MCKV Institute of Engineering

243 G. T. Road (N), Liluah, Howrah – 711204

Subject: **Database Management System Lab** Code:**PC-CS692**
Stream: CSE-1 Credit: 2

Assignment–1

Create the following tables using the schema given below and insert given data set accordingly.

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

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

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

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

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

ASSIGNMENT-1

Create the following tables using the schema given below and insert given data set accordingly.
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	

QUERY:

```
CREATE TABLE CLIENT_MASTER( CLIENT_NO VARCHAR2(6) primary key check(CLIENT_NO like 'C%'),
NAME VARCHAR2(20) NOT NULL,
ADDRESS1 VARCHAR2(20),
ADDRESS2 VARCHAR2(20),
CITY VARCHAR2(15),
STATE VARCHAR2(15),
PINCODE NUMBER(6),
BALANCE_DUE NUMBER(10,2)
);
```

OUTPUT:

TABLE CREATED

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

QUERY:

```
INSERT INTO CLIENT_MASTER(CLIENT_NO,NAME,ADDRESS1,ADDRESS2,CITY,STATE,PINCODE,BALANCE_DUE)
VALUES('C001','Ivan Bayross','P-76','Worli','Bombay','Maharastra',400054,15000);

INSERT INTO CLIENT_MASTER VALUES ('C002','VandanaSatiwal','128','Adams Street','Madras','TamilNadu',780001,0);

INSERT INTO CLIENT_MASTER VALUES ('C003','PramadaJaguste','157','Gopalpur','Kolkata','West Bengal',700058,5000);

INSERT INTO CLIENT_MASTER VALUES ('C004','BasuNavindgi','A/12','Nariman','Bombay','Maharastra',400056,0);

INSERT INTO CLIENT_MASTER VALUES ('C005','Ravi Shreedharan','B/34','Rajnagar','Delhi','Delhi',100001,2000);

INSERT INTO CLIENT_MASTER VALUES ('C006','Rukmini','Q-12','Bandra','Bombay','Maharastra',400050,0);

SELECT * FROM CLIENT_MASTER;
```

OUTPUT:

CLIENT	NAME	ADDRESS1	ADDRESS2	CITY	STATE	PINCODE	BALANCE_DUE
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 Shreedharan	B/34	Rajnagar	Delhi	Delhi	100001	2000
C006	Rukmini	Q-12	Bandra	Bombay	Maharastra	400050	0

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

QUERY:

```
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)
);
```

OUTPUT:

Table Created.

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

QUERY:

```
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);

SELECT * FROM PRODUCT_MASTER;
```

OUTPUT:

```
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.
```

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

QUERY:

```
CREATE TABLE SALESMAN_MASTER(
SALESMAN_NO VARCHAR2(6) primary key check(SALESMAN_NO like 'S%'),
SALESMAN_NAME VARCHAR2(30) NOT NULL,
ADDRESS1 VARCHAR2(30),
ADDRESS2 VARCHAR2(30),
CITY VARCHAR2(20),
PINCODE NUMBER(8),
STATE VARCHAR2(20),
SAL_AMT NUMBER(8,2)
);
```

OUTPUT: Table Created.

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

QUERY:

```
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);

SELECT * FROM SALESMAN_MASTER;
```

OUTPUT:

```
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
```

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')

QUERY:

```
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) check(ORDER_STATUS in('InProcess', 'Fullfilled', 'BackOrder','Cancelled')),
constraint C1 check(DELIVERY_DATE>ORDER_DATE)
);
```

OUTPUT: Table Created.

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

QUERY:

```
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');

SELECT * FROM SALES_ORDER;
```

OUTPUT:

```
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
```

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	

QUERY:

```
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)
);
```

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

QUERY:

```
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);

SELECT * FROM SALES_ORDER_DETALS;
```

OUTPUT:

```
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
```

Assignment 2

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

QUERY and OUTPUT

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

NAME
--------------------
VandanaSatiwal
BasuNavindgi
Ravi Shreedharan
```

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

QUERY and OUTPUT

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

NAME                 CITY
-------------------- ---------------
VandanaSatiwal       Madras
PramadaJaguste       Kolkata
Ravi Shreedharan     Delhi
```

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

QUERY and 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’.

QUERY and OUTPUT

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

NAME                 CITY
-------------------- ---------------
Ivan Bayross         Bombay
BasuNavindgi         Bombay
Ravi Shreedharan     Delhi
Rukmini              Bombay
```

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

QUERY and OUTPUT:

```
SQL> select name,city from client_master where BALANCE_DUE>10000;

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

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

QUERY and 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’.

QUERY and 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.

QUERY and 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.

QUERY and OUTPUT:

```
SQL> select description from product_master where sell_price>1500 ;

DESCRIPTION
----------------------------------------
Monitors
Keyboard
CD Drive
540 HDD
```

```
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’.

QUERY and OUTPUT:

```
SQL> select name,city,state from client_master where state!='Maharastra';

NAME                 CITY            STATE
-------------------- --------------- ---------------
VandanaSatiwal       Madras          TamilNadu
PramadaJaguste       Kolkata         West Bengal
Ravi Shreedharan     Delhi           Delhi
```

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

QUERY and 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

QUERY and 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
```

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

QUERY and OUTPUT:

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

NEW_DATE
---------
22-FEB-25
```

ASSIGNMENT 3

1. Count the total number of orders.

QUERY and OUTPUT:

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

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

2. Calculate the average price of all the products.

QUERY and OUTPUT:

```
SQL> select avg(product_rate) FROM sales_order_details;

AVG(PRODUCT_RATE)
-----------------
       3482.14286
```

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

QUERY and OUTPUT:

```
SQL> select count(product_no) from product_master where sell_price>=1500;

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

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

QUERY and OUTPUT:

```
SQL> select  max(sell_price) max_price, min(sell_price) min_price from product_master;

 MAX_PRICE  MIN_PRICE
---------- ----------
     12000        525
```

5. Change the City of the Client_no ‘C005’ to ‘Madras’.

OUTPUT and QUERY:

```
SQL> Update client_master set city='Madras' where client_no='C005';

1 row updated.

SQL> select city from client_master where client_no='C005';

CITY
---------------
Madras
```

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

OUTPUT and QUERY:

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

1 row updated.

SQL> select balance_due from client_master where client_no='C005';

BALANCE_DUE
-----------
       3000
```

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

OUTPUT and QUERY:

```
SQL> delete from client_master where state='TamilNadu';
delete from client_master where state='TamilNadu'
*
ERROR at line 1:
ORA-02292: integrity constraint (CSE154.SYS_C00120114) violated - child record found
```

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

QUERY and OUTPUT:

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

Table altered.

SQL> desc client_master;
 Name
                                                                    Null?    Type
 -----------------------------------------------------------------------------------
----------------------------------------------------------------------- -------- ----------
-------------------------------------------------------------------------------------
----------------------------
 CLIENT_NO
                                                                    NOT NULL VARCH
AR2(6)
 NAME
                                                                    NOT NULL VARCH
AR2(20)
 ADDRESS1
                                                                             VARCH
AR2(20)
 ADDRESS2
                                                                             VARCH
AR2(20)
 CITY
                                                                             VARCH
AR2(15)
 STATE
                                                                             VARCH
AR2(15)
 PINCODE
                                                                             NUMBE
R(6)
 BALANCE_DUE
                                                                             NUMBE
R(10,2)
 TELEPHONE
                                                                             NUMBE
R(10)
```

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

QUERY and OUTPUT:

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

Table altered.

SQL> desc client_master;
 Name
                                                                    Null?    Type
 -----------------------------------------------------------------------------------
----------------------------------------------------------------------- -------- ----------
-------------------------------------------------------------------------------------
----------------------------
 CLIENT_NO
                                                                    NOT NULL VARCH
AR2(6)
 NAME
                                                                    NOT NULL VARCH
AR2(20)
 ADDRESS1
                                                                             VARCH
AR2(20)
 ADDRESS2
                                                                             VARCH
AR2(20)
 CITY
                                                                             VARCH
AR2(15)
 STATE
                                                                             VARCH
AR2(15)
 PINCODE
                                                                             NUMBE
R(10)
 BALANCE_DUE
                                                                             NUMBE
R(10,2)
 TELEPHONE
                                                                             NUMBE
R(10)
```

10. Drop the column Address2 from the table client_master.

QUERY and 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(20)
 ADDRESS1                                                               VARCHAR2(20)
 CITY                                                                   VARCHAR2(15)
 STATE                                                                  VARCHAR2(15)
 PINCODE                                                                NUMBER(10)
 BALANCE_DUE                                                            NUMBER(10,2)
 TELEPHONE                                                              NUMBER(10)
```

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

QUERY and 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(20)
 ADDRESS1                                                               VARCHAR2(20)
 CITY                                                                   VARCHAR2(15)
 STATE                                                                  VARCHAR2(15)
 PINCODE                                                                NUMBER(10)
 BALANCE_DUE                                                            NUMBER(10,2)
 TELEPHONE                                                              NUMBER(10)
```

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

QUERY and OUTPUT

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

6 rows created.
```

```
SQL> select * from client_master_duplicate;

CLIENT NAME                ADDRESS1           CITY           STATE              PINCODE BALANCE_DUE  TELEPHONE
------ ------------------- ------------------ -------------- -------------- ---------- ----------- ----------
C001   Ivan Bayross        P-76               Bombay         Maharastra         400054       15000
C002   VandanaSatiwal      128                Madras         TamilNadu          780001           0
C003   PramadaJaguste      157                Kolkata        West Bengal        700058        5000
C004   BasuNavindgi        A/12               Bombay         Maharastra         400056           0
C005   Ravi Shreedharan    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.

QUERY and OUTPUT;

```
SQL> rename client_master_duplicate to c_master;

Table renamed.

SQL> desc c_master;
 Name                                                                                                Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------------------
 CLIENT_NO                                                                                                    VARCHAR2(6)
 NAME                                                                                                NOT NULL VARCHAR2(20)
 ADDRESS1                                                                                                     VARCHAR2(20)
 CITY                                                                                                         VARCHAR2(15)
 STATE                                                                                                        VARCHAR2(15)
 PINCODE                                                                                                      NUMBER(10)
 BALANCE_DUE                                                                                                  NUMBER(10,2)
 TELEPHONE                                                                                                    NUMBER(10)

SQL> desc client_master_duplicate;
ERROR:
ORA-04043: object client_master_duplicate does not exist
```

14. Destroy the table c_master with its data.

QUERY and OUTPUT;

```
SQL> DROP TABLE C_MASTER;

Table dropped.
```

```
SQL> desc c_master;
ERROR:
ORA-04043: object c_master does not exist
```

ASSIGNMENT 4

1.Display the names of all employees, right-aligning them to 15 characters

QUERY and OUTPUT:

```
SQL> select lpad(ename,15) from emp;

LPAD(ENAME,15)
---------------------------------------------------------------
           KING
          BLAKE
          CLARK
          JONES
          SCOTT
           FORD
          SMITH
          ALLEN
           WARD
         MARTIN
         TURNER

LPAD(ENAME,15)
---------------------------------------------------------------
          ADAMS
          JAMES
         MILLER

14 rows selected.
```

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

QUERY and OUTPUT:

```
SQL> select lpad(ename,15,'*') from emp;

LPAD(ENAME,15,'*')
---------------------------------------------------------------
**********KING
**********BLAKE
**********CLARK
**********JONES
**********SCOTT
**********FORD
**********SMITH
**********ALLEN
**********WARD
*********MARTIN
*********TURNER

LPAD(ENAME,15,'*')
---------------------------------------------------------------
**********ADAMS
**********JAMES
*********MILLER

14 rows selected.
```

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

QUERY and OUTPUT:

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

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      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
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7876 ADAMS      CLERK           7788 13-JUL-87       1100                    20
```

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

QUERY and OUTPUT:

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

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:

QUERY and OUTPUT:

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

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:

QUERY and OUTPUT:

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

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

7. List the names of all employees who are either ‘CLERKS’, ‘SALESMAN’, or ‘ANALYST’:

QUERY and OUTPUT:

```
SQL> select ename,job from emp where job in('CLERKS','SALESMAN','ANALYST');

ENAME      JOB
---------- ---------
SCOTT      ANALYST
FORD       ANALYST
ALLEN      SALESMAN
WARD       SALESMAN
MARTIN     SALESMAN
TURNER     SALESMAN

6 rows selected.
```

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

QUERY and OUTPUT:

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

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

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

QUERY and OUTPUT:

```
SQL> select count(ename), avg(sal) from emp where deptno=30;

COUNT(ENAME)   AVG(SAL)
------------ ----------
           6 1566.66667
```

10. Calculate the minimum salary earned by 'CLERKS':

QUERY and OUTPUT:

```
SQL> SELECT MIN(sal) AS min_salary FROM emp WHERE job = 'CLERK';

MIN_SALARY
----------
       800
```

11. Calculate the maximum salary earned by 'SALESMAN':

QUERY and OUTPUT:

```
SQL> SELECT MAX(sal) AS max_salary FROM emp WHERE job = 'SALESMAN';

MAX_SALARY
----------
      1600
```

12. Find the names of those employees whose immediate boss is in a different department:
QUERY and OUTPUT:

```
SQL> SELECT e.ename
  2  FROM emp e, emp m
  3  WHERE e.mgr = m.empno
  4  AND e.deptno != m.deptno;

ENAME
----------
JONES
BLAKE
```

13. Calculate the number of employees who are not getting any commission:

QUERY and OUTPUT:

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

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

14. Find the department(s) which do not have any employee:

QUERY and OUTPUT:

```
SQL> SELECT deptno FROM dept WHERE deptno NOT IN (SELECT  deptno FROM emp);

    DEPTNO
----------
        40
```

ASSIGNMENT 5

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

QUERY AND 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

QUERY AND 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 13-JUL-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.

QUERY AND OUTPUT :

```
SQL> select ename,job,dname,loc from emp e,dept d where loc='DALLAS';

ENAME      JOB       DNAME          LOC
---------- --------- -------------- -------------
KING       PRESIDENT RESEARCH       DALLAS
BLAKE      MANAGER   RESEARCH       DALLAS
CLARK      MANAGER   RESEARCH       DALLAS
JONES      MANAGER   RESEARCH       DALLAS
SCOTT      ANALYST   RESEARCH       DALLAS
FORD       ANALYST   RESEARCH       DALLAS
SMITH      CLERK     RESEARCH       DALLAS
ALLEN      SALESMAN  RESEARCH       DALLAS
WARD       SALESMAN  RESEARCH       DALLAS
MARTIN     SALESMAN  RESEARCH       DALLAS
TURNER     SALESMAN  RESEARCH       DALLAS

ENAME      JOB       DNAME          LOC
---------- --------- -------------- -------------
ADAMS      CLERK     RESEARCH       DALLAS
JAMES      CLERK     RESEARCH       DALLAS
MILLER     CLERK     RESEARCH       DALLAS

14 rows selected.
```

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.

QUERY and OUTPUT :

```
SQL> select ename,sal,sal * 0.10 PF, sal * 0.6 HRA ,sal * 0.4 DA,(sal + sal * 0.10 + sal * 0.6 +sal * 0.4) Gross from emp order by gross;

ENAME             SAL         PF        HRA         DA      GROSS
---------- ---------- ---------- ---------- ---------- ----------
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
---------- ---------- ---------- ---------- ---------- ----------
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.

QUERY and OUTPUT :

```
SQL> SELECT e.ename, e.sal FROM emp e, dept d WHERE e.deptno = d.deptno
  2  AND d.loc = 'DALLAS' AND e.sal > (SELECT MAX(e1.sal) FROM emp e1, dept d1
  3  WHERE e1.deptno = d1.deptno AND d1.loc = 'CHICAGO');

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

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

QUERY and OUTPUT :

```
SQL> select ename,job,dname,loc from emp e,dept d where loc='DALLAS';

ENAME      JOB       DNAME          LOC
---------- --------- -------------- -------------
KING       PRESIDENT RESEARCH       DALLAS
BLAKE      MANAGER   RESEARCH       DALLAS
CLARK      MANAGER   RESEARCH       DALLAS
JONES      MANAGER   RESEARCH       DALLAS
SCOTT      ANALYST   RESEARCH       DALLAS
FORD       ANALYST   RESEARCH       DALLAS
SMITH      CLERK     RESEARCH       DALLAS
ALLEN      SALESMAN  RESEARCH       DALLAS
WARD       SALESMAN  RESEARCH       DALLAS
MARTIN     SALESMAN  RESEARCH       DALLAS
TURNER     SALESMAN  RESEARCH       DALLAS

ENAME      JOB       DNAME          LOC
---------- --------- -------------- -------------
ADAMS      CLERK     RESEARCH       DALLAS
JAMES      CLERK     RESEARCH       DALLAS
MILLER     CLERK     RESEARCH       DALLAS
```

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

QUERY and 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.

QUERY and 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.

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

QUERY and 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/-.

QUERY and 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.

QUERY and 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

1. **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	

Create a view vw_client_master using Client_no, Name, Address1 and Bal_due

```
SQL> create view vw_client_master as select client_no,Name,Address1,Balance_due from client_master;

View created.
```

a. Insert at least 3 records to vw_client_master.

```
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(client_no,name,address1,balance_due)values('C007','ABHISEK SINGH','BALLY',15000);

1 row created.

SQL> Insert into vw_client_master(client_no,name,address1,balance_due)values('C008','DIPAK DAS','SALTLAKE',10000);

1 row created.
```

```
SQL> select * from vw_client_master;

CLIENT NAME                 ADDRESS1             BALANCE_DUE
------ -------------------- -------------------- -----------
C001   Ivan Bayross         P-76                       15000
C002   VandanaSatiwal       128                            0
C003   PramadaJaguste       157                         5000
C004   BasuNavindgi         A/12                           0
C005   Ravi Shreedharan     B/34                        3000
C006   Rukmini              Q-12                           0
C007   ABHISEK SINGH        BALLY                      15000
C008   DIPAK DAS            SALTLAKE                   10000
C009   RAMAN GUPTA          HOWRAH                     20000

9 rows selected.
```

b. Update a record to vw_client_master.

```
SQL> Update vw_client_master set Balance_due=8000 where client_no='C008';

1 row updated.

SQL> select * from vw_client_master;

CLIENT NAME                ADDRESS1             BALANCE_DUE
------ ------------------- -------------------- -----------
C001   Ivan Bayross        P-76                       15000
C002   VandanaSatiwal      128                            0
C003   PramadaJaguste      157                         5000
C004   BasuNavindgi        A/12                           0
C005   Ravi Shreedharan    B/34                        3000
C006   Rukmini             Q-12                           0
C007   ABHISEK SINGH       BALLY                      15000
C008   DIPAK DAS           SALTLAKE                    8000
C009   RAMAN GUPTA         HOWRAH                     20000
```

c. Delete a record from vw_client_master.

```
SQL> Delete from vw_client_master where client_no = 'C007';

1 row deleted.

SQL> select * from vw_client_master;

CLIENT NAME                ADDRESS1             BALANCE_DUE
------ ------------------- -------------------- -----------
C001   Ivan Bayross        P-76                       15000
C002   VandanaSatiwal      128                            0
C003   PramadaJaguste      157                         5000
C004   BasuNavindgi        A/12                           0
C005   Ravi Shreedharan    B/34                        3000
C006   Rukmini             Q-12                           0
C008   DIPAK DAS           SALTLAKE                    8000
C009   RAMAN GUPTA         HOWRAH                     20000

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).

QUERY: 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';

a. Insert a record to vw_sales_det.

```
SQL> Insert into vw_sales_det(client_no,order_date,product_no,qty_ordered,order_status) values('C007','O19251','12-Jan-91','P00091',100,'InProcess');
Insert into vw_sales_det(client_no,order_date,product_no,qty_ordered,order_status) values('C007','O19251','12-Jan-91','P00091',100,'InProcess')
            *
ERROR at line 1:
ORA-00913: too many values
```

b. Update the client_no for a particular order_no.

```
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
```

c. Delete a record.

```
SQL> Delete from vw_sales_det where client_no='C002';

1 row deleted.
```

d. Remove the views from database.

```
SQL> Drop view vw_sales_det;

View dropped.
```

ASSIGNMENT 7

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

CODE :

```
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> @D:DBMS154\Assg7_1.sql;
Enter value for x: 5
old   6:     n := &x;
new   6:     n := 5;
Factorial of 5is120

PL/SQL procedure successfully completed.
```

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

CODE :

```
set serveroutput on;
declare
n number;
i number;
s number := 0;
begin
   n:= &x;
   for i in 1..n
   loop
   s:=s+i;
   end loop;
   dbms_output.put_line('Sum of first '||n||' number is '||s);
end;
/
```

OUTPUT :

```
SQL> @D:DBMS154\Assg7_2.sql;
Enter value for x: 5
old   6:     n:= &x;
new   6:     n:= 5;
Sum of first 5 number is 15

PL/SQL procedure successfully completed.
```

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

CODE :

```
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> @D:DBMS154\Assg7_3.sql;
Enter value for x: 1999
old   4: y:= &x;
new   4: y:= 1999;
Not a Leap year

PL/SQL procedure successfully completed.

SQL> @D:DBMS154\Assg7_3.sql;
Enter value for x: 2024
old   4: y:= &x;
new   4: y:= 2024;
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).

CODE :

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

OUTPUT :

```
SQL> @D:\DBMS154\Assg7_4.sql;
Enter value for a: 100
old   6: a:=&a;
new   6: a:=100;
Enter value for b: 22
old   7: b:=&b;
new   7: b:=22;
Enter value for c: 3
old   8: c:=&c;
new   8: c:=3;
100 is the largest number
```

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.

CODE :

```
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*r*r;
insert into Areas values(r,area);
end loop;
end;
/
```

OUTPUT :

```
SQL> @D:\DBMS154\Assg7_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.

CODE :

```
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 :

```
PL/SQL procedure successfully completed.

SQL> @D:\DBMS154\Assg7_6.sql;
Enter value for client_no: C003
old   2: cli_no varchar2(6):= '&client_no';
new   2: cli_no varchar2(6):= 'C003';
The balance is below 6000
```

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

CODE :

```
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:\DBMS154\Ass8_1.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.

CODE :

```
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:\DBMS154\Ass8_2.sql;
Enter value for n: 2
old   9: num:=&n;
new   9: num:=2;
---------------------

PL/SQL procedure successfully completed.

SQL> @D:\DBMS154\Ass8_2.sql;
Enter value for n: 6
old   9: num:=&n;
new   9: num:=6;
---------------------
6 is not prime

PL/SQL procedure successfully completed.

SQL> select * from prime_entry;

    NUM_ID  PRIME_NUM
---------- ----------
         1          2
```

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.

CODE :

```
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;
n number;
i 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 exist in Table');
end if;
end if;
end;
/
```

OUTPUT :

```
SQL> @D:\DBMS154\Ass8_3.sql;

Function created.

Enter value for n: 5
old   9: num:=&n;
new   9: num:=5;
---------------------

PL/SQL procedure successfully completed.

SQL> @D:\DBMS154\Ass8_3.sql;

Function created.

Enter value for n: 5
old   9: num:=&n;
new   9: num:=5;
---------------------
Already exist in Table

PL/SQL procedure successfully completed.
```

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.

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_cost 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');


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
);
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');
```

```
SQL> select * from Acc_details;

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
```

```
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 13-JUL-12
```

COMMANDS : delete from Acc_details where Acc_no='001';

```
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.

CODE:

```
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');
```

CODE :

```
set serveroutput on
declare
n number(20);
t number(20);
amt number:=500;
begin
n:=&n;
update Acc_details set Total_cost = Total_cost-amt where Acc_no = n;
commit;
savepoint s;
update Acc_details set Total_cost= Total_cost-amt where Acc_no=n;
select Total_cost 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_cost into t from Acc_details where Acc_no=n;
dbms_output.put_line('Balance after commit'||t);
end if;
end;
/
```

OUTPUT :

```
SQL> select * from Acc_details;

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
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
```

```
SQL> @D:\DBMS154\Ass9_1.sql;
Enter value for n: 001
old   6: n:=&n;
new   6: n:=001;
Balance after commit11000
```

```
SQL> select * from Acc_details;

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      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.

CODE :

```
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;
/
```

OUTPUT :

```
SQL> @D:\DBMS154\Ass9_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.

CODE :

```
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;
/
```

OUTPUT :

```
SQL> @D:\DBMS154\Ass9_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.

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 13-JUL-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
```

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.

CODE :

```
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-mmyyyy'),2850, null, 30);
exception
when IN_ERR then
dbms_output.put_line(' Cannot insert Null values in not Null column. ');
end;
/
```

OUTPUT :

```
SQL> @D:\DBMS154\Ass9_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)

CODE :

```
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;
/
```

OUTPUT :

```
SQL> @D:\DBMS154\Ass9_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 13-JUL-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 04-APR-25
      7788       3060 04-APR-25
      7902       3060 04-APR-25
      7369        816 04-APR-25
      7876       1122 04-APR-25
```

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.

i) **CODE :**

```
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> @D:\DBMS154\Assg10_1.sql;
001 is updated
002 is updated
003 is updated
```

```
SQL> select * from Acc_details;

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      11220 A
002      SUMIT               AB-125                                                 10-FEB-12 M 9830073258 13-JAN-12       1530 A
003      RAMIT               BG-350                                                 25-JAN-13 M 9877363258 15-JUL-12      10200 A
```

ii) **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;
/
```

CODE :

```
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> @D:\DBMS154\Assg10_2.sql;

Function created.


Procedure created.
```

```
001   is updated
Data inserted with Id   7
002   is updated
Data inserted with Id   8
003   is updated
Data inserted with Id   9
```

```
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 13-JUL-12
             4 001         11673.29            0 CHQ            0 11-APR-25
             5 002          1591.81            0 CHQ            0 11-APR-25
             6 003         10612.08            0 CHQ            0 11-APR-25
             7 001         11906.76            0 CHQ            0 11-APR-25
             8 002          1623.65            0 CHQ            0 11-APR-25
             9 003         10824.32            0 CHQ            0 11-APR-25

8 rows selected.
```

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.

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> @D:\DBMS154\Assg10_3.sql;

Table created.

drop trigger trg_sal
*
ERROR at line 1:
ORA-04080: trigger 'TRG_SAL' does not exist



Trigger created.
```

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

1 row updated.

SQL> select * from Emp_audit;

    EMP_NO    DEPT_NO STATUS        SALARY AUDIT_DAT
---------- ---------- -------- ---------- ---------
      7499          30 UPDATE         1600 11-APR-25
```

```
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 13-JUL-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       2050        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.
```

ASSIGNMENT NO. :- 10

PROBLEM STATEMENT : Write a program to implement sliding window protocol.

CODE :- (Client side)

```
#include<stdio.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<unistd.h>
#define CLIENT_IP "127.0.0.1"
#define CLIENT_PORT 6542
#define SERVER_IP "127.0.0.1"
#define SERVER_PORT 6555
void main(){
	struct sockaddr_in client,server;
	int sd,n,i,j,count=0;
	Ichar msg[512],msg1[512],ack[512];
	bzero((char*)&server,sizeof(server));
	server.sin_family = AF_INET;
	server.sin_addr.s_addr = inet_addr(SERVER_IP);
	server.sin_port = htons(SERVER_PORT);
	bzero((char*)&client,sizeof(client));
	client.sin_family = AF_INET;
	client.sin_addr.s_addr=inet_addr(CLIENT_IP);
	client.sin_port=htons(CLIENT_PORT);
	sd =  socket(AF_INET,SOCK_STREAM,0);
	connect(sd,(struct sockaddr*)&server,sizeof(server));
	do{
		printf("Enter a message:");
		scanf("%s",msg);
		printf("Enter Window size:");
		scanf("%d",&n);
		j=0;
		for(i=0;i<strlen(msg);i++){
			if(j<n){
				msg1[j++]=msg[i];
			}
			if(j==n || i==strlen(msg)-1){
				msg1[j]='\0';
				send(sd,msg1,strlen(msg1)+1,0);
				memset(ack,0x0,512);
				recv(sd,ack,512,0);
				printf("%s%d\n",ack,count++);
				j=0;
		}
	    }

	}while(strcmp(msg,"stop"));
	close(sd);

}
```

CODE:- (Server Side)

```
#include<stdio.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<unistd.h>
#define SERVER_IP "127.0.0.1"
#define SERVER_PORT 6555
void main(){
	struct sockaddr_in client,server;
	int sd,nsd,clen=sizeof(client);
	char msg[512],ack[]="Acknowledgement received";
	bzero((char*)&server,sizeof(server));
	server.sin_family = AF_INET;
	server.sin_addr.s_addr = inet_addr(SERVER_IP);
	server.sin_port = htons(SERVER_PORT);
	sd =  socket(AF_INET,SOCK_STREAM,0);
	bind(sd,(struct sockaddr*)&server,sizeof(server));
	listen(sd,5);
	while(1){
		nsd =accept(sd,(struct sockaddr*)&client,&clen);
	do{
		memset(msg,0x0,512);
		recv(nsd,msg,512,0);
		printf("\nData receievd:%s\n",msg);
		send(nsd,ack,strlen(ack)+1,0);

	}while(strcmp(msg,"stop"));
	close(sd);
	}

}
```

Output :

CLIENT

```
[root@localhost client]# gcc -o client10 client10.c
[root@localhost client]# ./client10
Enter a message:MCKVIE
Enter Window size:2
Acknowledgement received0
Acknowledgement received1
Acknowledgement received2
Enter a message:ABHISHEK
Enter Window size:4
Acknowledgement received3
Acknowledgement received4
Enter a message:QWERTYUIOP
Enter Window size:2
Acknowledgement received5
Acknowledgement received6
Acknowledgement received7
Acknowledgement received8
Acknowledgement received9
Enter a message:^C
```

SERVER

```
[root@localhost server]# gcc -o server10 server10.c
[root@localhost server]# ./server10

Data receievd:MC

Data receievd:KV

Data receievd:IE

Data receievd:ABHI

Data receievd:SHEK

Data receievd:QW

Data receievd:ER

Data receievd:TY

Data receievd:UI

Data receievd:OP

Data receievd:

Data receievd:
```