**DIFFERENT TYPES OF DATABASES**

Database technology has changed and evolved over the years. Relational, NoSQL, hierarchical…it can start to get confusing. Storing data doesn’t have to be a headache. If you’re trying to pick the right database for your organization, here’s a guide to the properties and uses of each type.

## What are the types of databases?

### 1. Relational databases

Relational databases have been around since the 1970s. The name comes from the way that data is stored in multiple, related tables. Within the tables, data is stored in rows and columns. The relational database management system (RDBMS) is the program that allows you to create, update, and administer a relational database. Structured Query Language (SQL) is the most common language for reading, creating, updating and deleting data. Relational databases are very reliable. They are compliant with ACID (Atomicity, Consistency, Isolation, Durability), which is a standard set of properties for reliable database transactions. Relational databases work well with structured data. Organizations that have a lot of unstructured or semi-structured data should not be considering a relational database.

Examples: Microsoft SQL Server, Oracle Database, MySQL, PostgreSQL and IBM Db2

### 2. NoSQL databases

NoSQL is a broad category that includes any database that doesn’t use SQL as its primary data access language. These types of databases are also sometimes referred to as non-relational databases. Unlike in relational databases, data in a NoSQL database doesn’t have to conform to a pre-defined schema, so these types of databases are great for organizations seeking to store unstructured or semi-structured data. One advantage of NoSQL databases is that developers can make changes to the database on the fly, without affecting applications that are using the database.

Examples: Apache Cassandra, MongoDB, CouchDB, and CouchBase

**3. Network Database Model**

A network database is a type of database model wherein multiple member records or files can be linked to multiple owner files and vice versa. The model can be viewed as an upside-down tree where each member information is the branch linked to the owner, which is the bottom of the tree. Essentially, relationships are in a net-like form where a single element can point to multiple data elements and can itself be pointed to by multiple data elements.

The network database model allows each record to have multiple parent and multiple child records, which, when visualized, form a web-like structure of networked records. In contrast, a hierarchical model data member can only have a single parent record but can have many child records.  
  
This property of having multiple links applies in two ways: the schema and the database itself can be seen as a generalized graph of record types that are connected by relationship types. The main advantage of a network database is that it allows for a more natural modeling of relationships between records or entities, as opposed to the hierarchical model. However, the relational database model has started to win over both the network and the hierarchical models because its added flexibility and productivity has become more evident as hardware technology has become faster.

**Examples:** Integrated Data Store (IDS), IDMS (Integrated Database Management System)

### 4. Key-value databases

One of the simplest types of NoSQL databases, key-value databases save data as a group of key-value pairs made up of two data items each. They’re also sometimes referred to as a key-value store. Key-value databases are highly scalable and can handle high volumes of traffic, making them ideal for processes such as session management for web applications, user sessions for massive multi-player online games, and online shopping carts.

Examples: Amazon DynamoDB, Redis

### 5. Hierarchical databases

Hierarchical databases use a parent-child model to store data. If you were to draw a picture of a hierarchical database, it would look like a family tree, with one object on top branching down to multiple objects beneath it. The one-to-many format is rigid, so child records can’t have more than one parent record. Originally developed by IBM in the early 1960s, hierarchical databases are commonly used to support high-performance and high availability applications.

Examples: IBM Information Management System (IMS), Windows Registry

### 6. Graph databases

Graph databases are a type of NoSQL database that are based on graph theory. Graph-Oriented Database Management Systems (DBMS) software is designed to identify and work with the connections between data points. Therefore graph databases are often used to analyze the relationships between heterogeneous data points, such as in fraud prevention or for mining data about customers from social media.

Examples: Datastax Enterprise Graph, Neo4J

### 7. Cloud databases

A cloud database refers to any database that’s designed to run in the cloud. Like other cloud-based applications, cloud databases offer flexibility and scalability, along with high availability. Cloud databases are also often low-maintenance, since many are offered via a SaaS model.

Examples: Microsoft Azure SQL Database, Amazon Relational Database Service, Oracle Autonomous Database.

### 8. Columnar databases

Also referred to as column data stores, columnar databases store data in columns rather than rows. These types of databases are often used in data warehouses because they’re great at handling analytical queries. When you’re querying a columnar database, it essentially ignores all of the data that doesn’t apply to the query, because you can retrieve the information from only the columns you want.

Examples: Google BigQuery, Cassandra, HBase, MariaDB, Azure SQL Data Warehouse

### 9. Wide column databases

Wide column databases, also known as wide column stores, are schema-agnostic. Data is stored in column families, rather than in rows and columns. Highly scalable, wide column databases can handle petabytes of data, making them ideal for supporting real-time big data applications.

Examples: BigTable, Apache Cassandra and Scylla

### 10. Object-oriented databases

An object-oriented database is based on object-oriented programming, so data and all of its attributes, are tied together as an object. Object-oriented databases are managed by object-oriented database management systems (OODBMS). These databases work well with object-oriented programming languages, such as C++ and Java. Like relational databases, object-oriented databases conform to ACID standards.

Examples: Wakanda, ObjectStore

### 11. Document databases

Document databases, also known as document stores, use JSON-like documents to model data instead of rows and columns. Sometimes referred to as document-oriented databases, document databases are designed to store and manage document-oriented information, also referred to as semi-structured data. Document databases are simple and scalable, making them useful for mobile apps that need fast iterations.

Examples: MongoDB, Amazon DocumentDB, Apache CouchDB

### 12. Time series databases

A time series database is a database optimized for time-stamped, or time series, data. Examples of this type of data include network data, sensor data, and application performance monitoring data. All of those Internet of Things sensors that are getting attached to everything put out a constant stream of time series data.

Examples: Druid, eXtremeDB, InfluxDB

**MySQL:** is an open-source Relational DataBase Management System (RDBMS).

* Its name is a combination of "My", the name of co-founder Micheal Wideneu's daughter, and "SQL", the abbreviation for Structured Query Language.
* A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data.
* SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database.
* In addition to relational databases and SQL, an RDBMS like 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.
* MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.
* MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Miccrosystems (now Oracle Corporation).
* In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.
* MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.
* MySQL is a component of the LAMP web application software stack, which is an acronym for Linux*, Apache, MySQL, Perl/PHP/Python*.
* MySQL is used by many database-driven web applications, including Drupal, Joomla, PHP-BB, and WordPress.
* MySQL is also used by many popular websites, including Facebook, Flicker, MediaWiki, Twitter, and YouTube.

SQLite: is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine.

* It is the most used database engine in the world. It is an in-process library and its code is publicly available.
* It is free for use for any purpose, commercial or private.
* It is basically an embedded SQL database engine.
* Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL.
* The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems.
* Due to all these features, it is a popular choice as an Application File Format.

Features of SQLite

* The transactions follow ACID properties i.e. atomicity, consistency, isolation, and durability even after system crashes and power failures.
* The configuration process is very easy, no setup or administration needed.
* All the features of SQL are implemented in it with some additional features like partial indexes, indexes on expressions, JSON, and common table expressions.
* Sometimes it is faster than the direct file system I/O.
* It supports terabyte-sized databases and gigabyte-sized strings and blobs.
* Almost all OS supports SQLite like Android, BSD, iOS, Linux, Mac, Solaris, VxWorks, and Windows (Win32, WinCE, etc. It is very much easy to port to other systems.
* Complete database can be stored in a single cross-platform disk file.

Applications of SQLite

* Due to its small code print and efficient usage of memory, it is the popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets.
* It is used as an alternative for open to writing XML, JSON, CSV or some proprietary format into disk files used by the application.
* As it has no complication for configuration and easily stores file in an ordinary disk file, so it can be used as a database for small to medium sized websites.
* It is faster and accessible through a wide variety of third-party tools, so it has great application in different software platforms.

Some DDL and DML Commands:

It is same as compared to previous technology like MySQL, Oracle.

* Creating Table:  
  CREATE TABLE STUDENT(  
  ID INT PRIMARY KEY NOT NULL,  
  NAME TEXT NOT NULL,  
  AGE INT NOT NULL,  
  PLACE CHAR(50),  
  FEES REAL  
  );
* Insert Command:  
  INSERT INTO STUDENT (ID, NAME, AGE, PLACE, FEES)  
  VALUES (1, 'Shamjad', 21, 'Trivandrum', 25000.00);
* Drop Table:  
  Drop Table Student;

Disadvantages of SQLite

* It is only used where there is low to medium traffic requests are there.
* The database size is restricted i.e. it is 2GB in most cases.