ScholarHat

Swipe

Database Management Systems (DBMS)

ROADMAP

2024 EDITION

LEVEL UP YOUR CAREER

AMIT KUMAR GHOSH

MENTOR AND VICE PRESIDENT AT SCHOLARHAT

DBMS Career Scope

Why learn DBMS?

DBMS skills enable effective organization, storage, retrieval, and management of data, vital for optimizing business operations and information handling.

Scalability and Performance Optimization:

Data Retrieval and Querying

Concurrency Control and Transaction Management

Data Organization and Management

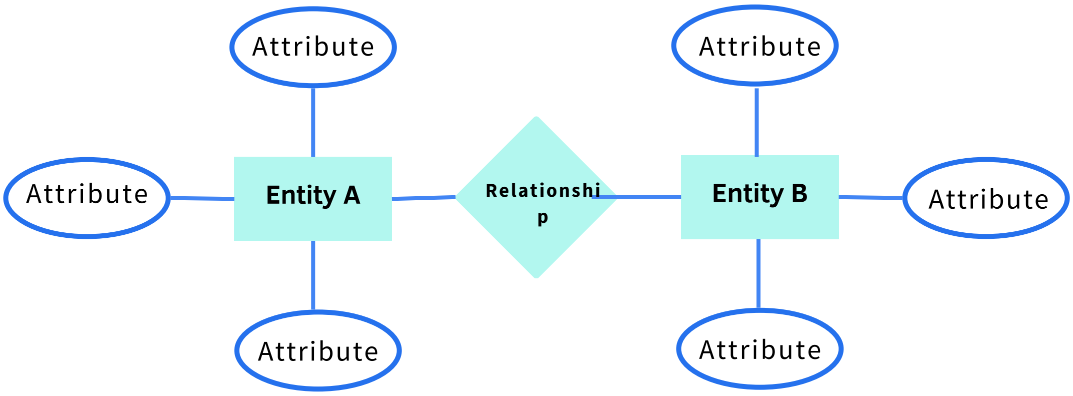
Data Security and Access Control

Integration with Applications

World Top Companies Using DBMS

1 Introduction to DBMS

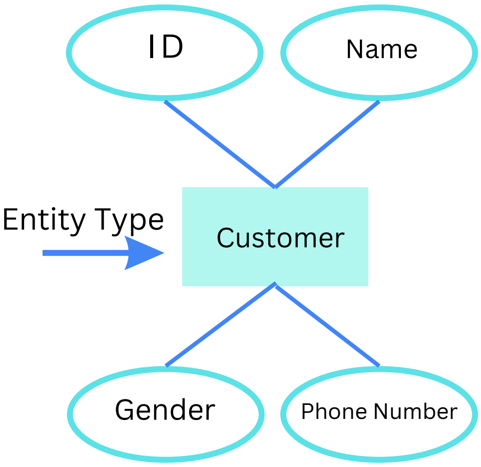
1. **File System vs. DBMS:**
    - Differences between File System and DBMS
    - Know the **challenges** with File Systems.
    - **Advantages** of DBMS such as data security, retrieval, data sharing etc.
2. **ER Model (Entity-Relationship Model):**
    - Identify the main **entities** in your system objects about which data needs to be stored.
    - For each entity, define its **attributes** (characteristics of the entity).
    - Define **relationships** between entities (Unary, Binary, n-ary).
    - Specify the **Cardinality** of each relationship (one-to-one, one-to-many, many-to-many).



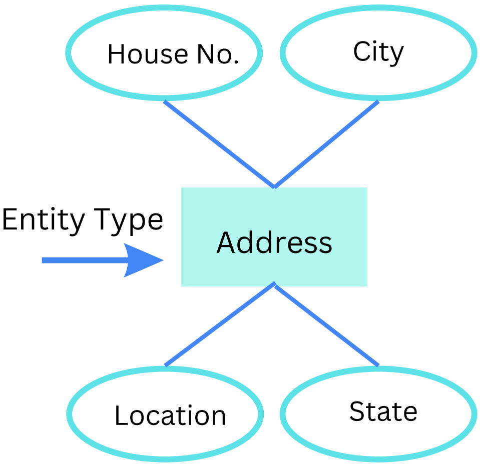
2 Entity Types

Entities are real-world objects that we store information about. These entities are then represented in a database schema using entity-relationship diagrams (ERDs).

- know about the types of entities, which are tangible entities and intangible entities.
- **Within the world of entities, there are two main types:**
  - **Strong Entities:** has its own existence and is independent.
  - **Weak Entities:** does not have its own existence and relies on a strong entity for its existence.



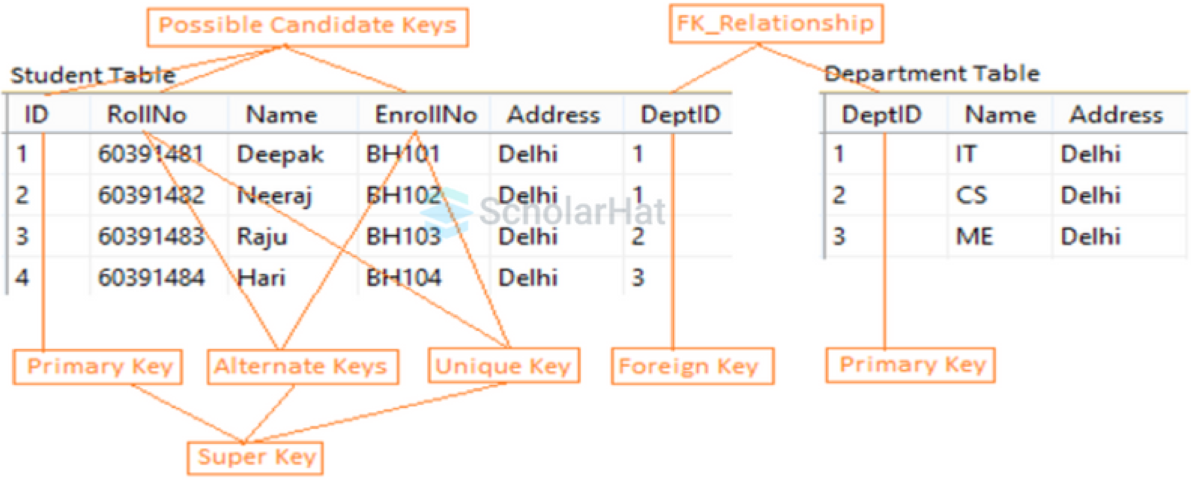
Strong Entity Type



Weak Entity Type

3 DBMS Keys

- **Super Key**: A set of one or more columns that can uniquely identify a record in a table.
- **Candidate Key**: A minimal set of columns that can uniquely identify each row in a table.
- **Primary Key**: A specific candidate key selected to uniquely identify database records, cannot be null.
- **Foreign Key**: A set of columns in one table that references the primary key of another table to make a relationship.
- **Unique Key**: A set of columns, where all values must be unique across the dataset, can contain nulls.
- **Alternate Key**: A candidate key that is not chosen as primary key but is capable of uniquely identifying each row.
- **Composite/Compound Key**: consisting of two or more columns used together as a unique identifier for a record.



Student Table

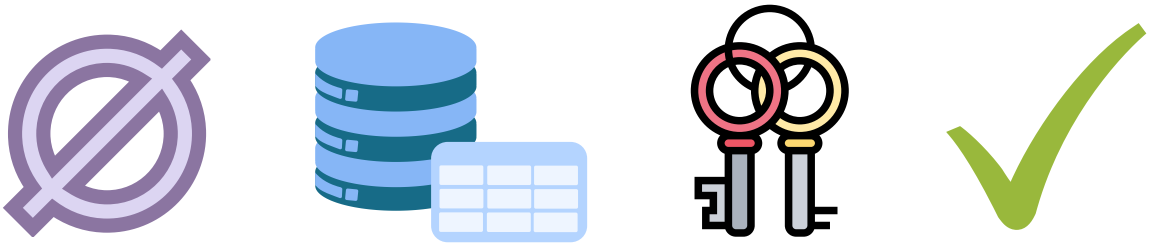
ID	RollNo	Name	EnrollNo	Address	DeptID
1	60391481	Deepak	BH101	Delhi	1
2	60391482	Neeraj	BH102	Delhi	1
3	60391483	Raju	BH103	Delhi	2
4	60391484	Hari	BH104	Delhi	3

Department Table

DeptID	Name	Address
1	IT	Delhi
2	CS	Delhi
3	ME	Delhi

4 DBMS Constraints

- Learn what constraints are and why they are important.
- Understand their role in maintaining data integrity.
- Explore some real-life examples how constraints can be implemented.
- SQL Constraints are as**:**
  - **NOT NULL**: Ensures that a column **cannot have a NULL value**.
  - **UNIQUE**: Guarantees that **all values in a column are different**.
  - **PRIMARY KEY**: A **combination of NOT NULL and UNIQUE**, uniquely identifies each row in a table.
  - **FOREIGN KEY**: Ensures referential integrity by **linking a column or a group of columns to a PRIMARY KEY** in another table.
  - **CHECK**: **Enforces a specific rule** on each row in a table, ensuring that certain conditions are true or false.
  - **DEFAULT**: **Set a default value** for a column when no value is given to that column.



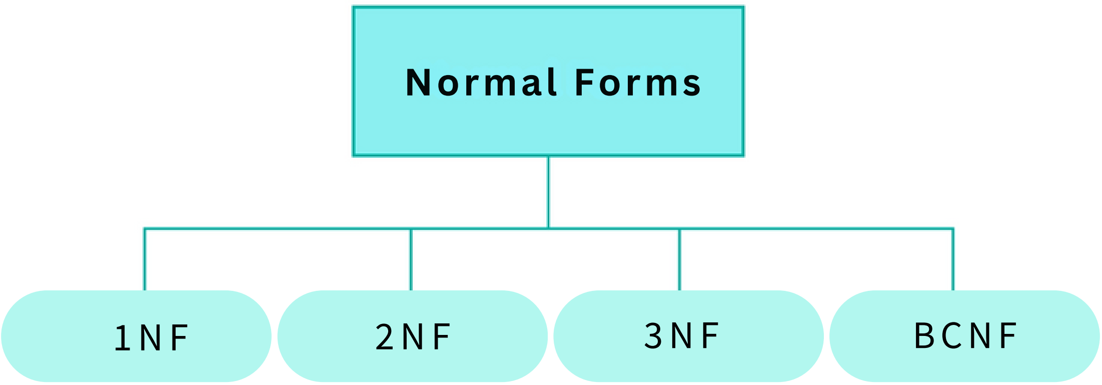
Swipe

5 Normalization

It refers to the process of organizing data in a database efficiently, reducing redundancy and dependency, and ensuring data integrity.

Normalization Forms: A Step-by-Step Approach

- **First Normal Form (1NF):** Ensures each cell contains a single atomic value (indivisible unit of data).
- **Second Normal Form (2NF):** Builds upon 1NF and eliminates partial dependencies.
- **Third Normal Form (3NF):** Builds upon 2NF and eliminates transitive dependencies.
- **Boyce-Codd Normal Form (BCNF):** A stricter form of 3NF.



Normal Forms

1NF

2NF

3NF

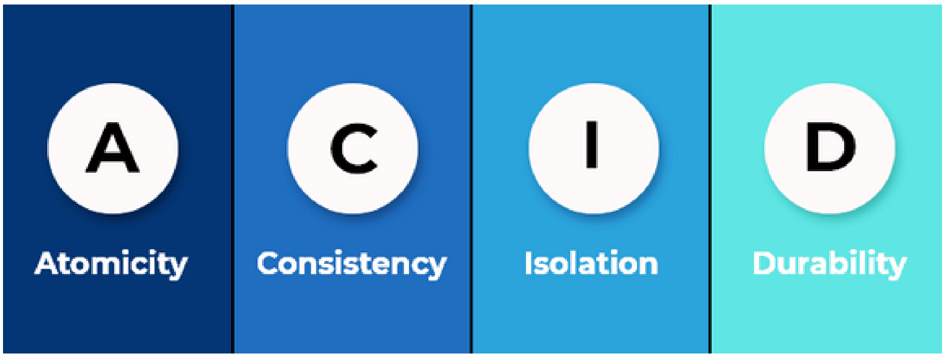
BCNF

6 ACID Properties

ACID Properties are fundamental principles that ensure reliable and consistent database transactions.

ACID is an acronym that encapsulates four properties:

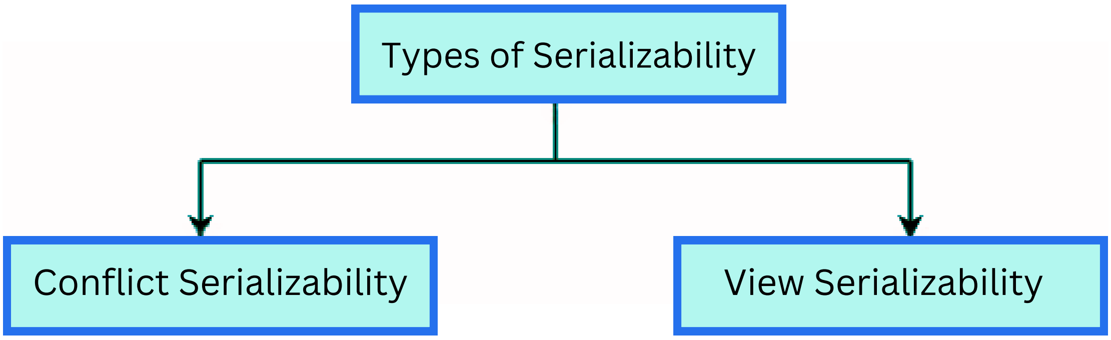
- **Atomicity:** ensures that a transaction is treated as a single, indivisible unit of work.
- **Consistency:** ensures that database remains in a valid state before and after the execution of a transaction.
- **Isolation:** ensures that intermediate states of transactions are invisible to other transactions until they are completed.
- **Durability:** Durability guarantees that once a transaction is committed, its effects persist even in the event of system failures.



7 Serializability

Serializability guarantees that the execution of multiple transactions in parallel does not produce any unexpected or incorrect results.

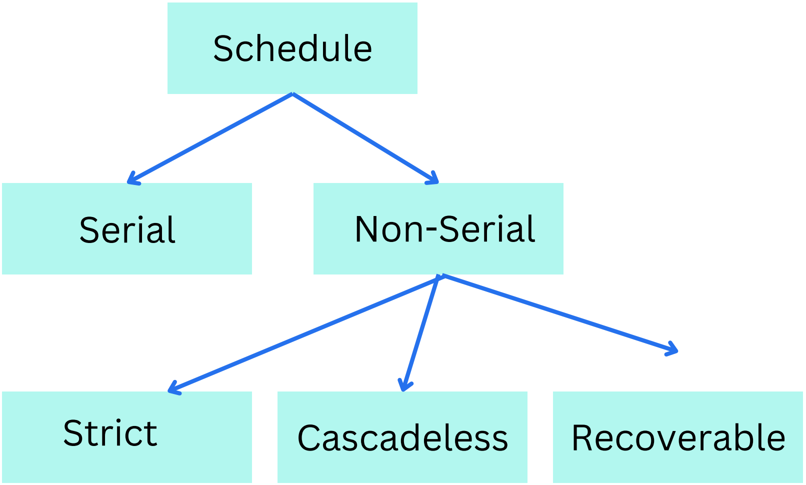
- Know about **Concurrency Control** to ensure serializability.
- DBMS provides **isolation levels** (e.g., Read Uncommitted, Read Committed, Repeatable Read, Serializable) to control the visibility of changes made by concurrent transactions.
- **Here are the main types of serializability:**
  - **Conflict Serializability:** based on identifying and resolving conflicts between transactions.
  - **View Serializability:** a procedure to check if the given schedule is consistent or not.



8 Schedules

A **Schedule** represents the sequence in which the operations of multiple transactions are executed.

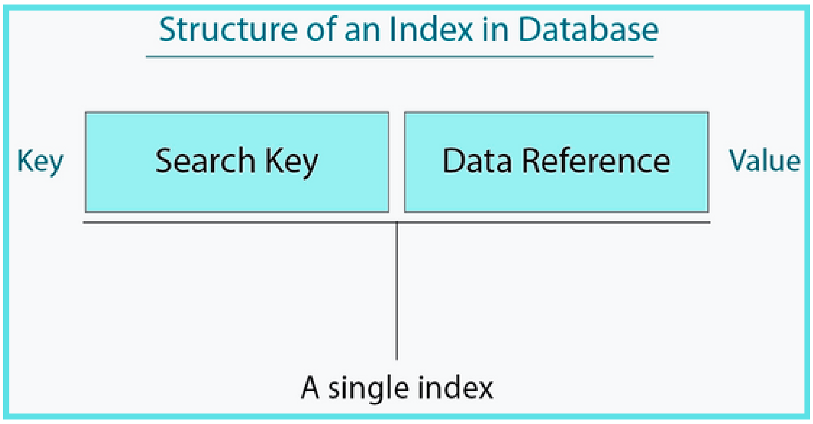
- Differentiate between:
  - **Serial Schedule:** transactions are executed sequentially, one after the other.
  - **Non-serial Schedule:** involve concurrent execution of transactions, where operations from different transactions can be interleaved.
- **Let's break down these schedule types in a DBMS:**
  - **Recoverable Schedules**
  - **Cascadeless Schedule**
  - **Strict Schedule**



9 Indexing

Indexing is a technique in DBMS that significantly improves the performance of data retrieval operations.

1. **Types of Indexing**
    - Clustered Indexing
    - Non-Clustered Indexing
    - Primary Index
    - Secondary Index
2. **Index Files and Implementation**
    - Structure of Index File
    - Implementation of Index
3. **Multi-Level Indexing**
    - B-Tree Indexing
    - B+Tree Indexing



10 Transaction and Concurrency Control

Concurrency control mechanisms manage simultaneous access to shared resources (e.g., database tables, records) by multiple transactions to prevent data corruption and ensure consistency.

- Learn about **Locking Protocols:**
  i. **Basic Locking**
  ii. **Conservative Locking**
  iii. **Strict Locking**
  iv. **Rigorous Locking**
- Understand **Graph-based Protocols:** represent transactions as nodes and edges in a directed acyclic graph (DAG).
- Know about **Time-Stamp Ordering:** A concurrency control protocol that utilizes timestamps assigned to transactions to determine their execution order.

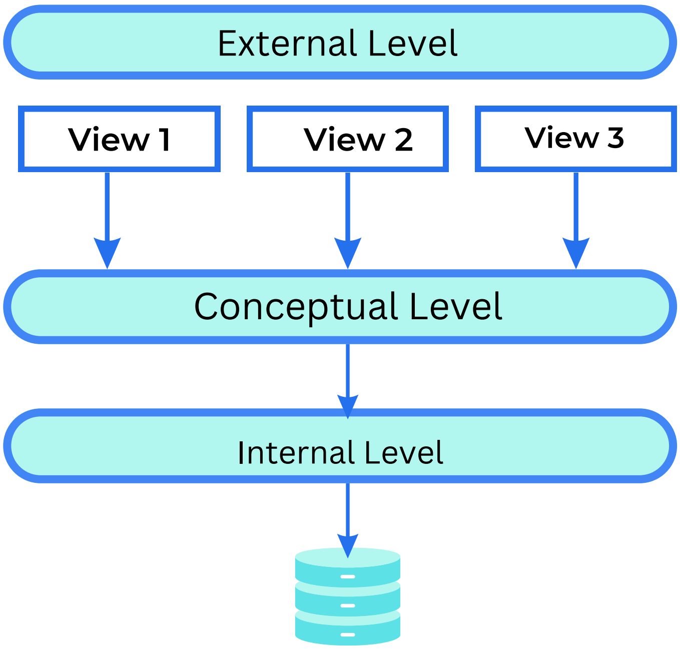
Swipe

11 Levels of Database

Understanding the different **levels of data abstraction** plays a crucial role in effectively interacting with and managing data.

Each level caters to specific needs:

- **Physical level (Internal):** Efficient storage and retrieval (handled by DBMS)
- **Logical level (Conceptual):** User-friendly data organization and schema management
- **View level (External):** Tailored data access control and simplified user interfaces



Swipe

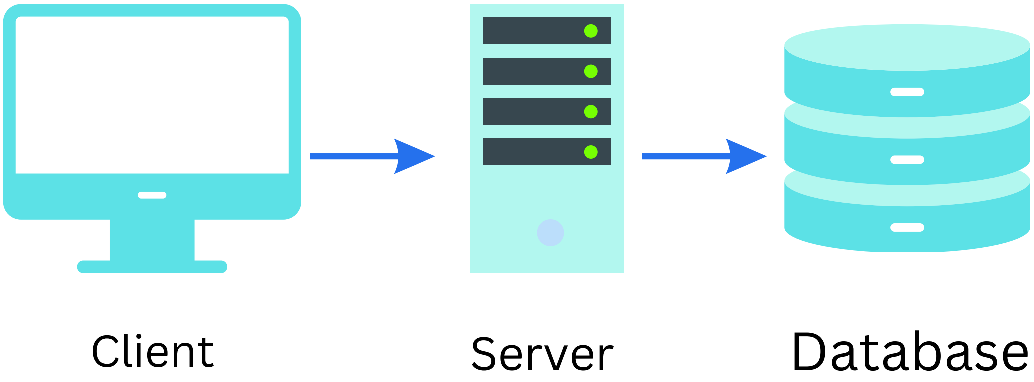
12 Tiers of Architecture

The architecture typically follows a **three-tiered approach** to organize and manage various components and functionalities.

Here are the common three tiers of architecture in a DBMS:

1. **Presentation Tier (User Interface):** responsible for presenting information to the user.
2. **Business Logic Tier (Application Layer):** handles the core application logic.
3. **Data Access Tier (Data Layer):** manages the interaction with the database itself.

Three Tier Architecture

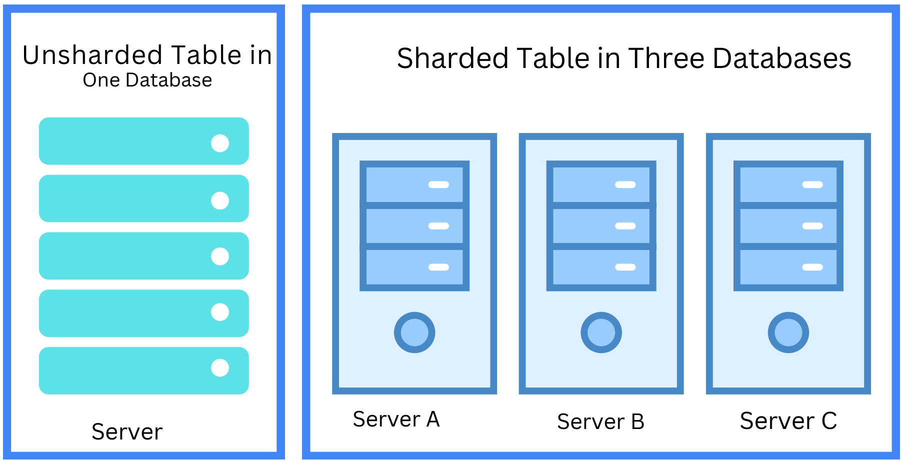


13 Sharding in DBMS

Sharding (horizontal partitioning) is a technique to scale horizontally by distributing data across multiple database servers (shards).

Here's a roadmap to guide you in learning and implementing sharding in DBMS:

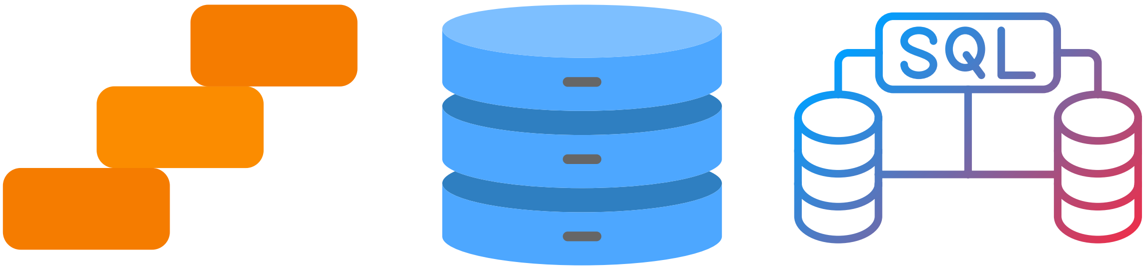
- Understand the concept of sharding.
- Explore the **benefits** and **limitations** of sharding
- Learn about various **sharding key options (Hashing and Range Partitioning)** used to determine which shard stores a specific data record.
- Gain knowledge on essential sharding **management tasks (Shard key management, Query Routing, Data rebalancing).**



14 Procedural vs. Non-Procedural Language

Understanding the **differences between procedural and non-procedural** languages is essential for choosing the most appropriate tool for interacting with data and achieving desired outcomes.

- **Procedural languages** focus on specifying the exact steps (procedures) the DBMS needs to follow to perform a specific task.
- Know about the Procedural i.e, SQL and PL/SQL.
- **Non-Procedural languages** focus on describing what needs to be done (the desired outcome) rather than the specific steps to achieve it.
- Know about the Non-Procedural i.e, Query by Example (QBE), Search languages.

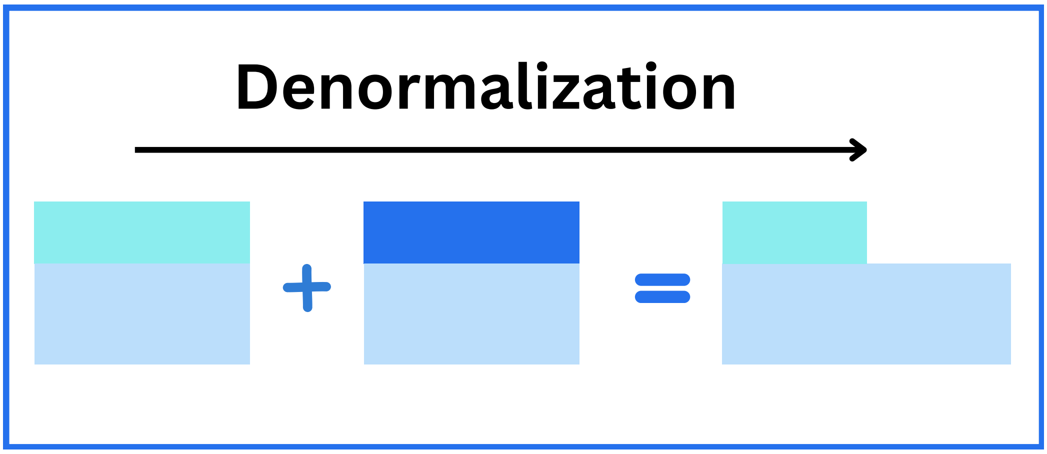


15 Denormalization

Denormalization involves intentionally violating the principles of normalization to improve the performance of data retrieval operations.

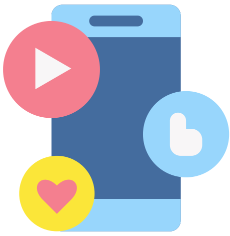
Here's a roadmap to master denormalization:

- Understand the **core concepts** of normalization and denormalization.
- Recognize the **trade-offs** (Benefits and Drawbacks).
- Implement **Denormalization Strategies** (Adding redundant columns, Pre-calculated aggregates).
- Plan for **managing denormalized data** (Establish data consistency procedures, Monitor storage usage).



16 Build Real-World Projects

- **Library Management System:** Design a system to manage books, borrowers, and transactions in a library.
- **Online Shopping System:** Develop an e-commerce platform with features like product management, order processing and customer accounts.
- **Student Information System:** Create a system to manage students records, grades and attendance for educational institutions.
- **Hospital Management System:** Build a system for managing patient records, appointments and medical histories in a hospital.
- **Social Media Platform:** Develop a simplified social media platform with user profiles, posts, comments and friend requests.



30 Common DBMS Interview Questions

1. What is a Database Management System (DBMS)?
2. Explain the difference between a DBMS and a file system.
3. Define the terms "Entity," "Attribute," and "Relationship" in the context of a database.
4. What are the different types of relationships in a database?
5. Explain the concepts of "Simple," "Composite," and "Multivalued" attributes.
6. What is a primary key? How is it different from a foreign key?
7. Describe the normalization process in databases.
8. What is the difference between 1st Normal Form (1NF), 2nd Normal Form (2NF), and 3rd Normal Form (3NF)?
9. What is the purpose of indexing in a database?
10. Differentiate between clustered and non-clustered indexing.
11. Explain the ACID properties in the context of database transactions.
12. What is serializability in database transactions?
13. What is a foreign key constraint? Why is it important?

Swipe

30 Common DBMS Interview Questions

1. Describe the types of serializability: View Serializability and Conflict Serializability.
2. Explain the concepts of "Total" and "Partial" participation in relationships.
3. What is a super key and how is it different from a candidate key?
4. What is the difference between a weak entity and a strong entity in a database?
5. How does a B-tree differ from a B+ tree in the context of indexing?
6. What is the 2-Phase Locking Protocol in transaction management?
7. Explain the concept of referential integrity in databases.
8. What is denormalization, and when is it appropriate to use in a database?
9. What is a transaction in a database?
10. Describe the different levels of database architecture.
11. What is a database view?

Swipe

30 Common DBMS Interview Questions

1. Explain the concept of sharding in the context of databases.
2. Differentiate between a procedural and a non-procedural language in the context of databases.
3. What are ACID properties, and why are they essential in database transactions?
4. Explain the concept of a composite key.
5. How do you ensure referential integrity in a relational database?
6. What are the advantages and disadvantages of denormalization?

How to follow this roadmap?

At ScholarHat, we believe **mastering a technology** is a **three-step process** as mentioned below:

- **Step1 - Learn Skills:** You can learn DBMS skills by using **various blogs on DBMS** or **through Videos** on YouTube or **Videos based courses**. For topic revision and recalling make **short notes**.
- **Step2 - Build Experience:** You can build hands-on experience by creating **data storage workflow** like Social Media Platform, Library Management System, etc. Further **build end-to-end real world applications** like Online Forum, car Rental System.
- **Step3 - Empower Yourself:** Build your **strong profile** by mentioning all the above skills with **hands-on experience** on projects. Prepare yourself with interview **Q&A about DBMS** to crack your next job interview.

Congrats!

You are just one interview away!

WAS THIS HELPFUL?

Share with your friend who needs it!

Love. Like. Comment. Share.

Learn. Build. Empower.