TABLE DESIGN

1.Table Name : CLIENTMASTER_10

Description : Used to store client information

Column_name	Datatype	Size	Attributes
CLIENT_NO	VARCHAR	6	Primary key , First letter must start with 'C'
NAME	VARCHAR	8	Not Null
ADDRESS1	VARCHAR	8	
ADDRESS2	VARCHAR	8	
CITY	VARCHAR	5	
PINCODE	NUMBER	8	
STATE	NUMBER	8	
BAL_DUE	NUMBER	5	

2.Table Name:PRODUCT_MASTER_10

Description : Used to store product information

Column_name	Datatype	Size	Attributes
PRODUCT_NO	VARCHAR	6	Primary key , First letter must start with 'P'
DESCRIPTION	VARCHAR	8	Not Null
PROFIT_PERCENT	NUMBER	2.1	
UNIT_MEASURE	VARCHAR	5	
QTY_ON_HAND	NUMBER	3	
RECODER_LVL	NUMBER	3	
SELL_PRICE	NUMBER	5	
COST_PRICE	NUMBER	5	

3 .Table Name : SALESMAN_MASTER_10

Description : Used to store salesman working for the company

Column_name	Datatype	Size	Attributes
SALESMAN_NO	VARCHAR	6	Primary key , First letter must start with 'S'
SALESMAN_NAME	VARCHAR	8	Not Null
ADDRESS	VARCHAR	5	Not Null
CITY	VARCHAR	8	
PINCODE	NUMBER	6	
STATE	VARCHAR	8	
SAL_AMT	NUMBER	6	Not Null, Cannot be Zero.
TGT_TO_GET	NUMBER	3	Not Null, Cannot be Zero.
TTD_SALES	NUMBER	3	Not Null
REMARKS	VARCHAR	4	

4.Table Name : SALES_ORDER_10

Description : Used to store client's orders

Column_name	Datatype	Size	Attributes
ORDER_NO	VARCHAR	7	Primary key , First letter must start with 'O'
ORDER_DATE	DATE		
CLIENT_NO	VARCHAR	6	Foreign key references CLIENT_NO of CLIENTMASTER_10
DELY_ADDR	VARCHAR	10	
SALESMN_NO	VARCHAR	6	Foreign key references SALESMN_NO of SALESMAN_MASTER_10
DELYTYPE	VARCHAR	3	
BILLED_YN	VARCHAR	3	
DELY_DATE	DATE		
ORDER_STATUS	VARCHAR	10	Values ('in process','fullfilled', 'backorder','cancelled')

5.Table Name : SALES_ORDER_DETAILS_10

Description : Used to store client's orders with details of each product ordered

Column_name	Datatype	Size	Attributes
ORDER_NO	VARCHAR	6	Primary key reference to ORDER_NO of SALES_ORDER_10
PRODUCT_NO	NUMBER	6	Foreign key reference to PRODUCT_NO of PRODUCR_MASTER34
QTY_ORDERED	NUMBER	3	
QTY_DISP	NUMBER	3	
PRODUCT_RATE	NUMBER	5	

6.Table Name : ITEM_ORDER_10

Description : Used to store item ordered by client.

Column_name	Datatype	Size	Attributes
CUST_NAME	VARCHAR	10	
ADDRESS	VARCHAR	10	
ITEM	VARCHAR	10	
PRICE	NUMBER	5	

TABLE DATA

1. Data for Client_master

Client_no	Name	Address1	Address2	City	Pincode	State	Bal_due
C00001	Ivan bayross	Wandon	Worli	Mumbai	450005	Maharashtra	15000
C00002	Vandana saitwa	Don Street	Bandra	Madras	780001	TamilNadu	0
C00003	pramadajagust	Mandon	Dadar	Mumbai	450007	Maharashtra	5000
C00004	Basu navindgi	Jerome	Juhu	Mumbai	450009	Maharashtra	0
C00005	Ravisreedharan	Dadar	Dadra	Delhi	100003	Delhi	2000
C00006	Rukmini	Rourk	Bandra	Mumbai	450002	Maharashtra	0

2.Data for Product_master

Product_no	Description	Profit_percent	Unit_measure	Qty_on_hand	Reorder_lvl	Sell_price	Cost_price
P00001	1.44 floppies	5	Piece	100	20	525	500
P03453	Monitors	6	Piece	10	3	12000	11280
P45789	Mouse	5	Piece	20	5	1050	1000
P44783	Keyboards	5	Piece	100	20	3150	3050
P45123	Cd drive	2	Piece	10	3	5250	5100
P35412	540 HDD	2,5	Piece	10	3	8400	8000

3.Data for
Salesman_master

Salesman_no	Salesman_name	Address	City	Pincode	State	Sal_amt	Tgt_to_get	Ttd_sales	Remarks
S00001	Kiran	a/14	Worli	450001	Bombay	3000	100	50	Good
S00002	Maneesh	j-65	Nariman	450001	Bombay	3000	200	100	Good
S00003	Ravi	p-7	Bandra	400003	Bombay	3000	200	100	Good
S00004	Ashish	a/5	Juhu	400041	Bombay	3000	200	150	Good

4. Data for Sales_order

Order_no	Order_date	Client_no	Dely_addr	Salesmn_no	Delytype	Billed_yn	Dely_date	Order_status
O19001	12-jan-96	C00001	Wandon	S00001	F	N	20-jan-96	In Process
O19002	25-jan-96	C00002	Don Street	S00002	P	N	27-jan-96	Cancelled
O46865	18-feb-96	C00003	Mandon	S00003	F	Y	20-feb-96	Fulfilled
O19003	03-apr-96	C00001	Jerome	S00001	F	Y	07-apr-96	Fulfilled
O46866	20-may-96	C00004	Dadar	S00002	P	N	22-may-96	Cancelled
O19008	24-may-96	C00005	Rourk	S00004	F	N	26-may-96	In Process

5. Data for Sales_order_details

Order_no	Product_no	Qty_ordered	Qty_disp	Product_rate
O19001	P00001	4	4	525
O19002	P00001	10	0	525
O46865	P00001	10	10	525
O46865	P03453	4	4	1050
O19003	P03453	2	2	1050
O19008	P00001	10	5	525

6. Data for item_order

Cust_name	Address	Item	Price
Basu	Worli	Monitor	1000
Ivan	Juhu	Keyboard	800
Ivan	Worli	Printer	1500
Bayross	Wandon	Printer	1500
Basu	Dadar	Mouse	500
Vandana	Wondon	Monitor	1000
Vandana	Wondon	Keyboard	800
Bayross	Juhu	Mouse	5000

DATE: …….

EXPERIMENT NO :1

CREATION AND ALTERATION OF TABLES

1 .CREATION OF TABLES

AIM: To perform creation and alteration of tables on

1. CLIENTMASTER_10
2. PRODUCT_MASTER_10
3. SALESMAN_MASTER_10
4. SALES_ORDER_10
5. SALES_ORDER_DETAILS_10
6. ITEM_ORDER_10

a) Create the following tables: CLIENTMASTER_10, PRODUCT_MASTER_10, SALESMAN_MASTER_10,
SALES_ORDER_10 ,Sales_order_deatils34 and ITEM_ORDER_10.

1. Table name: CLIENTMASTER_10

QUERY

```
SQL>  CREATE TABLE CLIENTMASTER_10
                        (CLIENT_NO VARCHAR2(6) PRIMARY KEY CHECK(CLIENT_NO LIKE 'C%'),
                        NAME VARCHAR2(8)NOT NULL,
                        ADDRESS1 VARCHAR2(8),
                        ADDRESS2 VARCHAR2(8),
                        CITY VARCHAR2(5),
                        PINCODE NUMBER(8),
                        STATE VARCHAR2(8),
                        BAL_DUE NUMBER(5));
```

OUTPUT

Table created

2. Table name: PRODUCT_MASTER_10

QUERY

```
 SQL>   CREATE TABLE PRODUCT_MASTER_10
                        (PRODUCT_NO VARCHAR2(6) PRIMARY KEY CHECK(CLIENT_NO LIKE 'P%'),
                        DESCRIPTION VARCHAR2(8) NOT NULL,
```

```
PROFIT_PERCENT NUMBER(2,1),
UNIT_MEASURE VARCHAR2(5),
QTY_ON_HAND NUMBER(3),
RECODER_LVL NUMBER(3),
SELL_PRICE NUMBER(5),
COST_PRICE NUMBER(5));
```

OUTPUT

Table created

3. Table name: SALESMAN_MASTER_10

QUERY

```
SQL>   CREATE TABLE SALESMAN_MASTER_10
                (SALESMAN_NO VARCHAR2(6) PRIMARY KEY CHECK(SALESMAN_NO LIKE 'S%'),
                 SALESMAN_NAME VARCHAR2(8) NOT NULL,
                ADDRESS VARCHAR(5)NOT NULL,
                CITY VARCHAR(8),PINCODE NUMBER(6),
                STATE VARCHAR(8),
                SAL_AMT NUMBER(6) CHECK(SAL_AMT>0) NOT NULL,
                TGT_TO_GET NUMBER(3) CHECK(TGT_TO_GET>0) NOT NULL,
                TTD_SALES NUMBER(3) NOT NULL,
                REMARKS VARCHAR(4));
```

OUTPUT

Table created

4. Table name: SALES_ORDER_10

QUERY

```
 SQL>   CREATE TABLE  SALES_ORDER_10
                (ORDER_NO VARCHAR2(7) PRIMARY KEY CHECK(ORDER_NO LIKE'O%'),
                 ORDER_DATE DATE,CLIENT_NO VARCHAR(6),
                 DELY_ADDR VARCHAR2(10),SALESMN_NO VARCHAR(6),
                 DELYTYPE VARCHAR2(3) DEFAULT 'F' CHECK(DELYTYPE IN('P','F')),
                 BILLED_YN VARCHAR2(3),
                 DELY_DATE DATE,
                ORDER_STATUS VARCHAR2(10) CHECK(ORDER_STATUS IN
                ('IN  PROCESS','FULLFILLED','BACKORDER','CANCELLED')));
```

OUTPUT

Table created

5. Table name: SALES_ORDER_DETAILS_10

QUERY

```
SQL>  CREATE TABLE SALES_ORDER_DETAILS_10
                    (ORDER_NO VARCHAR2(6),
                     PRODUCT_NO VARCHAR2(6),
                     QTY_ORDERED NUMBER(3),
                     QTY_DISP NUMBER(3),
                     PRODUCT_RATE NUMBER(5));
```

OUTPUT

Table created

6. Table name: ITEM_ORDER_10

QUERY

```
SQL> CREATE TABLE ITEM_ORDER_10
                    (CUST_NAME VARCHAR(10),
                    ADDRESS VARCHAR(10),
                    ITEM VARCHAR(10),
                    PRICE NUMBER(5));
```

OUTPUT

Table created.

2 .ALTERATION OF TABLES

b) Rename the Client_master table to clientmaster

QUERY

```
SQL> ALTER TABLE  CLIENT_MASTER_10  RENAME  TO CLIENTMASTER_10;
```

OUTPUT

Table altered

c) Add constraint to the Sales_order table to ensure that dely_date should be followed by the order_date.

QUERY

SQL> ALTER TABLE SALES_ORDER_10 ADD CHECK(DELY_DATE>ORDER_DATE);

OUTPUT

Table altered

d) Alter the product_master table to add the constraint that the cost_price must be greater than zero.

QUERY

SQL> ALTER TABLE PRODUCT_MASTER_10 ADD CHECK(COST_PRICE>0);

OUTPUT

Table altered

e) Alter the product_master table to add the constraint that the cost_price should always has a value.

QUERY

SQL> ALTER TABLE PRODUCT_MASTER_10 MODIFY (COST_PRICE NUMBER(5) NOT NULL);

OUTPUT

Table altered

f) Alter the product_master table to add the constraint that the sell_price must be greater than zero.

QUERY

SQL> ALTER TABLE PRODUCT_MASTER_10 ADD CHECK(SELL_PRICE>0);

OUTPUT

Table altered

RESULT

-

The creation and alteration of tables

1. CLIENTMASTER_10
2. PRODUCT_MASTER_10
3. SALESMAN_MASTER_10
4. SALES_ORDER_10
5. SALES_ORDER_DETAILS_10
6. ITEM_ORDER_10

were performed and output verified.

DATE: …….

EXPERIMENT NO :2

INSERTION OF VALUES INTO THE TABLES

AIM :To insert the table data in to the tables

1. CLIENTMASTER_10
2. PRODUCT_MASTER_10
3. SALESMAN_MASTER_10
4. SALES_ORDER_10
5. SALES_ORDER_DETAILS_10
6. ITEM_ORDER_10

1. Table name : CLIENTMASTER_10

QUERY

```
SQL> INSERT INTO  CLIENTMASTER_10 VALUES

        ('C00001','IVAN BAYROSS','WONDON','WORLI','MUMBAI','450005','MAHARASHTRA','15000');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO  CLIENTMASTER_10 VALUES
        ('C00002','VANDANA SAITWA','DON STREET','BANDRA','MADRAS','780001','TAMIL NADU','0');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO  CLIENTMASTER_10 VALUES
        (' C00003','PRAMADAJAGUST','MANDON','DADAR','MUMBAI','450007','MAHARASHTRA','5000');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO  CLIENTMASTER_10 VALUES
        ('C00004','BASUNAVINDGI','JEROME','JUHU','MUMBAI','450009','MAHARASHTRA','0');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO  CLIENTMASTER_10VALUES
        ('C00005','RAVISREEDHARAN','DADAR','DADRA','DELHI','100003','DELHI','2000');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO  CLIENTMASTER_10 VALUES
        ('C00006','RUKMINI','ROURK','BANDRA','MUMBAI','450002','MAHARASHTRA','0');
```

OUTPUT

1 row created.

2.Table Name:PRODUCT_MASTER_10

QUERY

```
SQL> INSERT INTO PRODUCT_MASTER_10 VALUES
        ('P00001','1.44 FLOPPIES',5,'PIECE','100','20','525','500');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO PRODUCT_MASTER_10 VALUES
        ('P03453','MONITORS',6,'PIECE','10','3','12000','11280');
```

OUTPUT

1 row created.
QUERY

```
SQL> INSERT INTO PRODUCT_MASTER_10 VALUES
        ('P45789','MOUSE',5,'PIECE','20','5','1050','1000');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO PRODUCT_MASTER_10 VALUES
        ('P44783','KEY BOARDS',5,'PIECE','100','20','3150','3050');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO PRODUCT_MASTER_10 VALUES
        ('P45123','CD DRIVE',2,'PIECE','10','3','5250','5100');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO PRODUCT_MASTER_10 VALUES
        ('P35412','540 HDD',2.5,'PIECE','10','3','8400','8000');
```

OUTPUT

1 row created.

3 .Table Name : SALESMAN_MASTER_10

QUERY

```
SQL> INSERT INTO SALESMAN_MASTER_10 VALUES
        ('S00001','KIRAN','a/4','WORLI',450001,'BOMBAY',3000,100,50,'GOOD');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALESMAN_MASTER_10 VALUES
```

```
('S00002','MANEESH','j-65','NARIMAN',450001,'BOMBAY',3000,200,100,'GOOD');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALESMAN_MASTER_10 VALUES
('S00003','RAVI','P-7','BANDRA',450003,'BOMBAY',3000,200,100,'GOOD');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALESMAN_MASTER_10 VALUES
('S00004','ASHISH','a/5','JUHU',450041,'BOMBAY',3000,200,150,'GOOD');
```

OUTPUT

1 row created.

4.Table Name : SALES_ORDER_10

QUERY

```
SQL> INSERT INTO SALES_ORDER_10 VALUES
('O19001','12-JAN-96','C00001','WANDON','S00001','F','N','20-JAN-96','IN PROCESS');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALES_ORDER_10 VALUES
('O19002','25-JAN-96','C00002','DON STREET','S00002','P','N','27-JAN-96','CANCELLED');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALES_ORDER_10 VALUES
('O46865','18-FEB-96','C00003','MANDON','S00003','F','Y','20-FEB-96','FULLFILLED');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALES_ORDER_10 VALUES
        ('O19003','03-APR-96','C00001','JEROME','S00001','F','Y','07-APR-96','FULLFILLED');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALES_ORDER_10 VALUES
        ('O46866','20-MAY-96','C00004','DADAR','S00002','P','N','22-MAY-96','CANCELLED');
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALES_ORDER_10 VALUES
        ('O19008','24-MAY-96','C00005','ROURK','S00004','F','N','26-MAY-96','IN PROCESS');
```

OUTPUT

1 row created.

5.Table Name : SALES_ORDER_DETAILS_10

QUERY

```
SQL> INSERT INTO SALES_ORDER_DETAILS_10 VALUES('O19001','P00001',4,4,525);
```

OUTPUT

1 row created.

QUERY

```
SQL> INSERT INTO SALES_ORDER_DETAILS_10 VALUES('O19002','P00001',10,0,525);
```

OUTPUT

1 row created.

QUERY

SQL> INSERT INTO SALES_ORDER_DETAILS_10 VALUES('O46865','P00001',10,10,525);

OUTPUT

1 row created.

QUERY

SQL> INSERT INTO SALES_ORDER_DETAILS_10 VALUES('O46865','P03453',4,4,1050);

OUTPUT

1 row created.

QUERY

SQL> INSERT INTO SALES_ORDER_DETAILS_10 VALUES('O19003','P03453',2,2,1050);

OUTPUT

1 row created.

QUERY

SQL> INSERT INTO SALES_ORDER_DETAILS_10 VALUES('O19008','P00001',10,5,525);

OUTPUT

1 row created.

6. Table name: ITEM_ORDER_10

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('BASU','WORLI','MONITOR',1000);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('IVAN','JUHU','KEYBOARD',800);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('IVAN','WORLI','PRINTER',1500);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('BAYROSS','WANDON','PRINTER',1500);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('BASU','DADAR','MOUSE',500);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('VANDANA','WONDON','MONITOR',1000);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('VANDANA','WONDON','KEYBOARD',800);

OUTPUT

1 row created.

QUERY

SQL> insert into ITEM_ORDER_10 VALUES ('BAYROSS','JUHU','MOUSE',500);

OUTPUT

1 row created.

RESULT

- The data are inserted into the tables and the output verified

DATE: …….

EXPERIMENT NO :3

COMPUTATION OF TABLE DATA

AIM : To perform computation on table data

a) Find the names of all salesman having ‘A’ as second letter in their name.

QUERY

```
SQL> select SALESMAN_NAME FROM SALESMAN_MASTER_10 WHERE SALESMAN_NAME LIKE '_A
%';
```

OUTPUT

```
SALESMAN
---------------
MANEESH
RAVI
```

b) Find all client whose name starts with ‘R’ and stays in ‘Delhi’ from client master table.

QUERY

```
SQL> select NAME,CITY FROM CLIENTMASTER_10 WHERE (NAME LIKE 'R%' AND CITY LIKE
'DELHI');
```

OUTPUT

```
NAME                          CITY
------------------------------  ----------
RAVISREEDHARAN       DELHI
```

c) List the name and address of client who stays in ‘Delhi’ or ‘Maharshtra’ from client_master table.

QUERY

```
SQL> select NAME,ADDRESS1,ADDRESS2 FROM CLIENTMASTER_10 WHERE
                                        (STATE LIKE 'DELHI' OR STATE  LIKE 'MAHARASHTRA');
```

OUTPUT

NAME	ADDRESS1	ADDRESS2
IVAN BAYROSS	WONDON	WORLI
PRAMADAJAGUST	MANDON	DADAR
BASU NAVINDGI	JEROME	JUHU
RAVISREEDHARAN	DADAR	DADRA
RUKMINI	ROURK	BANDR

d) List the product which cost more than Rs.5000 from product_master table.

QUERY

```
SQL> SELECT DESCRIPTION,COST_PRICE FROM PRODUCT_MASTER_10 WHERE COST_PRICE>5000;
```

OUTPUT

DESCRIPTION	COST_PRICE
MONITORS	11280
CD DRIVE	5100
540 HDD	8000

e) Print the information of orders placed in the month of ‘May’ from sales_order table.

QUERY

```
SQL>  SELECT * FROM SALES_ORDER_10 WHERE ORDER_DATE LIKE '%%-MAY-%%';
```

OUTPUT

ORDER_N	ORDER_DAT	CLIENT	DELY_ADDR	SALESM	DEL	BIL	DELY_DATE	ORDER_STAT
O46866	20-MAY-96	C00004	DADAR	S00002	P	N	22-MAY-96	CANCELLED
O19008	24-MAY-96	C00005	ROURK	S00004	F	N	26-MAY-96	IN PROCESS

f) Display the information of salesman whose salesman numbers are ‘s00001’ and ‘s00002’ from salesman_master.

QUERY

```
SQL> SELECT * FROM SALESMAN_MASTER_10 WHERE SALESMAN_NO LIKE 'S00001' OR
SALESMAN_NO LIKE 'S00002';
```

OUTPUT

S_NO	SALE	ADDRESS	CITY	PINCODE	STATE	SAL_AMT	TGT_TO_GET	TTD_SALES	REMA
S00001	KIRAN	a/4	WORLI	450001	BOMBAY	3000	100	50	GOOD
S00002	MANEESH	j-65	NARIMAN	450001	BOMBAY	3000	200	100	GOOD

g) Find out the product whose selling price is greater than 2000 and less than or equal to 5000 from product_master table.

QUERY

```
SQL> SELECT DESCRIPTION,SELL_PRICE FROM  PRODUCT_MASTER_10 WHERE SELL_PRICE>2000
AND SELL_PRICE<=5000;
```

OUTPUT

```
DESCRIPTION   SELL_PRICE
-------------------  ------------------
KEY BOARDS      3150
```

h) Find out the product whose selling price is more than 3000 and calculate a new selling-price as 20% of the original selling-price and display both from the product_master.

QUERY

```
SQL> SELECT DESCRIPTION,SELL_PRICE,(SELL_PRICE+.2*SELL_PRICE) AS HIKE PRICE FROM
PRODUCT_MASTER_10 WHERE SELL_PRICE>3000;
```

OUTPUT

```
DESCRIPTION    SELL_PRICE                    HIKE PRICE
-------------- ---------- ----------------------------------------------------
MONITORS         12000                          14400
KEY BOARDS       3150                            3780
CD DRIVE         5250                            6300
540 HDD          8400                           10080
```

i) Count the total number of product from product_master table.

QUERY

```
SQL> select COUNT(DESCRIPTION) FROM PRODUCT_MASTER_10
```

OUTPUT

```
COUNT(DESCRIPTION)
---------------------------------
                6
```

j) Count the total number of orders ‘fulfilled’ in sales_order table.

QUERY

```
SQL> select COUNT(ORDER_STATUS) FROM SALES_ORDER_10
                                                WHERE ORDER_STATUS LIKE 'FULLFILLED';
```

OUTPUT

```
COUNT(ORDER_STATUS)
-----------------------------------
                  2
```

k) Calculate the average price of all product from product_master table.

QUERY

```
SQL> SELECT AVG(COST_PRICE)FROM PRODUCT_MASTER_10;
```

OUTPUT

```
AVG(COST_PRICE)
--------------
    4821.66667
```

RESULT

The SQL block of computation on table data are executed successfully and the output verified.

DATE: …….

EXPERIMENT NO :4

DATE MANIPULATION

AIM: To working with date functions.

a) Display the order_no and day on which clients placed their orders.

QUERY

```
SQL> SELECT ORDER_NO,ORDER_DATE,TO_CHAR(TO_DATE(ORDER_DATE) ,'DAY') AS  DAY FROM
SALES_ORDER_10;
```

OUTPUT

```
ORDER_NO   ORDER_DATE    DAY
------------------------------------------------------------------------
O19001          12-JAN-96              FRIDAY
O19002          25-JAN-96              THURSDAY
O46865          18-FEB-96              SUNDAY
O19003          03-APR-96              WEDNESDAY
O46866           20-MAY-96             MONDAY
O19008          24-MAY-96               FRIDAY
```

6 rows selected.

b) Display the month and date which the order must be delivered.

QUERY

```
SQL> SELECT TO_CHAR(DELY_DATE,'MONTH') AS MONTH ,DELY_DATE FROM SALES_ORDER_10;
```

OUTPUT

```
MONTH           DELY_DATE
---------------------------------------
JANUARY     20-JAN-96
JANUARY     27-JAN-96
FEBRUARY   20-FEB-96
APRIL          07-APR-96
MAY            22-MAY-96
MAY            26-MAY-96
```

6 rows selected.

c) Display the order_date in the format ‘dd/month/yy’.

QUERY

```
SQL> select ORDER_NO,TO_CHAR
                        (ORDER_DATE,'DD/MONTH/YY')ORDER_DATE FROM SALES_ORDER_10;
```

OUTPUT

```
ORDER_NO        ORDER_DATE
-------------------------------------------
O19001          12/JANUARY  /96
O19002          25/JANUARY  /96
O46865          18/FEBRUARY /96
O19003          03/APRIL    /96
O46866          20/MAY      /96
O19008          24/MAY      /96
```

d) Find the date,15 days after today’s date.

QUERY

```
SQL> SELECT SYSDATE+15 FROM DUAL;
```

OUTPUT

```
SYSDATE+1
---------
24-OCT-17
```

e) Find the no. of days elapsed between order-date and delivery date of the order placed by the clients.

QUERY

```
SELECT DELY_DATE-ORDER_DATE AS DATE_ELAPSED FROM SALES_ORDER_10;
```

OUTPUT

```
DATE_ELAPSED
------------
           8
           2
           2
           4
           2
           2
```

RESULT

The SQL block for date manipulation was successfully completed and the output verified.

DATE: …….

EXPERIMENT NO :5

HAVING AND GROUP BY CLAUSES

AIM : To perform queries with having and group by clause.

a) Display total cost price of each item.

QUERY

```
SQL> select Item,SUM(Price) from ITEM_ORDER_10 GROUP BY Item;
```

OUTPUT

```
ITEM      SUM(PRICE)
---------- ----------
MONITOR        2000
PRINTER        3000
MOUSE                  1000
KEYBOARD     1600
```

b) Display no. of items ordered by the customer Ivan.

QUERY

```
SQL>  SELECT Cust_name,Count(Item) as Total_item_ordered from ITEM_ORDER_10
                                        GROUP BY Cust_name HAVING Cust_name='IVAN';
```

OUTPUT

```
CUST_NAME       TOTAL_ITEM_ORDERED
-------------------- ------------------------------------
IVAN                           2
```

c) Display no. of items that are ordered from the address Juhu.

QUERY

```
SQL> SELECT Address,count(item)as No_of_items from ITEM_ORDER_10
                                        GROUP BY Address HAVING Address='JUHU';
```

OUTPUT

```
ADDRESS    NO_OF_ITEMS
---------- ----------
JUHU                2
```

d) Display the name of customers whose ordered price is greater than 2000

QUERY

```
SQL> SELECT Cust_name,SUM(Price) as Total_Price from ITEM_ORDER_10 Group by Cust_name  having
sum(price) >'2000';
```

OUTPUT

```
CUST_NAME  TOTAL_PRICE
---------- -----------
IVAN              2300
```

RESULT

The queries for performing the having and group by clause on the table data were executed successfully and verified the ouput.

DATE: …….

EXPERIMENT NO :6

SUBQUERIES

AIM : To perform manipulation on table data using subqueries

a) Find customer_name, address1, address2, city, pincode for clients who has placed order_no ‘o19001’.

QUERY

```
SQL> SELECT NAME,ADDRESS1,ADDRESS2,CITY,PINCODE
                        FROM CLIENTMASTER_10 where CLIENT_NO IN
                        (SELECT CLIENT_NO FROM SALES_ORDER WHERE ORDER_NO='O19001');
```

OUTPUT

```
NAME                 ADDRESS1  ADDRESS2       CITY          PINCODE
-------------------------------------------------------------------------------------------
IVAN BAYROSS         WONDON    WORLI          MUMBAI        450005
```

b) Find the client names that have placed orders before the month of 01-may-96

QUERY

```
SQL> SELECT NAME FROM CLIENTMASTER_10 where CLIENT_NO IN
                (SELECT CLIENT_NO FROM SALES_ORDER_10 WHERE ORDER_DATE < '01-MAY-96');
```

OUTPUT

```
NAME
--------------------
IVAN BAYROSS
VANDANA SAITWA
PRAMADAJAGUST
```

c) Find out if the product ‘1.44 floppies’ has been ordered by any client and print the client_no, name to whom it was sold.

QUERY

```
SQL> SELECT CLIENT_NO,NAME FROM CLIENTMASTER_10
                WHERE CLIENT_NO IN(SELECT CLIENT_NO FROM SALES_ORDER_10
                WHERE ORDER_NO IN (SELECT ORDER_NO FROM SALES_ORDER_DETAILS_10
                WHERE PRODUCT_NO IN( SELECT PRODUCT_NO FROM PRODUCT_MASTER_10
                WHERE DESCRIPTION='1.44 FLOPPIES')));
```

OUTPUT

```
CLIENT_NO       NAME
--------------- ------------------
C00001          IVAN BAYROSS
C00002          VANDANA SAITWA
C00003          PRAMADAJAGUST
C00005          RAVISREEDHARAN
```

d) Find the product_no and description of non-moving products i.e, products not being sold.

QUERY

```
SELECT SALES_ORDER_DETAILS_10.PRODUCT_NO,DESCRIPTION
        FROM SALES_ORDER_DETAILS_10,PRODUCT_MASTER_10 WHERE QTY_DISP=0 AND
        SALES_ORDER_DETAILS_10.PRODUCT_NO=PRODUCT_MASTER_10.PRODUCT_NO;
```

OUTPUT

```
PRODUC DESCRIPTION
------ ---------------
P00001 1.44 FLOPPIES
```

e) Find the names of clients who have placed orders worth Rs. 10000 or more

QUERY

```
SQL> SELECT NAME FROM CLIENTMASTER_10
                WHERE CLIENT_NO IN(SELECT CLIENT_NO FROM SALES_ORDER_10
                WHERE ORDER_NO IN)SELECT ORDER_NO FROM SALES_ORDER_DETAILS_10
                WHERE (QTY_ORDERED*PRODUCT_RATE)>=10000));
```

OUTPUT

```
NAME               PRODUCT_NO SELL_PRICE
------------------- ---------- -------------------------------
IVAN BAYROSS            P03453          12000
PRAMADAJAGUST           P03453          12000
```

RESULT

The queries for performing manipulation on the table data using subqueries were executed successfully and verified the ouput.

DATE: ……

EXPERIMENT NO :7

VIEWS IN SQL

AIM: To create a view of a table.

a) Create a view of the table client_master (Client_no, Name, City)

QUERY

```
SQL> CREATE VIEW CLIENT_DETAILS AS SELECT CLIENT_NO,NAME,CITY FROM
CLIENTMASTER_10;
```

OUTPUT

View created.

```
SQL> SELECT * FROM CLIENT_DETAILS;

CLIENT NAME                    CITY
------ -------------------- ----------
C00001 IVAN BAYROSS            MUMBAI
C00002 VANDANA SAITWA          MADRAS
C00003 PRAMADAJAGUST           MUMBAI
C00004 BASU NAVINDGI           MUMBAI
C00005 RAVISREEDHARAN          DELHI
C00006 RUKMINI                 MUMBAI
```

b) Updating the view and checking the table data

QUERY

```
SQL> update CLIENT_DETAILS SET CITY='TRIVANDRUM' WHERE CLIENT_NO='C00003";
```

OUTPUT

1 row updated.

QUERY

```
SQL> SELECT * FROM CLIENT_DETAILS;
```

OUTPUT

CLIENT NAME	CITY
C00001 IVAN BAYROSS	MUMBAI
C00002 VANDANA SAITWA	MADRAS
C00003 PRAMADAJAGUST	TRIVANDRUM
C00004 BASU NAVINDGI	MUMBAI
C00005 RAVISREEDHARAN	DELHI
C00006 RUKMINI	MUMBAI

6 rows selected.

RESULT

The queries for performing manipulation on the table data by creating a view were executed successfully and verified the ouput.