NAME:AVINASH BHIWALE ROLL NO.:T1851050 PRN NO.:71901284J DIV:A

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 wherever necessary.

Objective :

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

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 a database
- INSERT – insert data into a table
- UPDATE – updates existing data within a table
- DELETE – deletes all records from a table

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

a) Inserting a single record

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 multiple record

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,column2.

VLUES (value1,

2. <u>SELECT: This is used to Retrieve data from one or more tables.</u>

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

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 WHERE condition;

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 multiple characters
- _ - The underscore represents a single character

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

Example:

```
SELECT * FROM Customers
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 * FROM Products
WHERE Price BETWEEN 10 AND 20;

f) WHERE with - *AND* LOGICAL Operator

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 becomes true.	(A <= B) is true.

Example: SELECT agent_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 a relation.

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 that relation.

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

Syntax: DELETE FROM relation_name;

Example: DELETE FROM std;

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

Syntax: DELETE FROM relation_name WHERE condition;

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 that data.
2. List Cust_id along with customer name.
3. List Cust_id of depositors having amount greater than 10000.
4. List account date of customer **'**Anil**'**.
5. List Cust_id of customers who have opened account after 01/01/2016.
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 their name.
10. List Cust_id, Loan no and Loan amount of borrowers.
11. List cust_id and C_name of depositors.
12. List all the customers who are depositors but not borrowers.
13. List all the customers who are both depositors and borrowers.
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 all depositors.
17. Update 10% to all depositors living in **'**Nagpur**'**.
18. Change living city of the **'**Nigdi**'** branch borrowers to Nagpur.

19. Delete branches having deposit from Nagpur.
20. Delete depositors of branches having number of customers between 1 and 3.
21. Delete depositors having deposit less than Rs500.

Conclusion:-

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

Output:

```
Enter password: ***************
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+----------------------------+
| Database                   |
+----------------------------+
| dbms_assignments           |
| information_schema         |
| mysql                      |
| performance_schema         |
| student_management_system  |
| sys                        |
+----------------------------+
6 rows in set (0.00 sec)

mysql> use dbms_assignments;
Database changed
mysql> create table CUSTOMER (Cust_id varchar(10) Primary key, C_name varchar(15) not
null, city varchar(10));
```

Query OK, 0 rows affected (0.09 sec)

mysql> create table BRANCH (Branch_id varchar(5) Primary Key, bname varchar(15), city varchar(10));

Query OK, 0 rows affected (0.04 sec)

mysql> create table DEPOSIT(

-> Acc_no varchar(10) Primary Key, Cust_id Varchar(10) Not NULL, Amount int, Branch_id Varchar(5), Open_date date);

Query OK, 0 rows affected (0.04 sec)

mysql> Create table BORROW(

-> Loan_no Varchar(5) Primary Key, Cust_id Varchar (10), Branch_id varchar(5), Amount int);

Query OK, 0 rows affected (0.07 sec)

mysql>(10) Primary key, C_name varchar(15) not null, city varchar(10));

mysql> desc CUSTOMER;

Field	Type	Null	Key	Default	Extra
Cust_id	varchar(10)	NO	PRI	NULL	
C_name	varchar(15)	NO		NULL	
city	varchar(10)	YES		NULL	

3 rows in set (0.03 sec)

mysql> desc BRANCH;

Field	Type	Null	Key	Default	Extra
Branch_id	varchar(5)	NO	PRI	NULL	
bname	varchar(15)	YES		NULL	
city	varchar(10)	YES		NULL	

3 rows in set (0.01 sec)

mysql> DESC DEPOSIT;

```
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| Acc_no      | varchar(10) | NO   | PRI | NULL    |       |
| Cust_id     | varchar(10) | NO   |     | NULL    |       |
| Amount      | int        | YES  |     | NULL    |       |
| Branch_id   | varchar(5)  | YES  |     | NULL    |       |
| Open_date   | date       | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql> DESC BORROW;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| Loan_no     | varchar(5)  | NO   | PRI | NULL    |       |
| Cust_id     | varchar(10) | YES  |     | NULL    |       |
| Branch_id   | varchar(5)  | YES  |     | NULL    |       |
| Amount      | int        | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

DROP DATABASE:

```
mysql> create database test1;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+--------------------------+
| Database                 |
+--------------------------+
| dbms_assignments         |
| information_schema       |
| mysql                    |
| performance_schema       |
| student_management_system |
| sys                      |
| test1                    |
+--------------------------+
```

7 rows in set (0.00 sec)

mysql> drop database test1;
Query OK, 0 rows affected (0.01 sec)

mysql> show databases;

```
+----------------------------+
| Database                   |
+----------------------------+
| dbms_assignments           |
| information_schema         |
| mysql                      |
| performance_schema         |
| student_management_system  |
| sys                        |
+----------------------------+
```

6 rows in set (0.00 sec)

mysql> ALTER TABLE CUSTOMER ADD ADDRESS
VARCHAR(20); Query OK, 0 rows affected (0.08 sec) Records: 0
Duplicates: 0 Warnings: 0

mysql> DESC CUSTOMER;

```
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Cust_id | varchar(10) | NO   | PRI | NULL    |       |
| C_name  | varchar(15) | NO   |     | NULL    |       |
| city    | varchar(10) | YES  |     | NULL    |       |
| ADDRESS | varchar(20) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

4 rows in set (0.01 sec)

6.2) mysql> ALTER TABLE CUSTOMER RENAME column C_name to Cust_name;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc customer;

```
+---------+-----------------+-------+---+---------+---------+
| Field   | Type            | Null  | Key | Default | Extra   |
+---------+-----------------+-------+-----+---------+---------+
| Cust_id | varchar(10)     | NO    | PRI | NULL    |         |
| Cust_name | varchar(15)   | NO    |     | NULL    |         |
| city    | varchar(10)     | YES   |     | NULL    |         |
| ADDRESS | varchar(20)     | YES   |     | NULL    |         |
+---------+-----------------+-------+-----+---------+---------+
4 rows in set (0.01 sec)
```

6.3) alter table CUSTOMER rename column ADDRESS to new_address;
Query OK, 0 rows affected (0.02 sec) Records: 0 Duplicates: 0 Warnings: 0

```
mysql> desc customer;
+---------+-----------------+-------+---+---------+---------+
| Field   | Type            | Null  | Key | Default | Extra   |
+---------+-----------------+-------+-----+---------+---------+
| Cust_id | varchar(10)     | NO    | PRI | NULL    |         |
| Cust_name | varchar(15)   | NO    |     | NULL    |         |
| city    | varchar(10)     | YES   |     | NULL    |         |
| new_address | varchar(20) | YES   |     | NULL    |         |
+---------+-----------------+-------+-----+---------+---------+
4 rows in set (0.00 sec)
```

6.4) mysql> alter table customer drop column new_address; Query OK, 0 rows affected (0.11 sec) Records: 0 Duplicates: 0 Warnings: 0

```
mysql> desc customer;
+---------+-----------------+-------+---+---------+---------+
| Field   | Type            | Null  | Key | Default | Extra   |
+---------+-----------------+-------+-----+---------+---------+
| Cust_id | varchar(10)     | NO    | PRI | NULL    |         |
| Cust_name | varchar(15)   | NO    |     | NULL    |         |
| city    | varchar(10)     | YES   |     | NULL    |         |
+---------+-----------------+-------+-----+---------+---------+
3 rows in set (0.01 sec)
```

6.5) mysql> alter table BRANCH rename to BRANCH1;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables
-> ;
+----------------------------+
| Tables_in_dbms_assignments |
+----------------------------+
| borrow |
| branch1 |
| customer |
| deposit |
+----------------------------+
4 rows in set (0.04 sec)

7) mysql> create user 'newuser'@'localhost' identified by
'newuser_pwd'; Query OK, 0 rows affected (0.02 sec)

mysql> select user from mysql.user;
+ +
| user |
+ +
| akash136 |
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| newuser |
| root |
| student_management_system |
+ +
7 rows in set (0.00 sec)

mysql> grant create on dbms_assignments.customer to 'newuser'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.02 sec)

```
mysql> show grants for 'newuser'@'localhost';
+-----------------------------------------------------------------+
| Grants for newuser@localhost                                    |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `newuser`@`localhost`                     |
| GRANT CREATE ON `dbms_assignments`.`customer` TO `newuser`@`localhost` |
+-----------------------------------------------------------------
+ 2 rows in set (0.00 sec)

mysql> revoke create on dbms_assignments.customer from 'newuser'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> show grants for 'newuser'@'localhost';
+-----------------------------------------------+
| Grants for newuser@localhost                  |
+-----------------------------------------------+
| GRANT USAGE ON *.* TO `newuser`@`localhost` |
+---------------------------------------------+ 1 row in set
(0.00 sec)
mysql> insert into customer values (1, 'Akash', 'Pune'), (2, 'Raj', 'Delhi');
Query OK, 2 rows affected (0.01 sec) Records: 2 Duplicates: 0
Warnings: 0

mysql> select * from customer;
+---------+-----------+-------+
| Cust_id | Cust_name | city |
+---------+-----------+-------+
| 1       | Akash     | Pune  |
| 2       | Raj       | Delhi |
+---------+-----------+-------+
2 rows in set (0.00 sec)

mysql> truncate table customer;
Query OK, 0 rows affected (0.06 sec)

mysql> select * from customer;
Empty set (0.01 sec)
```

8) Drop table:

```
mysql> drop table customer;
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+---------------------------+
| Tables_in_dbms_assignments |
+---------------------------+
| borrow                    |
| branch1                   |
| deposit                   |
+---------------------------+
3 rows in set (0.01 sec)
```

ER Diagram: