

Database Principles Appendix A

The Sedona Conference

Copyright 2022, The Sedona Conference. All rights reserved.



Introduction

There are many types of databases in use today. Historically a database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (“DBMS”). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. Database management has evolved significantly and this appendix will cover the most common database platforms used in business today.

Relational Database

Relational Database Management Systems (“RDBMS”) are the most common and rigid of all database types. RDBMS was created for mapping datasets at IBM in 1970. RDBMS is the basis for common database platforms such as SQL Server, Oracle, and MySQL.

Data in an RDBMS is organized in tables, columns, and rows. Each table contains data that is stored vertically in columns, also known as fields, and horizontally as rows, also known as records. rows and columns. Each row is considered a single record listed horizontally. Each column is a field that describes what is being stored which is stored “vertically.” The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

RDBMS’ make up more than 70% of all databases in production as of 2022, though that number is dropping. They are used for a wide variety of applications, old and new, and are frequently the underlying technology in legacy systems. Example RDBMS engines include MSSQL, MySQL, and Oracle.

The advantages of an RDBMS is its scalability, security, industry standard and are generally easy to understand.

The disadvantages are the lack support for complex datatypes (e.g., images/video), complex data analysis routines and the data to be stored must be known and must conform to the database structure for it to be added as tables.

Object Oriented Database

Object Oriented Databases Management Systems (“OODBMS”) are often used in devices, compiled software, and real-time systems. OODBMS are different in structure from Relational Databases. Because the world rarely looks like an RDBMS’ tables, objects can be designed and implemented to reflect reality more closely.

For example, in an RDBMS, data about a car could be stored in multiple tables. One table would track the parts of the car and their most recent review while another table would track maintenance and associated costs. A separate table would contain all the manufacturers of parts as aligned with a primary key and those manufacturers names would not be stored with the parts, rather the number would be. This makes updating a single datum easy as it only must be updated in one place.

In an OODBMS, by contrast, a car would be defined as a “class” and everything about a specific car, from parts to ownership history, would be stored as a single “instance” of the object. OODBMS are conceptually akin to Plato’s Theory of Forms in that an OODBMS relies on defined “classes” like Plato’s Forms as opposed to an RDBMS, which relies on enforcement of its “schema.” A class defines how data will be stored for objects of that class. Each object is an “instance” of that object and can use some or all of the features of the class, much like shadows on Plato’s allegorical cave wall.

Object Oriented databases are designed to store “objects” used in Object Oriented Programming environments. The storage of objects is different than the storage of tables in a Relational database. One of the inherent features in an Object-Oriented Database is Polymorphism.

In an Object-Oriented Database, an Object is defined by its “class” and actions on Object Oriented Databases are performed via “methods” rather than queries. Data is grouped by object rather than by table which can have implications for both data collection and production.

Where an RDBMS is designed in advance with a specific schema, an Object-Oriented Database may allow for user generated schema.

Examples include ObjectDB, Objectivity/DB, and Versant.

- OQL not SQL
- Designed for speed with Object Oriented programming languages
- Objects contain unique IDs which can be directly referenced via pointer
- Very useful for BLOBS (Binary Large Objects) like images and videos.

OODB Image: Car “Class” with different Car “Instances”

NOSQL Databases

NOSQL (which stands for Not Only SQL) databases were designed to address the exponential growth of real time data in the modern Internet age. Internet data is growing at more than 3 quintillion bytes of data per day, almost all of it unstructured. Relational databases just can’t keep up.

NOSQL databases are specifically designed to work efficiently with unstructured data, such as that found in social media, email and documents. NOSQL has a simple query language, is designed from the ground up to be distributed across many servers, is highly scalable and very reliable.

Examples of systems that use NOSQL databases include Amazon, Netflix and Facebook.

Key-Value Database

Since database storage is a technique for organizing, storing, and accessing data, different types of data require different types of database structures. NoSQL databases, whether implemented using key-value arrangements, document stores, column-oriented structures, or graph configurations, store their data differently than an RDBMS does.

The simplest of the NoSQL databases is a Key-Value database. A Key-Value database is designed around two pieces of information - a unique key and a value associated with that key. Values do not have type constraints as they do in an RDBMS, and they can generally not be queried using a SQL query.

While a Key-Value database does not have a schema in the same way a Relational database does, it does have a “keyspace” at the logical top level of a Key-Value store to organize the keys in the key-value pairs. Data is retrieved via exact key matching. The two central features of a Key-Value database are that every record has a unique key and that every object in the database is tied to one of those keys. The value in a Key-Value database can be anything so long as the value is tied to a unique key, which is a significant deviation from an RDBMS’ schema. Each value in a key-value pair may have its own

The simplicity of a Key-Value database makes it difficult to search on a per-field basis as storage is explicitly and exclusively tied to the single Key for each record.

Additionally, Key-Value databases can be used in partitioned and mirrored environments over multiple servers, which may create evidentiary issues should the “state” of a database be relevant.

A Key-Value database is easy to update and change due to the simplicity of the structure of the system. A value may only be retrieved by a specific key and depending on the way the Key-Value database is deployed, the value may not be searchable at all.

Examples of a Key-Value database engines include Apache Cassandra, Redis, and Amazon DynamoDB.

Key-Value Image Goes Here of a Primary Key and a Car Image BLOB

Columnar Database

A column store database, or a “columnar” database, stores data within single columns as files and each column is saved on its own. Each row has one or more Columns in it which contain a key-value pair as well as a time stamp. Every row in a specific field is stored in a single column.

Wide Column Store

A wide column store is, despite the name, closer to an RDBMS than a columnar database. One important difference is that column formats and data types can be different in rows within the same column and do not have the rigid enforcement of datatype that Relational databases do.

Document Database

One common variety of Key-Value databases is a Document Store Database. A Document Database is a Key-Value database in which the values are referred to as documents which are comprised of structured or semi-structured data. One significant differentiating feature between a Key-Value database and a Document Database is that the fields within a document database are indexed and thus can be queried, allowing for field-based evaluation and querying which standard Key-Value databases do not. This allows for the retrieval of data subsets within a “document.” With a standard Key-Value database, the result of a search is the entire value which can not be searched until it is exported from the database. One important distinction between Document and Relational databases is that all data for each

“Document” is the entirety of data about that document – there is no mechanism for joining documents together as one can in a relational database.

Document Databases often store their data in JSON.

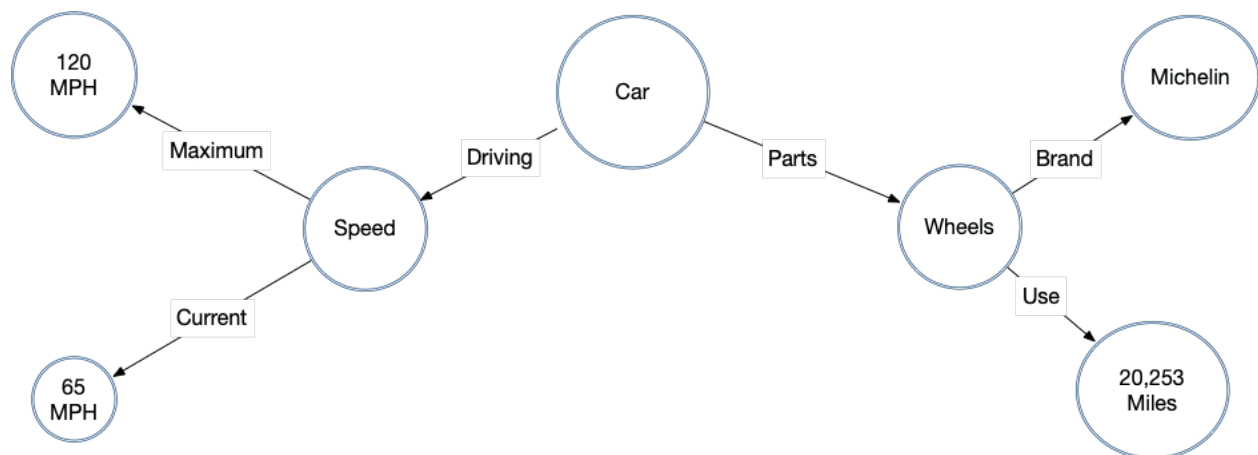
These are often used for Content Management Systems, eCommerce sites, and analytics. Example systems include - MongoDB, CouchDB, and Amazon’s DocumentDB.

Graph Database

A graph database has an entirely different metaphor than an RDBMS or any other database type described supra. A graph database is often used to measure and describe relationships between things. The real world is rarely as well organized as a database table, and a graph database is designed to focus on the interconnectedness of things – whether objects, people, or data.

Within a graph database, all entries are a set of discrete objects called “nodes” which are related to zero or more of the other objects within the database via “edges.” Each node contains some data, called a “property.”

Graph databases are used to explore relationships and are used extensively for Fraud Detection, social network graphs, and to drive a Recommendation engine.



Multi Model Database

https://en.wikipedia.org/wiki/Multi-model_database

Data Exchange Formats

JSON

Javascript Object Notation (“JSON”) is an open-source standard data exchange format used for data transfer and storage. JSON notation is simple and generic and can be used to represent any type of data structure, regardless of the underlying database system.

XML

eXtensible Markup Language is a markup language originally used for websites (it was the data storage equivalent of the website's Hypertext Markup Language or HTML). However, XML has evolved into a generic data representation language, in a similar way to JSON. The major difference between XML and JSON is that XML defines a set of rules for encoding data (the XML Schema), whereas JSON is (mostly) unstructured and can represent different data sets in the same JSON file.

CSV

A Comma Separated Value ("csv") file is a text file used to transmit fielded data. The fields are indicated or "delimited" by an identified character, often a comma, a pipe ("|"), or other unique character. CSV files are commonly used with relational databases to represent the data in a single table, or the results of a SQL query.

Other Formats

While Excel, Numbers, or other Spreadsheet document types may be used to transfer data, metadata unrelated to an export may be created

Database vs. Data Warehouse vs. Data Lake

Database

- often relational, designed to capture and record data (OLTP - Online Transaction Processing)
- Live, real-time data
- Data stored in tables with rows and columns
- Data is highly detailed
- Flexible schema
- ETL - Extract, Transform, Load
- Database is a structure for storage of information that allows for the retrieval of data at a later point in time. Different databases use different structures to solve different problems.

Data Warehouse

- Data warehouse
 - Used for reporting and analysis
 - Collectively gathered information from multiple sources
 - Top-Down approach which creates data marts for specific groups after the warehouse has been created and bottom up approach which creates a data warehouse which contains everything.
 - Data Marts - for pulling data in specific areas

- Multiple databases can be put into a single location
 - Create a single data model across multiple sources
 - Restructure data as needed.
- Will always have historical data but not necessarily current data
 - Summarized data rather than every row and column

Data Lake

- Captures any raw data at all (Structured, Semi-Structured, and Unstructured)
- Large quantities
- Used for ML and AI in its current state or processing the data for Analytics
- Organize the data on receipt in order to send results (after cleaning) into a DB or DW)