Assignment 1

SIXTY SELF REVIEW SQL SENTENCE CONSTRUCTS FOR PRACTICE

1. Single table retrieval

1) Find out the names of all the clients.

```
SELECT name FROM client_master
```

2) Print the entire client_master table.

```
SELECT * FROM client_master
```

3) Retrieve the list of names and the cities of all the clients

```
SELECT name, city FROM client_master;
```

4) List the various products available from the product_master table.

```
SELECT description "Product Name" FROM product_master;
```

5) Find the names of all clients having ‘a’ as the second letter in their table.

```
SELECT name FROM client_master WHERE name LIKE '_a%';
```

6) Find the names of all clients who stay in a city whose second letter is ‘a’

```
SELECT name FROM client_master WHERE city LIKE '_a%';
```

7) Find out the clients who stay in a city ‘Bombay’ or city ‘Delhi’ or city ‘Madras’.

```
SELECT name FROM client_master WHERE city IN('Bombay','Delhi','Madras');
```

8) List all the clients who are located in Bombay.

```
SELECT name FROM client_master WHERE city='Bombay';
```

9) Print the list of clients whose bal_due are greater than value 10000

```
SELECT name FROM client_master WHERE bal_due > 10000 ;
```

10) Print the information from sales_order table of orders placed in the month of January.

```
SELECT * FROM sales_order WHERE EXTRACT(MONTH FROM
s_order_date)='jan';
```

11) Display the order information for client_no ‘C00001’ and ‘C00002’

```
SELECT * FROM sales_order WHERE client_no = 'C00001' AND client_no =
'C00002';
```

12) Find the products with description as ‘1.44 Drive’ and ‘1.22 Drive’

```
SELECT product_no,description FROM sales_order WHERE description IN('1.44
Drive','1.22 Drive');
```

13) Find the products whose selling price is greater than 2000 and less than or equal to 5000

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

14) Find the products whose selling price is more than 1500 and also find the new selling price as original selling price * 15

```
SELECT product_no,description,sell_price*15 FROM product_master WHERE sell_price > 1500;
```

15) Rename the new column in the above query as new_price

```
ALTER TABLE product_master RENAME sell_price*15 TO new_price;
```

16) Find the products whose cost price is less than 1500

```
SELECT product_no,description FROM product_master WHERE cost_price < 1500;
```

17) List the products in sorted order of their description

```
SELECT product_no, description FROM product_master ORDER BY description;.
```

18) Calculate the square root the price of each product.

```
SELECT SQRT(sell_price),SQRT(cost_price) FROM product_master;
```

19) Divide the cost of product ‘540 HDD’ by difference between its price and 100

```
UPDATE product_master SET cost_price = (cost_price\(cost_price-100)) WHERE description = '540 HD';
```

20) List the names, city and state of clients not in the state of Maharashtra

```
SELECT name,city,state from client_master WHERE state != ‘Maharashtra';
```

21) List the product_no, description, sell_price of products whose description begin with letter ‘M’

```
SELECT product_no, description,sell_price FROM product_master WHERE description LIKE 'M%';
```

22) List all the orders that were canceled in the month of May.

```
SELECT s_order_no,s_order_date,delay_date from sales_order WHERE EXTRACT(MONTH from dely_date) = 'May';
```

2. Set Functions and Concatenation :

23) Count the total number of orders.

```
SELECT count(s_order_no) FROM sales_order;
```

24) Calculate the average price of all the products.

```
SELECT AVG(sell_price),AVG(cost_price) FROM product_master;
```

25) Calculate the minimum price of products.

```
SELECT MIN(sell_price),MIN(cost_price) FROM product_master;
```

26) Determine the maximum and minimum product prices. Rename the title as max_price and min_price respectively.

```
SELECT MAX(cost_price) AS max_cost_price,MAX(sell_price) AS max_sell_price
,MIN(cost_price) AS min_cost_price ,MIN(sell_price) AS min_sell_price;
```

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

```
SELECT COUNT(product_no) FROM product_master WHERE sell_price >= 1500;
```

28) Find all the products whose qty_on_hand is less than reorder level.

```
SELECT product_no,description FROM product_master WHERE qty_on_hand<
reorder_lvl;
```

29) Print the information of client_master, product_master, sales_order table in the following formate for all the records

{cust_name} has placed order {order_no} on {s_order_date}.

```
SELECT name || ‘has placed order’ || s_order_no ||’on’|| s_order_date from
Client_master c,sales_order s WHERE s.client_no=c.client_no;
```

3. Having and Group by:

30) Print the description and total qty sold for each product.

```
SELECT P.PRODUCT_NO, P.DESCRIPTION, SUM(S.QTY_DISP) AS TOTAL_QTY_SOLD FROM
PRODUCT_MASTER P LEFT JOIN SALES_ORDER_DETAILS S ON P.PRODUCT_NO
=S.PRODUCT_NO GROUP BY P.PRODUCT_NO, P.DESCRIPTION;
```

31) Find the value of each product sold.

```
SELECT S.PRODUCT_NO, P.DESCRIPTION, SUM(S.QTY_DISP) AS TOTAL_QTY_SOLD,
SUM(S.QTY_DISP*P.SELL_PRICE) AS TOTAL_VALUE_SOLD FROM SALES_ORDER_DETAILS S
INNER JOIN PRODUCT_MASTER P ON S.PRODUCT_NO =P.PRODUCT_NO GROUP BY
S.PRODUCT_NO, P.DESCRIPTION;
```

32) Calculate the average qty sold for each client that has a maximum order value of 15000.00

```
SELECT S.CLIENT_NO,AVG(D.QTY_DISP) AS AVG_QTY_SOLD FROM SALES_ORDER
S,SALES_ORDER_DETAILS D WHERE D.PRODUCT_RATE > 15000 GROUP BY S.CLIENT_NO;
```

33) Find out the total sales amount receivable for the month of jan. it will be the sum total of all the billed orders for the month

```
SELECT s.s_order_no, s.s_order_date,
sum(so.qty_ordered*so.product_rate),sum(so.qty_disp * so.product_rate) FROM
sales_order s, sales_order_details so WHERE so.s_order_no = s.s_order_no AND
s.billed_yn = 'Y' AND EXTRACT(MONTH FROM s.s_order_date) = 1  GROUP BY
s.s_order_no,s.s_order_date;
```

34) Print the information of product_master, order_detail table in the following format for all the records

{Description} worth Rs. {Total sales for the product} was sold.

```
SELECT p.description || ' Worth Rs ' || sum(d.qty_disp*d.product_rate) FROM
product_master p, sales_order_details d WHERE p.product_no = d.product_no
GROUP BY p.description;
```

35) Print the information of product_master, order_detail table in the following format for all the records

{Description} worth Rs. {Total sales for the product} was produced in the month of {s_order_date} in month formate.

```
SELECT p.description || ' Worth Rs ' || sum(d.qty_disp * d.product_rate) || ' was
produced in the month of '||s.s_order_date 'MONTH' FROM product_master p,
sales_order_details d, sales_order s WHERE p.product_no = d.product_no AND
s.s_order_no = d.s_order_no GROUP BY p.description,s.s_order_date;
```

4. Nested Queries :

36) Find the product_no and description of non-moving products.

37) Find the customer name, address1, address2, city and pin code for the client who has placed order no ‘O19001’

```
SELECT c.name, c.address1, c.address2, c.city, c.pincode FROM Client_master c
INNER JOIN Sales_order s ON c.client_no = s.client_no
WHERE s.s_order_no = 'O19001';
```

38) Find the client names who have placed orders before the month of May, 1996

```
SELECT DISTINCT c.name FROM Client_master c INNER JOIN Sales_order s ON
c.client_no = s.client_no WHERE s.s_order_date < TO_DATE('01/05/1996',
'DD/MM/YYYY');
```

39) Find out if product ‘1.44 Drive’ is ordered by client and print the client_no, name to whom it is was sold.

```
SELECT DISTINCT c.name FROM Client_master c INNER JOIN sales_order s ON
c.client_no = s.client_no WHERE s.s_order_date < TO_DATE('1996-05-01','YYYY-
MM-DD');
```

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

5. Queries using Date:

41) Display the order number and day o which clients placed their order

```
SELECT s_order_no, TO_CHAR(s_order_date, 'Day') AS order_day
FROM sales_order;
```

42) Display the month (in alphabets) and date when the order must deliver

```
SELECT TO_CHAR(dely_date, 'Month') AS delivery_month, TO_CHAR(dely_date,
'DD') AS delivery_day FROM sales_order;
```

43) Display the s_order_date in the format ‘DD-MM-YY’. E.g. 12-February-1996

```
SELECT TO_CHAR(s_order_date, 'DD-MM-YY') AS formatted_order_date
FROM sales_order;
```

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

```
SELECT SYSDATE+10 FROM dual;
```

45) Find the number of days elapsed between today’s date and the delivery date of the orders placed by the clients.

```
SELECT s.s_order_no,
    TO_DATE(s.dely_date, 'YYYY-MM-DD') AS delivery_date,
    TO_DATE(SYSDATE, 'YYYY-MM-DD') AS current_date,
    TO_DATE(s.dely_date, 'YYYY-MM-DD') - TO_DATE(SYSDATE, 'YYYY-MM-DD')
AS days_elapsed FROM sales_order s;
```

6. Table Updations:

46) Change the s_order_date of client_no ‘C00001’ to 24/07/96.

```
UPDATE sales_order SET s_order_date = TO_DATE('24/07/96', 'DD/MM/YY')
WHERE client_no = 'C00001';
```

47) Change the selling price of ‘1.44 Floppy Drive’ to Rs. 1150.00

```
UPDATE product_master SET sell_price = 1150.00 WHERE description = '1.44 Floppy Drive';
```

48) Delete the records with order number ‘O19001’ from the order table.

```
DELETE FROM sales_order WHERE s_order_no = 'O19001';
```

49) Delete all the records having delivery date before 10th July’96

```
DELETE FROM sales_order WHERE dely_date < TO_DATE('1996-07-10', 'YYYY-MM-DD');
```

50) Change the city of client_no ‘C00005’ to ’Bombay’.

```
UPDATE client_master SET city = 'Bombay' WHERE client_no = 'C00005';
```

51) Change the delivery date of order number ‘O10008” to 16/08/96

```
UPDATE Sales_order SET dely_date = TO_DATE('16/08/96', 'DD/MM/YY') WHERE s_order_no = 'O10008';
```

52) Change the bal_due of client_no ‘C00001’ to 1000

```
UPDATE Client_master SET bal_due = 1000 WHERE client_no = 'C00001';
```

53) Change the cost price of ‘1.22 Floppy Drive’ to Rs. 950.00.

```
UPDATE Product_master SET cost_price = 950.00 WHERE description = '1.22 Floppy Drive';
```

Assignment 1