**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.*

1. **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 :**



* 1. **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 :**

****

* 1. **Find out the name of city whose second last character is ‘a’.**

**Code :**

SELECT city

FROM client\_master

WHERE city LIKE '%a\_';

**Output :**



* 1. **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 :**



* 1. **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 :**



* 1. **Display the order information for client\_no ‘C003’ and ‘C001’.**

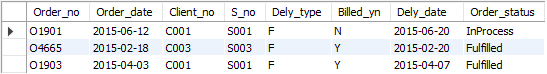
**Code :**

SELECT \*

FROM sales\_order

WHERE client\_no IN ('C003', 'C001');

**Output :**

****

* 1. **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 :**

****

* 1. **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.**

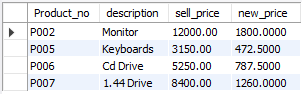
**Code :**

SELECT p\_no, description, sell\_price, sell\_price \* 0.15 AS new\_price

FROM product\_master

WHERE sell\_price > 1500;

**Output :**



* 1. **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 :**

****

* 1. **Count the total number of orders.**

**Code :**

SELECT COUNT(\*) AS total\_orders

FROM sales\_order;

**Output :**

****

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

**Code :**

SELECT AVG(sell\_price) AS average\_price

FROM product\_master;

**Output :**

****

* 1. **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 :**



* 1. **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 :**

****

* 1. **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 :**

****

* 1. **Create table cmaster from client\_master table.**

**Code :**

CREATE TABLE cmaster AS

SELECT \*

FROM client\_master;

**Output :**

****

* 1. **Insert data in cmaster from client\_master where city=’bombay’.**

**Code :**

INSERT INTO cmaster

SELECT \*

FROM client\_master

WHERE city = 'Bombay';

**Output :**

****

* 1. **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 :**

****

* 1. **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 :**

****

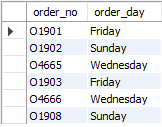
1. **Queries on Date manipulation:**
   1. **Display the order number and day on which clients placed their order.**

**Code :**

SELECT order\_no, DATE\_FORMAT(order\_date, '%W') AS order\_day

FROM sales\_order;

**Output :**

****

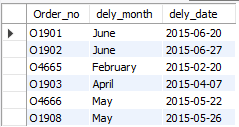
* 1. **Display the month (in alphabets) and date when the order must be delivered.**

**Code :**

SELECT Order\_no, DATE\_FORMAT(dely\_date, '%M') AS dely\_month, dely\_date

FROM sales\_order;

**Output :**

****

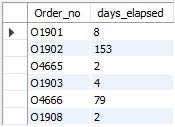
* 1. **Find the number of days elapsed between delivery date and order date from sales\_order table.**

**Code :**

SELECT Order\_no, DATEDIFF(dely\_date, Order\_date) AS days\_elapsed

FROM sales\_order;

**Output :**

****

* 1. **Find the date, 15 days after today’s date.**

**Code :**

SELECT CURDATE() + INTERVAL 15 DAY AS date\_15\_days\_later;

**Output :**



* 1. **Display current date and time.**

**Code :**

SELECT NOW() AS current\_date\_time;

**Output :**

****

* 1. **Display system time.**

**Code :**

SELECT CURTIME() AS system\_time;

**Output :**

****