**What Is a Database?**

A **database** is an organised collection of data stored electronically. It allows efficient access, modification, and management of data using software called a **Database Management System (DBMS)**. Databases can store:

* **Structured data** (like names and dates),
* **Semi-structured data** (like JSON or XML), and
* **Unstructured data** (like videos and images).

**Key Features of All Databases**

* **Organisation**: Data is structured logically, often using tables or objects.
* **Efficiency**: Queries and updates are fast and accurate.
* **Scalability**: Can handle growing data over time.
* **Security**: Includes features for authentication, access control, and backups.

**Database main types**

Here’s a breakdown of the database types the article discusses, along with their pros and cons:

**1. Hierarchical Databases**

* **Structure**: Tree-like, with parent-child relationships.
* **Use Case**: Simple data with clear hierarchy (e.g., university departments).
* **Pros**: Fast and efficient when hierarchy is known.
* **Cons**: Inflexible and hard to modify if the structure changes.

**2. Network Databases**

* **Structure**: Web-like, allows many-to-many relationships.
* **Use Case**: Complex data with multiple relationships (e.g., students in multiple clubs).
* **Pros**: More flexible than hierarchical.
* **Cons**: Hard to design and maintain.

**3. Object-Oriented Databases**

* **Structure**: Data is stored as objects (like in OOP languages).
* **Use Case**: Multimedia, CAD systems, complex apps.
* **Pros**: Supports methods and reusable objects.
* **Cons**: Requires OOP knowledge, not widely supported.

**4. Relational Databases (RDBMS)**

* **Structure**: Tables (rows = records, columns = attributes).
* **Use Case**: Business, web apps, systems requiring data validation.
* **Pros**: Simple, reliable, supports relationships with **primary and foreign keys**.
* **Cons**: Schema must be designed carefully; less flexible with unstructured data.

**5. Cloud Databases**

* **Hosted online** using platforms like AWS or Azure.
* **Pros**: Scalable, low maintenance, pay-as-you-go.
* **Cons**: Dependent on internet access; potential security concerns.

**6. Centralised Databases**

* **All data stored in one location/server**.
* **Pros**: High security, consistency.
* **Cons**: Slower response, harder to scale.

**7. Personal Databases**

* Designed for single users (e.g., Microsoft Access, SQLite).
* **Pros**: Easy to use, lightweight.
* **Cons**: Not suitable for large systems or teams.

**8. Operational Databases**

* Store **real-time, transactional data** (e.g., for SAP systems).
* **Pros**: Supports fast updates and daily operations.
* **Cons**: Needs constant monitoring.

**9. NoSQL Databases**

* **Structure**: Key-value pairs, documents, graphs—**not** tables.
* **Use Case**: Unstructured data, fast access, flexible schema.
* **Pros**: Easy to scale, handles big data.
* **Cons**: Less support for complex relationships, limited GUI, harder backups.

**Final Takeaways**

* **Relational Databases** are the **most widely used**, especially in enterprise and compliance-heavy environments.
* **NoSQL** is good for flexibility and unstructured data.
* The choice of database depends on **what kind of data you need to store**, **how it relates**, and **how you plan to query or report it**.

**What is useful for my project**

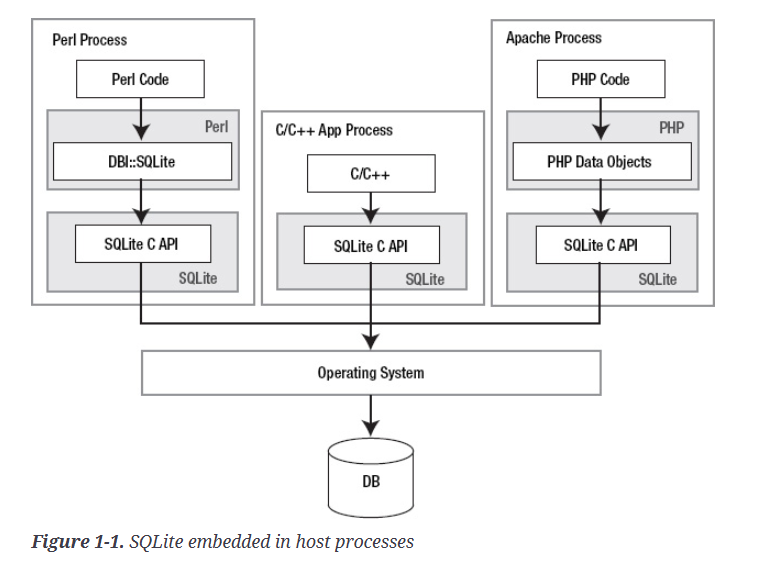
* **relational databases** are ideal for **structured, auditable data**, which your system for inspection logs requires.
* **NoSQL and hierarchical models are less suited** due to weaker support for relationships and validation.

**Article:**

Types of Databases. (2025). GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/types-of-databases/> (Accessed 15 April 2025)

SQLite

SQLite is an open source embedded relational database.



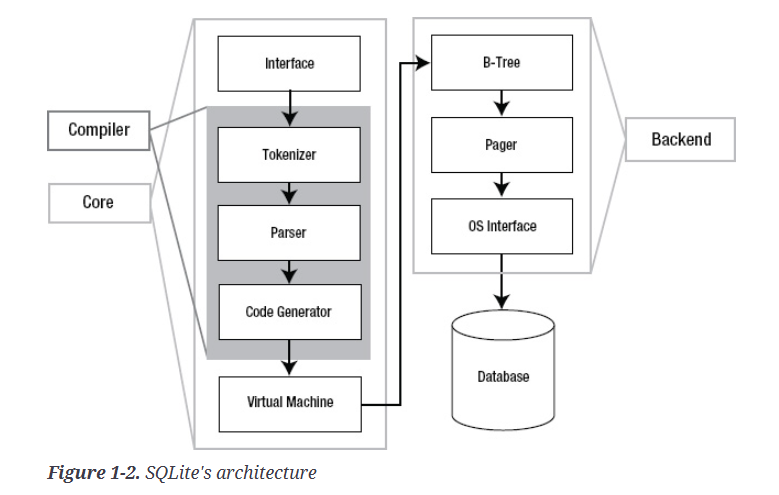
It is a database, a programming library, and a command-line tool, as well an excellent learning tool that provides a good introduction to relational databases. There are many ways to use it - in embedded environments, websites, operating system services, scripts, and applications.

SQLite can use variety of scripting languages such as Perl, Python, TCL, and Ruby.

Where SQLite is applied:

* in Sun’s Solaris operating systems
* in Mozilla Project mozStorage C++ and Mozilla JavaScript API layer
* besides SQLite is used in the native Symbian OS platform to provide SQL database support
* Google used SQLite in the Android mobile phone operating system
* SQLite is used extensively in native Android devices
* Although the SQLite is not as popular as SQL Server or MySQL, it is used in variety of consumer products with examples like: D-Link, Squeezebox music player, and Phillips GoGear (personal music player)

SQLite architecture:



SQLite architecture is made up of eight separate modules (modular), grouped within three main subsystems. The purpose of them modules is to divide query processing into discrete tasks that work one by one.