# **Acropolis Institute of Technology & Research**



Subject: Database Management System (CY-405)

Name: Anushka Bagwan

Enrollment no: 0827CY221013

Branch: CY(Cyber Security)

Semester: 4th (2nd Year)

Submitted to: Prof. Nidhi Nigam

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Experiment | Date of Experiment | Date of submission | Grade |
| 1. | To study of DBMS and RDBMS,its characteristics and database softwares. | 11/03/24 | 18/03/24 |  |
| 2. | Introduction SQL-SQL Lite and installation |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**LAB 1 – INTRODUCTION TO DBMS**

Objective:To study of DBMS and RDBMS.

DBMS: Database Management System is a software or technology used to manage data from a database. Some popular databases are MySQL, Oracle, MongoDB, etc. DBMS provides many operations e.g. creating a database, Storing in the database, updating an existing database, delete from the database. DBMS is a system that enables you to store, modify and retrieve data in an organized way. It also provides security to the database.

CHARACTERISTICS OF DBMS:

* A database management system is able to store any kind of data in a database.
* The database management system has to support ACID (atomicity, consistency, isolation, durability) properties.
* The Database management system allows so many users to access databases at the same time.
* Backup and recovery are the two main methods which allow users to protect the data from damage or loss.
* It also provides multiple views for different users in a single organization.
* It follows the concept of normalization which is helpful to minimize the redundancy of a relation.

TYPES OF DBMS

## **1) Centralized Database**

It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications.

## **2) Distributed Database**

Unlike a centralized database system, in distributed systems, data is distributed among different database systems of an organization. These database systems are connected via communication links. Such links help the end-users to access the data easily.

## **3) Relational Database**

This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation).

## **4) NoSQL Database**

Non-SQL/Not Only SQL is a type of database that is used for storing a wide range of data sets. It is not a relational database as it stores data not only in tabular form but in several different ways.

## **5) Cloud Database**

A type of database where data is stored in a virtual environment and executes over the cloud computing platform. It provides users with various cloud computing services (SaaS, PaaS, IaaS, etc.) for accessing the database.

## **6) Object-oriented Databases**

The type of database that uses the object-based data model approach for storing data in the database system. The data is represented and stored as objects which are similar to the objects used in the object-oriented programming language.

RDBMS:

**RDBMS stands for relational database management system and it is a software system and it is a software system that is used to store only data in the form oftables.Data is handled and stored in rows and columns ,which is reffered to as tuples and attributes.**

DIFFERENCE between DBMS AND RDBMS:

|  |  |
| --- | --- |
| **DBMS** | **RDBMS** |
| Data is stored in a database management system (DBMS) as a file | Tables are used to store information |
| Data is stored in a database management system (DBMS) in either a navigational or hierarchical format | RDBMS employs a tabular format, with column names as headers and associated data as rows |
| Only a single user is supported by the DBMS | It may be used by numerous people |
| Software and hardware requirements are minimal | Higher hardware and software requirements are required |
| Normalization is not supported by DBMS. | A relational database management system (RDBMS) can be normalised |
| Distributed databases are not supported by DBMS | Distributed databases are supported by RBMS |
| The DBMS system is mostly used to manage tiny amounts of data | The RDBMS database is built to manage a vast volume of data |

Data softwares used in DBMS

* [MySQL](https://www.guru99.com/sql.html)
* [Microsoft Access](https://www.guru99.com/ms-access-tutorial.html)
* Oracle
* [PostgreSQL](https://www.guru99.com/postgresql-tutorial.html)
* dBASE
* FoxPro
* [SQLite](https://www.guru99.com/sqlite-tutorial.html)
* IBM DB2
* LibreOffice Base
* [MariaDB](https://www.guru99.com/mariadb-tutorial-install.html)
* Microsoft [SQL Server](https://www.guru99.com/ms-sql-server-tutorial.html)

# **LAB 2-INTRODUCTION TO SQL**

**Objective:** Introduction to SQL-SQL Lite and installation.

**SQL:** Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL statements to store, update, remove, search, and retrieve information from the database. You can also use SQL to maintain and optimize database performance.

## **Why is SQL important?**

Structured query language (SQL) is a popular query language that is frequently used in all types of applications. Data analysts and developers learn and use SQL because it integrates well with different programming languages. For example, they can embed SQL queries with the Java programming language to build high-performing data processing applications with major SQL database systems such as Oracle or MS SQL Server. SQL is also fairly easy to learn as it uses common English keywords in its statements

### **History of SQL**

SQL was invented in the 1970s based on the relational data model. It was initially known as the structured English query language (SEQUEL). The term was later shortened to SQL. Oracle, formerly known as Relational Software, became the first vendor to offer a commercial SQL relational database management system.

## **components of a SQL system**

Relational database management systems use structured query language (SQL) to store and manage data. The system stores multiple database tables that relate to each other. MS SQL Server, MySQL, or MS Access are examples of relational database management systems. The following are the components of such a system.

### **SQL table**

A SQL table is the basic element of a relational database. The SQL database table consists of rows and columns. Database engineers create relationships between multiple database tables to optimize data storage space.

For example, the database engineer creates a SQL table for products in a store:

|  |  |  |
| --- | --- | --- |
| **Product ID** | **Product Name** | **Color ID** |
| 0001 | Mattress | Color 1 |
| 0002 | Pillow | Color 2 |

Then the database engineer links the product table to the color table with the Color ID:

|  |  |
| --- | --- |
| **Color ID** | **Color Name** |
| Color 1 | Blue |
| Color 2 | Red |

### **SQL statements**

SQL statements, or SQL queries, are valid instructions that relational database management systems understand. Software developers build SQL statements by using different SQL language elements. SQL language elements are components such as identifiers, variables, and search conditions that form a correct SQL statement.

For example, the following SQL statement uses a SQL INSERT command to store Mattress Brand A, priced $499, into a table named Mattress\_table, with column names brand\_name and cost:

INSERT INTO Mattress\_table (brand\_name, cost)

VALUES(‘A’,’499’);

### **Stored procedures**

Stored procedures are a collection of one or more SQL statements stored in the relational database. Software developers use stored procedures to improve efficiency and performance. For example, they can create a stored procedure for updating sales tables instead of writing the same SQL statement in different applications.

## **SQL Lite:** SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a popular choice as an embedded database for local/client storage in application software such as web browsers. It is also used in many other applications that need a lightweight, embedded database.

### **SQLite History :** SQLite was created in the year 2000 by D. Richard Hipp, who continues to lead the development of the software today. SQLite was designed to be a lightweight and simple database engine that could be easily embedded into other applications. It was created as an alternative to more complex and heavyweight database engines, such as MySQL and PostgreSQL. Over the years, SQLite has gained widespread adoption and is now one of the most widely used database engines in the world. It is used in many applications, including web browsers, mobile phones, and a wide variety of other software.

### **Why we use SQLite?**

There are several reasons why you might choose to use SQLite in your project:

1. **Ease of use:** SQLite is very easy to get started with, as it requires no setup or configuration. You can simply include the library in your project and start using it.
2. **Embeddability:**SQLite is designed to be embedded into other applications. It is a self-contained, serverless database engine, which means you can include it in your application without the need for a separate database server.
3. **Lightweight:**SQLite is a very lightweight database engine, with a small library size (typically less than 1MB). This makes it well-suited for use in applications where the database is embedded directly into the application binary, such as mobile apps.
4. **Serverless:**As mentioned earlier, SQLite is a serverless database engine, which means there is no need to set up and maintain a separate database server process. This makes it easy to deploy and manage, as there are no additional dependencies to worry about.
5. **Cross-platform:**SQLite is available on many platforms, including Linux, macOS, and Windows, making it a good choice for cross-platform development.
6. **Standalone:** SQLite stores all of the data in a single file on the filesystem, which makes it easy to copy or backup the database.
7. **High reliability:**SQLite has been widely tested and used in production systems for many years, and has a reputation for being a reliable and robust database engine.

## **Installation of SQLite :**

* **Step 1** − Go to [SQLite download page](https://www.sqlite.org/download.html), and download precompiled binaries from Windows section.
* **Step 2** − Download sqlite-shell-win32-\*.zip and sqlite-dll-win32-\*.zip zipped files.
* **Step 3** − Create a folder C:\>sqlite and unzip above two zipped files in this folder, which will give you sqlite3.def, sqlite3.dll and sqlite3.exe files.
* **Step 4** − Add C:\>sqlite in your PATH environment variable and finally go to the command prompt and issue sqlite3 command, which should display the following result.