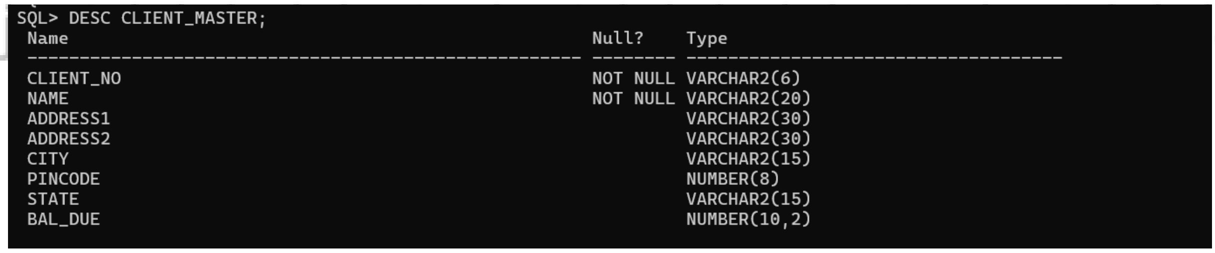
ORACLE LAB ASSIGNMENT-4

- CREATE FOLLOWING TABLES AND INSERT RECORDS IN THE TABLES.

A)TABLE NAME : CLIENT_MASTER

DESCRIPTION : USED TO STORE CLIENT INFORMATION

```
CREATE TABLE CLIENT_MASTER (
   CLIENT_NO VARCHAR2(6) PRIMARY KEY CHECK (CLIENT_NO LIKE 'C%'),
   NAME VARCHAR2(20) NOT NULL,
   ADDRESS1 VARCHAR2(30),
   ADDRESS2 VARCHAR2(30),
   CITY VARCHAR2(15),
   PINCODE NUMBER(8),
   STATE VARCHAR2(15),
   BAL_DUE NUMBER(10, 2)
);
```

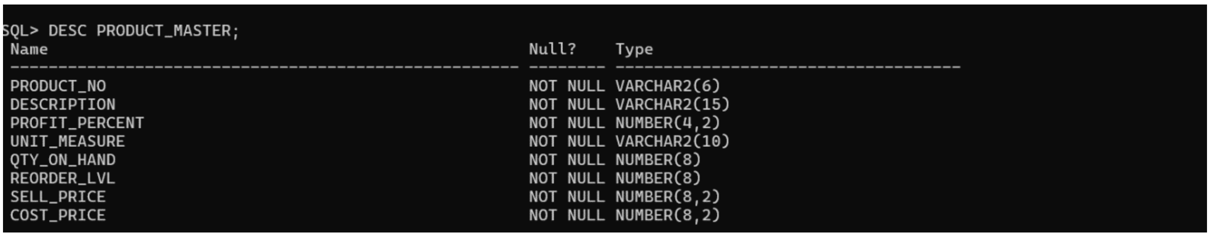


```
SQL> DESC CLIENT_MASTER;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 CLIENT_NO                                             NOT NULL VARCHAR2(6)
 NAME                                                  NOT NULL VARCHAR2(20)
 ADDRESS1                                                       VARCHAR2(30)
 ADDRESS2                                                       VARCHAR2(30)
 CITY                                                           VARCHAR2(15)
 PINCODE                                                        NUMBER(8)
 STATE                                                          VARCHAR2(15)
 BAL_DUE                                                        NUMBER(10,2)
```

B) TABLE NAME : PRODUCT_MASTER

DESCRIPTION : USED TO STORE PRODUCT INFORMATION.

```
CREATE TABLE PRODUCT_MASTER (
   PRODUCT_NO VARCHAR2(6) PRIMARY KEY CHECK (PRODUCT_NO LIKE 'P%'),
   DESCRIPTION VARCHAR2(15) NOT NULL,
   PROFIT_PERCENT NUMBER(4, 2) NOT NULL,
   UNIT_MEASURE VARCHAR2(10) NOT NULL,
   QTY_ON_HAND NUMBER(8) NOT NULL,
   REORDER_LVL 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)
);
```

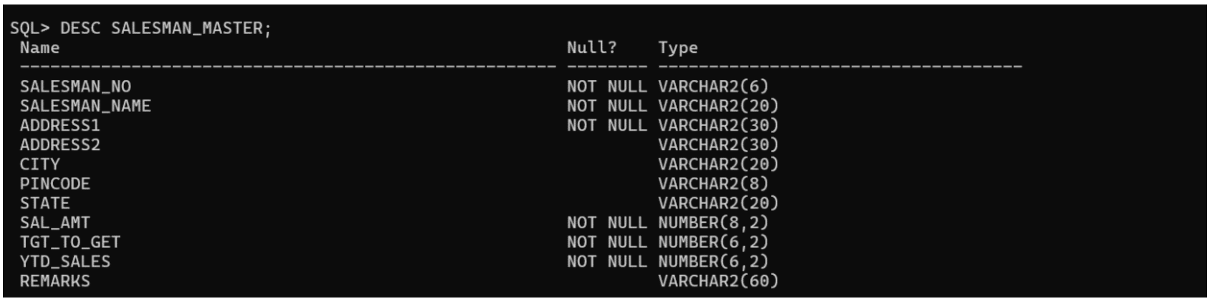


```
SQL> DESC PRODUCT_MASTER;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 PRODUCT_NO                                            NOT NULL VARCHAR2(6)
 DESCRIPTION                                           NOT NULL VARCHAR2(15)
 PROFIT_PERCENT                                        NOT NULL NUMBER(4,2)
 UNIT_MEASURE                                          NOT NULL VARCHAR2(10)
 QTY_ON_HAND                                           NOT NULL NUMBER(8)
 REORDER_LVL                                           NOT NULL NUMBER(8)
 SELL_PRICE                                            NOT NULL NUMBER(8,2)
 COST_PRICE                                            NOT NULL NUMBER(8,2)
```

C) TABLE NAME : SALESMAN_MASTER

DESCRIPTION : USED TO STORE SALESMAN WORKING FOR THE COMPANY.

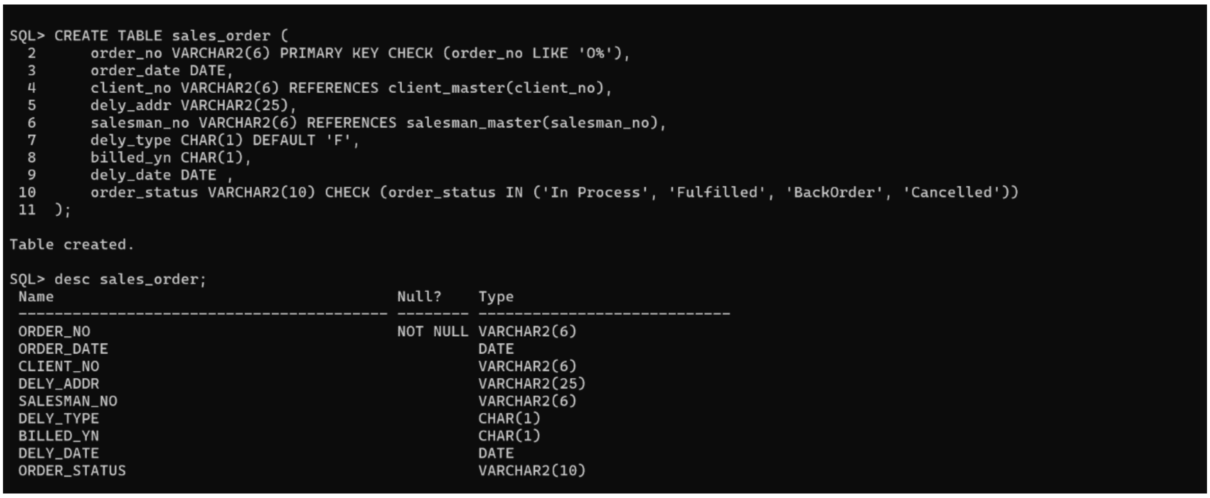
```
CREATE TABLE SALESMAN_MASTER (
   SALESMAN_NO VARCHAR2(6) PRIMARY KEY CHECK (SALESMAN_NO LIKE 'S%'),
   SALESMAN_NAME VARCHAR2(20) NOT NULL,
   ADDRESS1 VARCHAR2(30) NOT NULL,
   ADDRESS2 VARCHAR2(30),
   CITY VARCHAR2(20),
   PINCODE VARCHAR2(8),
   STATE VARCHAR2(20),
   SAL_AMT NUMBER(8, 2) NOT NULL CHECK (SAL_AMT > 0),
   TGT_TO_GET NUMBER(6, 2) NOT NULL CHECK ( TGT_TO_GET > 0),
   YTD_SALES NUMBER(6, 2) NOT NULL,
   REMARKS VARCHAR2(60)
);
```



```
SQL> DESC SALESMAN_MASTER;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 SALESMAN_NO                                           NOT NULL VARCHAR2(6)
 SALESMAN_NAME                                         NOT NULL VARCHAR2(20)
 ADDRESS1                                              NOT NULL VARCHAR2(30)
 ADDRESS2                                                       VARCHAR2(30)
 CITY                                                           VARCHAR2(20)
 PINCODE                                                        VARCHAR2(8)
 STATE                                                          VARCHAR2(20)
 SAL_AMT                                               NOT NULL NUMBER(8,2)
 TGT_TO_GET                                            NOT NULL NUMBER(6,2)
 YTD_SALES                                             NOT NULL NUMBER(6,2)
 REMARKS                                                        VARCHAR2(60)
```

D) TABLE NAME : SALES_ORDER

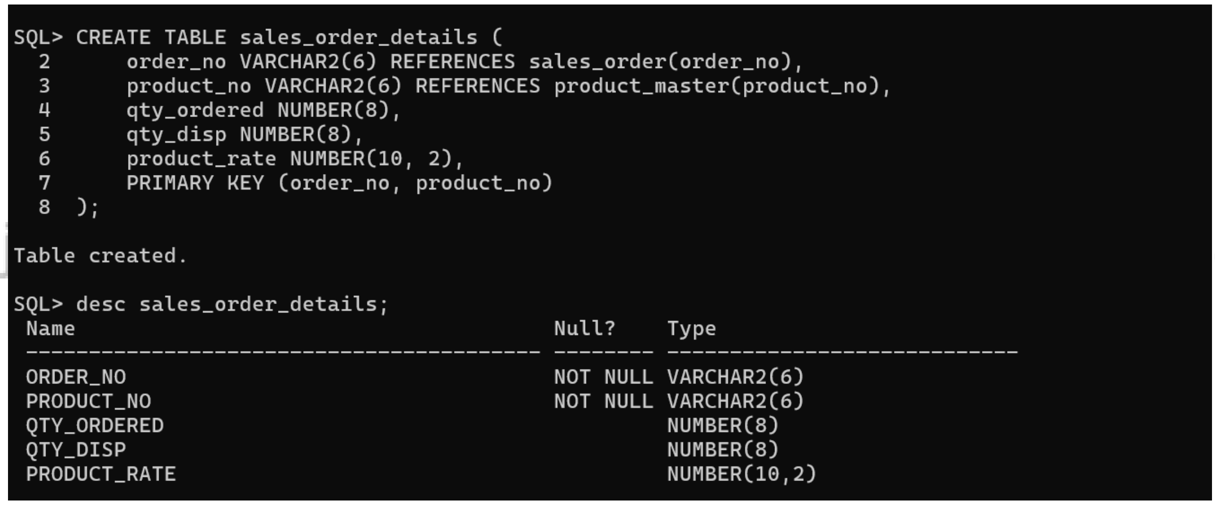
DESCRIPTION : USED TO STORE CLIENT’S ORDERS.



```
SQL> CREATE TABLE sales_order (
  2      order_no VARCHAR2(6) PRIMARY KEY CHECK (order_no LIKE 'O%'),
  3      order_date DATE,
  4      client_no VARCHAR2(6) REFERENCES client_master(client_no),
  5      dely_addr VARCHAR2(25),
  6      salesman_no VARCHAR2(6) REFERENCES salesman_master(salesman_no),
  7      dely_type CHAR(1) DEFAULT 'F',
  8      billed_yn CHAR(1),
  9      dely_date DATE ,
 10      order_status VARCHAR2(10) CHECK (order_status IN ('In Process', 'Fulfilled', 'BackOrder', 'Cancelled'))
 11  );

Table created.

SQL> desc sales_order;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_NO                                  NOT NULL VARCHAR2(6)
 ORDER_DATE                                         DATE
 CLIENT_NO                                          VARCHAR2(6)
 DELY_ADDR                                          VARCHAR2(25)
 SALESMAN_NO                                        VARCHAR2(6)
 DELY_TYPE                                          CHAR(1)
 BILLED_YN                                          CHAR(1)
 DELY_DATE                                          DATE
 ORDER_STATUS                                       VARCHAR2(10)
```

E) TABLE NAME : SALES_ORDER_DETAILS

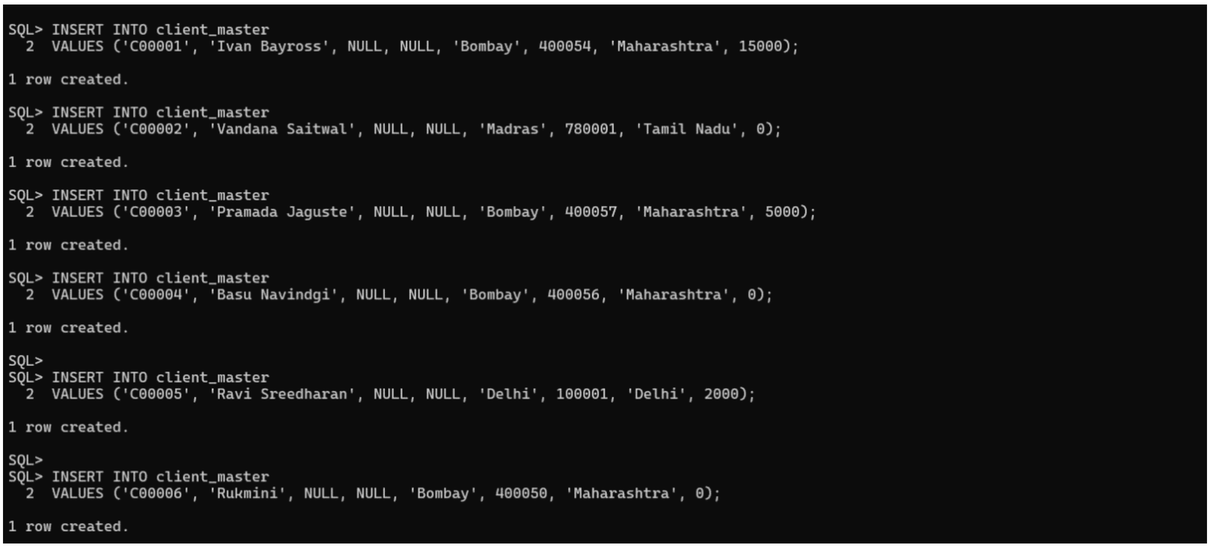
DESCRIPTION : USED TO STORE CLIENT’S ORDERS WITH DETAILS OF EACH PRODUCT ORDERED.



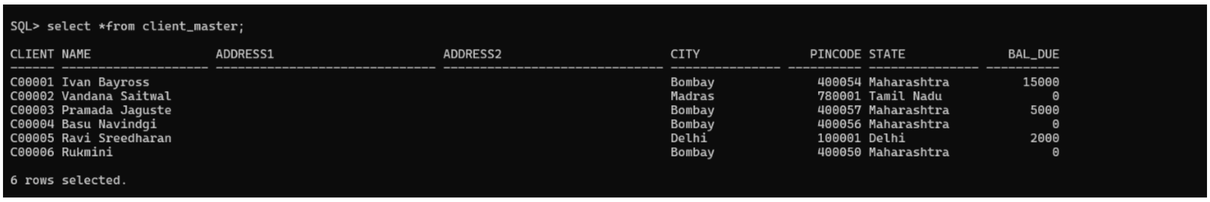
```
SQL> CREATE TABLE sales_order_details (
  2      order_no VARCHAR2(6) REFERENCES sales_order(order_no),
  3      product_no VARCHAR2(6) REFERENCES product_master(product_no),
  4      qty_ordered NUMBER(8),
  5      qty_disp NUMBER(8),
  6      product_rate NUMBER(10, 2),
  7      PRIMARY KEY (order_no, product_no)
  8  );

Table created.

SQL> desc sales_order_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_NO                                  NOT NULL VARCHAR2(6)
 PRODUCT_NO                                NOT NULL VARCHAR2(6)
 QTY_ORDERED                                        NUMBER(8)
 QTY_DISP                                           NUMBER(8)
 PRODUCT_RATE                                       NUMBER(10,2)
```

2) INSERT THE FOLLOWING DATA INTO THEIR RESPECTIVE TABLES:

A) DATA FOR CLIENT_MASTER TABLE:

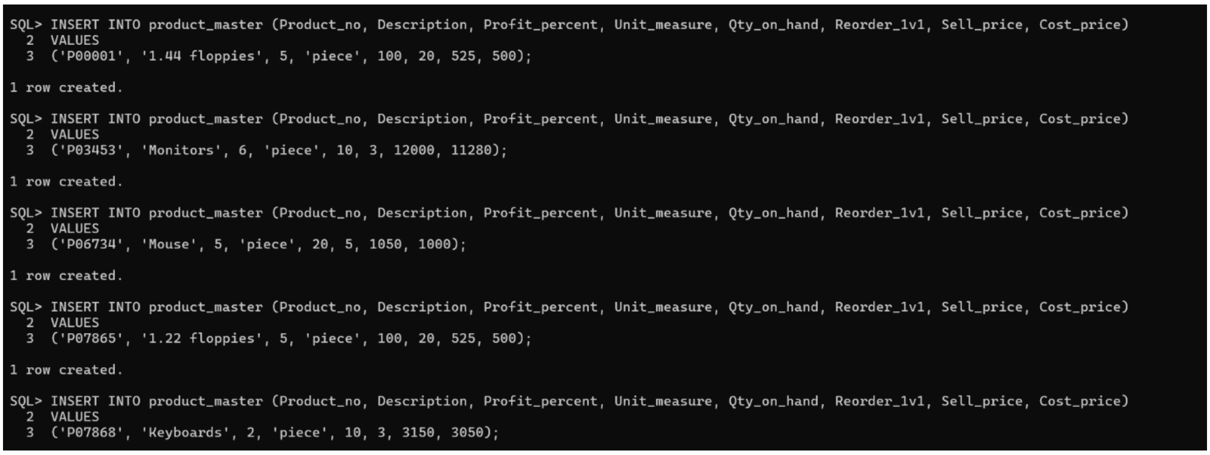


```
SQL> INSERT INTO client_master
  2  VALUES ('C00001', 'Ivan Bayross', NULL, NULL, 'Bombay', 400054, 'Maharashtra', 15000);

1 row created.

SQL> INSERT INTO client_master
  2  VALUES ('C00002', 'Vandana Saitwal', NULL, NULL, 'Madras', 780001, 'Tamil Nadu', 0);

1 row created.

SQL> INSERT INTO client_master
  2  VALUES ('C00003', 'Pramada Jaguste', NULL, NULL, 'Bombay', 400057, 'Maharashtra', 5000);

1 row created.

SQL> INSERT INTO client_master
  2  VALUES ('C00004', 'Basu Navindgi', NULL, NULL, 'Bombay', 400056, 'Maharashtra', 0);

1 row created.

SQL>
SQL> INSERT INTO client_master
  2  VALUES ('C00005', 'Ravi Sreedharan', NULL, NULL, 'Delhi', 100001, 'Delhi', 2000);

1 row created.

SQL>
SQL> INSERT INTO client_master
  2  VALUES ('C00006', 'Rukmini', NULL, NULL, 'Bombay', 400050, 'Maharashtra', 0);

1 row created.
```

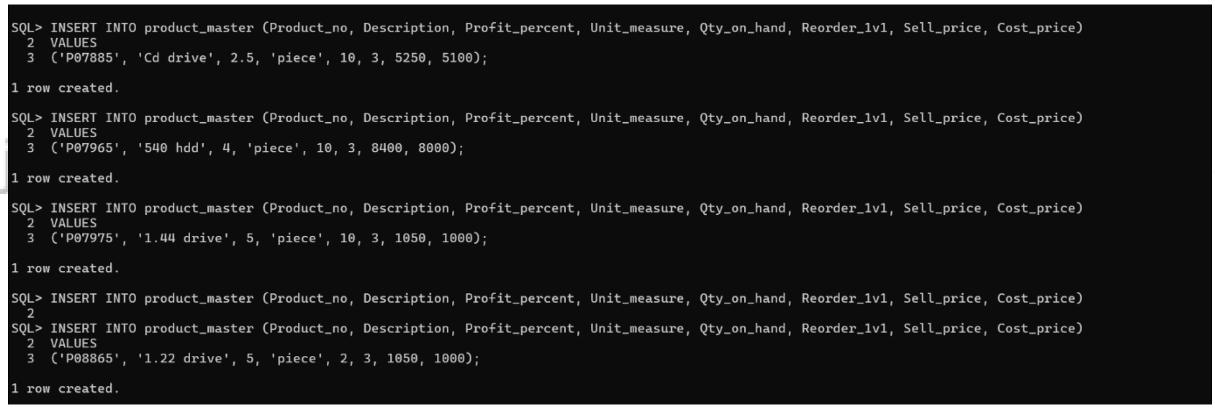


```
SQL> select *from client_master;

CLIENT NAME                 ADDRESS1                       ADDRESS2                       CITY               PINCODE STATE                BAL_DUE
------ -------------------- ------------------------------ ------------------------------ --------------- ---------- --------------- ----------
C00001 Ivan Bayross                                                                       Bombay              400054 Maharashtra          15000
C00002 Vandana Saitwal                                                                    Madras              780001 Tamil Nadu               0
C00003 Pramada Jaguste                                                                    Bombay              400057 Maharashtra           5000
C00004 Basu Navindgi                                                                      Bombay              400056 Maharashtra              0
C00005 Ravi Sreedharan                                                                    Delhi               100001 Delhi                 2000
C00006 Rukmini                                                                            Bombay              400050 Maharashtra              0

6 rows selected.
```

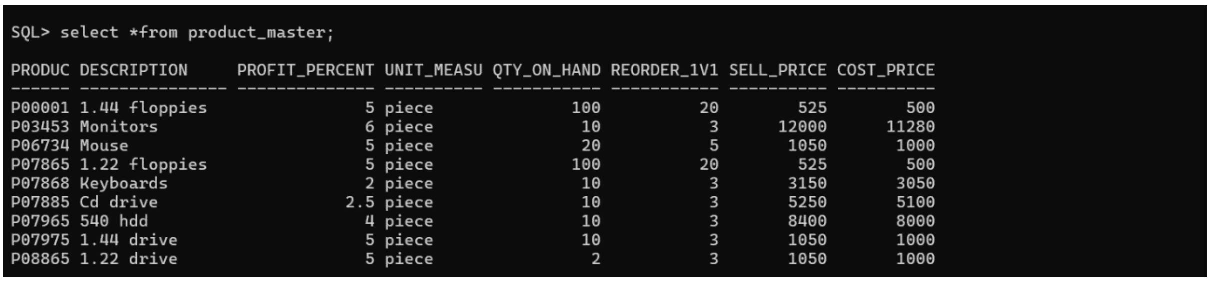
B) DATA FOR PRODUCT_MASTER TABLE:



```
SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P00001', '1.44 floppies', 5, 'piece', 100, 20, 525, 500);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P03453', 'Monitors', 6, 'piece', 10, 3, 12000, 11280);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P06734', 'Mouse', 5, 'piece', 20, 5, 1050, 1000);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P07865', '1.22 floppies', 5, 'piece', 100, 20, 525, 500);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P07868', 'Keyboards', 2, 'piece', 10, 3, 3150, 3050);
```

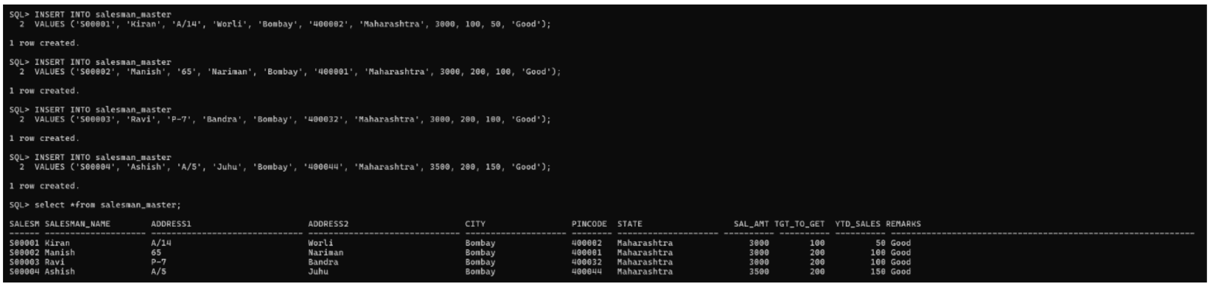


```
SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P07885', 'Cd drive', 2.5, 'piece', 10, 3, 5250, 5100);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P07965', '540 hdd', 4, 'piece', 10, 3, 8400, 8000);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P07975', '1.44 drive', 5, 'piece', 10, 3, 1050, 1000);

1 row created.

SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2
SQL> INSERT INTO product_master (Product_no, Description, Profit_percent, Unit_measure, Qty_on_hand, Reorder_1v1, Sell_price, Cost_price)
  2  VALUES
  3  ('P08865', '1.22 drive', 5, 'piece', 2, 3, 1050, 1000);

1 row created.
```



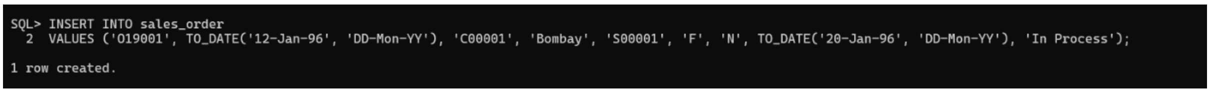
```
SQL> select *from product_master;

PRODUC DESCRIPTION     PROFIT_PERCENT UNIT_MEASU QTY_ON_HAND REORDER_1V1 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 Keyboards                    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       1000
P08865 1.22 drive                   5 piece                2           3       1050       1000
```

C) DATA FOR SALESMAN_MASTER TABLE:

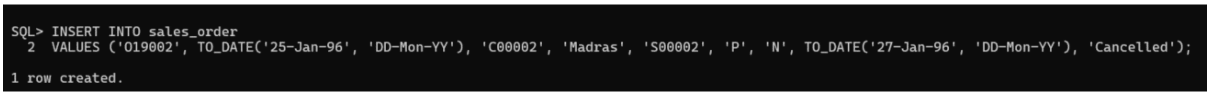


```
SQL> INSERT INTO salesman_master
  2  VALUES ('S00001', 'Kiran', 'A/14', 'Worli', 'Bombay', '400002', 'Maharashtra', 3000, 100, 50, 'Good');

1 row created.

SQL> INSERT INTO salesman_master
  2  VALUES ('S00002', 'Manish', '65', 'Nariman', 'Bombay', '400001', 'Maharashtra', 3000, 200, 100, 'Good');

1 row created.

SQL> INSERT INTO salesman_master
  2  VALUES ('S00003', 'Ravi', 'P-7', 'Bandra', 'Bombay', '400032', 'Maharashtra', 3000, 200, 100, 'Good');

1 row created.

SQL> INSERT INTO salesman_master
  2  VALUES ('S00004', 'Ashish', 'A/5', 'Juhu', 'Bombay', '400044', 'Maharashtra', 3500, 200, 150, 'Good');

1 row created.

SQL> select *from salesman_master;

SALESM SALESMAN_NAME        ADDRESS1                       ADDRESS2                       CITY                 PINCODE  STATE                   SAL_AMT TGT_TO_GET  YTD_SALES REMARKS
------ -------------------- ------------------------------ ------------------------------ -------------------- -------- -------------------- ---------- ---------- ---------- ------------------------------------------------------------
S00001 Kiran                A/14                           Worli                          Bombay               400002   Maharashtra                3000        100         50 Good
S00002 Manish               65                             Nariman                        Bombay               400001   Maharashtra                3000        200        100 Good
S00003 Ravi                 P-7                            Bandra                         Bombay               400032   Maharashtra                3000        200        100 Good
S00004 Ashish               A/5                            Juhu                           Bombay               400044   Maharashtra                3500        200        150 Good
```

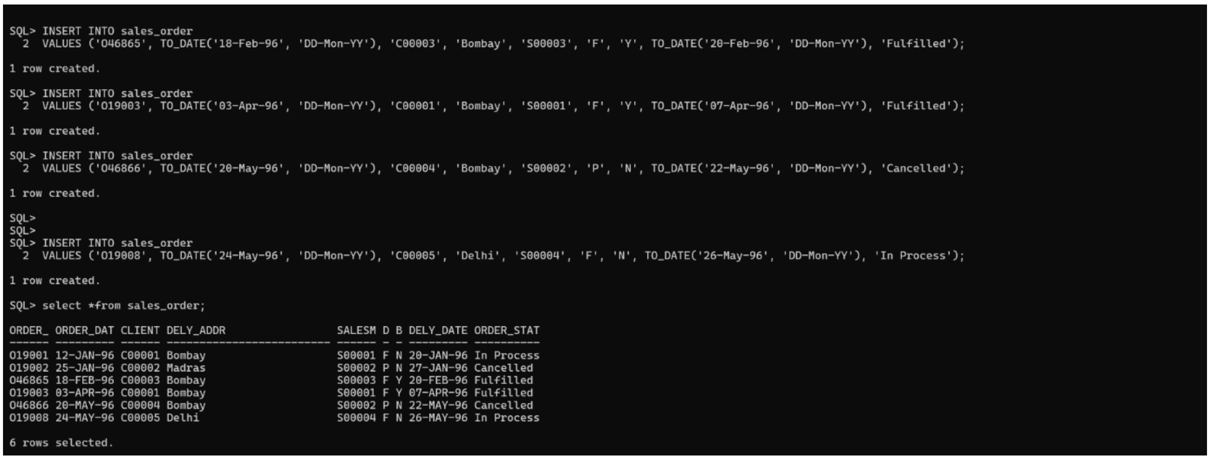
D) DATA FOR SALES_ORDER TABLE



```
SQL> INSERT INTO sales_order
  2  VALUES ('O19001', TO_DATE('12-Jan-96', 'DD-Mon-YY'), 'C00001', 'Bombay', 'S00001', 'F', 'N', TO_DATE('20-Jan-96', 'DD-Mon-YY'), 'In Process');

1 row created.
```

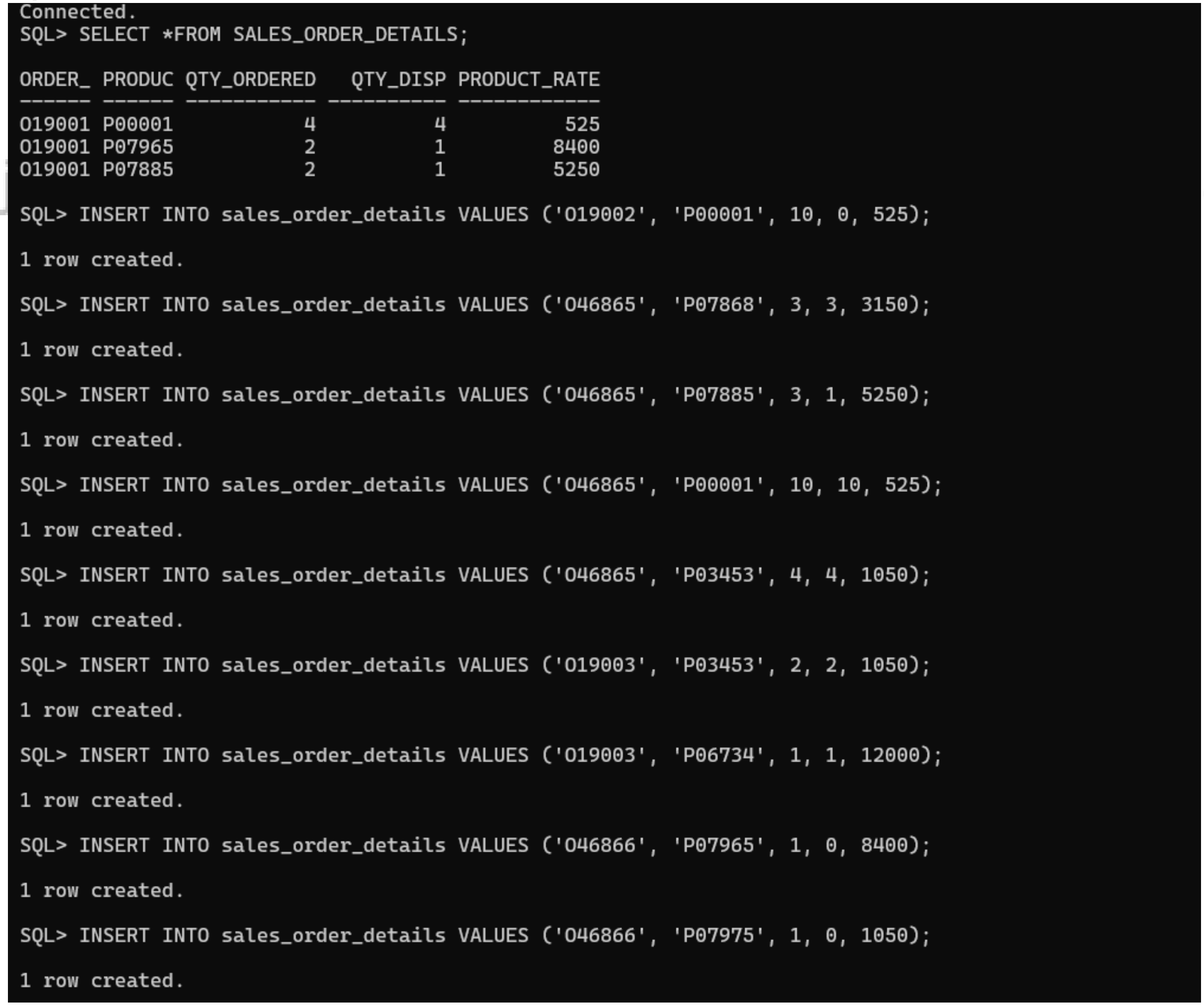


```
SQL> INSERT INTO sales_order
  2  VALUES ('O19002', TO_DATE('25-Jan-96', 'DD-Mon-YY'), 'C00002', 'Madras', 'S00002', 'P', 'N', TO_DATE('27-Jan-96', 'DD-Mon-YY'), 'Cancelled');

1 row created.
```

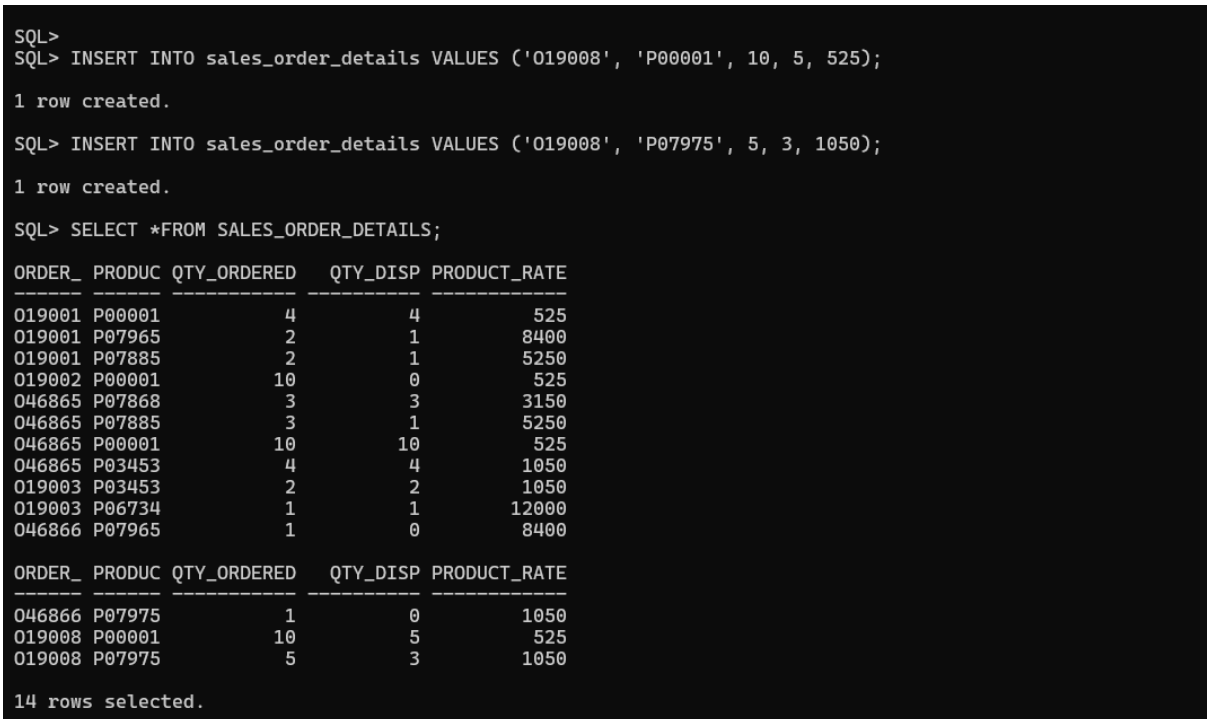


```
SQL> INSERT INTO sales_order
  2  VALUES ('O46865', TO_DATE('18-Feb-96', 'DD-Mon-YY'), 'C00003', 'Bombay', 'S00003', 'F', 'Y', TO_DATE('20-Feb-96', 'DD-Mon-YY'), 'Fulfilled');

1 row created.

SQL> INSERT INTO sales_order
  2  VALUES ('O19003', TO_DATE('03-Apr-96', 'DD-Mon-YY'), 'C00001', 'Bombay', 'S00001', 'F', 'Y', TO_DATE('07-Apr-96', 'DD-Mon-YY'), 'Fulfilled');

1 row created.

SQL> INSERT INTO sales_order
  2  VALUES ('O46866', TO_DATE('20-May-96', 'DD-Mon-YY'), 'C00004', 'Bombay', 'S00002', 'P', 'N', TO_DATE('22-May-96', 'DD-Mon-YY'), 'Cancelled');

1 row created.

SQL>
SQL>
SQL> INSERT INTO sales_order
  2  VALUES ('O19008', TO_DATE('24-May-96', 'DD-Mon-YY'), 'C00005', 'Delhi', 'S00004', 'F', 'N', TO_DATE('26-May-96', 'DD-Mon-YY'), 'In Process');

1 row created.

SQL> select *from sales_order;

ORDER_ ORDER_DAT CLIENT DELY_ADDR                SALESM D B DELY_DATE ORDER_STAT
------ --------- ------ ------------------------ ------ - - --------- ----------
O19001 12-JAN-96 C00001 Bombay                   S00001 F N 20-JAN-96 In Process
O19002 25-JAN-96 C00002 Madras                   S00002 P N 27-JAN-96 Cancelled
O46865 18-FEB-96 C00003 Bombay                   S00003 F Y 20-FEB-96 Fulfilled
O19003 03-APR-96 C00001 Bombay                   S00001 F Y 07-APR-96 Fulfilled
O46866 20-MAY-96 C00004 Bombay                   S00002 P N 22-MAY-96 Cancelled
O19008 24-MAY-96 C00005 Delhi                    S00004 F N 26-MAY-96 In Process

6 rows selected.
```

E) DATA FOR SALES_ORDER_DETAILS TABLE:



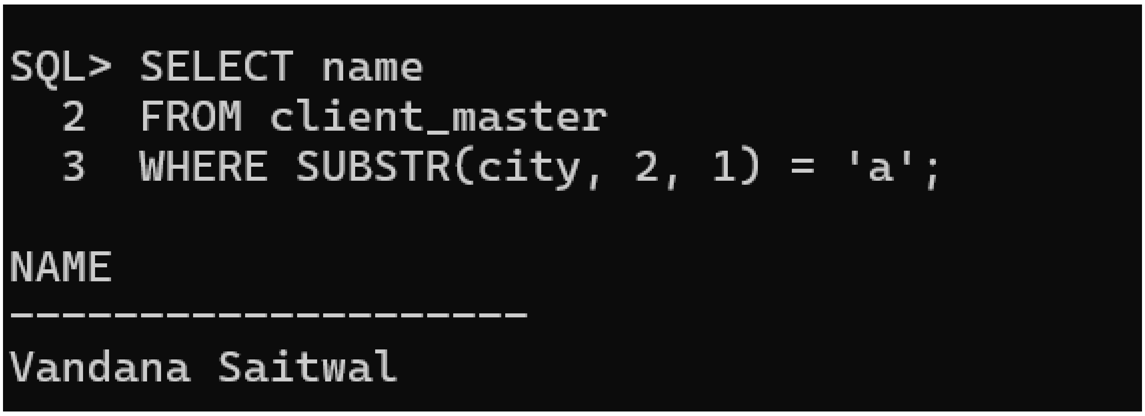
```
Connected.
SQL> SELECT *FROM SALES_ORDER_DETAILS;

ORDER_ PRODUC QTY_ORDERED   QTY_DISP PRODUCT_RATE
------ ------ ----------- ---------- ------------
O19001 P00001           4          4          525
O19001 P07965           2          1         8400
O19001 P07885           2          1         5250

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

1 row created.

SQL> INSERT INTO sales_order_details VALUES ('O46865', 'P07868', 3, 3, 3150);

1 row created.

SQL> INSERT INTO sales_order_details VALUES ('O46865', 'P07885', 3, 1, 5250);

1 row created.

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

1 row created.

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

1 row created.

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

1 row created.

SQL> INSERT INTO sales_order_details VALUES ('O19003', 'P06734', 1, 1, 12000);

1 row created.

SQL> INSERT INTO sales_order_details VALUES ('O46866', 'P07965', 1, 0, 8400);

1 row created.

SQL> INSERT INTO sales_order_details VALUES ('O46866', 'P07975', 1, 0, 1050);

1 row created.
```



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

1 row created.

SQL> INSERT INTO sales_order_details VALUES ('O19008', 'P07975', 5, 3, 1050);

1 row created.

SQL> SELECT *FROM SALES_ORDER_DETAILS;

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

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

14 rows selected.
```

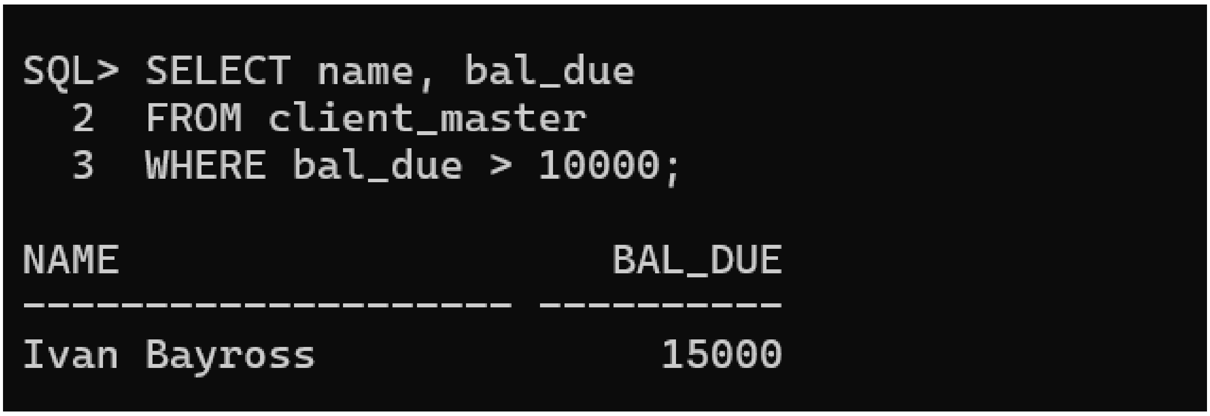
Q.1 QUERIES BASED ON THOSE TABLES COMPUTATION ON TABLE DATA, DATE MANIPULATION, HAVING AND GROUP BY, JOINS AND SUBQUERIES.

1) FIND OUT THE NAMES OF CLIENTS WHO STAYS IN A CITY WHOSE SECOND LETTER IS ‘A’.



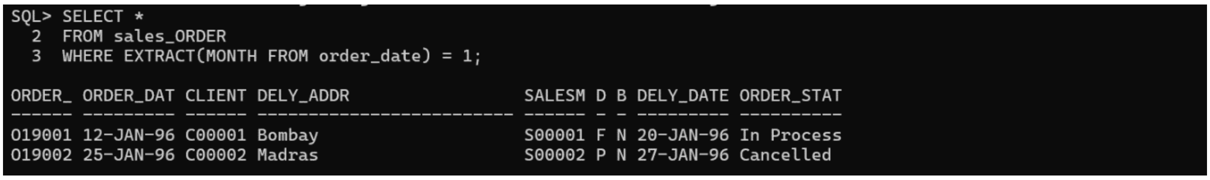
```
SQL> SELECT name
  2  FROM client_master
  3  WHERE SUBSTR(city, 2, 1) = 'a';

NAME
--------------------
Vandana Saitwal
```

2) PRINT THE LIST OF CLIENTS WHOSE BAL_DUE IS GREATER THAN VALUE 10000.



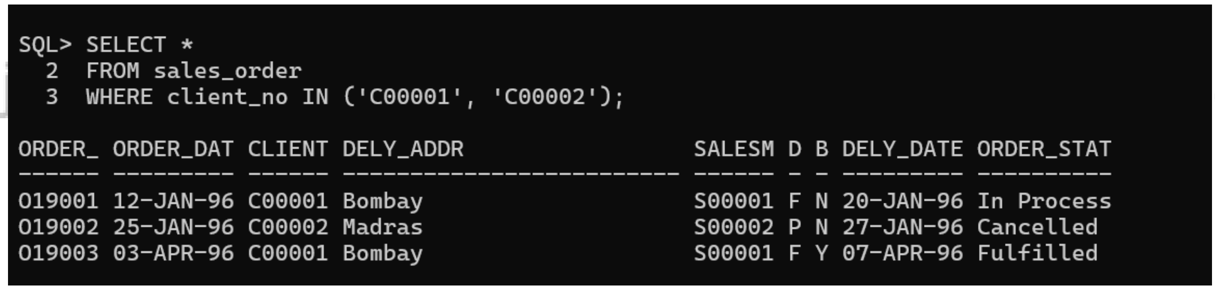
```
SQL> SELECT name, bal_due
  2  FROM client_master
  3  WHERE bal_due > 10000;

NAME                    BAL_DUE
-------------------- ----------
Ivan Bayross              15000
```

3) PRINT THE INFORMATION FROM SALES_ORDER TABLE FOR ORDERS PLACED IN THE MONTH OF JANUARY.



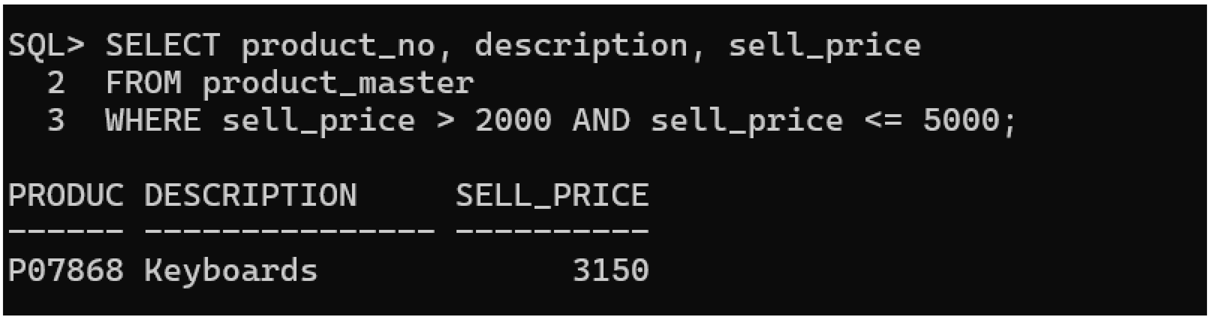
```
SQL> SELECT *
  2  FROM sales_ORDER
  3  WHERE EXTRACT(MONTH FROM order_date) = 1;

ORDER_ ORDER_DAT CLIENT DELY_ADDR                 SALESM D B DELY_DATE ORDER_STAT
------ --------- ------ ------------------------- ------ - - --------- ----------
O19001 12-JAN-96 C00001 Bombay                    S00001 F N 20-JAN-96 In Process
O19002 25-JAN-96 C00002 Madras                    S00002 P N 27-JAN-96 Cancelled
```

4) DISPLAY THE ORDER INFORMATION FOR CLIENT_NO ‘C00001’ AND ‘C00002’.



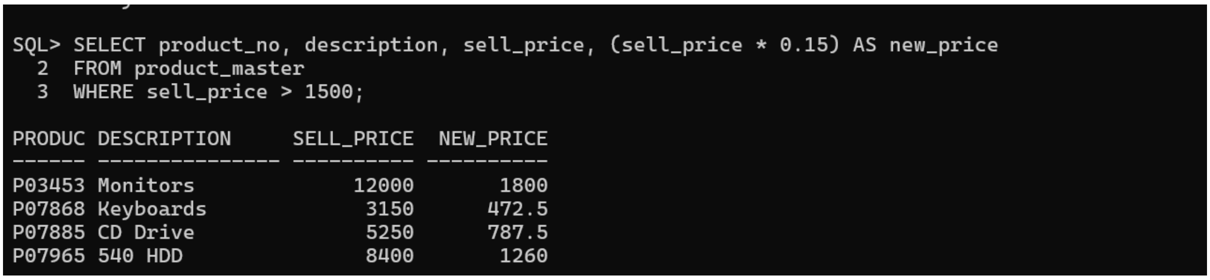
```
SQL> SELECT *
  2  FROM sales_order
  3  WHERE client_no IN ('C00001', 'C00002');

ORDER_ ORDER_DAT CLIENT DELY_ADDR                 SALESM D B DELY_DATE ORDER_STAT
------ --------- ------ ------------------------- ------ - - --------- ----------
O19001 12-JAN-96 C00001 Bombay                    S00001 F N 20-JAN-96 In Process
O19002 25-JAN-96 C00002 Madras                    S00002 P N 27-JAN-96 Cancelled
O19003 03-APR-96 C00001 Bombay                    S00001 F Y 07-APR-96 Fulfilled
```

5) FIND PRODUCTS WHOSE SELLING PRICE IS GREATER THAN 2000 AND LESS THAN OR EQAL TO 5000.



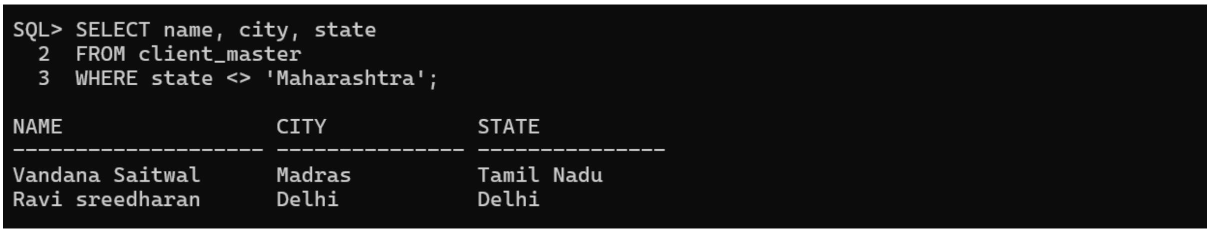
```
SQL> SELECT product_no, description, sell_price
  2  FROM product_master
  3  WHERE sell_price > 2000 AND sell_price <= 5000;

PRODUC DESCRIPTION     SELL_PRICE
------ --------------- ----------
P07868 Keyboards             3150
```

6) FIND PRODUCTS WHOSE SELLING PRICE IS MORE THAN 1500. CALCULATE A NEW SELLING PRICE AS, ORIGINAL SELLING PRICE *.15. RENAME THE NEW COLUMN IN THE ABOVE QUERY AS NEW_PRICE.



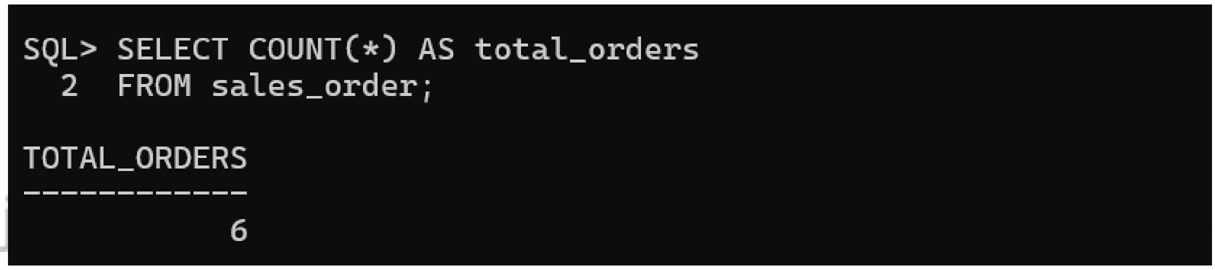
```
SQL> SELECT product_no, description, sell_price, (sell_price * 0.15) AS new_price
  2  FROM product_master
  3  WHERE sell_price > 1500;

PRODUC DESCRIPTION      SELL_PRICE  NEW_PRICE
------ --------------- ---------- ----------
P03453 Monitors             12000       1800
P07868 Keyboards             3150      472.5
P07885 CD Drive              5250      787.5
P07965 540 HDD               8400       1260
```

7) LIST THE NAMES, CITY AND STATE OF CLIENTS WHO ARE NOT IN THE STATE OF ‘MAHARASHTRA’.



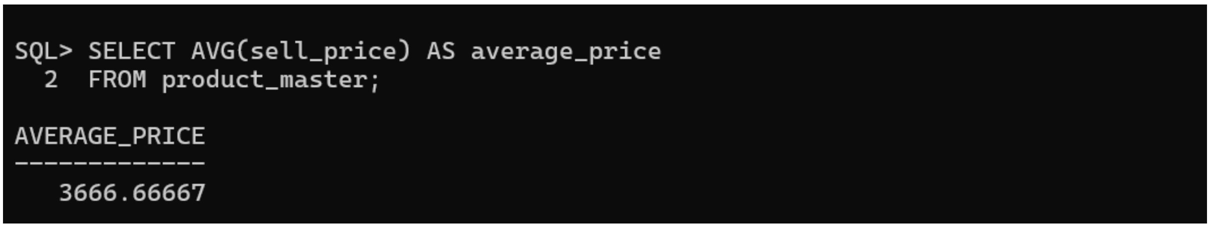
```
SQL> SELECT name, city, state
  2  FROM client_master
  3  WHERE state <> 'Maharashtra';

NAME                 CITY            STATE
-------------------- --------------- ---------------
Vandana Saitwal      Madras          Tamil Nadu
Ravi sreedharan      Delhi           Delhi
```

8) COUNT THE TOTAL NUMBER OF ORDERS.



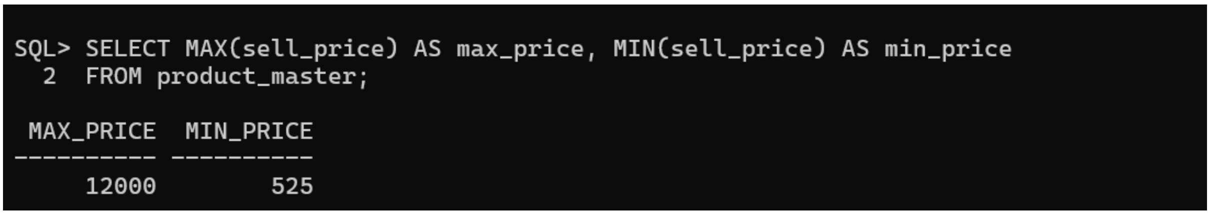
```
SQL> SELECT COUNT(*) AS total_orders
  2  FROM sales_order;

TOTAL_ORDERS
------------
           6
```

9) CALCULATE THE AVERAGE PRICE OF ALL THE PRODUCTS.



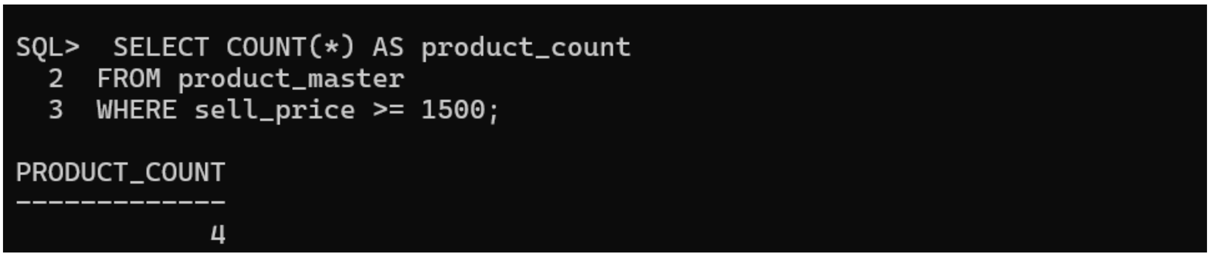
```
SQL> SELECT AVG(sell_price) AS average_price
  2  FROM product_master;

AVERAGE_PRICE
-------------
   3666.66667
```

10) DETERMINE THE MAXIMUM AND MINIMUM PRODUCT PRICES. RENAME THE OUTPUT AS MAX_PRICE AND MIN_PRICE RESPECTIVELY.



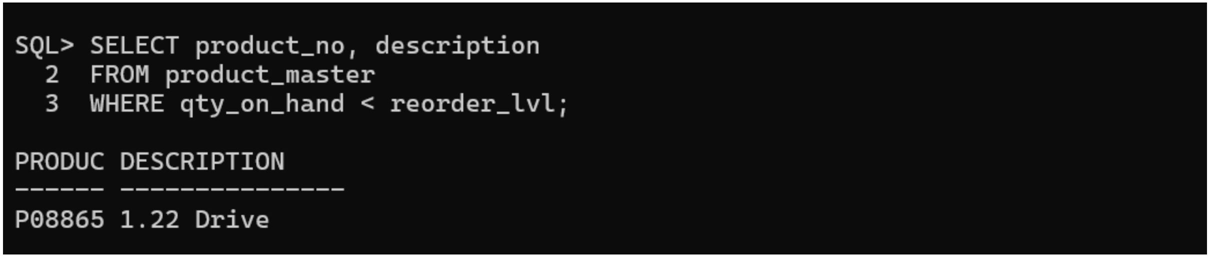
```
SQL> SELECT MAX(sell_price) AS max_price, MIN(sell_price) AS min_price
  2  FROM product_master;

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

11) COUNT THE NUMBER OF PRODUCTS HAVING PRICE GREATER THAN OR EQAL TO 1500.



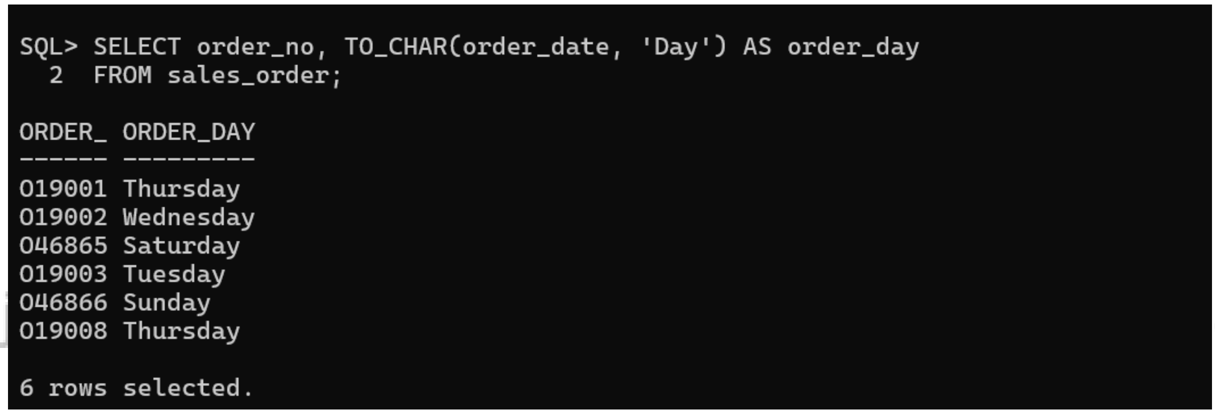
```
SQL>  SELECT COUNT(*) AS product_count
  2  FROM product_master
  3  WHERE sell_price >= 1500;

PRODUCT_COUNT
-------------
            4
```

12) FIND ALL THE PRODUCTS WHOSE QTY_ON_HAND IS LESS THAN REORDER LEVEL.



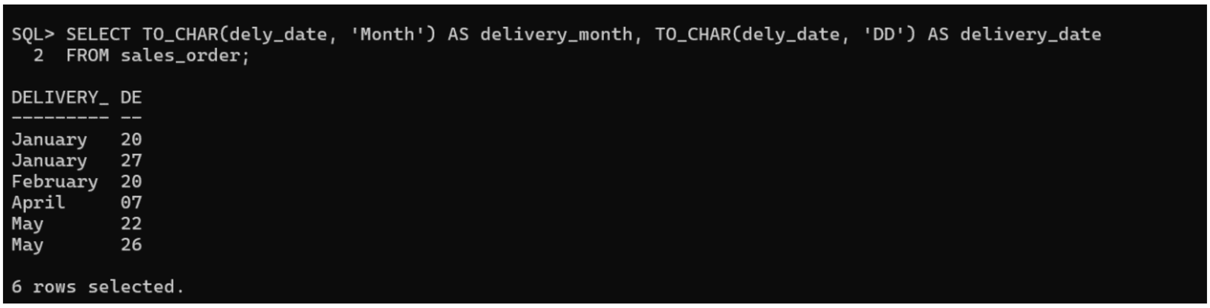
```
SQL> SELECT product_no, description
  2  FROM product_master
  3  WHERE qty_on_hand < reorder_lvl;

PRODUC DESCRIPTION
------ ---------------
P08865 1.22 Drive
```

13) DISPLAY THE ORDER NUMBER AND DAY ON WHICH CLIENTS PLACED THEIR ORDER.



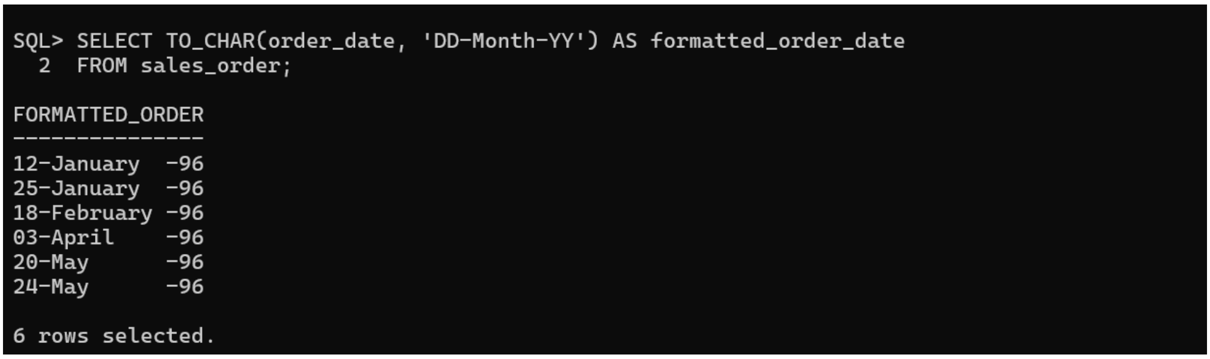
```
SQL> SELECT order_no, TO_CHAR(order_date, 'Day') AS order_day
  2  FROM sales_order;

ORDER_ ORDER_DAY
------ ---------
O19001 Thursday
O19002 Wednesday
O46865 Saturday
O19003 Tuesday
O46866 Sunday
O19008 Thursday

6 rows selected.
```

14) DISPLAY THE MONTH(IN ALPHBETS) AND DATE WHEN THE ORDER MUST BE DELIVERED.



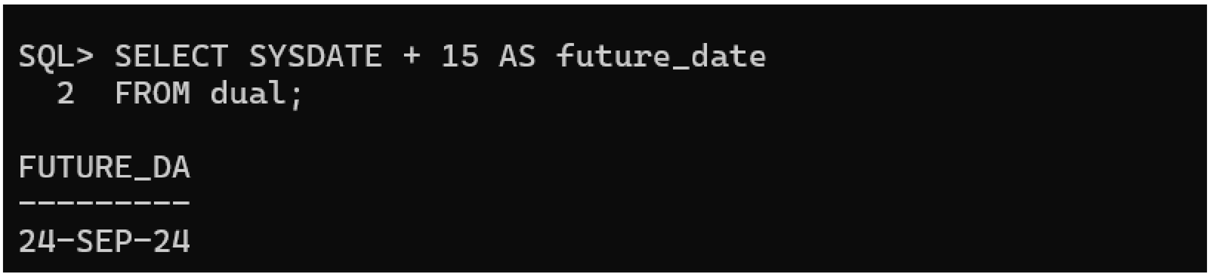
```
SQL> SELECT TO_CHAR(dely_date, 'Month') AS delivery_month, TO_CHAR(dely_date, 'DD') AS delivery_date
  2  FROM sales_order;

DELIVERY_ DE
--------- --
January   20
January   27
February  20
April     07
May       22
May       26

6 rows selected.
```

15) DISPLAY THE ORDER_DATE IN THE FORMATE 'DD- MONTH – YY'.



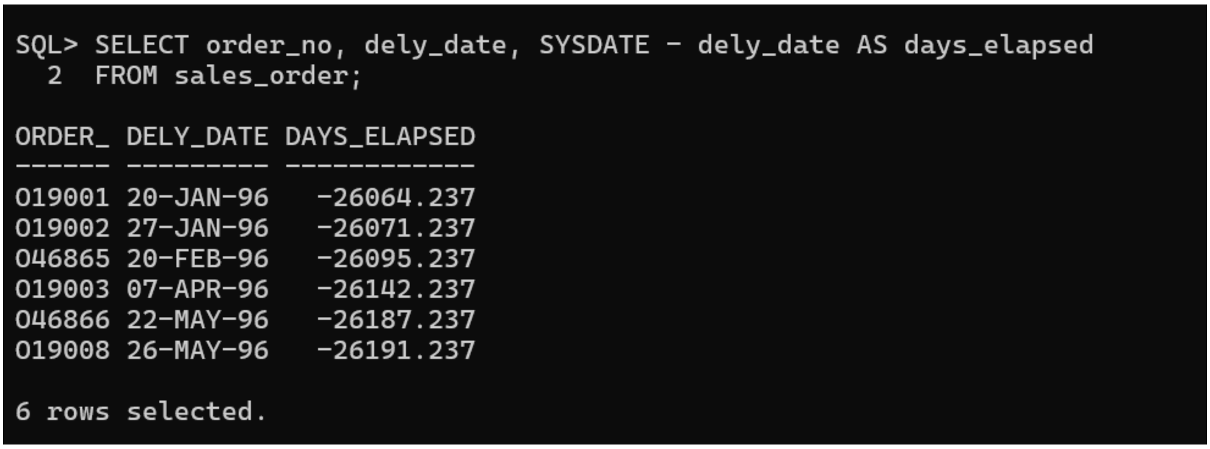
```
SQL> SELECT TO_CHAR(order_date, 'DD-Month-YY') AS formatted_order_date
  2  FROM sales_order;

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

6 rows selected.
```

16) FIND THE DATE, 15 DAYS AFTER TODAY'S DATE.



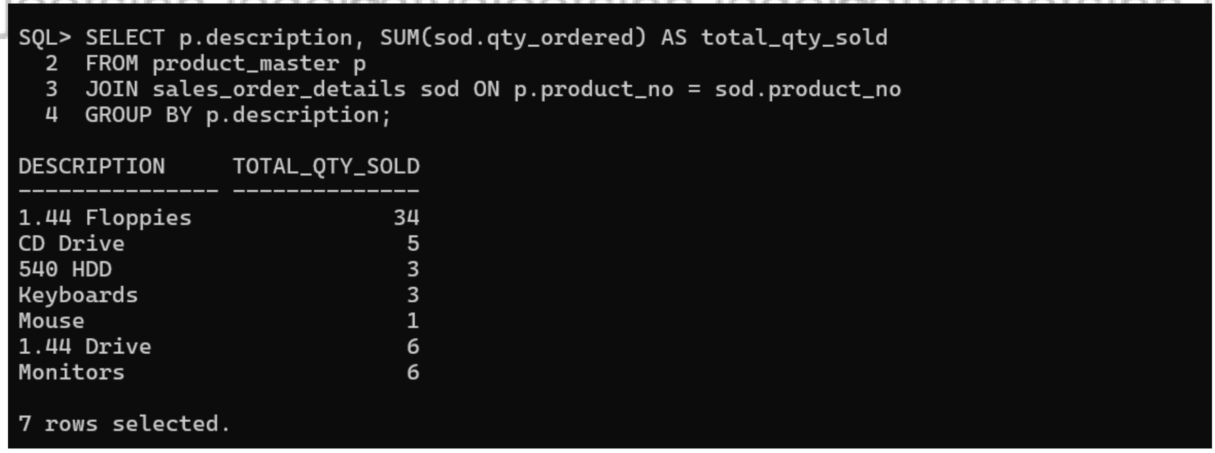
```
SQL> SELECT SYSDATE + 15 AS future_date
  2  FROM dual;

FUTURE_DA
---------
24-SEP-24
```

17) FIND THE NUMBER OF DAYS ELAPSED BETWEEN TODAY'S DATE AND THE DELIVERY DATE OF THE ORDERS PLACED BY THE CLIENTS..



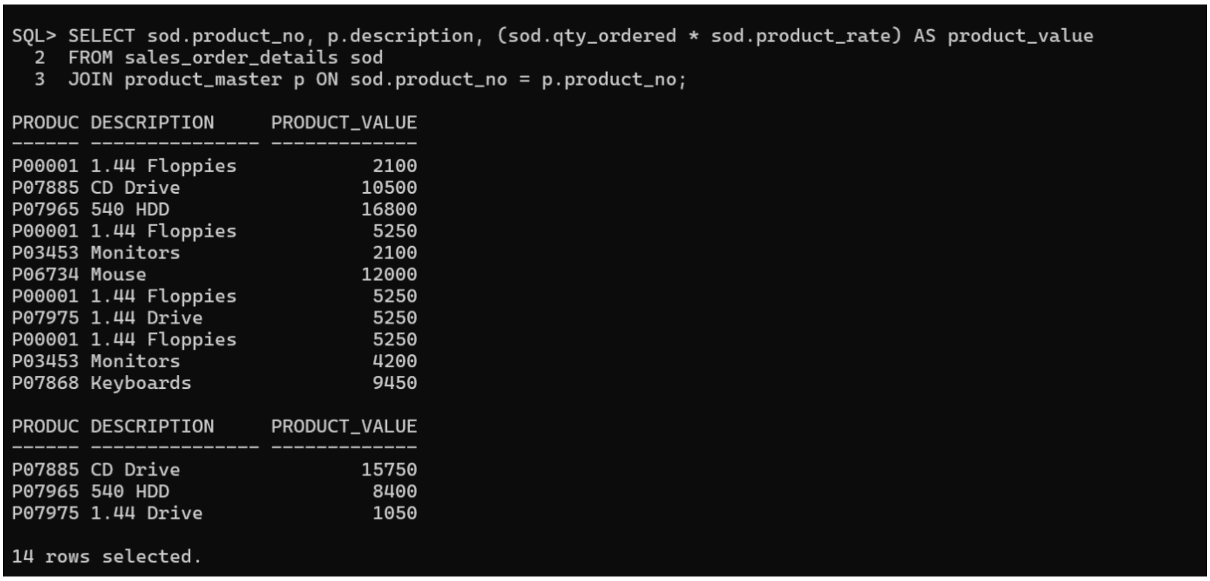
```
SQL> SELECT order_no, dely_date, SYSDATE - dely_date AS days_elapsed
  2  FROM sales_order;

ORDER_ DELY_DATE DAYS_ELAPSED
------ --------- ------------
O19001 20-JAN-96   -26064.237
O19002 27-JAN-96   -26071.237
O46865 20-FEB-96   -26095.237
O19003 07-APR-96   -26142.237
O46866 22-MAY-96   -26187.237
O19008 26-MAY-96   -26191.237

6 rows selected.
```

18) PRINT THE DESCRIPTION AND TOTAL QTY SOLD FOR EACH PRODUCT.



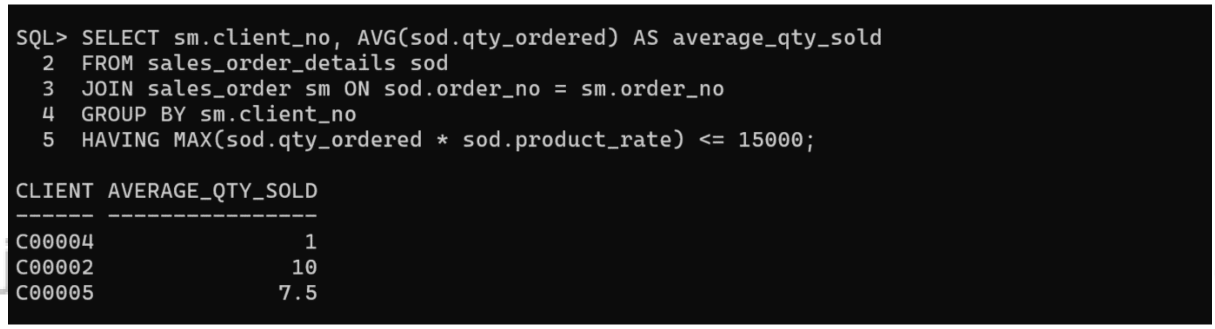
```
SQL> SELECT p.description, SUM(sod.qty_ordered) AS total_qty_sold
  2  FROM product_master p
  3  JOIN sales_order_details sod ON p.product_no = sod.product_no
  4  GROUP BY p.description;

DESCRIPTION     TOTAL_QTY_SOLD
--------------- --------------
1.44 Floppies               34
CD Drive                     5
540 HDD                      3
Keyboards                    3
Mouse                        1
1.44 Drive                   6
Monitors                     6

7 rows selected.
```

19) FIND THE VALUE OF EACH PRODUCT SOLD.



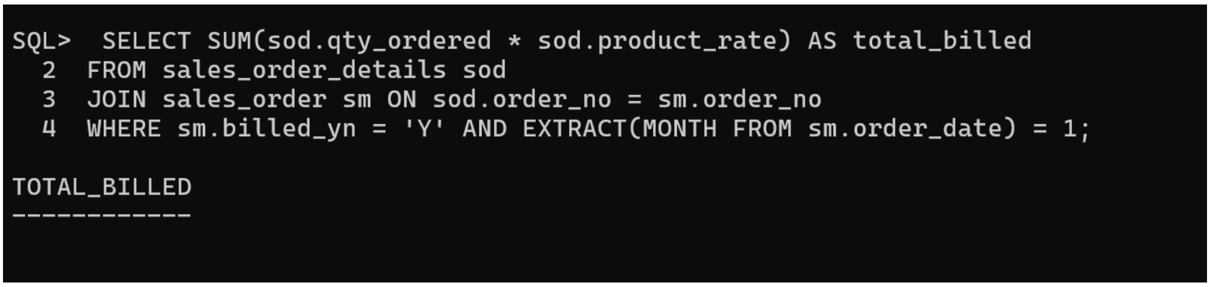
```
SQL> SELECT sod.product_no, p.description, (sod.qty_ordered * sod.product_rate) AS product_value
  2  FROM sales_order_details sod
  3  JOIN product_master p ON sod.product_no = p.product_no;

PRODUC DESCRIPTION     PRODUCT_VALUE
------ --------------- -------------
P00001 1.44 Floppies            2100
P07885 CD Drive                10500
P07965 540 HDD                 16800
P00001 1.44 Floppies            5250
P03453 Monitors                 2100
P06734 Mouse                   12000
P00001 1.44 Floppies            5250
P07975 1.44 Drive               5250
P00001 1.44 Floppies            5250
P03453 Monitors                 4200
P07868 Keyboards                9450

PRODUC DESCRIPTION     PRODUCT_VALUE
------ --------------- -------------
P07885 CD Drive                15750
P07965 540 HDD                  8400
P07975 1.44 Drive               1050

14 rows selected.
```

20) CALCULATE THE AVERAGE QTY SOLD FOR EACH CLIENT THAT HAS A MAXIMUM ORDER VALUE OF 15000.



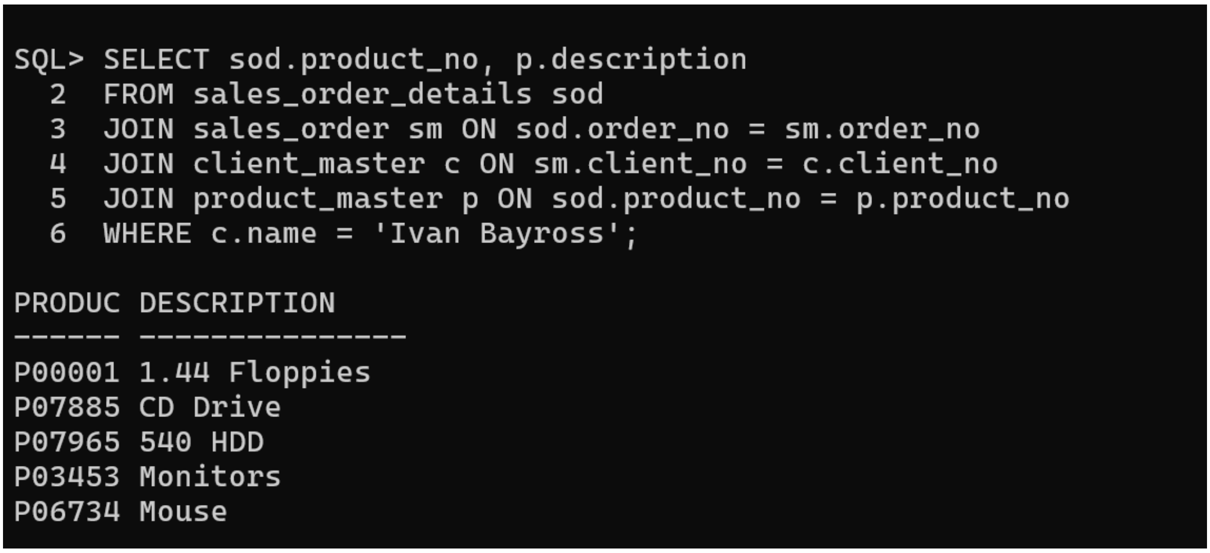
```
SQL> SELECT sm.client_no, AVG(sod.qty_ordered) AS average_qty_sold
  2  FROM sales_order_details sod
  3  JOIN sales_order sm ON sod.order_no = sm.order_no
  4  GROUP BY sm.client_no
  5  HAVING MAX(sod.qty_ordered * sod.product_rate) <= 15000;

CLIENT AVERAGE_QTY_SOLD
------ ----------------
C00004                1
C00002               10
C00005              7.5
```

21) FIND OUT THE SUM TOTAL OF ALL BILLED ORDERS FOR THE MONTH OF JANUARY.



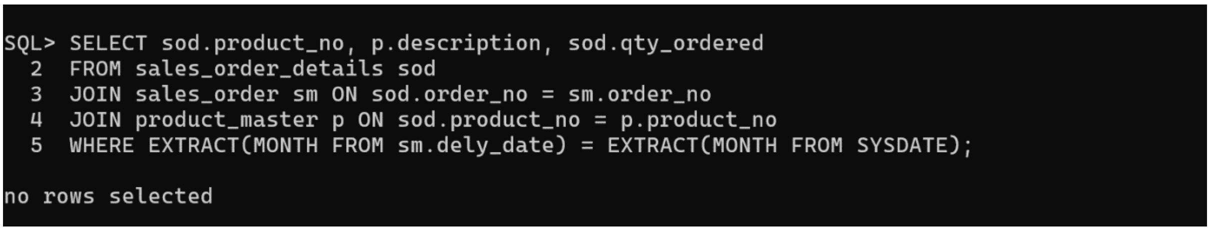
```
SQL>  SELECT SUM(sod.qty_ordered * sod.product_rate) AS total_billed
  2  FROM sales_order_details sod
  3  JOIN sales_order sm ON sod.order_no = sm.order_no
  4  WHERE sm.billed_yn = 'Y' AND EXTRACT(MONTH FROM sm.order_date) = 1;

TOTAL_BILLED
------------
```

22) FIND OUT THE PRODUCTS, WHICH HAVE BEEN SOLD TO ‘IVAN BAYROSS’.



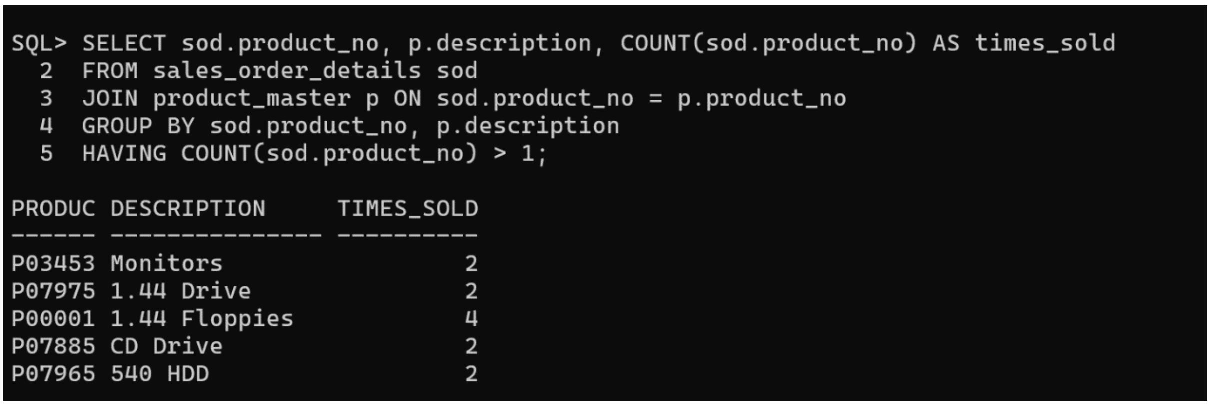
```
SQL> SELECT sod.product_no, p.description
     FROM sales_order_details sod
     JOIN sales_order sm ON sod.order_no = sm.order_no
     JOIN client_master c ON sm.client_no = c.client_no
     JOIN product_master p ON sod.product_no = p.product_no
     WHERE c.name = 'Ivan Bayross';

PRODUC DESCRIPTION
------ ---------------
P00001 1.44 Floppies
P07885 CD Drive
P07965 540 HDD
P03453 Monitors
P06734 Mouse
```

23) FIND OUT THE PRODUCTS AND THEIR QUANTITIES THAT WILL HAVE TO BE DELIVERED IN THE CURRENT MONTH.



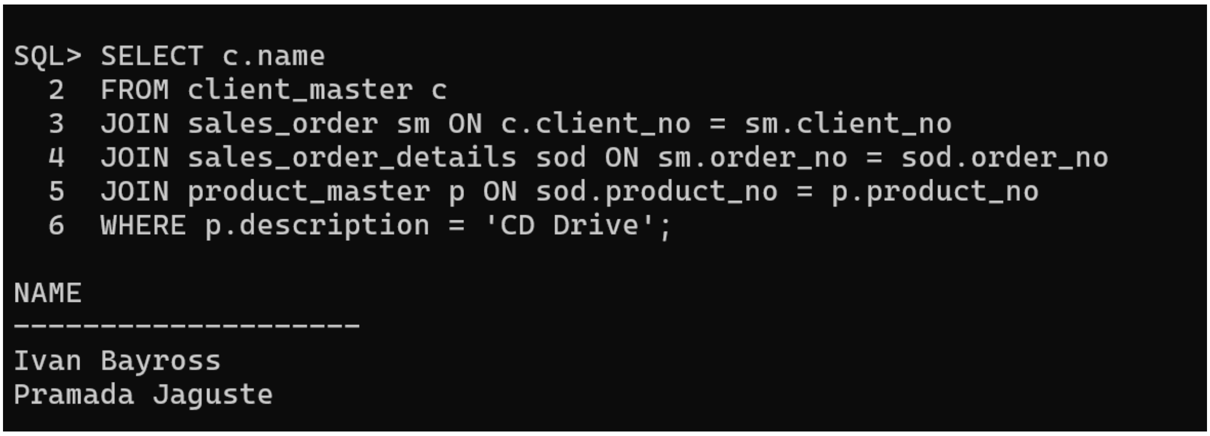
```
SQL> SELECT sod.product_no, p.description, sod.qty_ordered
     FROM sales_order_details sod
     JOIN sales_order sm ON sod.order_no = sm.order_no
     JOIN product_master p ON sod.product_no = p.product_no
     WHERE EXTRACT(MONTH FROM sm.dely_date) = EXTRACT(MONTH FROM SYSDATE);

no rows selected
```

24) FIND THE PRODUCT_NO AND DESCRIPTION OF CONSTANTLY SOLD I.E. RAPIDLY MOVING PRODUCTS.



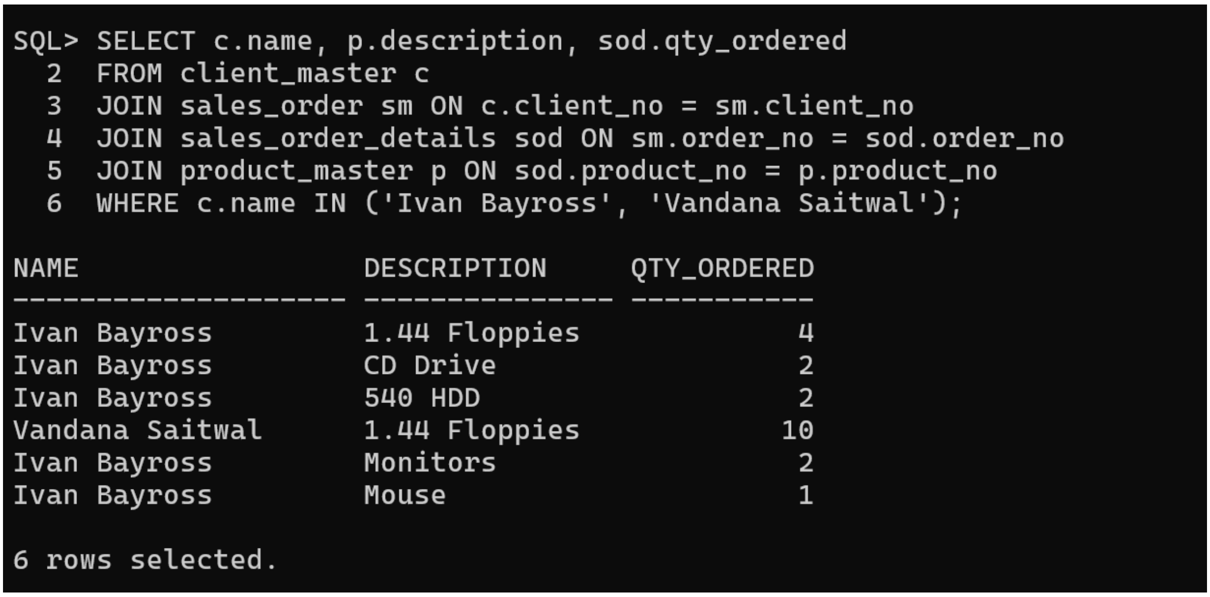
```
SQL> SELECT sod.product_no, p.description, COUNT(sod.product_no) AS times_sold
     FROM sales_order_details sod
     JOIN product_master p ON sod.product_no = p.product_no
     GROUP BY sod.product_no, p.description
     HAVING COUNT(sod.product_no) > 1;

PRODUC DESCRIPTION     TIMES_SOLD
------ --------------- ----------
P03453 Monitors                 2
P07975 1.44 Drive               2
P00001 1.44 Floppies            4
P07885 CD Drive                 2
P07965 540 HDD                  2
```

25) FIND THE NAMES OF CLIENTS WHO HAVE PURCHASED ‘CD DRIVE’.



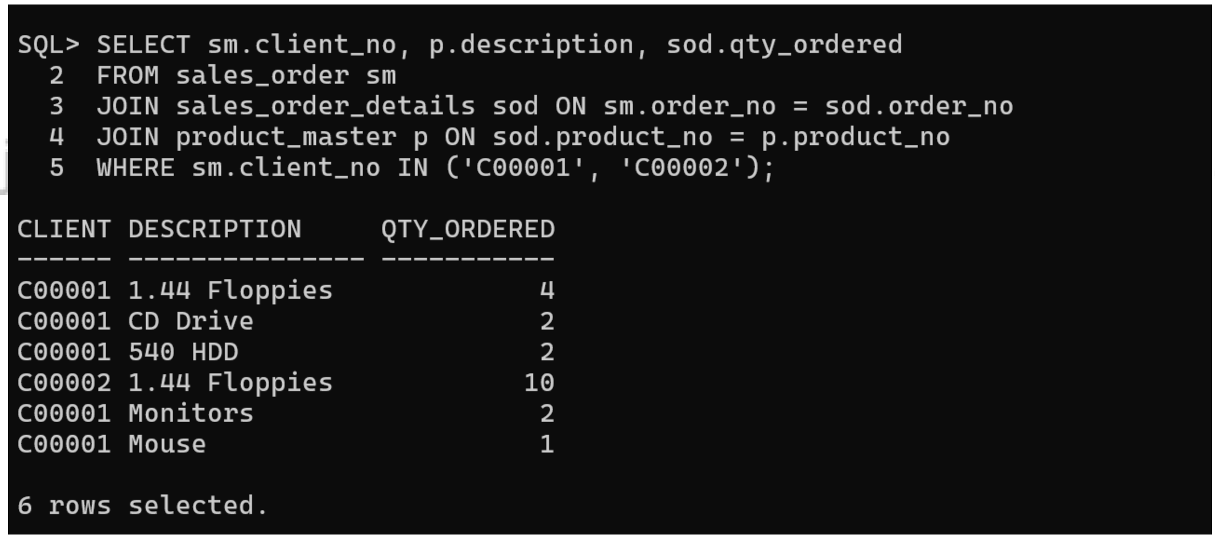
```
SQL> SELECT c.name
     FROM client_master c
     JOIN sales_order sm ON c.client_no = sm.client_no
     JOIN sales_order_details sod ON sm.order_no = sod.order_no
     JOIN product_master p ON sod.product_no = p.product_no
     WHERE p.description = 'CD Drive';

NAME
--------------------
Ivan Bayross
Pramada Jaguste
```

26) FIND THE PRODUCTS AND THEIR QUANTITIES FOR THE ORDERS PLACED BY ‘IVAN BAYROSS’ AND ‘VANDANA SAITWAL’.



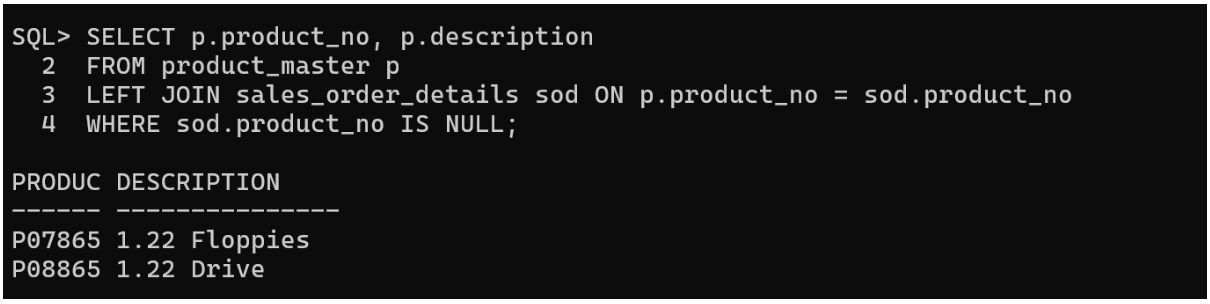
```
SQL> SELECT c.name, p.description, sod.qty_ordered
  2  FROM client_master c
  3  JOIN sales_order sm ON c.client_no = sm.client_no
  4  JOIN sales_order_details sod ON sm.order_no = sod.order_no
  5  JOIN product_master p ON sod.product_no = p.product_no
  6  WHERE c.name IN ('Ivan Bayross', 'Vandana Saitwal');

NAME                 DESCRIPTION     QTY_ORDERED
-------------------- --------------- -----------
Ivan Bayross         1.44 Floppies             4
Ivan Bayross         CD Drive                  2
Ivan Bayross         540 HDD                   2
Vandana Saitwal      1.44 Floppies            10
Ivan Bayross         Monitors                  2
Ivan Bayross         Mouse                     1

6 rows selected.
```

27) FIND THE PRODUCTS AND THEIR QUANTITIES FOR THE ORDERS PLACED BY CLIENT_NO ‘C00001’ AND ‘C00002’.



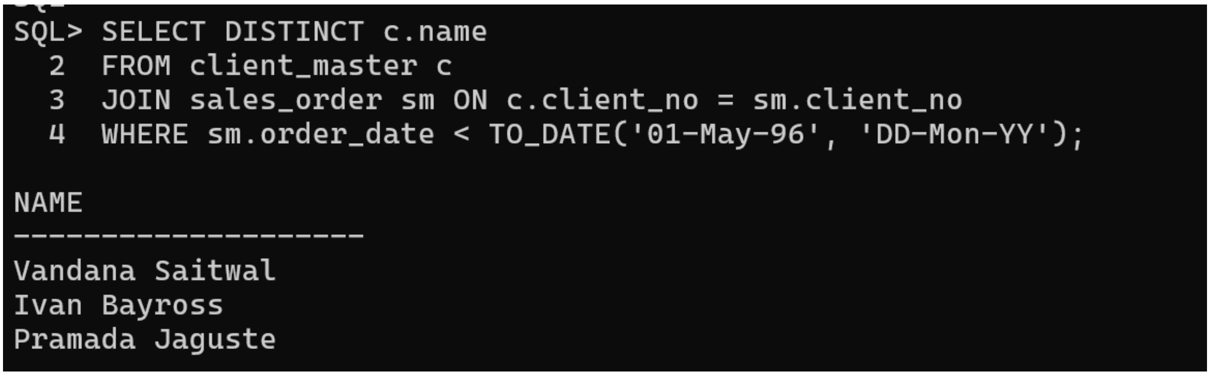
```
SQL> SELECT sm.client_no, p.description, sod.qty_ordered
  2  FROM sales_order sm
  3  JOIN sales_order_details sod ON sm.order_no = sod.order_no
  4  JOIN product_master p ON sod.product_no = p.product_no
  5  WHERE sm.client_no IN ('C00001', 'C00002');

CLIENT DESCRIPTION     QTY_ORDERED
------ --------------- -----------
C00001 1.44 Floppies             4
C00001 CD Drive                  2
C00001 540 HDD                   2
C00002 1.44 Floppies            10
C00001 Monitors                  2
C00001 Mouse                     1

6 rows selected.
```

28) FIND THE PRODUCT_NO AND DESCRIPTION OF NON MOVING PRODUCTS I.E. PRODUCTS NOT BEING SOLD.



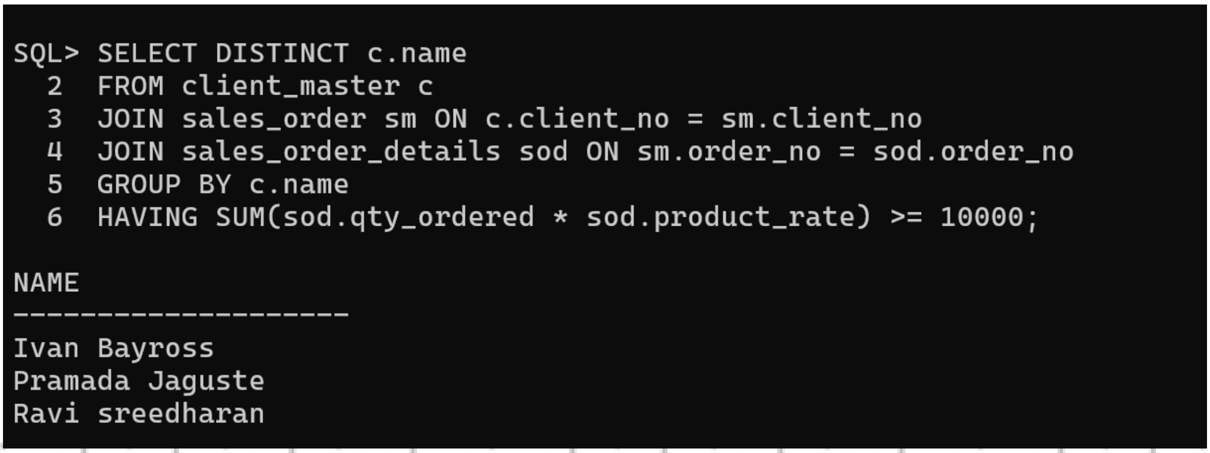
```
SQL> SELECT p.product_no, p.description
  2  FROM product_master p
  3  LEFT JOIN sales_order_details sod ON p.product_no = sod.product_no
  4  WHERE sod.product_no IS NULL;

PRODUC DESCRIPTION
------ ---------------
P07865 1.22 Floppies
P08865 1.22 Drive
```

29) FIND THE CLIENT NAMES WHO HAVE PLACED ORDER BEFORE THE MONTH OF MAY ’96.



```
SQL> SELECT DISTINCT c.name
  2  FROM client_master c
  3  JOIN sales_order sm ON c.client_no = sm.client_no
  4  WHERE sm.order_date < TO_DATE('01-May-96', 'DD-Mon-YY');

NAME
--------------------
Vandana Saitwal
Ivan Bayross
Pramada Jaguste
```

30) FIND THE NAMES OF CLIENTS WHO HAVE PALCED ORDERS WORTH RS. 10000 OR MORE.



```
SQL> SELECT DISTINCT c.name
  2  FROM client_master c
  3  JOIN sales_order sm ON c.client_no = sm.client_no
  4  JOIN sales_order_details sod ON sm.order_no = sod.order_no
  5  GROUP BY c.name
  6  HAVING SUM(sod.qty_ordered * sod.product_rate) >= 10000;

NAME
--------------------
Ivan Bayross
Pramada Jaguste
Ravi sreedharan
```