SOLUTIONS TO HANDS ON EXERCISES

7. INTERACTIVE SQL PART - I

1. **SQL Statement for creating the tables:**

a) **Table Name:** CLIENT_MASTER

```
CREATE TABLE CLIENT_MASTER(CLIENTNO varchar2(6), NAME varchar2(20),
    ADDRESS1 varchar2(30), ADDRESS2 varchar2(30), CITY varchar2(15),
    PINCODE number(8), STATE varchar2(15), BALDUE number(10,2));
```

b) **Table Name:** PRODUCT_MASTER

```
CREATE TABLE PRODUCT_MASTER(PRODUCTNO varchar2(6), DESCRIPTION varchar2(15),
    PROFITPERCENT number(4,2), UNITMEASURE varchar2(10), QTYONHAND number(8),
    REORDERLVL number(8), SELLPRICE number(8,2), COSTPRICE number(8,2));
```

c) **Table Name:** SALESMAN_MASTER

```
CREATE TABLE SALESMAN_MASTER(SALESMANNO varchar2(6),
    SALESMANNAME varchar2(20), ADDRESS1 varchar2(30), ADDRESS2 varchar2(30),
    CITY varchar2(20), PINCODE number(8), STATE varchar2(20), SALAMT number(8,2),
    TGTTOGET number(6,2), YTDSALES number(6,2), REMARKS varchar2(60));
```

2. **SQL Statement for inserting into their respective tables:**

a) Data for **CLIENT_MASTER** table:

```
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00001', 'Ivan Bayross', 'Mumbai', 400054, 'Maharashtra', 15000);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00002', 'Mamta Muzumdar', 'Madras'', 780001, 'Tamil Nadu', 0);
INSERT INTO Client_Master (ClientNo, Name, City, Pincode, State, BalDue)
    VALUES ('C00003', 'Chhaya Bankar', 'Mumbai', 400057, 'Maharashtra', 5000);
INSFRT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00004', 'Ashwini Joshi', 'Bangalore', 560001, 'Karnataka', 0);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00005', 'Hansel Colaco', 'Mumbai', 400060,  'Maharashtra', 2000);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00006', 'Deepak Sharma', 'Mangalore', 560050, 'Karnataka', 0);
```

b) Data for **PRODUCT_MASTER** table

```
INSERT INTO Product_Master VALUES ('P00001', 'T-Shirts', 5, 'Piece', 200, 50, 350, 250);
INSERT INTO Product_Master VALUES ('P03453', 'Shirts', 6, 'Piece', 150, 50, 500, 350);
INSERT INTO Product_Master VALUES ('P06734', 'Cotton Jeans', 5, 'Piece', 100, 20, 600, 450);
INSERT INTO Product_Master VALUES ('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500);
INSERT INTO Product_Master VALUES ('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550);
INSERT INTO Product_Master VALUES ('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450);
INSERT INTO Product_Master VALUES ('P07965', 'Denim Shirts', 4, 'Piece', 100, 40, 350, 250);
INSERT INTO Product_Master VALUES ('P07975', 'Lycra Tops', 5, 'Piece', 70, 30, 300, 175);
INSERT INTO Product_Master VALUES ('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);
```

d) Data for **SALESMAN_MASTER** table

```
INSERT INTO Salesman_Master VALUES ('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002,
                                    'Maharashtra', 3000, 100, 50, 'Good');
INSERT INTO Salesman_Master VALUES ('S00002', 'Omkar', '65', 'Nariman', 'Mumbai', 400001,
                                    'Maharashtra', 3000, 200, 100, 'Good');
INSERT INTO Salesman_Master VALUES ('S00003', 'Raj', 'P-7', 'Bandra', 'Mumbai', 400032,
                                    'Maharashtra', 3000, 200, 100, 'Good');
INSERT INTO Salesman_Master VALUES ('S00004', 'Ashish', 'A/5', 'Juhu', 'Bombay', 400044,
                                    'Maharashtra', 3500, 200, 150, 'Good');
```

3. **SQL Statement for retrieving records from a table:**

a) Find out the names of all the clients.

```
SELECT Name FROM Client_Master;
```

b) Retrieve the entire contents of the Client_Master table.

```
SELECT * FROM Client_Master;
```

c) Retrieve the list of names, city and the sate of all the clients.

```
SELECT Name, City, State FROM Client_Master;
```

d) List the various products available from the Product_Master table.

```
SELECT Description FROM Product_Master;
```

e) List all the clients who are located in Mumbai.

```
SELECT * FROM Client_Master WHERE City = 'Mumbai';
```

f) Find the names of salesmen who have a salary equal to Rs.3000.

```
SELECT Salesman_name FROM Salesman_Master WHERE SalAmt = 3000;
```

4. **SQL Statement for updating records in a table:**

a) Change the city of ClientNo 'C00005' to 'Bangalore'.

```
UPDATE Client_Master SET City = 'Bangalore' WHERE ClientNo = 'C00005';
```

b) Change the BalDue of ClientNo 'C00001' to Rs. 1000.

```
UPDATE Client_Master SET BalDue = 1000 WHERE Client_no = 'C00001';
```

c) Change the cost price of 'Trousers' to Rs. 950.00.

```
UPDATE Product_Master SET CostPrice = 950.00 WHERE Description = 'Trousers';
```

d) Change the city of the salesman to Pune.

```
UPDATE Client_Master SET City = 'Pune';
```

5. **SQL Statement for deleting records in a table:**

a) Delete all salesmen from the Salesman_Master whose salaries are equal to Rs. 3500.

```
DELETE FROM Salesman_Master WHERE SalAmt = 3500;
```

b) Delete all products from Product_Master where the quantity on hand is equal to 100.

```
DELETE FROM Product_Master WHERE QtyOnHand = 100;
```

c) Delete from Client_Master where the column state holds the value 'Tamil Nadu'.

```
DELETE FROM Client_Master WHERE State = 'Tamil Nadu';
```

6. **SQL Statement for altering the table structure:**

a) Add a column called 'Telephone' of data type 'number' and size ='10' to the Client_Master table.

```
ALTER TABLE Client_Master ADD (Telephone number(10));
```

b) Change the size of SellPrice column in Product_Master to 10,2.

```
ALTER TABLE Product_Master MODIFY (SellPrice number(10,2));
```

7. **SQL Statement for deleting the table structure along with the data:**

a) Destroy the table Client_Master along with its data.

```
DROP TABLE Client_Master;
```

8. **SQL Statement for renaming the table:**

a) Change the name of the Salesman_Master table to sman_mast.

```
RENAME Salesman_Master TO sman_mast;
```

8. INTERACTIVE SQL PART - II

1. **SQL Statement for creating the tables:**

Note

Before, executing the CREATE TABLE with the Data Constraints ensure that the tables with similar names do not exist within the database. Executing the following SQL commands will eliminate the problem of existing tables.

```
DROP TABLE IF EXISTS SALES_ORDER_DETAILS;
DROP TABLE IF EXISTS SALES_ORDER;
DROP TABLE IF EXISTS SALESMAN_MASTER;
DROP TABLE IF EXISTS PRODUCT_MASTER;
DROP TABLE IF EXISTS CLIENT_MASTER;
```

a) **Table Name:** CLIENT_MASTER

```
CREATE TABLE CLIENT_MASTER(CLIENTNO varchar2(6) PRIMARY KEY,
    NAME varchar2(20) NOT NULL, ADDRESS1 varchar2(30), ADDRESS2 varchar2(30),
    CITY varchar2(15), PINCODE number(8), STATE varchar2(15), BALDUE number(10,2),
    CONSTRAINT ck_client CHECK (CLIENTNO like 'C%'));
```

b) **Table Name:** PRODUCT_MASTER

```
CREATE TABLE PRODUCT_MASTER(PRODUCTNO varchar2(6) PRIMARY KEY,
    DESCRIPTION varchar2(15) NOT NULL, PROFITPERCENT number(4,2) NOT NULL,
    UNITMEASURE varchar2(10) NOT NULL, QTYONHAND number(8) NOT NULL,
    REORDERLVL number(8) NOT NULL, SELLPRICE number(8,2) NOT NULL,
    COSTPRICE number(8,2) NOT NULL,
    CONSTRAINT ck_product CHECK (PRODUCTNO like 'P%'),
    CONSTRAINT ck_sell CHECK (SELLPRICE <> 0),
    CONSTRAINT ck_cost CHECK (COSTPRICE <> 0));
```

c) **Table Name:** SALESMAN_MASTER

```
CREATE TABLE SALESMAN_MASTER(SALESMANNO varchar2(6) PRIMARY KEY,
    SALESMANNAME varchar2(20) NOT NULL, ADDRESS1 varchar2(30) NOT NULL,
    Address2 varchar2(30), CITY varchar2(20), PINCODE number(8), State varchar2(20),
    SALAMT number(8,2) NOT NULL, TGTTOGET number(6,2) NOT NULL,
    YTDSALES number(6,2) NOT NULL, REMARKS varchar2(60),
    CONSTRAINT ck_salesman CHECK (SALESMANNO like 'S%'),
    CONSTRAINT ck_sal CHECK (SALAMT <> 0),
    CONSTRAINT ck_target CHECK (TGTTOGET <> 0));
```

d) **Table Name:** SALES_ORDER

```
CREATE TABLE SALES_ORDER(ORDERNO varchar2(6) PRIMARY KEY,
    CLIENTNO varchar2(6) REFERENCES CLIENT_MASTER, ORDERDATE date,
    DELYADDR varchar2(25), SALESMANNO varchar2(6) REFERENCES SALESMAN_MASTER,
    DELYTYPE char(1) DEFAULT 'F', BILLEDYN char(1), DELYDATE date,
    ORDERSTATUS varchar2(10), CONSTRAINT ck_order CHECK (ORDERNO like 'O%'),
    CONSTRAINT ck_dely_type CHECK (DELYTYPE IN ('P', 'F')),
    CONSTRAINT ck_ord_status
        CHECK(ORDERSTATUS IN ('In Process', 'Fulfilled', 'Backorder', 'Cancelled')));
```

Table Name: SALES_ORDER_DETAILS

d)
```
CREATE TABLE SALES_ORDER_DETAILS(
    ORDERNO varchar2(6) REFERENCES SALES_ORDER,
    PRODUCTNO varchar2(6) REFERENCES PRODUCT_MASTER,
    QTYORDERED number(8), QTYDISP number(8), PRODUCTRATE number(10,2),
    PRIMARY KEY (ORDERNO, PRODUCTNO));
```

2. **SQL Statement for inserting data into their respective tables:**

a) Data for **CLIENT_MASTER** table:
```
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00001', 'Ivan Bayross', 'Mumbai', 400054, 'Maharashtra', 15000);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00002', 'Mamta Muzumdar', 'Madras'', 780001, 'Tamil Nadu', 0);
INSERT INTO Client_Master (ClientNo, Name, City, Pincode, State, BalDue)
    VALUES ('C00003', 'Chhaya Bankar', 'Mumbai', 400057, 'Maharashtra', 5000);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00004', 'Ashwini Joshi', 'Bangalore', 560001, 'Karnataka', 0);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00005', 'Hansel Colaco', 'Mumbai', 400060, 'Maharashtra', 2000);
INSERT INTO Client_Master (ClientNo, Name, City, PinCode, State, BalDue)
    VALUES ('C00006', 'Deepak Sharma', 'Mangalore', 560050, 'Karnataka', 0);
```

b) Data for **PRODUCT_MASTER** table
```
INSERT INTO Product_Master VALUES ('P00001', 'T-Shirts', 5, 'Piece', 200, 50, 350, 250);
INSERT INTO Product_Master VALUES ('P03453', 'Shirts', 6, 'Piece', 150, 50, 500, 350);
INSERT INTO Product_Master VALUES ('P06734', 'Cotton Jeans', 5, 'Piece', 100, 20, 600, 450);
INSERT INTO Product_Master VALUES ('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500);
INSERT INTO Product_Master VALUES ('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550);
INSERT INTO Product_Master VALUES ('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450);
INSERT INTO Product_Master VALUES ('P07965', 'Denim Shirts', 4, 'Piece', 100, 40, 350, 250);
INSERT INTO Product_Master VALUES ('P07975', 'Lycra Tops', 5, 'Piece', 70, 30, 300, 175);
INSERT INTO Product_Master VALUES ('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);
```

c) Data for **SALESMAN_MASTER** table
```
INSERT INTO Salesman_Master VALUES ('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002,
                                    'Maharashtra', 3000, 100, 50, 'Good');
INSERT INTO Salesman_Master VALUES ('S00002', 'Omkar', '65', 'Nariman', 'Mumbai', 400001,
                                    'Maharashtra', 3000, 200, 100, 'Good');
INSERT INTO Salesman_Master VALUES ('S00003', 'Raj', 'P-7', 'Bandra', 'Mumbai', 400032,
                                    'Maharashtra', 3000, 200, 100, 'Good');
INSERT INTO Salesman_Master VALUES ('S00004', 'Ashish', 'A/5', 'Juhu', 'Bombay', 400044,
                                    'Maharashtra', 3500, 200, 150, 'Good');
```

d) Data for **SALES_ORDER** table
```
                                                          BilledYn, SalesmanNo, DelyDate,
INSERT INTO Sales_Order (OrderNo, OrderDate, ClientNo, DelyType, ' N', 'S00001', '20-july-02', 'In Process');
    OrderStatus) VALUES('O19001', '12-june-02', 'C00001', 'F', BilledYn, SalesmanNo, DelyDate,
INSERT INTO Sales_Order (OrderNo, OrderDate, ClientNo, DelyType, ' N', 'S00002', '27-july-02', 'Cancelled');
    OrderStatus) VALUES('O19002', '25-june-02', 'C00002', 'P', BilledYn, SalesmanNo, DelyDate,
INSERT INTO Sales_Order (OrderNo, OrderDate, ClientNo, DelyType, ' Y', 'S00003', '20-feb-02', 'Fulfilled');
    OrderStatus) VALUES('O19003', '18-feb-02', 'C00003', 'F', BilledYn, SalesmanNo, DelyDate,
INSERT INTO Sales_Order (OrderNo, OrderDate, ClientNo, DelyType, 'Y', 'S00001', '07-apr-02', 'Fulfilled');
    OrderStatus) VALUES('O19003', '03-apr-02', 'C00001', 'F', BilledYn, SalesmanNo, DelyDate,
INSERT INTO Sales_Order (OrderNo, OrderDate, ClientNo, DelyType, 'N', 'S00002', '22-may-02', 'Cancelled');
    OrderStatus) VALUES('O46866', '20-may-02', 'C00004', 'P',
INSERT INTO Sales_Order (OrderNo, OrderDate, ClientNo, DelyType, BilledYn, SalesmanNo, DelyDate,
    OrderStatus) VALUES('O19008', '24-may-02', 'C00005', 'F', 'N', 'S00004', '26-july-96', 'In Process');
```

e) Data for **SALES_ORDER_DETAILS** table

```
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19001', 'P00001', 4, 4, 525);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19001', 'P07965', 2, 1, 8400);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19001', 'P07885', 2, 1, 5250);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19002', 'P00001', 10, 0, 525);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O46865', 'P07868', 3, 3, 3150);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O46865', 'P07885', 3, 1, 5250);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O46865', 'P00001', 10, 10, 525);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O46865', 'P03453', 4, 4, 1050);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19003', 'P03453', 2, 2, 1050);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19003', 'P06734', 1, 1, 12000);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O46866', 'P07965', 1, 0, 8400);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O46866', 'P07975', 1, 0, 1050);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19008', 'P00001', 10, 5, 525);
INSERT INTO Sales_Order_Details (OrderNo, ProductNo, QtyOrdered, QtyDisp, ProductRate)
    VALUES('O19008', 'P07975', 5, 3, 1050);
```

9. INTERACTIVE SQL PART - III

1. Generate SQL Statements to perform the following computations on table data:

a. Listing of the names of all clients having 'a' as the second letter in their names.
```
SELECT Name FROM Client_Master WHERE Name like '_a%';
```

b. Listing of clients who stay in a city whose first letter is 'M'.
```
SELECT ClientNo, Name FROM Client_Master WHERE City LIKE 'M%';
```

c. List all clients who stay in 'Bangalore' or 'Mangalore'
```
SELECT ClientNo, Name FROM Client_Master WHERE City IN('Bangalore', 'Mangalore');
```

d. List all clients whose BalDue is greater than value 10000.
```
SELECT ClientNo, Name FROM Client_Master WHERE Baldue > 10000;
```

e. Print the information from Sales_Order table for orders placed in the month of June.
```
SELECT * FROM Sales_Order WHERE TO_CHAR(OrderDate,'MON') = 'JUN';
```

f. Displaying the order information of ClientNo 'C00001' and 'C00002'.
```
SELECT * FROM Sales_Order WHERE ClientNo IN('C00001', 'C00002');
```

g. List products whose selling price is greater than 500 and less than or equal to 750.
```
SELECT ProductNo, Description FROM Product_Master WHERE SellPrice > 500 and SellPrice < 750;
```

h. Listing of products whose selling price is more than 500 with the new selling price calculated as original selling price plus 15%.

```
SELECT ProductNo, Description, SellPrice, SellPrice*15 new_price FROM Product_Master
    WHERE SellPrice > 500;
```

i. Listing of 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');
```

j. Count the total number of orders.

```
SELECT COUNT(OrderNo) 'No. Of Order' FROM Sales_Order;
```

k. Calculating the average price of all the products.

```
SELECT AVG(SellPrice) FROM Product_Master;
```

l. Determining the maximum and minimum price for the product prices.

```
SELECT MAX(SellPrice) max_price, MIN(SellPrice) min_price FROM Product_Master;
```

m. Count the number of products having price greater than or equal to 500.

```
SELECT COUNT(ProductNo) FROM Product_Master WHERE SellPrice <= 1500;
```

n. Find all the products whose QtyOnHand is less than reorder level.

```
SELECT ProductNo, Description FROM Product_Master WHERE QtyOnHand < ReorderLvl;
```

2. **SQL Statements for Date Manipulation:**

a. Display the order number and day on which clients placed their order.

```
SELECT OrderNo, TO_CHAR(OrderDate, 'day') FROM Sales_Order;
```

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

```
SELECT TO_CHAR(DelyDate, 'month'), DelyDate FROM Sales_Order
    ORDER BY TO_CHAR(DelyDate, 'month');
```

c. Display the OrderDate in the format 'DD-Month-YY'. E.g. 12-February-03

```
SELECT DATE_FORMAT(OrderDATE '%d-%M-%Y') FROM Sales_Order;
```

d. List the OrderDate in the format 'DD-Month-YY'. e.g. 12-February-02.

```
SELECT TO_CHAR(Orderdate, 'DD-Month-YY') FROM Sales_Order;
```

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

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

10. INTERACTIVE SQL PART - IV

1. **SQL statements for using Having and Group By Clauses:**

a. Printing the description and total quantity sold for each product.

```
SELECT description, SUM(QtyDisp) FROM Product_Master, Sales_Order_Details
    WHERE Product_Master.ProductNo = Sales_Order_Details.ProductNo
    GROUP BY Description;
```

b. Finding the value of each product sold.

```
SELECT Sales_Order_Details.ProductNo, Product_Master.Description,
    SUM(Sales_Order_Details.QtyDisp * Sales_Order_Details.ProductRate) 'Sales Per Product'
    FROM Sales_Order_Details, Product_Master
        WHERE Product_Master.ProductNo = Sales_Order_Details.ProductNo
        GROUP BY Sales_Order_Details.ProductNo, Product_Master.Description;
```

c. Calculating the average quantity sold for each client that has a maximum order value of 15000.00.

```
SELECT CM.ClientNo, CM.Name, AVG(SOD.QtyDisp) 'Avg. Sales'
    FROM Sales_Order_Details SOD, Sales_Order SO, Client_Master CM
        WHERE CM.ClientNo = SO.ClientNo AND SO.OrderNo = SOD.OrderNo
    GROUP BY CM.ClientNo, Name HAVING MAX(SOD.QtyOrdered * SOD.ProductRate) > 15000;
```

d. Finding out the total of all the billed orders for the month of June.

```
SELECT SO.OrderNo, SO.OrderDate, SUM(SOD.QtyOrdered * SOD.ProductRate) 'Order Billed'
    FROM Sales_Order SO, Sales_Order_Details SOD WHERE SOD.OrderNo = SO.OrderNo
    AND SO.Billed = 'Y' AND to_char(OrderDate, 'MON') = 'Jun' GROUP BY SO.OrderNo;
```

2. Exercises on Joins and Correlation:

a. Find out the products, which have been sold to 'Ivan Bayross'.

```
SELECT SOD.ProductNo, PM.Description
    FROM Sales_Order_Details SOD, Sales_Order SO, Product_Master PM, Client_Master CM
        WHERE PM.ProductNo = SOD.ProductNo AND SO.OrderNo = SOD.OrderNo
            AND CM.ClientNo = SO.ClientNo AND CM.Name = 'Ivan Bayross';
```

b. Finding out the products and their quantities that will have to be delivered in the current month.

```
SELECT SOD.ProductNo, PM.Description, SUM(SOD.QtyOrdered)
    FROM Sales_Order_Details SOD, Sales_Order SO, Product_Master PM
        WHERE PM.ProductNo = SOD.ProductNo AND SO.OrderNo = SOD.OrderNo
            AND TO_CHAR(DelyDate, 'MON-YY') = TO_CHAR(SYSDATE, 'MON-YY')
        GROUP BY SOD.ProductNo, PM.Description;
```

c. Listing the ProductNo and description of constantly sold (i.e. rapidly moving) products.

```
SELECT DISTINCT Product_Master.ProductNo, Description
    FROM Sales_Order_Details, Product_Master
        WHERE Product_Master.ProductNo =Sales_Order_Details.ProductNo;
```

d. Finding the names of clients who have purchased 'Trousers'.

```
SELECT DISTINCT Sales_Order.ClientNo, Client_Master.Name
    FROM Sales_Order_Details, Sales_Order, Product_Master, Client_Master
        WHERE Product_Master.ProductNo = Sales_Order_Details.ProductNo
            AND Sales_Order.OrderNo = Sales_Order_Details.OrderNo
            AND Client_Master.ClientNo = Sales_Order.ClientNo
            AND Description = 'Trousers';
```

e. Listing the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.

```
SELECT Sales_Order_Details.ProductNo, Sales_Order_Details.OrderNo
    FROM Sales_Order_Details, Sales_Order, Product_Master
        WHERE Sales_Order.OrderNo = Sales_Order_Details.OrderNo
        AND Product_Master.ProductNo = Sales_Order_Details.ProductNo
        AND Sales_Order_Details.QtyOrdered < 5 AND Product_Master.Description = 'Pull Overs';
```

f. Finding the products and their quantities for the orders placed by 'Ivan Bayross' and 'Mamta Muzumdar'.

```
SELECT SOD.ProductNo, PM.Description, SUM(QtyOrdered) 'Units Ordered'
    FROM Sales_Order_Details SOD, Sales_Order SO, Product_Master PM, Client_Master CM
        WHERE SO.OrderNo = SOD.OrderNo AND PM.ProductNo = SOD.ProductNo
            AND CM.ClientNo = SO.ClientNo
            AND (CM.Name = 'Ivan Bayross' OR CM.Name = 'Mamta Muzumdar')
        GROUP BY SOD.ProductNo, PM.Description;
```

g. Finding the products and their quantities for the orders placed by ClientNo 'C00001' and 'C00002'.

```
SELECT SO.ClientNo, SOD.ProductNo, PM.Description, SUM(QtyOrdered) 'Units Ordered'
    FROM Sales_Order SO, Sales_Order_Details SOD, Product_Master PM, Client_Master CM
        WHERE SO.OrderNo = SOD.OrderNo AND SOD.ProductNo = PM.ProductNo
            AND SO.ClientNo = CM.ClientNo
        GROUP BY SO.ClientNo, SOD.ProductNo, PM.Description
            HAVING SO.ClientNo = 'C00001' OR SO.ClientNo='C00002';
```

3. SQL statements for exercises on Sub-queries:

a. Finding the non-moving products i.e. products not being sold.

```
SELECT ProductNo, Description FROM Product_Master
    WHERE ProductNo NOT IN(SELECT ProductNo FROM Sales_Order_Details);
```

b. Finding the name and complete address for the customer who has placed Order number 'O19001'.

```
SELECT Name ,Address1, Address2, City, State, PinCode FROM Client_Master
    WHERE ClientNo IN(SELECT ClientNo FROM Sales_Order WHERE OrderNo = 'O19001');
```

c. Finding the clients who have placed orders before the month of May'02.

```
SELECT ClientNo, Name FROM Client_Master WHERE ClientNo IN(SELECT ClientNo
    FROM Sales_Order WHERE TO_CHAR(OrderDate, 'MON,YY') < 'MAY,02');
```

d. Find out if the product 'Lycra Tops' has been ordered by any client and print the ClientNo, Name to whom it was sold.

```
SELECT ClientNo, Name FROM Client_Master WHERE ClientNo
    IN(SELECT ClientNo FROM Sales_Order WHERE OrderNo IN(SELECT OrderNo
        FROM Sales_Order_Details WHERE ProductNo IN(SELECT ProductNo
            FROM Product_Master WHERE Description = 'Lycra Tops')));
```

e. Find 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
        WHERE (QtyOrdered * ProductRate) >= 10000));
```