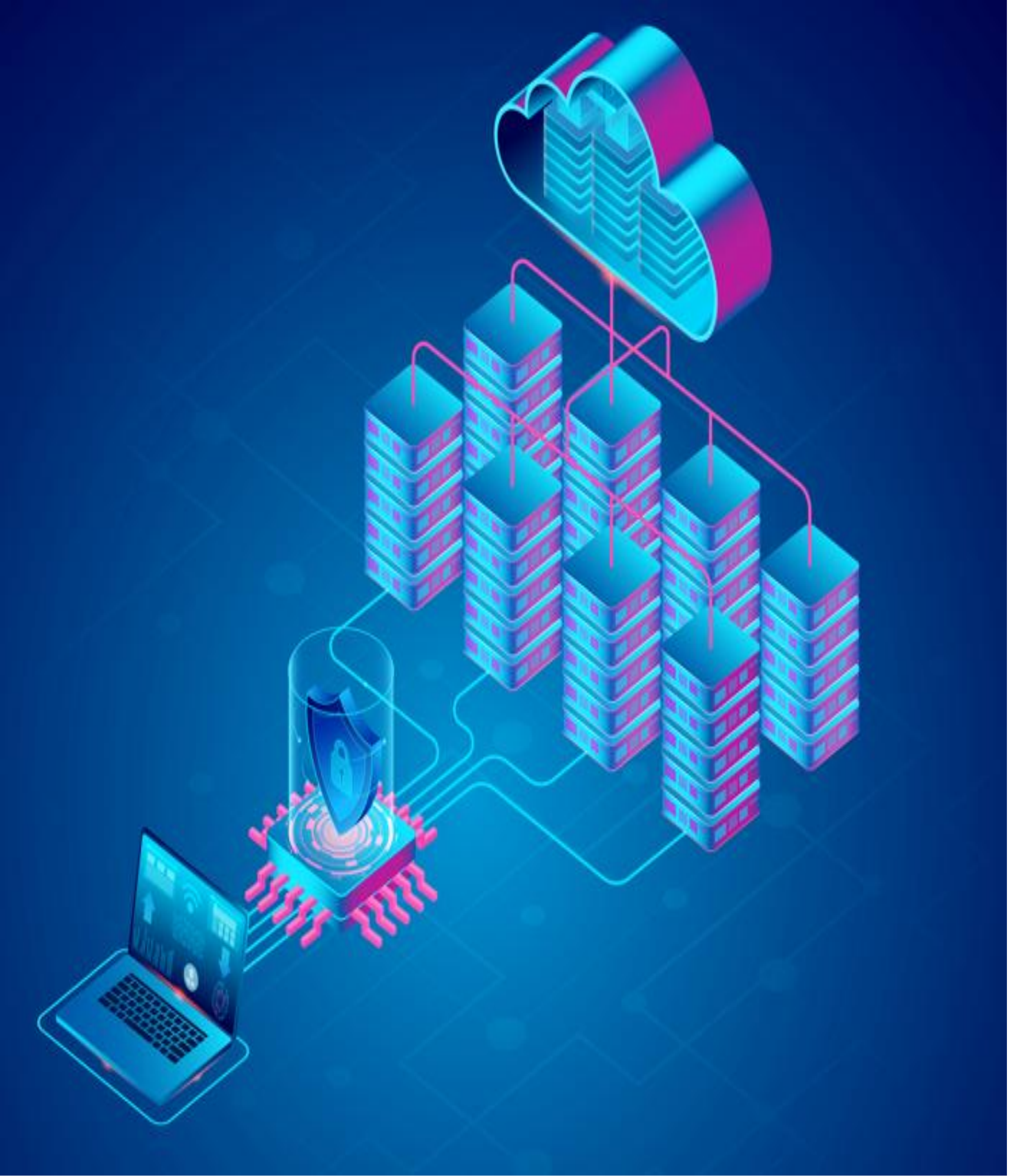


Database Fundamentals

Lecture 1

Eng. Doaa Soleiman

Outline

1 – Database Development Phases

2- Database processing history

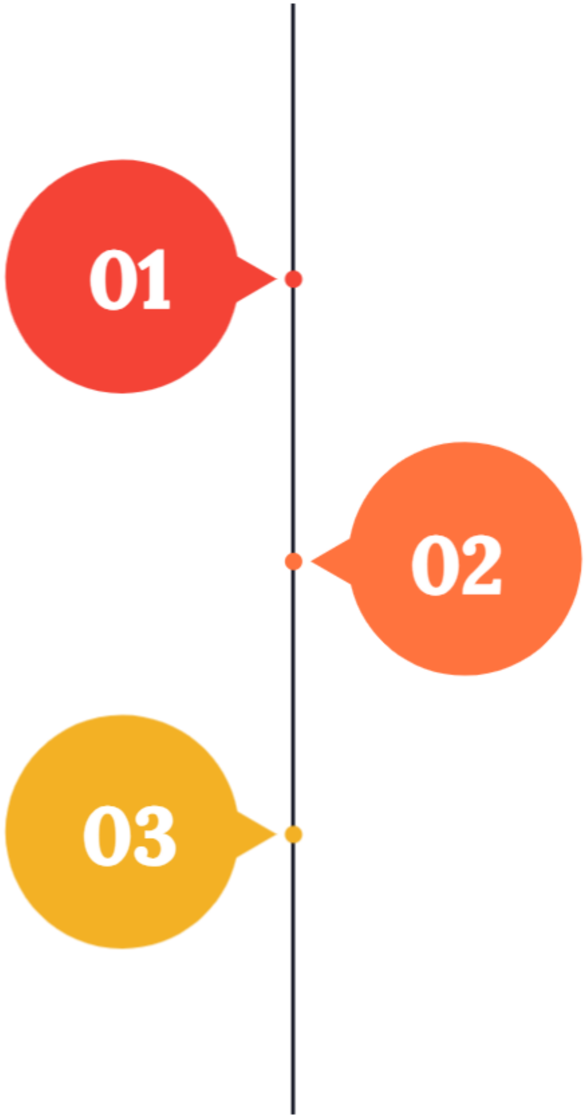
2 - Definitions

- Database
- Database Management System (DBMS)
- Database System
- Database Users
- Database Environment
- DBMS Architecture
- Data Model

Examples of Database Systems in our daily life

Database Development Phases in SDLC

Aligning database creation with user needs and practical implementation



Analysis

Identify business needs, define system features, and develop a functional specification to ensure usefulness and feasibility.

Design

Convert requirements into detailed specifications including data models forms, reports, and processing rules that define data structure and access.

Implementation

Develop and test programs, create data files, train users, and complete documentation to deploy the system reliably.

1- Analysis

- Purpose:
  - Analyze business situation
  - Determining requirements
  - Structuring requirements
  - Selecting competing system features
- Deliverables:
  - Functional specification of system
  - Meeting user requirements
  - Meeting feasibility to develop the system
  - Feasibility to implement the system

2- Design

- Purpose:
  - To elicit and structure all information requirements
- Deliverables:
  - Detailed functional specification of:
    - Data
    - Forms
    - Reports
    - Displays
    - Processing Rules

3- Implementation

- Purpose:
  - To write programs
  - Build data files
  - Test & install new system
  - Train users
  - Finalize documentation
- Deliverables:
  - Programs that work accurately to specification
  - Documentation
  - Training materials

Database Processing: A Transformative Journey Through Time

01

1950

- Manual Paper Records
- Manual paper records dominated, causing inefficiency and high manpower costs.

02

1950

- Magnetic Tapes & Punched Cards
- Sequential data processing began using magnetic tapes and punched card input.

03

1960

- File Processing Systems
- Hard disks enabled direct data access but led to data duplication and program dependence.

Database Processing: A Transformative Journey Through Time

04

1960

- Hierarchical & Network Models
- Introduced tree-like and graph structures to better organize data

05

1970

- Relational Model by Edgar F. Codd
- Revolutionized databases by storing data in tables with defined relationships.

06

1990

- Object-Oriented Databases
- Integrated object-oriented programming concepts to support complex data types and behaviors.

07

2000

- NoSQL & Document-Oriented Databases
- Emerged to address scalability and flexibility for big data and web applications.

Definitions

Excel vs Database … Which is better???

Database

- **Database**: Logical collection of Non-Redundant, shareable data that is used by different application systems of some given enterprise, this data represents some aspects of the real world.
- **Properties:**
  - Redundancy can be reduced.
  - Allow data sharing by concurrent users.
  - Security restrictions can be applied.
  - Data integrity can be maintained.

Database Management System (DBMS)

A software package/ system to facilitate the creation and maintenance of a computerized database.

It is an intermediate layer between database and the program that access the data. It enables users to create and maintain database and handle all requests from users to access the database.

DBMS Core Functions and Architecture

Understanding how DBMS operates and organizes data for reliability and flexibility

Defining and Constructing Databases

Establishes database structure and schema to organize data effectively

Data Manipulation

Enables querying, updating, and managing data dynamically

Security and Integrity

Protects data access and ensures accuracy through constraints and controls

Concurrency Management

Coordinates simultaneous data access for multiple users without conflicts

DBMS Core Functions and Architecture

Understanding how DBMS operates and organizes data for reliability and flexibility

05

Recovery from Failures

Restores data and system state after crashes or errors

06

Data Dictionary Maintenance

Maintains metadata describing database structure and rules

07

Performance Optimization

Improves efficiency by optimizing queries and storage access

Understanding Databases and DBMS Benefits

Key Features, Advantages, and Practical Considerations

A **Database** is a structured collection of non-redundant, shareable data modeling real-world aspects

DBMS software manages database creation, maintenance, and secure data manipulation

Controls data redundancy and inconsistency to ensure accuracy

Restricts **unauthorized access** to protect sensitive information

Supports **concurrent data sharing** among multiple users

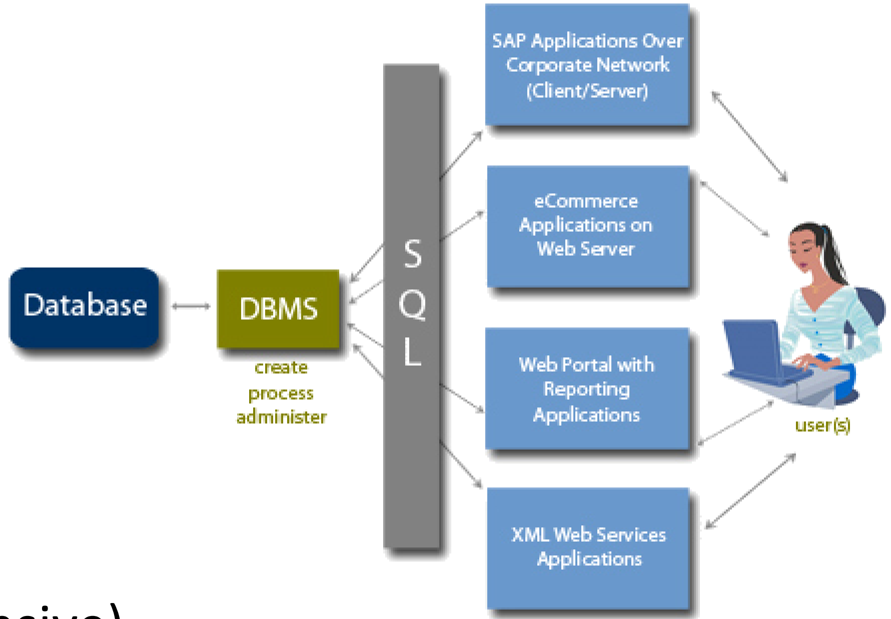
Enforces **data integrity constraints** to maintain reliable data

Provides robust **backup and recovery** mechanisms for data safety

Requires expertise and can be costly, but benefits outweigh challenges

Database Management System

- **DBMS Advantages:**
  - Controlling Redundancy.
  - Restricting Unauthorized Access.
  - Sharing data.
  - Enforcing Integrity Constraints
  - Inconsistency can be avoided.
  - Providing Backup and Recovery.
- **DBMS Advantages:**
  - It needs expertise to use (which is expensive)
  - DBMS itself is expensive

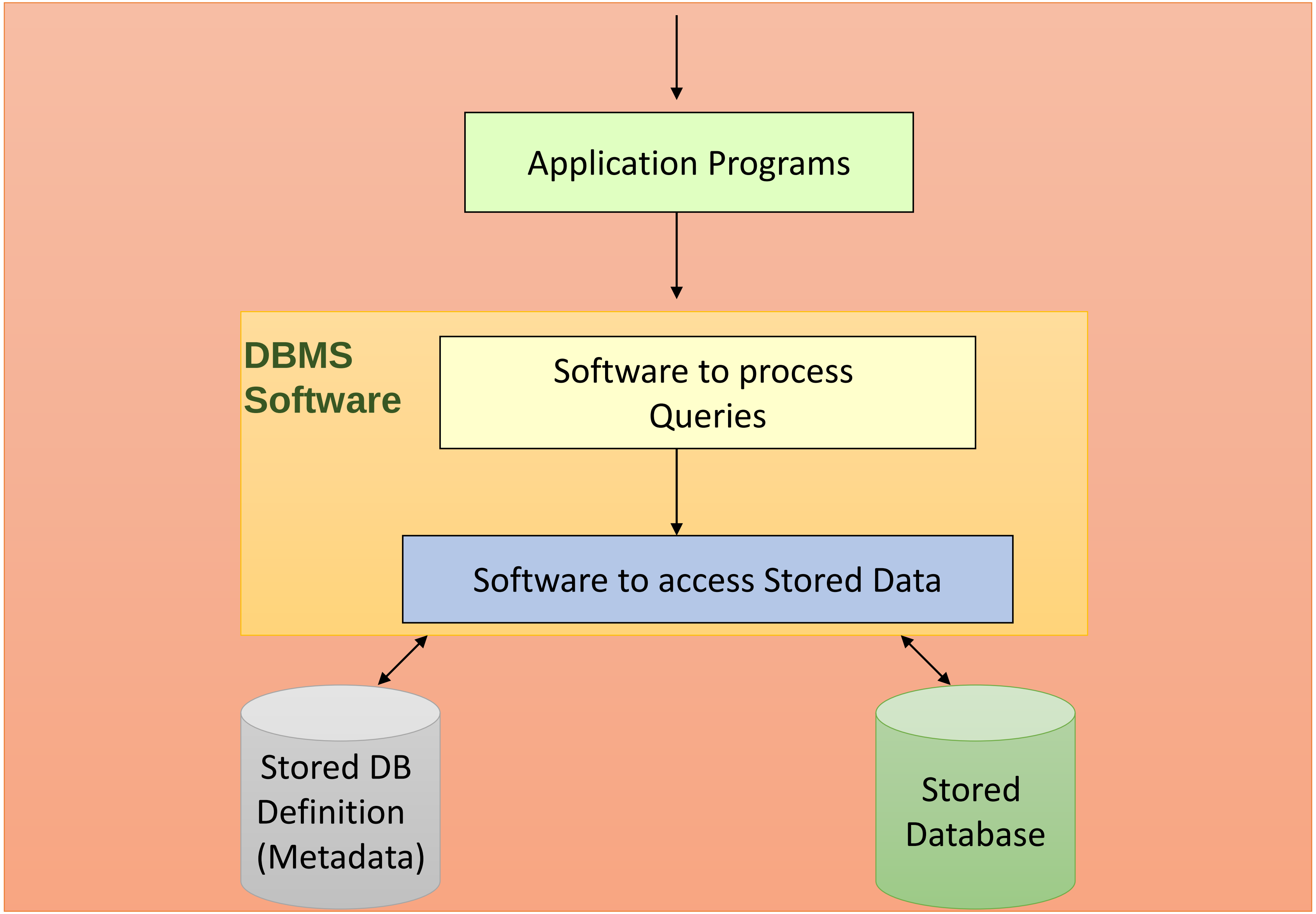

Database
DBMS
create
process
administer
S
Q
L
SAP Applications Over Corporate Network (Client/Server)
eCommerce Applications on Web Server
Web Portal with Reporting Applications
XML Web Services Applications
user(s)

Database System

The DBMS software together with the data itself. Sometimes, the applications are also included.

(Software + Database)

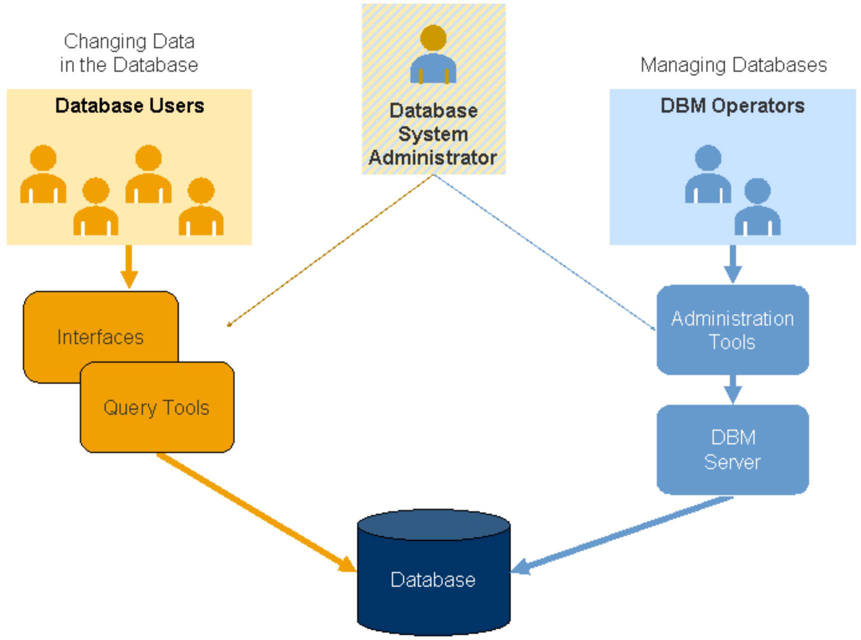
Database System



Application Programs
DBMS Software
Software to process Queries
Software to access Stored Data
Stored DB Definition (Metadata)
Stored Database

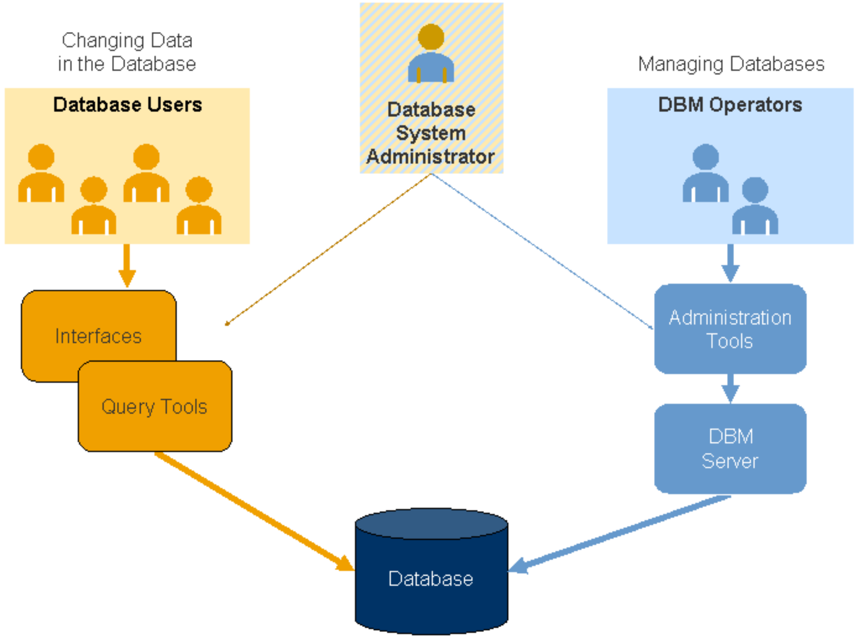
Database Users

- Database Administrator (DBA)
  - Put and implement suitable strategy & policy of the whole database
  - Define internal schema
  - Define security & integrity checks
  - Define backup & recovery procedure
  - Monitor performance
- System Analysts
  - Determine the user requirements

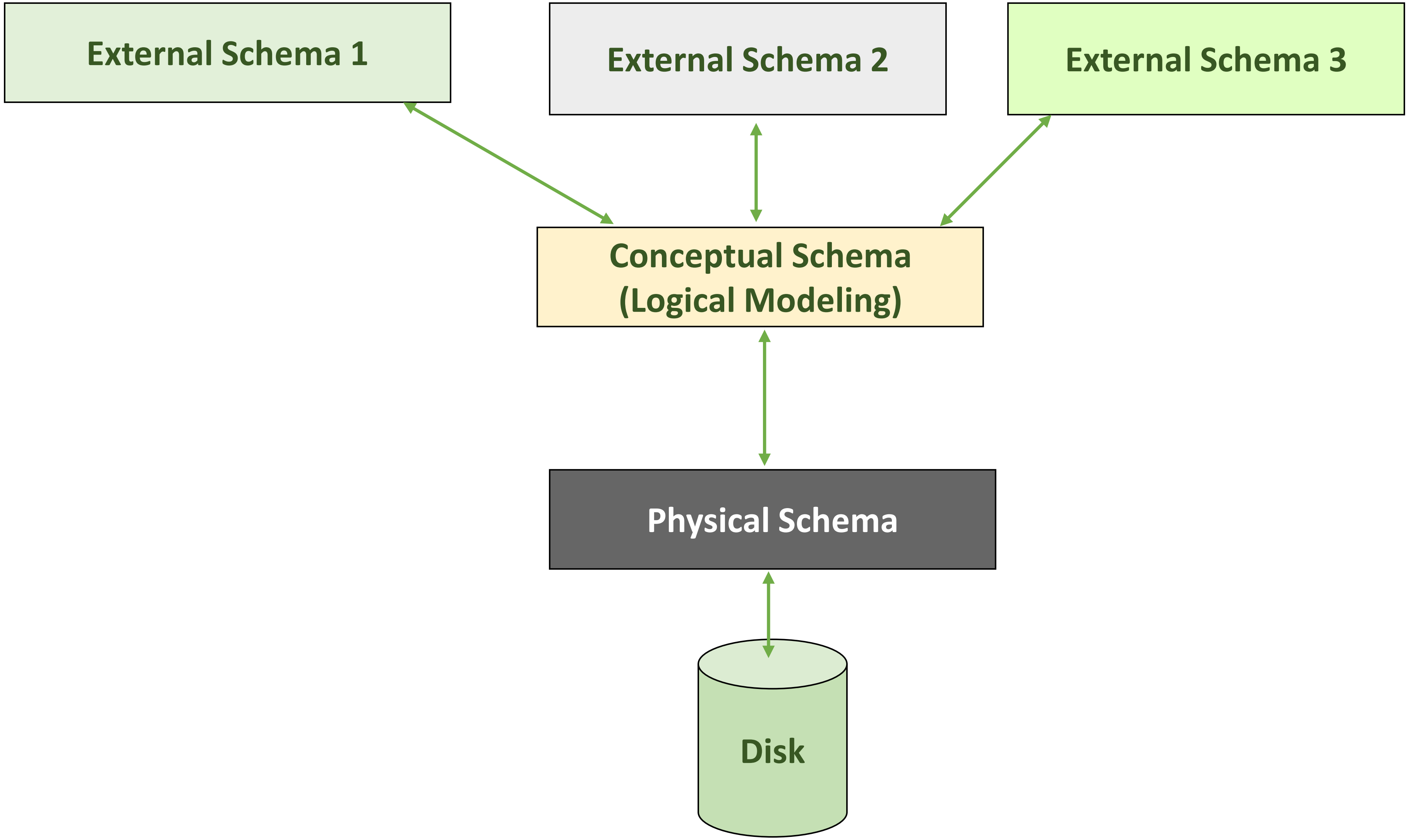


Database Users

- Database Designer
  - Choose the appropriate structure to represent data
- Application programmers
  - Implement the specifications as programs with high level language
  - Testing, Debugging, Documenting & maintaining transactions
- End users
  - Casual
  - Naïve
  - Sophisticated
- Software Testers
- Data analytics & BI team
- SQL Data Scientist , Etc..



DBMS Architecture



Three Level/Schema Architecture

- **External**

  concerned with what data the user will see and how the data will be presented to the user (for each user Views).

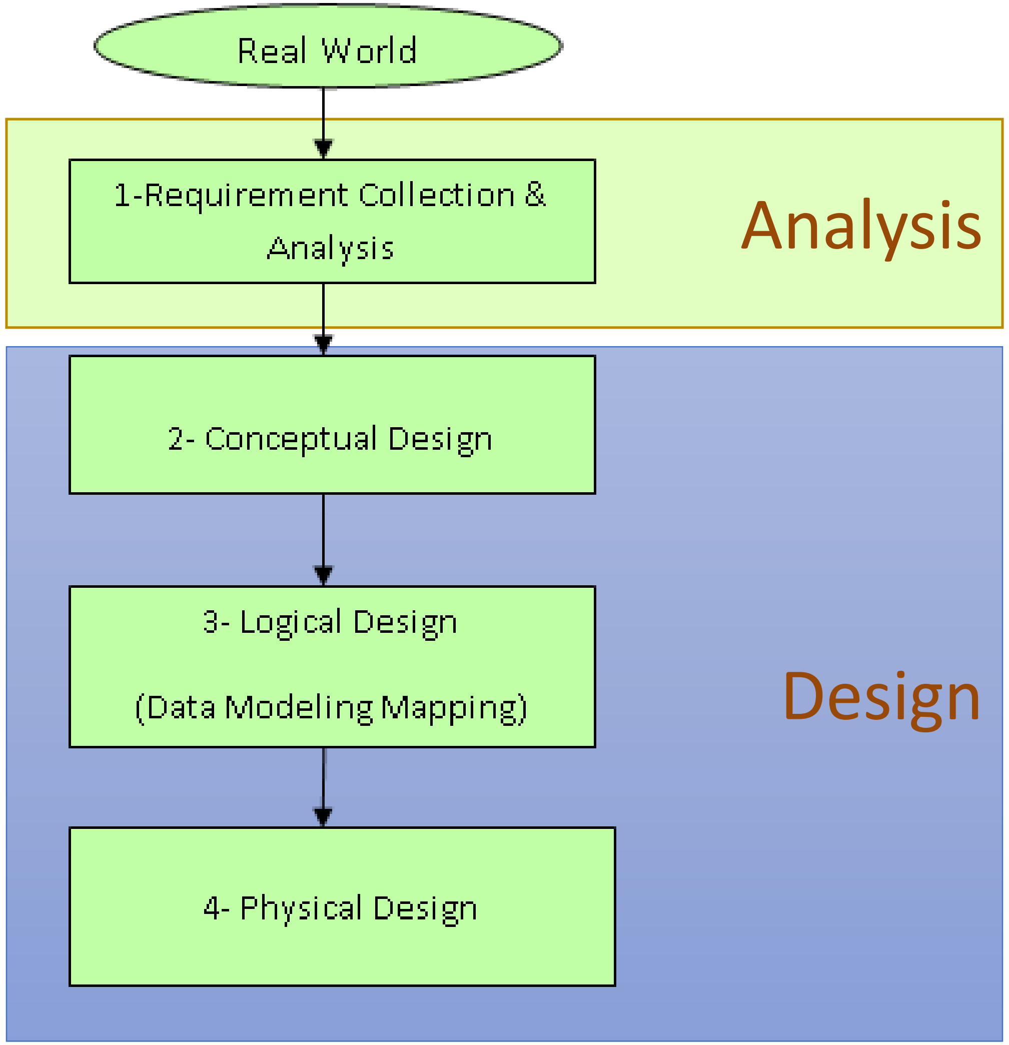
- **Conceptual (The logical model)**

  concerned with what is represented rather than how it is represented.(define database structures such as tables and constraints)

- **Internal (The physical model)**

  emphasis on how the data are represented in the database or on how the data structures are implemented. (Physical storage of database using physical data model)

Database Design Phases



Real World

1-Requirement Collection & Analysis

Analysis

2- Conceptual Design

3- Logical Design

(Data Modeling Mapping)

4- Physical Design

Design

Data Models

- High Level or Conceptual data models provide concepts that are close to the way many users perceive data, entities, attributes and relationships. (Ex. ERD)
- Physical data models describes how data is stored in the computer and the access path needed to access and search for data.

Database Models: Relational vs. NoSQL

Understanding Key Differences, Features, and Use Cases

Relational Databases

1. Store data in tables with rows and columns linked by relationships
2. Introduced by E.F. Codd, supporting powerful SQL queries
3. Enforce strict schemas ideal for structured data
4. Use complex joins to maintain data integrity
5. Best for consistent, transactional systems

Non-Relational Databases (NoSQL)

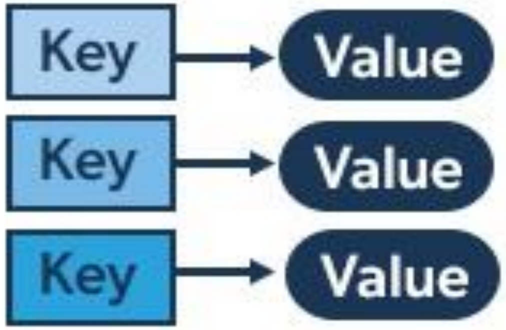
1. Handle large-scale, unstructured, or rapidly changing data
2. Do not require fixed schemas; avoid complex joins
3. Types: Document (MongoDB), Key-Value (Redis), Wide-Column (Cassandra), Graph (Neo4j)
4. Enable better scalability and flexibility
5. Support big data, real-time analytics, and cloud apps (used by Google, Facebook)

Non-relational Database (Nosql)

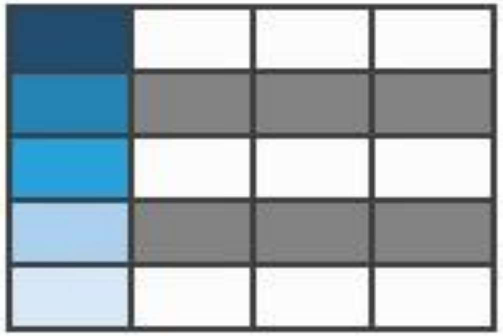
- NoSQL databases stands for Not Only SQL.
- Non-Relational Database: unlike the relational database (doesn’t require a fixed schema), there are no tables, rows, primary keys or foreign keys.
- Instead, the non-relational database uses a storage model optimized for specific requirements of the type of data being stored.
- NoSQL databases avoid joins and are easy to scale.
- Companies like Facebook, Google, and Twitter use NoSQL for their big data and real-time web applications, collecting terabytes of user data every single day.
- NoSQL can use other types of query language rather than SQL.

NoSQL

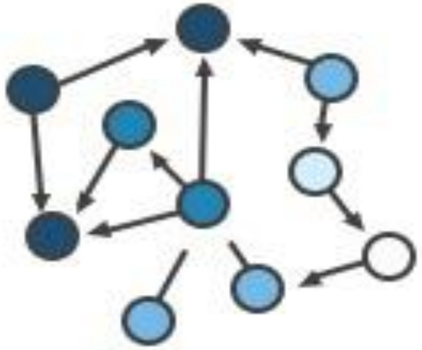
Key-Value



Column-Family

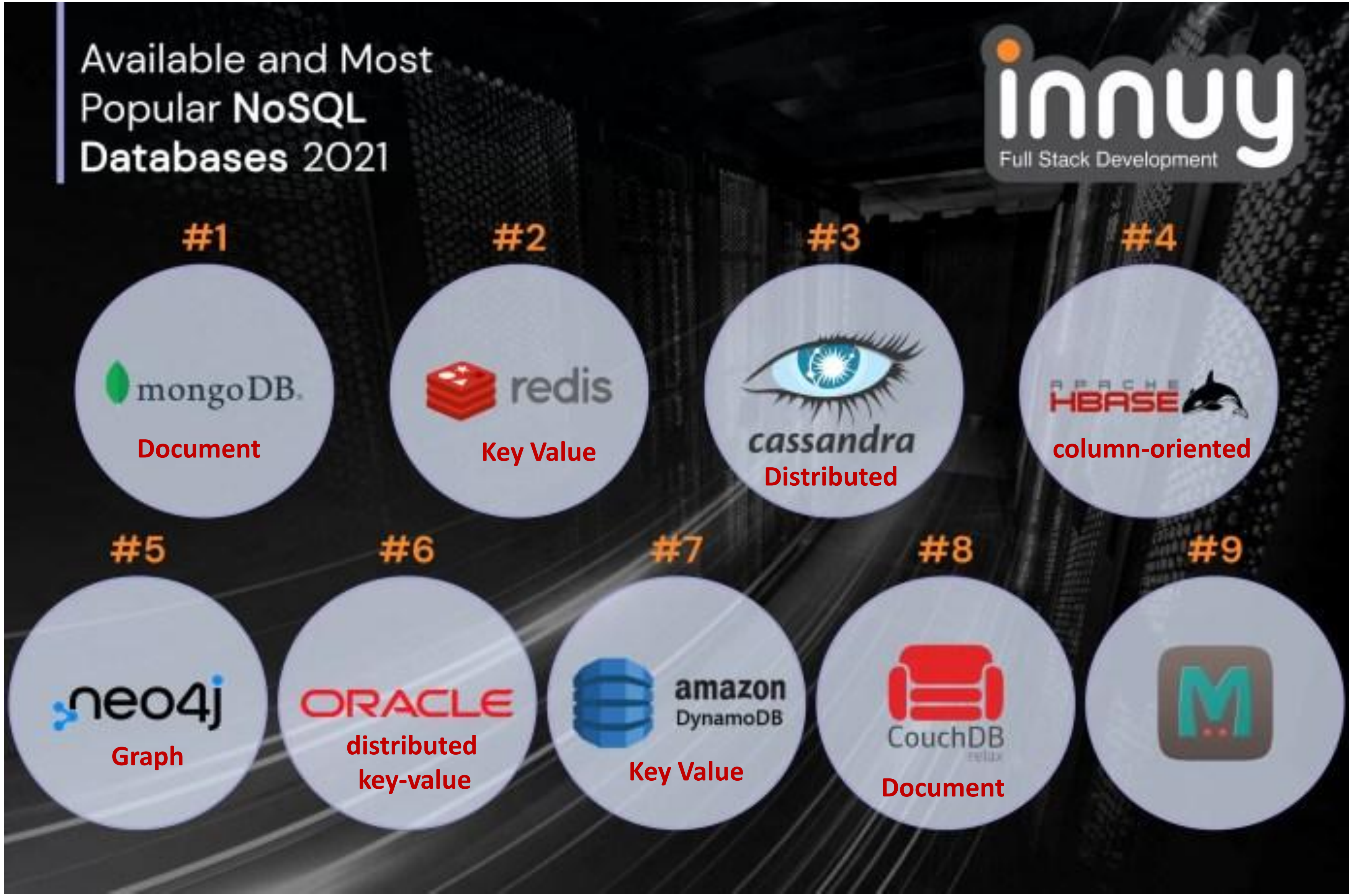


Graph



Document

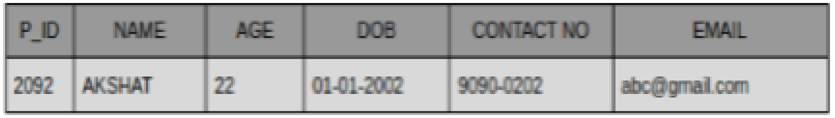
NoSQL Databases



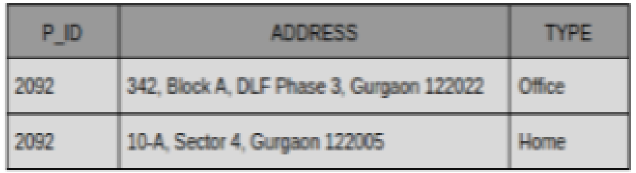
1- Document Oriented

- A document database stores data in JSON, BSON, or XML documents

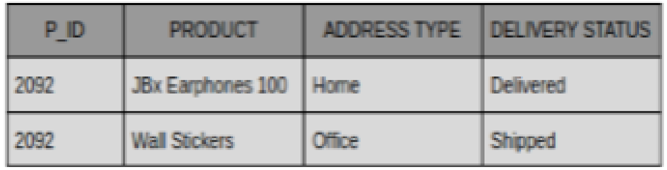
- **Example:** MongoDB



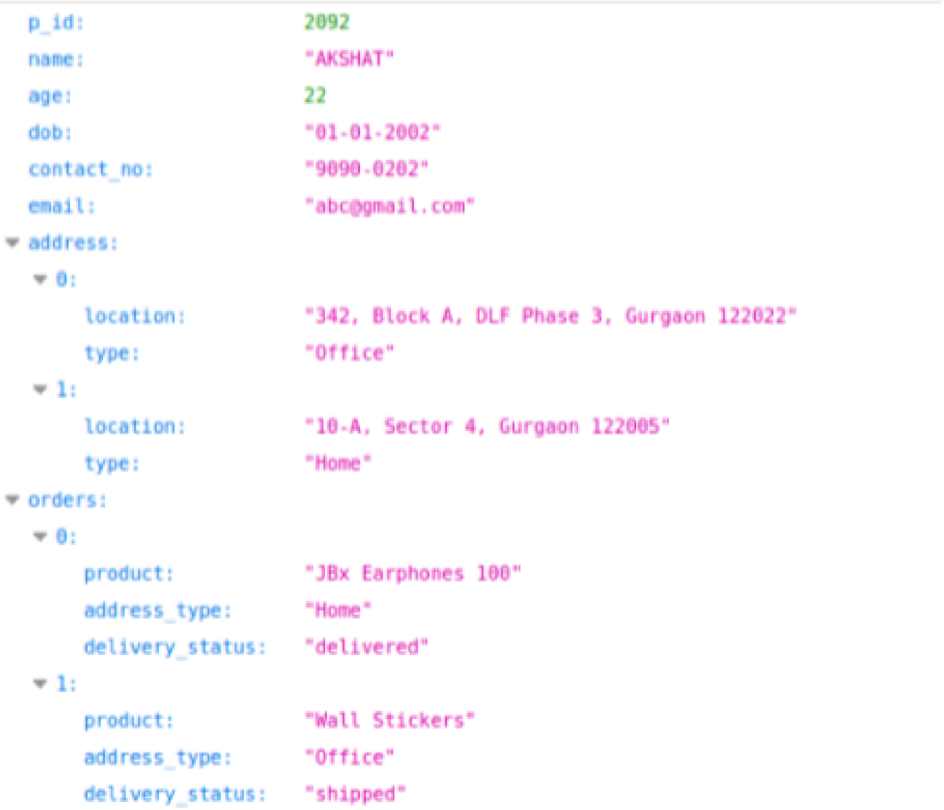
P_ID	NAME	AGE	DOB	CONTACT NO	EMAIL
2092	AKSHAT	22	01-01-2002	9090-0202	abc@gmail.com



P_ID	ADDRESS	TYPE
2092	342, Block A, DLF Phase 3, Gurgaon 122022	Office
2092	10-A, Sector 4, Gurgaon 122005	Home



P_ID	PRODUCT	ADDRESS TYPE	DELIVERY STATUS
2092	JBx Earphones 100	Home	Delivered
2092	Wall Stickers	Office	Shipped



```
p_id:                   2092
name:                   "AKSHAT"
age:                    22
dob:                    "01-01-2002"
contact_no:             "9090-0202"
email:                  "abc@gmail.com"
▾ address:
  ▾ 0:
      location:         "342, Block A, DLF Phase 3, Gurgaon 122022"
      type:             "Office"
  ▾ 1:
      location:         "10-A, Sector 4, Gurgaon 122005"
      type:             "Home"
▾ orders:
  ▾ 0:
      product:          "JBx Earphones 100"
      address_type:     "Home"
      delivery_status:  "delivered"
  ▾ 1:
      product:          "Wall Stickers"
      address_type:     "Office"
      delivery_status:  "shipped"
```

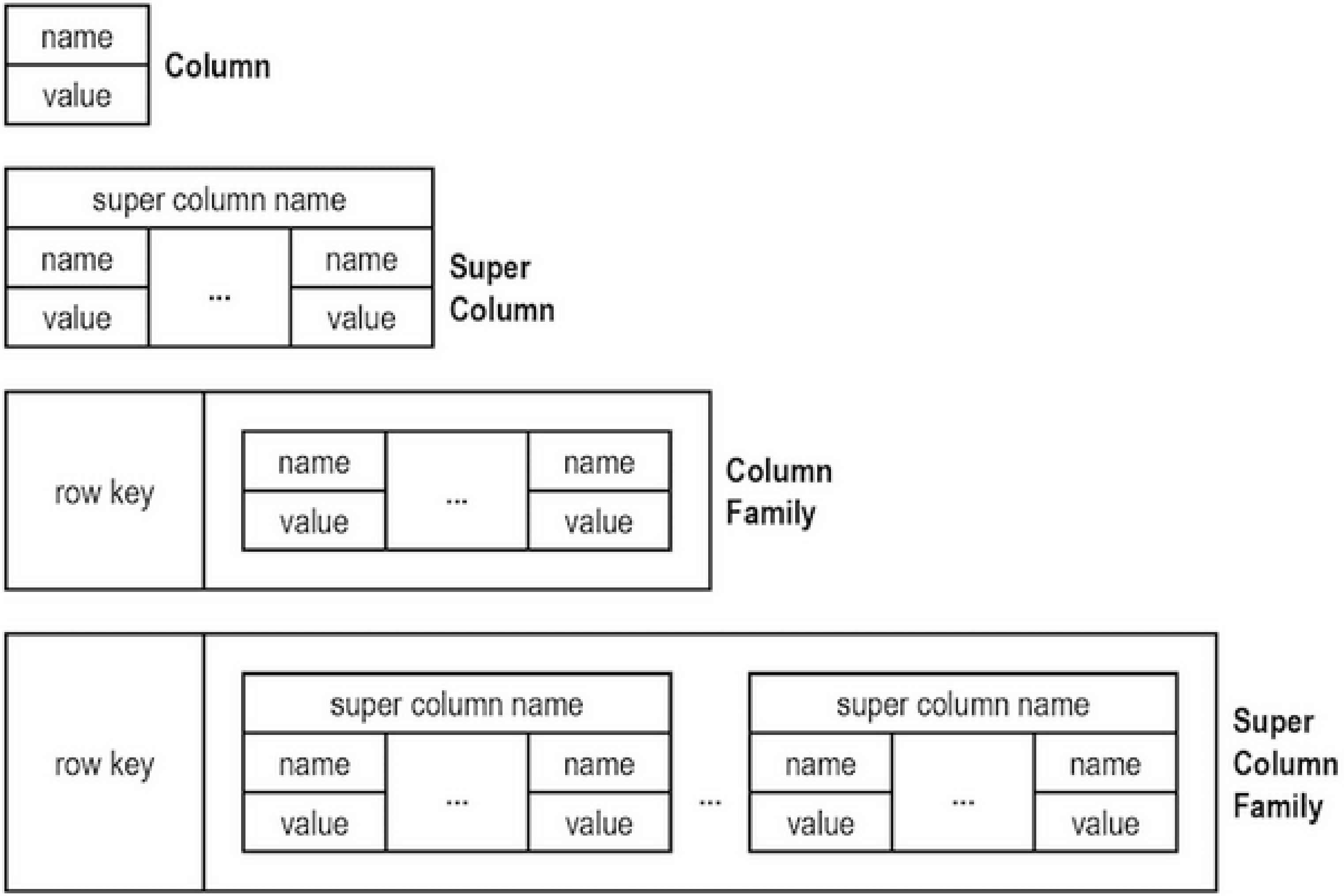
2. Key-Value

Every data element in the database is stored as a key value pair consisting of an attribute name (or "key") and a value

Examples: DynamoDB, Redis, and Aerospike.

3- Columnar Oriented

This database stores the data in records similar to any relational database but it has the ability to store very large numbers of dynamic columns. It groups the columns logically into column families



4- Graph based

- They store the data in the form of nodes and edges. The node part of the database stores information about the main entities like people, places, products, etc., and the edges part stores the relationships between them.

The use of database in AI

- Databases are fundamental to training all sorts of machine learning and artificial intelligence (AI) models.
- Over the last two decades, there has been an explosion of datasets available on the market, making it far more challenging to choose the right one for your tasks.
- At the same time, the larger number of datasets means you can find the perfect fit for whichever application you’re aiming towards.

The use of database in AI

1. For **data collection and storage** through the training of machine learning models, or to deploy real-time prediction endpoints.

2. To **ingest, explore, analyze, and visualize** fast-moving, complex data in milliseconds.

3. For applications that require faster ingestion and more processing speed.

4. To **build a model pipeline platform** that performs data-science-driven model hosting.

5. To address the **volume, velocity, and complex data** governance and management challenges associated with training machine-learning and deep-learning models to save time and optimize resources.

Vector Database

- A vector database stores, manages, and indexes data as high-dimensional numerical vectors, also known as **embeddings**
- These embeddings are mathematical representations of unstructured data like text, images, or audio, and the database allows for fast, similarity-based searches by finding vectors that are closest to each other in a high-dimensional space.
- This enables applications like semantic search, recommendation systems, and natural language processing to understand data contextually, rather than through exact keyword matching.

Break slide.

Questions
?
?
Answers
?