Intro to Databases

Before we learn about SQL injections, we need to learn more about databases and Structured Query Language (SQL), which databases will perform the necessary queries. Web applications utilize back-end databases to store various content and information related to the web application. This can be core web application assets like images and files, content like posts and updates, or user data like usernames and passwords.

There are many different types of databases, each of which fits a particular type of use. Traditionally, an application used file-based databases, which was very slow with the increase in size. This led to the adoption of `Database Management Systems` (`DBMS`).

Database Management Systems

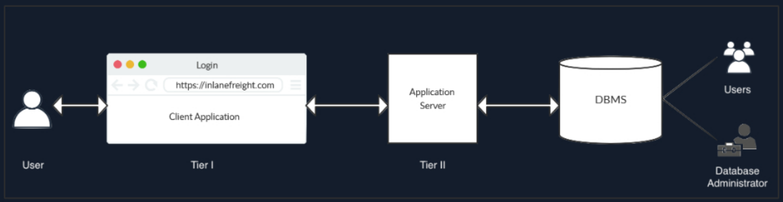
A Database Management System (DBMS) helps create, define, host, and manage databases. Various kinds of DBMS were designed over time, such as file-based, Relational DBMS (RDBMS), NoSQL, Graph based, and Key/Value stores.

There are multiple ways to interact with a DBMS, such as command-line tools, graphical interfaces, or even APIs (Application Programming Interfaces). DBMS is used in various banking, finance, and education sectors to record large amounts of data. Some of the essential features of a DBMS include:

Feature	Description
`Concurrency`	A real-world application might have multiple users interacting with it simultaneously. A DBMS makes sure that these concurrent interactions succeed without corrupting or losing any data.
`Consistency`	With so many concurrent interactions, the DBMS needs to ensure that the data remains consistent and valid throughout the database.
`Security`	DBMS provides fine-grained security controls through user authentication and permissions. This will prevent unauthorized viewing or editing of sensitive data.
`Reliability`	It is easy to backup databases and rolls them back to a previous state in case of data loss or a breach.
`Structured Query Language`	SQL simplifies user interaction with the database with an intuitive syntax supporting various operations.

Architecture

The diagram below details a two-tiered architecture.



`Tier I` usually consists of client-side applications such as websites or GUI programs. These applications consist of high-level interactions such as user login or commenting. The data from these interactions is passed to `Tier II` through API calls or other requests.

The second tier is the middleware, which interprets these events and puts them in a form required by the DBMS. Finally, the application layer uses specific libraries and drivers based on the type of DBMS to interact with them. The DBMS receives queries from the second tier and performs the requested operations. These operations could include insertion, retrieval, deletion, or updating of data. After processing, the DBMS returns any requested data or error codes in the event of invalid queries.

It is possible to host the application server as well as the DBMS on the same host. However, databases with large amounts of data supporting many users are typically hosted separately to improve performance and scalability.

Previous Next Mark Complete & Next

Cheat Sheet

Table of Contents

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left