**BASIC SQL QUERIES**

**EXPT NO: 1 DATE:**

**AIM:** To study basic SQL queries.

**THEORY:**

SQL is Structured Query Language, which is a computer language for storing,   
manipulating and retrieving data stored in a relational database. SQL is the   
standard language for Relational Database System. All the Relational Database   
Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase,   
Informix, Postgres and SQL Server use SQL as their standard database   
language.   
SQL is a hybrid of 4 languages which are Data Definition Language (DDL), Data   
Manipulation Language (DML), Data Control Language (DCL), Data Query   
Language (DQL).   
   
There are 2 types of databases:   
i) Relational database   
ii) Non-relational database

|  |  |
| --- | --- |
| **Relational Database** | **Non-Relational Database** |
| A Relational database management system (RDBMS) is a database  management system (DBMS) that is based on the relational model as  introduced by E. F. Codd. It consists of Data arranged in the form of  tables(relations), rows(tuples) and columns(attributes). Relational databases  work best when the data they contain doesn’t change very often, and when  accuracy is crucial. Relational databases are, for instance, often found in  financial applications. Popular examples of standard relational databases  include Microsoft SQL Server, Oracle Database, MySQL and IBM DB2. | Non-relational databases (often called NoSQL databases) are different from  traditional relational databases in that they store their data in a non-tabular  form. Instead, non-relational databases might be based on data structures like  documents. A document can be highly detailed while containing a range of  different types of information in different formats. This ability to digest and  organize various types of information side by side makes non-relational  databases much more flexible than relational databases. MongoDB, Apache  Cassandra, Redis, Couchbase and Apache HBase. They are best for Rapid  Application Development. |

MySQL

* MySQL is a widely used relational database management system (RDBMS).
* MySQL is free and open-source.
* MySQL is ideal for both small and large applications.
* MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.
* It allows us to implement database operations on tables, rows, columns, and indexes. It defines the database relationship in the form of tables (collection of rows and columns), also known as relations. It provides the Referential Integrity between rows or columns of various tables. It allows us to updates the table indexes automatically. It uses many SQL queries and combines useful information from multiple tables for the end-users.

|  |  |
| --- | --- |
| **COMMAND** | **FUNCTION** |
| CREATE | The CREATE DATABASE statement is used to create a new SQL database.     Syntax CREATE DATABASE databasename;    The CREATE TABLE statement is used to create a new table in a database.     Syntax CREATE TABLE table\_name (   column1 datatype,   column2 datatype,   column3 datatype,   .... ); |
| SELECT | The SELECT statement is used to select data from a database.  The data returned is stored in a result table, called the result-set.     Syntax SELECT column1, column2, ... FROM table\_name;    If we want to select all the fields available in the table, use the following syntax:    SELECT \* FROM table\_name; |
| SELECT WHERE | The WHERE clause is used to filter records.  It is used to extract only those records that fulfill a specified condition.     Syntax SELECT \* FROM table\_name WHERE condition; |
| SELECT WHERE AND | The WHERE clause can be combined with the AND operators.  The AND operator are used to filter records based on more than one condition:  The AND operator displays a record if all the conditions separated by AND are TRUE.     Syntax SELECT column1, column2, ... FROM table\_name   WHERE condition1 AND condition2 AND condition3 ...; |
| SELECT WHERE OR | The WHERE clause can be combined with the OR operator.  The OR operator are used to filter records based on more than one condition:  The OR operator displays a record if any of the conditions separated by OR is TRUE.     Syntax SELECT column1, column2, ... FROM table\_name   WHERE condition1 OR condition2 OR condition3 ...; |
| AGGREGATE  FUNCTIONS | AVG() Returns the average of the values in the selected column  COUNT() Returns the number of rows returned for a selection  MAX() Returns the maximum value for a column  MIN() Returns the minimum value of a column  SUM() Returns the sum of the values in a specified column |
| LIKE | The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.  There are two wildcards often used in conjunction with the LIKE operator:  The percent sign (%) represents zero, one, or multiple characters  The underscore sign (\_) represents one, single character  The percent sign and the underscore can also be used in combinations     Syntax SELECT column1, column2, ... FROM table\_name WHERE columnN LIKE pattern; |
| DISTINCT | The SELECT DISTINCT statement is used to return only distinct (different) values.  A column often contains many duplicate values, and sometimes we only want to list the different values.     Syntax SELECT DISTINCT column1, column2, ... FROM table\_name; |

**QUERIES**

1) Create the following tables:

1. Book:

|  |  |  |
| --- | --- | --- |
| **Name** | **Constraint** | **Data Type** |
| BookId | Primary key | int |
| Title |  | varchar(25) |
| PublisherName |  | varchar(25) |
| Publish\_date |  | date |

1. Book\_authors

|  |  |  |
| --- | --- | --- |
| **Name** | **Constraint** | **Data Type** |
| BookId | Primary key | int |
| Authorname | Primary key | varchar(25) |

1. Publisher

|  |  |  |
| --- | --- | --- |
| **Name** | **Constraint** | **Data Type** |
| Name | Primary key | varchar(25) |
| Addr |  | varchar(25) |
| Phone |  | int |

1. book\_copies

|  |  |  |
| --- | --- | --- |
| **Name** | **Constraint** | **Data Type** |
| Bookid | Primary key | int |
| Branch |  | varchar(25) |
| no\_of\_copies |  | int |

1. Insert atleast 5 rows in each table. (Insert data in such a way that every query below should generate an output).
2. Answer the following queries:
3. List information about all books in the database.
4. List information about all publishers in the database.
5. List the bookid and title of all the books in the database.
6. List the books with more than 1000 copies at the 'farmagudi' branch.
7. List name and phone number of all the publishers.
8. List the books with their authors.
9. List the names of all the authors in the database.
10. List names of all authors whose names are 5 characters long.

1. List names of all authors whose names end with letter 'D'.
2. List names of all books whose titles start with 'A' or 'G'.
3. Select all branches whose names fall between 'A' and 'C' alphabetical range.
4. List all the book details in the ascending order of no of copies.
5. List the books branchwise(i.e. In the alphabetical order of the branch and within the branch in the descending order of the no\_of\_copies).
6. List the books in the descending order of publish date.
7. List the names of books published atleast 5 years ago.

p) List the books published in the month of 'July'.

**CONCLUSION:**

All fundamental SQL commands were successfully comprehended and implemented.

The experiment demonstrated the basic knowledge of database creation, table creation, data insertion, and SQL queries.