ASSIGNMENT-1

A. Create and Insert the following table

Table Name- client _master
Description- Used to store client information

Column No	Column Name	Data Type	Size	Attributes
1	Client_no	Varchar2	6	Primary key, first letter must start with 'C'
2	Name	Varchar2	30	Not NULL
3	Address1	Varchar2	30	
4	Address2	Varchar2	30	
5	City	Varchar2	15	
6	State	Varchar2	15	
7	Pincode	Number	6	
8	Balance_due	Number	10,2	

Data of client_master table

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
C001	Ivan Bayross	P-76	Worli	Bombay	Maharastra	400054	15000
C002	VandanaSatiwal	128	Adams Street	Madras	TamilNadu	780001	0
C003	PramadaJaguste	157	Gopalpur	Kolkata	West Bengal	700058	5000
C004	BasuNavindgi	A/12	Nariman	Bombay	Maharastra	400056	0
C005	Ravi Sreedharan	B/34	Rajnagar	Delhi	Delhi	100001	2000
C006	Rukmini	Q-12	Bandra	Bombay	Maharastra	400050	0

SQL Commands:

```
create table client_master(
  Client_no varchar2(6) primary key check (Client_no like 'C%'),
  Name varchar2(30) NOT NULL,
  Address1 varchar2(30) ,
  Address2 varchar2(30) ,
  City varchar2(15) ,
  State varchar2(15) ,
  Pincode Number(6) ,
  Balance_due Number(10,2)
);
INSERT INTO client_master VALUES('C001','Ivan Bayross','P-76','Worli','Bombay','Maharastra',400054,15000);
INSERT INTO client_master VALUES('C002','Vandana Satiwal','128','Adams Street','Madras','Tamil Nadu',780001,0);
INSERT INTO client_master VALUES('C003','Pramada Jaguste','157','Gopalpur','Kolkata','West Bengal',70058,5000);
INSERT INTO client_master VALUES('C004','Basu Navindgi','A/12','Nariman','Bombay','Maharastra',400056,0);
INSERT INTO client_master VALUES('C005','Ravi Sreedharan','B/34','Rajnagar','Delhi','Delhi',100001,2000);
INSERT INTO client_master VALUES('C006','Rukmini','Q-12','Bandra','Bombay','Maharastra',400050,0);
```

SQL Prompt Output:

```
SQL> @E:\CSE_30\Client

Table created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.

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

```
SQL> set linesize 500
SQL> select * from client_master;

CLIENT NAME                       ADDRESS1                    ADDRESS2                    CITY           STATE            PINCODE BALANCE_DUE
------ ---------------------------- ---------------------------- ---------------------------- --------------- --------------- --------- -----------
C001   Ivan Bayross                P-76                        Worli                       Bombay         Maharastra         400054       15000
C002   Vandana Satiwal             128                         Adams Street                Madras         Tamil Nadu         780001           0
C003   Pramada Jaguste             157                         Gopalpur                    Kolkata        Weste Bengal        70058        5000
C004   Basu Navindgi               A/12                        Nariman                     Bombay         Maharastra         400056           0
C005   Ravi Sreedharan             B/34                        Rajnagar                    Delhi          Delhi              100001        2000
C006   Rukmini                     Q-12                        Bandra                      Bombay         Maharastra         400050           0

6 rows selected.
```

B. Create and Insert the following table

Table Name- product_master:
Description- Used to store product information

Column No	Column Name	Data Type	Size	Attributes
1	Product_no	Varchar2	6	Primary key, First letter must start with 'P'
2	Description	Varchar2	40	Not null
3	Profit_percent	Number	4,2	Not null
4	Unit_measure	Varchar2	10	Not null
5	Qty_on_hand	Number	8	Not null
6	Reorder_level	Number	8	Not null
7	Sell_price	Number	8,2	Not null, cannot be 0
8	Cost_price	Number	8,2	Not null, cannot be 0

Data of product_master table

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
P00001	1.44 Floppies	5	Piece	100	20	525	500
P03453	Monitors	6	Piece	10	3	12000	11280
P06734	Mouse	5	Piece	20	5	1050	1000
P07865	1.22 Floppies	5	Piece	100	20	525	500
P07868	Keyboard	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	900
P08865	1.22 Drive	5	Piece	2	3	1025	850

SQL Commands:

```
create table product_master(
  Product_no varchar2(6) Primary key check (Product_no like 'P%'),
  Description varchar2(40) NOT NULL,
  Profit_percent number(4,2) not null ,
  Unit_measure varchar2(10) not null ,
  Qty_on_hand number(8) not null,
  Reorder_level number(8) not null,
  Sell_price number(8,2) not null check(Sell_price > 0),
    Cost_price number(8,2) not null check(Cost_price > 0)
);
Insert into product_master values('P00001','1.44 Floppies',5,'Piece',100,20,525,500);
Insert into product_master values('P03453','Monitors',6,'Piece',10,3,12000,11280);
Insert into product_master values('P06734','Mouse',5,'Piece',20,5,1050,1000);
Insert into product_master values('P07865','1.22 Floppies',5,'Piece',100,20,525,500);
Insert into product_master values('P07868','Keyboard',2,'Piece',10,3,3150,3050);
Insert into product_master values('P07885','CD Drive',2.5,'Piece',10,3,5250,5100);
Insert into product_master values('P07965','540 HDD',4,'Piece',10,3,8400,8000);
Insert into product_master values('P07975','1.44 Drive',5,'Piece',10,3,1050,900);
Insert into product_master values('P08865','1.22 Drive',5,'Piece',2,3,1025,850);
```

SQL Prompt Output:

```
SQL> @E:\CSE_30\Product
Table created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.
```

```
SQL> select * from product_master;

PRODUC DESCRIPTION                          PROFIT_PERCENT UNIT_MEASU QTY_ON_HAND REORDER_LEVEL SELL_PRICE COST_PRICE
------ ------------------------------------ -------------- ---------- ----------- ------------- ---------- ----------
P00001 1.44 Floppies                                    5 Piece             100            20        525        500
P03453 Monitors                                         6 Piece              10             3      12000      11280
P06734 Mouse                                            5 Piece              20             5       1050       1000
P07865 1.22 Floppies                                    5 Piece             100            20        525        500
P07868 Keyboard                                         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        900
P08865 1.22 Drive                                       5 Piece               2             3       1025        850

9 rows selected.

SQL> commit;

Commit complete.

SQL> spool off
```

C. Create and Insert the following table

Table Name- salesman_master:
Description- Used to store salesman working for company

Column No	Column Name	Data Type	Size	Attributes
1	Salesman_no	Varchar2	6	Primary key,first letter must start with 'S'
2	Salesman _name	Varchar2	30	Not null
3	Address1	Varchar2	30	Not null
4	Address2	Varchar2	30	
5	City	Varchar2	20	
6	Pincode	Number	8	
7	State	Varchar2	20	
8	Sal_amt	Number	8, 2	Not null, cannot be 0

Data of salesman_master table

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
S001	Kiran	A/14	Worli	Bombay	400002	Maharastra	3000
S002	Manish	65	Nariman	Bombay	400001	Maharastra	3000
S003	Ravi	P-7	Bandra	Bombay	400032	Maharastra	3000
S004	Asish	A/5	Juhu	Bombay	400044	Maharastra	3000

SQL Commands:

```
CREATE table salesman_master(
  Salesman_No varchar2(6) PRIMARY key ,
  CHECK(Salesman_No like 'S%'),
  Salesman_name varchar2(30) not null,
  Address1 varchar2(30) not null,
  Address2 varchar2(30),
  City varchar2(20),
  Pincode NUMBER(8),
  State VARCHAR2(20),
  Sal_amt number(8,2) not null check(Sal_amt > 0)
);
insert into salesman_master values ('S001','Kiran','A/14','Worli','Bombay',400002,'Maharastra',3000);
insert into salesman_master values ('S002','Manish','65','Nariman','Bombay',400001,'Maharastra',3000);
insert into salesman_master values ('S003','Ravi','P-7','Bandra','Bombay',400032,'Maharastra',3000);
insert into salesman_master values ('S004','Asish','A/5','Juhu','Bombay',400044,'Maharastra',3000);
```

SQL Prompt Output:

```
SQL> desc salesman_master
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SALESMAN_NO                               NOT NULL VARCHAR2(6)
 SALESMAN_NAME                             NOT NULL VARCHAR2(30)
 ADDRESS1                                  NOT NULL VARCHAR2(30)
 ADDRESS2                                           VARCHAR2(30)
 CITY                                               VARCHAR2(20)
 PINCODE                                            NUMBER(8)
 STATE                                              VARCHAR2(20)
 SAL_AMT                                   NOT NULL NUMBER(8,2)
```

```
SQL> set linesize 500;
SQL> select * from salesman_master;

SALESM SALESMAN_NAME                 ADDRESS1                   ADDRESS2                   CITY                 PINCODE STATE                  SAL_AMT
------ ---------------------------- -------------------------- -------------------------- -------------------- ------- -------------------- -------
S001   Kiran                        A/14                       Worli                      Bombay               400002 Maharastra                3000
S002   Manish                       65                         Nariman                    Bombay               400001 Maharastra                3000
S003   Ravi                         P-7                        Bandra                     Bombay               400032 Maharastra                3000
S004   Asish                        A/5                        Juhu                       Bombay               400044 Maharastra                3000
```

D. Create and Insert the following table

Table Name- sales_order:
Description- Used to store client's orders

Column No	Column Name	Data Type	Size	Attributes
1	Order_no	Varchar2	6	Primary key, first letter must start with 'O'
2	Order_date	Date		
3	Client_no	Varchar2	6	Foreign key references Client_master table
4	Salesman_no	Varchar2	6	Foreign key references salesman _master table
5	Delivery_type	Char	1	Delivery part(P),full(F) Default 'F'
6	Bill_y_n	Char	1	
7	Delivery_date	Date		Cannot be less than Order_date
8	Order_status	Varchar2	10	Values('InProcess',' Fullfilled', 'BackOrder', 'Cancelled')

Data of sales_order table

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
O19001	12-Jan-96	C001	S001	F	N	20-Jan-96	InProcess
O19002	25-Jan-96	C002	S002	P	N	27-Jan-96	BackOrder
O46865	18-Feb-96	C003	S003	F	Y	20-Feb-96	Fullfilled
O19003	03-Apr-96	C001	S001	F	Y	07-Apr-96	Fullfilled
O46866	20-May-96	C004	S002	P	N	22-May-96	Cancelled
O19008	24-May-96	C005	S004	F	N	26-May-96	InProcess

SQL Commands:

```
create table sales_order(
  Order_No varchar2(6) PRIMARY key,
  check(Order_No like 'O%'),
  Order_date Date,
  Client_No varchar2(6) REFERENCES client_master(Client_No),
  Salesman_No varchar2(6) REFERENCES salesman_master(Salesman_No),
  Delivery_type Char(1) DEFAULT 'F' check(Delivery_type in ('P','F')),
  Bill_y_n char(1),
  Delivery_date date ,
  Order_status varchar2(10),
  constraint ck_order_status check(Order_status in ('InProcess','Fullfilled','BackOrder','Cancelled')),
  constraint ck_delivery_date check (Delivery_date >= Order_date)
);
insert into sales_order VALUES ('O19001','12-Jan-96','C001','S001','F','N','20-Jan-96','InProcess');
insert into sales_order VALUES ('O19002','25-Jan-96','C002','S002','P','N','27-Jan-96','BackOrder');
insert into sales_order VALUES ('O46865','18-Feb-96','C003','S003','F','Y','20-Feb-96','Fullfilled');
insert into sales_order VALUES ('O19003','03-Apr-96','C001','S001','F','Y','07-Apr-96','Fullfilled');
insert into sales_order VALUES ('O46866','20-May-96','C004','S002','P','N','22-May-96','Cancelled');
insert into sales_order VALUES ('O19008','24-May-96','C005','S004','F','N','26-May-96','InProcess');
```

SQL Prompt Output:

```
SQL> @E:\CSE_30\Ass1\ass_1_4

Table created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.
```

```
SQL> desc sales_order
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_NO                                  NOT NULL VARCHAR2(6)
 ORDER_DATE                                         DATE
 CLIENT_NO                                          VARCHAR2(6)
 SALESMAN_NO                                        VARCHAR2(6)
 DELIVERY_TYPE                                      CHAR(1)
 BILL_Y_N                                           CHAR(1)
 DELIVERY_DATE                                      DATE
 ORDER_STATUS                                       VARCHAR2(10)
```

```
SQL> set linesize 500;
SQL> select * from sales_order;

ORDER_ ORDER_DAT CLIENT SALESM D B DELIVERY_ ORDER_STAT
------ --------- ------ ------ - - --------- ----------
O19001 12-JAN-96 C001   S001   F N 20-JAN-96 InProcess
O19002 25-JAN-96 C002   S002   P N 27-JAN-96 BackOrder
O46865 18-FEB-96 C003   S003   F Y 20-FEB-96 Fullfilled
O19003 03-APR-96 C001   S001   F Y 07-APR-96 Fullfilled
O46866 20-MAY-96 C004   S002   P N 22-MAY-96 Cancelled
O19008 24-MAY-96 C005   S004   F N 26-MAY-96 InProcess

6 rows selected.
```

E. Create and Insert the following table

Table Name- sales_order_details:
Description- Used to store client's orders with details of each product ordered

Column No	Column Name	Data Type	Size	Attributes
1	Order _no	Varchar2	6	Foreign key referencessales_order table
2	Product_no	Varchar2	6	Foreign key references product_master table
3	Qty_ordered	Number	8	
4	Qty_disp	Number	8	
5	Product_rate	Number	10, 2	

Data of sales_order_details

Col-1	Col-2	Col-3	Col-4	Col-5
O19001	P00001	4	4	525
O19001	P07965	2	1	8400
O19001	P07885	2	1	5250
O19002	P00001	10	0	525
O46865	P07868	3	3	3150
O46865	P07885	3	1	5250
O46865	P00001	10	10	525
O46865	P03453	4	4	1050
O19003	P03453	2	2	1050
O19003	P06734	1	1	12000
O46866	P07965	1	0	8400
O46866	P07975	1	0	1050
O19008	P00001	10	5	525
O19008	P07975	5	3	1050

SQL Commands:

```
create table sales_order_details(
  Order_No varchar2(6) references sales_order(Order_No),
  Product_No varchar2(6) references product_master(Product_No),
  Qty_ordered NUMBER(8),
  Qty_disp number(8),
  Product_rate number(10,2)
);
desc sales_order_details;
insert into sales_order_details VALUES ('O19001','P00001',4,4,525);
insert into sales_order_details VALUES ('O19001','P07965',2,1,8400);
insert into sales_order_details VALUES ('O19001','P07885',2,1,5250);
insert into sales_order_details VALUES ('O19002','P00001',10,0,525);
insert into sales_order_details VALUES ('O46865','P07868',3,3,3150);
insert into sales_order_details VALUES ('O46865','P07885',3,1,5250);
insert into sales_order_details VALUES ('O46865','P00001',10,10,525);
insert into sales_order_details VALUES ('O46865','P03453',4,4,1050);
insert into sales_order_details VALUES ('O19003','P03453',2,2,1050);
insert into sales_order_details VALUES ('O19003','P06734',1,1,12000);
insert into sales_order_details VALUES ('O46866','P07965',1,0,8400);
insert into sales_order_details VALUES ('O46866','P07975',1,0,1050);
insert into sales_order_details VALUES ('O19008','P00001',10,5,525);
insert into sales_order_details VALUES ('O19008','P07975',5,3,1050);
```

SQL Prompt Output:

```
SQL> desc sales_order_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_NO                                           VARCHAR2(6)
 PRODUCT_NO                                         VARCHAR2(6)
 QTY_ORDERED                                        NUMBER(8)
 QTY_DISP                                           NUMBER(8)
 PRODUCT_RATE                                       NUMBER(10,2)
```

```
SQL> select * from sales_order_details;

ORDER_ PRODUC QTY_ORDERED   QTY_DISP PRODUCT_RATE
------ ------ ----------- ---------- ------------
O19001 P00001           4          4          525
O19001 P07965           2          1         8400
O19001 P07885           2          1         5250
O19002 P00001          10          0          525
O46865 P07868           3          3         3150
O46865 P07885           3          1         5250
O46865 P00001          10         10          525
O46865 P03453           4          4         1050
O19003 P03453           2          2         1050
O19003 P06734           1          1        12000
O46866 P07965           1          0         8400

ORDER_ PRODUC QTY_ORDERED   QTY_DISP PRODUCT_RATE
------ ------ ----------- ---------- ------------
O46866 P07975           1          0         1050
O19008 P00001          10          5          525
O19008 P07975           5          3         1050

14 rows selected.

SQL> spool off
```

ASSIGNMENT-2

1. **Find the names of all clients having ‘a’ as the second letter in their names.**

   SQL Command:

   select name from client_master where name like '_a%' ;

   Output:

```
SQL> select name from client_master where name like '_a%' ;

NAME
-----------------------------
Vandana Satiwal
Basu Navindgi
Ravi Sreedharan
```

2. **Find out the clients who do not stay in a city whose first letter is ‘B’.**

   SQL Command:

   select name from client_master where city not like 'B%' ;

   Output:

```
SQL> select name from client_master where city not like 'B%';

NAME
-----------------------------
Vandana Satiwal
Pramada Jaguste
Ravi Sreedharan
```

3. **List the names and city of all clients who have exactly 12 characters in length and starts with ‘I’.**

   SQL Command:

   select name, city from client_master where length(name)=12 and name like 'I%' ;

   Output:

```
SQL> select name, city from client_master where length(name)=12 and name like 'I%';

NAME                          CITY
----------------------------- ---------------
Ivan Bayross                  Bombay
```

4. **Find the list of all clients who stay in ‘Bombay’ or ‘Delhi’.**

   SQL Command:

   select name from client_master where city='Bombay' or city='Delhi' ;

Output:

```
SQL> select name from client_master where city='Bombay' or city='Delhi';

NAME
------------------------------
Ivan Bayross
Basu Navindgi
Ravi Sreedharan
Rukmini
```

5. **Print the list of all clients whose bal_due is greater than value 10,000.**

SQL Command:

select name from client_master where balance_due>10000;

Output:

```
SQL> select name from client_master where balance_due>10000;

NAME
------------------------------
Ivan Bayross
```

6. **Print the information from sales_order table for orders places in the month of January.**

SQL Command:

select * from sales_order where to_char (order_date,'Mon')='Jan' ;

Output:

```
SQL> select * from sales_order where to_char (order_date,'Mon')='Jan' ;

ORDER_ ORDER_DAT CLIENT SALESM D B DELIVERY_ ORDER_STAT
------ --------- ------ ------ - - --------- ----------
O19001 12-JAN-96 C001   S001   F N 20-JAN-96 InProcess
O19002 25-JAN-96 C002   S002   P N 27-JAN-96 BackOrder
```

7. **Display the order information for client_no 'C001' and 'C002'.**

SQL Command:

select * from sales_order where client_no in ('C001', 'C002');

Output:

```
SQL> select * from sales_order where client_no in ('C001', 'C002');

ORDER_ ORDER_DAT CLIENT SALESM D B DELIVERY_ ORDER_STAT
------ --------- ------ ------ - - --------- ----------
O19001 12-JAN-96 C001   S001   F N 20-JAN-96 InProcess
O19002 25-JAN-96 C002   S002   P N 27-JAN-96 BackOrder
O19003 03-APR-96 C001   S001   F Y 07-APR-96 Fullfilled
```

8. Find products whose selling price greater than 2000 and less than 5000.

SQL Command:

select * from product_master where Sell_price>2000 and sell_price<5000;

Output:

```
SQL> select * from product_master where Sell_price>2000 and sell_price<5000;

PRODUC DESCRIPTION                                PROFIT_PERCENT UNIT_MEASU
------ ---------------------------------------- -------------- ----------
QTY_ON_HAND REORDER_LEVEL SELL_PRICE COST_PRICE
----------- ------------- ---------- ----------
P07868 Keyboard                                            2 Piece
        10             3       3150       3050
```

9. Find products whose selling price is more than 1500.Calculate a new selling price as original selling price*1.15. Rename the new column in the above query is New_price.

SQL Command:

select Sell_price, Sell_price*1.15 New_Price from product_master;

Output:

```
SQL> select Sell_price, Sell_price*1.15 New_Price from product_master;

SELL_PRICE  NEW_PRICE
---------- ----------
       525     603.75
     12000      13800
      1050     1207.5
       525     603.75
      3150     3622.5
      5250     6037.5
      8400       9660
      1050     1207.5
      1025    1178.75

9 rows selected.
```

10. List the names, city and state of clients who are not in the state of 'Maharastra'.

SQL Command:

select name, city, state from Client_master where State!='Maharashtra' ;

Output:

```
SQL> select name, city, state from Client_master where State!='Maharashtra' ;

NAME                           CITY           STATE
------------------------------ -------------- --------------
Ivan Bayross                   Bombay         Maharastra
Vandana Satiwal                Madras         Tamil Nadu
Pramada Jaguste                Kolkata        Weste Bengal
Basu Navindgi                  Bombay         Maharastra
Ravi Sreedharan                Delhi          Delhi
Rukmini                        Bombay         Maharastra

6 rows selected.
```

11. Display the month (in alphabets) and date when the order must be delivered.

SQL Command:

select to_char(delivery_date, 'Month-dd') from Sales_order;

Output:

```
SQL> select to_char(delivery_date, 'Month-dd') from Sales_order;

TO_CHAR(DELIVERY_DATE,'MONTH-DD')
-------------------------------------
January  -20
January  -27
February -20
April    -07
May      -22
May      -26

6 rows selected.
```

12. Display the Order_date in the format 'DD-Month-YY' e.g., 12-February-13.

SQL Command:

select to_char(order_date, 'DD-Month-YY') O_date from Sales_order;

Output:

```
SQL> select to_char(order_date, 'DD-Month-YY') O_date from Sales_order;

O_DATE
----------------------------------------
12-January  -96
25-January  -96
18-February -96
03-April    -96
20-May      -96
24-May      -96

6 rows selected.
```

13. Find the date, 15 days after today's date

SQL Command:

select sysdate+15 New_date from dual;

Output:

```
SQL> select sysdate+15 New_date from dual;

NEW_DATE
---------
15-MAR-23
```

ASSIGNMENT-3

1. **Count the total number of orders.**

   SQL Command:

   select count(*) from sales_order;

   Output:

```
SQL> select count(*) from sales_order;

  COUNT(*)
----------
         6
```

2. **Calculate the average price of all the products.**

   SQL Command:

   select avg(sell_price) from product_master;

   select avg(cost_price) from product_master;

   Output:

```
SQL> select avg(sell_price) from product_master
  2  ;

AVG(SELL_PRICE)
---------------
     3663.88889
```

```
SQL> select avg(cost_price) from product_master;

AVG(COST_PRICE)
---------------
     3464.44444
```

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

   SQL Command:

   select count(*) from product_master where (cost_price>1500);

   Output:

```
SQL> select count(*) from product_master where (cost_price>1500);

  COUNT(*)
----------
         4
```

4. Determine the maximum and minimum product prices. Rename the output as max_price and min_price respectively.

SQL Command:

select min(cost_price) MIN_PRICE,max(cost_price) MAX_PRICE from product_master;

Output:

```
SQL> select min(cost_price) MIN_PRICE,max(cost_price) MAX_PRICE from product_master;

 MIN_PRICE  MAX_PRICE
---------- ----------
       500      11280
```

5. Change the City of the Client_no 'C005' to 'Madras'.

SQL Command:

update client_master set city='MADRAS' where client_no='C005';

Output:

```
SQL> update client_master set city='MADRAS' where client_no='C005'
  2  ;

1 row updated.

SQL> select * from client_master
  2  ;

CLIENT NAME                          ADDRESS1                     ADDRESS2                     CITY           STATE            PINCODE BALANCE_DUE
------ ----------------------------- ---------------------------- ---------------------------- -------------- -------------- ------- -----------
C001   Ivan Bayross                  P-76                         Worli                        Bombay         Maharastra       400054       15000
C002   Vandana Satiwal               128                          Adams Street                 Madras         Tamil Nadu       780001           0
C003   Pramada Jaguste               157                          Gopalpur                     Kolkata        Weste Bengal      70058        5000
C004   Basu Navindgi                 A/12                         Nariman                      Bombay         Maharastra       400056           0
C005   Ravi Sreedharan               B/34                         Rajnagar                     MADRAS         Delhi            100001        2000
C006   Rukmini                       Q-12                         Bandra                       Bombay         Maharastra       400050           0

6 rows selected.
```

6. Change the Bal_due of Client_no 'C005' to Rs.3000/-.

SQL Command:

update client_master set balance_due='3000' where client_no='C005';

Output:

```
SQL> update client_master set balance_due='3000' where client_no='C005';

1 row updated.

SQL> select * from client_master
  2  ;

CLIENT NAME                          ADDRESS1                     ADDRESS2                     CITY           STATE            PINCODE BALANCE_DUE
------ ----------------------------- ---------------------------- ---------------------------- -------------- -------------- ------- -----------
C001   Ivan Bayross                  P-76                         Worli                        Bombay         Maharastra       400054       15000
C002   Vandana Satiwal               128                          Adams Street                 Madras         Tamil Nadu       780001           0
C003   Pramada Jaguste               157                          Gopalpur                     Kolkata        Weste Bengal      70058        5000
C004   Basu Navindgi                 A/12                         Nariman                      Bombay         Maharastra       400056           0
C005   Ravi Sreedharan               B/34                         Rajnagar                     MADRAS         Delhi            100001        3000
C006   Rukmini                       Q-12                         Bandra                       Bombay         Maharastra       400050           0

6 rows selected.
```

7. Delete from client_master where the column state holds the value ‘Tamil Nadu’.

SQL Command:

delete from client_master where state='Tamil Nadu';

Output:

```
SQL> delete from client_master where state='Tamil Nadu';
delete from client_master where state='Tamil Nadu'
*
ERROR at line 1:
ORA-02292: integrity constraint (CSE30.SYS_C00102035) violated - child record found
```

8. Add a column called ‘Telephone’ of data type ‘number’ and size 10 in the table client_master.

SQL Command:

Alter table client_master add(Telephone number(10));

Output:

```
SQL> Alter table client_master add(Telephone number(10));

Table altered.

SQL> select * from client_master
  2  ;

CLIENT NAME                          ADDRESS1                     ADDRESS2                     CITY           STATE             PINCODE BALANCE_DUE  TELEPHONE
------ ----------------------------- ---------------------------- ---------------------------- -------------- --------------- ------- ----------- ----------
C001   Ivan Bayross                  P-76                         Worli                        Bombay         Maharastra        400054       15000
C002   Vandana Satiwal               128                          Adams Street                 Madras         Tamil Nadu        780001           0
C003   Pramada Jaguste               157                          Gopalpur                     Kolkata        Weste Bengal       70058        5000
C004   Basu Navindgi                 A/12                         Nariman                      Bombay         Maharastra        400056           0
C005   Ravi Sreedharan               B/34                         Rajnagar                     MADRAS         Delhi             100001        3000
C006   Rukmini                       Q-12                         Bandra                       Bombay         Maharastra        400050           0

6 rows selected.
```

9. Change the size of data type Pin_code to 10 in the table client_master

SQL Command:

Alter table client_master modify(pincode number(10));

Output:

```
SQL> Alter table client_master modify(pincode number(10));

Table altered.

SQL> select * from client_master
  2  ;

CLIENT NAME                          ADDRESS1                     ADDRESS2                     CITY           STATE             PINCODE BALANCE_DUE  TELEPHONE
------ ----------------------------- ---------------------------- ---------------------------- -------------- --------------- ------- ----------- ----------
C001   Ivan Bayross                  P-76                         Worli                        Bombay         Maharastra        400054       15000
C002   Vandana Satiwal               128                          Adams Street                 Madras         Tamil Nadu        780001           0
C003   Pramada Jaguste               157                          Gopalpur                     Kolkata        Weste Bengal       70058        5000
C004   Basu Navindgi                 A/12                         Nariman                      Bombay         Maharastra        400056           0
C005   Ravi Sreedharan               B/34                         Rajnagar                     MADRAS         Delhi             100001        3000
C006   Rukmini                       Q-12                         Bandra                       Bombay         Maharastra        400050           0

6 rows selected.

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

10. Drop the column Address2 from the table client_master

SQL Command:

Alter table client_master drop(address2);

Output:

```
SQL> Alter table client_master drop(address2);

Table altered.

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

SQL> select * from client_master
  2  ;

CLIENT NAME                           ADDRESS1                      CITY           STATE              PINCODE BALANCE_DUE  TELEPHONE
------ ---------------------------- ---------------------------- --------------- --------------- ---------- ----------- ----------
C001   Ivan Bayross                  P-76                          Bombay         Maharastra          400054       15000
C002   Vandana Satiwal               128                           Madras         Tamil Nadu          780001           0
C003   Pramada Jaguste              157                           Kolkata        Weste Bengal         70058        5000
C004   Basu Navindgi                 A/12                          Bombay         Maharastra          400056           0
C005   Ravi Sreedharan               B/34                          MADRAS         Delhi               100001        3000
C006   Rukmini                       Q-12                          Bombay         Maharastra          400050           0

6 rows selected.
```

11. Create another table client_master_duplicate with the same structure of client_master (without copying the data of the table client_master).

SQL Command:

create table client_master_duplicate as select * from client_master where 1=2;

Output:

```
SQL> create table client_master_duplicate as select * from client_master where 1=2;

Table created.

SQL> select * from client_master_duplicate;

no rows selected

SQL> desc client_master_duplicate;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_NO                                          VARCHAR2(6)
 NAME                                      NOT NULL VARCHAR2(30)
 ADDRESS1                                           VARCHAR2(30)
 CITY                                               VARCHAR2(15)
 STATE                                              VARCHAR2(15)
 PINCODE                                            NUMBER(10)
 BALANCE_DUE                                        NUMBER(10,2)
 TELEPHONE                                          NUMBER(10)

SQL> insert into client_master_duplicate select  * from client_master;

6 rows created.
```

12. Insert the data into client_master_duplicate table from client_master table.

SQL Command:

insert into client_master_duplicate select * from client_master;

Output:

```
SQL> insert into client_master_duplicate select  * from client_master;

6 rows created.
```

```
CLIENT NAME                          ADDRESS1                       CITY           STATE            PINCODE BALANCE_DUE  TELEPHONE
------ ------------------------------ ------------------------------ --------------- --------------- ---------- ----------- ----------
C001   Ivan Bayross                   P-76                           Bombay         Maharastra         400054       15000
C002   Vandana Satiwal                128                            Madras         TamilNadu          780001           0
C003   Pramada Jaguste                157                            Kolkata        West Bengal        700058        5000
C004   Basu Navindgi                  A/12                           Bombay         Maharastra         400056           0
C005   Ravi Sreedharan                B/34                           Madras         Delhi              100001        3000
C006   Rukmini                        Q-12                           Bombay         Maharastra         400050           0

6 rows selected.
```

13. Rename the table client_master_duplicate to c_master.

SQL Command:

rename client_master_duplicate to c_master;

Output:

```
SQL> rename client_master_duplicate to c_master;

Table renamed.

SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
STUDENTTABLE                   TABLE
STUDENT                        TABLE
STD                            TABLE
SALES_ORDER_DETAILS            TABLE
SALES_ORDER                    TABLE
SALESMAN_MASTER                TABLE
PRODUCT_MASTER                 TABLE
DEPT                           TABLE
DATAFLAIR                      TABLE
C_MASTER                       TABLE
CLIENT_MASTER                  TABLE

11 rows selected.
```

14. Destroy the table c_master with its data.

SQL Command:

drop table c_master;

Output:

```
SQL> drop table c_master;

Table dropped.

SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
STUDENTTABLE                   TABLE
STUDENT                        TABLE
STD                            TABLE
SALES_ORDER_DETAILS            TABLE
SALES_ORDER                    TABLE
SALESMAN_MASTER                TABLE
PRODUCT_MASTER                 TABLE
DEPT                           TABLE
DATAFLAIR                      TABLE
CLIENT_MASTER                  TABLE
BIN$EOi+Ux8XQo6Qa5U/AetZnA==$0 TABLE

11 rows selected.
```

ASSIGNMENT-4

Dept Table Creation and Insertion:

```
create table dept(
        deptno number(2,0),
        dname  varchar2(14),
        loc    varchar2(13),
        constraint pk_dept primary key (deptno)
);

insert into dept
values(10, 'ACCOUNTING', 'NEW YORK');
insert into dept
values(20, 'RESEARCH', 'DALLAS');
insert into dept
values(30, 'SALES', 'CHICAGO');
insert into dept
values(40, 'OPERATIONS', 'BOSTON');
```

Output:

```
select * from dept;

DEPTNO DNAME          LOC
------ -------------- -------------
    10 ACCOUNTING     NEW YORK
    20 RESEARCH       DALLAS
    30 SALES          CHICAGO
    40 OPERATIONS     BOSTON
```

Emp Table Creation and Insertion:

```
create table emp(
        empno number(4,0),
        ename varchar2(10),
        job varchar2(9),
        mgr number(4,0),
        hiredate date,
        sal number(7,2),
        comm number(7,2),
        deptno number(2,0),
```

```
        constraint pk_emp primary key (empno),
        constraint fk_deptno foreign key (deptno) references dept (deptno)
);


insert into emp
values(7839, 'KING', 'PRESIDENT', null, to_date('17-11-1981','dd-mm-yyyy'), 5000, null, 10);
insert into emp
values(7698, 'BLAKE', 'MANAGER', 7839, to_date('1-5-1981','dd-mm-yyyy'), 2850, null, 30);
insert into emp
values(7782, 'CLARK', 'MANAGER', 7839, to_date('9-6-1981','dd-mm-yyyy'),2450, null, 10);
insert into emp
values(7566, 'JONES', 'MANAGER', 7839, to_date('2-4-1981','dd-mm-yyyy'),2975, null, 20);
insert into emp
values(7788, 'SCOTT', 'ANALYST', 7566, to_date('13-JUL-87','dd-mm-rr') - 85,3000, null, 20);
insert into emp
values(7902,'FORD', 'ANALYST', 7566, to_date('3-12-1981','dd-mm-yyyy'),3000, null, 20);
insert into emp
values(7369, 'SMITH', 'CLERK', 7902, to_date('17-12-1980','dd-mm-yyyy'),800, null, 20);
insert into emp
values(7499, 'ALLEN', 'SALESMAN', 7698, to_date('20-2-1981','dd-mm-yyyy'),1600, 300, 30);
insert into emp
values(7521, 'WARD', 'SALESMAN', 7698, to_date('22-2-1981','dd-mm-yyyy'),1250, 500, 30);
insert into emp
values(7654, 'MARTIN', 'SALESMAN', 7698, to_date('28-9-1981','dd-mm-yyyy'),1250, 1400, 30);
insert into emp
values(7844, 'TURNER', 'SALESMAN', 7698, to_date('8-9-1981','dd-mm-yyyy'),1500, 0, 30);
insert into emp
values(7876, 'ADAMS', 'CLERK', 7788, to_date('13-JUL-87','dd-mm-yyyy'),1100, null, 20);
insert into emp
values(7900, 'JAMES', 'CLERK', 7698, to_date('3-12-1981','dd-mm-yyyy'),950, null, 30);
insert into emp
values(7934,'MILLER','CLERK', 7782, to_date('23-1-1982','dd-mm-yyyy'),1300, null, 10);
```

Output:

```
SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER          7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER          7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER          7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST          7566 19-APR-87       3000                    20
      7902 FORD       ANALYST          7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK            7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN         7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN         7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN         7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN         7698 08-SEP-81       1500          0         30

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7876 ADAMS      CLERK            7788 13-JUL-87       1100                    20
      7900 JAMES      CLERK            7698 03-DEC-81        950                    30
      7934 MILLER     CLERK            7782 23-JAN-82       1300                    10

14 rows selected.
```

1. **Display the names of all employees' right aligning them to 15 characters.**

   SQL Command:

   select lpad(empno,15) from emp;

   Output:

```
LPAD(EMPNO,15)
------------------------------
           7369
           7499
           7521
           7566
           7654
           7698
           7782
           7788
           7839
           7844
           7876
```

2. **Display the names of all employees' padding them to the right up to 15 characters with '*'.**

   SQL Command:

   select lpad(empno,15,'*') from emp;

   Output:

```
LPAD(EMPNO,15,'*')
--------------------
***********7369
***********7499
***********7521
***********7566
***********7654
***********7698
***********7782
***********7788
***********7839
***********7844
***********7876
```

3. **Find the details of all the managers in department 10 and all clerks in department 20 and all employees who are neither managers nor clerks but whose salary is more than or equal to 2000/-.**

SQL Command:

select * from emp where (job='MANAGER' and deptno=10) or (job='CLERK' and deptno=20) or (job!='MANAGER' and job != 'CLERK'and sal>=2000);

Output:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7876	ADAMS	CLERK	7788	13-JUL-87	1100		20

6 rows selected.

4. **List all the employees who have joined between 01/02/81 and 31/08/81.**

SQL Command:

select ename,hiredate from emp where hiredate between '1-FEB-81' and '31-AUG-81';

Output:

ENAME	HIREDATE
BLAKE	01-MAY-81
CLARK	09-JUN-81
JONES	02-APR-81
ALLEN	20-FEB-81
WARD	22-FEB-81

5. **List all the employees who were joined as manager during 1981.**

SQL Command:

select ename,hiredate,job from emp where job='MANAGER' and to_char(hiredate,'YY')=81;

Output:

ENAME	HIREDATE	JOB
BLAKE	01-MAY-81	MANAGER
CLARK	09-JUN-81	MANAGER
JONES	02-APR-81	MANAGER

6. List the employees whose salaries are 800, 1600 or 2450.

SQL Command:

select ename, sal from emp where sal in (800,1600,2450);

Output:

```
ENAME             SAL
---------- ----------
CLARK            2450
SMITH             800
ALLEN            1600
```

7. List the names of all employees who are either 'clerks' or 'salesman' or 'analyst'.

SQL Command:

select ename, job from emp where job in ('MANAGER','CLERK','ANALYST');

Output:

```
ENAME      JOB
---------- ---------
BLAKE      MANAGER
CLARK      MANAGER
JONES      MANAGER
SCOTT      ANALYST
FORD       ANALYST
SMITH      CLERK
ADAMS      CLERK
JAMES      CLERK
MILLER     CLERK
```

8. List the total number of employees and the average salaries of the different departments.

SQL Command:

select count(ename), avg(sal) from emp group by deptno;

Output:

```
COUNT(ENAME)   AVG(SAL)
------------ ----------
           6 1566.66667
           5       2175
           3 2916.66667
```

9. Calculate the average salary of all employees whose department is 30.

SQL Command:

select avg(sal) from emp where deptno=30;

Output:

```
  AVG(SAL)
----------
1566.66667
```

10. Calculate the minimum salary earn by clerks.

SQL Command:

select min(sal) from emp where job='CLERK';

Output:

```
  MIN(SAL)
----------
       800
```

11. Calculate the maximum salary earn by salesmen.

SQL Command:

select max(sal) from emp where job='SALESMAN';

Output:

```
  MAX(SAL)
----------
      1600
```

12. ----------

13. Calculate the no. of employees who are not getting any commission.

SQL Command:

select count(ename) from emp where comm is NULL;

Output:

```
COUNT(ENAME)
------------
          10
```

14. Find the department is not having any employee.

SQL Command:

select deptno from emp;

select dname from dept where deptno not in (10,20,30);

Output:

```
    DEPTNO
----------
        10
        30
        10
        20
        20
        20
        20
        30
        30
        30
        30

    DEPTNO
----------
        20
        30
        10
```

```
SQL> select dname from dept where deptno not in (10,20,30);

DNAME
--------------
OPERATIONS
```

ASSIGNMENT-5

1. List all the employee names, dept name and the city, in department name order.

SQL command:

select ename,dname,loc from emp,dept where emp.deptno=dept.deptno order by dname;

Output:

```
SQL> select ename,dname,loc from emp,dept where emp.deptno=dept.deptno order by dname;

ENAME      DNAME          LOC
---------- -------------- -------------
CLARK      ACCOUNTING     NEW YORK
MILLER     ACCOUNTING     NEW YORK
KING       ACCOUNTING     NEW YORK
FORD       RESEARCH       DALLAS
SCOTT      RESEARCH       DALLAS
JONES      RESEARCH       DALLAS
SMITH      RESEARCH       DALLAS
ADAMS      RESEARCH       DALLAS
WARD       SALES          CHICAGO
MARTIN     SALES          CHICAGO
TURNER     SALES          CHICAGO

ENAME      DNAME          LOC
---------- -------------- -------------
JAMES      SALES          CHICAGO
ALLEN      SALES          CHICAGO
BLAKE      SALES          CHICAGO

14 rows selected.
```

2. List all employees working in Dallas in descending order of salary.

SQL command:

select * from emp where deptno in (select deptno from dept where loc='DALLAS') order by sal desc;

Output:

```
SQL> select * from emp where deptno in (select deptno from dept where loc='DALLAS') order by sal desc;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7876 ADAMS      CLERK           7788 13-JUL-87       1100                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
```

3. List employee name, department name, job and location of all employees who work in DALLAS.

SQL command:

select ename,dname,job,loc from emp,dept where emp.deptno=dept.deptno and loc='DALLAS';

Output:

```
SQL> select ename,dname,job,loc from emp,dept where emp.deptno=dept.deptno and loc='DALLAS';

ENAME      DNAME          JOB         LOC
---------- -------------- --------- -------------
JONES      RESEARCH       MANAGER     DALLAS
SCOTT      RESEARCH       ANALYST     DALLAS
FORD       RESEARCH       ANALYST     DALLAS
SMITH      RESEARCH       CLERK       DALLAS
ADAMS      RESEARCH       CLERK       DALLAS
```

4. List the employee name, salary, PF, HRA, DA and gross salary; order the result in ascending order of gross. PF is 10% of salary, HRA is 60% of salary and DA is 40% of salary.

SQL command:

select ename,sal,sal*0.1 PF,sal*0.6 HRA, sal*0.4 DA,(sal*0.1+sal*0.6+sal*0.4+sal) Gross_sal from emp order by Gross_sal;

Output:

```
SQL> select ename,sal,sal*0.1 PF,sal*0.6 HRA, sal*0.4 DA,(sal*0.1+sal*0.6+sal*0.4+sal) Gross_sal from emp order by Gross_sal;

ENAME             SAL         PF        HRA         DA  GROSS_SAL
---------- ---------- ---------- ---------- ---------- ----------
SMITH             800         80        480        320       1680
JAMES             950         95        570        380       1995
ADAMS            1100        110        660        440       2310
MARTIN           1250        125        750        500       2625
WARD             1250        125        750        500       2625
MILLER           1300        130        780        520       2730
TURNER           1500        150        900        600       3150
ALLEN            1600        160        960        640       3360
CLARK            2450        245       1470        980       5145
BLAKE            2850        285       1710       1140       5985
JONES            2975      297.5       1785       1190     6247.5

ENAME             SAL         PF        HRA         DA  GROSS_SAL
---------- ---------- ---------- ---------- ---------- ----------
FORD             3000        300       1800       1200       6300
SCOTT            3000        300       1800       1200       6300
KING             5000        500       3000       2000      10500

14 rows selected.
```

5. Display names and salary of all the employees who report to KING.

SQL command:

select ename,sal from emp where mgr=(select empno from emp where ename='KING');

Output:

```
SQL> select ename,sal from emp where mgr=(select empno from emp where ename='KING');

ENAME             SAL
---------- ----------
BLAKE             2850
CLARK             2450
JONES             2975
```

6. List all employees who work in DALLAS and earn more than any employee working in Chicago.

SQL command:

select ename, loc, sal from emp, dept where emp.deptno=dept.deptno and loc = 'DALLAS' and sal>(select max(sal) from emp, dept where loc= 'CHICAGO' and emp.deptno=dept.deptno);

Output:

```
SQL> select ename, loc, sal from emp, dept where emp.deptno=dept.deptno and loc = 'DALLAS' and sal>(select max(sal) from emp, dept where loc= 'CHICAGO'
and emp.deptno=dept.deptno);

ENAME      LOC                 SAL
---------- ------------- ----------
JONES      DALLAS              2975
SCOTT      DALLAS              3000
FORD       DALLAS              3000
```

7. List all employees who work in the same post as Smith.

SQL command:

Select ename, job from emp where job=(select job from emp where ename= 'SMITH');

Output:

```
SQL> Select ename, job from emp where job=(select job from emp where ename= 'SMITH');

ENAME      JOB
---------- ---------
SMITH      CLERK
ADAMS      CLERK
JAMES      CLERK
MILLER     CLERK
```

8. Find the job with the highest average salary.

SQL command:

Select job from emp where sal = (select max(avg(sal)) from emp group by job);

Output:

```
SQL> Select job from emp where sal = (select max(avg(sal)) from emp group by job);

JOB
---------
PRESIDENT
```

9. List the top 10 earners in the company.

SQL command:

Select ename, sal from (select ename, sal from emp order by sal desc) where rownum <=10;

Output:

```
SQL> Select ename, sal from (select ename, sal from emp order by sal desc) where rownum <=10;

ENAME             SAL
---------- ----------
KING             5000
SCOTT            3000
FORD             3000
JONES            2975
BLAKE            2850
CLARK            2450
ALLEN            1600
TURNER           1500
MILLER           1300
WARD             1250

10 rows selected.
```

10. Display the names of all employees' replacing 'A' with 'a'.

SQL command:

Select replace (ename, 'A', 'a') from emp;

Output:

```
SQL> Select replace (ename, 'A', 'a') from emp;

REPLACE(EN
----------
KING
BLaKE
CLaRK
JONES
SCOTT
FORD
SMITH
aLLEN
WaRD
MaRTIN
TURNER

REPLACE(EN
----------
aDaMS
JaMES
MILLER

14 rows selected.
```

11. Show the salary of all the employees rounding it to the nearest Rs.1000/-.

SQL command:

Select ename, sal, round (sal, -3) from emp;

Output:

```
SQL> Select ename, sal, round (sal, -3) from emp;

ENAME             SAL ROUND(SAL,-3)
---------- ---------- -------------
KING             5000          5000
BLAKE            2850          3000
CLARK            2450          2000
JONES            2975          3000
SCOTT            3000          3000
FORD             3000          3000
SMITH             800          1000
ALLEN            1600          2000
WARD             1250          1000
MARTIN           1250          1000
TURNER           1500          2000

ENAME             SAL ROUND(SAL,-3)
---------- ---------- -------------
ADAMS            1100          1000
JAMES             950          1000
MILLER           1300          1000

14 rows selected.
```

12. Show the first three and last three characters of the names of all the employees.

SQL command:

Select substr(ename,1,3), substr(ename, -3) from emp;

Output:

```
SQL> Select substr(ename,1,3), substr(ename, -3) from emp;

SUBSTR(ENAME SUBSTR(ENAME
------------ ------------
KIN          ING
BLA          AKE
CLA          ARK
JON          NES
SCO          OTT
FOR          ORD
SMI          ITH
ALL          LEN
WAR          ARD
MAR          TIN
TUR          NER

SUBSTR(ENAME SUBSTR(ENAME
------------ ------------
ADA          AMS
JAM          MES
MIL          LER

14 rows selected.
```

ASSIGNMENT – 6

Table: Client_master

Column_Name	Data type	Size	Attributes
Client_no	Varchar2	8	Primary Key
Name	Varchar2	20	Not Null
Address1	Varchar2	20	Not Null
Address2	Varchar2	20	
City	Varchar2	15	
State	Varchar2	15	
Pincode	Varchar2	8	
Bal_due	Number	8,3	

1. **Create a view vw_client_master using Client_no, Name, Address1 and Bal_due**

   a. Insert at least 3 records to vw_client_master.

   b. Update a record to vw_client_master.

   c. Delete a record from vw_client_master.

   And check that the due to the above operation if the base table is affected or not.

SQL COMMANDS:

```
create view vw_client_master as select client_no, Name, Address1, Balance_due from client_master;
    insert into vw_client_master values ('C007', 'ABHISEK SINGH', 'BALLY', 15000);
    insert into vw_client_master values ('C008', 'DIPAK DAS', 'SALTLAKE', 10000);
    insert into vw_client_master values ('C009', 'RAMAN GUPTA', 'HOWRAH', 20000);
    Update vw_client_master set Balance_due=8000 where client_no = 'C008';
    Delete from vw_client_master where client_no = 'C007';

select * from vw_client_master;
select * from client_master;
```

OUTPUT:

```
SQL> create view vw_client_master as select client_no, Name, Address1, Balance_due from client_master;
View created.
SQL> insert into vw_client_master values ('C007', 'ABHISEK SINGH', 'BALLY', 15000);
1 row created.
SQL> insert into vw_client_master values ('C008', 'DIPAK DAS', 'SALTLAKE', 10000);
1 row created.
SQL> insert into vw_client_master values ('C009', 'RAMAN GUPTA', 'HOWRAH', 20000);
1 row created.
SQL> Update vw_client_master set Balance_due=8000 where client_no = 'C008';
1 row updated.
SQL> Delete from vw_client_master where client_no = 'C007';
1 row deleted.
```

```
SQL> select * from vw_client_master;

CLIENT NAME                           ADDRESS1                        BALANCE_DUE
------ ------------------------------ ------------------------------ -----------
C008   DIPAK DAS                      SALTLAKE                              8000
C009   RAMAN GUPTA                    HOWRAH                               20000
C001   Ivan Bayross                   P-76                                 15000
C002   Vandana Satiwal                128                                      0
C003   Pramada Jaguste                157                                   5000
C004   Basu Navindgi                  A/12                                     0
C005   Ravi Sreedharan                B/34                                  3000
C006   Rukmini                        Q-12                                     0

8 rows selected.

SQL> set linesize 500;
SQL> select * from client_master;

CLIENT NAME                           ADDRESS1                        CITY           STATE              PINCODE BALANCE_DUE  TELEPHONE
------ ------------------------------ ------------------------------ --------------- --------------- ---------- ----------- ----------
C008   DIPAK DAS                      SALTLAKE                                                                        8000
C009   RAMAN GUPTA                    HOWRAH                                                                         20000
C001   Ivan Bayross                   P-76                           Bombay          Maharastra          400054       15000
C002   Vandana Satiwal                128                            Madras          Tamil Nadu          780001           0
C003   Pramada Jaguste                157                            Kolkata         Weste Bengal         70058        5000
C004   Basu Navindgi                  A/12                           Bombay          Maharastra          400056           0
C005   Ravi Sreedharan                B/34                           MADRAS          Delhi               100001        3000
C006   Rukmini                        Q-12                           Bombay          Maharastra          400050           0

8 rows selected.
```

2. Create a view Vw_sales_det using Client_no, Order_no, Order_date, Product_no, Qty_ordered, and order_status for all order which have already marked as 'Backorder'. (Using the tables sales_order, sales_order_details).

a. Insert a record to vw_sales_det.

b. Update the client_no for a particular order_no.

c. Delete a record.

d. Remove the views from database.

<u>SQL COMMANDS:</u>

create view vw_sales_det as select s1.client_no, s1.order_no, s1.order_date, s2.product_no, s2.qty_ordered, s1.order_status from sales_order s1, sales_order_details s2 where s1.order_no=s2.order_no and s1.order_status = 'BackOrder';

insert into vw_sales_det values ('C007', 'O19251', '12-Jan-91', 'P00091', 100, 'InProcess');

update vw_sales_det set client_no = 'C006' where order_no = 'O19002';

select * from vw_sales_det;

delete from vw_sales_det where client_no= 'C005';

select * from vw_sales_det;

drop view vw_sales_det;

OUTPUT:

```
SQL> create view vw_sales_det as select s1.client_no, s1.order_no, s1.order_date, s2.product_no, s2.qty_ordered, s1.order_status from sales_order s1, sales_order_details s2 where
s1.order_no=s2.order_no and s1.order_status = 'BackOrder';

View created.

SQL> insert into vw_sales_det values ('C007', 'O19251', '12-Jan-91', 'P00091', 100, 'InProcess');
insert into vw_sales_det values ('C007', 'O19251', '12-Jan-91', 'P00091', 100, 'InProcess')
*
ERROR at line 1:
ORA-01779: cannot modify a column which maps to a non key-preserved table


SQL> update vw_sales_det set client_no = 'C006' where order_no = 'O19002';
update vw_sales_det set client_no = 'C006' where order_no = 'O19002'
                    *
ERROR at line 1:
ORA-01779: cannot modify a column which maps to a non key-preserved table
```

```
SQL> select * from vw_sales_det;

CLIENT ORDER_ ORDER_DAT PRODUC QTY_ORDERED ORDER_STAT
------ ------ --------- ------ ----------- ----------
C002   O19002 25-JAN-96 P00001          10 BackOrder

SQL> delete from vw_sales_det where client_no= 'C005';

0 rows deleted.

SQL> select * from vw_sales_det;

CLIENT ORDER_ ORDER_DAT PRODUC QTY_ORDERED ORDER_STAT
------ ------ --------- ------ ----------- ----------
C002   O19002 25-JAN-96 P00001          10 BackOrder

SQL> drop view vw_sales_det;

View dropped.
```

ASSIGNMENT-7

1. Write a PL/SQL code for finding factorial of a given number

PL/SQL commands:

```
set serveroutput on
declare
    n number;
    i number;
    f number:=1;
begin
    n:=&x;
    for i in 1..n
    loop
        f:=f*i;
    end loop;
    dbms_output.put_line('Factorial of '||n||' is ' || f);
end;
/
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_1.sql
Enter value for x: 5
old   6:    n:=&x;
new   6:    n:=5;
Factorial of 5 is 120

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL code for calculating finding the sum of N numbers.

PL/SQL commands:

```
set serveroutput on
declare
    n number;
    i number;
    s number:=0;
begin
    n:=&n;
    for i in 1..n
    loop
```

```
            s:=s+i;
        end loop;
        dbms_output.put_line('Sum of first '||n||' numbers is ' || s);
    end;
    /
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_2.sql
Enter value for n: 10
old   6:    n:=&n;
new   6:    n:=10;
Sum of first 10 numbers is 55

PL/SQL procedure successfully completed.
```

3. Write a PL/SQL code for finds a given year is leap year or not.

PL/SQL commands:

```
set serveroutput on
declare
        y number;
begin
        y:=&x;
        if(mod(y,400)=0)then
                dbms_output.put_line('Leap Year');
        elsif((mod(y,4)=0) and (mod(y,100)!=0))then
                dbms_output.put_line('Leap Year');
        else
                dbms_output.put_line('NOT a Leap Year');
        end if;
end;
/
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_3.sql
Enter value for x: 2004
old   4:    y:=&x;
new   4:    y:=2004;
Leap Year

PL/SQL procedure successfully completed.
```

4. Write a PL/SQL code for finding maximum of three numbers. (Input will be given by the user).

PL/SQL commands:

```
set serveroutput on
declare
        a number;
        b number;
```

```
    c number;
begin
    a:=&a;
    b:=&b;
    c:=&c;
    if(a>b and a>c)then
            dbms_output.put_line(a || ' is the maximum');
    elsif(b>c)then
            dbms_output.put_line(b || ' is the maximum');
    else
            dbms_output.put_line(c || ' is the maximum');
    end if;
end;
/
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_4.sql
Enter value for a: 7
old   6:    a:=&a;
new   6:    a:=7;
Enter value for b: 21
old   7:    b:=&b;
new   7:    b:=21;
Enter value for c: 10
old   8:    c:=&c;
new   8:    c:=10;
21 is the maximum

PL/SQL procedure successfully completed.
```

5. **Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 6 to 10. Store the radius and corresponding values of calculated area in an empty table named Areas, Consisting of two columns Radius and Area.**

PL/SQL commands:

```
set serveroutput on
drop table Areas;
create table Areas(radius number(5,3), area number(10,3));
declare
    r number;
    pi constant number(8,2):=3.14;
    area number(10,2);
begin
```

```
    for r in 6..10
    loop
        area:=pi*power(r,2);
        insert into Areas values(r,area);
    end loop;
end;
/
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_5.sql

Table dropped.


Table created.


PL/SQL procedure successfully completed.

SQL> select * from Areas;

    RADIUS       AREA
---------- ----------
         6     113.04
         7     153.86
         8     200.96
         9     254.34
        10        314
```

6. **Write a PL/SQL code block that will accept a client_no from the user and adds the amount of Rs. 1000 to bal_due column, has a minimum balance of Rs. 6000. The process is fire on client_master.**

PL/SQL commands:

```
set serveroutput on
declare
    cli_no varchar2(6):= '&client_no';
    t_c_no number(10,2);
begin
    select balance_due into t_c_no from client_master where client_no=cli_no;
    if(t_c_no>=6000)then
        t_c_no := t_c_no + 1000;
        update client_master set balance_due=t_c_no where client_no=cli_no;
    else
        dbms_output.put_line('The balance is below 6000.');
```

```
    end if;
end;
/
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_6.sql
Enter value for client_no: C001
old   2:    cli_no varchar2(6):= '&client_no';
new   2:    cli_no varchar2(6):= 'C001';

PL/SQL procedure successfully completed.


SQL> set linesize 500;
SQL> select client_no, balance_due from client_master;

CLIENT BALANCE_DUE
------ -----------
C008          8000
C009         20000
C001         16000
C002             0
C003          5000
C004             0
C005          3000
C006             0

8 rows selected.
```

ASSIGNMENT-8

1. **a) Create a table whose structure will be as follows:**

Table Name: Prime_Entry

Column Name	Data Type	Attributes
Num_id	Number(3)	Primary Key
Prime_num	Number(3)	Not Null

COMMANDS:

```
set serveroutput on;
create table prime_entry(
    num_id number(3) primary key,
    prime_num number(3) not null
);

create sequence seq
start with 1
increment by 1
/
```

OUTPUT:

```
SQL> @D:\SQL\DBMS\Ass8\asg8_1a.sql

Table created.


Sequence created.
```

b) Write a PL/SQL block of code that will take a number from user and test whether the number is prime or not. If the number is prime, then enter into above table by generating NUMID automatically.

COMMANDS:

```
set serveroutput on;

declare
    num number;
    j number;
    n number;
    i number;
    flag number;
    g number;

begin
```

```
    num:=&n;
    n:=TRUNC(num/2);
    for i in 2..n
    loop
            if(mod(num,i)=0)then
                    flag:=1;
                    exit;
            else
                    flag:=0;
            end if;
    end loop;
    dbms_output.put_line('--------------------');
    if(flag=1)then
            dbms_output.put_line(num||' is not prime');
    else
            select seq.nextval into g from dual;
            insert into prime_entry values(g,num);
    end if;
end;
/
```

OUTPUT:

```
SQL> @D:\SQL\DBMS\Ass8\asg8_1b.sql
Enter value for n: 4
old  10:        num:=&n;
new  10:        num:=4;
--------------------
4 is not prime

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1b.sql
Enter value for n: 5
old  10:        num:=&n;
new  10:        num:=5;
--------------------

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1b.sql
Enter value for n: 7
old  10:        num:=&n;
new  10:        num:=7;
--------------------

PL/SQL procedure successfully completed.

SQL> select * from prime_entry;

    NUM_ID  PRIME_NUM
---------- ----------
         1          5
         2          7
```

c) Now add a checking for same prime number entry. It will show - 'Number already exists in database' for same prime number entry. Write a function to test whether given number exist or not.

<u>COMMANDS</u>:

```
set serveroutput on;

create or replace function prime_test(id number) return number
is
num number(20);
begin
    select num_id into num from prime_entry where prime_num=id;
    return 1;
    exception
            when no_data_found then
            return 0;

end;
/

declare
    num number;
    j number;
    i number;
    n number;
    flag number;
    x number;
begin
    num:=&n;
    n:=TRUNC(num/2);
    for i in 2..n
    loop
            if(mod(num,i)=0)then
                    flag:=1;
                    exit;
            else
                    flag:=0;
            end if;
    end loop;
    dbms_output.put_line('---------------------');
    if(flag=1)then
            dbms_output.put_line(num||' is not prime');
    else
            x:=prime_test(num);
```

```
            if(x=0)then
                insert into prime_entry values(seq.nextval, num);
            else
                dbms_output.put_line('Already exists in the table. ');
            end if;
    end if;
end;
/
```

OUTPUT:

```
SQL> @D:\SQL\DBMS\Ass8\asg8_1c.sql

Function created.

Enter value for n: 4
old   9:        num:=&n;
new   9:        num:=4;
---------------------
4 is not prime

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1c.sql

Function created.

Enter value for n: 7
old   9:        num:=&n;
new   9:        num:=7;
---------------------
Already exists in the table.

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1c.sql

Function created.

Enter value for n: 11
old   9:        num:=&n;
new   9:        num:=11;
---------------------

PL/SQL procedure successfully completed.
```

```
SQL> select * from prime_entry;

    NUM_ID  PRIME_NUM
---------- ----------
         1          5
         2          7
         3         11
```

2. **Create the following table:**

Table Name: Acc_details

Column_Name	Data type	Size	Attributes
Acc_no	Varchar2	8	Primary Key
Name	Varchar2	20	Not Null
Address	Varchar2	20	Not Null
DOB	Date		Not Null
Sex	Char	1	Not Null, Values ('M', 'F')
Contact_no	Number	10	Not Null
Last_trans_date	Date		Not Null
Total_amt	Number	12,4	Not Null
Acc_status	Char	1	Not Null, Values ('A', 'I')

Table Name: Transactions_Acc

Column_Name	Data type	Size	Attributes
Transaction_id	Number	8	Primary Key
Acc_no	Number	8	References Acc_details.Acc_no
Deposit_amt	Number	12,4	
Withdraw_amt	Number	12,4	
Mode_trans	Char	5	Not Null
Cheque_no	Number	6	Default 0
Trans_date	Date		Not Null

When a specific account will be deleted then all the transaction details from Transactions_Acc will be deleted for that account number.

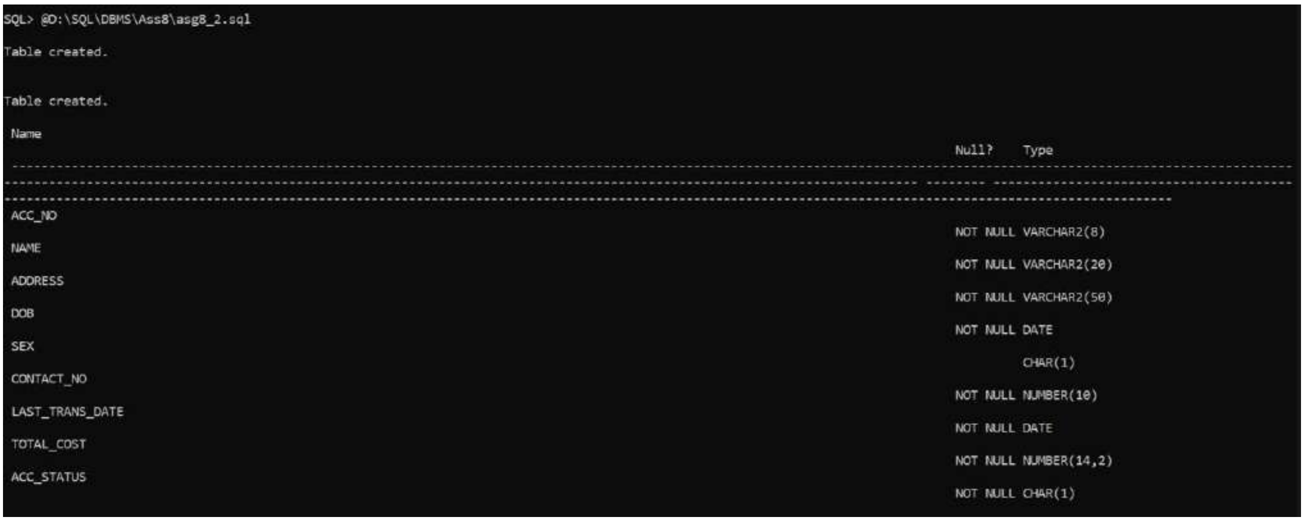
<u>COMMANDS:</u>

```
create table Acc_details(
    Acc_No varchar2(8) primary key,
    Name varchar2(20) not null,
    Address varchar2(50) not null,
    DOB date not null,
    sex char(1) check (sex in ('M', 'F')),
    contact_no number(10) not null,
    last_trans_date date not null,
    Total_cost number(14,2) not null,
    Acc_status char(1) not null check(Acc_status in ('A', 'I'))
);
```

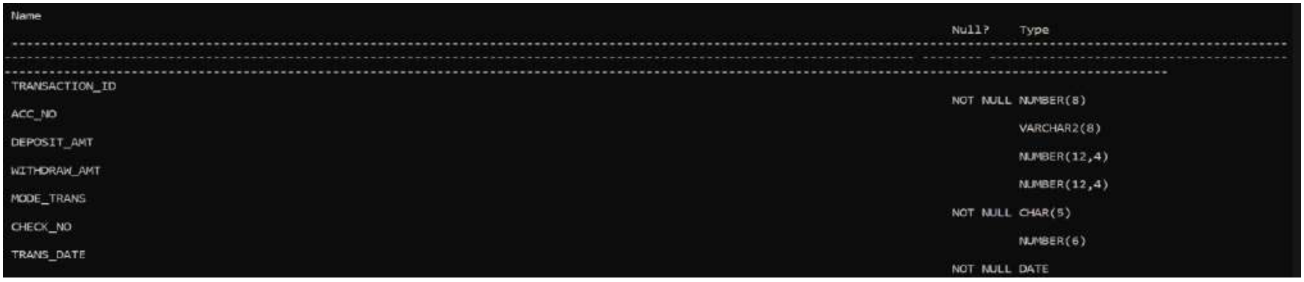
```
create table Transaction_Acc(
    Transaction_Id number(8) primary key,
    Acc_No varchar2(8) references Acc_details on DELETE CASCADE,
    Deposit_amt number(12,4),
```

```
    Withdraw_amt number(12,4),
    Mode_trans char(5) not null,
    Check_no number(6) default 0,
    Trans_date date not null
);

desc Acc_details;
desc Transaction_acc;

insert into Acc_details values('001', 'AMIT', 'BK-256', '12-JAN-2012', 'M', 9836773258,
'13-JUN-2012', 12000, 'A');

select * from Acc_details;

insert into Transaction_Acc values(002, '001', 11000, 5000, 'A', 101, '12-JUN-2012');
insert into Transaction_Acc values(003, '001', 12000, 6000, 'B', 102, '13-JUL-2012');

select * from Transaction_Acc;



delete from Acc_details where Acc_no='001';

select * from Acc_details;

select * from Transaction_Acc;
```

OUTPUT:



```
SQL> @D:\SQL\DBMS\Ass8\asg8_2.sql

Table created.


Table created.

 Name
                                                                    Null?    Type
 ------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------- -------- ---------------------
-------------------------------------------------------------------------------------
 ACC_NO
                                                                    NOT NULL VARCHAR2(8)
 NAME
                                                                    NOT NULL VARCHAR2(20)
 ADDRESS
                                                                    NOT NULL VARCHAR2(50)
 DOB
                                                                    NOT NULL DATE
 SEX
                                                                             CHAR(1)
 CONTACT_NO
                                                                    NOT NULL NUMBER(10)
 LAST_TRANS_DATE
                                                                    NOT NULL DATE
 TOTAL_COST
                                                                    NOT NULL NUMBER(14,2)
 ACC_STATUS
                                                                    NOT NULL CHAR(1)
```



```
 Name
                                                                    Null?    Type
 ------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------- -------- ---------------------
-------------------------------------------------------------------------------------
 TRANSACTION_ID
                                                                    NOT NULL NUMBER(8)
 ACC_NO
                                                                             VARCHAR2(8)
 DEPOSIT_AMT
                                                                             NUMBER(12,4)
 WITHDRAW_AMT
                                                                             NUMBER(12,4)
 MODE_TRANS
                                                                    NOT NULL CHAR(5)
 CHECK_NO
                                                                             NUMBER(6)
 TRANS_DATE
                                                                    NOT NULL DATE
```

```
1 row created.

ACC_NO   NAME                 ADDRESS                                            DOB       S CONTACT_NO LAST_TRAN TOTAL_COST A
-------- -------------------- -------------------------------------------------- --------- - ---------- --------- ---------- -
001      AMIT                 BK-256                                             12-JAN-12 M 9836773258 13-JUN-12      12000 A

1 row created.

1 row created.

TRANSACTION_ID ACC_NO   DEPOSIT_AMT WITHDRAW_AMT MODE_   CHECK_NO TRANS_DAT
-------------- -------- ----------- ------------ ----- ---------- ---------
             2 001           11000         5000 A            101 12-JUN-12
             3 001           12000         6000 B            102 12-JUL-12
```

```
SQL> delete from Acc_details where Acc_no='001';

1 row deleted.
```

```
SQL> select * from Acc_details;

no rows selected
```

```
SQL> select * from Transaction_Acc;

no rows selected
```

ASSIGNMENT-9

1. **Write a PL/SQL block of code that first withdraws an amount of Rs. 500. Then again withdraws Rs. 500. Now if the current balance of a specific account number is less than Rs. 1000 then undo the last withdraw just made.**

COMMANDS:

```
create table Acc_details
(
  Acc_No varchar2(8) primary key,
  Name varchar2(20) not null,
  Address varchar2(50) not null,
  DOB date not null,
  sex char(1) check (sex in ('M', 'F')),
  contact_no number(10) not null,
  last_trans_date date not null,
  Total_amt number(14,2) not null,
  Acc_status char(1) not null check(Acc_status in ('A', 'I'))
);

insert into Acc_details values('001', 'AMIT', 'BK-256', '12-JAN-2012', 'M', 9836773258,
'13-JUN-2012', 12000, 'A');
insert into Acc_details values('002', 'SUMIT', 'AB-125', '10-FEB-2012', 'M', 9830073258,
'13-JAN-2012', 1500, 'A');
insert into Acc_details values('003', 'RAMIT', 'BG-350', '25-JAN-2013', 'M', 9877363258,
'15-JUL-2012', 10000, 'A');

set serveroutput on
declare
  n number(20);
  t number(20);
  amt number:=500;
begin
  n:=&n;
  update Acc_details set Total_amt=Total_amt-amt where
  Acc_No=n;
  commit;
  savepoint s;
  update Acc_details set Total_amt=Total_amt-amt where
  Acc_No=n;
  select Total_amt into t from Acc_details where Acc_No=n;
  if(t < 1000) then
    dbms_output.put_line('Balance after 2nd Transaction = ' || t);
    dbms_output.put_line('!!!!! Insufficient Balance !!!! ');
```

```
    rollback to savepoint s;
    dbms_output.put_line('Balance after Rollback = ' || t);
  else
    commit;
    select Total_Amt into t from Acc_details where Acc_No=n;
    dbms_output.put_line('Balance after COMMIT = ' || t);
  end if;
end;
/
```

```
desc Acc_details;

select * from Acc_details;
```

OUTPUTS:

```
SQL> @D:\SQL\DBMS\Ass9\asg9_1.sql

Table created.


1 row created.


1 row created.


1 row created.

Enter value for n: 001
old   6:   n:=&n;
new   6:   n:=001;
Balance after COMMIT = 11000

PL/SQL procedure successfully completed.
```

```
SQL> desc Acc_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ACC_NO                                    NOT NULL VARCHAR2(8)
 NAME                                      NOT NULL VARCHAR2(20)
 ADDRESS                                   NOT NULL VARCHAR2(50)
 DOB                                       NOT NULL DATE
 SEX                                                CHAR(1)
 CONTACT_NO                                NOT NULL NUMBER(10)
 LAST_TRANS_DATE                           NOT NULL DATE
 TOTAL_AMT                                 NOT NULL NUMBER(14,2)
 ACC_STATUS                                NOT NULL CHAR(1)
```

```
SQL> select * from Acc_details;

ACC_NO   NAME                ADDRESS                                            DOB       S CONTACT_NO LAST_TRAN  TOTAL_AMT A
-------- ------------------- -------------------------------------------------- --------- - ---------- --------- ---------- -
001      AMIT                BK-256                                             12-JAN-12 M 9836773258 13-JUN-12      11000 A
002      SUMIT               AB-125                                             10-FEB-12 M 9830073258 13-JAN-12       1500 A
003      RAMIT               BG-350                                             25-JAN-13 M 9877363258 15-JUL-12      10000 A
```

2. **Write a PL/SQL block of code to update the location of specific department number that will be taken from user. Display an appropriate message using SQL%FOUND based on existence of the record in the Department table and display an appropriate message using SQL%NOTFOUND based on the non-existence of the record in Department Table.**

COMMANDS:

```
select * from dept;
set serveroutput on
declare
  dno number:=&dno;
  loc1 varchar2(10):='&loc';
begin
  update Dept set loc=loc1 where Deptno=dno;
  if sql%found then
    dbms_output.put_line(' The updated loc is ' || loc1);
  end if;
  if sql%notfound then
    dbms_output.put_line(' The updated loc is not found. ');
  end if;
end;
/
```

OUTPUTS:

```
SQL> @D:\SQL\DBMS\Ass9\asg9_2.sql

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

Enter value for dno: 20
old   2:   dno number:=&dno;
new   2:   dno number:=20;
Enter value for loc: MUMBAI
old   3:   loc1 varchar2(10):='&loc';
new   3:   loc1 varchar2(10):='MUMBAI';
The updated loc is MUMBAI

PL/SQL procedure successfully completed.

SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       MUMBAI
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

3. **Write a PL/SQL block that will show an Employee name for a given Employee number. Here you try to enter a wrong Employee number and show an appropriate message, i.e. NOT FOUND using exception handling.**

COMMANDS:

```
set serveroutput on
declare
  ename varchar2(20);
  Eno number:=&Eno;
begin
  select ename into ename from Emp where Empno=Eno;
  dbms_output.put_line(' The Employee name is ' || ename);
  exception
  when NO_DATA_FOUND then
  dbms_output.put_line(' The Employee is not found for the given Emp No. ');
end;
/

select * from emp;
```

OUTPUTS:

```
SQL> @D:\SQL\DBMS\Ass9\asg9_3.sql
Enter value for eno: 7934
old   3:   Eno number:=&Eno;
new   3:   Eno number:=7934;
The Employee name is MILLER

PL/SQL procedure successfully completed.


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7876 ADAMS      CLERK           7788 13-JUL-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

4. **Write a PL/SQL block of code using your own exception handling that will show an error message whenever you want to insert a null value in a not null column.**

COMMANDS:

```
set serveroutput on
declare
  IN_ERR exception;
  Pragma
  exception_init(IN_ERR, -01400);
begin
  insert into Emp values (null,'BLAKE','MANAGER',7839,to_date('1-5-1981','dd-mm-
yyyy'),2850, null, 30);
    exception
    when IN_ERR then
      dbms_output.put_line(' Cannot insert Null values in not Null column. ');
end;
/
```

OUTPUTS:

```
SQL> @D:\SQL\DBMS\Ass9\asg9_4.sql
Cannot insert Null values in not Null column.

PL/SQL procedure successfully completed.
```

5. **a) Create a table Emp_sal_inc that have three column(Emp_id, Cur_sal, Inc_date).**
b) Now write a PL/SQL block of code will allow 2% salary increment of all employee of RESEARCH department. After that all records are to be inserted into the above table (i.e., Emp_sal_inc)

COMMANDS:

```
set serveroutput on
create table Emp_sal_inc(
  Emp_id number(10),
  cur_sal number(20,4),
  inc_date date
);
declare
  cursor cur is
    select Empno, Sal from Emp where Deptno=(Select Deptno from Dept where
Dname='RESEARCH');
  Emp_id number;
  Emp_sal Emp.Sal%type;
begin
  open cur;
  if cur%isopen then
    loop
```

```
      fetch cur into Emp_id, Emp_sal;
      exit when cur%notfound;
      update Emp set Sal=Sal*1.02 where Empno=Emp_id;
      select Sal into Emp_sal from Emp where Empno=Emp_id;
      insert into Emp_sal_inc values(Emp_id, Emp_sal, SYSDATE);
    end loop;
    commit;
    dbms_output.put_line(cur%rowcount);
  else
    dbms_output.put_line(' Cursor not open....');
  end if;
  close cur;
end;
/
select * from Dept;
select * from Emp;
select * from Emp_sal_inc;
```

OUTPUTS:

```
SQL> @D:\SQL\DBMS\Ass9\asg9_5.sql

Table created.

5

PL/SQL procedure successfully completed.

SQL> select * from Dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       MUMBAI
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

SQL> select * from Emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81     3034.5                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3060                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3060                    20
      7369 SMITH      CLERK           7902 17-DEC-80        816                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7876 ADAMS      CLERK           7788 13-JUL-87       1122                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

```
SQL> select * from Emp_sal_inc;

    EMP_ID    CUR_SAL INC_DATE
---------- ---------- ---------
      7566     3034.5 16-MAY-23
      7788       3060 16-MAY-23
      7902       3060 16-MAY-23
      7369        816 16-MAY-23
      7876       1122 16-MAY-23
```

ASSIGNMENT-10

1. **Write a PL/SQL block that will add 2% interest of all customer of a bank for active account.**
   i) For updating Acc_details updating, you have to use Cursor.
   ii) For entry in Transaction_Acc, you have to use procedure.
   iii) For Generation Transaction_id, you have to use function.

PL-SQL CODE:

i)

```
set serveroutput on
declare
   cursor add_interest
   is
   select Acc_no,Total_cost from Acc_details where Acc_status='A';
   varaccn Acc_details.Acc_no%type;
   varamt Acc_details.Total_cost%type;
      begin
      open add_interest;
      if add_interest % isopen then
         loop
         fetch add_interest into varaccn,varamt;
         exit when add_interest%notfound;
         update Acc_details set Total_cost=varamt*1.02 where Acc_no=varaccn;
         dbms_output.put_line(varaccn || ' is updated');
         end loop;
      else
         dbms_output.put_line('Curson not opened.');
      end if;
      close add_interest;
      commit;
      end;
/
```

OUTPUT:

```
SQL> @"C:\Users\monis\Desktop\ass10-1.sql"
001 is updated

PL/SQL procedure successfully completed.
```

ii) and iii)

PL-SQL CODE:

```
set serveroutput on
create function Max_id return number
is
    var_id number(4);
    begin
```

```
    select max(Transaction_id) into var_id from Transaction_acc;
            if var_id is null then
            var_id:=200;
            else
            var_id:=var_id+1;
            end if;
            return var_id;
            exception
            when no_data_found then
            return var_id;
    end;
/

create procedure Transaction_entry(varaccn in Acc_details.Acc_no%type, varamt in
Acc_details.Total_cost%type)
    is
    vartid Transaction_acc.Transaction_id%type;
    begin
    vartid:=Max_id();
   insert into Transaction_acc values(vartid, varaccn,varamt, 0, 'CHQ',0,Sysdate);
    dbms_output.put_line('  Data inserted with Id   ' ||vartid);
    end;
/
declare
            cursor add_interest
            is
select Acc_no, Total_cost from Acc_details where Acc_status='A';
            varaccn Acc_details.Acc_no%type;
            varamt Acc_details.Total_cost%type;
            begin
            open add_interest;
            if add_interest%isopen then
                    loop
                    fetch add_interest into varaccn, varamt;
                    exit when add_interest%notfound;
update Acc_details set Total_cost=varamt*1.02 where Acc_no=varaccn;
            dbms_output.put_line( varaccn || '   is updated  ');
            varamt:=varamt*1.02;
            Transaction_entry(varaccn, varamt);
            end loop;
            else
                    dbms_output.put_line('Cursor not opened.  ');
            end if;
            close add_interest;
```

```
        commit;
        end;
/
```

OUTPUT:

```
SQL> @"C:\Users\monis\Desktop\ass10-2.sql"

Function created.


Procedure created.

001   is updated
Data inserted with Id   4

PL/SQL procedure successfully completed.

SQL> select * from Transaction_Acc;

TRANSACTION_ID ACC_NO   DEPOSIT_AMT WITHDRAW_AMT MODE_   CHECK_NO TRANS_DAT
-------------- -------- ----------- ------------ ----- ---------- ---------
             2 001            11000         5000 A            101 12-JUN-12
             3 001            12000         6000 B            102 12-JUL-12
             4 001          12484.8            0 CHQ            0 30-MAY-23
```

2. **a) Create the following table: (Table Name:– Emp_audit)**

Column_Name	Data type	Size	Attributes
Emp_no	Number	4	Primary Key
Dept_no	Number	4	Not Null, Ref. department.dept_no
Status	Varchar 2	8	
Salary	Number	8,2	Not Null
Audit_date	Date		Not Null

b)Write a trigger that must keep track of records (in above table) that are being deleted or updated from Employee table.

c) Write a SQL command to update the employee entry and describe the output.

COMMANDS:

a, b & c)

PL-SQL CODE:

```
set serveroutput on
create table Emp_audit
(Emp_no number(4) primary key,
 Dept_no number(4) not null references Dept,
 Status varchar2(8),
 Salary number(8,2) not null,
 Audit_date date not null);
```

```
set serveroutput on
drop trigger trg_sal;
```

```
create trigger trg_sal after
update or delete on Emp for each row
  declare
            status varchar2(20);
            begin
                    if updating then
                    status:='UPDATE';
                    end if;
                    if deleting then
                    status:='DELETE';
                    end if;
          insert into Emp_audit values(:Old.empno, :Old.deptno,status, :Old.Sal,SYSDATE);

            end;
/
```

OUTPUT:

```
SQL> @"C:\Users\monis\Desktop\ass11.sql"

Table created.


Trigger dropped.


Trigger created.
```

COMMAND:

SQL> update Emp set Sal=2050 where Empno=7499;

OUTPUT:

```
SQL> update Emp set Sal=2050 where Empno=7499;

1 row updated.
```

COMMAND:

SQL> select * from Emp_audit;

OUTPUT:

```
SQL> select * from Emp_audit;

    EMP_NO    DEPT_NO STATUS       SALARY AUDIT_DAT
---------- ---------- -------- ---------- ---------
      7499         30 UPDATE         1600 30-MAY-23
```