

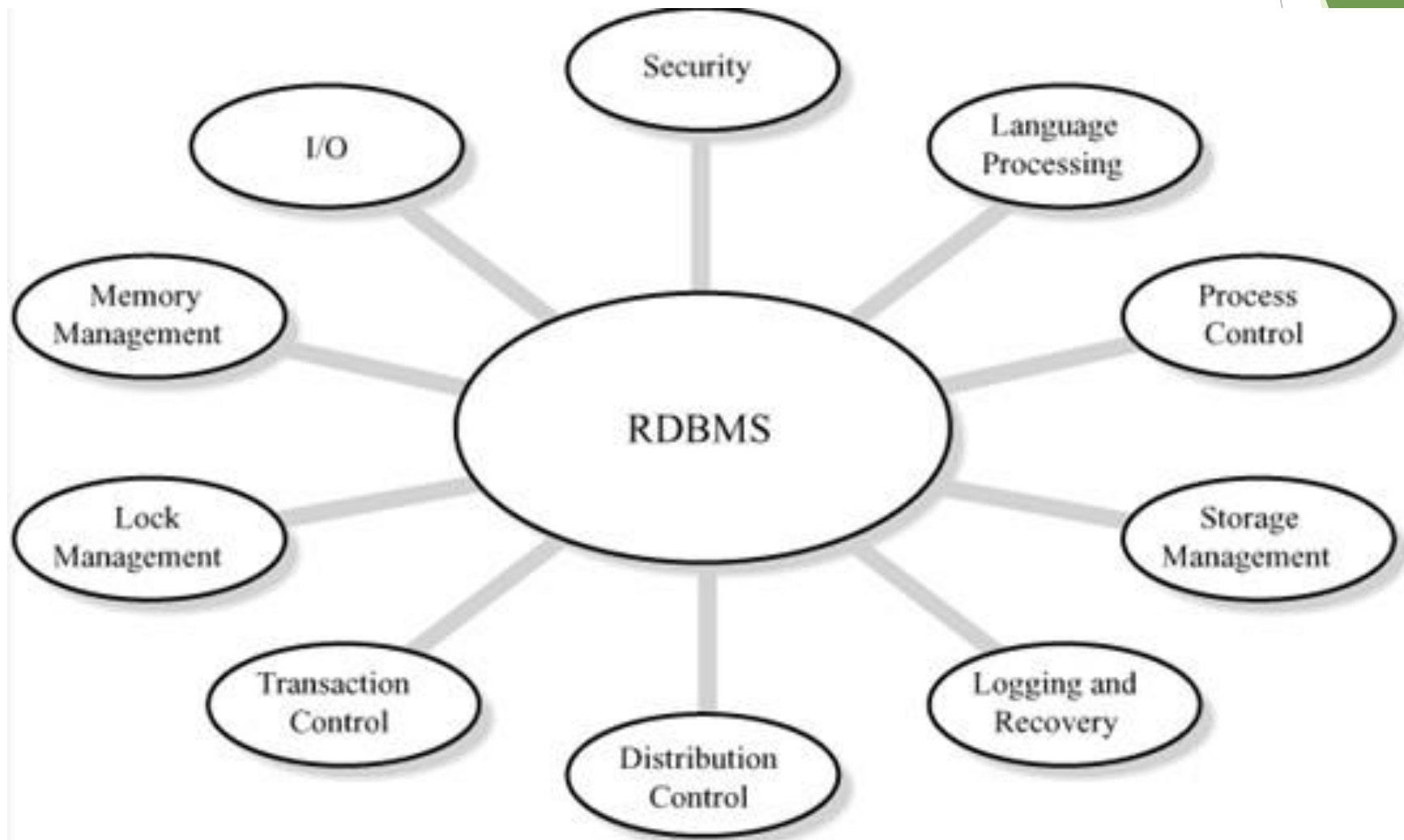
Relational Database Management System (RDBMS)

240110102

Relational Vs. Non Relational DBMS

Relational	Non-Relational
Data stored in structured format	No fix structure
Store and retrieval of data is easy	Not easy to manipulate data
Can use SQL	No fix language, it depends on the data storage format
Data stored in form of rows and column (relation)	Data in form of document , column-oriented, key-value store and graph
Any application can use the database	These types are used for a single application.
Can define key for unique identification of record, integrity constraints can be impose	There are no tables, rows, primary keys or foreign keys
Most popular relational databases have been Microsoft SQL Server, Oracle Database, MySQL and IBM DB2. Several free versions of these RDBMS platforms have gained popularity over the years, such as SQL Server Express, PostgreSQL, SQLite, MySQL and MariaDB.	Popular NoSQL databases are MongoDB, Apache Cassandra, Redis, Couchbase and Apache HBase

Benefits of using RDBMS



Benefits of using RDBMS

- Possible to design complex data storage and retrieval systems with ease (and without programming).
- Support for ACID transactions
 - Atomic
 - Consistent
 - Independent
 - Durable
- Easy for backup and recovery.
- Easy accessibility and query based protocols ensure good interactivity.
- Security and privacy ensured by maintenance of a key (Primary / Foreign key)
- Supporting multiple format file structures with multi level indexing.

Types of Databases

- ▶ Text/ file databases
 - ▶ A large portion of the information available worldwide in electronic form is actually in text form (other popular forms are structured and multimedia information).
- ▶ Relational database
 - ▶ Microsoft SQL Server, Oracle Database, MySQL and IBM DB2.
- ▶ NoSQL database
 - ▶ many different types of NoSQL databases, with different specifications: Key value, graph etc...
 - ▶ MongoDB is a most popular NoSQL Database (document-oriented database) that stores JSON-like documents in dynamic schemas
 - ▶ Cassandra DB is also highly scalable. Facebook® created Cassandra DB
 - ▶ The Firebase Realtime Database is a cloud-hosted NoSQL database
- ▶ Cloud database
 - ▶ 5 Best Cloud Databases to Use in 2021. Amazon DynamoDB, Google BigQuery, Azure SQL Server, Azure Cosmos DB, Amazon Redshift
- ▶ Object-oriented database
 - ▶ 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
 - ▶ Examples: Wakanda, ObjectStore
- ▶ Hierarchical database
 - ▶ 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.
 - ▶ Examples: IBM Information Management System (IMS), Windows Registry

Components of DBMS

1. Database Engine

The core service for accessing and processing data. It handles storage, query processing, and transaction management.

2. Data Definition Language (DDL) Compiler

Interprets and processes DDL statements like CREATE, ALTER, DROP to define schema and structure of the database.

3. Data Manipulation Language (DML) Compiler

Translates DML statements (INSERT, UPDATE, DELETE, SELECT) into low-level instructions the DBMS engine can execute.

4. Query Processor

Analyzes, parses, and optimizes SQL queries before execution. Converts high-level queries into efficient execution plans.

5. Transaction Manager

Ensures ACID properties (Atomicity, Consistency, Isolation, Durability) during transactions. Manages concurrency and crash recovery.

6. Storage Manager

Manages storage space on disk and memory. It handles data files, indexes, and data buffering for optimized access.

7. Buffer Manager

Controls data transfer between disk and main memory (RAM). Keeps frequently used data in memory for faster access.

8. Authorization and Integrity Manager

Enforces user access controls and data validation rules. Ensures only authorized users can access or modify the data.

9. Catalog/Metadata Manager

Maintains metadata — data about database structure like tables, columns, data types, indexes, constraints, etc.

10. User Interface or Query Interface

Allows users to interact with the DBMS. Interfaces may include command-line (e.g., SQL prompt) or graphical tools (e.g., MySQL Workbench).

DBMS Languages

- ▶ In a **Database Management System (DBMS)**, special languages are used to perform tasks such as defining the structure of data, manipulating data, and controlling user access. These are called **DBMS languages** and can be categorized into:
 - ▶ DDL - Data Definition Language
 - ▶ DML - Data Manipulation Language
 - ▶ DCL - Data Control Language

DDL - Data Definition Language

- ▶ DDL is used to define the structure of database objects such as tables, schemas, indexes, and views.
- ▶ DDL statements do not manipulate data.
- ▶ They only define or change the structure of database objects.
- ▶ Common DDL Commands:
 - ▶ Create
 - ▶ Alter
 - ▶ Drop
 - ▶ Truncate

DML - Data Manipulation Language

- ▶ DML is used to manipulate and manage data in the database.
- ▶ These commands allow inserting, updating, deleting, and retrieving data.
- ▶ Common DML Commands:
 - ▶ Select
 - ▶ Insert
 - ▶ Update
 - ▶ Delete

DCL - Data Control Language

- ▶ DCL is used to control access to data and database objects. It manages user privileges and permissions.
- ▶ Common DCL Commands:
 - ▶ Grant
 - ▶ Revoke

Traditional RDBMS

- ▶ Based on **Relational Model** (tables, rows, columns).
- ▶ Use **Structured Query Language (SQL)**.
- ▶ Great for **structured data**
- ▶ Strong **ACID** (**A**tomicity, **C**onsistency, **I**solation, and **D**urability) compliance.
- ▶ Examples: Oracle, MySQL, SQL Server, PostgreSQL

NoSQL

- ▶ **NoSQL** = “Not only SQL” - a broad class of **non-relational** databases designed for:
- ▶ High **scalability**
- ▶ Flexible **schema-less** structure
- ▶ **Unstructured/semi-structured data**

🔍 Key Characteristics:

- ▶ Horizontal scalability (easier distributed architecture)
- ▶ Schema flexibility (can change structure anytime)
- ▶ High performance for specific workloads
- ▶ For better performance and availability use **BASE**: Basically Available, Soft state, Eventually consistent (NOT ACID)

Types of NoSQL Databases:

<u>Type</u>	<u>Description</u>	<u>Example</u>
Document-based	Store data in JSON-like documents	MongoDB, CouchDB
Key-Value Stores	Simple key-value pairs	Redis, DynamoDB
Column-based	Store data in columns for fast retrieval	Cassandra, HBase
Graph-based	Store data as nodes and relationships	Neo4j, ArangoDB

When is NoSQL Used?

- ▶ Real-time analytics
- ▶ IoT applications
- ▶ Social media platforms
- ▶ Content management systems
- ▶ Applications with massive amounts of unstructured data

NewSQL

- ▶ NewSQL refers to modern relational databases
- ▶ Use SQL as query language
- ▶ Provide ACID compliance
- ▶ Offer scalability like NoSQL
- ▶ NewSQL = Best of both worlds: SQL + Scalability

Why NewSQL?

- ▶ To address scalability limitations of traditional RDBMS without giving up SQL features or transaction integrity.
- ▶ Designed for cloud-native applications.

Key Features of NewSQL:

- ▶ **Distributed architecture** for high availability
- ▶ **Full SQL support** (joins, indexes, ACID)
- ▶ **In-memory capabilities** for speed
- ▶ **Auto-sharding** (sharding refers to the practice of horizontally partitioning a large database into smaller, more manageable pieces called shards, which are distributed across multiple servers) and **replication** (process of creating and maintaining multiple copies of data across different nodes or servers)

Popular NewSQL Databases

Database

Google Spanner

CockroachDB

VoltDB

NuoDB

TiDB

Features

Global distribution, strong consistency

Fault-tolerant, scalable SQL

In-memory processing for speed

Cloud-native, elastic scaling

MySQL compatible, supports hybrid workloads

Comparison: RDBMS vs NoSQL vs NewSQL

Feature	RDBMS	NoSQL	NewSQL
Data Model	Relational	Non-relational	Relational
Query Language	SQL	Varies	SQL
Scalability	Vertical	Horizontal	Horizontal
ACID	Yes	No (BASE)	Yes
Schema	Fixed	Flexible	Flexible
Best Use	Structured data	Big data, IoT, social apps	Cloud apps needing SQL + scale

NoSQL = flexible, fast, scalable → good for modern apps with unstructured data.

NewSQL = power of SQL with scalability → best for enterprises needing reliability + scale.