Assignment No: 5

Aim :

Design and implement a database and apply at least 10 different DML queries for the following task. For a given input string display only those records which match the given pattern or a phrase in the search string. Make use of wild characters and LIKE operator for the same. Make use of Boolean and arithmetic operators wherevernecessary.

Objective :

- To understand the different issues involved in the design and implementation of a databasesystem
- To understand and use Data Manipulation Language to query to manage adatabase

Theory :

DATA MANIPULATION LANGUAGE (DML): The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

- SELECT – retrieve data from the adatabase
- INSERT – insert data into atable
- UPDATE – updates existing data within atable
- DELETE – deletes all records from atable

1. INSERT INTO: This is used to add records into a relation. These are three type of INSERT INTO queries which areas

a) Inserting a singlerecord

Syntax: INSERT INTO < relation/table name>
(field_1,field_2……field_n)VALUES (data_1,data_2,.
..... data_n);

Example: INSERT INTO student(sno,sname,address)VALUES

(1,'Ravi','M.Tech','Palakol');

b) To insert multiplerecord

Here, we are going to insert record in the "cus_tbl" table of "customers" database. INSERT INTO student

(cus_id, cus_firstname,
cus_surname) VALUES

(5, 'Ajeet', 'Maurya'),

(6, 'Deepika', 'Chopra'),

(7, 'Vimal',
'Jaiswal');
table(column1,colu
mn2.)

VLUES (value1,

2. SELECT:This is used to Retrieve data from one or moretables.

a) SELECT FROM: To display all fields for allrecords.

Syntax : SELECT * FROM relation_name;

Example : SQL> select * from dept;

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

b) SELECT - FROM -WHERE: This query is used to display a selected set of fields for a selected set of records of a relation.

Syntax: SELECT a set of fields FROM relation_name WHERECondition;

Example: SQL> select * FROM dept WHERE deptno<=20;

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS

c) SELECT - FROM -WHERE-LIKE

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column. There are two wildcards used in conjunction with the LIKE operator:

- % - The percent sign represents zero, one, or multiplecharacters
- _ - The underscore represents a singlecharacter

Syntax: SELECT*column1, column2, ...*
FROM *table_name*
WHERE *columnN* LIKE *pattern*;

Example: SELECT *
FROMCustomers
WHERE CustomerName LIKE 'a%';

LIKE Operator	**Description**
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"

WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

d) SELECT -DISTINCT

The SELECT DISTINCT statement is used to return only distinct (different) values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values. The SELECT DISTINCT statement is used to return only distinct (different) values.

Syntax: SELECT DISTINCT*column1*,*column2,* ...
FROM *table_name*;

Example: SELECT COUNT(DISTINCT Country) FROM Customers;

e) SELECT -BETWEEN

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

Syntax:

```
SELECT
column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

Example:

```
SELECT *
FROMProducts
WHERE Price BETWEEN 10 AND 20;
```

f) WHERE with - *AND* LOGICALOperator

The WHERE clause when used together with the AND logical operator, is only executed if ALL filter criteria specified are met.

```
SELECT * FROM `movies` WHERE `category_id` = 2 AND `year_released` =2008;
```

g) WHERE with - *OR* LOGICAL Operator

The WHERE clause when used together with the OR operator, is only executed if any or the entire specified filter criteria is met.

The following script gets all the movies in either category 1 or category 2

```
SELECT * FROM `movies` WHERE `category_id` = 1 OR `category_id` = 2;
```

h) **WHERE with - *Arithmetic* Operator**

Operator	Description	Example

=	Checks if the values of the two operands are equal or not, if yes, then the condition becomes true.	(A = B) is not true.
!=	Checks if the values of the two operands are equal or not, if the values are not equal then the condition becomes true.	(A != B) is true.
>	Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true.	(A > B) is not true.
<	Checks if the value of the left operand is less than the value of the right operand, if yes then the condition becomes true.	(A < B) is true.
>=	Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true.	(A >= B) is not true.
<=	Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomestrue.	(A <= B) is true.

Example: SELECTagent_code agent_name,

working_area,
(commission*2)
FROM agents

WHERE (commission*2)>0.25;

3. UPDATE-SET-WHERE: This is used to update the content of a record in arelation.

Syntax: UPDATE relation name SET Field_name1=data,field_name2=data,

WHERE field_name=data;

Example: UPDATE student SET **sname = 'kumar' WHERE sno=1;**

4. DELETE-FROM: This is used to delete all the records of a relation but it will retain the structure of thatrelation.

a) DELETE-FROM: This is used to delete all the records ofrelation.

Syntax: DELETE FROM relation_name;

Example: DELETE FROMstd;

b) DELETE -FROM-WHERE: This is used to delete a selected record from arelation.

Syntax: DELETE FROM relation_name WHEREcondition;

Example: DELETE FROM student WHERE sno = 2;

LAB PRACTICE ASSIGNMENT:

Consider the following table structure for this assignment:

CUSTOMER(Cust_id, C_name, City)

BRANCH(Branch_id, bname, City)

DEPOSIT(Acc_no , Cust_id, Amount, Branch_id, Open_date)

BORROW(Loan_no, Cust_id, Branch_id, Amount)

Perform the following queries on the above table:

1. Insert minimum 10 rows on each table and display thatdata.
2. List Cust_id along with customername.
3. List Cust_id of depositors having amount greater than10000.
4. List account date of customer‘Anil’.
5. List Cust_id of customers who have opened account after01/01/2016.

6. List account no., amount and Cust_id of customers having amount between 40,000 and 80,000.
7. List customer name starting with‘S’.
8. List customer from depositor starting with‘_a%’.
9. List customer name from customer having exactly 5 characters in theirname.
10. List Cust_id, Loan no and Loan amount ofborrowers.
11. List cust_id and C_name ofdepositors.
12. List all the customers who are depositors but notborrowers.
13. List all the customers who are both depositors andborrowers.
14. List all the customers along with their amount who are either borrowers or depositors.
15. List the cites of depositor having branch‘Akurdi’.
16. Update 10% interest to alldepositors.
17. Update 10% to all depositors living in‘Nagpur’.
18. Change living city of the ‘Nigdi’ branch borrowers toNagpur.Delete branches having deposit fromNagpur.
19. Delete depositors of branches having number of customers between 1 and3.
20. Delete depositors having deposit less thanRs500.

Conclusion:-

We have designed and implemented a database and applied different DML queries.

OUTPUT -

```
MySQL> CREATE TABLE CUSTOMER(CUST_ID INT PRIMARY KEY, CUST_NAME
VARCHAR(30) NOT NULL, ADDRESS VARCHAR(20), ACCOUNT_BALANCE INT,
ACCOUNT_OPENING_DATE DATE);

Query OK, 0 rows affected (0.326 sec)


MySQL> DESC CUSTOMER;

+---------------------+-------------+------+-----+---------+-------+

| Field               | Type        | Null | Key | Default | Extra |

+---------------------+-------------+------+-----+---------+-------+

| CUST_ID             | int(11)     | NO   | PRI | NULL    |       |

| CUST_NAME           | varchar(30) | NO   |     | NULL    |       |

| ADDRESS             | varchar(20) | YES  |     | NULL    |       |

| ACCOUNT_BALANCE     | int(11)     | YES  |     | NULL    |       |
```

| ACCOUNT_OPENING_DATE | date | YES | | NULL | |

+----------------------+-------------+------+-----+---------+-------+

5 rows in set (0.037 sec)

CREATE BORROW TABLE

MySQL> CREATE TABLE BORROW(LOAN_NO VARCHAR(5) PRIMARY KEY, CUST_ID INT, BRANCH_ID INT, AMOUNT INT);;

Query OK, 0 rows affected (0.481 sec)

MySQL> DESC BORROW;

+-----------+------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+------------+------+-----+---------+-------+

| LOAN_NO | varchar(5) | NO | PRI | NULL | |

| CUST_ID | int(11) | YES | | NULL | |

| BRANCH_ID | int(11) | YES | | NULL | |

| AMOUNT | int(11) | YES | | NULL | |

+-----------+------------+------+-----+---------+-------+

4 rows in set (0.066 sec)

1 - INSERT 10 RECORDS

MySQL> MySQL> INSERT INTO CUSTOMER VALUES(1, 'Shreyas', 'Pune', 100000, '2019-07-29'),(2, 'Anil', 'Akurdi', 500000, '2019-07-30'),(3, 'Piyush', 'Nagpur', 10000, '2019-07-22'),(4, 'Aditya', 'Nigdi', 200000, '2019-07-19'),(5, 'Ankit', 'Mumbai', 25000, '2019-07-21'),(6, 'Patil', 'Pune', 105000, '2019-06-29'),(7, 'Yash', 'Pimpri Chinchwa', 90000, '2019-05-29'),(8, 'Shubham', 'Jalgaon', 30000, '2019-01-29'),(9, 'Varun', 'Pune', 1900000, '2018-07-29'),(10, 'Nilesh', 'Nashik', 1000, '2019-09-29');

Query OK, 10 rows affected (0.127 sec)

Records: 10 Duplicates: 0 Warnings: 0

```
MySQL> INSERT INTO BORROW VALUES('00111', 1, 1, 100000),('10111', 2, 2,
150000),('20111', 3, 3, 10000),('30111', 4, 9, 5000);

Query OK, 4 rows affected (0.097 sec)

Records: 4  Duplicates: 0  Warnings: 0
```

2. List Cust_id along with customername –

```
MySQL> SELECT ID, NAME FROM CUSTOMER;

+----+---------+
| ID | NAME    |
+----+---------+
|  1 | Shreyas |
|  2 | Anil    |
|  3 | Piyush  |
|  4 | Aditya  |
|  5 | Ankit   |
|  6 | Patil   |
|  7 | Yash    |
|  8 | Shubham |
|  9 | Varun   |
| 10 | Nilesh  |
+----+---------+
10 rows in set (0.038 sec)
```

3. List Cust_id of depositors having amount greater than10000.

```
MySQL> SELECT CUST_ID FROM DEPOSIT WHERE AMOUNT > 10000;

+---------+
| CUST_ID |
+---------+
```

| Shubham |

+---------+

1 row in set (0.002 sec)

4 - List Account date of customer 'Anil' -

MySQL> SELECT ACCOUNT_OPENING_DATE FROM CUSTOMER WHERE CUST_NAME='Anil';

+----------------------+

| ACCOUNT_OPENING_DATE |

+----------------------+

| 2019-07-30 |

+----------------------+

1 row in set (0.002 sec)

5. List Cust_id of customers who have opened account after01/01/2016 –

MySQL> SELECT CUST_ID FROM CUSTOMER WHERE OPEN_DATE > '01/01/2016';

+---------+

| CUST_ID |

+---------+

| Piyush |

| Ankit |

| Aditya |

| Shreyas |

| Patil |

| Yash |

| Shubham |

| Nilesh |

| Anil |

+---------+

9 rows in set, 1 warning (0.002 sec)

7. Display customer name whose name starts from 'S'-

```
MySQL> SELECT CUST_NAME FROM CUSTOMER WHERE CUST_NAME LIKE 'S%';
+-----------+
| CUST_NAME |
+-----------+
| Shreyas   |
| Shubham   |
+-----------+
2 rows in set (0.002 sec)
```

9. **Display name whose length is 5 -**

```
MySQL> SELECT CUST_NAME FROM CUSTOMER WHERE LENGTH(CUST_NAME) = 5;
+-------+
| NAME  |
+-------+
| Ankit |
| Patil |
| Varun |
+-------+
3 rows in set (0.027 sec)
```

10. **Display Customer ID, Load No., Branch ID from borrow-**

```
MySQL> SELECT CUST_ID, LOAN_NO, BRANCH_ID FROM BORROW;
+---------+---------+-----------+
| CUST_ID | LOAN_NO | BRANCH_ID |
+---------+---------+-----------+
|       1 | 00111   |         1 |
|       2 | 10111   |         2 |
|       3 | 20111   |         3 |
|       4 | 30111   |         9 |
+---------+---------+-----------+
```

4 rows in set (0.002 sec)

14. **Select Customer name and amount who borrowed money.**

MySQL> SELECT CUST_NAME, AMOUNT FROM CUSTOMER C, BORROW B WHERE C.CUST_ID=B.CUST_ID;

+----------+--------+

| CUST_NAME | AMOUNT |

+----------+--------+

| Shreyas | 100000 |

| Anil | 150000 |

| Piyush | 10000 |

| Aditya | 5000 |

+----------+--------+

4 rows in set (0.002 sec)