# Overview on SQL Database

**Structured query language (SQL)** is a programming language for storing and processing information in a relational database.

A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values.

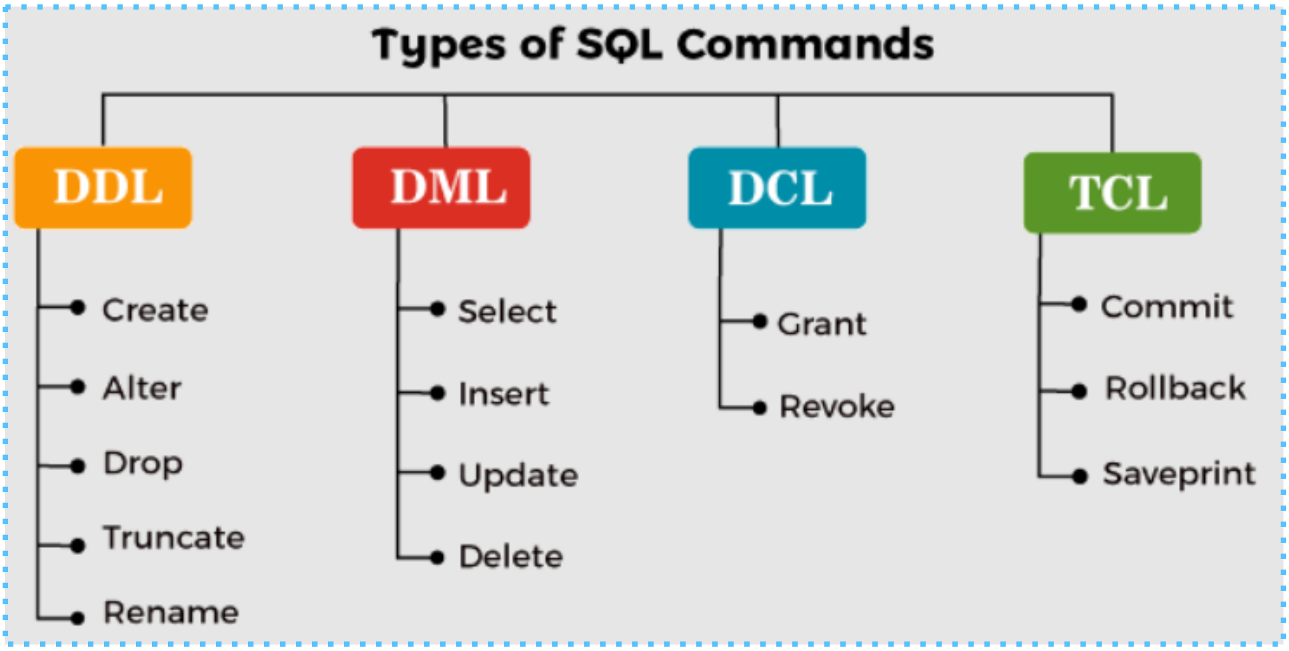
- Structured Query Language (SQL) is the standard language used for writing queries in a databases and frequently used in all types of applications.
- SQL is not case-sensitive. It can define data structure, modify data in a database, specify security constraints and can perform many more tasks.
- It was approved by **ISO** (International Standard Organization) and **ANSI** (American National Standards Institute).
- Most relational databases, including MySQL, ORACLE, SQL Server, etc., support (SQL) query language.

## History of SQL

- SQL was invented in the 1970s based on the relational data model.
- It was initially known as the structured English query language (SEQUEL). The term was later shortened to SQL.
- **Oracle**, formerly known as Relational Software, became the first vendor to offer a commercial SQL relational database management system.

# SQL commands

- Structured query language (SQL) commands are specific keywords or SQL statements that developers use to manipulate the data stored in a relational database.
- You can categorize SQL commands as follows.



## Data definition language

- Data definition language (DDL) refers to SQL commands that design the database structure.
- Database engineers use DDL to create and modify database objects based on the business requirements.
- For example, the database engineer uses the CREATE command to create database objects such as tables, views, and indexes.

## Data query language

- Data query language (DQL) consists of instructions for retrieving data stored in relational databases.
- Software applications use the SELECT command to filter and return specific results from a SQL table.

## Data manipulation language

- Data manipulation language (DML) statements write new information or modify existing records in a relational database.
- For example, an application uses the INSERT command to store a new record in the database.

## Data control language

- Database administrators use data control language (DCL) to manage or authorize database access for other users.
- For example, they can use the GRANT command to permit certain applications to manipulate one or more tables.

## Transaction control language

- The relational engine uses transaction control language (TCL) to automatically make database changes.
- For example, the database uses the ROLLBACK command to undo an erroneous transaction.

# Features of SQL

Features of Structured Query Language (SQL) are

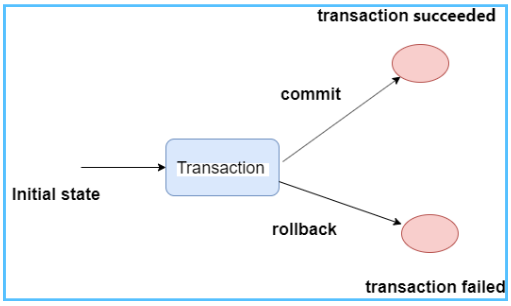
- **Flexibility and Scalability**
    - SQL offers users flexibility and scalability for relational database management systems. With SQL, it is easier to create new tables while dropping or deleting previously-created or scantily used tables.
- **A Comprehensive Application Development Tool**
    - Programmers use SQL to program applications to access a database, making it a comprehensive and effective application development tool.
    - SQL is suitable for every large or small organization, no matter the size. It has rich transactional support.
    - It is capable of handling large records while also managing several other transactions is top-notch.
- **High Performance**
    - It offers a high-performance programming capability for high usage, incredibly transactional, and heavy workload database systems.
- **High Availability**
    - SQL is compatible with other databases such as Microsoft SQL Server, Oracle Database, MS Access, MySQL, SAP Adaptive Server, and more.
- **High Security**
    - SQL also has high security as one of its notable features.

    - It’s easy to give permissions on views, procedures, and tables.

- **SQL’s Management Ease**
    - Common and standard SQL commands include “Delete,” “Insert,” “Select,” “Update,” and “Drop.” These commands help users manage large amounts of data from a database efficiently and quickly.
- **Open Source**
    - Structured Query Language has the feature of being an open-source programming language great for building relational database management systems (RDBMS).
    - This makes it a great pick for developers and programmers who are looking for a community of professionals to learn from off, and it’s also one of the benefits of SQL.
- **Data Integrity**
    - SQL includes features to enforce data integrity, such as the ability to specify constraints on the values that can be inserted or updated in a table, and to enforce referential integrity between tables.
- **User Access Control**
    - SQL provides mechanisms to control user access to a database, including the ability to grant and revoke privileges to perform certain operations on the database.
- **Portability**
    - SQL is a standardized language, meaning that SQL code written for one database management system can be used on another system with minimal modification.

# ACID (Atomicity, Consistency, Isolation, and Durabilability)

- A transaction is a logical unit that read and write operations that only succeeds if all the operations within succeed.



- Transactions can impact a single record or multiple records.
- ACID is an acronym that stands for atomicity, consistency, isolation, and durability.

### Atomicity

- Atomicity guarantees that all of the commands that make up a transaction are treated as a single unit and either succeed or fail together.
- Atomicity is also known as the 'All or nothing rule'.
  - It involves the following two operations.

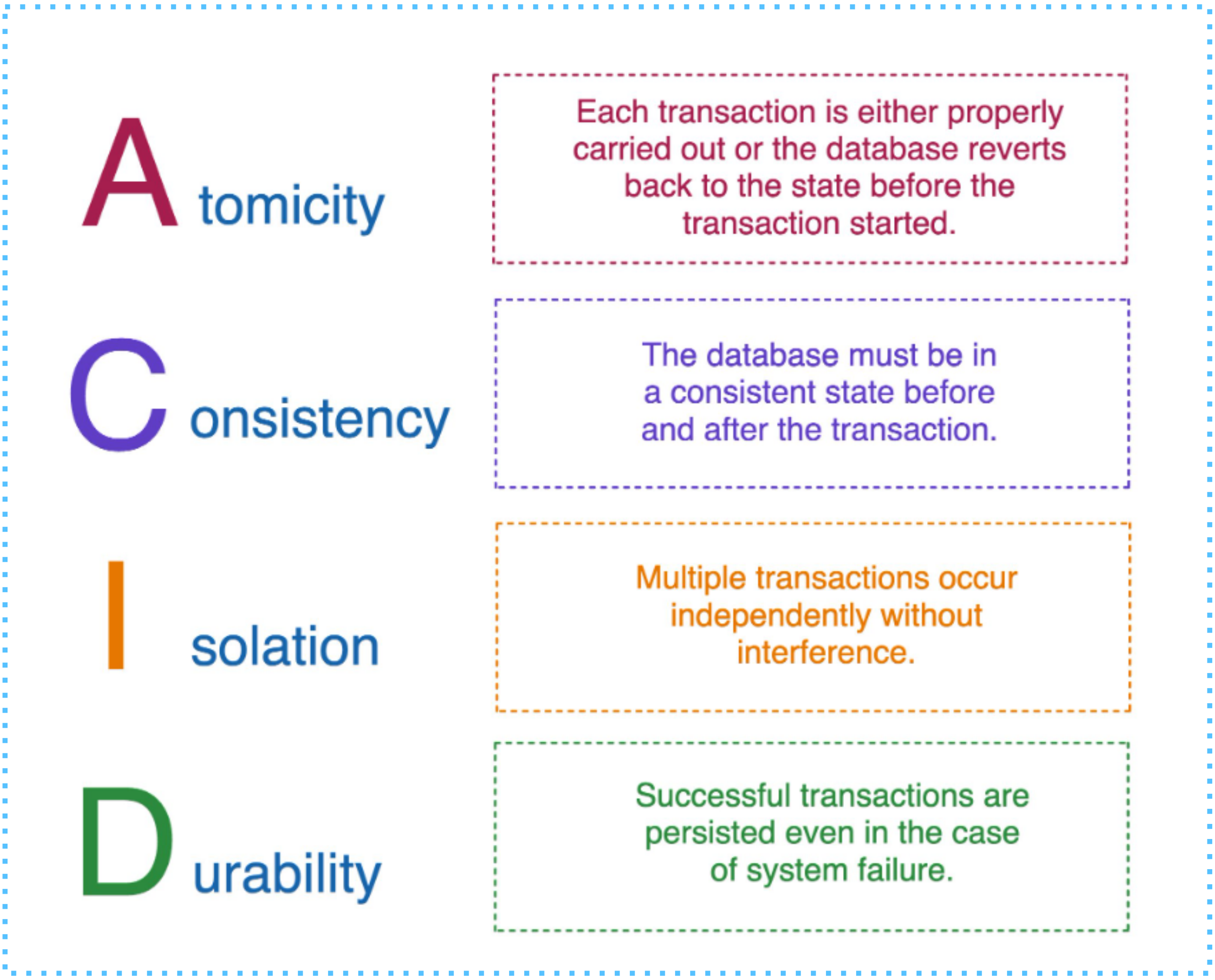
    —Abort: If a transaction aborts, changes made to the database are not visible.

    —Commit: If a transaction commits, changes made are visible.

### Consistency

- Consistency guarantees that changes made within a transaction are consistent, it means integrity constraints must be maintained so that the database is consistent before and after the transaction.

- This includes all rules, constraints, and triggers. If the data gets into an illegal state, the whole transaction fails.


A tomicity
Each transaction is either properly carried out or the database reverts back to the state before the transaction started.
C onsistency
The database must be in a consistent state before and after the transaction.
I solation
Multiple transactions occur independently without interference.
D urability
Successful transactions are persisted even in the case of system failure.

### Isolation

- This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state.
- Transactions occur independently without interference.

### Durability

- Durability guarantees that once the transaction completes and changes are written to the database, they are persisted.
- This ensures that data within the system will persist even in the case of system failures like crashes or power outages.

# Advantages of SQL

### Quicker Query Processing

- SQL Server has the advantage of offering faster and quicker query processing.
- You can efficiently and quickly retrieve a large amount of data.

### Standardized Language

- SQL offers a uniform platform globally to its numerous users because of documentation and long establishment through the years, making it a standardized language for analysts and developers.

### No Coding Skills Required

- When carrying out data retrieval, numerous lines of code aren't needed.
- The syntactical rules aren't so complex in SQL either, making it a user-friendly language.

### Interactive

- The SQL server is interactive, and the language is easy to understand and learn.
- Users can receive the answers to very complex queries in seconds.

### Portable

- Portability is one of the benefits of SQL Server. You can use SQL in programs in laptops, servers, or personal computers independent of any existing platform or operating system.

### Multiple or Different Data Views

- The SQL language helps users to make different views of a database structure while also making databases for different users.

# Disadvantages of SQL

### Difficult Interface

- SQL's interface is a little complex and difficult. This alone makes it difficult for new users to deal with the database at first.

### Partial Control

- The presence of hidden business rules means that the database isn't given complete control.

### Cost

- The fact that some versions of SQL are costly makes accessibility difficult for some programmers.

# Benefits of SQL

Below are benefits of SQL and reasons to prefer it:

- Several users of a database access db via SQL. SQL is easy to learn it consists of mainly English statements, making it easier to understand, learn, and write SQL queries.
- People use SQL to create and manage databases and their security.
- SQL is a complete language for a database, and its main use is in retrieving, updating, and sharing data with more than one user.
- SQL in a 3-tiered internet architecture, including a database, a client, and an application server.
- SQL is widely considered a very desirable skill with many jobs such as web development, IT support, and business data analysts needed the skills in SQL.

- All the major database management system vendors, including Microsoft, IBM, and Oracle, use SQL.

# Row-Based Databases

- Row-Based Databases is a data storage and management approach where data is organized and stored in rows.
- In this structure, each row represents an individual record or entity, and each column contains specific information or attributes associated with the record.
- In a row-based database, data is stored in a horizontal format.
- This means that all the values for a particular record are stored together, making it easier to access and modify individual records.
- It is a common structure used in traditional relational databases.
- They are optimized for reading and writing rows efficiently.
- Common row oriented databases:
  - Postgres
  - MySQL

## Why Row-Based Databases is important

Row-based databases offer several advantages for businesses:

- **Flexibility**
  - Row-based databases provide flexibility in terms of data retrieval and manipulation. The structure allows for easy insertion, deletion, and updating of individual records.
- **Transaction Support**
  - Row-based databases are well-suited for transactional operations that involve modifying individual records. They ensure the integrity of

data by providing transaction support, allowing multiple users to access and modify records simultaneously.

- **Schema Enforcement**
  - Row-based databases enforce schemas, ensuring that the data conforms to a predefined structure. This helps maintain data consistency and integrity.
- **Compatibility**
  - Row-based databases are widely supported and compatible with popular database management systems, making it easier to integrate with existing systems and tools.
- **Pros**
  - Modifying data is easier
  - Ideal for OLTP
- **Cons**
  - Slower data aggregation
  - Insufficient compression
  - Requires additional space