Welcome

Hello! I Am

SAJIB KHAN

Executive, Data Analytics & Strategy
at pathao HERE WITH YOU

Lead Data Instructor,
at Interactive CARES

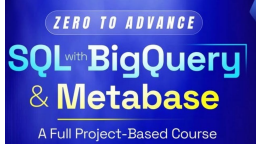
sajibkhansk.netlify.app
/in/sajibkhansk
/sajibkhanofficial
+88 01717 015761
sajibkhansk19@gmail.com

Class 1: Introduction to SQL and Database Fundamentals

INTRODUCTION TO DATABASE & DBMS

- **Database:** A structured collection of data that can be easily accessed, managed, and updated.

- **Database Management System (DBMS):** Software that interacts with the user, applications, and the database itself to capture and analyze data.



Relational Database Management System (RDBMS)

What is an RDBMS?

- A software system for creating, managing, and querying relational databases.
- Organizes data into tables with rows and columns.
- Uses Structured Query Language (SQL) for data manipulation.

Key Features

- **Data Integrity**: Enforces constraints (e.g., primary keys, foreign keys).
- **Scalability**: Handles large datasets and multiple users.
- **Query Optimization**: Efficiently retrieves data using indexes.
- **ACID Compliance**: Ensures reliable transactions (Atomicity, Consistency, Isolation, Durability).

Popular RDBMS Examples

- MySQL
- PostgreSQL
- Oracle Database
- Microsoft SQL Server

Relational Database Management System (RDBMS)

Benefits

- Structured data storage.
- Easy data retrieval and updates.
- Supports complex queries and joins.
- Ensures data consistency and security.

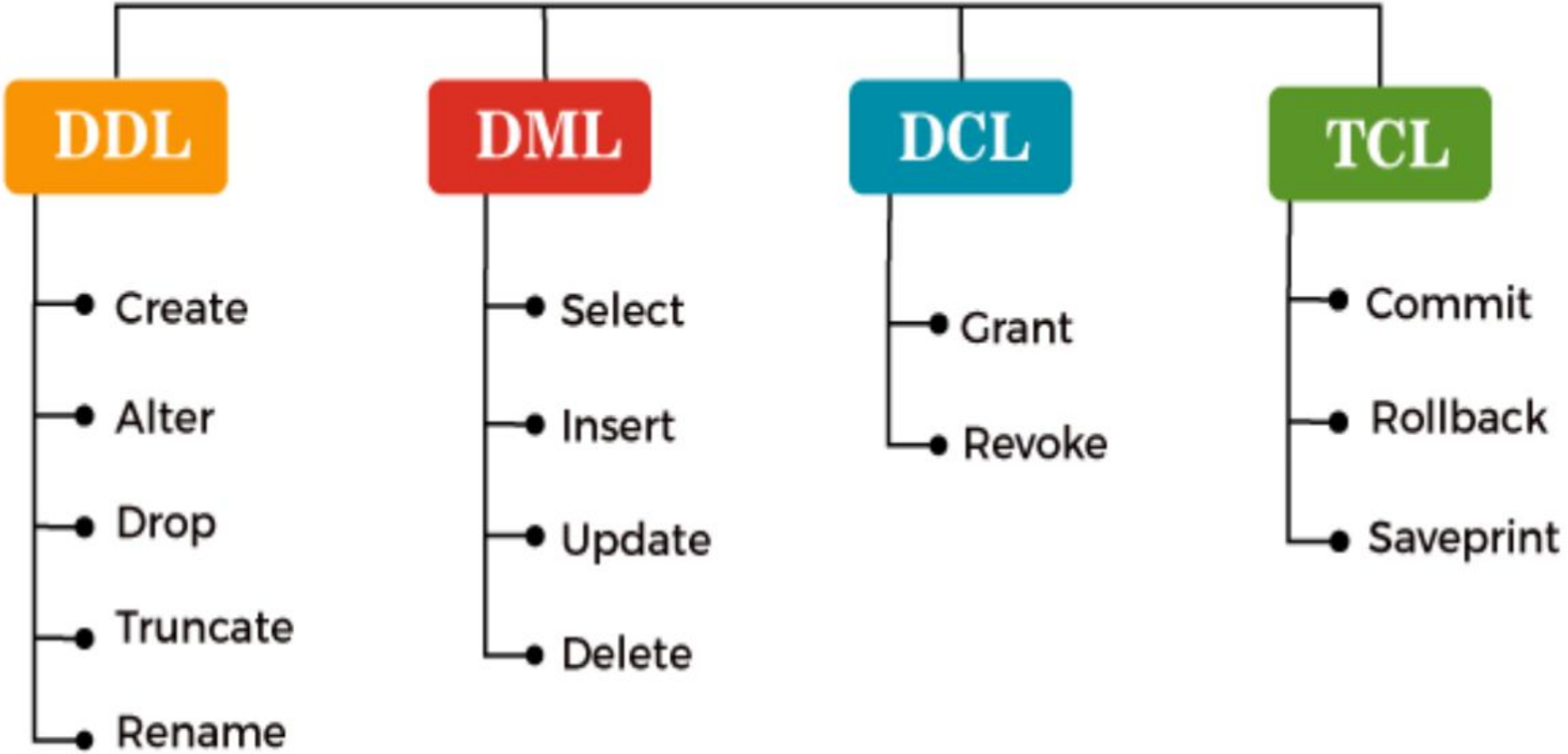
Use Cases

- Financial systems (e.g., banking).
- E-commerce platforms.
- Customer relationship management (CRM).
- Data analytics and reporting.

INTRODUCTION TO SQL

SQL (Structured Query Language): A standardized programming language used for managing and manipulating relational databases.

Types of SQL Commands



Understanding ACID Properties

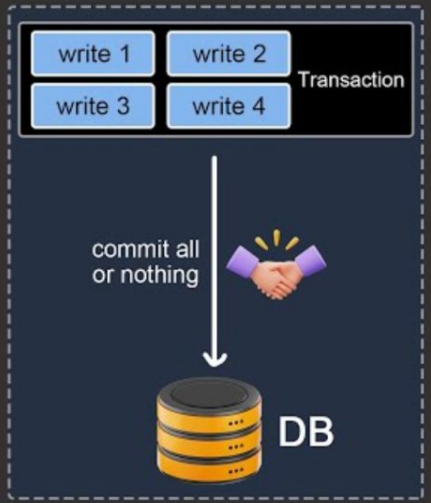
These properties guarantee reliable and secure database transactions, ensuring data integrity and consistency even in the face of errors or failures.

- **ACID**: Ensures reliable database transactions
- **Atomicity**: Ensures transactions are all-or-nothing (fully completed or not executed)
- **Consistency**: Maintains database integrity, ensuring valid states before and after transactions
- **Isolation**: Transactions are independent; partial changes are not visible until complete
- **Durability**: Committed transactions are permanently saved, even in case of system failure

What does ACID Really Mean?

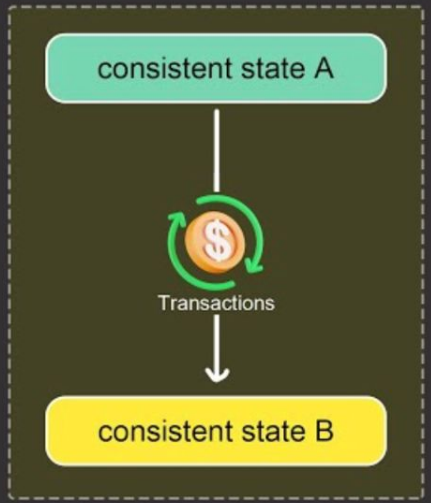
Atomicity

All or nothing



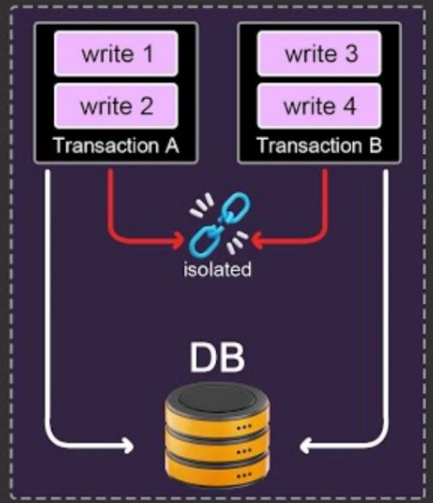
Consistency

Preserving database invariants



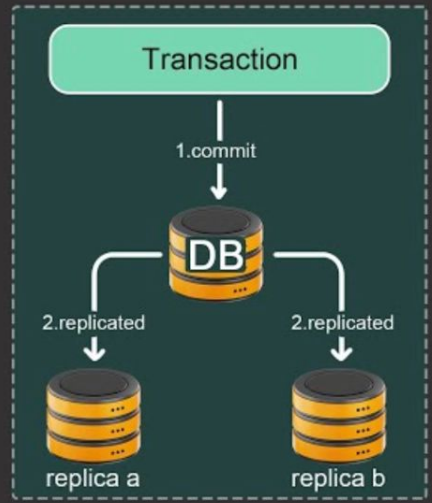
Isolation

Concurrent transactions are isolated from each other



Durability

Data is persisted after transaction is committed even in a system failure



NORMALIZATION

The process of organizing data in a database to minimize redundancy and dependency by dividing large tables into smaller, related tables.

Benefits:

- Eliminates Redundancy: Reduces repeated data, saving storage and improving efficiency.
- Improves Data Integrity: Ensures consistency and reduces the chances of data anomalies.
- Optimizes Query Performance: Makes data retrieval more efficient by structuring data logically.

A NORMALIZED DATABASE

Employee

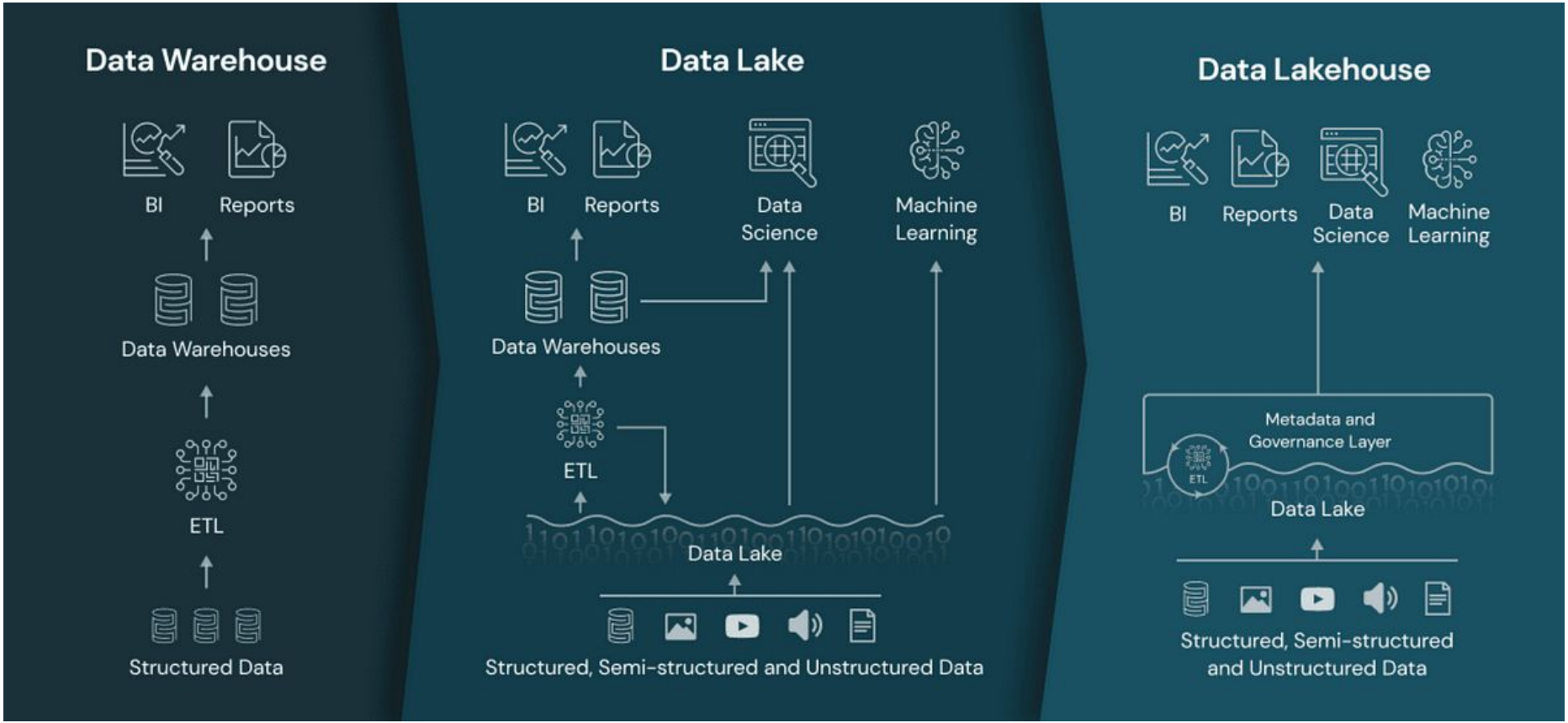
employeeID	employeeName	managerID	sectorID
1	David D.	1	4
2	Eugene E.	1	3
3	George G.	2	2
4	Henry H.	2	1
5	Ingrid I.	2	4
6	James J.	3	1
7	Katy K.	3	4

Sector

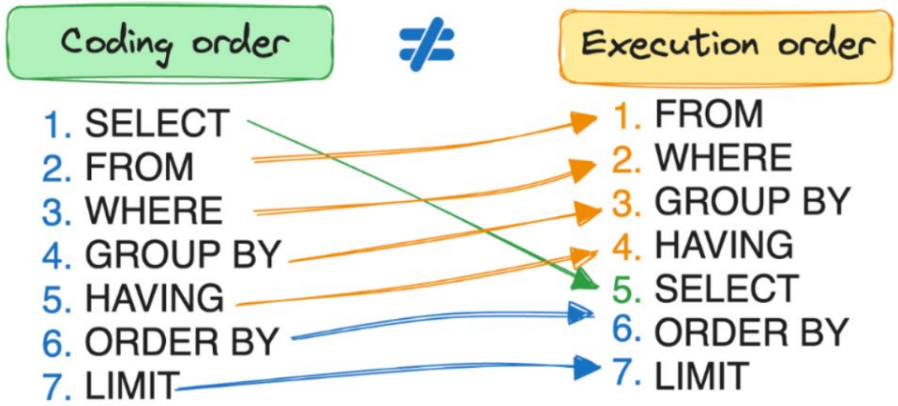
sectorID	sectorName
1	Administration
2	Security
3	IT
4	Finance

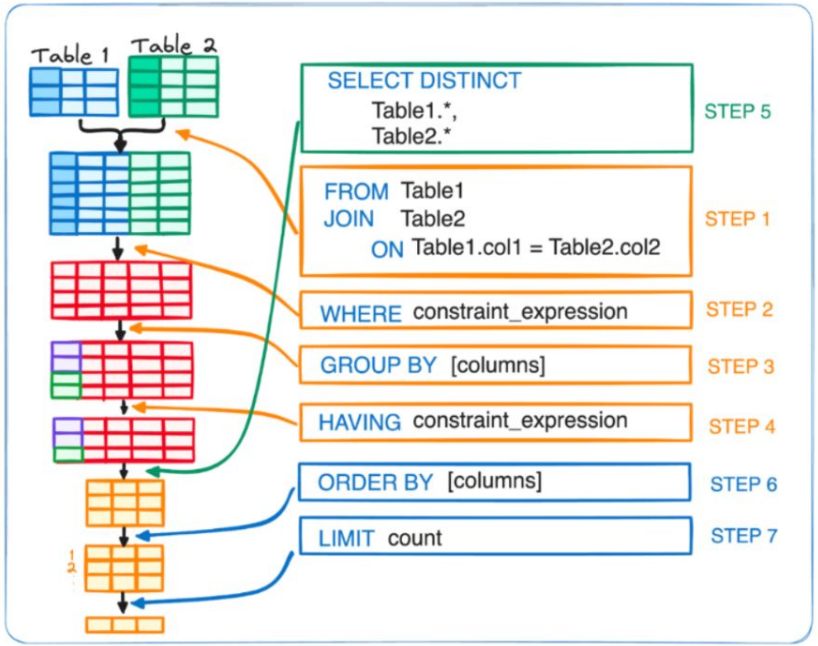
Manager

managerID	managerName	area
1	Adam A.	East
2	Betty B.	West
3	Carl C.	North

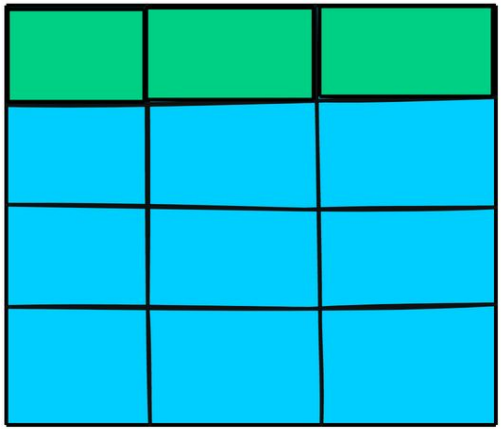

Data Warehouse
BI
Reports
Data Warehouses
ETL
Structured Data
Data Lake
BI
Reports
Data Science
Machine Learning
Data Warehouses
ETL
Data Lake
Structured, Semi-structured and Unstructured Data
Data Lakehouse
BI
Reports
Data Science
Machine Learning
Metadata and Governance Layer
ETL
Data Lake
Structured, Semi-structured and Unstructured Data

EXECUTION ORDER OF SQL QUERY



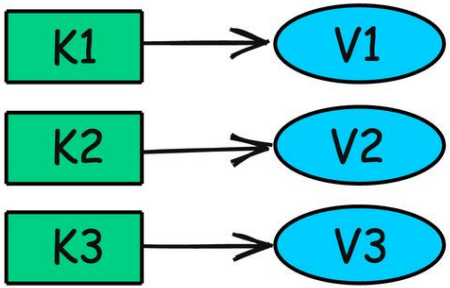


SQL

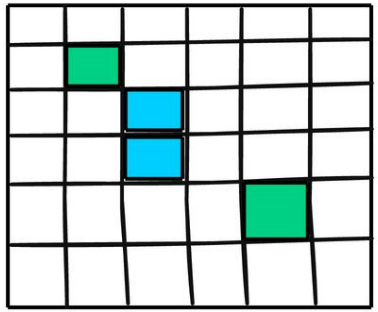


Relational

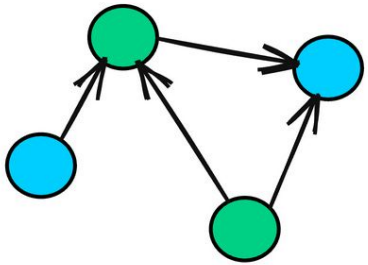
NoSQL



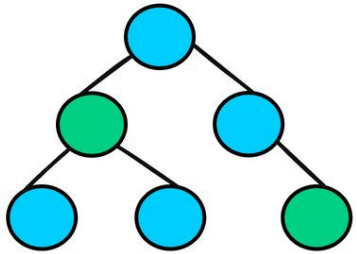
Key-Value



Column Store



Graph



Document

Main Differences Between SQL and NoSQL Databases

Feature	SQL Databases	NoSQL Databases
Key Focus	Reducing data duplication	Scaling and rapid application change
Data Storage Model	Tables with fixed rows and colums	Document: JSON documents; Key-value: key-value pairs; Wide-column: tables with rows and dynamic columns; Graph: nodes and edges
Schemas	Rigid	Flexible
Data to Object Mapping	Requires ORM (object-relational mapping)	Typically doesn't require ORMs. E.g. MongoDB documents map directly to data structures in popular programming languages.
Scaling	Vertical (scale-up with a larger server)	Horizontal (scale-out across commodity servers)

WHY SHOULD YOU LEARN SQL IN 2025 ?

- ✅ 1. Essential for Data-Driven Decision Making
- ✅ 2. Universal in Databases (MySQL, PostgreSQL, SQL Server, BigQuery, Snowflake, etc.)
- ✅ 3. Must-Have for BI & Data Analytics (Power BI, Tableau, Looker, etc.)
- ✅ 4. High Demand in the Job Market
- ✅ 5. Crucial for AI & Machine Learning (Data Preprocessing & Feature Engineering)
- ✅ 6. Compatible with Cloud & Big Data (Google Cloud, AWS, Azure, Databricks, etc.)
- ✅ 7. Powers Automation in Data Pipelines (ETL & Workflow Automation)
- ✅ 8. Easy to Learn & User-Friendly (Simple Syntax & Readability)

CHALLENGES YOU WILL FACE WITHOUT SQL KNOWLEDGE

- ❌ **Reduced Job Opportunities** (SQL is a key requirement for data-related roles)
- ❌ **Limited Data Access & Analysis** (Struggle to retrieve and analyze structured data efficiently)
- ❌ **Dependency on Others** (Need developers or IT teams for data extraction and reports)
- ❌ **Slower Decision-Making** (Unable to quickly query databases for insights)
- ❌ **Difficulty in Using BI Tools** (Power BI, Tableau, and other tools rely on SQL for advanced analysis)
- ❌ **Challenges in AI & Machine Learning** (Harder to preprocess and structure data for models)
- ❌ **Inefficiency in Data Engineering & ETL** (Unable to automate data workflows and transformations)
- ❌ **Struggle with Big Data & Cloud Platforms** (BigQuery, Snowflake, AWS Redshift require SQL for queries)
- ❌ **Inability to Optimize Performance** (Poor understanding of indexing, joins, and query optimization)
- ❌ **Falling Behind in a Data-Driven World** (SQL remains the foundation of data handling)


ZERO TO ADVANCE
SQL with BigQuery
& Metabase
A Full Project-Based Course
THANK YOU