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 :	3rd /6th	**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 Procedures in PL/SQL
8.	To implement the concept of Functions in PL/SQL
9.	To implement the concept of Cursor in PL/SQL
10.	To implement the concept of Trigger in PL/SQL

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	PALAK KHURANA
Registration No.	RA1911003030437
Section- Batch	B.TECH CSE I – Batch II

SRM Institute of Science & Technology, Delhi NCR Campus
Department of Computer Science & Engineering

INDEX

Exp. No.	Experiment Name	Date of Conduction	Date of Submission	Faculty Signature
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 procedures.			
8.	To implement the concept of functions.			
9.	To implement the concept of cursors.			
10.	To implement the concept of triggers.			

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

```
SQL> CREATE table client_master(
  2  client_no varchar2(6),
  3  name varchar2(20),
  4  address1 varchar2(30),
  5  address2 varchar2(30),
  6  city varchar2(15),
  7  state varchar2(15),
  8  pincode number(6),
  9  bal_due number(10,2)
 10  );

Table created.
```

```
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>
```

ii) Product_master

Columnname	datatype	size
Product_no	varchar2	
Description	varchar2	
Profit_percent	number	
Unit_measure	varchar2	
Qty_on_hand	number	
Reoder_lvlnumber		
Sell_price	number	
Cost_price number		

```
SQL> create table product_master(
  2  product_no varchar2(8),
  3  description varchar2(15),
  4  profile_percent number(5),
  5  unit_measure varchar2(10),
  6  qty_on_hand number(10),
  7  reorder_lvlnumber number(10),
  8  sell_price number(10),
  9  cost_price number(10)
 10  );

Table created.
```

```
SQL> desc product_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_NO                                         VARCHAR2(8)
 DESCRIPTION                                        VARCHAR2(15)
 PROFILE_PERCENT                                    NUMBER(5)
 UNIT_MEASURE                                       VARCHAR2(10)
 QTY_ON_HAND                                        NUMBER(10)
 REORDER_LVL                                        NUMBER(10)
 SELL_PRICE                                         NUMBER(10)
 COST_PRICE                                         NUMBER(10)

SQL>
```

Q2- Insert the following data into their respective tables:

Clientno	Name	city	pinc de	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

```
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
3      PRAMADA                                                                      BOMBAY         MAHARASHTRA          40057       5000
4      BASU                                                                         BOMBAY         MAHARASHTRA          40056          0
5      RAVI                                                                         DELHI                              100001       2000
6      RUKMINI                                                                      BOMBAY         MAHARASHTRA         400050          0

6 rows selected.
```

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

```
SQL> SELECT * FROM PRODUCT_MASTER;

PRODUCT_ DESCRIPTION     PROFILE_PERCENT UNIT_MEASU 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   1.22FLOPPIES                  5 PIECE              100          20        525        500
P07868   KEYBOARDS                     2 PIECE               10           3       3150       3050
P07885   CD DRIVE                      3 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

9 rows selected.
```

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.
viii) Find the list of all clients who stay in city ‘Bombay’ or city ‘Delhi’ or ‘Madras’.
ix) Find the product whose selling price is greater than 2000 and less than or equal to 5000.
x) List the name, city and state of clients not in the state

Run SQL Command Line

```
SQL> SELECT NAME FROM CLIENT_MASTER;

NAME
--------------------
IVAN
Vandana
PRAMADA
BASU
RAVI
RUKMINI

6 rows selected.

SQL> SELECT NAME, CITY FROM CLIENT_MASTER;

NAME                 CITY
-------------------- ---------------
IVAN                 BOMBAY
Vandana              MADRAS
PRAMADA              BOMBAY
BASU                 BOMBAY
RAVI                 DELHI
RUKMINI              BOMBAY

6 rows selected.

SQL> SELECT DESCRIPTION FROM PRODUCT_MASTER;

DESCRIPTION
---------------
1.44FLOPPIES
MONITORS
MOUSE
1.22FLOPPIES
KEYBOARDS
CD DRIVE
540 HDD
1.44 DRIVE
1.22 DRIVE

9 rows selected.

SQL> SELECT NAME FROM CLIENT_MASTER WHERE CITY='BOMBAY';

NAME
--------------------
IVAN
PRAMADA
BASU
```

Run SQL Command Line

```
SQL> SELECT NAME FROM CLIENT_MASTER WHERE CITY='BOMBAY';

NAME
--------------------
IVAN
PRAMADA
BASU
RUKMINI

SQL> SELECT * FROM CLIENT_MASTER WHERE CLIENT_NO='1' OR CLIENT_NO='2';

CLIENT NAME                 ADDRESS1                      ADDRESS2                      CITY            STATE              PINCODE    BAL_DUE
------ -------------------- ----------------------------- ----------------------------- --------------- --------------- ---------- ----------
1      IVAN                                                                             BOMBAY          MAHARASHTRA         400054      15000
2      Vandana                                                                          MADRAS          TAMILNADU           780001          0

SQL> SELECT * FROM PRODUCT_MASTER WHERE DESCRIPTION='1.44 DRIVE' OR DESCRIPTION='1.22 DRIVE';

PRODUCT_ DESCRIPTION     PROFILE_PERCENT UNIT_MEASU QTY_ON_HAND REORDER_LVL SELL_PRICE COST_PRICE
-------- --------------- --------------- ---------- ----------- ----------- ---------- ----------
P07975   1.44 DRIVE                    5 PIECE               10           3       1050       1000
P08865   1.22 DRIVE                    5 PIECE                2           3       1050       1000

SQL> SELECT DESCRIPTION FROM PRODUCT_MASTER WHERE SELL_PRICE > 5000;

DESCRIPTION
---------------
MONITORS
CD DRIVE
540 HDD

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

NAME
--------------------
IVAN
Vandana
PRAMADA
BASU
RAVI
RUKMINI

6 rows selected.

SQL>
```

Run SQL Command Line

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

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

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.

Ques 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

```
SQL> UPDATE PRODUCT_MASTER SET SELL_PRICE=1150 WHERE DESCRIPTION='1.44FLOPPIES';

1 row updated.

SQL> SELECT * FROM PRODUCT_MASTER;

PRODUCT_ DESCRIPTION     PROFIT_PERCENT UNIT_MEASU QTY_ON_HAND REORDER_LVL SELL_PRICE COST_PRICE
-------- --------------- -------------- ---------- ----------- ----------- ---------- ----------
P00001   1.44FLOPPIES                 5 PIECE              100          20       1150        500
P03453   MONITORS                     6 PIECE               10           3      12000      11200
P06734   MOUSE                        5 PIECE               20           5       1050        500
P07865   1.22FLOPPIES                 5 PIECE              100          20        525        500
P07868   KEYBOARDS                    2 PIECE               10           3       3150       3050
P07885   CD DRIVE                     3 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

9 rows selected.
```

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

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

1 row deleted.

SQL> SET LINESIZE 400;
SQL> SELECT * FROM CLIENT_MASTER;

CLIENT_NO NAME                 ADDRESS1                     ADDRESS2                     CITY           STATE            PINCODE    BAL_DUE   PHONE_NO
--------- -------------------- ---------------------------- ---------------------------- -------------- -------------- --------- --------- ----------
2         Vandana                                                                        MADRAS         TAMILNADU         780001         0
3         PRAMADA                                                                        BOMBAY         MAHARASHTRA        40057      5000
4         BASU                                                                           BOMBAY         MAHARASHTRA        40056         0
5         RAVI                                                                           DELHI                            100001      2000
6         RUKMINI                                                                        BOMBAY         MAHARASHTRA       400050         0
```

iii. Change the city of client_no ‘0005’ to Bombay.

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

1 row updated.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT_NO NAME                 ADDRESS1                     ADDRESS2                     CITY           STATE            PINCODE    BAL_DUE   PHONE_NO
--------- -------------------- ---------------------------- ---------------------------- -------------- -------------- --------- --------- ----------
2         Vandana                                                                        MADRAS         TAMILNADU         780001         0
3         PRAMADA                                                                        BOMBAY         MAHARASHTRA        40057      5000
4         BASU                                                                           BOMBAY         MAHARASHTRA        40056         0
5         RAVI                                                                           BOMBAY                           100001      2000
6         RUKMINI                                                                        BOMBAY         MAHARASHTRA       400050         0
```

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

```
SQL> UPDATE CLIENT_MASTER SET BAL_DUE=1000 WHERE CLIENT_NO='0001';

1 row updated.

SQL> SELECT * FROM CLIENT_MASTER;

CLIENT_NO NAME                ADDRESS1                    ADDRESS2                    CITY           STATE            PINCODE    BAL_DUE   PHONE_NO
--------- ------------------- --------------------------- --------------------------- -------------- -------------- ---------- --------- ----------
2         Vandana                                                                     MADRAS         TAMILNADU          780001         0
3         PRAMADA                                                                     BOMBAY         MAHARASHTRA         40057      5000
4         BASU                                                                        BOMBAY         MAHARASHTRA         40056         0
5         RAVI                                                                        BOMBAY                            100001      2000
6         RUKMINI                                                                     BOMBAY         MAHARASHTRA        400050         0
```

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

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

PRODUCT_ 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_NO NAME                ADDRESS1                    ADDRESS2                    CITY           STATE            PINCODE    BAL_DUE   PHONE_NO
--------- ------------------- --------------------------- --------------------------- -------------- -------------- ---------- --------- ----------
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 NAME FROM CLIENT_MASTER WHERE NAME LIKE '_A%' OR NAME LIKE '_a%';

NAME
--------------------
Vandana
BASU
RAVI
```

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

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

DESCRIPTION
---------------
1.22 DRIVE
1.22FLOPPIES
1.44 DRIVE
1.44FLOPPIES
540 HDD
CD DRIVE
KEYBOARDS
MONITORS
MOUSE
```

ix. Count the total number of orders

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

SUM(REORDER_LVL)
----------------
              63
```

x. Calculate the average price of all the products.

```
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
```

Experiment No. 3

Aim : To implement the restrictions / constraints on the table.

Question 1. Create the following tables:

i) Sales_master

Columnname	Datatype	Size
Salesman_no	varchar2	6
Sal_name	varchar2	20
Address	varchar2	
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

```
SQL> create table sales_master(
  2  salesman_no varchar2(6),
  3  sal_name varchar2(20),
  4  address varchar2(20),
  5  city varchar2(20),
  6  pincode number(6),
  7  state varchar2(20),
  8  sal_amt number(8,2),
  9  tgt_to_get number(6,2),
 10  ytd_sales number(6,2),
 11  remarks varchar2(30)
 12  );

Table created.
```

```
Name                                                      Null?    Type
--------------------------------------------------------- -------- ----------------------------------------
SALESMAN_NO                                                        VARCHAR2(6)
SAL_NAME                                                           VARCHAR2(20)
ADDRESS                                                            VARCHAR2(20)
CITY                                                               VARCHAR2(20)
PINCODE                                                            NUMBER(6)
STATE                                                              VARCHAR2(20)
SAL_AMT                                                            NUMBER(8,2)
TGT_TO_GET                                                         NUMBER(6,2)
YTD_SALES                                                          NUMBER(6,2)
REMARKS                                                            VARCHAR2(30)
```

ii) Sales_order

Columnname	Datatype	Size
S_order_no	varchar2	6
S_order_date	Date	6
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> create table sales_order(
  2  s_order_no varchar2(6),
  3  s_order_date date,
  4  client_no varchar2(25),
  5  dely_add varchar2(6),
  6  dely_type char(1),
  7  billed_yn char(1),
  8  salesman_no varchar2(6),
  9  dely_date date,
 10  order_status varchar2(10)
 11  );

Table created.
```

```
SQL> desc sales_order;
 Name                                                      Null?    Type
 --------------------------------------------------------- -------- ----------------------------------------
 S_ORDER_NO                                                         VARCHAR2(6)
 S_ORDER_DATE                                                       DATE
 CLIENT_NO                                                          VARCHAR2(25)
 DELY_ADD                                                           VARCHAR2(6)
 DELY_TYPE                                                          CHAR(1)
 BILLED_YN                                                          CHAR(1)
 SALESMAN_NO                                                        VARCHAR2(6)
 DELY_DATE                                                          DATE
 ORDER_STATUS                                                       VARCHAR2(10)
```

iii) Sales_order_details

Column	Datatype	Size
S_order_no	Varchar2	6
Product_no	Varchar2	6
Qty_order	Number	8
Qty_disp	Number	8
Product_rate	Number	10,2

```
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)
  7  );

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)
```

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

i) Data for sales_master 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

```
SQL> insert into sales_master values('500001' , 'Kiran', 'A/14 worli' , 'Bombay', 400002, 'Mah' , 3000 , 100, 50, 'Good');

1 row created.

SQL> insert into sales_master values('500002' , 'Manish', '65, nariman' , 'Bombay', 400001, 'Mah' , 3000 , 200, 100, 'Good');

1 row created.

SQL> insert into sales_master values('500002' , 'Ravi', 'P-7 Bandra' , 'Bombay', 400032, 'Mah' , 3000 , 200, 100, 'Good');

1 row created.

SQL> insert into sales_master values('500002' , 'Ashish', 'A/5 Juhu' , 'Bombay', 400044, 'Mah' , 3500 , 200, 150, 'Good');

1 row created.
```

```
SQL> select * from sales_master;

SALESM SAL_NAME            ADDRESS            CITY                 PINCODE STATE                SAL_AMT TGT_TO_GET  YTD_SALES REMARKS
------ -------------------- -------------------- -------------------- --------- -------------------- --------- ---------- ---------- ------------------------------
500001 Kiran               A/14 worli         Bombay                 400002 Mah                     3000        100         50 Good
500002 Manish              65, nariman        Bombay                 400001 Mah                     3000        200        100 Good
500002 Ravi                P-7 Bandra         Bombay                 400032 Mah                     3000        200        100 Good
500002 Ashish              A/5 Juhu           Bombay                 400044 Mah                     3500        200        150 Good

SQL>
```

i) Data for salesorder table :

S_ordern o	S_orderdate	Client no	Del y typ e	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

```
SQL> insert into sales_order(s_order_no , s_order_date, client_no, dely_type, billed_yn, salesman_no, dely_date, order_status) values('019001',TO_DATE('12-jan-96' , 'dd-mon-yy'), '0001' , 'F' , 'N','50001',TO_DA
TE('20-jan-96' , 'dd-mon-yy'), 'Ip');

1 row created.

SQL>  insert into sales_order(s_order_no , s_order_date, client_no, dely_type, billed_yn, salesman_no, dely_date, order_status) values('019002',TO_DATE('25-jan-96' , 'dd-mon-yy'), '0002' , 'P' , 'N','50002',TO_D
ATE('27-jan-96' , 'dd-mon-yy'), 'C');

1 row created.

SQL> insert into sales_order(s_order_no , s_order_date, client_no, dely_type, billed_yn, salesman_no, dely_date, order_status) values('016865',TO_DATE('18-feb-96' , 'dd-mon-yy'), '0003' , 'F' , 'Y','500003',TO_D
ATE('20-feb-96' , 'dd-mon-yy'), 'F');

1 row created.

SQL> insert into sales_order(s_order_no , s_order_date, client_no, dely_type, billed_yn, salesman_no, dely_date, order_status) values('019003',TO_DATE('18-apr-96' , 'dd-mon-yy'), '0001' , 'F' , 'Y','500001',TO_D
ATE('07-apr-96' , 'dd-mon-yy'), 'F');

1 row created.

SQL> insert into sales_order(s_order_no , s_order_date, client_no, dely_type, billed_yn, salesman_no, dely_date, order_status) values('046866',TO_DATE('20-may-96' , 'dd-mon-yy'), '0004' , 'P' , 'N','500002',TO_D
ATE('22-may-96' , 'dd-mon-yy'), 'C');

1 row created.

SQL> insert into sales_order(s_order_no , s_order_date, client_no, dely_type, billed_yn, salesman_no, dely_date, order_status) values('010008',TO_DATE('24-may-96' , 'dd-mon-yy'), '0005' , 'F' , 'N','500004',TO_D
ATE('26-may-96' , 'dd-mon-yy'), 'Ip');

1 row created.
```

```
SQL> select * from sales_order;

S_ORDE S_ORDER_D CLIENT_NO                DELY_A D B SALESM DELY_DATE ORDER_STAT
------ --------- ------------------------ ------ - - ------ --------- ----------
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 18-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
	P00001	10	5	525

```
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' , 'P07885',  10 , 10, 5250);

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('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' , 'P07975',  1 , 0, 1050);

1 row created.

SQL> insert into sales_order_details values('010008' , 'P00001',  10 , 5, 525);

1 row created.
```

```
SQL> select * from sales_order_details;

S_ORDE PRODUC  QTY_ORDER   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
046866 P06734          1          1        12000
046866 P07965          1          0         8400
010008 P07975          1          0         1050
010008 P00001         10          5          525
```

Experiment No. 4

Aim : To alter the structure of the table.

Question 1. 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’

```
SQL> create table challan_header
  2  (
  3  Challan_no    varchar2(6) Primary key,
  4  s_order_no    varchar2(6) references sales_order,
  5  challan_date    date     not null,
  6  billed_yn    char(1)    DEFAULT 'N',
  7  CHECK (billed_yn in ('Y','N'))
  8  );

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                        CHAR(1)
```

Table Name : Challan_Details

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key/Foreign key references
Product_no	varchar2	8	Product_no of product_master
Qty_disp	number	4,2	not null

```
SQL> CREATE TABLE CHALLAN_DETAILS(
  2  CHALLAN_NO VARCHAR2(6),
  3  PRODUCT_NO VARCHAR2(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)
```

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

Challan No	S_order No	Challan Date	Billed
CH9001	019001	12-DEC-95	Y
CH865	046865	12-NOV-95	Y
CH3965	010008	12-OCT-95	Y

```
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
```

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

```
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.

```
SQL> ALTER TABLE CLIENT_MASTER ADD PRIMARY KEY(CLIENT_NO);

Table altered.

SQL> DESC CLIENT_MASTER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                 NOT NULL VARCHAR2(10)
 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(12)
```

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

```
SQL> ALTER TABLE CLIENT_MASTER ADD PHONE_NO NUMBER(12);

Table altered.

SQL> DESC CLIENT_MASTER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                 NOT NULL VARCHAR2(10)
 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(12)
```

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

```
SQL> ALTER TABLE PRODUCT_MASTER MODIFY ( DESCRIPTION NOT NULL , PROFIT_PERCENT NOT NULL , SELL_PRICE NOT NULL , COST_PRICE NOT NULL);

Table altered.

SQL> DESC PRODUCT_MASTER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_NO                                NOT NULL VARCHAR2(8)
 DESCRIPTION                               NOT NULL VARCHAR2(15)
 PROFIT_PERCENT                            NOT NULL NUMBER(5)
 UNIT_MEASURE                                       VARCHAR2(10)
 QTY_ON_HAND                                        NUMBER(10)
 REORDER_LVL                                        NUMBER(10)
 SELL_PRICE                                NOT NULL NUMBER(10)
 COST_PRICE                                NOT NULL NUMBER(10)
```

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

```
SQL> ALTER TABLE CLIENT_MASTER MODIFY CLIENT_NO VARCHAR(10);

Table altered.

SQL> DESC CLIENT_MASTER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                 NOT NULL VARCHAR2(10)
 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(12)
```

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

```
SQL> SELECT PRODUCT_NO , DESCRIPTION FROM PRODUCT_MASTER WHERE PROFIT_PERCENT BETWEEN 20 AND 30;

no rows selected

SQL>
```

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*)**

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

***Due_Date*)LIBRARY_BRANCH (*Branch_id, Branch_Name, Address*)**

BOOK :

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

SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                PUB_YEAR            PUBLISHER_NAME
---------- -------------------- -------------------- --------------------
         1 DBMS                 JAN-2017            MCGRAW-HILL
         2 ADBMS                JUN-2016            MCGRAW-HILL
         4 CG                   SEP-2015            GRUPO PLANETA
         5 OS                   MAY-2016            PEARSON
```

BOOK_AUTHORS :

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

SQL> SELECT * FROM BOOK_AUTHORS;

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

PUBLISHER :

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

SQL> SELECT * FROM PUBLISHER;

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

BOOK_COPIES :

```
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> SELECT * FROM BOOK_COPIES;

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

6 rows selected.
```

BOOK_LENDING :

```
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> SELECT * FROM BOOK_LENDING;

DATE_OUT  DUE_DATE     BOOK_ID  BRANCH_ID    CARD_NO
--------- --------- ---------- ---------- ----------
01-JAN-17 01-JUN-17          1         10        101
21-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
```

LIBRARY_BRANCH :

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

SQL> SELECT * FROM LIBRARY_BRANCH;

 BRANCH_ID BRANCH_NAME                                          ADDRESS
---------- -------------------------------------------------- ------------------------------------
        10 RR NAGAR                                            BANGALORE
        11 RNSIT                                               BANGALORE
        12 RAJAJI NAGAR                                        BANGALORE
        13 NITTE                                               MANGALORE
        14 MANIPAL                                             UDUPI
```

Write SQL queries to :

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

```
SQL> SELECT B.BOOK_ID,B.TITLE,B.PUBLISHER_NAME,A.AUTHOR_NAME,C.NO_OF_COPIES,L.BRANCH_ID
  2  FROM BOOK B,BOOK_AUTHORS A,BOOK_COPIES C,LIBRARY_BRANCH L
  3  WHERE B.BOOK_ID=A.BOOK_ID
  4  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
         2 ADBMS                MCGRAW-HILL          NAVATHE                         5         13
         3 CN                   PEARSON              TANENBAUM                       7         14
         5 OS                   PEARSON              GALVIN                          1         10
         4 CG                   GRUPO PLANETA        EDWARD ANGEL                    3         11

7 rows selected.
```

2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun2017.

```
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;

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

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

```
SQL> DELETE FROM BOOK 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                JUN-2016             MCGRAW-HILL
         4 CG                   SEP-2015             GRUPO PLANETA
         5 OS                   MAY-2016             PEARSON
```

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 FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH L
  2  WHERE B.BOOK_ID=C.BOOK_ID   AND C.BRANCH_ID=L.BRANCH_ID;

View created.

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.
```

EXPERIMENT NO. 6

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

Aim : To implement the concept of Procedures in PL/SQL

Run SQL Command Line

```
SQL> create table user1
  2  (
  3  id number(10) primary key,
  4  name varchar2(100)
  5  );

Table created.
```

```
SQL> create or replace 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> begin
  2  insertuser1(101 , 'Rahul');
  3  dbms_output.put_line('record inserted successfully');
  4  end;
  5  /

PL/SQL procedure successfully completed.
```

```
SQL> select * from user1;

        ID NAME
---------- ------------------
       101 Rahul
```

EXPERIMENT NO. 8

Aim : To implement the concept of Functions in PL/SQL

```
SQL> create or replace 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> 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.
```

EXPERIMENT NO. 9

Aim : To implement the concept of Cursors in PL/SQL

```
SQL> create table customers(
  2  id number(5),
  3  name varchar2(15),
  4  age number(3),
  5  address varchar2(20),
  6  salary number(10)
  7  );

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.
```

```
SQL> set serveroutput on;
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  /
6 customers updated

PL/SQL procedure successfully completed.

SQL> select * from customers;

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

6 rows selected.
```

EXPERIMENT NO. 10

Aim : To implement the concept of Triggers in PL/SQL

```
SQL> create table customers(
  2  id number(5),
  3  name varchar2(15),
  4  age number(3),
  5  address varchar2(20),
  6  salary number(10)
  7  );

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.

SQL> set linesize 300;
SQL> select * from customers;

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

6 rows selected.
```

```
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.
```

```
SQL> set serveroutput on;
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  /
Old salary:20000
New salary:25000
Salary difference:5000
Old salary:22000
New salary:27000
Salary difference:5000
Old salary:24000
New salary:29000
Salary difference:5000
Old salary:26000
New salary:31000
Salary difference:5000
Old salary:28000
New salary:33000
Salary difference:5000
Old salary:30000
New salary:35000
Salary difference:5000
6 customers updated

PL/SQL procedure successfully completed.

SQL>
```