim: To implement various DML Operations on table

Question.1 Using the table client master and product master answer the following Queries.

i. Change the selling price of ‘1.44 floppy drive to Rs.1150.00

Run SQL Command Line

```
SQL> UPDATE PRODUCT_MASTER SET SELL_PRICE=1150 WHERE DESCRIPTIO'1.44FLOPPIES';
UPDATE PRODUCT_MASTER SET SELL_PRICE=1150 WHERE DESCRIPTIO'1.44FLOPPIES'
                                                          *
ERROR at line 1:
ORA-00920: invalid relational operator


SQL> ED
Wrote file afiedt.buf

  1* UPDATE PRODUCT_MASTER SET SELL_PRICE=1150 WHERE DESCRIPTION='1.44FLOPPIES'
SQL> /

1 row updated.

SQL> SELECT * FROM PRODUCT_MASTER;

PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P00001               1.44FLOPPIES                                5
PIECE                        100       1150        500

P03453               MONITORS                                    6
PIECE                         10      12000      11200

P06734               MOUSE                                       5
PIECE                         20       1050        500


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P07865               1.22FLOPPIES                                5
PIECE                        100        525        500

P07868               KEYBOARDS                                   2
PIECE                         10       3150       3050

P07885               CD DRIVE                                    3
PIECE                         10       5250       5100


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P07965               540 HDD                                     4
```

ii. Delete the record with client 0001 from the client master table.

Run SQL Command Line

```
SQL> DELETE FROM CLIENT_MASTER WHERE CLIENT_NO='1';

1 row deleted.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY           STATE             PINCODE
------------------------------ -------------- --------------- ----------
   BAL_DUE
----------
2      VANDANA
                              MADRAS         TAMILNADU          780001
         0

3      PRAMADA
                              BOMBAY         MAHARASHTRA        400057
      5000

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY           STATE             PINCODE
------------------------------ -------------- --------------- ----------
   BAL_DUE
----------

4      BASU
                              BOMBAY         MAHARASHTRA        400056
         0

5      RAVI
                              DELHI                             100001

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY           STATE             PINCODE
------------------------------ -------------- --------------- ----------
   BAL_DUE
----------
      2000

6      RUKMINI
                              BOMBAY         MAHARASHTRA        400050
         0
```

iii. Change the city of client_no'0005' to Bombay.

Run SQL Command Line

```
SQL> UPDATE CLIENT_MASTER SET CITY='BOMBAY' WHERE CLIENT_NO='5';

1 row updated.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT NAME                ADDRESS1
------ ------------------- ------------------------------
ADDRESS2                      CITY            STATE              PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
2      VANDANA
                              MADRAS          TAMILNADU           780001
         0

3      PRAMADA
                              BOMBAY          MAHARASHTRA         400057
      5000

CLIENT NAME                ADDRESS1
------ ------------------- ------------------------------
ADDRESS2                      CITY            STATE              PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------

4      BASU
                              BOMBAY          MAHARASHTRA         400056
         0

5      RAVI
                              BOMBAY                              100001

CLIENT NAME                ADDRESS1
------ ------------------- ------------------------------
ADDRESS2                      CITY            STATE              PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
      2000

6      RUKMINI
                              BOMBAY          MAHARASHTRA         400050
         0
```

iv. Change the bal_due of client_no ‘0001, to 1000.

```
SQL> INSERT INTO CLIENT_MASTER VALUES(0001, "IVAN" , NULL , NULL ,"BOMBAY" , "MAHARASHTRA" , 400054 , 15000);
INSERT INTO CLIENT_MASTER VALUES(0001, "IVAN" , NULL , NULL ,"BOMBAY" , "MAHARASHTRA" , 400054 , 15000)
                                                            *
ERROR at line 1:
ORA-00984: column not allowed here


SQL> ED
Wrote file afiedt.buf

  1* INSERT INTO CLIENT_MASTER VALUES(0001, 'IVAN' , NULL , NULL ,'BOMBAY' , 'MAHARASHTRA' , 400054 , 15000)
SQL> /

1 row created.

SQL> UPDATE CLIENT_MASTER SET BAL_DUE=1000 WHERE CLIENT_NO='1';

1 row updated.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY           STATE              PINCODE
------------------------------ -------------- -------------- ----------
   BAL_DUE
----------
1      IVAN
                              BOMBAY         MAHARASHTRA         400054
      1000

2      VANDANA
                              MADRAS         TAMILNADU           780001
        0

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                      CITY           STATE              PINCODE
------------------------------ -------------- -------------- ----------
   BAL_DUE
----------

3      PRAMADA
                              BOMBAY         MAHARASHTRA         400057
      5000
```

v. Find the products whose selling price is more than 1500 and also find the new selling price as original selling price *15.

Run SQL Command Line

```
6 rows selected.

SQL> SELECT PRODUCT_NO,DESCRIPTION,SELL_PRICE*15 AS NEW_SELL_PRICE FROM PRODUCT_MASTER WHERE SELL_PRICE>1500;

PRODUCT_NO          DESCRIPTION                    NEW_SELL_PRICE
------------------- ------------------------------ --------------
P03453              MONITORS                               180000
P07868              KEYBOARDS                               47250
P07885              CD DRIVE                                78750
P07965              540 HDD                                126000
```

vi. Find out the clients who stay in a city whose second letter is a.

```
SQL> SELECT * FROM CLIENT_MASTER WHERE CITY LIKE'_A%';

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                        CITY            STATE               PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
2      VANDANA
                                MADRAS          TAMILNADU           780001
         0
```

vii. Find out the name of all clients having ‘a’ as the second letter in their names.

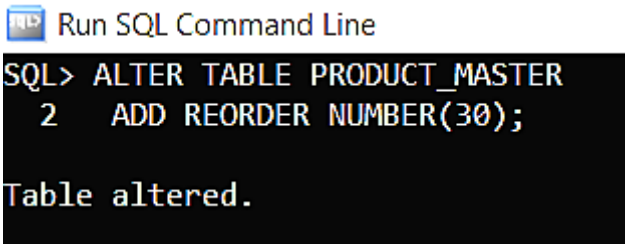
```
SQL>  SELECT * FROM CLIENT_MASTER WHERE NAME LIKE'_A%';

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                        CITY            STATE               PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
2      VANDANA
                                MADRAS          TAMILNADU           780001
         0

4      BASU
                                BOMBAY          MAHARASHTRA         400056
         0

CLIENT NAME                ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                        CITY            STATE               PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------

5      RAVI
                                BOMBAY                              100001
      2000
```

viii. List the products in sorted order of their description.

Run SQL Command Line

```
SQL> SELECT * FROM PRODUCT_MASTER ORDER BY DESCRIPTION;

PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P08865               1.22 DRIVE                                5
PIECE                          2       1050       1000

P07865               1.22FLOPPIES                              5
PIECE                        100        525        500

P07975               1.44 DRIVE                                5
PIECE                         10       1050       1000


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P00001               1.44FLOPPIES                              5
PIECE                        100       1150        500

P07965               540 HDD                                   4
PIECE                         10       8400       8000

P07885               CD DRIVE                                  3
PIECE                         10       5250       5100


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE
-------------------- ----------- ---------- ----------
P07868               KEYBOARDS                                 2
PIECE                         10       3150       3050

P03453               MONITORS                                  6
PIECE                         10      12000      11200

P06734               MOUSE                                     5
PIECE                         20       1050        500


9 rows selected.
```

ix. Count the total number of orders.



Run SQL Command Line

```
SQL> UPDATE PRODUCT_MASTER SET REORDER='20' WHERE DESCRIPTION='1.44FLOPPIES';

1 row updated.

SQL> SELECT * FROM PRODUCT_MASTER;

PRODUCT_NO          DESCRIPTION                    PROFIT_PERCENT
------------------- ------------------------------ --------------
UNIT_MEASURE        QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
------------------- ----------- ---------- ---------- ----------
P00001              1.44FLOPPIES                              5
PIECE                        100       1150        500         20

P03453              MONITORS                                  6
PIECE                         10      12000      11200

P06734              MOUSE                                     5
PIECE                         20       1050        500


PRODUCT_NO          DESCRIPTION                    PROFIT_PERCENT
------------------- ------------------------------ --------------
UNIT_MEASURE        QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
------------------- ----------- ---------- ---------- ----------
P07865              1.22FLOPPIES                              5
PIECE                        100        525        500

P07868              KEYBOARDS                                 2
PIECE                         10       3150       3050

P07885              CD DRIVE                                  3
PIECE                         10       5250       5100


PRODUCT_NO          DESCRIPTION                    PROFIT_PERCENT
------------------- ------------------------------ --------------
UNIT_MEASURE        QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
------------------- ----------- ---------- ---------- ----------
P07965              540 HDD                                   4
PIECE                         10       8400       8000

P07975              1.44 DRIVE                                5
PIECE                         10       1050       1000

P08865              1.22 DRIVE                                5
PIECE                          2       1050       1000


9 rows selected.
```

Run SQL Command Line

```
9 rows selected.

SQL> UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='MONITORS';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='5' WHERE DESCRIPTION='MOUSE';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='20' WHERE DESCRIPTION='1.22FLOPPIES';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='5' WHERE DESCRIPTION='KEYBOARDS';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='5' WHERE DESCRIPTION='CD DRIVE';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='540 HDD DRIVE';

0 rows updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='540 HDD';

1 row updated.

SQL>  UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='CD DRIVE';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='KEYBOARDS';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='1.44 DRIVE';

1 row updated.

SQL> UPDATE PRODUCT_MASTER SET REORDER='3' WHERE DESCRIPTION='1.22 DRIVE';

1 row updated.
```

Run SQL Command Line

```
SQL> SELECT * FROM PRODUCT MASTER;
SELECT * FROM PRODUCT MASTER
              *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> ED
Wrote file afiedt.buf

  1* SELECT * FROM PRODUCT_MASTER
SQL> /

PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
-------------------- ----------- ---------- ---------- ----------
P00001               1.44FLOPPIES                              5
PIECE                        100       1150        500         20

P03453               MONITORS                                  6
PIECE                         10      12000      11200          3

P06734               MOUSE                                     5
PIECE                         20       1050        500          5


PRODUCT_NO           DESCRIPTION                    PROFIT_PERCENT
-------------------- ------------------------------ --------------
UNIT_MEASURE         QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
-------------------- ----------- ---------- ---------- ----------
P07865               1.22FLOPPIES                              5
PIECE                        100        525        500         20

P07868               KEYBOARDS                                 2
PIECE                         10       3150       3050          3

P07885               CD DRIVE                                  3
PIECE                         10       5250       5100          3
```

Run SQL Command Line

```
P00001                1.44FLOPPIES                                  5
PIECE                        100        1150         500          20

P03453                MONITORS                                      6
PIECE                         10       12000       11200           3

P06734                MOUSE                                         5
PIECE                         20        1050         500           5


PRODUCT_NO            DESCRIPTION                    PROFIT_PERCENT
--------------------- ------------------------------ --------------
UNIT_MEASURE          QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
--------------------- ----------- ---------- ---------- ----------
P07865                1.22FLOPPIES                                  5
PIECE                        100         525         500          20

P07868                KEYBOARDS                                     2
PIECE                         10        3150        3050           3

P07885                CD DRIVE                                      3
PIECE                         10        5250        5100           3


PRODUCT_NO            DESCRIPTION                    PROFIT_PERCENT
--------------------- ------------------------------ --------------
UNIT_MEASURE          QTY_ON_HAND SELL_PRICE COST_PRICE    REORDER
--------------------- ----------- ---------- ---------- ----------
P07965                540 HDD                                       4
PIECE                         10        8400        8000           3

P07975                1.44 DRIVE                                    5
PIECE                         10        1050        1000           3

P08865                1.22 DRIVE                                    5
PIECE                          2        1050        1000           3


9 rows selected.
```

```
SQL> SELECT SUM(REORDER) FROM PRODUCT_MASTER;

SUM(REORDER)
------------
          63
```

x. Calculate the average price of all the products.

Run SQL Command Line

```
SQL> SELECT AVG(COST_PRICE) FROM PRODUCT_MASTER;

AVG(COST_PRICE)
---------------
     3427.77778
```

xi. Calculate the minimum price of products.

```
SQL> SELECT MIN(COST_PRICE) FROM PRODUCT_MASTER;

MIN(COST_PRICE)
---------------
           500
```

xii. Determine the maximum and minimum prices . Rename the tittle as ‘max_price’ and min_price respectively.

```
SQL> SELECT MAX(COST_PRICE) MAX_PRICE,MIN(COST_PRICE) MIN_PRICE FROM PRODUCT_MASTER;

 MAX_PRICE  MIN_PRICE
---------- ----------
     11200        500
```

xiii. Count the number of products having price greater than or equal to 1500.

```
SQL> SELECT COUNT(PRODUCT_NO) FROM PRODUCT_MASTER WHERE SELL_PRICE>=1500;

COUNT(PRODUCT_NO)
-----------------
                4
```

USE OF PRIMARY KEY -

Run SQL Command Line

```
SQL> ALTER TABLE Client_Master
  2  MODIFY Client_no varchar(6) PRIMARY KEY;

Table altered.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT NAME                 ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                       CITY            STATE              PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
1      IVAN
                               BOMBAY          MAHARASHTRA         400054
      1000

2      VANDANA
                               MADRAS          TAMILNADU           780001
         0

CLIENT NAME                 ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                       CITY            STATE              PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------

3      PRAMADA
                               BOMBAY          MAHARASHTRA         400057
      5000

4      BASU
                               BOMBAY          MAHARASHTRA         400056

CLIENT NAME                 ADDRESS1
------ -------------------- ------------------------------
ADDRESS2                       CITY            STATE              PINCODE
------------------------------ --------------- --------------- ----------
   BAL_DUE
----------
         0

5      RAVI
                               BOMBAY                              100001
      2000

6      RUKMINI

CLIENT NAME                 ADDRESS1
```

```
CLIENT NAME                ADDRESS1
------ ------------------- ------------------------------
ADDRESS2                      CITY            STATE              PINCODE
----------------------------- --------------- --------------- ----------
   BAL_DUE
----------
         0

5      RAVI
                              BOMBAY                              100001
      2000

6      RUKMINI

CLIENT NAME                ADDRESS1
------ ------------------- ------------------------------
ADDRESS2                      CITY            STATE              PINCODE
----------------------------- --------------- --------------- ----------
   BAL_DUE
----------
                              BOMBAY          MAHARASHTRA         400050
         0


6 rows selected.

SQL>
```

Experiment No.3

Aim: To implement the restrictions/constraints on the table

Question.2 Create the following tables:

i. **Sales_master**

Columnname	Datatype	Size	Attributes
Salesman_no	varchar2	6	Primary key/first letter must start with 's'
Sal_name	varchar2	20	Not null
Address	varchar2		Not null
City	varchar2	20	
State	varchar2	20	
Pincode	Number	6	
Sal_amt	Number	8,2	Not null, cannot be 0
Tgt_to_get	Number	6,2	Not null, cannot be 0
Ytd_sales	Number	6,2	Not null, cannot be 0
Remarks	Varchar2	30	

ii. **Sales_order**

Columnname	Datatype	Size	Attributes
S_order_no	varchar2	6	Primary/first letter must be 0
S_order_date	Date	6	Primary key reference clientno of client_master table
Client_no	Varchar2	25	
Dely_add	Varchar2	6	
Salesman_no	Varchar2	6	Foreign key references salesman_no of salesman_master table
Dely_type	Char	1	Delivery part(p)/full(f),default f
Billed_yn	Char	1	
Dely_date	Date		Can not be lessthan s_order_date
Order_status	Varchar2	10	Values ('in process';'fulfilled';back order';'canceled

I. **Sales_order_details**

Column	Datatype	Size	Attributes
S_order_no	Varchar2	6	Primary key/foreign key references s_order_no of sales_order
Product_no	Varchar2	6	Primary key/foreign key references product_no of product_master
Qty_order	Number	8	
Qty_disp	Number	8	

Product_rate	Number	10,2	

Insert the following data into their respective tables using insert statement:

Data for sales_man master table

Salesman_no	**Salesman name**	**Address**	**City**	**Pin code**	**State**	**Salamt**	**Tgt_to_get**	**Ytd Sales**	**Remark**
500001	Kiran	A/14 worli	Bombay	400002	Mah	3000	100	50	Good
500002	Manish	65,nariman	Bombay	400001	Mah	3000	200	100	Good
500003	Ravi	P-7 Bandra	Bombay	400032	Mah	3000	200	100	Good
500004	Ashish	A/5 Juhu	Bombay	400044	Mah	3500	200	150	Good

(i) **Data for salesorder table:**

S_orderno	S_orderdate	Client no	Dely type	Bill yn	Salesman no	Delay date	Orderstatus
019001	12-jan-96	0001	F	N	50001	20-jan-96	Ip
019002	25-jan-96	0002	P	N	50002	27-jan-96	C
016865	18-feb-96	0003	F	Y	500003	20-feb-96	F
019003	03-apr-96	0001	F	Y	500001	07-apr-96	F
046866	20-may-96	0004	P	N	500002	22-may-96	C
010008	24-may-96	0005	F	N	500004	26-may-96	Ip

(ii) **Data for sales_order_details table:**

S_order no	Product no	Qty ordered	Qty disp	Product_rate
019001	P00001	4	4	525
019001	P07965	2	1	8400
019001	P07885	2	1	5250
019002	P00001	10	0	525
046865	P07868	3	3	3150
046865	P07885	10	10	5250
019003	P00001	4	4	1050
019003	P03453	2	2	1050
046866	P06734	1	1	12000
046866	P07965	1	0	8400
010008	P07975	1	0	1050
010008	P00001	10	5	525

SALES_MASTER TABLE:

Run SQL Command Line

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Feb 14 09:32:57 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> CONNECT
Enter user-name: SYSTEM
Enter password:
Connected.
SQL> CREATE TABLE SALES_MASTER
  2  (SALESMAN_N0 VARCHAR2 (6) PRIMARY KEY,
  3  SAL_NAME VARCHAR2 (20),
  4  ADDRESS VARCHAR2 (20),
  5  CITY VARCHAR2 (20),
  6  STATE VARCHAR2 (20),
  7  PINCODE NUMBER (6),
  8  SAL_AMT NUMBER (8.2),
  9  TGT_TO_GET NUMBER (6.2),
 10  YTD_SALES NUMBER (6.2),
 11  REMARKS VARCHAR2 (30));
SAL_AMT NUMBER (8.2),
                *
ERROR at line 8:
ORA-02017: integer value required


SQL> ED
Wrote file afiedt.buf

  1  CREATE TABLE SALES_MASTER
  2  (SALESMAN_N0 VARCHAR2 (6) PRIMARY KEY,
  3  SAL_NAME VARCHAR2 (20),
  4  ADDRESS VARCHAR2 (20),
  5  CITY VARCHAR2 (20),
  6  STATE VARCHAR2 (20),
  7  PINCODE NUMBER (6),
  8  SAL_AMT NUMBER (8,2),
  9  TGT_TO_GET NUMBER (6,2),
 10  YTD_SALES NUMBER (6,2),
 11* REMARKS VARCHAR2 (30))
SQL> /

Table created.
```

Run SQL Command Line

```
Table created.

SQL> DESC SALES_MASTER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SALESMAN_N0                               NOT NULL VARCHAR2(6)
 SAL_NAME                                           VARCHAR2(20)
 ADDRESS                                            VARCHAR2(20)
 CITY                                               VARCHAR2(20)
 STATE                                              VARCHAR2(20)
 PINCODE                                            NUMBER(6)
 SAL_AMT                                            NUMBER(8,2)
 TGT_TO_GET                                         NUMBER(6,2)
 YTD_SALES                                          NUMBER(6,2)
 REMARKS                                            VARCHAR2(30)
```

INSERTION IN SALES_MASTER –

Run SQL Command Line

```
SQL> INSERT INTO SALES_MASTER VALUES(500001,'KIRAN','A/14 WORLI','BOMBAY','MAHARASHTRA',400002,3000,100,50,'GOOD');

1 row created.

SQL> INSERT INTO SALES_MASTER VALUES(500002,'MANISH','65 NARIMAN','BOMBAY','MAHARASHTRA',400001,3000,200,100,'GOOD');

1 row created.

SQL> INSERT INTO SALES_MASTER VALUES(500003,'RAVI','P-7 BANDRA','BOMBAY','MAHARASHTRA',400032,3000,200,100,'GOOD');

1 row created.

SQL> INSERT INTO SALES_MASTER VALUES(500004,'ASHISH','A/5 JUHU','BOMBAY','MAHARASHTRA',400002,3500,200,150,'GOOD');

1 row created.
```

Run SQL Command Line

```
SQL> INSERT INTO SALES_MASTER VALUES(500003,'RAVI','P-7 BANDRA','BOMBAY','MAHARASHTRA',400032,3000,200,100,'GOOD');

1 row created.

SQL> INSERT INTO SALES_MASTER VALUES(500004,'ASHISH','A/5 JUHU','BOMBAY','MAHARASHTRA',400002,3500,200,150,'GOOD');

1 row created.

SQL> SELECT * FROM SALES_MASTER;

SALESM SAL_NAME             ADDRESS             CITY
------ -------------------- ------------------- --------------------
STATE                   PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------
500001 KIRAN                A/14 WORLI          BOMBAY
MAHARASHTRA             400002       3000        100         50
GOOD

500002 MANISH               65 NARIMAN          BOMBAY
MAHARASHTRA             400001       3000        200        100
GOOD

SALESM SAL_NAME             ADDRESS             CITY
------ -------------------- ------------------- --------------------
STATE                   PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------

500003 RAVI                 P-7 BANDRA          BOMBAY
MAHARASHTRA             400032       3000        200        100
GOOD

500004 ASHISH               A/5 JUHU            BOMBAY
MAHARASHTRA             400002       3500        200        150

SALESM SAL_NAME             ADDRESS             CITY
------ -------------------- ------------------- --------------------
STATE                   PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------
GOOD


SQL>
```

AFTER ALL ATTRIBUTES:-

Run SQL Command Line

```
SQL> ALTER TABLE SALES_MASTER
  2  MODIFY SAL_NAME VARCHAR2(20) NOT NULL;

Table altered.

SQL> ALTER TABLE SALES_MASTER
  2   MODIFY ADDRESS VARCHAR2(20) NOT NULL;

Table altered.

SQL> ALTER TABLE SALES_MASTER
  2  MODIFY SAL_AMT NUMBER(8,2) NOT NULL;

Table altered.

SQL> ALTER TABLE SALES_MASTER
  2   MODIFY TGT_TO_GET NUMBER(6,2) NOT NULL;

Table altered.

SQL> ALTER TABLE SALES_MASTER
  2  MODIFY YTD_SALES NUMBER(6,2) NOT NULL;

Table altered.

SQL> SELECT * FROM SALES_MASTER;

SALESM SAL_NAME             ADDRESS              CITY
------ -------------------- -------------------- --------------------
STATE                   PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------
500001 KIRAN                A/14 WORLI           BOMBAY
MAHARASHTRA             400002       3000        100         50
GOOD

500002 MANISH               65 NARIMAN           BOMBAY
MAHARASHTRA             400001       3000        200        100
GOOD

SALESM SAL_NAME             ADDRESS              CITY
------ -------------------- -------------------- --------------------
STATE                   PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------
```

Run SQL Command Line

```
SQL> ALTER TABLE SALES_MASTER
  2   MODIFY TGT_TO_GET NUMBER(6,2) NOT NULL;

Table altered.

SQL> ALTER TABLE SALES_MASTER
  2  MODIFY YTD_SALES NUMBER(6,2) NOT NULL;

Table altered.

SQL> SELECT * FROM SALES_MASTER;

SALESM SAL_NAME             ADDRESS              CITY
------ -------------------- -------------------- --------------------
STATE                PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------
500001 KIRAN                A/14 WORLI           BOMBAY
MAHARASHTRA             400002       3000        100         50
GOOD

500002 MANISH               65 NARIMAN           BOMBAY
MAHARASHTRA             400001       3000        200        100
GOOD

SALESM SAL_NAME             ADDRESS              CITY
------ -------------------- -------------------- --------------------
STATE                PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------

500003 RAVI                 P-7 BANDRA           BOMBAY
MAHARASHTRA             400032       3000        200        100
GOOD

500004 ASHISH               A/5 JUHU             BOMBAY
MAHARASHTRA             400002       3500        200        150

SALESM SAL_NAME             ADDRESS              CITY
------ -------------------- -------------------- --------------------
STATE                PINCODE    SAL_AMT TGT_TO_GET  YTD_SALES
-------------------- ---------- ---------- ---------- ----------
REMARKS
------------------------------
GOOD
```

SALES_ORDER TABLE:

```
SQL> ED
Wrote file afiedt.buf

  1  CREATE TABLE SALES_ORDER
  2  (S_ORDER_NO VARCHAR2 (6) PRIMARY KEY,
  3  S_ORDER_DATE DATE,
  4  CLIENT_NO VARCHAR2 (25),
  5  DELY_ADD VARCHAR2 (6),
  6  SALESMAN_NO VARCHAR2 (6),
  7  DELY_TYPE CHAR (1) DEFAULT 'F',
  8  BILLED_YN CHAR (1),
  9  DELY_DATE DATE,
 10* ORDER_STATUS VARCHAR2 (10))
SQL> /

Table created.
```

```
SQL> DESC SALES_ORDER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 S_ORDER_NO                                NOT NULL VARCHAR2(6)
 S_ORDER_DATE                                       DATE
 CLIENT_NO                                          VARCHAR2(25)
 DELY_ADD                                           VARCHAR2(6)
 SALESMAN_NO                                        VARCHAR2(6)
 DELY_TYPE                                          CHAR(1)
 BILLED_YN                                          CHAR(1)
 DELY_DATE                                          DATE
 ORDER_STATUS                                       VARCHAR2(10)

SQL>
```

INSERTION IN SALES_ORDER –

Run SQL Command Line

```
SQL> INSERT INTO SALES_ORDER VALUES(019001,'12-JAN-96',0001,NULL,50001,'F','N','20-JAN-96','IP');

1 row created.

SQL> INSERT INTO SALES_ORDER VALUES(019002,'25-JAN-96',0002,NULL,50002,'P','N','27-JAN-96','C');

1 row created.

SQL> INSERT INTO SALES_ORDER VALUES(016865,'18-FEB-96',0003,NULL,500003,'F','Y','20-FEB-96','F');

1 row created.

SQL> INSERT INTO SALES_ORDER VALUES(019003,'03-APR-96',0001,NULL,500001,'F','Y','07-APR-96','F');

1 row created.

SQL> INSERT INTO SALES_ORDER VALUES(046866,'20-MAY-96',0004,NULL,500002,'P','N','22-MAY-96','C');

1 row created.

SQL> INSERT INTO SALES_ORDER VALUES(010008,'24-MAY-96',0005,NULL,500004,'F','N','26-MAY-96','IP');

1 row created.
```

Run SQL Command Line

```
1 row created.

SQL> INSERT INTO SALES_ORDER VALUES(010008,'24-MAY-96',0005,NULL,500004,'F','N','26-MAY-96','IP');

1 row created.

SQL> SELECT * FROM SALES_ORDER;

S_ORDE S_ORDER_D CLIENT_NO                DELY_A SALESM D B DELY_DATE
------ --------- ------------------------ ------ ------ - - ---------
ORDER_STAT
----------
19001  12-JAN-96 1                             50001  F N 20-JAN-96
IP

19002  25-JAN-96 2                             50002  P N 27-JAN-96
C

16865  18-FEB-96 3                             500003 F Y 20-FEB-96
F


S_ORDE S_ORDER_D CLIENT_NO                DELY_A SALESM D B DELY_DATE
------ --------- ------------------------ ------ ------ - - ---------
ORDER_STAT
----------
19003  03-APR-96 1                             500001 F Y 07-APR-96
F

46866  20-MAY-96 4                             500002 P N 22-MAY-96
C

10008  24-MAY-96 5                             500004 F N 26-MAY-96
IP


6 rows selected.

SQL>
```

SALES_ORDER_DETAILS TABLE:

Run SQL Command Line

```
 ORDER_STATUS                                       VARCHAR2(10)

SQL> CREATE TABLE SALES_ORDER_DETAILS
  2  (S_ORDER_NO VARCHAR2 (6),
  3  PRODUCT_NO VARCHAR2 (6),
  4  QTY_ORDER NUMBER (8),
  5  QTY_DISP NUMBER (8),
  6  PRODUCT_RATE NUMBER (10,2));

Table created.

SQL> DESC SALES_ORDER_DETAILS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 S_ORDER_NO                                         VARCHAR2(6)
 PRODUCT_NO                                         VARCHAR2(6)
 QTY_ORDER                                          NUMBER(8)
 QTY_DISP                                           NUMBER(8)
 PRODUCT_RATE                                       NUMBER(10,2)

SQL>
```

Run SQL Command Line

```
SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(019001,'P00001',4,4,525);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(019001,'P07965',2,1,8400);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(019001,'P07885',2,1,5250);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(019002,'P00001',10,0,525);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(046865,'P07868',3,3,3150);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(046865,'PO7885',10,10,5250);

1 row created.

SQL> ED
Wrote file afiedt.buf

  1* INSERT INTO SALES_ORDER_DETAILS VALUES(046865,'P07885',10,10,5250)
SQL> /

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(019003,'P00001',4,4,1050);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(019003,'P03453',2,2,1050);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(046866,'P06734',1,1,12000);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(046866,'P07965',1,0,8400);

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(010008,'PO7975',1,0,1050);
```

Run SQL Command Line

```
SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(010008,'P07975',1,0,1050);

1 row created.

SQL> ED
Wrote file afiedt.buf

  1* INSERT INTO SALES_ORDER_DETAILS VALUES(010008,'P07975',1,0,1050)
SQL> /

1 row created.

SQL> INSERT INTO SALES_ORDER_DETAILS VALUES(010008,'P00001',10,5,525);

1 row created.
```

Run SQL Command Line

```
SQL> SELECT * FROM SALES_ORDER_DETAILS;

S_ORDE PRODUC  QTY_ORDER   QTY_DISP PRODUCT_RATE
------ ------ ---------- ---------- ------------
19001  P00001          4          4          525
19001  P07965          2          1         8400
19001  P07885          2          1         5250
19002  P00001         10          0          525
46865  P07868          3          3         3150
46865  P07885         10         10         5250
19003  P00001          4          4         1050
19003  P03453          2          2         1050
46866  P06734          1          1        12000
46866  P07965          1          0         8400
10008  P07975          1          0         1050

S_ORDE PRODUC  QTY_ORDER   QTY_DISP PRODUCT_RATE
------ ------ ---------- ---------- ------------
10008  P00001         10          5          525

12 rows selected.
```

Experiment No.4

Aim: To Alter the structure of the table

Q1. Create the following tables:

Challan_Header

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key
s_order_no	varchar2	6	Foreign key references s_order_no of sales_order table
challan_date	date		not null
billed_yn	char	1	values ('Y','N'). Default 'N'

Table Name : Challan_Details

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key
Product No	Varchar2	10	Foreign Key references Product_no of product_master
Qty_disp	number	4,2	not null

Q2. Insert the following values into the challan header and challan_details tables:

(i)

Challan No	S_order No	Challan Date	Billed
CH9001	019001	12-DEC-95	Y
CH865	046866	12-NOV-95	Y
CH3965	010008	12-OCT-95	Y

Data for challan_details table

Challan No	Product No	Qty Disp
CH9001	P00001	4
CH9001	P07965	1
CH9001	P07885	1
CH6865	P07868	3
CH6865	P03453	4
CH6865	P00001	10
CH3965	P00001	5
CH3965	P07975	2

CHALLAN HEADER TABLE –

Run SQL Command Line

```
SQL> create table challan_header
  2  (challan_no varchar2(6) primary key,
  3  s_order_no varchar2(6) references sales_order,
  4  challan_date date not null,
  5  billed_yn varchar2(1) default ' N' ,
  6  check (billed_yn in ('Y','N'));
check (billed_yn in ('Y','N'))
                     *
ERROR at line 6:
ORA-00907: missing right parenthesis


SQL> ed
Wrote file afiedt.buf

  1  create table challan_header
  2  (challan_no varchar2(6) primary key,
  3  s_order_no varchar2(6) references sales_order,
  4  challan_date date not null,
  5  billed_yn varchar2(1) default ' N' ,
  6* check (billed_yn in ('Y','N')))
SQL> /
billed_yn varchar2(1) default ' N' ,
                      *
ERROR at line 5:
ORA-01401: inserted value too large for column


SQL> ed
Wrote file afiedt.buf

  1  create table challan_header
  2  (challan_no varchar2(6) primary key,
  3  s_order_no varchar2(6) references sales_order,
  4  challan_date date not null,
  5  billed_yn varchar2(1) default 'N' ,
  6* check (billed_yn in ('Y','N')))
SQL> /

Table created.

SQL> desc challan_header;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CHALLAN_NO                                NOT NULL VARCHAR2(6)
 S_ORDER_NO                                         VARCHAR2(6)
 CHALLAN_DATE                              NOT NULL DATE
 BILLED_YN                                          VARCHAR2(1)
```

```
SQL> INSERT INTO CHALLAN_HEADER VALUES('CH9001','019001','12-DEC-95','Y');

1 row created.

SQL> INSERT INTO CHALLAN_HEADER VALUES('CH865','046866','12-NOV-95','Y');

1 row created.

SQL> INSERT INTO CHALLAN_HEADER VALUES('CH3965','010008','12-OCT-95','Y');

1 row created.

SQL> SELECT * FROM CHALLAN_HEADER;

CHALLA S_ORDE CHALLAN_D B
------ ------ --------- -
CH9001 019001 12-DEC-95 Y
CH865  046866 12-NOV-95 Y
CH3965 010008 12-OCT-95 Y
```

CHALLAN_DETAILS TABLE-

Run SQL Command Line

```
SQL> create table challan_details(
  2  challan_no varchar(6),
  3  product_no varchar(8) references product_master,
  4  qty_disp number(4,2) not null,
  5  primary key(challan_no,product_no));

Table created.

SQL> desc challan_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CHALLAN_NO                                NOT NULL VARCHAR2(6)
 PRODUCT_NO                                NOT NULL VARCHAR2(8)
 QTY_DISP                                  NOT NULL NUMBER(4,2)
```

```
SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH9001','P00001',4);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH9001','P07965',1);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH9001','P07885',1);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH6865','P07868',3);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH6865','P03453',4);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH6865','P00001',10);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH3965','P00001',5);

1 row created.

SQL> INSERT INTO CHALLAN_DETAILS VALUES('CH3965','P07975',2);

1 row created.

SQL> SELECT * FROM CHALLAN_DETAILS;

CHALLA PRODUCT_   QTY_DISP
------ -------- ----------
CH9001 P00001            4
CH9001 P07965            1
CH9001 P07885            1
CH6865 P07868            3
CH6865 P03453            4
CH6865 P00001           10
CH3965 P00001            5
CH3965 P07975            2
```

Objective – Answer the following Queries

Q1. Make the primary key to client_no in client_master.

Run SQL Command Line

```
SQL>  alter table client_master
  2  add primary key(client_no);

Table altered.

SQL> desc client_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                 NOT NULL VARCHAR2(6)
 NAME                                               VARCHAR2(20)
 ADDRESS1                                           VARCHAR2(30)
 ADDRESS2                                           VARCHAR2(30)
 CITY                                               VARCHAR2(15)
 STATE                                              VARCHAR2(15)
 PINCODE                                            NUMBER(6)
 BAL_DUE                                            NUMBER(10,2)

SQL>
```

Q2. Add a new column phone_no in the client_master table.

Run SQL Command Line

```
 BAL_DUE                                            NUMBER(10,2)

SQL> alter table client_master
  2  add phone_no number(10);

Table altered.

SQL> desc client_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                 NOT NULL VARCHAR2(6)
 NAME                                               VARCHAR2(20)
 ADDRESS1                                           VARCHAR2(30)
 ADDRESS2                                           VARCHAR2(30)
 CITY                                               VARCHAR2(15)
 STATE                                              VARCHAR2(15)
 PINCODE                                            NUMBER(6)
 BAL_DUE                                            NUMBER(10,2)
 PHONE_NO                                           NUMBER(10)

SQL>
```

Q3. Add the not null constraint in the product_master table with the columns description, profit percent , sell price and cost price.

Run SQL Command Line

```
 PHONE_NO                                           NUMBER(10)

SQL> alter table product_master modify (description not null, profit_percent not null, sell_price not null,
  2  cost_price not null);

Table altered.
```

Run SQL Command Line

```
SQL> desc product_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 PRODUCT_NO                                NOT NULL VARCHAR2(20)
 DESCRIPTION                               NOT NULL VARCHAR2(30)
 PROFIT_PERCENT                            NOT NULL NUMBER(20)
 UNIT_MEASURE                                       VARCHAR2(20)
 QTY_ON_HAND                                        NUMBER(20)
 SELL_PRICE                                NOT NULL NUMBER(20)
 COST_PRICE                                NOT NULL NUMBER(20)
 REORDER                                            NUMBER(30)
```

Q4. Change the size of client_no field in the client_master table.

Run SQL Command Line

```
Commit complete.

SQL> alter table client_master
  2  modify client_no varchar2(25);

Table altered.

SQL> desc client_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                 NOT NULL VARCHAR2(25)
 NAME                                               VARCHAR2(20)
 ADDRESS1                                           VARCHAR2(30)
 ADDRESS2                                           VARCHAR2(30)
 CITY                                               VARCHAR2(15)
 STATE                                              VARCHAR2(15)
 PINCODE                                            NUMBER(6)
 BAL_DUE                                            NUMBER(10,2)
 PHONE_NO                                           NUMBER(10)

SQL>
```

Q5. Select product_no, description where profit percent is between 20 and 30 both inclusive.

Run SQL Command Line

```
SQL> select product_no, description from product_master where profit_percent >20 and profit_percent <30;

no rows selected
```

Experiment No.5

Aim: to implement the concept of Joins

A. Consider the following schema for a Library Database:

BOOK (*Book_id, Title, Publisher_Name, Pub_Year*)

BOOK_AUTHORS (Book_id, Author_*Name*)

PUBLISHER (*Name, Address, Phone*)

BOOK_COPIES (*Book_id, Branch_id, No-of_Copies*)

CARD(Card_No)

BOOK_LENDING (*Book_id, Branch_id, Card_No, Date_Out, Due_Date*)

LIBRARY_BRANCH (*Branch_id, Branch_Name, Address*)

1- PUBLISHER TABLE –

Run SQL Command Line

```
SQL> CREATE TABLE PUBLISHER
  2  (NAME VARCHAR2 (20) PRIMARY KEY,
  3  PHONE INTEGER,
  4  ADDRESS VARCHAR2 (20));

Table created.

SQL> DESC PUBLISHER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL VARCHAR2(20)
 PHONE                                              NUMBER(38)
 ADDRESS                                            VARCHAR2(20)

SQL> INSERT INTO PUBLISHER VALUES ('MCGRAW-HILL', 9989076587, 'BANGALORE');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('PEARSON', 9889076565, 'NEWDELHI');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('RANDOM HOUSE', 7455679345, 'HYDRABAD');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('HACHETTE LIVRA', 8970862340, 'CHENNAI');

1 row created.

SQL> INSERT INTO PUBLISHER VALUES ('GRUPO PLANETA', 7756120238, 'BANGALORE');

1 row created.

SQL> SELECT * FROM PUBLISHER;

NAME                      PHONE ADDRESS
-------------------- ---------- --------------------
MCGRAW-HILL           9989076587 BANGALORE
PEARSON               9889076565 NEWDELHI
RANDOM HOUSE          7455679345 HYDRABAD
HACHETTE LIVRA        8970862340 CHENNAI
GRUPO PLANETA         7756120238 BANGALORE

SQL>
```

2- BOOK TABLE-

Run SQL Command Line

```
SQL> CREATE TABLE BOOK
  2   (BOOK_ID INTEGER PRIMARY KEY,
  3  TITLE VARCHAR2 (20),
  4  PUB_YEAR VARCHAR2 (20),
  5  PUBLISHER_NAME REFERENCES PUBLISHER (NAME) ON DELETE CASCADE);

Table created.

SQL> DESC BOOK;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOK_ID                                   NOT NULL NUMBER(38)
 TITLE                                              VARCHAR2(20)
 PUB_YEAR                                           VARCHAR2(20)
 PUBLISHER_NAME                                     VARCHAR2(20)

SQL> INSERT INTO BOOK VALUES (1,'DBMS','JAN-2017', 'MCGRAW-HILL');

1 row created.

SQL> INSERT INTO BOOK VALUES (2,'ADBMS','JUNE-2016', 'MCGRAW-HILL');

1 row created.

SQL> INSERT INTO BOOK VALUES (3,'CN','SEP-2019','PEARSON');

1 row created.

SQL> INSERT INTO BOOK VALUES (4,'CG','SEP-2016','GRUPO PLANETA');

1 row created.

SQL> INSERT INTO BOOK VALUES (5,'OS','MAY-2016','PEARSON');

1 row created.

SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                PUB_YEAR             PUBLISHER_NAME
---------- -------------------- -------------------- --------------------
         1 DBMS                 JAN-2017             MCGRAW-HILL
         2 ADBMS                JUNE-2016            MCGRAW-HILL
         3 CN                   SEP-2019             PEARSON
         4 CG                   SEP-2016             GRUPO PLANETA
         5 OS                   MAY-2016             PEARSON
```

3- BOOK_AUTHORS TABLE –

Run SQL Command Line

```
SQL> CREATE TABLE BOOK_AUTHORS
  2  (AUTHOR_NAME VARCHAR2 (20),
  3  BOOK_ID REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
  4  PRIMARY KEY(BOOK_ID, AUTHOR_NAME));

Table created.

SQL> DESC BOOK_AUTHORS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 AUTHOR_NAME                               NOT NULL VARCHAR2(20)
 BOOK_ID                                   NOT NULL NUMBER(38)

SQL> INSERT INTO BOOK_AUTHORS VALUES ('NAVATHE', 1);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('NAVATHE', 2);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('TANENBAUM', 3);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('EDWARD ANGEL', 4);

1 row created.

SQL> INSERT INTO BOOK_AUTHORS VALUES ('GALVIN',5);

1 row created.
```

Run SQL Command Line

```
         5 OS                    MAY-2016              PEA

SQL> SELECT * FROM BOOK_AUTHORS;

AUTHOR_NAME             BOOK_ID
-------------------- ----------
NAVATHE                       1
NAVATHE                       2
TANENBAUM                     3
EDWARD ANGEL                  4
GALVIN                        5
```

4- LIBRARY_BRANCH TABLE –

Run SQL Command Line

```
SQL> CREATE TABLE LIBRARY_BRANCH
  2  (BRANCH_ID INTEGER PRIMARY KEY,
  3  BRANCH_NAME VARCHAR2 (50),
  4  ADDRESS VARCHAR2 (50));

Table created.

SQL> DESC LIBRARY_BRANCH;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRANCH_ID                                 NOT NULL NUMBER(38)
 BRANCH_NAME                                        VARCHAR2(50)
 ADDRESS                                            VARCHAR2(50)

SQL> INSERT INTO LIBRARY_BRANCH VALUES (10,'RR NAGAR','BANGALORE');

1 row created.

SQL>  INSERT INTO LIBRARY_BRANCH VALUES (11, 'RNSIT','BANGALORE');

1 row created.

SQL>  INSERT INTO LIBRARY_BRANCH VALUES (12,'RAJAJI NAGAR','BANGALORE');

1 row created.

SQL>  INSERT INTO LIBRARY_BRANCH VALUES (13,'NITTE','MANGALORE');

1 row created.

SQL>  INSERT INTO LIBRARY_BRANCH VALUES (14,'MANIPAL','UDUPI');

1 row created.
```

```
SQL> SELECT * FROM LIBRARY_BRANCH;

 BRANCH_ID BRANCH_NAME
---------- --------------------------------------------------
ADDRESS
--------------------------------------------------
        10 RR NAGAR
BANGALORE

        11 RNSIT
BANGALORE

        12 RAJAJI NAGAR
BANGALORE


 BRANCH_ID BRANCH_NAME
---------- --------------------------------------------------
ADDRESS
--------------------------------------------------
        13 NITTE
MANGALORE

        14 MANIPAL
UDUPI


SQL>
```

5- BOOK_COPIES TABLE-

Run SQL Command Line

```
SQL> CREATE TABLE BOOK_COPIES
  2  (NO_OF_COPIES INTEGER,
  3  BOOK_ID REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
  4  BRANCH_ID REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON DELETE CASCADE,
  5  PRIMARY KEY (BOOK_ID, BRANCH_ID));

Table created.

SQL> INSERT INTO BOOK_COPIES VALUES (10, 1, 10);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (5, 1, 11);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (2, 2, 12);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (5, 2, 13);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (7, 3, 14);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (1, 5, 10);

1 row created.

SQL> INSERT INTO BOOK_COPIES VALUES (3, 4, 11);

1 row created.

SQL> SELECT * FROM BOOK_COPIES;

NO_OF_COPIES    BOOK_ID  BRANCH_ID
------------ ---------- ----------
          10          1         10
           5          1         11
           2          2         12
           5          2         13
           7          3         14
           1          5         10
           3          4         11

7 rows selected.
```

```
SQL> DESC BOOK_COPIES;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NO_OF_COPIES                                       NUMBER(38)
 BOOK_ID                                   NOT NULL NUMBER(38)
 BRANCH_ID                                 NOT NULL NUMBER(38)

SQL>
```

6- CARD TABLE-

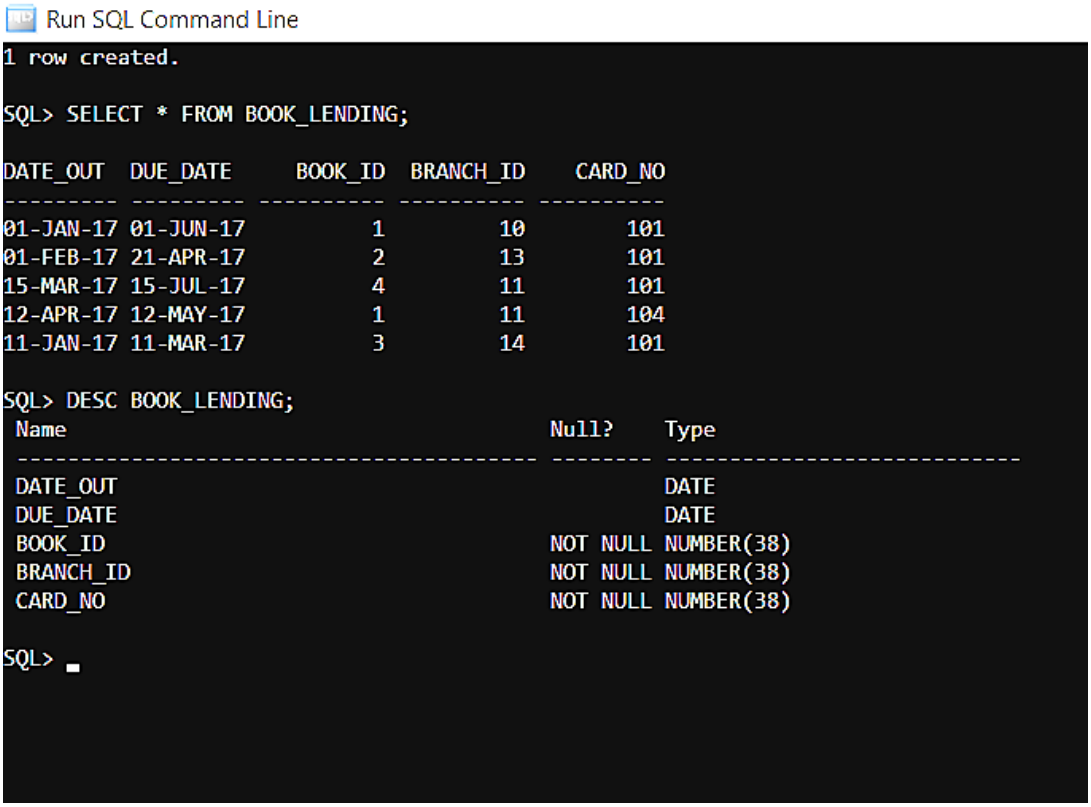
Run SQL Command Line

```
SQL> CREATE TABLE CARD
  2  (CARD_NO INTEGER PRIMARY KEY);

Table created.

SQL> INSERT INTO CARD VALUES (100);

1 row created.

SQL> INSERT INTO CARD VALUES (101);

1 row created.

SQL> INSERT INTO CARD VALUES (102);

1 row created.

SQL> INSERT INTO CARD VALUES (103);

1 row created.

SQL> INSERT INTO CARD VALUES (104);

1 row created.

SQL> SELECT * FROM CARD;

   CARD_NO
----------
       100
       101
       102
       103
       104

SQL> DESC CARD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CARD_NO                                   NOT NULL NUMBER(38)

SQL>
```

7- BOOK_LENDING TABLE-

Run SQL Command Line

```
SQL> CREATE TABLE BOOK_LENDING
  2  (DATE_OUT DATE,
  3  DUE_DATE DATE,
  4  BOOK_ID REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
  5  BRANCH_ID REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON DELETE CASCADE,
  6  CARD_NO REFERENCES CARD (CARD_NO) ON DELETE CASCADE,
  7  PRIMARY KEY (BOOK_ID, BRANCH_ID, CARD_NO));

Table created.

SQL> INSERT INTO BOOK_LENDING VALUES ('01-JAN-17','01-JUN-17', 1, 10, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('11-MAR-17','11-MAR-17', 3, 14, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('01-FEB-17','21-APR-17', 2, 13, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('15-MAR-17','15-JUL-17', 4, 11, 101);

1 row created.

SQL> INSERT INTO BOOK_LENDING VALUES ('12-APR-17','12-MAY-17', 1, 11, 104);

1 row created.

SQL> COMMIT;

Commit complete.

SQL> DELETE FROM BOOK_LENDING WHERE CARD_NO=14;

0 rows deleted.

SQL> DELETE FROM BOOK_LENDING WHERE DATE_OUT='11-MAR-17';

1 row deleted.

SQL>  INSERT INTO BOOK_LENDING VALUES ('11-JAN-17','11-MAR-17', 3, 14, 101);

1 row created.

SQL>
```


Run SQL Command Line
1 row created.
SQL> SELECT * FROM BOOK_LENDING;
DATE_OUT DUE_DATE BOOK_ID BRANCH_ID CARD_NO
01-JAN-17 01-JUN-17 1 10 101
01-FEB-17 21-APR-17 2 13 101
15-MAR-17 15-JUL-17 4 11 101
12-APR-17 12-MAY-17 1 11 104
11-JAN-17 11-MAR-17 3 14 101
SQL> DESC BOOK_LENDING;
Name Null? Type
DATE_OUT DATE
DUE_DATE DATE
BOOK_ID NOT NULL NUMBER(38)
BRANCH_ID NOT NULL NUMBER(38)
CARD_NO NOT NULL NUMBER(38)
SQL>

Write SQL queries to :

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.

Run SQL Command Line

```
SQL> SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME, A.AUTHOR_NAME, C.NO_OF_COPIES,
  2  L.BRANCH_ID
  3  FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_BRANCH L
  4  WHERE B.BOOK_ID=A.BOOK_ID
  5  AND B.BOOK_ID=C.BOOK_ID AND L.BRANCH_ID=C.BRANCH_ID;

   BOOK_ID TITLE                PUBLISHER_NAME       AUTHOR_NAME
---------- -------------------- -------------------- --------------------
NO_OF_COPIES  BRANCH_ID
------------ ----------
         1 DBMS                 MCGRAW-HILL          NAVATHE
          10         10

         1 DBMS                 MCGRAW-HILL          NAVATHE
           5         11

         2 ADBMS                MCGRAW-HILL          NAVATHE
           2         12


   BOOK_ID TITLE                PUBLISHER_NAME       AUTHOR_NAME
---------- -------------------- -------------------- --------------------
NO_OF_COPIES  BRANCH_ID
------------ ----------
         2 ADBMS                MCGRAW-HILL          NAVATHE
           5         13

         3 CN                   PEARSON              TANENBAUM
           7         14

         5 OS                   PEARSON              GALVIN
           1         10


   BOOK_ID TITLE                PUBLISHER_NAME       AUTHOR_NAME
---------- -------------------- -------------------- --------------------
NO_OF_COPIES  BRANCH_ID
------------ ----------
         4 CG                   GRUPO PLANETA        EDWARD ANGEL
           3         11


7 rows selected.

SQL>
```

2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.

Run SQL Command Line

```
SQL> SELECT CARD_NO FROM BOOK_LENDING
  2  WHERE DATE_OUT BETWEEN '01-JAN-2017' AND '01-JUL-2017'
  3  GROUP BY CARD_NO HAVING COUNT (*)>3
  4  /

   CARD_NO
----------
       101
```

3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.

Run SQL Command Line

```
SQL> DELETE FROM BOOK
  2  WHERE BOOK_ID=3;

1 row deleted.

SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE               PUB_YEAR            PUBLISHER_NAME
---------- ------------------- ------------------- --------------------
         1 DBMS                JAN-2017            MCGRAW-HILL
         2 ADBMS               JUNE-2016           MCGRAW-HILL
         4 CG                  SEP-2016            GRUPO PLANETA
         5 OS                  MAY-2016            PEARSON

SQL>
```

4. Create a view of all books and its number of copies that are currently available in the Library.

```
SQL> CREATE VIEW V_BOOKS AS SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
  2  FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH L
  3  WHERE B.BOOK_ID=C.BOOK_ID AND C.BRANCH_ID=L.BRANCH_ID;

View created.

SQL>
```

Run SQL Command Line

```
SQL> SELECT * FROM V_BOOKS;

   BOOK_ID TITLE                NO_OF_COPIES
---------- -------------------- ------------
         1 DBMS                           10
         1 DBMS                            5
         2 ADBMS                           2
         2 ADBMS                           5
         5 OS                              1
         4 CG                              3

6 rows selected.

SQL> _
```

5. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

EXPERIMENT NO. 6

OBJECTIVE: Answer the following Queries:

Answer the following queries -

1. Find out the product which has been sold to 'Ivan'.

```
SQL> SELECT P.Product_no, P.Description, S.Qty_order
  2  FROM client_master C, Product_master P, Sales_order_details S, Sales_order S2
  3  WHERE P.Product_no = S.Product_no
  4  AND S.S_order_no = S2.S_order_no
  5  AND S2.Client_no = C.Client_no
  6  AND C.Name like '%Ivan%';

PRODUC DESCRIPTION           QTY_ORDER
------ -------------------- ----------
P07885 CD drive                      2
P07965 540 HDD                       2
P00001 1.44floppies                  4
P03453 Monitors                      2
P00001 1.44floppies                  4

SQL>
```

2. Find out the product and their quantities that will have do delivered.

```
SQL> SELECT P.Product_no, S.Qty_order
  2  FROM Product_master P, Sales_order_details S
  3  WHERE P.Product_no = S.Product_no;

PRODUC  QTY_ORDER
------ ----------
P00001          4
P07965          2
P07885          2
P00001         10
P07868          3
P07885         10
P00001          4
P03453          2
P07965          1
P07975          1
P00001         10

11 rows selected.

SQL>
```

3. Find the product_no and description of moving products.

```
SQL> SELECT P.Product_no, P.Description
  2  FROM Product_Master P
  3  WHERE P.Description = 'Monitors' OR P.Description = 'Mouse' OR P.Description ='keyboards';

PRODUC DESCRIPTION
------ --------------------
P03453 Monitors
P06734 Mouse
P07868 keyboards

SQL>
```

4. Find out the names of clients who have purchased 'CD DRIVE'.

```
SQL> SELECT C.Name
  2  FROM client_master C, Product_master P, Sales_order_details S, Sales_order S2
  3  WHERE P.Product_no = S.Product_no
  4  AND S.S_order_no = S2.S_order_no
  5  AND S2.Client_no = C.Client_no
  6  AND P.Description = 'CD drive';

NAME
--------------------
Ivan

SQL>
```

5. List the product_no and s_order_no of customers ha ving qty ordered less than from the order details table for the product "1.44 floppies".

```
SQL> SELECT S.Product_no, S.S_order_no
  2  FROM Product_master P, Sales_order_details S
  3  WHERE P.Description = '1.44floppies'
  4  AND S.Qty_order < 5
  5  AND P.Product_no = S.Product_no;

PRODUC S_ORDE
------ ------
P00001 019001
P00001 019003

SQL>
```

6. Find the products and their quantities for the orders placed by "Vandan Saitwal" and "Ivan Bayross".

```
SQL> SELECT P.Product_no, P.Description, S.Qty_order
  2  FROM Product_master P, client_master C, Sales_order_details S, Sales_order S2
  3  WHERE P.Product_no = S.Product_no
  4  AND S.S_order_no = S2.S_order_no
  5  AND S2.Client_no = C.Client_no
  6  AND C.Name like '%Ivan%';

PRODUC DESCRIPTION           QTY_ORDER
------ ------------------- ----------
P07885 CD drive                     2
P07965 540 HDD                      2
P00001 1.44floppies                 4
P03453 Monitors                     2
P00001 1.44floppies                 4

SQL>
```

7. Find the products and their quantities for the orders placed by client_no "C00001" and "C00002".

```
SQL> SELECT P.Description, S.Qty_order
  2  FROM Product_master P, Sales_order_details S, Sales_order S2
  3  WHERE P.Product_no = S.Product_no
  4  AND S.S_order_no = S2.S_order_no
  5  And S2.Client_no IN ('0001','0002');

DESCRIPTION          QTY_ORDER
-------------------- ----------
1.44floppies                 4
540 HDD                      2
CD drive                     2
1.44floppies                10
1.44floppies                 4
Monitors                     2

6 rows selected.

SQL>
```

8. Find the order No, Client No and salesman No. where a client has been received by more than one salesman.

```
SQL> SELECT S_order_no, Client_no, Salesman_no
  2  FROM Sales_order
  3  WHERE Client_no IN
  4  (SELECT Client_no FROM Sales_order
  5  GROUP BY Client_no HAVING COUNT(Salesman_no) > 1);

S_ORDE CLIENT_NO                 SALESM
------ ------------------------- ------
019003 0001                      500001
019001 0001                      50001

SQL>
```

9. Display the s_order_date in the format "dd-mm-yy" e.g., "12- feb-96".

```
SQL> SELECT S_order_date from sales_order;

S_ORDER_D
---------
12-JAN-96
25-JAN-96
18-FEB-96
03-APR-96
20-MAY-96
24-MAY-96

6 rows selected.
```

10.Find the date, 15 days after date.

```
SQL> SELECT S_order_date+15 from sales_order;

S_ORDER_D
---------
27-JAN-96
09-FEB-96
04-MAR-96
18-APR-96
04-JUN-96
08-JUN-96

6 rows selected.
```

Experiment No.7

Objective:- To implement the concept of procedure.

PL/SQL Concepts- Stored Procedure:-
The PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks. It is just like procedures in other programming languages.
The procedure contains a header and a body.

Header: The header contains the name of the procedure and the parameters or variables passed to the procedure.

Body: The body contains a declaration section, execution section and exception section similar to a general PL/SQL block.

How to pass parameters in procedure:
When you want to create a procedure or function, you have to define parameters .There is three ways to pass parameters in procedure:

1. IN parameters: The IN parameter can be referenced by the procedure or function. The value of the parameter cannot be overwritten by the procedure or the function.
2. OUT parameters: The OUT parameter cannot be referenced by the procedure or function, but the value of the parameter can be overwritten by the procedure or function.
3. INOUT parameters: The INOUT parameter can be referenced by the procedure or function and the value of the parameter can be overwritten by the procedure or function.
NOTE: A procedure may or may not return any value.

Syntax for creating procedure:

```
CREATE [OR REPLACE] PROCEDURE procedure_name
   [(parameter [,parameter]) ]
IS
   [declaration_section]
BEGIN
   executable_section
[EXCEPTION   exception_section]
END [procedure_name];
```

OBJECTIVE:
TO CREATE A PROCEDURE .

```
SQL> create table user1
  2  (ID number (10) primary key,
  3  Name varchar2(100));

Table created.

SQL> create procedure insertuser1 (ID in Number, Name In Varchar2)
  2  IS
  3  BEGIN
  4  INSERT into user1 values (ID, Name);
  5  end;
  6  /

Procedure created.

SQL> EXECUTE insertuser1(101, 'Rahul');

PL/SQL procedure successfully completed.
```

```
SQL> select * from user1;

        ID NAME
---------- ------------------------------------------------------------
------------------------------

       101 Rahul
       102 Rahul

SQL>
```

```
SQL> begin
  2  insertuser1(102, 'Rahul');
  3  dbms_output.put_line('record inserted sucessfully');
  4  end;
  5  /

PL/SQL procedure successfully completed.
```

Experiment NO.8

Objective:- To implement the concept of functions.

PL/SQL Concepts- Functions-

The PL/SQL Function is very similar to PL/SQL Procedure. The main difference between procedure and a function is, a function must always return a value, and on the other hand a procedure may or may not return a value. Except this, all the other things of PL/SQL procedure are true for PL/SQL function too.

Syntax to create a function:

1. CREATE [OR REPLACE] FUNCTION function_name [parameters]
2. [(parameter_name [IN | OUT | IN OUT] type [, ...])]
3. RETURN return_datatype
4. {IS | AS}
5. BEGIN
6. < function_body >
7. END [function_name];

Here:

1. Function_name: specifies the name of the function.
2. [OR REPLACE] option allows modifying an existing function.
3. The optional parameter list contains name, mode and types of the parameters.
4. IN represents that value will be passed from outside and OUT represents that this parameter will be used to return a value outside of the procedure.

The function must contain a return statement.

5. RETURN clause specifies that data type you are going to return from the function.
6. Function_body contains the executable part.
7. The AS keyword is used instead of the IS keyword for creating a standalone function.

Objective:-
TO CREATE FUNCTION-

```
SQL> create function adder(n1 in number, n2 in number)
  2  return number
  3  is
  4  n3 number (8);
  5  begin
  6  n3:=n1+n2;
  7  return n3;
  8  end;
  9  /

Function created.
```

```
SQL> set serveroutput on;
SQL> declare
  2  n3 number(2);
  3  begin
  4  n3:=adder(11, 22);
  5  dbms_output.put_line('Addition is: ' || n3);
  6  end;
  7  /
Addition is: 33

PL/SQL procedure successfully completed.

SQL>
```

Experiment No.9

Objective:- To implement the concept of Cursor.

Cursor– We have seen how oracle executes an SQL statement. Oracle DBA uses a work area for its internal processing. This work area is private to SQL's operation and is called a **cursor.**
The data that is stored in the cursor is called the **Active Data set.** The size of the cursor in memory is the size required to hold the number of rows in the Active Data Set.

Explicit Cursor- You can explicitly declare a cursor to process the rows individually. A cursor declared by the user is called **Explicit Cursor.** For Queries that return more than one row, You must declare a cursor explicitly.
The data that is stored in the cursor is called the **Active Data set.** The size of the cursor in memory is the size required to hold the number of rows in the Active

Why use an Explicit Cursor- Cursor can be used when the user wants to process data one row at a time.

Explicit Cursor Management- The steps involved in declaring a cursor and manipulating data in the active data set are:-

- Declare a cursor that specifies the SQL select statement that you want to process.
- Open the Cursor.
- Fetch the data from the cursor one row at a time.
- Close the cursor.

Explicit Cursor Attributes- Oracle provides certain attributes/ cursor variables to control the execution of the cursor. Whenever any cursor(explicit or implicit) is opened and used Oracle creates a set of four system variables via which Oracle keeps track of the 'Current' status of the cursor. You

- Declare a cursor that specifies the SQL select statement that you want to process.
- Open the Cursor.
- Fetch the data from the cursor one row at a time.
- Close the cursor.

How to Declare the Cursor:-

The General Syntex to create any perticular cursor is as follows:-
Cursor <Cursorname> is Sql Statement;

How to Open the Cursor:-

The General Syntex to Open any perticular cursor is as follows:-
Open Cursorname;

Fetching a record From the Cursor:-

The fetch statement retrieves the rows from the active set to the variables one at a time. Each time a fetch is executed. The focus of the DBA cursor advances to the next row in the Active set.
One can make use of any loop structute(Loop-End Loop along with While,For) to fetch the records from the cursor into variable one row at a time.

The General Syntex to Fetch the records from the cursor is as follows:-
Fetch cursorname into variable1,variable2,_____

Closing a Cursor:-

The General Syntex to Close the cursor is as follows:-
Close <cursorname>;

```
SQL> create table customer
  2  (ID number(5) primary key,
  3  Name varchar2(50),
  4  Age number(5),
  5  Address varchar2(50),
  6  Salary varchar2(10));

Table created.

SQL> insert into customer values(1, 'Ramesh', 23, 'Allahabad', '20000');

1 row created.

SQL> insert into customer values(2, 'Suresh', 22, 'Kanpur', '22000');

1 row created.

SQL> insert into customer values(3, 'Mahesh', 24, 'Ghaziabad', '24000');

1 row created.

SQL> insert into customer values(4, 'Chandan', 25, 'Noida', '26000');

1 row created.

SQL> insert into customer values(5, 'Alex', 21, 'Paris', '28000');

1 row created.

SQL> insert into customer values(6, 'Sunita', 20, 'Delhi', '30000');

1 row created.

SQL> select * from customer;

        ID NAME                                               AGE ADDRESS
---------- ---------------------------------------- ---------- -------
SALARY
----------
         1 Ramesh                                           23 Allahab
ad
20000
         2 Suresh                                           22 Kanpur
22000
         3 Mahesh                                           24 Ghaziab
ad
24000
         4 Chandan                                          25 Noida
26000
         5 Alex                                             21 Paris
28000
         6 Sunita                                           20 Delhi
30000

6 rows selected.

SQL>
```

```
SQL> declare
  2  Total_rows number(2);
  3  begin
  4  update customer
  5  set salary = salary + 5000;
  6  dbms_output.put_line(Total_rows || 'customer updated');
  7  end;
  8  /
customer updated

PL/SQL procedure successfully completed.

SQL> select * from customer;

        ID NAME                                               AGE ADDRESS
---------- ---------------------------------------- ---------- -------
SALARY
----------
         1 Ramesh                                           23 Allahab
ad
           25000
         2 Suresh                                           22 Kanpur
           27000
         3 Mahesh                                           24 Ghaziab
ad
           29000
         4 Chandan                                          25 Noida
           31000
         5 Alex                                             21 Paris
           33000
         6 Sunita                                           20 Delhi
           35000

6 rows selected.

SQL>
```

Experiment No.10

Objective:- To implement the concept of Trigger.

Database Triggers:-

Database triggers are procedures that are stored in the database and are implicitly executed(fired) when the contents of a table are changed.

Use of Database Triggers:-

Database triggers support Oracle to provide a highly customized database management system. Some of the uses to which the database triggers can be put to customize management information in Oracle are as follows:-

- A Trigger can permit DML statements against a table anly if they are issued, during regular bussiness hours or on predetermined weekdays.
- A trigger can also be used to keep an audit trail of a table along with the operation performed and the time on which the operation was performed.
- It can be used to prevent invalid transactions.
- Enforce complex security authorizations.

How to apply DataBase Triggers:-

A trigger has three basic parts:-

1. A triggering event or ststement.
2. A triger restriction
3. A trigger action.

Types of Triggers:-

Using the various options , four types of triggers can be created:-

1. **Before Statement Trigger:-** Before executing the triggering statement, the trigger action is executed.
2. **Before Row Trigger:-** Before modifying the each row affected by the triggering statement and before appropriate integrity constraints, the trigger is executed if the trigger restriction either evaluated to TRUE or was not included.'
3. **After Ststement Trigger:-** After executing the triggering statement and applying any deferred integrity canstraints, the trigger action is executed.
4. **After row Trigger:-** After modifying each row affected by the triggering statement and possibly applying appropriate integrity constraints, the trigger action is executed for the current row if the trigger restriction either evaluates to TRUE or was not included.

Syntex For Creating Trigger:-

The syntex for Creating the Trigger is as follows:-

Create or replace Trigger<Triggername> {Before,After} {Delete, Insert, Update } On <Tablename>
For Each row when Condition
Declare
<Variable declarations>;
<Constant Declarations>;
Begin
<PL/SQL> Subprogram Body;
Exception
Exception Pl/SQL block;
End;

How to Delete a Trigger:-

The syntex for Deleting the Trigger is as follows:-
Drop Trigger <Triggername>;

PL/SQL Trigger Example

Let's take a simple example to demonstrate the trigger. In this example, we are using the following CUSTOMERS table:

Create table and have records:

Run SQL Command Line

```
SQL> create table customers(
  2  id int NOT NULL,
  3  name varchar2(20),
  4   age int not null,
  5  address varchar2(30),
  6  salary varchar2(50));

Table created.

SQL> INSERT INTO CUSTOMERS VALUES(1,'RAMESH',23,'ALLAHABAD','20000');

1 row created.

SQL> INSERT INTO CUSTOMERS VALUES(2,'SURESH',22,'KANPUR','22000');

1 row created.

SQL> INSERT INTO CUSTOMERS VALUES(3,'MAHESH',24,'GHAZIABAD','24000');

1 row created.

SQL> INSERT INTO CUSTOMERS VALUES(4,'CHANDAN',25,'NOIDA','26000');

1 row created.

SQL> INSERT INTO CUSTOMERS VALUES(5,'ALEX',21,'PARIS','28000');

1 row created.

SQL> INSERT INTO CUSTOMERS VALUES(6,'SUNITA',20,'DELHI','30000');

1 row created.
```

Run SQL Command Line

```
SQL> SELECT * FROM CUSTOMERS;

        ID NAME                           AGE ADDRESS
---------- -------------------- ---------- ------------------------------
SALARY
----------------------------------------------------
         1 RAMESH                          23 ALLAHABAD
20000

         2 SURESH                          22 KANPUR
22000

         3 MAHESH                          24 GHAZIABAD
24000


        ID NAME                           AGE ADDRESS
---------- -------------------- ---------- ------------------------------
SALARY
----------------------------------------------------
         4 CHANDAN                         25 NOIDA
26000

         5 ALEX                            21 PARIS
28000

         6 SUNITA                          20 DELHI
30000


6 rows selected.
```

Create trigger: Let's take a program to create a row level trigger for the CUSTOMERS table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:

Run SQL Command Line

```
SQL> CREATE OR REPLACE TRIGGER display_salary_changes
  2  BEFORE DELETE OR INSERT OR UPDATE ON customers
  3  FOR EACH ROW
  4  WHEN (NEW.ID > 0)
  5  DECLARE
  6     sal_diff number;
  7  BEGIN
  8     sal_diff := :NEW.salary  - :OLD.salary;
  9     dbms_output.put_line('Old salary: ' || :OLD.salary);
 10     dbms_output.put_line('New salary: ' || :NEW.salary);
 11     dbms_output.put_line('Salary difference: ' || sal_diff);
 12  END;
 13  /

Trigger created.
```

Check the salary difference by procedure: Use the following code to get the old salary, new salary and salary difference after the trigger created. **Note:** As many times you executed this code, the old and new both salary is incremented by 5000 and hence the salary difference is always 5000.

Run SQL Command Line

```
SQL> DECLARE
  2     total_rows number(2);
  3  BEGIN
  4     UPDATE  customers
  5     SET salary = salary + 5000;
  6     IF sql%notfound THEN
  7        dbms_output.put_line('no customers updated');
  8     ELSIF sql%found THEN
  9        total_rows := sql%rowcount;
 10        dbms_output.put_line( total_rows || ' customers updated ');
 11     END IF;
 12  END;
 13  /

PL/SQL procedure successfully completed.
```

```
Old salary: 25000
New salary: 30000
Salary difference: 5000
Old salary: 27000
New salary: 32000
Salary difference: 5000
Old salary: 29000
New salary: 34000
Salary difference: 5000
Old salary: 31000
New salary: 36000
Salary difference: 5000
Old salary: 33000
New salary: 38000
Salary difference: 5000
Old salary: 35000
New salary: 40000
Salary difference: 5000
6 customers updated
```

Run SQL Command Line

```
  1* SELECT * FROM CUSTOMERS
SQL> /

        ID NAME                           AGE ADDRESS
---------- -------------------- ---------- ------------------------------
SALARY
----------------------------------------------------
         1 RAMESH                          23 ALLAHABAD
25000

         2 SURESH                          22 KANPUR
27000

         3 MAHESH                          24 GHAZIABAD
29000


        ID NAME                           AGE ADDRESS
---------- -------------------- ---------- ------------------------------
SALARY
----------------------------------------------------
         4 CHANDAN                         25 NOIDA
31000

         5 ALEX                            21 PARIS
33000

         6 SUNITA                          20 DELHI
35000


6 rows selected.

SQL>
```