Institute of Computer Technology

B. Tech. Computer Science and Engineering

Semester: III

Sub: Database Management System

Course Code: 2CSE301

Practical Number:3

Objective:

Performing queries with various operators and functions.

- Scenario: Mohan was worried about total income to getting raised every month for which he has to query differently in sales table. Thus he suggested IT Company to have multiple options to see the count of sales happen every day or weekly or monthly.

Exercise:

Solve the following queries using the database given in practical 1 and above table.

A) Queries on computation on table data:

1) Find the name of all clients having 'a ' as the second letter in their names.

Code :

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

Output :

	name
▶	Basu
	Ravi

2) Find out the clients whose name is four character ling and second letter is 'a'.

Code :

```
SELECT name
FROM client_master
WHERE name LIKE '_a__';
```

Output :

	name
▶	Basu
	Ravi

3) Find out the name of city whose second last character is ‘a’.

Code :

```
SELECT city
FROM client_master
WHERE city LIKE '%a_';
```

Output :

	city
▶	Bombay
	Bombay
	Bombay
	Bombay

4) Print the list of clients whose bal_due is greater than or equal to 10000.

Code :

```
SELECT *
FROM client_master
WHERE bal_due >= 10000;
```

Output :

	Client_no	Name	City	Pincode	State	Bal_due

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

Code :

```
SELECT *
FROM sales_order
WHERE TO_CHAR(order_date, 'MM') = '01';
```

Output :

	Order_no	Order_date	Client_no	S_no	Dely_type	Billed_yn	Dely_date	Order_status
▶	O1902	2015-01-25	C002	S002	P	N	2015-06-27	Cancelled

6) Display the order information for client_no ‘C003’ and ‘C001’.

Code :

```
SELECT *
FROM sales_order
WHERE client_no IN ('C003', 'C001');
```

Output :

Order_no	Order_date	Client_no	S_no	Dely_type	Billed_yn	Dely_date	Order_status
O1901	2015-06-12	C001	S001	F	N	2015-06-20	InProcess
O4665	2015-02-18	C003	S003	F	Y	2015-02-20	Fulfilled
O1903	2015-04-03	C001	S001	F	Y	2015-04-07	Fulfilled

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

Code :

```
SELECT *
FROM product_master
WHERE sell_price > 2000 AND sell_price <= 5000;
```

Output :

Product_no	Description	P_percent	U_measure	Qty_on_hand	Reorder_lvl	Sell_price	Cost_price
P005	Keyboards	2.00	Piece	10	3	3150.00	3050.00

8) 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.

Code :

```
SELECT p_no, description, sell_price, sell_price * 0.15 AS new_price
FROM product_master
WHERE sell_price > 1500;
```

Output :

Product_no	description	sell_price	new_price
P002	Monitor	12000.00	1800.0000
P005	Keyboards	3150.00	472.5000
P006	Cd Drive	5250.00	787.5000
P007	1.44 Drive	8400.00	1260.0000

9) List the names, city and state of clients who are not in the state of 'Maharashtra'.

Code :

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

Output :

	name	city	state
▶	Ravi	Delhi	Gujarat

10) Count the total number of orders.

Code :

```
SELECT COUNT(*) AS total_orders
FROM sales_order;
```

Output :

	total_orders
▶	6

11) Calculate the average price of all products.

Code :

```
SELECT AVG(sell_price) AS average_price
FROM product_master;
```

Output :

	average_price
▶	5970.000000

12) Determine the maximum and minimum product prices. Rename the output as max_price and min_price respectively.

Code :

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

Output :

	max_price	min_price
▶	12000.00	1050.00

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

Code :

```
SELECT COUNT(*) AS products_count
FROM product_master
WHERE sell_price >= 1500;
```

Output :

	products_count
▶	4

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

Code :

```
SELECT *
FROM product_master
WHERE qty_on_hand < reorder_level;
```

Output :

	Product_no	Description	P_percent	U_measure	Qty_on_hand	Reorder_lvl	Sell_price	Cost_price

15) Create table cmaster from client_master table.

Code :

```
CREATE TABLE cmaster AS
SELECT *
FROM client_master;
```

Output :

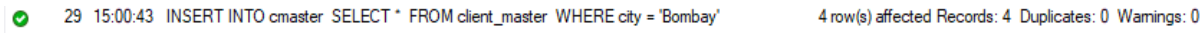
28 14:58:28 CREATE TABLE cmaster AS SELECT * FROM client_master 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

16) Insert data in cmaster from client_master where city='bombay'.

Code :

```
INSERT INTO cmaster
SELECT *
FROM client_master
WHERE city = 'Bombay';
```

Output :

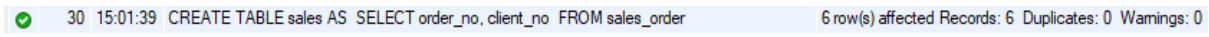


17) Create table sales from sales_order with order_no and client_no columns.

Code :

```
CREATE TABLE sales AS
SELECT order_no, client_no
FROM sales_order;
```

Output :

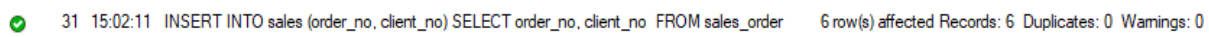


18) Insert data in sales from sales_order table.

Code :

```
INSERT INTO sales (order_no, client_no)
SELECT order_no, client_no
FROM sales_order;
```

Output :



B) Queries on Date manipulation:

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

Code :

```
SELECT order_no, DATE_FORMAT(order_date, '%W') AS order_day
FROM sales_order;
```

Output :

order_no	order_day
O1901	Friday
O1902	Sunday
O4665	Wednesday
O1903	Friday
O4666	Wednesday
O1908	Sunday

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

Code :

```
SELECT Order_no, DATE_FORMAT(dely_date, '%M') AS dely_month, dely_date
FROM sales_order;
```

Output :

Order_no	dely_month	dely_date
O1901	June	2015-06-20
O1902	June	2015-06-27
O4665	February	2015-02-20
O1903	April	2015-04-07
O4666	May	2015-05-22
O1908	May	2015-05-26

3) Find the number of days elapsed between delivery date and order date from sales_order table.

Code :

```
SELECT Order_no, DATEDIFF(dely_date, Order_date) AS days_elapsed
FROM sales_order;
```

Output :

Order_no	days_elapsed
O1901	8
O1902	153
O4665	2
O1903	4
O4666	79
O1908	2

4) Find the date, 15 days after today's date.

Code :

```
SELECT CURDATE() + INTERVAL 15 DAY AS date_15_days_later;
```

Output :

date_15_days_later
2024-09-02

5) Display current date and time.

Code :

```
SELECT NOW() AS current_date_time;
```

Output :

current_date_time
2024-08-18 15:31:18

6) Display system time.

Code :

```
SELECT CURTIME() AS system_time;
```

Output :

system_time
15:31:50