**ASSIGNMENT - 2**

**[A] Perform the following computations on table data:**

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

SELECT NAME FROM CLIENT\_MASTER WHERE NAME LIKE '\_a%';

**2. List the clients who stay in a city whose First letter is ‘M’**.

SELECT \* FROM CLIENT\_MASTER WHERE CITY LIKE 'M%';

**3. List all the clients who stay in ‘Bangalore’ or ‘Mangalore’.**

SELECT \* FROM CLIENT\_MASTER WHERE CITY IN('Bangalore', 'Mangalore');

**4. List all clients whose BalDue is greater than value 10000**.

SELECT \* FROM CLIENT\_MASTER WHERE BALDUE > 10000;

**5. List all information from the Sales\_Order table for orders placed in the month of June.**

SELECT \* FROM SALES\_ORDER WHERE MONTH(ORDERDATE) = '6’';

**6. List the order information for ClientNo. ‘C00001’ and ‘C00002’.**

SELECT \* FROM sales\_order WHERE CLIENTNO IN('C00001', 'C00002');

**7. List products whose selling price is greater than 500 and less than or equal to 750.**

SELECT \* FROM product\_master WHERE SELLPRICE > '500' AND SELLPRICE <= '750';

**8. List products whose selling price is more than 500.**

SELECT \* FROM product\_master WHERE SELLPRICE > '500';

**9. Calculate a new selling price as, original selling price \*0.15.**

SELECT SELLPRICE\*0.15 FROM product\_master;

**10. Rename the new column in the output of the above query as new\_price.**

SELECT SELLPRICE\*0.15 "NEW\_PRICE" FROM product\_master;

**11. List the names, city and state of clients who are not in the state of ‘Maharashtra’.**

SELECT NAME, CITY, STATE FROM client\_master WHERE STATE NOT IN('MAHARASHTRA');

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

SELECT AVG(COSTPRICE) FROM product\_master;

**13. Determine the maximum and minimum product prices. Rename the output as max\_price and min\_price respectively.**

SELECT MAX(COSTPRICE) "MAX\_PRICE", MIN(COSTPRICE) "MIN\_PRICE" FROM product\_master;

**14. Count the number of products having price less than or equal to 500**.

SELECT COUNT(PRODUCTNO) FROM product\_master WHERE COSTPRICE <= '500';

**15. List all the products whose QtyOnHand is less than reorder level.**

SELECT \* FROM product\_master WHERE QTYONHAND < REORDERLVL;

**[B]Exercise on Date Manipulation:**

**1. List the order number and day on which clients placed their order.**

SELECT ORDERNO, DAY(ORDERDATE) FROM sales\_order;

**2. List the month (in alphabets) and date when the orders must be delivered.**

SELECT MONTHNAME(DELYDATE), DAY(DELYDATE) FROM sales\_order;

**3. List the Order Date in the format ‘DD-Month-YY’. E.g. 20-November-02.**

SELECT DATE\_FORMAT(ORDERDATE, '%d-%M-%y') FROM sales\_order;

**4. List the date, 15 days after today’s date.**

SELECT DATE\_FORMAT(DATE\_ADD(CURDATE(), INTERVAL 15 DAY ), '%d-%M-%y');

**[C] Exercise on using Having and Group By Clauses:**

**1. Print the description and total qty sold for each product.**

SELECT product\_master.DESCRIPTION, SUM(sales\_order\_details.QTYORDERED) FROM sales\_order\_details INNER JOIN product\_master ON product\_master.PRODUCTNO = sales\_order\_details.PRODUCTNO GROUP BY product\_master.PRODUCTNO;

**2. Find the values of each product sold.**

SELECT product\_master.DESCRIPTION, SUM(sales\_order\_details.QTYORDERED \* sales\_order\_details.PRODUCTRATE)

FROM sales\_order\_details INNER JOIN product\_master ON product\_master.PRODUCTNO = sales\_order\_details.PRODUCTNO

GROUP BY product\_master.PRODUCTNO;

**3. Calculate the average qty sold for each client that has a maximum order value of 15000.00.**

SELECT sales\_order.ORDERNO, sales\_order.CLIENTNO, AVG(QTYDISP), MAX(sales\_order\_details.QTYORDERED\*sales\_order\_details.PRODUCTRATE)

FROM sales\_order INNER JOIN sales\_order\_details ON sales\_order.ORDERNO = sales\_order\_details.ORDERNO

GROUP BY sales\_order\_details.ORDERNO

HAVING SUM(sales\_order\_details.QTYORDERED\*sales\_order\_details.PRODUCTRATE) > 15000;

**4. Find out the total of all the billed orders for the month of June.**

**[D] Exercise on Joins and Correlation:**

**1. Find out the products, which have been sold to ‘Rakesh Joshi’.**

SELECT client\_master.name, sales\_order.ORDERNO, sales\_order\_details.PRODUCTNO, product\_master.DESCRIPTION FROM client\_master

INNER JOIN sales\_order ON client\_master.CLIENTNO = sales\_order.CLIENTNO

INNER JOIN sales\_order\_details ON sales\_order.ORDERNO = sales\_order\_details.ORDERNO

INNER JOIN product\_master ON product\_master.PRODUCTNO = sales\_order\_details.PRODUCTNO

where client\_master.NAME = 'Rakesh Joshi';

**2. Find out the products and their quantities that will have to be delivered in the current month.**

SELECT sales\_order.ORDERNO, sales\_order\_details.PRODUCTNO, product\_master.DESCRIPTION, sales\_order\_details.QTYDISP FROM sales\_order

INNER JOIN sales\_order\_details ON

sales\_order.ORDERNO = sales\_order\_details.ORDERNO

INNER JOIN product\_master ON

sales\_order\_details.PRODUCTNO = product\_master.PRODUCTNO

WHERE MONTH(DELYDATE) = '02';

**3. List the ProductNo and description of constantly sold (i.e. rapidly moving) products.**

SELECT product\_master.PRODUCTNO, product\_master.DESCRIPTION, COUNT(sales\_order\_details.QTYORDERED) FROM product\_master

INNER JOIN sales\_order\_details ON

product\_master.PRODUCTNO = sales\_order\_details.PRODUCTNO

GROUP BY PRODUCTNO HAVING COUNT(sales\_order\_details.QTYORDERED);

**4. Find the names of clients who have purchased ‘Trousers’.**

SELECT client\_master.CLIENTNO, client\_master.NAME FROM client\_master

INNER JOIN sales\_order ON

client\_master.CLIENTNO = sales\_order.CLIENTNO

INNER JOIN sales\_order\_details ON

sales\_order.ORDERNO = sales\_order\_details.ORDERNO

INNER JOIN product\_master ON

product\_master.PRODUCTNO = sales\_order\_details.PRODUCTNO

where product\_master.DESCRIPTION = 'Trousers';

**5. List the products and orders from customers who have ordered less than 5 units of ‘Pull Overs’.**

SELECT sales\_order.CLIENTNO, sales\_order.ORDERNO, product\_master.DESCRIPTION, sales\_order\_details.QTYORDERED FROM sales\_order

INNER JOIN sales\_order\_details ON

sales\_order.ORDERNO = sales\_order\_details.ORDERNO

INNER JOIN product\_master ON

sales\_order\_details.PRODUCTNO = product\_master.PRODUCTNO

WHERE product\_master.DESCRIPTION = 'Pull Overs' AND sales\_order\_details.QTYORDERED<5;

**6. Find the products and their quantities for the orders placed by ‘Rakesh Joshi’ and ‘Mayur Patel’.**

SELECT client\_master.CLIENTNO, client\_master.NAME, product\_master.PRODUCTNO, product\_master.DESCRIPTION, sales\_order\_details.QTYORDERED

FROM client\_master

INNER JOIN sales\_order ON

sales\_order.CLIENTNO = client\_master.CLIENTNO

INNER JOIN sales\_order\_details ON

sales\_order.ORDERNO = sales\_order\_details.ORDERNO

INNER JOIN product\_master ON

sales\_order\_details.PRODUCTNO = product\_master.PRODUCTNO

WHERE client\_master.NAME IN ('Rakesh Joshi', 'Mayur Patel')

**7. Find the products and their quantities for the orders placed by ClientNo ‘C00001’ and ‘C00002’.**

SELECT client\_master.CLIENTNO, client\_master.NAME, product\_master.PRODUCTNO, product\_master.DESCRIPTION, sales\_order\_details.QTYORDERED

FROM client\_master

INNER JOIN sales\_order ON

sales\_order.CLIENTNO = client\_master.CLIENTNO

INNER JOIN sales\_order\_details ON

sales\_order.ORDERNO = sales\_order\_details.ORDERNO

INNER JOIN product\_master ON

sales\_order\_details.PRODUCTNO = product\_master.PRODUCTNO

WHERE client\_master.CLIENTNO IN ('C00001', 'C00002')

**[E] Exercise on Subqueries:**

**1. Find the ProductNo and description of non-moving products i.e. products not being sold.**

SELECT product\_master.PRODUCTNO, product\_master.DESCRIPTION FROM product\_master WHERE PRODUCTNO NOT IN(SELECT sales\_order\_details.PRODUCTNO from sales\_order\_details);

**2. List the CustomerName, Address1, Address2, City and Pin Code for the client who has placed order no ‘O19001’.**

SELECT client\_master.NAME, client\_master.ADDRESS1, client\_master.ADDRESS2, client\_master.CITY, client\_master.PINCODE FROM client\_master

WHERE CLIENTNO = (SELECT sales\_order.CLIENTNO FROM sales\_order WHERE ORDERNO = 'O19001');

**3. List the client names that have placed orders before the month of May-04**

SELECT client\_master.NAME FROM client\_master

WHERE client\_master.CLIENTNO IN(SELECT sales\_order.CLIENTNO

FROM sales\_order WHERE MONTH(ORDERDATE) < '5' AND YEAR(ORDERDATE) = '2004');

**4. List if the product ‘Lycra Top’ has been ordered by any client and print the Client No Name to whom it was sold.**

SELECT client\_master.CLIENTNO, client\_master.NAME FROM client\_master WHERE client\_master.CLIENTNO IN

(SELECT sales\_order.CLIENTNO FROM sales\_order WHERE sales\_order.ORDERNO IN

((SELECT sales\_order\_details.ORDERNO FROM sales\_order\_details WHERE PRODUCTNO=

(SELECT product\_master.PRODUCTNO FROM product\_master WHERE product\_master.DESCRIPTION = 'Lycra Tops'))));

**5. List the names of clients who have placed orders worth Rs. 10000 or more.**

SELECT NAME FROM client\_master WHERE CLIENTNO IN

(SELECT CLIENTNO FROM sales\_order WHERE ORDERNO IN

(SELECT ORDERNO FROM sales\_order\_details GROUP BY ORDERNO HAVING SUM(QTYORDERED\*PRODUCTRATE) > 10000));

**Query for creation of database and insertion of data:**

1) create table client\_master(

CLIENTNO varchar(6) CHECK(CLIENTNO LIKE 'C%') PRIMARY KEY,

NAME varchar(20) NOT NULL,

ADDRESS1 varchar(30),

ADDRESS2 varchar(30),

CITY varchar(15),

PINCODE numeric(8),

STATE varchar(15),

BALDUE numeric(10,2)

);

2) INSERT INTO client\_master(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE)

VALUES

('C00001', 'Rakesh Joshi', 'Mumbai', '400054', 'Maharashtra', '15000'),

('C00002', 'Mayur Patel', 'Madras', '780001', 'Tamilnadu', '0'),

('C00003', 'Ishita Mehta', 'Mumbai', '400057', 'Maharashtra', '5000'),

('C00004', 'Amit Solanki', 'Bangalore', '560001', 'Karnataka', '0'),

('C00005', 'Hiren Pandya', 'Mumbai', '400060', 'Maharashtra', '2000'),

('C00006', 'Dipak Sharma', 'Mangalore', '560050', 'Karnataka', '0');

3) ALTER TABLE product\_master

ADD CONSTRAINT CHK CHECK(SELLPRICE <> 0 AND COSTPRICE <> 0);

4) ALTER TABLE sman\_mast

RENAME TO salesman\_master;

5) ALTER TABLE salesman\_master ADD CONSTRAINT Primary\_key PRIMARY KEY(SALESMANNO), ADD CONSTRAINT fl CHECK(SALESMANNO LIKE 'S%')

6) ALTER TABLE salesman\_master

MODIFY COLUMN SALESMANNAME VARCHAR(20) NOT NULL,

MODIFY ADDRESS1 varchar(30) NOT NULL,

MODIFY SALAMT NUMERIC(8, 2) NOT NULL,

MODIFY TGTTOGET NUMERIC(6, 2) NOT NULL,

MODIFY YTDSALES NUMERIC(6, 2) NOT NULL;

7) ALTER TABLE salesman\_master

ADD CONSTRAINT not\_zero CHECK(SALAMT <> 0 AND TGTTOGET <> 0 AND YTDSALES <> 0);

8) Create table sales\_order(

ORDERNO VARCHAR(6) PRIMARY KEY CHECK(ORDERNO LIKE 'O%'),

CLIENTNO VARCHAR(6),

ORDERDATE DATE NOT NULL,

DELYADDR VARCHAR(25),

SALESMANNO VARCHAR(6),

DELYTYPE CHAR(1) DEFAULT 'F',

BILLYN CHAR(1),

DELYDATE DATE,

ORDERSTATUS VARCHAR(10) CHECK(ORDERSTATUS IN ('In Process', 'Fullfilled', 'BackOrder', 'Cancelled')),

FOREIGN KEY(CLIENTNO) REFERENCES client\_master(CLIENTNO),

FOREIGN KEY(SALESMANNO) REFERENCES salesman\_master(SALESMANNO),

CONSTRAINT CHK\_DELYTYPE\_CHAR CHECK(DELYTYPE IN('P', 'F')),

CONSTRAINT CHK\_DATE CHECK(DELYDATE >= ORDERDATE)

);

9) ALTER TABLE sales\_order

DROP constraint CHK\_DATE

10) ALTER TABLE sales\_order

ADD CONSTRAINT CHK\_DELYDATE CHECK(STR\_TO\_DATE(DELYDATE) > STR\_TO\_DATE(ORDERDATE));

11) INSERT INTO sales\_order(ORDERNO, CLIENTNO, ORDERDATE, SALESMANNO, DELYTYPE, BILLYN, DELYDATE, ORDERSTATUS)

VALUES

('O19001', 'C00001', STR\_TO\_DATE('12-June-04', '%d-%M-%y'), 'S00001', 'F', 'N', STR\_TO\_DATE('20-July-04', '%d-%M-%y'), 'In Process'),

('O19002', 'C00002', STR\_TO\_DATE('25-June-04', '%d-%M-%y'), 'S00002', 'P', 'N', str\_to\_date('27-June-04', '%d-%M-%y'), 'Cancelled'),

('O46865', 'C00003', STR\_TO\_DATE('18-Feb-04', '%d-%M-%y'), 'S00003', 'F', 'Y', str\_to\_date('20-Feb-04', '%d-%M-%y'), 'Fulfilled'),

('O19003', 'C00004', STR\_TO\_DATE('03-Apr-04', '%d-%M-%y'), 'S00001', 'F', 'Y', str\_to\_date('07-Apr-02', '%d-%M-%y'), 'Fullfilled'),

('O46866', 'C00005', STR\_TO\_DATE('20-May-04', '%d-%M-%y'), 'S00002', 'P', 'N', str\_to\_date('22-May-04', '%d-%M-%y'), 'Cancelled'),

('O19008', 'C00006', STR\_TO\_DATE('24-May-04', '%d-%M-%y'), 'S00004', 'F', 'N', str\_to\_date('26-July-04', '%d-%M-%y'), 'Fullfilled');

12) create table sales\_order\_details(

ORDERNO VARCHAR(6),

PRODUCTNO VARCHAR(6),

QTYORDERED NUMERIC(8),

QTYDISP NUMERIC(8),

PRODUCTRATE NUMERIC(10, 2),

FOREIGN KEY(ORDERNO) REFERENCES sales\_order(ORDERNO),

FOREIGN KEY(PRODUCTNO) REFERENCES product\_master(PRODUCTNO)

);

13)INSERT INTO SALES\_ORDER\_DETAILS(ORDERNO, PRODUCTNO, QTYORDERED, QTYDISP, PRODUCTRATE)

VALUES

('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', 'P0345', '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')

;

END