



SQL and NoSql

PALLAVI VAIDYA

Agenda

Understand what SQL and NoSQL databases are

Identify key differences between SQL and NoSQL

Recognize popular database systems

Learn about ACID properties in SQL databases

Review basic SQL commands

Databases

A **database** is an organized collection of data.

Helps store, manage, and retrieve information efficiently.

Managed using **Database Management Systems (DBMS)**.

What is SQL?

SQL (Structured Query Language) databases are **relational**.

Data stored in **tables** (rows and columns).

Use **SQL** to define and manipulate data.

Examples: MySQL, PostgreSQL, Oracle, SQL Server.

NOSQL

NoSQL means “Not Only SQL.”

Designed for **unstructured or semi-structured data**.

Schema-less, flexible, and scalable.

Examples: MongoDB, Cassandra, Redis, CouchDB.

Key Differences Between SQL and NoSQL

Feature	SQL	NoSQL
Data Model	Relational (tables)	Non-relational (document, key-value, graph, column)
Schema	Fixed	Dynamic/Flexible
Scalability	Vertical	Horizontal
Query Language	SQL	Varies (JSON, APIs, etc.)
ACID	Strong	Often BASE (Eventually consistent)
Best for	Structured data	Large, unstructured data

Popular SQL Databases

MySQL – Open source, widely used

PostgreSQL – Advanced, supports JSON

Oracle DB – Enterprise-grade, robust

Microsoft SQL Server – Business-oriented

SQLite - Lightweight SQL Database, ideal for mobile apps, IoT devices, and testing environments.

SQLite

SQLite is a **self-contained, serverless, and zero-configuration** SQL database engine.

It stores the entire database in a **single file** on the local disk.

Advantages of SQLite

Embedded directly into applications (no separate server required).

Supports **standard SQL syntax**.

Cross-platform and **lightweight** (less than 1 MB).

Easy setup — no installation needed.

Fast for small to medium data volumes.

Great for **prototyping** and **local storage**.

Using SQLite in Python

SQLite comes built-in with Python — no installation required.

You can use the **sqlite3 module** to interact with databases.

Using SQLite in Python: Common Workflow

Connect to (or create) a database file

Create a cursor object

Execute SQL commands

Commit and close the connection

Popular NOSQL databases

MongoDB – Document-oriented (JSON-like)

Cassandra – High availability, distributed

Redis – In-memory key-value store

CouchDB – Uses HTTP/JSON for access

ACID Properties in SQL

A – Atomicity: All or nothing transactions

C – Consistency: Database rules maintained

I – Isolation: Concurrent transactions don't interfere

D – Durability: Data persists after system failure

Basic SQL commands

Category	Command	Description
Data Definition	CREATE, ALTER, DROP	Manage structure
Data Manipulation	INSERT, UPDATE, DELETE	Modify data
Data Query	SELECT	Retrieve data
Data Control	GRANT, REVOKE	Manage permissions

References

<https://www.mongodb.com/nosql-explained>

<https://www.postgresql.org>

<https://www.mysql.com>

<https://www.oracle.com/database/>