DATABASE CONCEPTS AND ARCHITECTURE

SHIVANI TYAGI

What we covered in Previous Class?

1. Data vs. Information

What we covered in Previous Class?

1. Data vs. Information

Data = raw facts (names, numbers, emails).
Information = processed data with meaning (organized tables, structured records).

What we covered in Previous Class?

1. Data vs. Information

Data = raw facts (names, numbers, emails).
Information = processed data with meaning (organized tables, structured records).

2. What is a Database?

What we covered in Previous Class?

1. Data vs. Information

Data = raw facts (names, numbers, emails).
Information = processed data with meaning (organized tables, structured records).

2. What is a Database?

- An **organized collection of data**, stored in structured form.
- Allows data to be accessed, updated, managed, and retrieved efficiently.

What we covered in Previous Class?

1. Data vs. Information → **Data** = raw facts (names, numbers, emails).
Information = processed data with meaning (organized tables, structured records).

2. What is a Database? →
- An **organized collection of data**, stored in structured form.
- Allows data to be accessed, updated, managed, and retrieved efficiently.

3.Why use Databases?

What we covered in Previous Class?

1. Data vs. Information → **Data** = raw facts (names, numbers, emails).
Information = processed data with meaning (organized tables, structured records).

2. What is a Database? →
- An **organized collection of data**, stored in structured form.
- Allows data to be accessed, updated, managed, and retrieved efficiently.

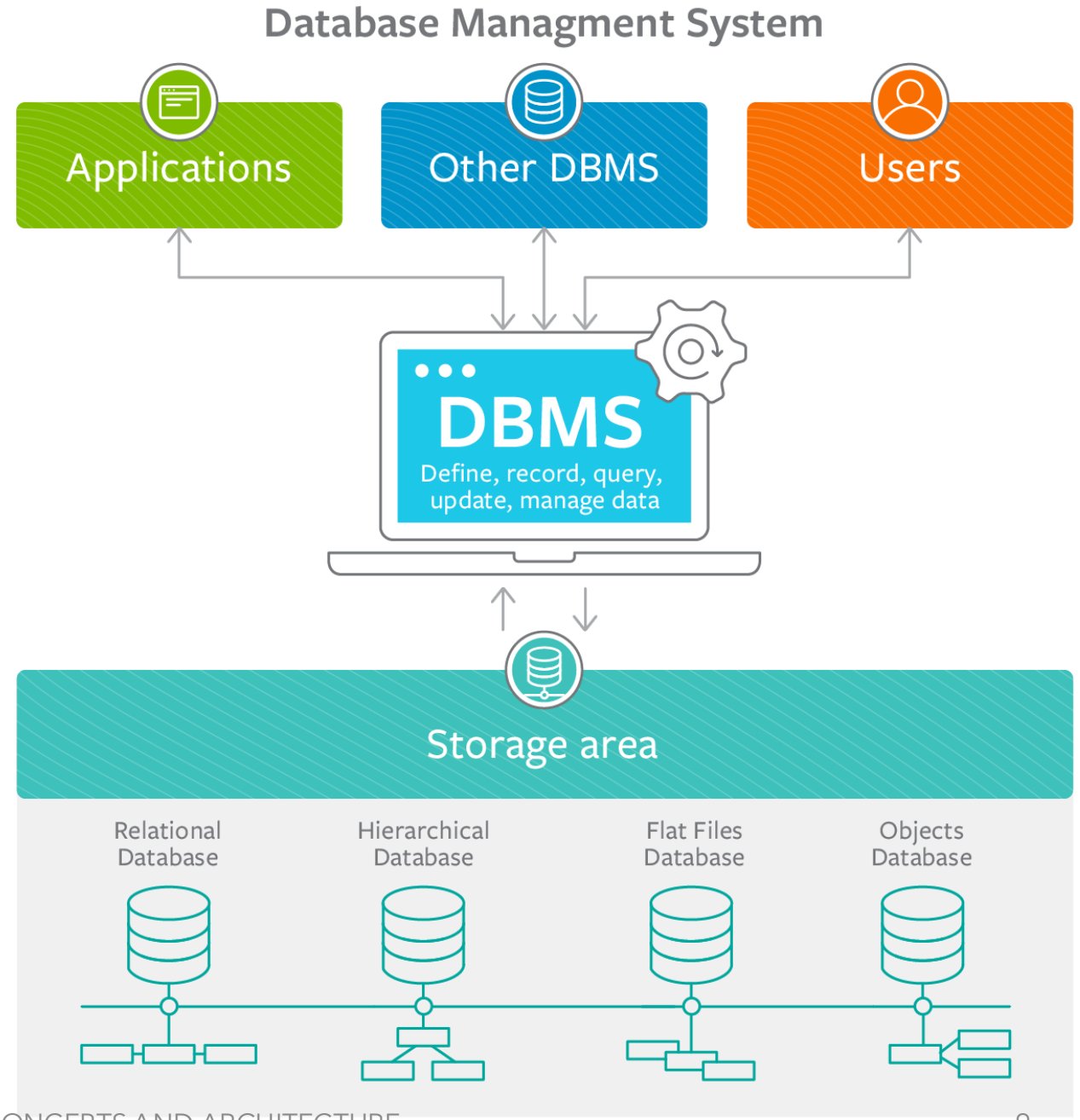
3.Why use Databases? → Databases ensure **consistency, accuracy, security, and fast access.**

LEARNING OBJECTIVES

1. Why do we use DBMS?
2. Users of DBMS
3. Applications of DBMS
4. Popular DBMS
5. Types of Database
6. File System Vs DBMS
7. Data Abstraction
8. Data Independence
9. Database Administrator & Its Role

DATABASE MANAGEMENT SYSTEM

A database management system is a software used to perform different operations, like addition, access, updating, and deletion of the data.



DBMS : Definition

A **Database Management System (DBMS)** consist of :

1. A collection of interrelated and persistent data (usually referred to as the **database (DB)**).
2. A set of application programs used to access, update and manage that data (which form the data **management system**)

WHY DO WE USE DBMS?

Why do we use DBMS?

1. Data Independence and efficient access.
2. Reduced Application development time.
3. Data integrity and security.
4. Uniform data administration.
5. Concurrent access, recovery from crashes.

Need of DBMS

1.Data Redundancy and inconsistency

- Same information may be duplicated in several places.
- All copies may not be updated properly.

2.Difficulty

- In new program to carry out each new task

3. Data isolation :

- Data in different formats.
- Difficult to write new application programs, files and format.

Need of DBMS

4. Security Problems

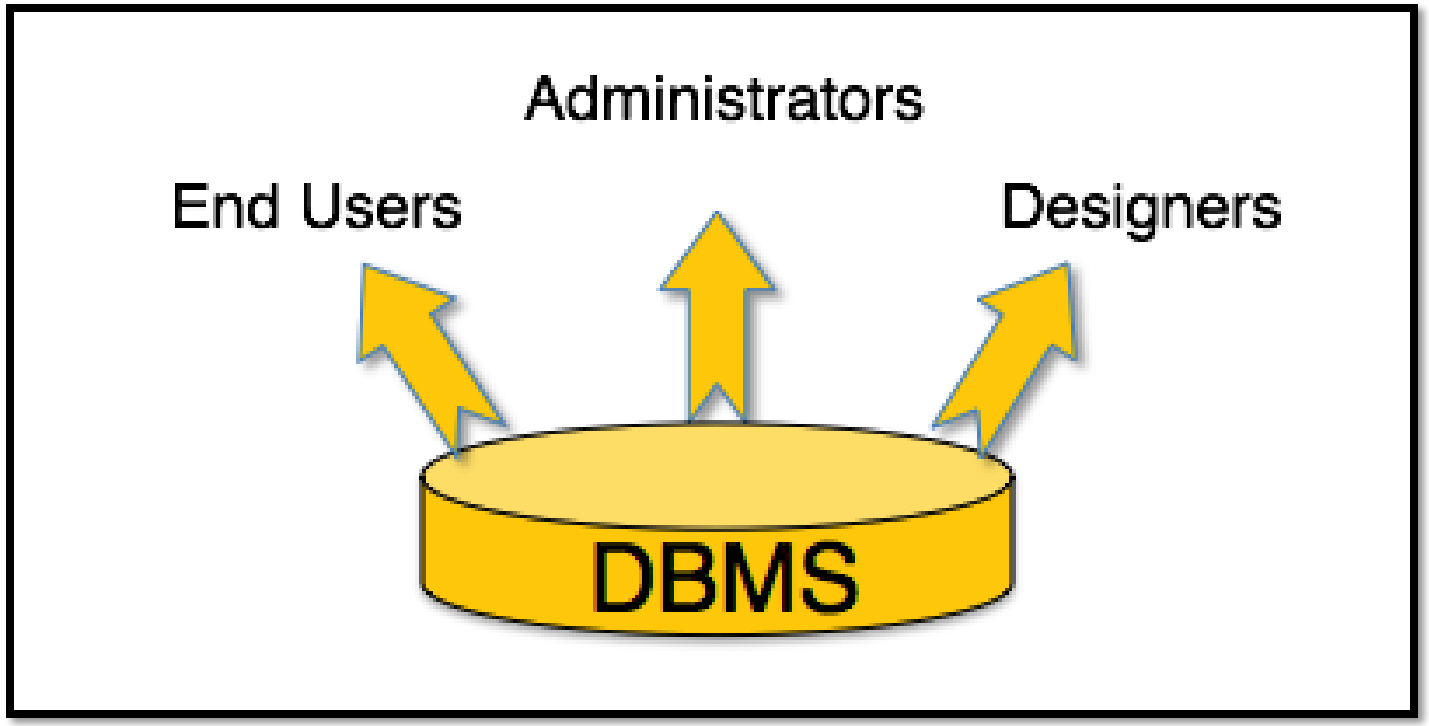
- Every user of the system should be able to access only the data they are permitted to see.
- **Examples**:
  - Payroll people only handle employee records and cannot see customer accounts.
  - Customer Service people only access customer data not payroll data.

5. Integrity Problems

- Data may be required to satisfy constraints.
- Difficult to enforce or to change constraints with the file-processing approach.

USERS OF DBMS

- A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows –



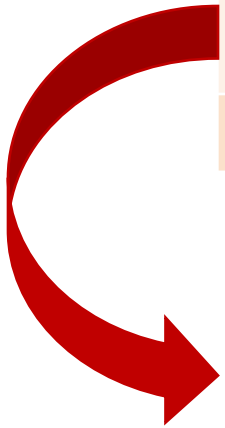
USERS OF DBMS : Application Programmers

COMPONENT NAME	TASK
Application Programmers	Write programs in various programming languages to interact with databases.
Database Administrators	
End-users	

Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.

USERS OF DBMS : Database Administrator

COMPONENT NAME	TASK
Application Programmers	Write programs in various programming languages to interact with databases.
Database Administrators	Responsible for managing the entire DBMS. They are called Database Admin/ DBA
End-users	



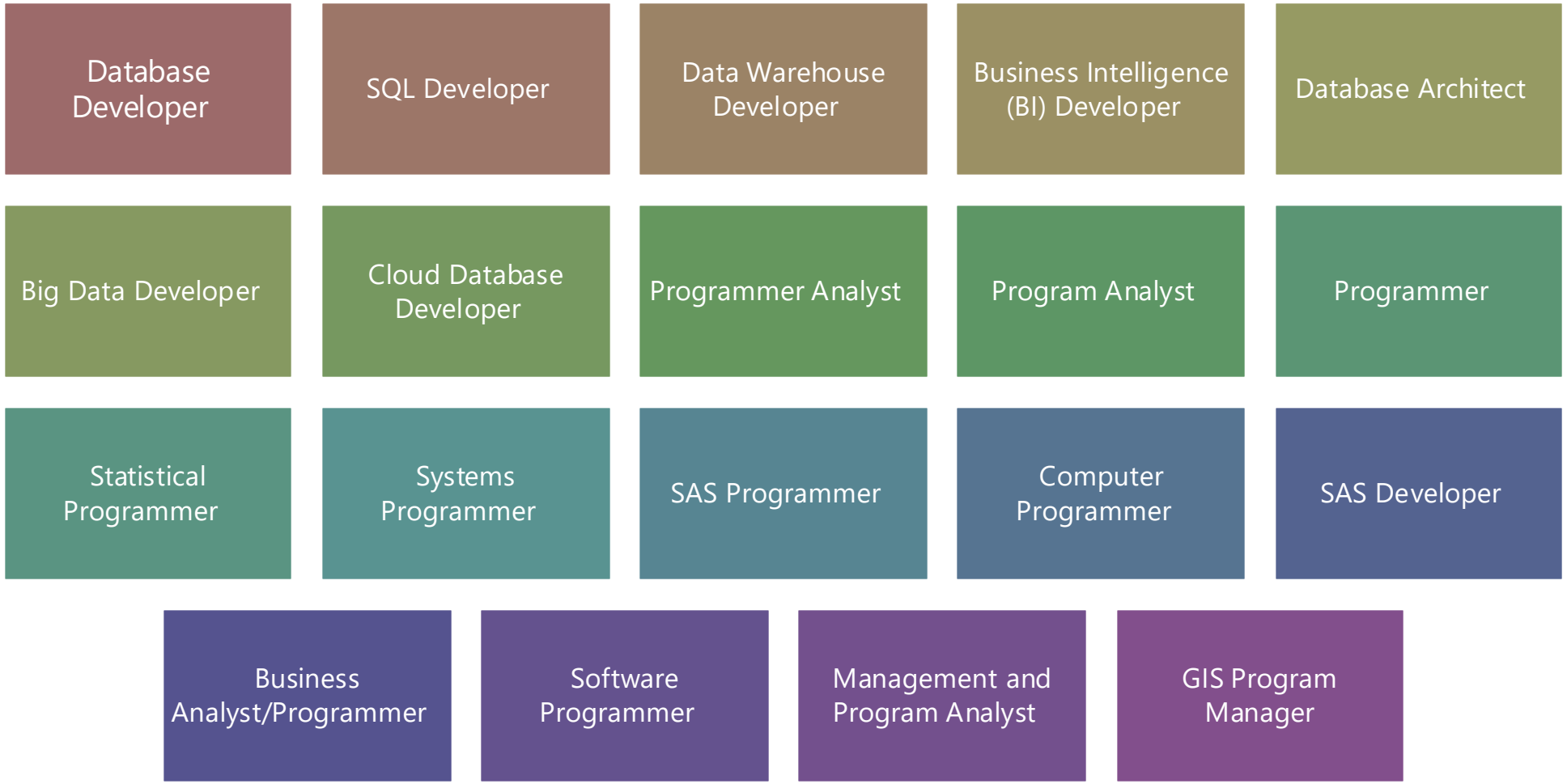
Administrators maintain the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.

USERS OF DBMS : End-Users

COMPONENT NAME	TASK
Application Programmers	Write programs in various programming languages to interact with databases.
Database Administrators	Responsible for managing the entire DBMS. They are called Database Admin/ DBA
End-users	Interact with DBMS & perform operations like retrieving, updating, deleting, etc.

End users are those who reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

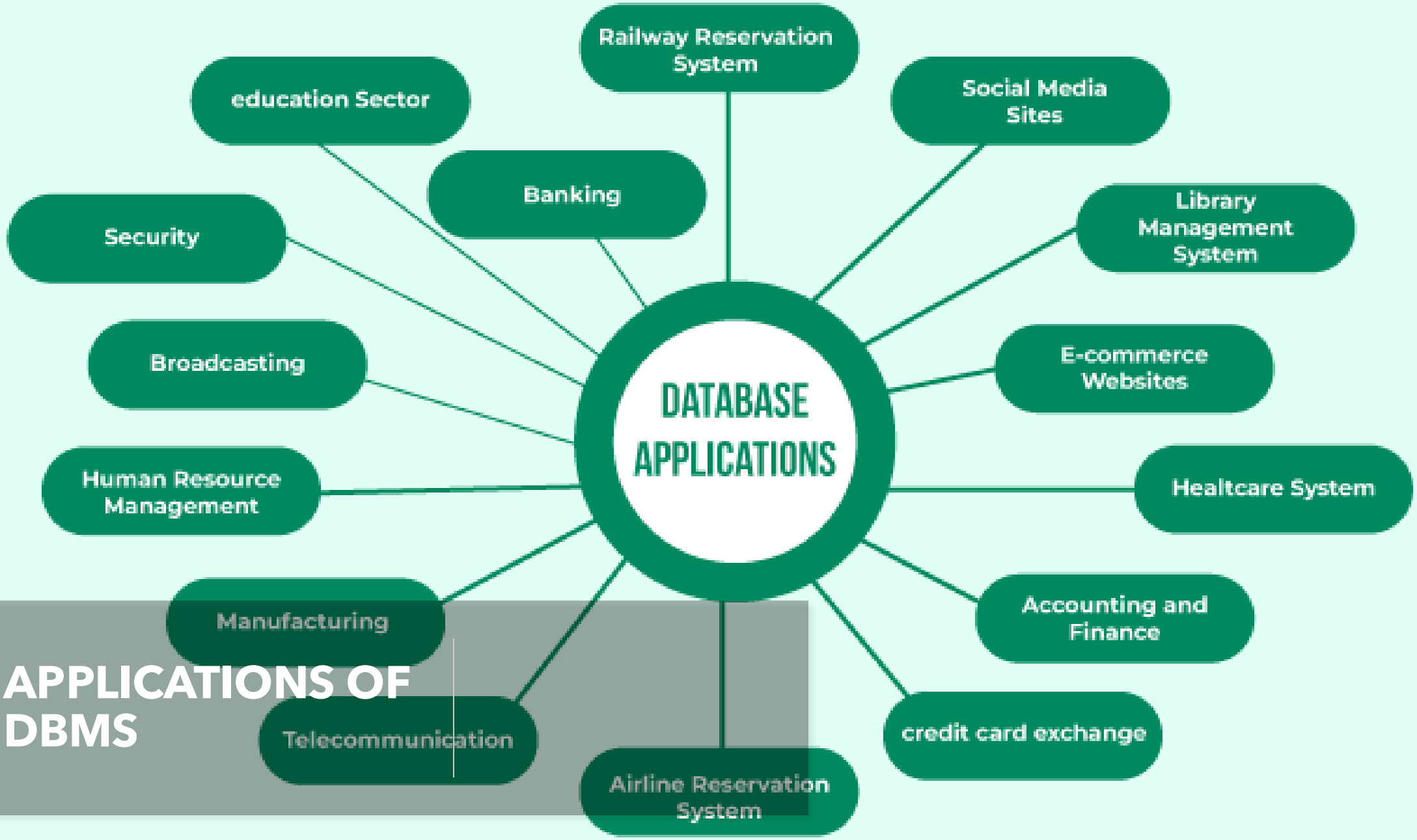
An Application Programmer Can be ....



A Database Administrator Can be ....

Junior Database Administrator	Database Administrator (DBA)	Senior Database Administrator	Database Manager	Database Architect	Data Warehouse Manager
Chief Data Officer (CDO)	Database Administration Manager	Database Analyst	Database Coordinator	Database Engineer	Systems Administrator
IS Administrator	Operations Administrator	Payroll Administrator	Marketing Administrator	Benefits Administrator	Contract Administrator
Business Administrator	Senior Oracle Database Administrator	Database Developer	Data Center Technician	Senior Database Engineer	Database Consultant

APPLICATIONS OF DBMS



Sector	Typical Use of DBMS
Railway Reservation System	Stores ticket bookings, seat availability, train schedules, and arrival/departure status; updates and surfaces delay information.
Library Management System	Maintains complete catalog (title, author), book availability, issue/return tracking, due dates, and member records.
Banking	Manages customer accounts and transactions, balances, loan records, auditing, and regulatory reporting.
Education Sector	Handles student registrations, courses, grades, attendance, fees, results, and online exam data.
Credit Card Exchanges	Records purchases/authorizations, settlements, and generates monthly statements; supports fraud monitoring.
Social Media Sites	Stores user profiles, posts, comments, connections, and feeds; supports search, personalization, and analytics.
Broadcast Communications	Keeps call detail records and produces monthly postpaid bills.
Accounting and Finance	Stores sales, invoices, ledgers, and positions in financial instruments (e.g., stocks, bonds) for reporting and analysis.

Sector	Typical Use of DBMS
E-Commerce Websites	Manages product catalog, inventory, carts, orders, payments, invoices/receipts, and customer profiles.
Human Resource Management	Maintains employee records, payroll, tax/benefits information, attendance, and employment history.
Manufacturing	Tracks bills of materials, inventory, procurement, orders, quantities, and supply-chain operations.
Airline Reservation System	Stores flight schedules, bookings, seat inventory, and live status (departure/arrival/delays).
Healthcare System	Manages patient data (EMR/EHR), appointments, test results, billing, and insurance claims.
Security	Enforces role-based access, authentication/authorization, encryption, auditing, and access logs.
Telecommunication	Manages high-volume data for billing, customer information, usage, and network optimization.

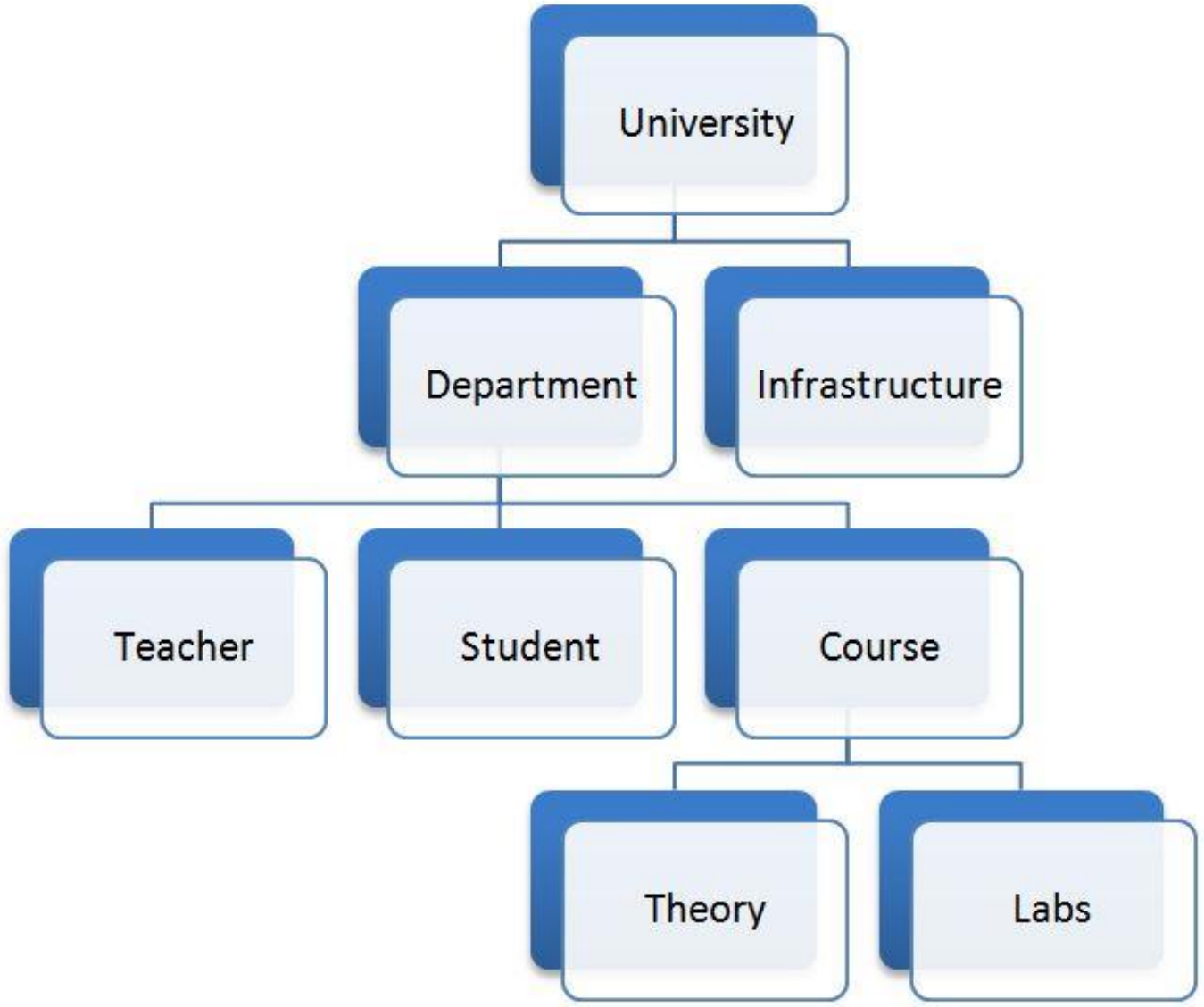
POPULAR DBMS

TYPES OF DATABASE

- Hierarchical Database
- Network Database
- Relational Database
- Object-Oriented Database
- Distributed Database
- Centralized Database
- Cloud Database

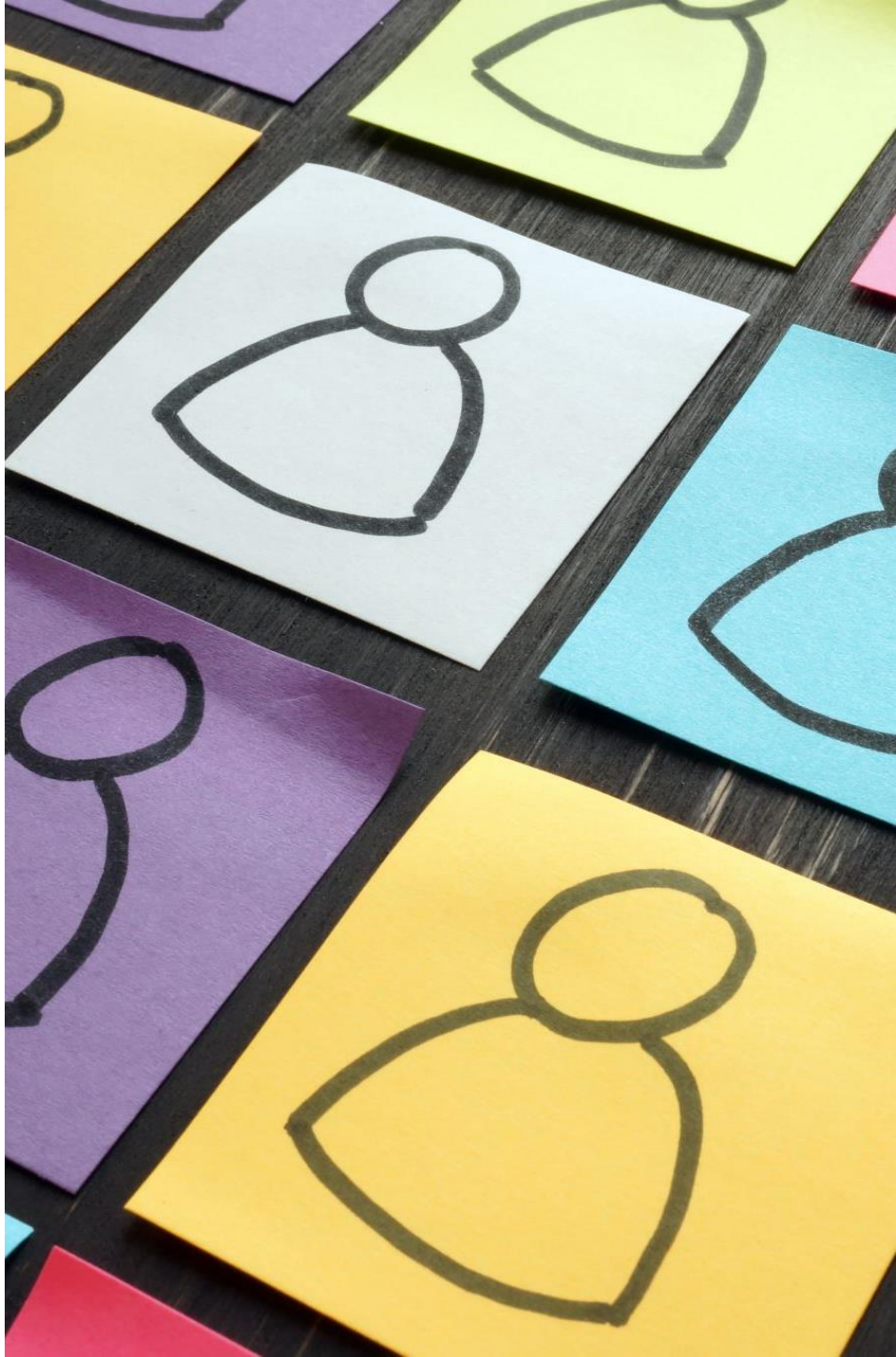
HIERARCHICAL DATABASE

- Hierarchical DBMs were popular from the late 1960s, with the introduction of **IBM's INFORMATION MANAGEMENT SYSTEM (IMS) DBMS**, through 1970.



HIERARCHICAL DATABASE

- These databases organize data in a tree-like structure with parent-child relationships. Each parent can have many children, but each child has only one parent.
- This model is like a file system with folders and subfolders.
- It is efficient for storing data with a clear hierarchy, such as organizational charts or file directories.
- Navigation is fast and predictable due to the fixed structure.
- It lacks flexibility and difficult to restructure or handle complex many-to-many relationships.

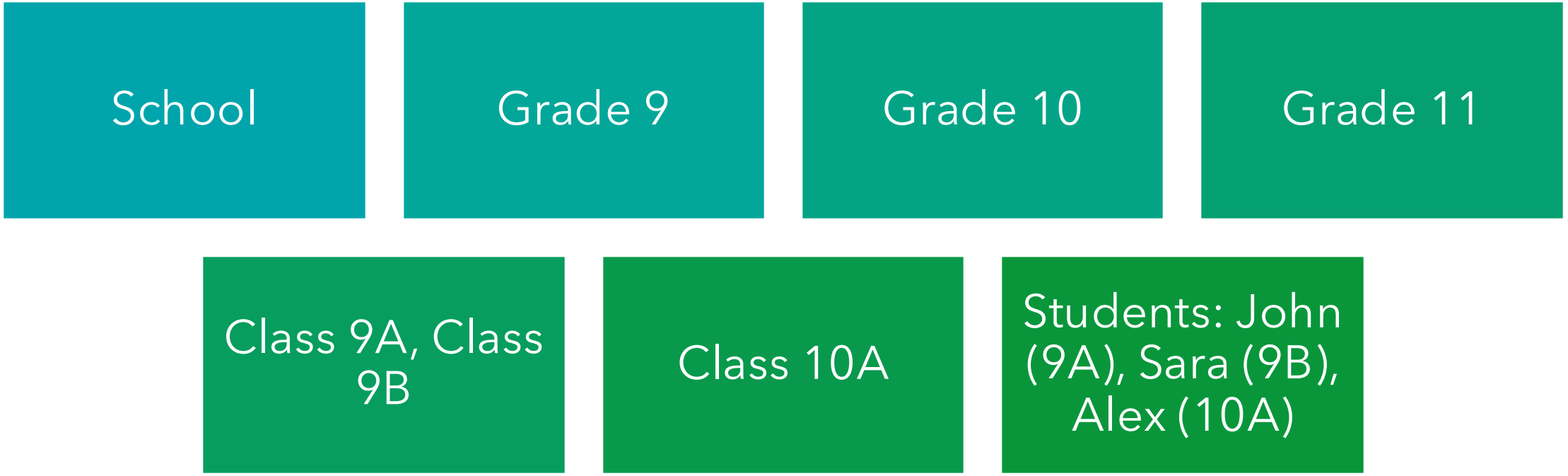


HIERARCHICAL DATABASE – Real Industry Use Cases

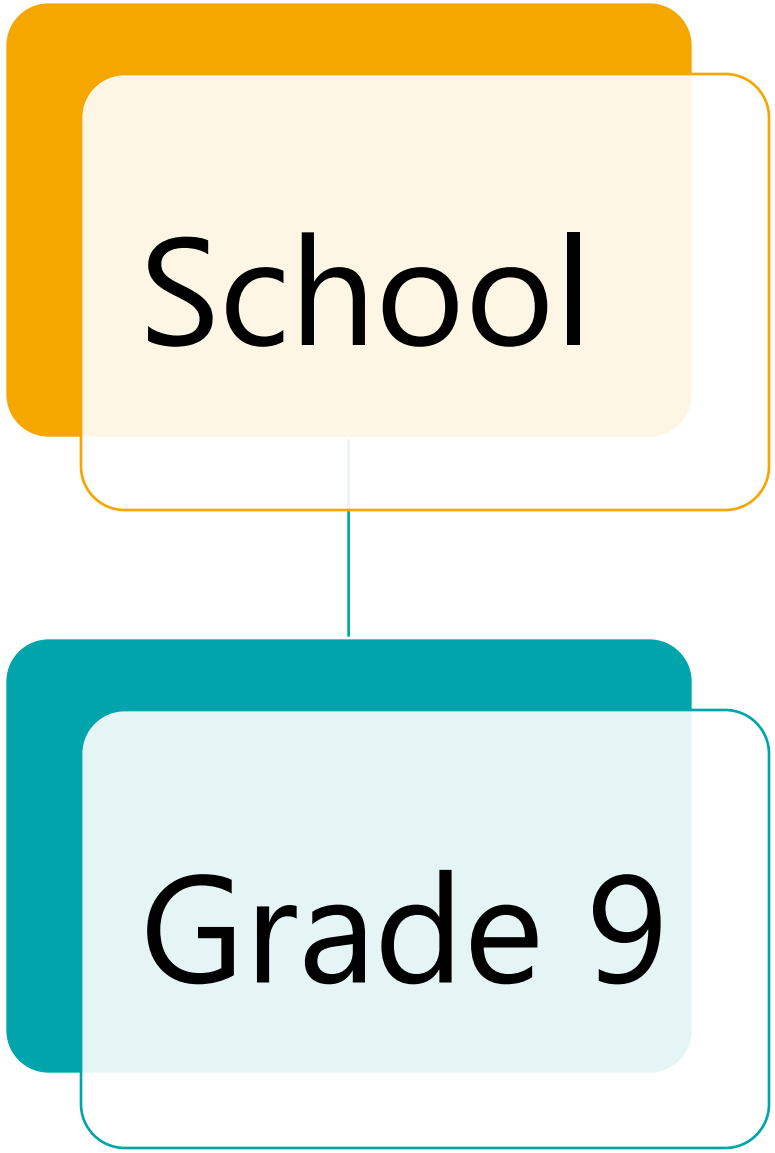
IBM IMS used by NASA Apollo program for inventory & billing management; IMS remains widely used in government and banking.

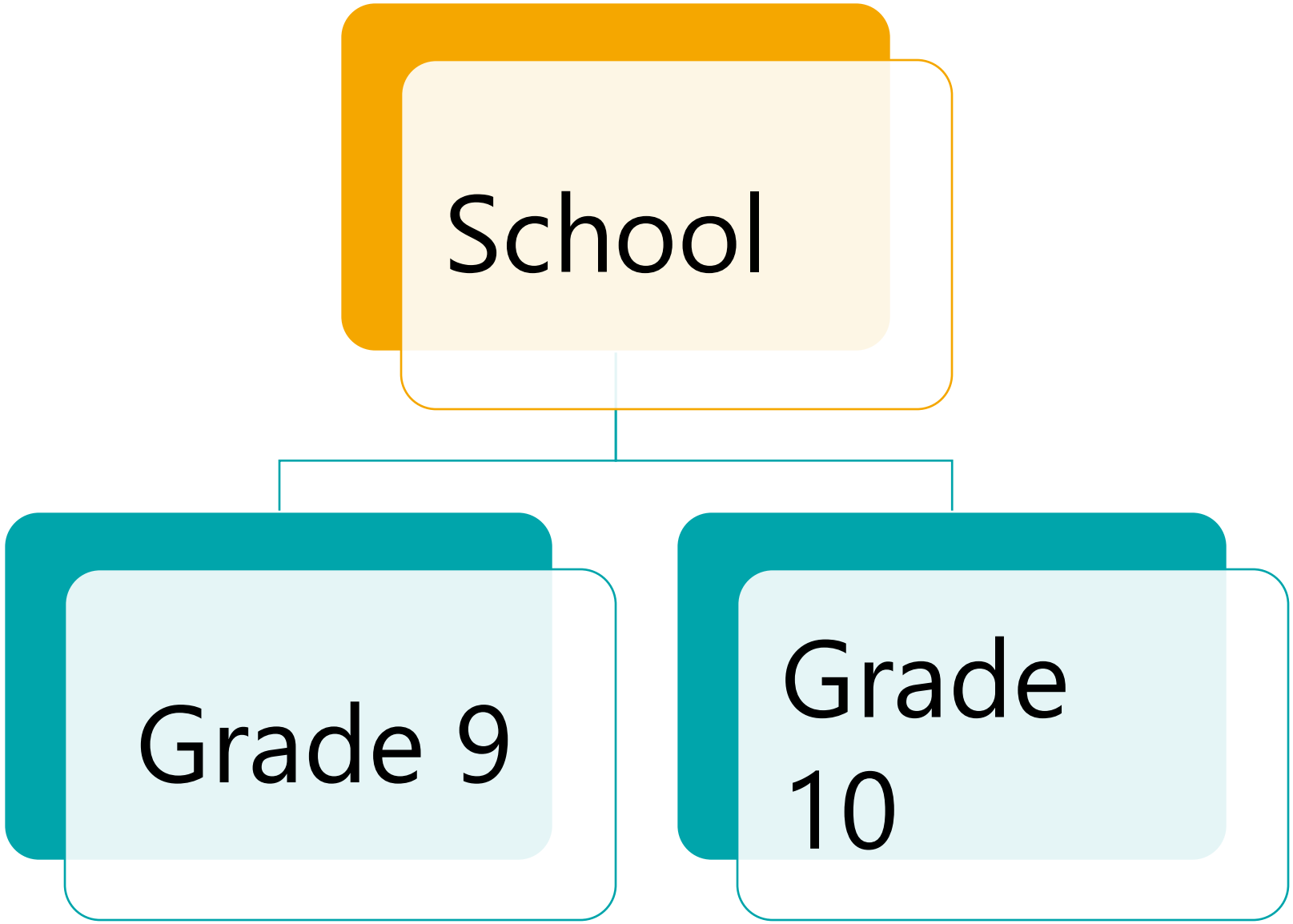
File systems, organizational charts, Windows Registry commonly use hierarchical structures.

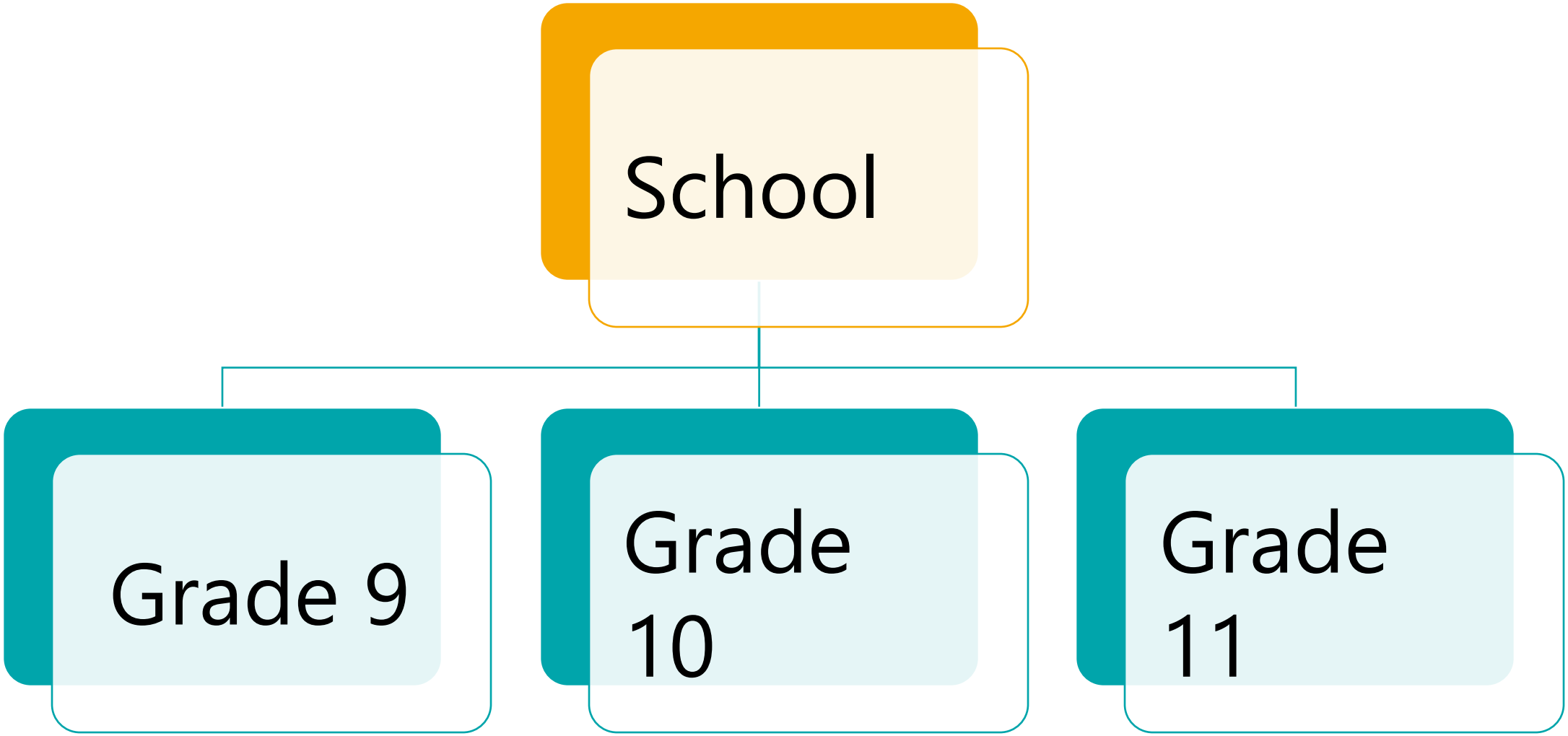
HANDS ON : CREATE HIERARCHICAL DATABASE

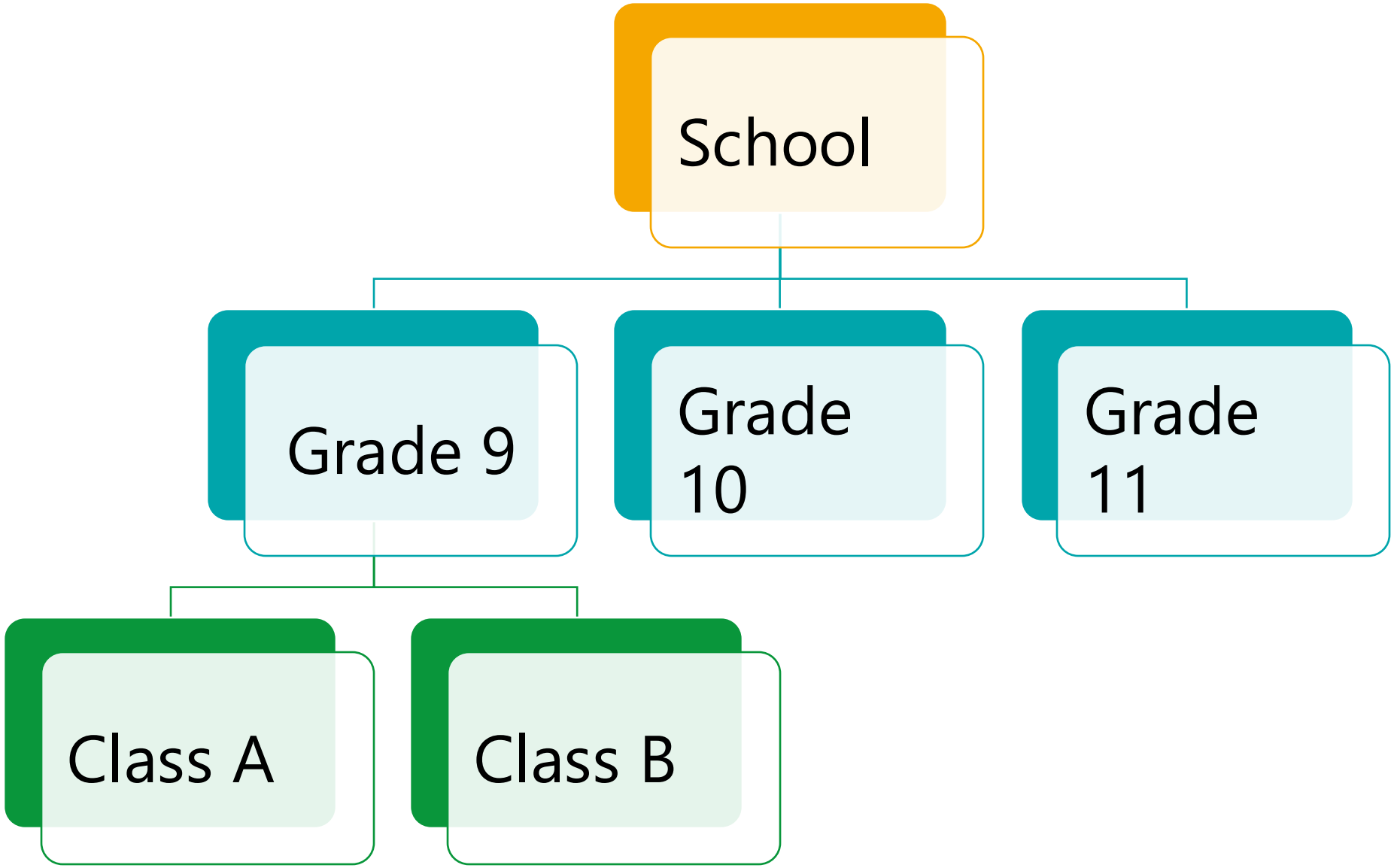


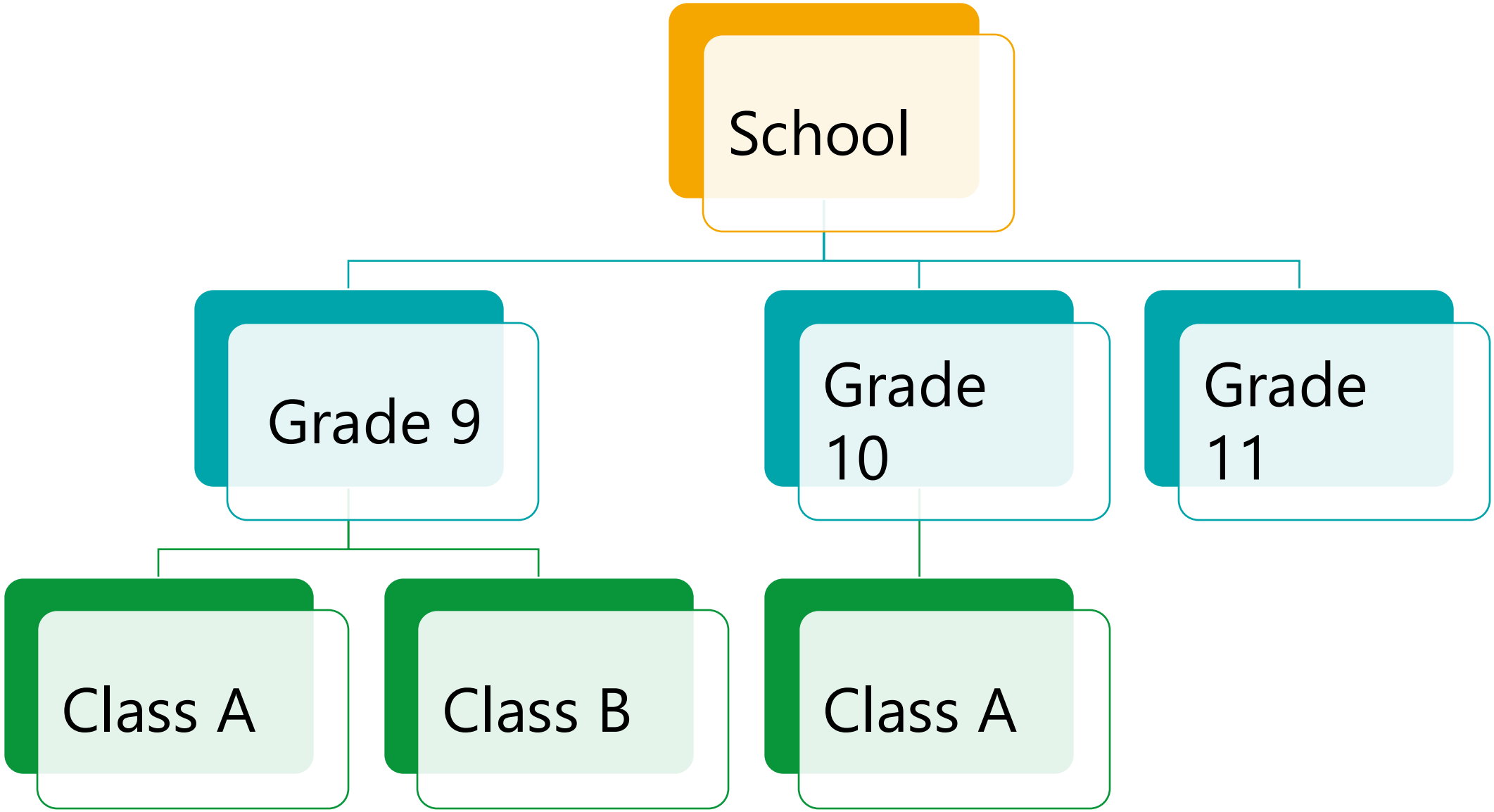
School

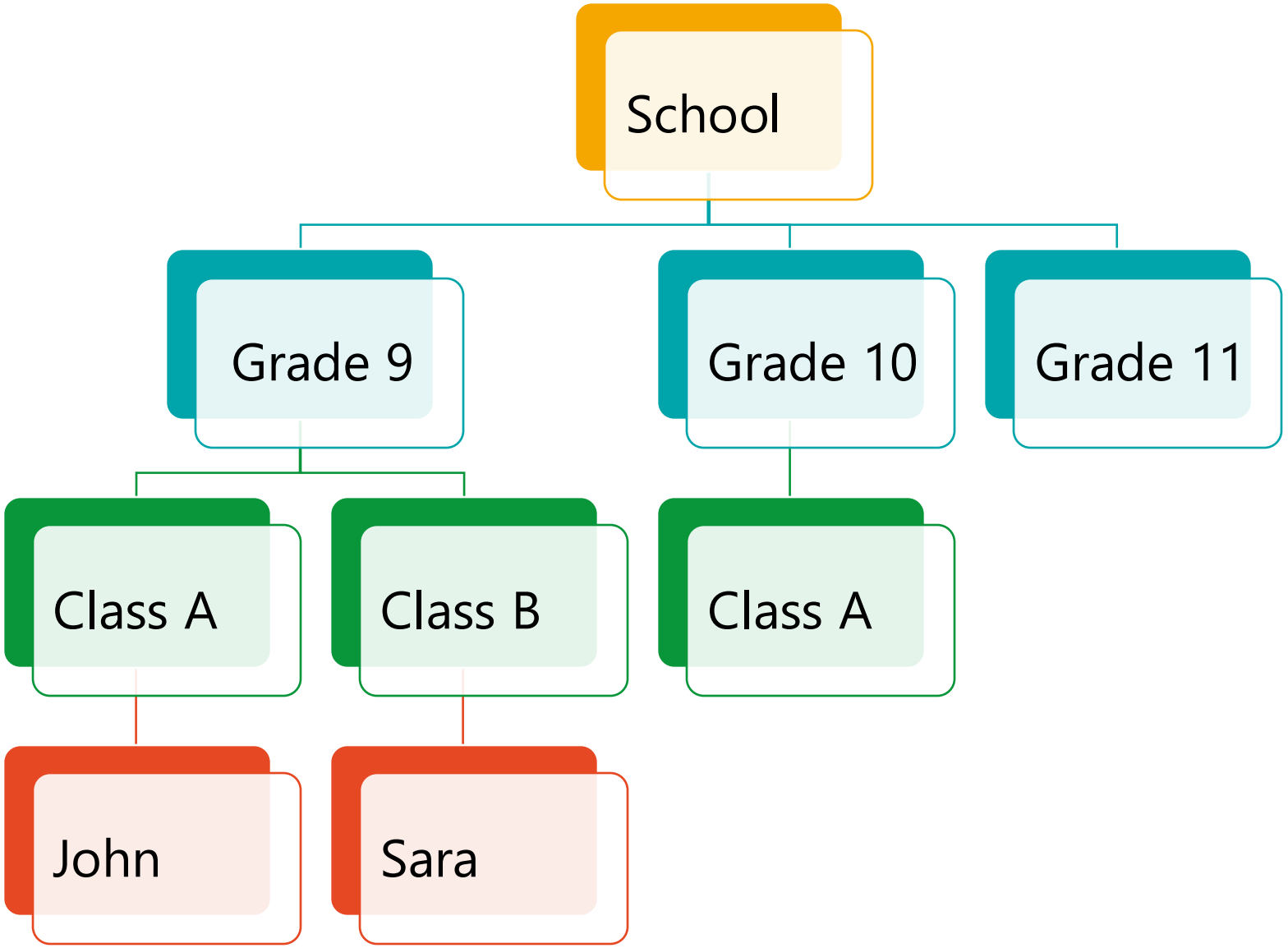

School
Grade 9

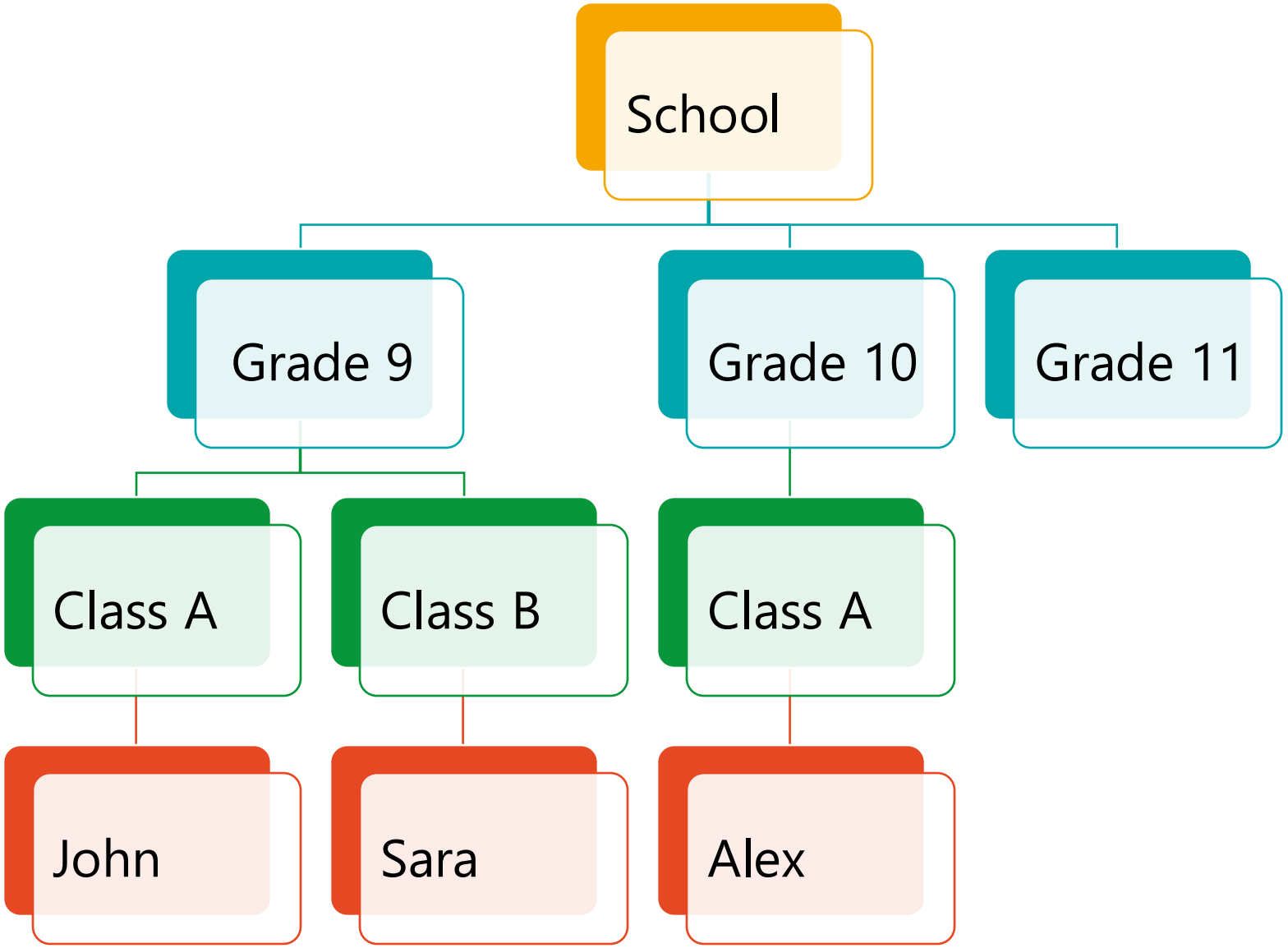

School
Grade 9
Grade 10

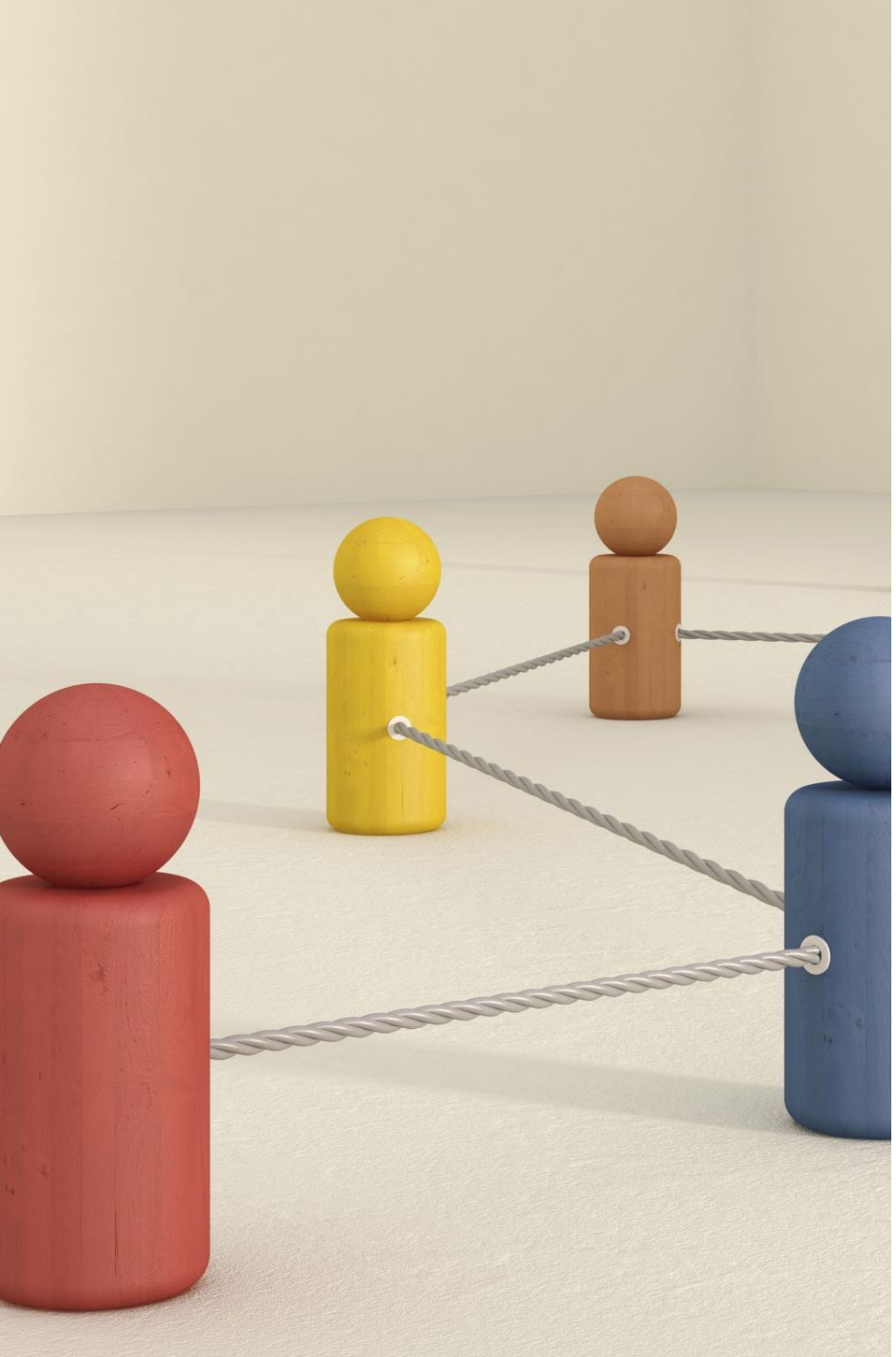

School
Grade 9
Grade 10
Grade 11


School
Grade 9
Grade 10
Grade 11
Class A
Class B


School
Grade 9
Grade 10
Grade 11
Class A
Class B
Class A

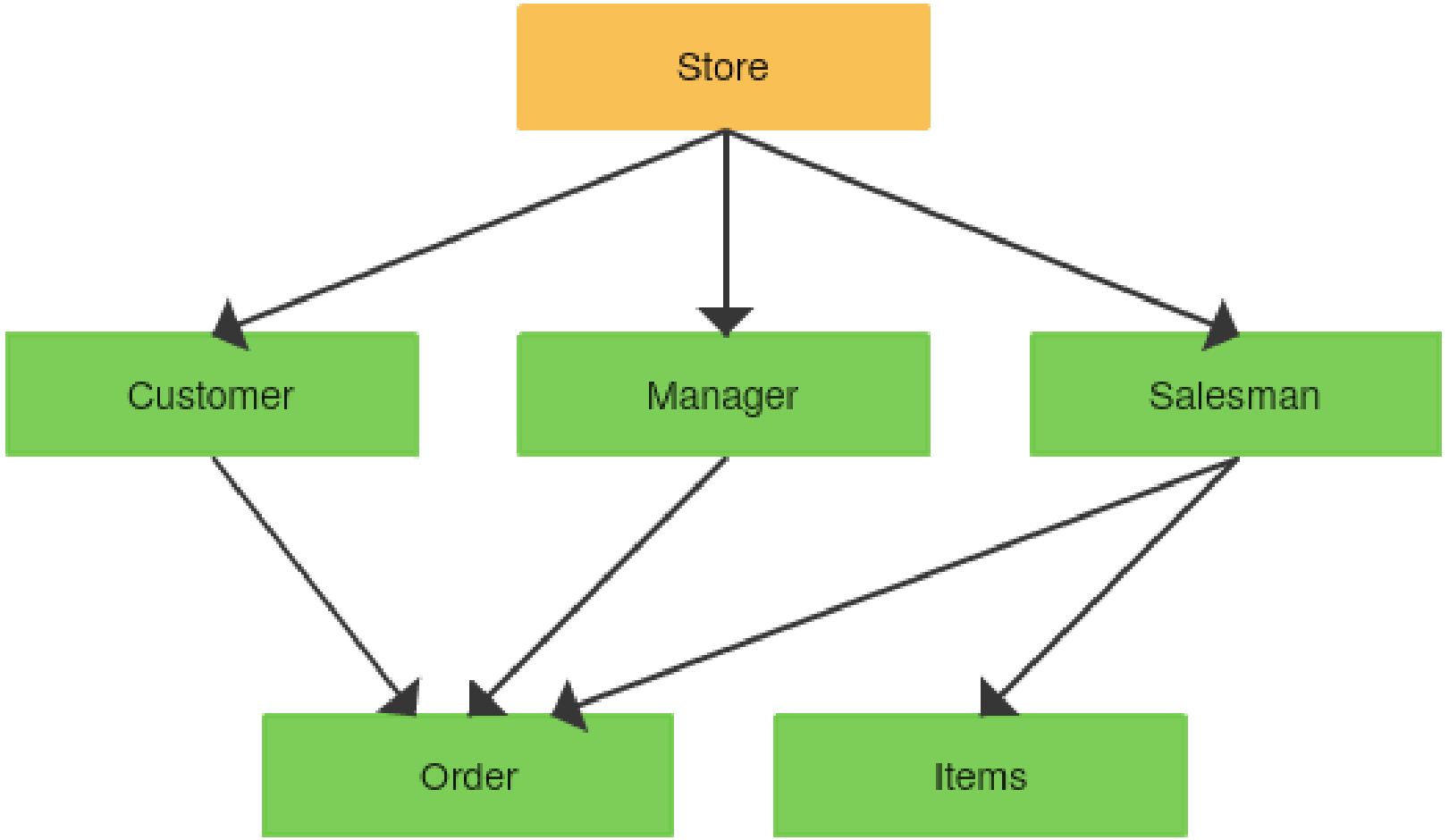

School
Grade 9
Grade 10
Grade 11
Class A
Class B
Class A
John
Sara


School
Grade 9
Grade 10
Grade 11
Class A
Class B
Class A
John
Sara
Alex



NETWORK DATABASE

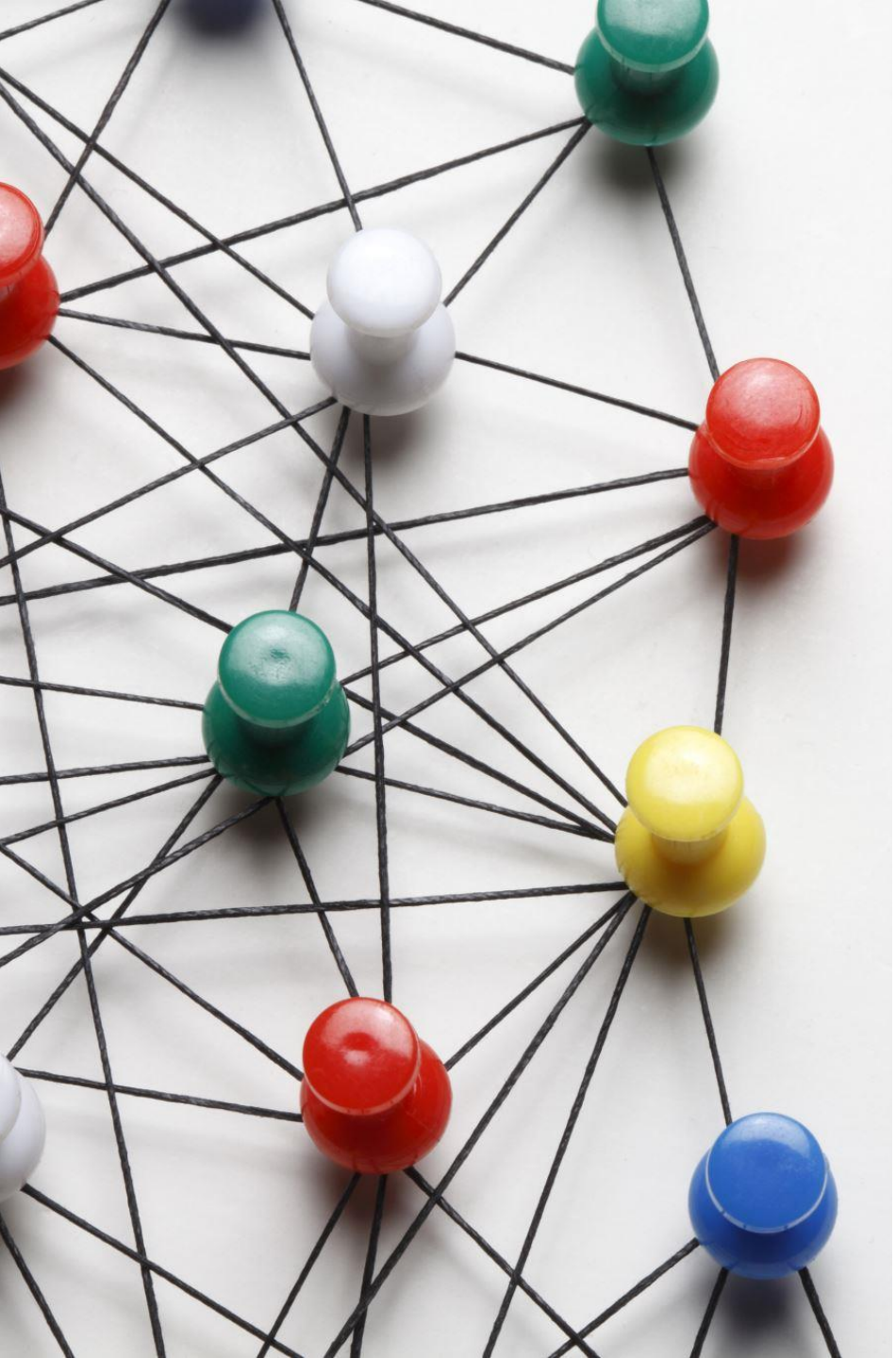
- The popularity of the network database model coincided with the Hierarchical database. Some data naturally modeled with more than one parent per child.
- So, the network model permitted the modeling of many-to-many relationships in data.
- Unlike the hierarchical database, it allows each record to have multiple children and parent nodes to form a generalized graph structure.


Store
Customer
Manager
Salesman
Order
Items

NETWORK DATABASE – Real Industry Use Cases

Banking, telecommunications, and manufacturing industries use network databases to manage deeply interconnected data.

Classic systems include **IDMS, TurboIMAGE, Raima Database Manager**, still used for legacy financial and enterprise apps



Mini-Quiz: Network Database Model

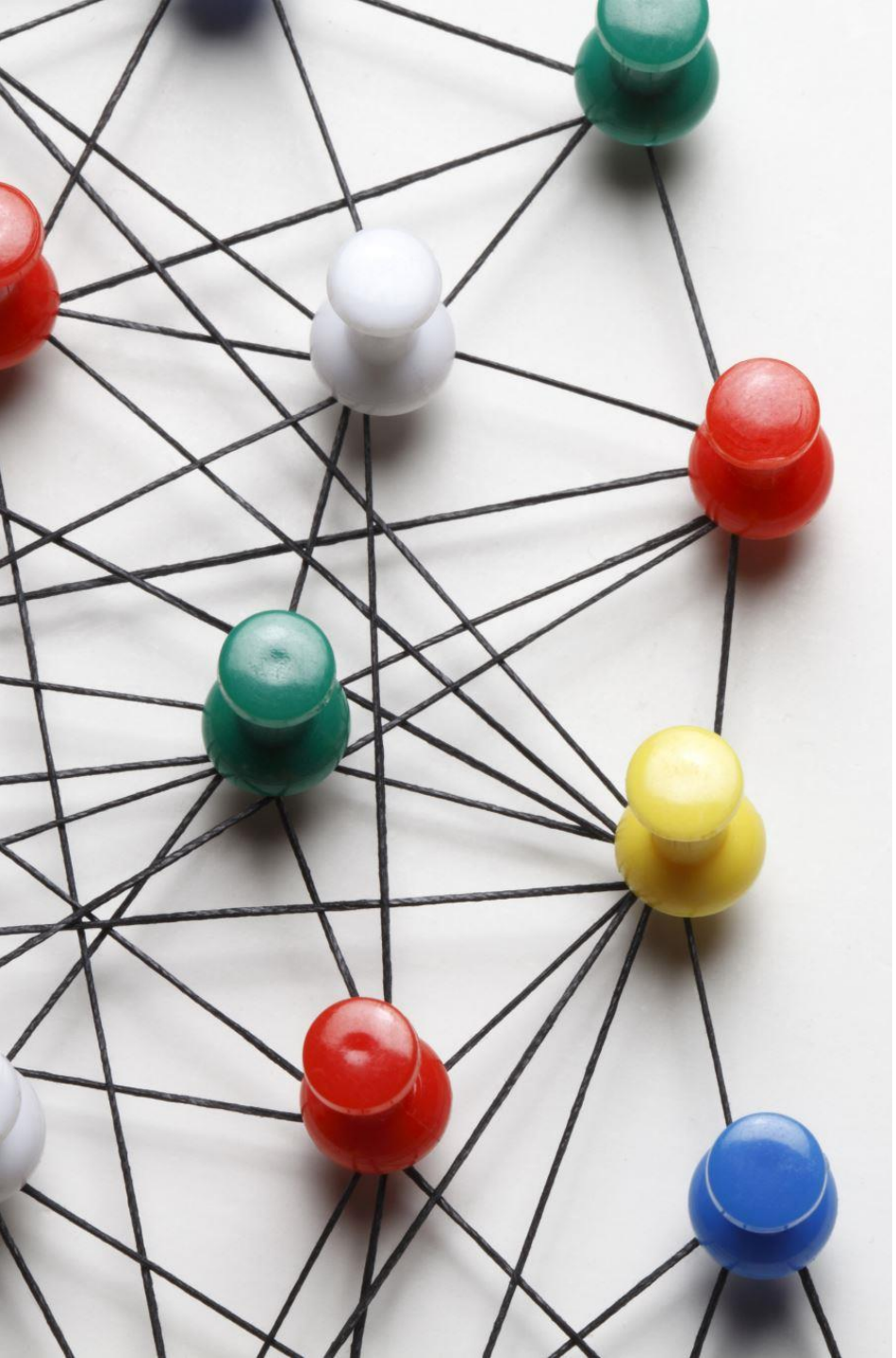
Which of the following BEST describes a network database model?

a) Data is arranged in a simple list

b) Data is arranged in a tree with one parent per child

c) Data is arranged in a graph allowing multiple parents and children

d) Data is stored without relationships



Mini-Quiz: Network Database Model

Which of the following BEST describes a network database model?

a) Data is arranged in a simple list

b) Data is arranged in a tree with one parent per child

c) Data is arranged in a graph allowing multiple parents and children

d) Data is stored without relationships

Mini-Quiz: Network Database Model

In the network model, a child can have more than one parent.
True / False

Mini-Quiz: Network Database Model

In the network model, a child can have more than one parent.

✓ True

✗ False

Mini-Quiz: Network Database Model

The network database model was mainly designed to handle:

a) Flat files

b) Many-to-many relationships

c) Only numerical data

d) Only hierarchical data

Mini-Quiz: Network Database Model

The network database model was mainly designed to handle:

a) Flat files

b) Many-to-many relationships

c) Only numerical data

d) Only hierarchical data

RELATIONAL DATABASE

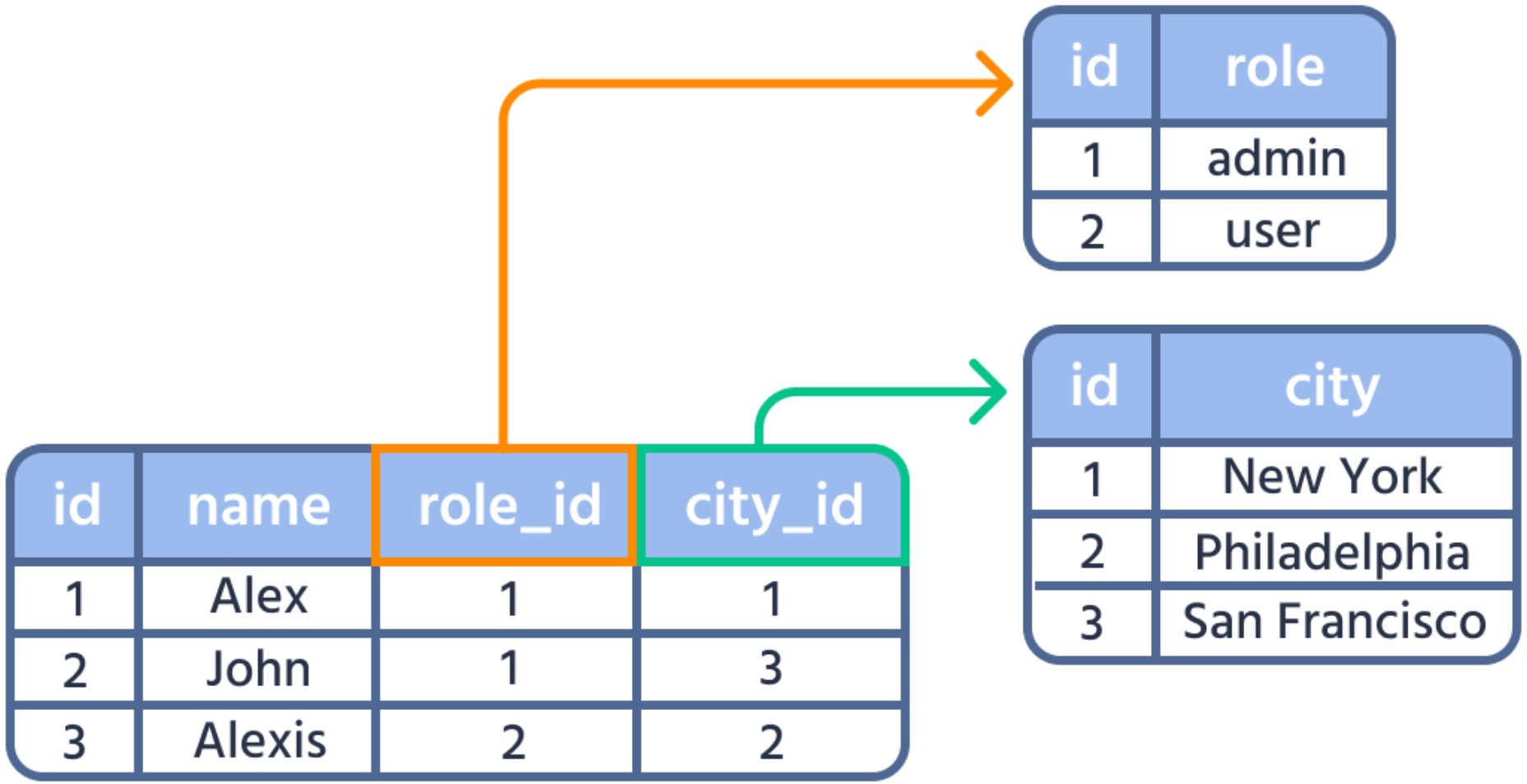
- RDBMS – A database based on the relational model developed by E.F Code.
- Relational database allows :
  - The definition of data structures
  - Storage & Retrieval operations
  - Integrity Constraints
- It uses SQL for storing, manipulating, as well as maintaining the data.

RELATIONAL DATABASE

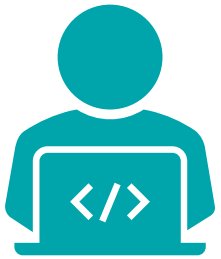
- In such a database, the data & relations between them are organized in tables.
- A **table** is collection of **records**.
- Each record in a table contains the same field.
- Stores data in form of :
  - Rows (tuple)
  - Columns (attributes)
  - Together forms a table (relation)

PROPERTIES OF RELATIONAL TABLES

- Values Are Atomic
- Each Row Is Unique
- Columns Values Are Of The Same Kind
- The Sequence Of Columns Is Insignificant
- The Sequence Of Rows Is Insignificant
- Each Column Has A Unique Name

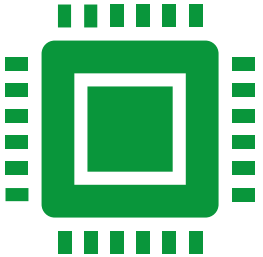

id
role
1
admin
2
user
id
city
1
New York
2
Philadelphia
3
San Francisco
id
name
role_id
city_id
1
Alex
1
1
2
John
1
3
3
Alexis
2
2

RELATIONAL DATABASE – Real Industry Use Cases



MySQL

Powers **web applications, CMS systems, and e-commerce platforms** (e.g., WordPress, Magento).

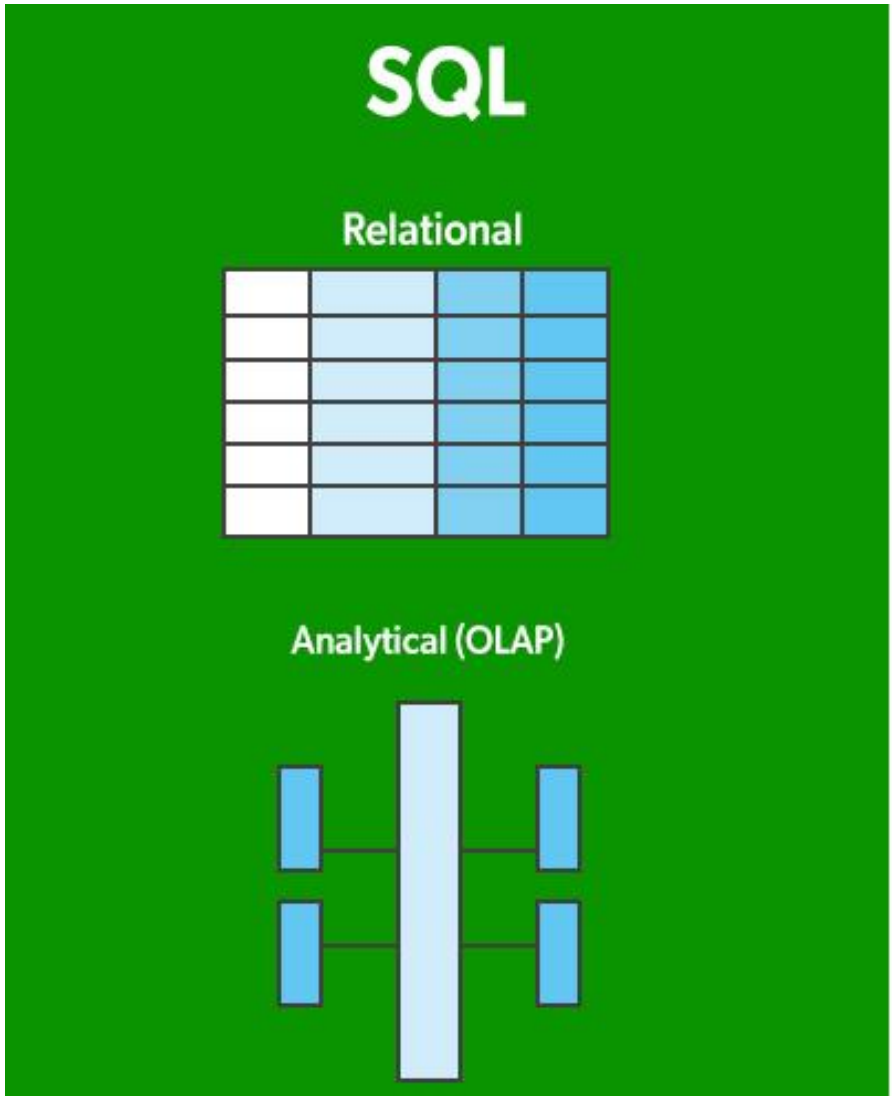


Oracle Database

Used in **banking, telecom, ERP systems**, handling millions of secure financial transactions daily.

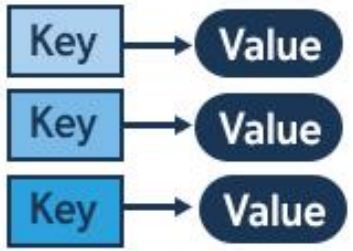
NoSQL DATABASE

- Used for storing a wide range of datasets.
- It came into existence when the demand for building modern applications increased.
- It presented wide variety of database technologies in response to the demands.

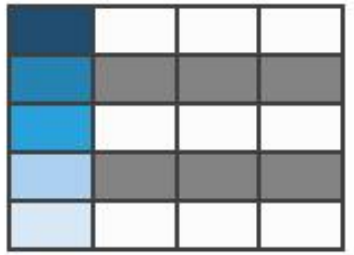


NoSQL

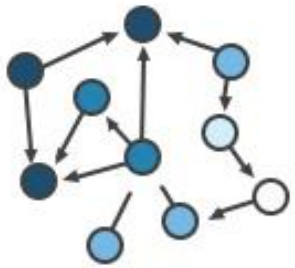
Key-Value



Column-Family



Graph



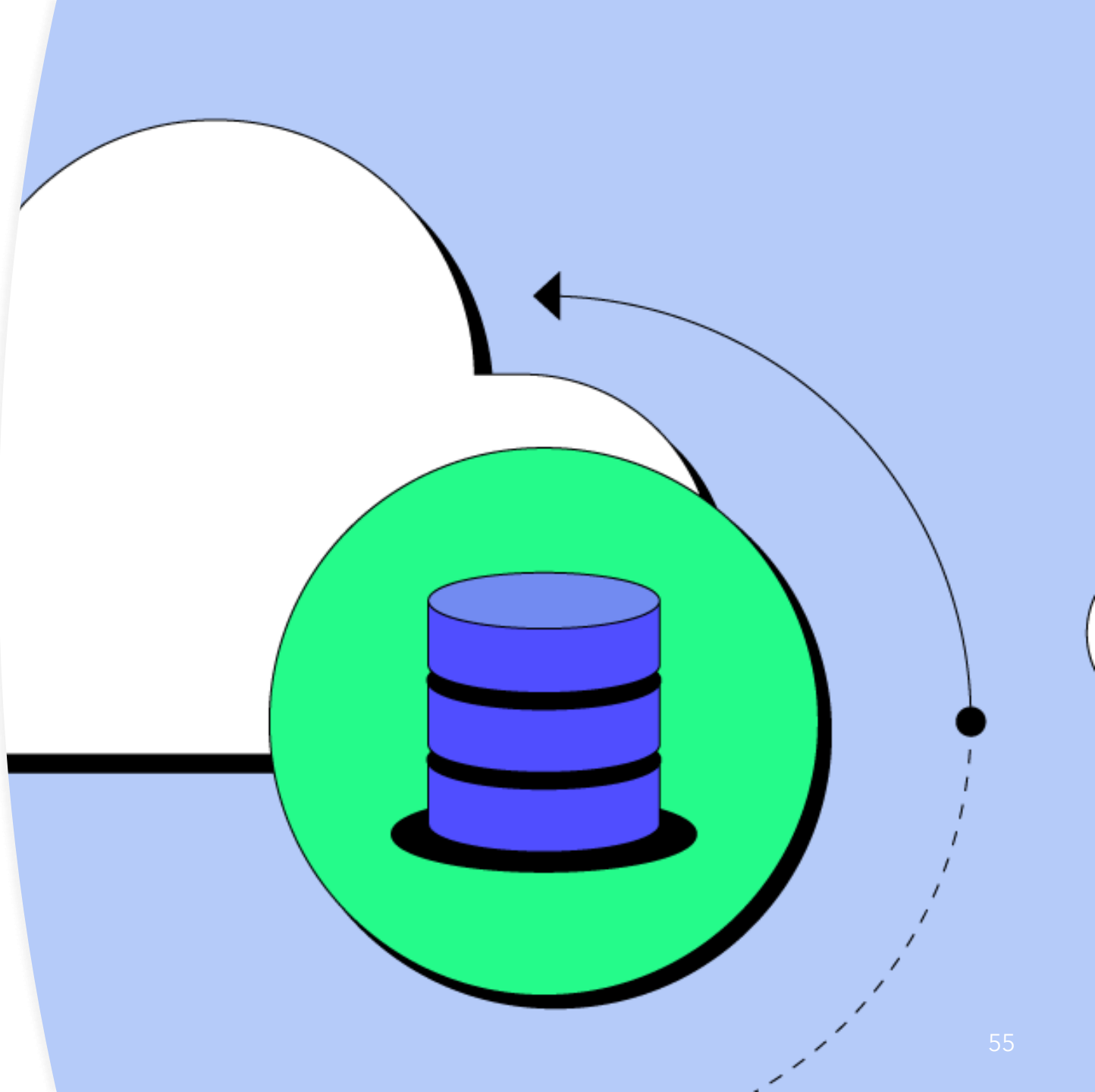
Document

NoSQL DATABASE – Real Industry Use Cases

- **Document Stores (MongoDB)**
  - **eBay and Forbes** use MongoDB to handle millions of flexible documents (products, articles).
- **Key-Value Stores (Redis)**
  - **Snapchat** uses Redis for session management to achieve high-speed performance at scale.
- **Column-Family Stores (Cassandra)**
  - Used in **finance, telecom, and e-commerce** for real-time, fault-tolerant, high-throughput applications.
- **Time-Series NoSQL (InfluxDB)**
  - **IoT & financial analytics** companies use InfluxDB for monitoring millions of events per second

CLOUD DATABASE

- Data is stored in a virtual environment and executes over the cloud computing platform.
- It provides users with various cloud computing services (SAAS, PAAS, IAAS, etc.) for accessing the database.



CLOUD DATABASE – Real Industry Use Cases

Personal cloud storage services, such as **Dropbox, Google Drive, iCloud**, store user files using cloud-based database infrastructures.

Online streaming, e-commerce, healthcare, finance, and **aviation** rely on cloud databases to manage massive, scalable real-time data workloads.

Familiarizing yourself with cloud technology not only equips you with practical skills but also future-proofs your career trajectory given the transformative trends in the tech industry. Your ability to showcase cloud skills will bolster your portfolio and strengthen your candidacy for a range of software development roles.

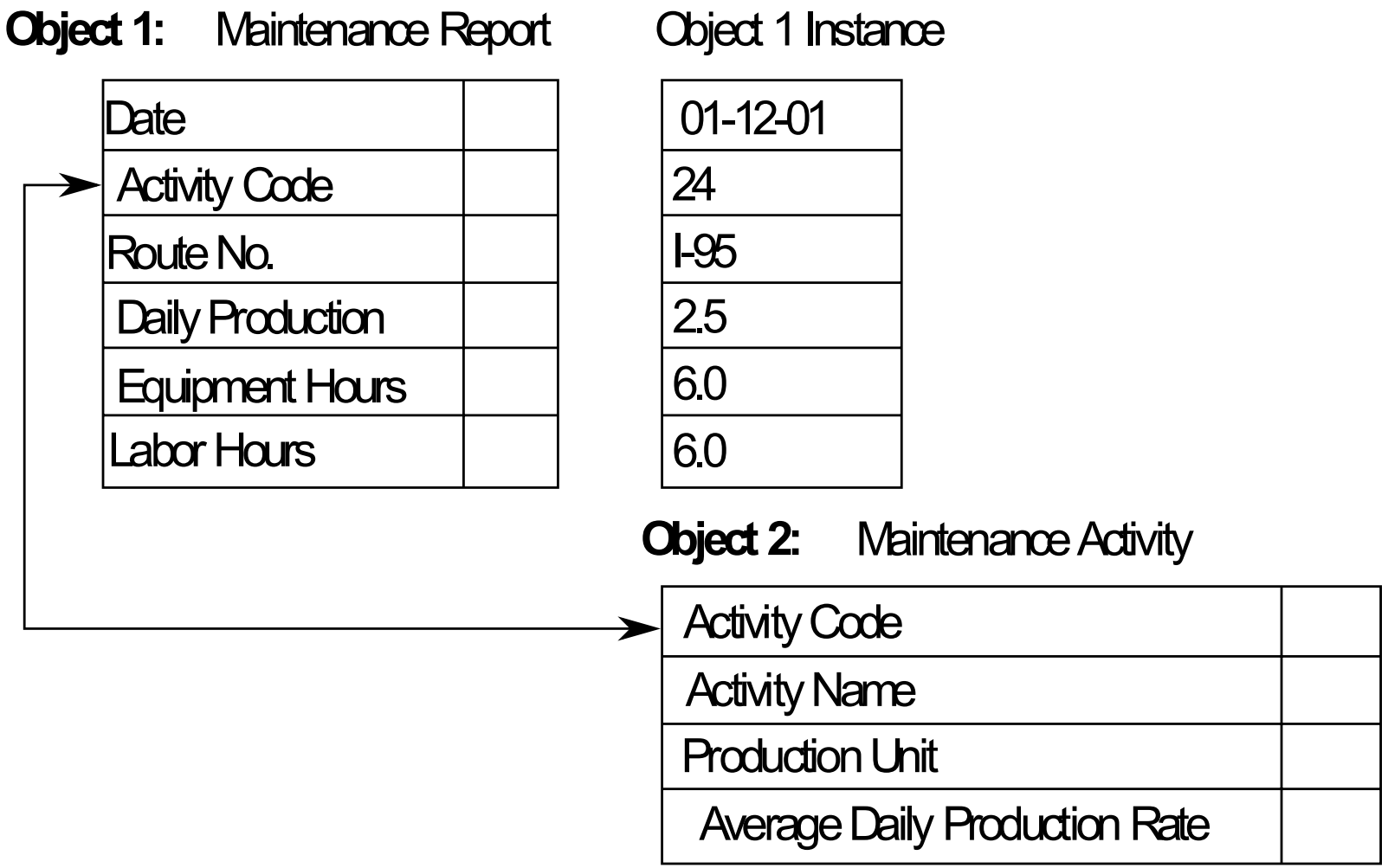
OBJECT-ORIENTED MODEL

- Object DBMSs add database functionality to object programming languages.
- They bring much more than persistent storage of programming language objects.
- A major benefit of this approach is the unification of the application and database development into a seamless data model and language environment.

OBJECT-ORIENTED MODEL

- It uses the object-based data model approach for sorting data in the database system.
- The data is represented and stored as objects which are like the objects used in the object-oriented programming language.

Object-Oriented Model



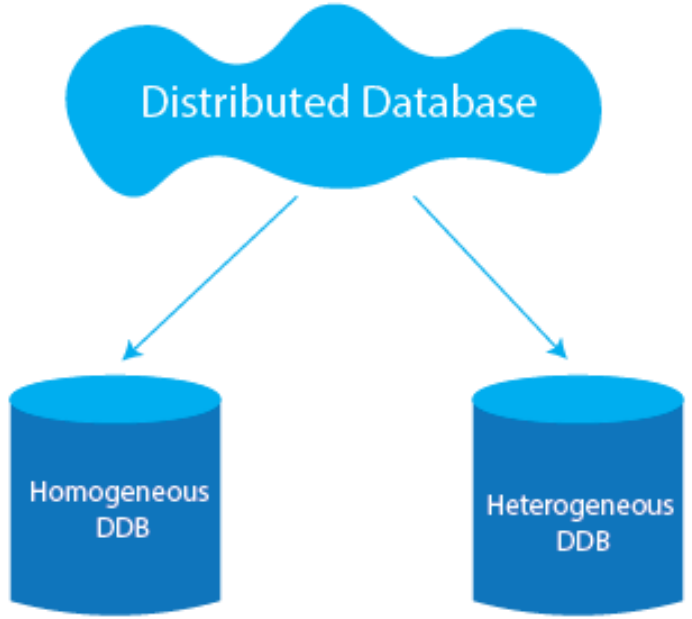
OBJECT-ORIENTED DATABASE – Real Industry Use Cases

Engineering design (CAD/CAM) and **geospatial systems**, where complex objects must be stored and manipulated.

Scientific simulations, molecular biology, embedded systems, and **telecommunications** where object identity and complex relationships are crucial.

DISTRIBUTED DATABASE

- Data is distributed among different database systems of an organization. These database systems are connected via communication links.
- One server failure will not affect the entire data set.
- **Homogeneous DDB:** database systems which execute on the same OS & use the same application process and carry the same hardware devices.
- **Heterogeneous DDB:** database systems which execute on different OS under different application procedures and carries different hardware devices.



DISTRIBUTED DATABASE – Real Industry Use Cases

E-commerce platforms using **distributed key-value stores like Amazon DynamoDB** to support massive scale and low-latency inventory/session workloads.

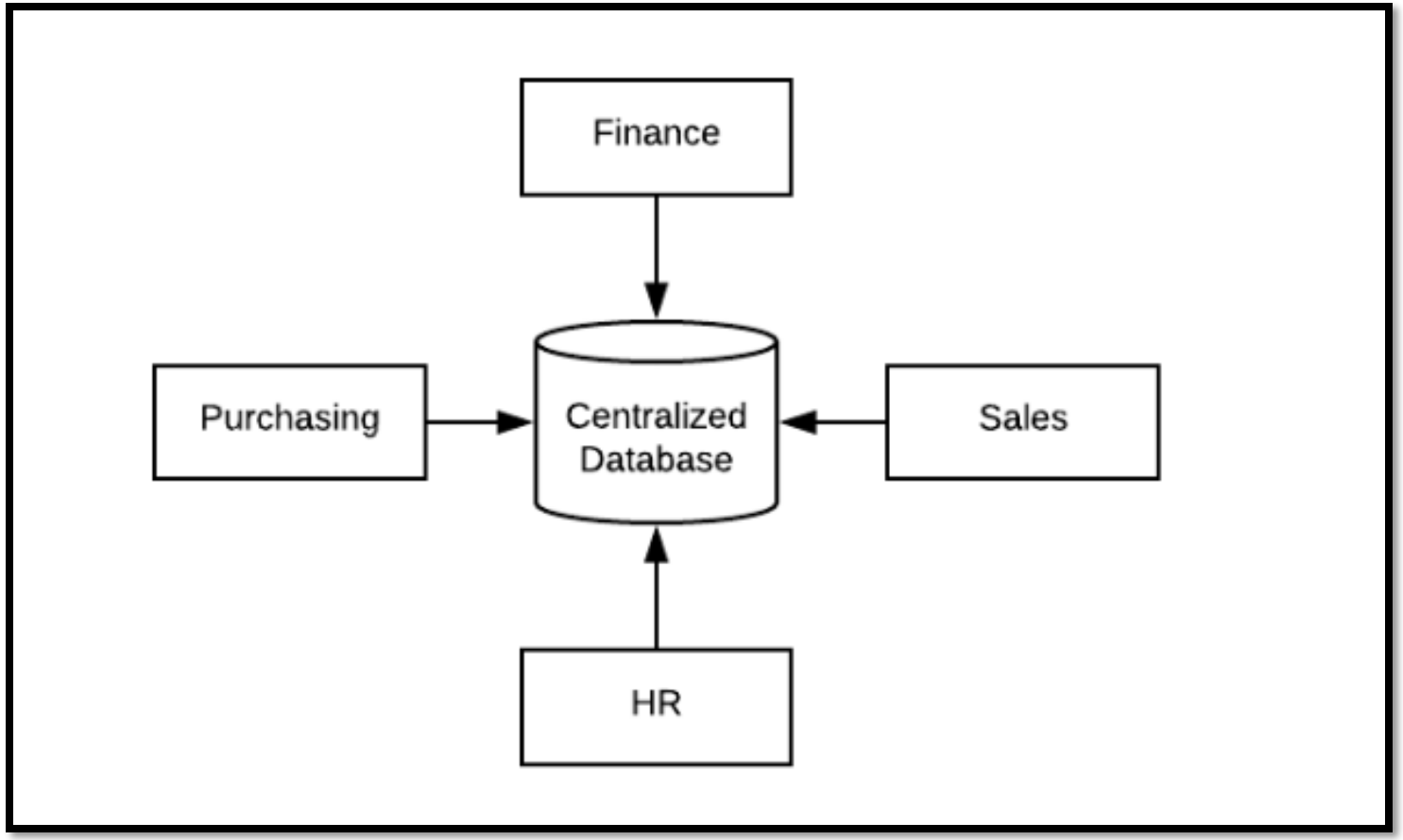
Gaming companies using DynamoDB for real-time, high-velocity read/write operations.

Social media platforms using graph-based distributed systems such as Neo4j for analyzing user relationships.

CENTRALIZED DATABASE

- Users can access the stored data from different locations through several applications.

- These applications contain the authentication process to let users access data securely.



CENTRALIZED DATABASE

Advantages:

1. Data consistency is maintained as it manages data in a central repository.
2. It provides better data quality, which enables organizations to establish data standards.

CENTRALIZED DATABASE

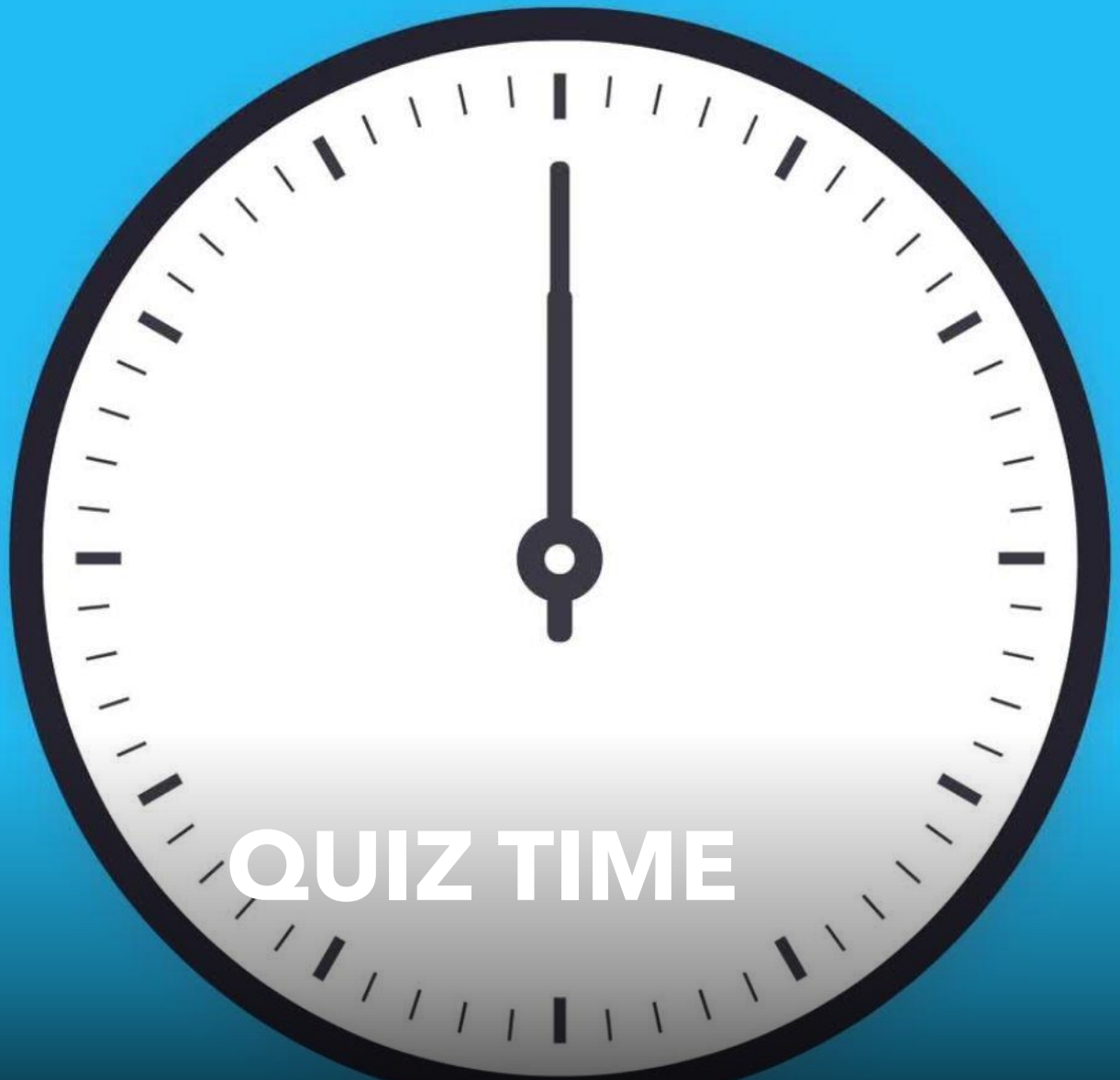
Disadvantages:

1. The size of the centralized database is large, which increases the response time for fetching the data.
2. It is not easy to update such an extensive database system.
3. If any server failure occurs, entire data will be lost, which could be a huge loss.

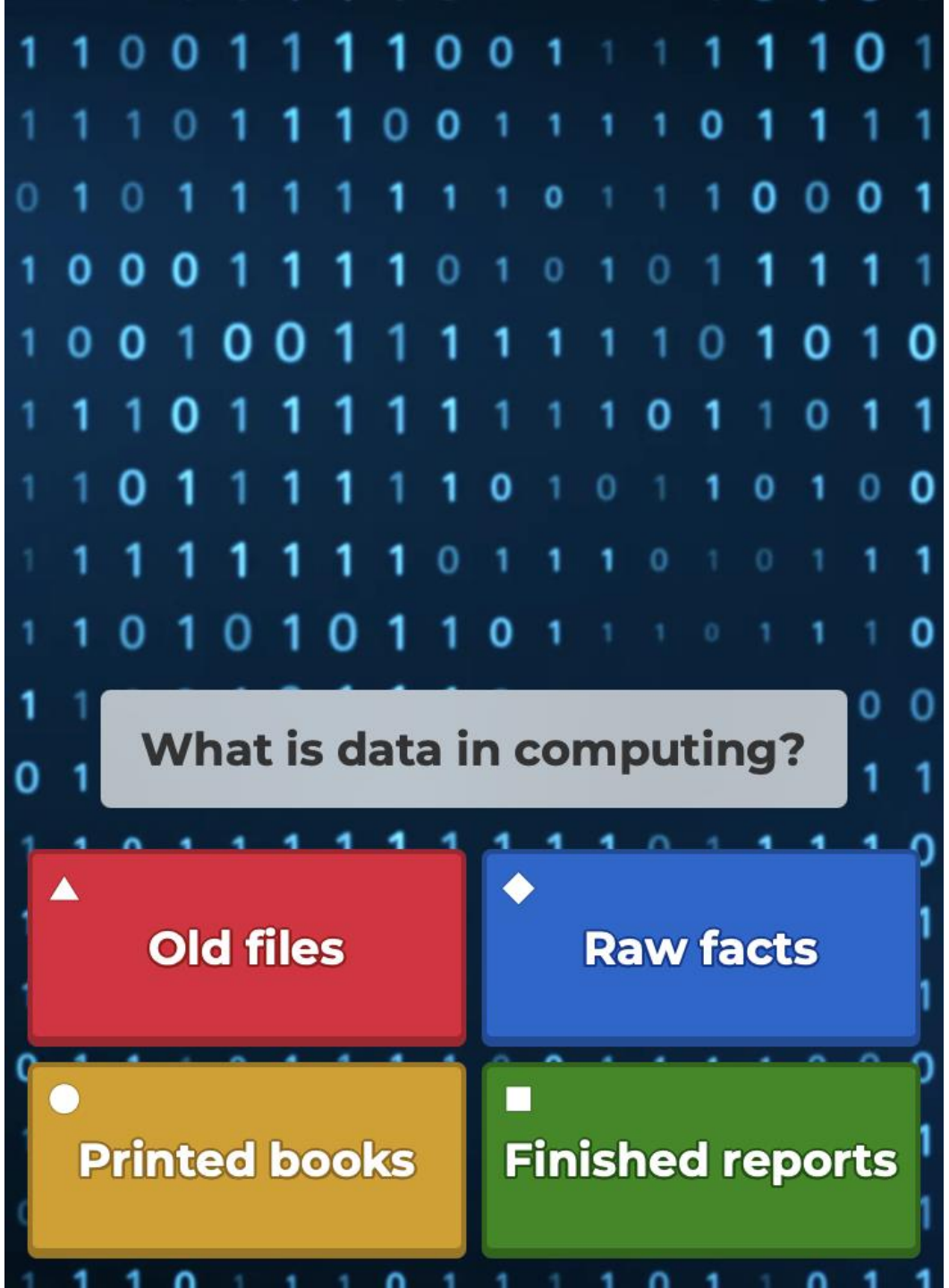
CENTRALIZED DATABASE — Real Industry Use Cases

Enterprises centralizing CRM, ERP, marketing, finance data to eliminate data silos and improve decision-making.

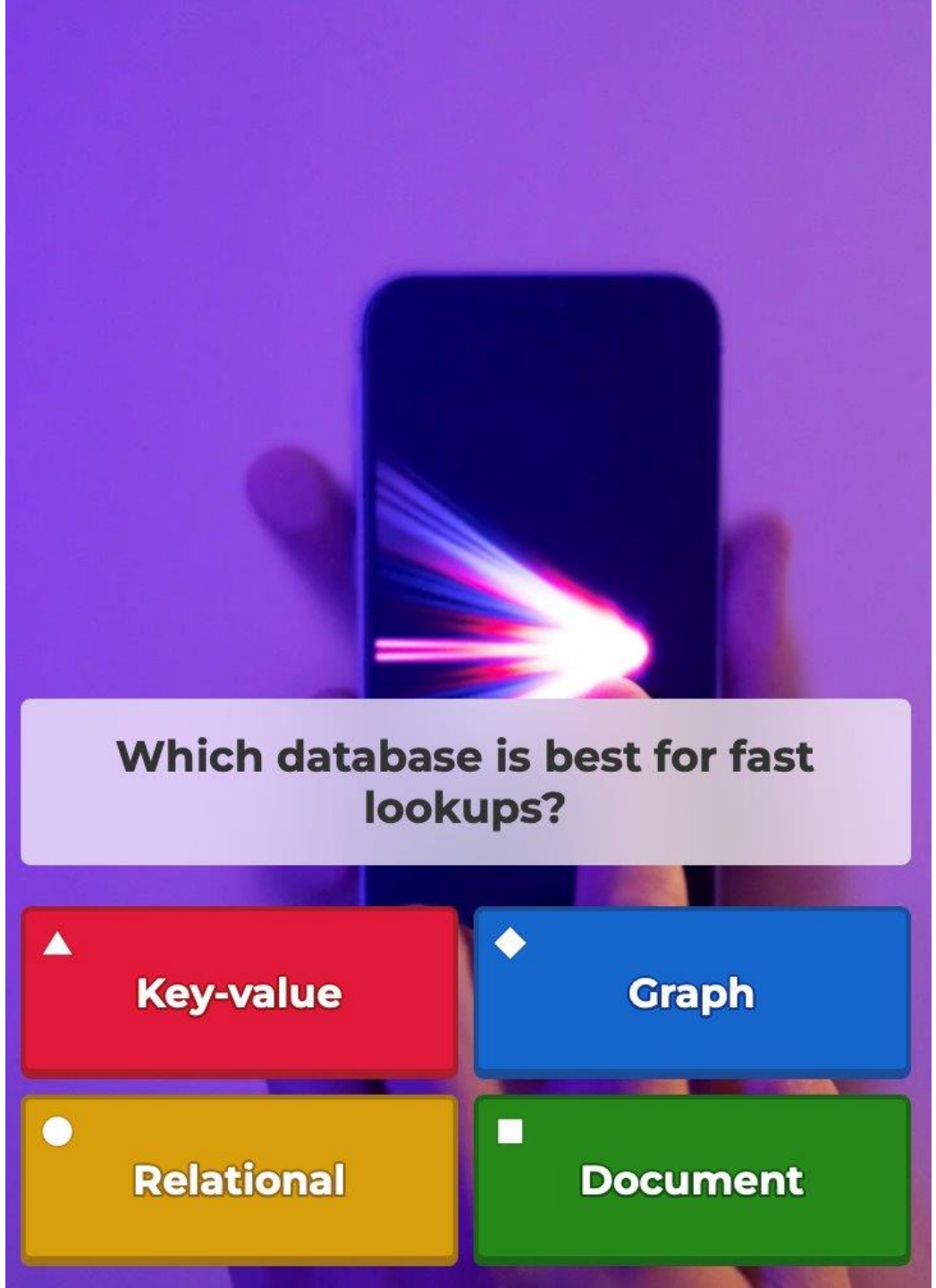
Organizations using centralized data warehouses (e.g., service, cleaning, and security industries) to unify multi-branch operations.

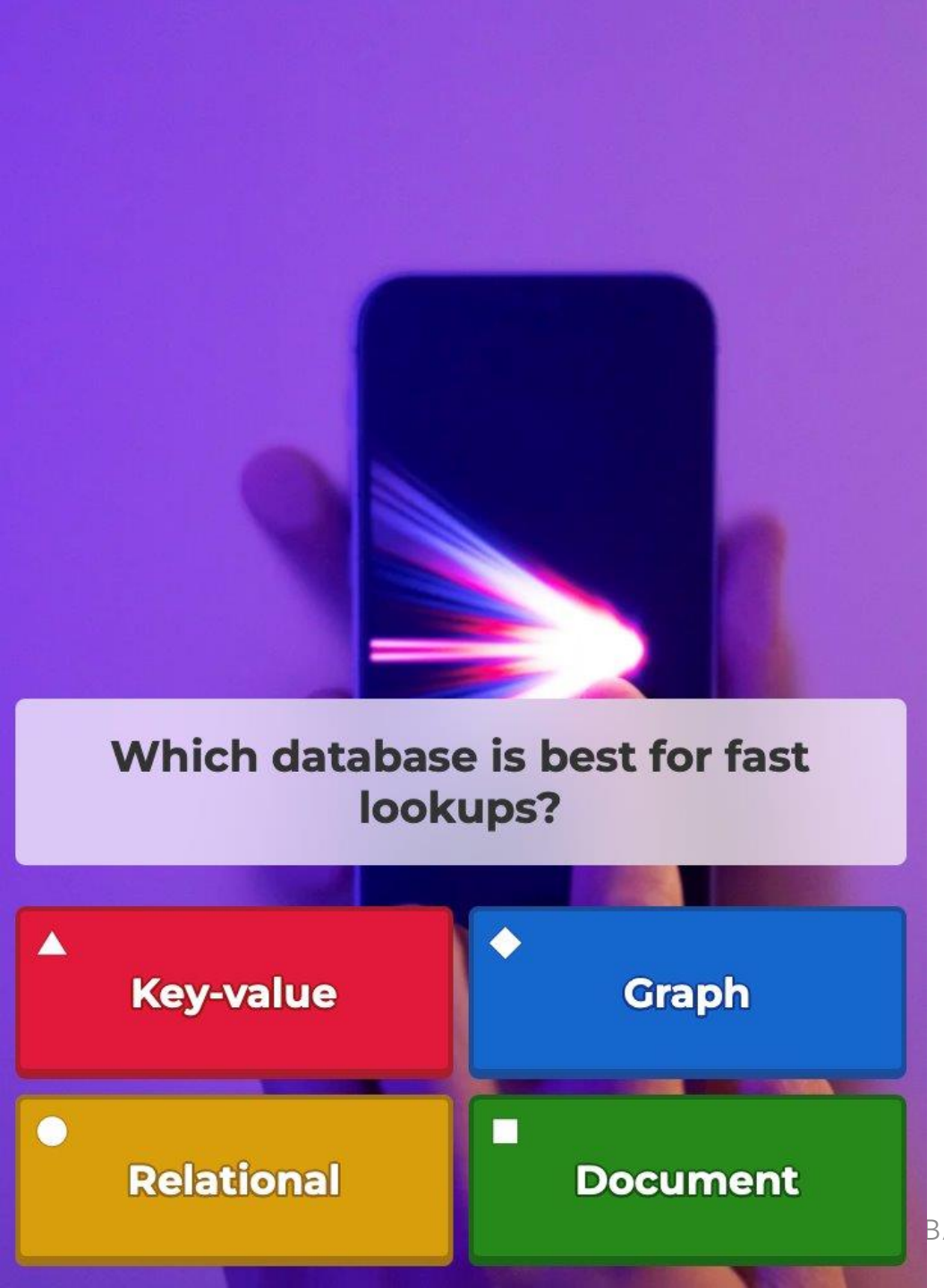


QUIZ TIME

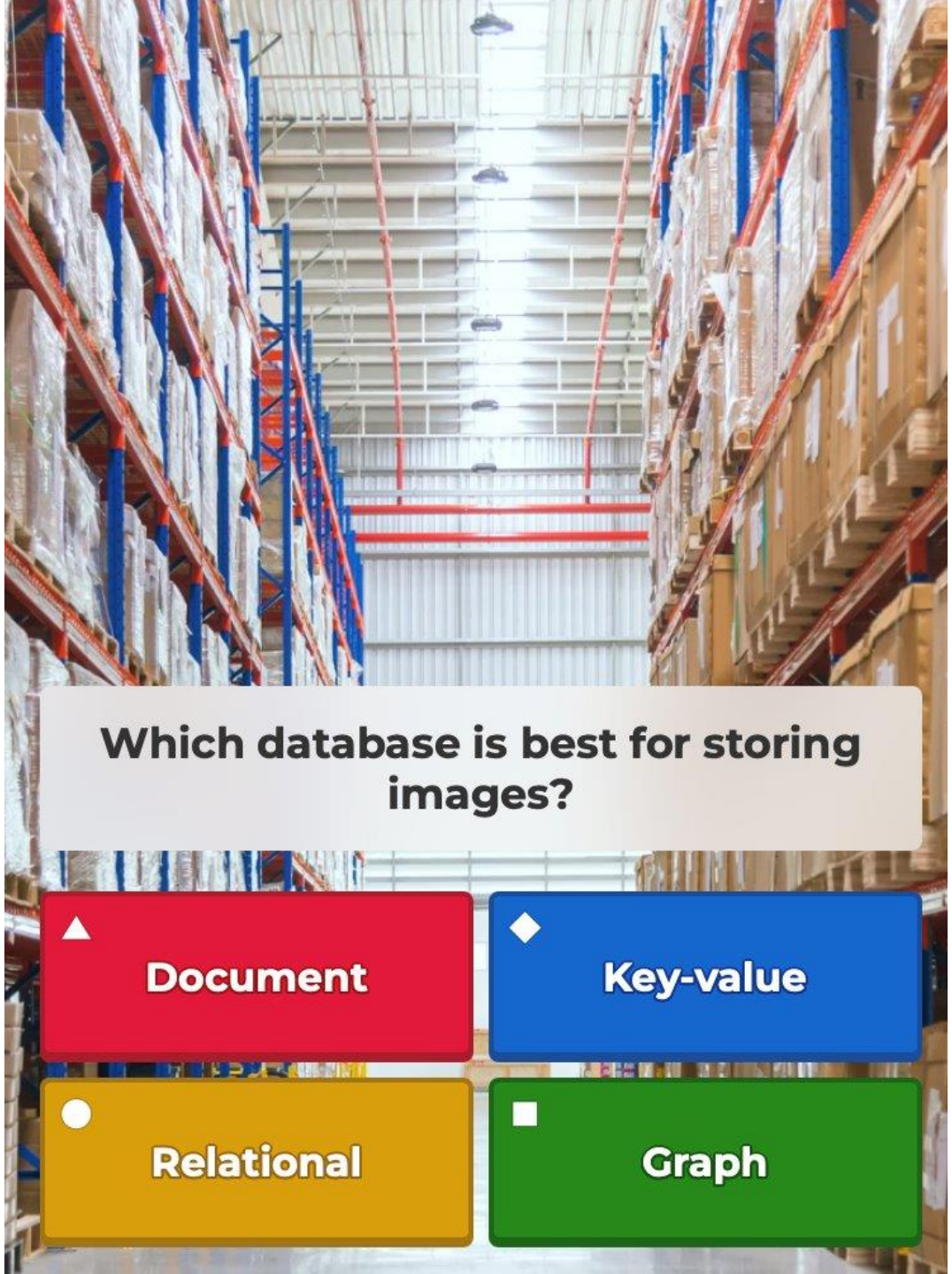

What is data in computing?
Old files
Raw facts
Printed books
Finished reports

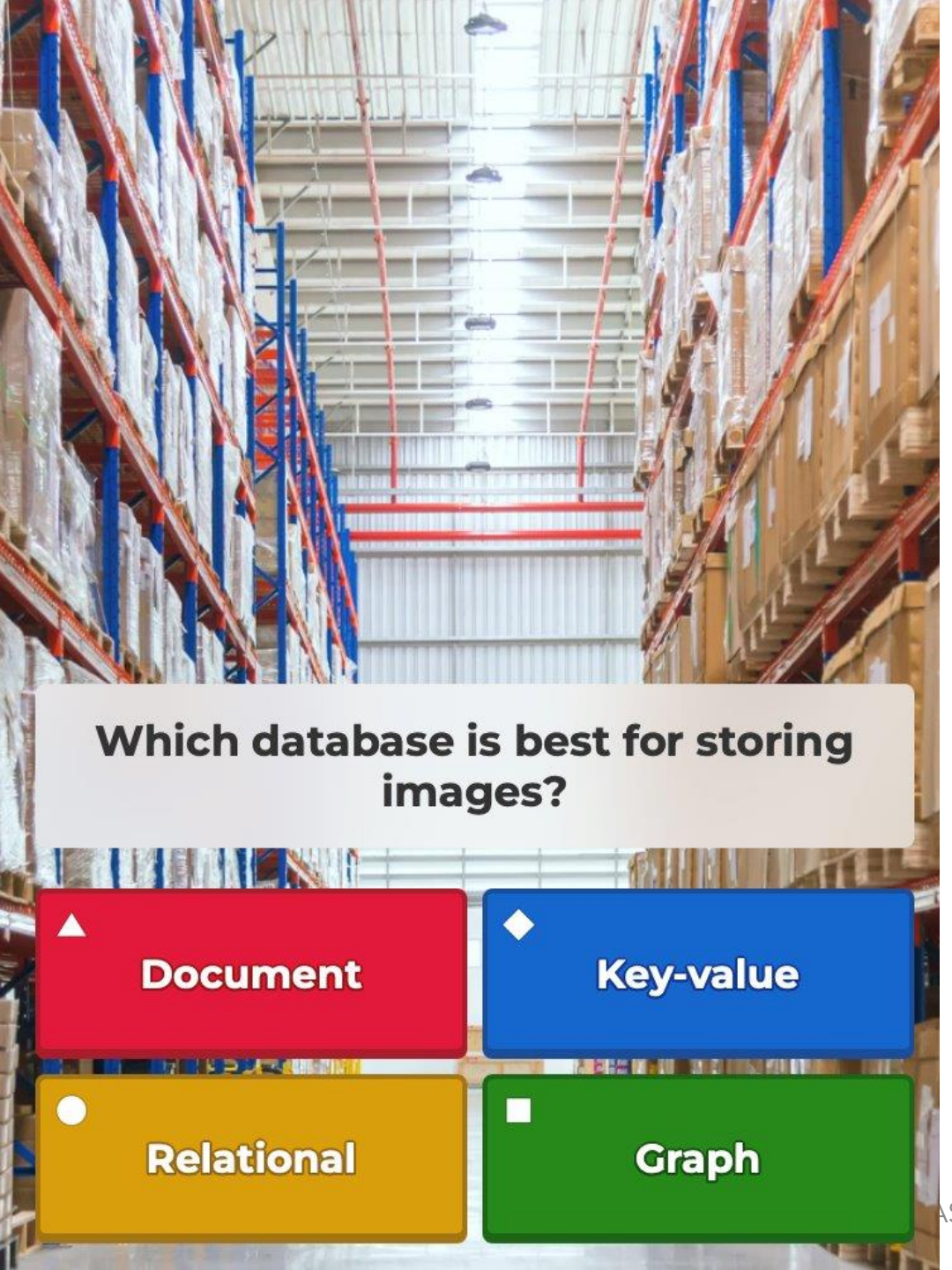
Data in computing is **RAW FACTS**


Which database is best for fast lookups?
Key-value
Graph
Relational
Document

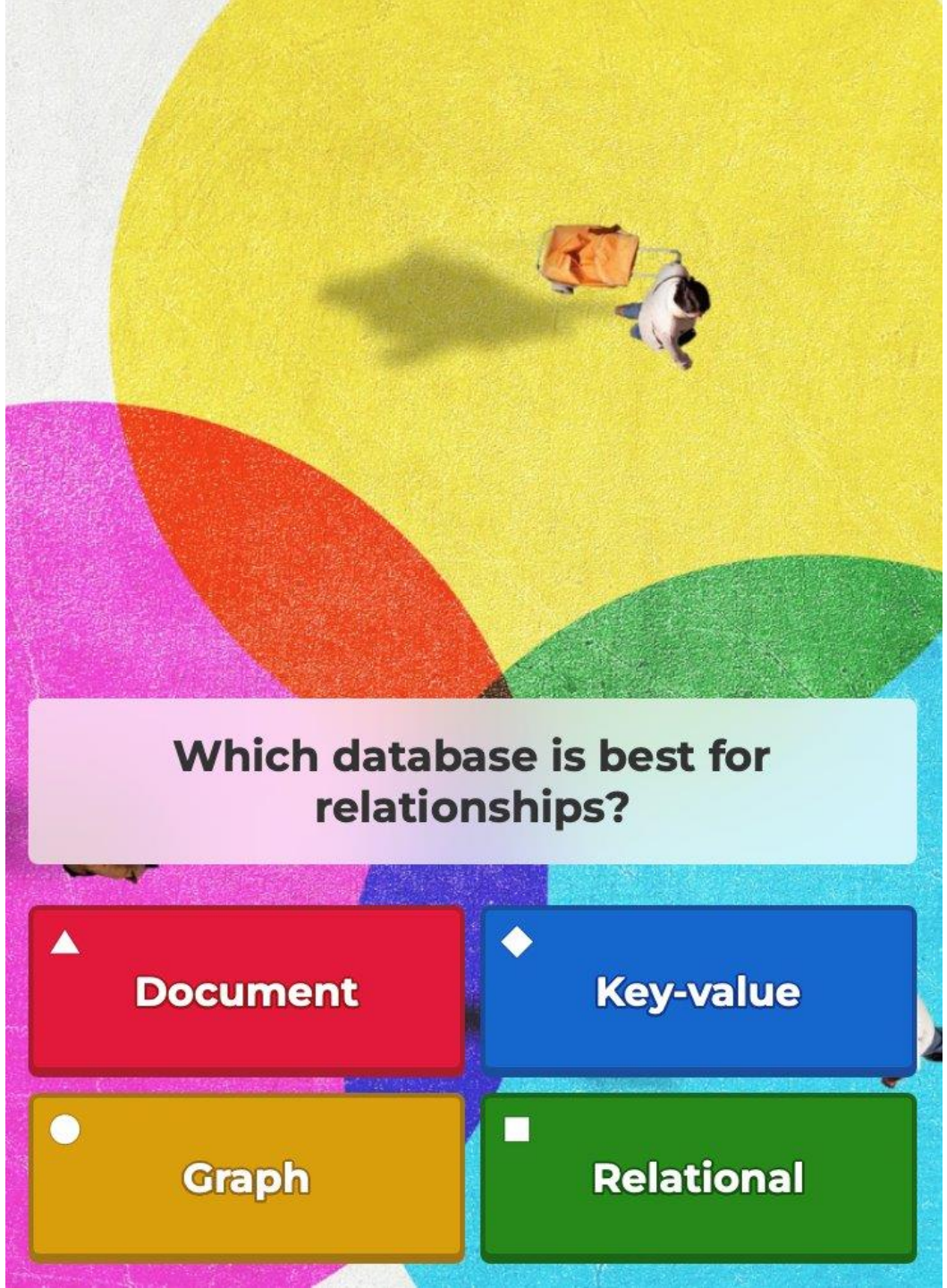


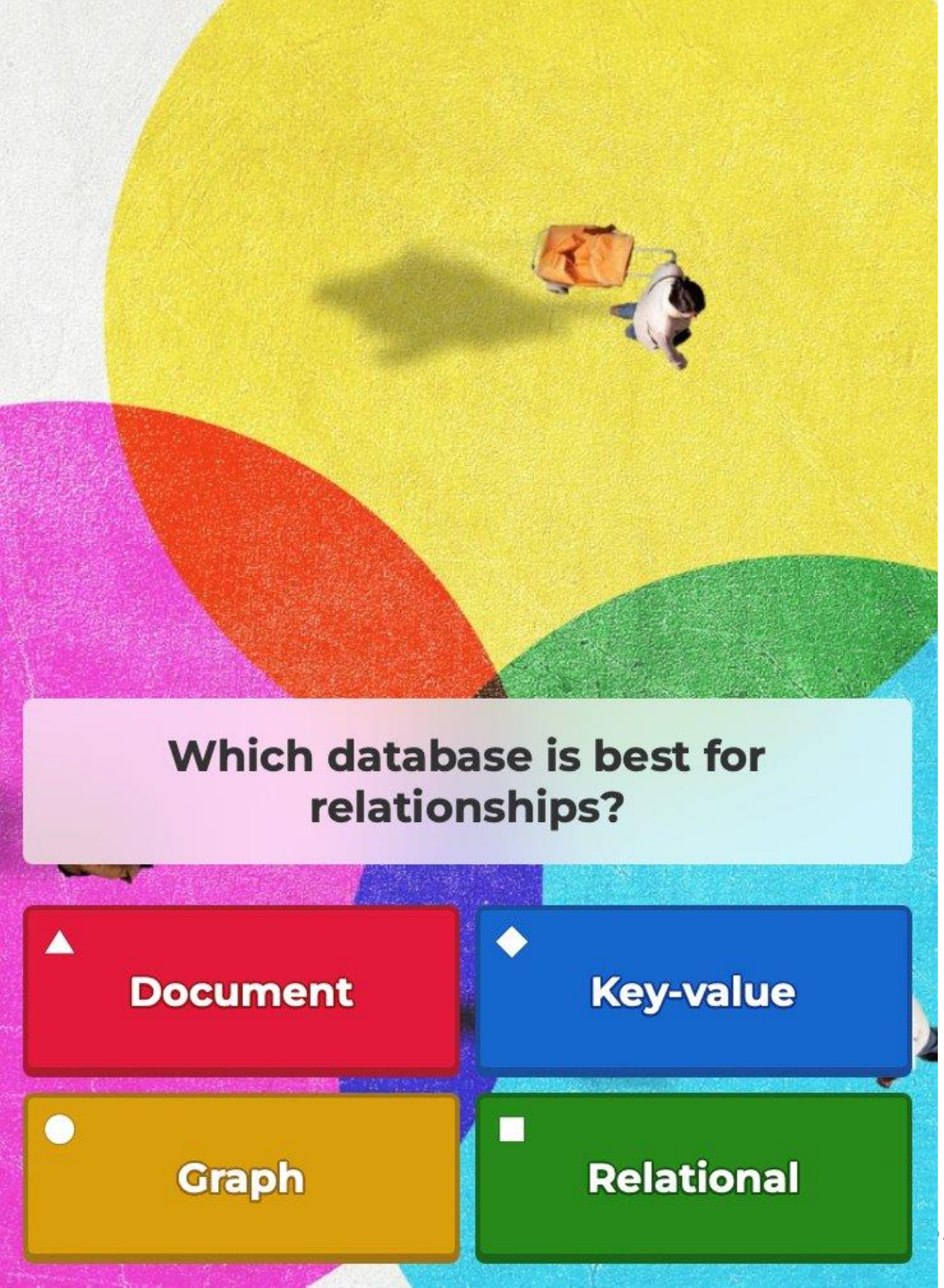
KEY-VALUE is best for lookup


Which database is best for storing images?
Document
Key-value
Relational
Graph



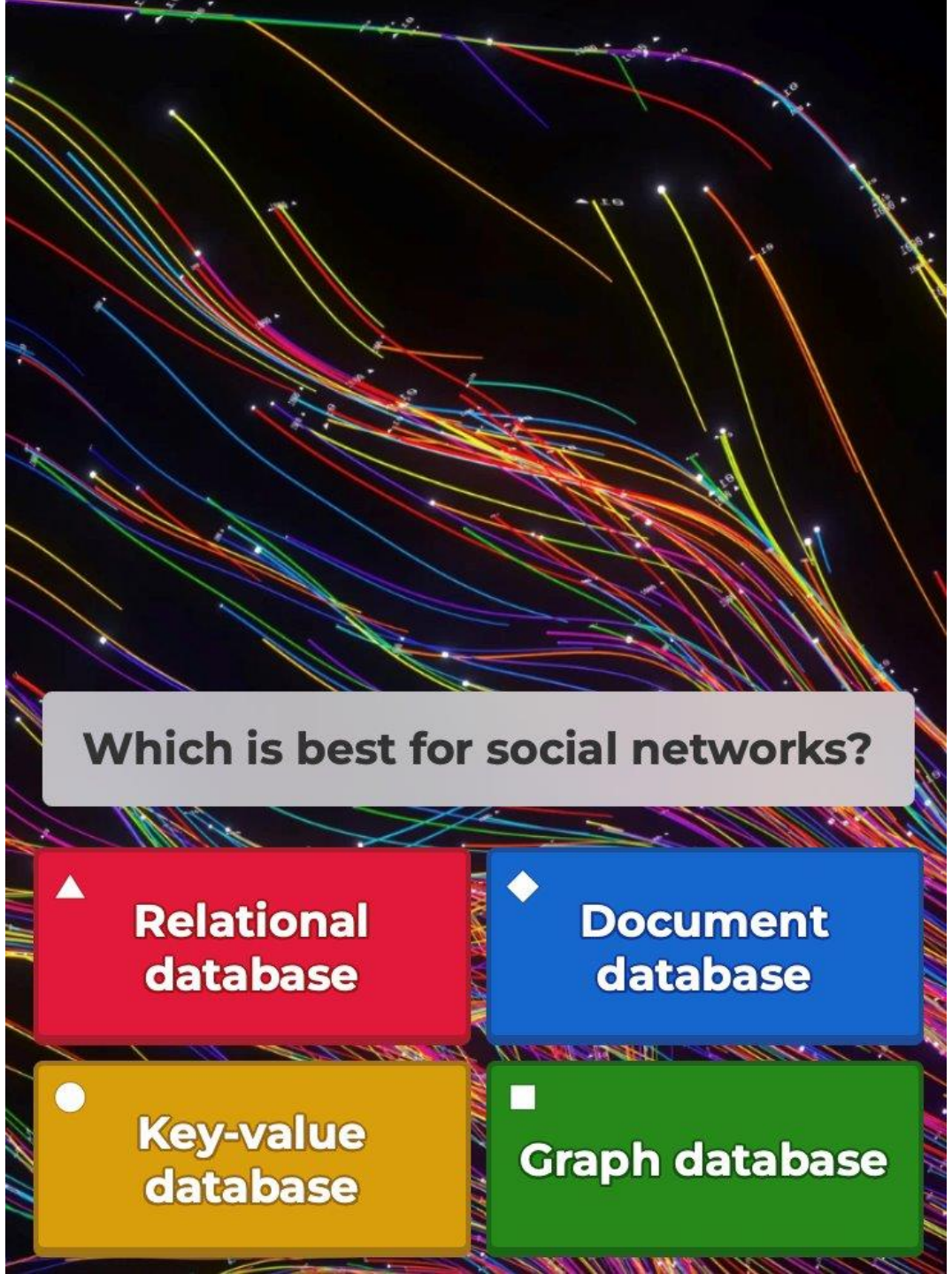
DOCUMENT is best for storing images

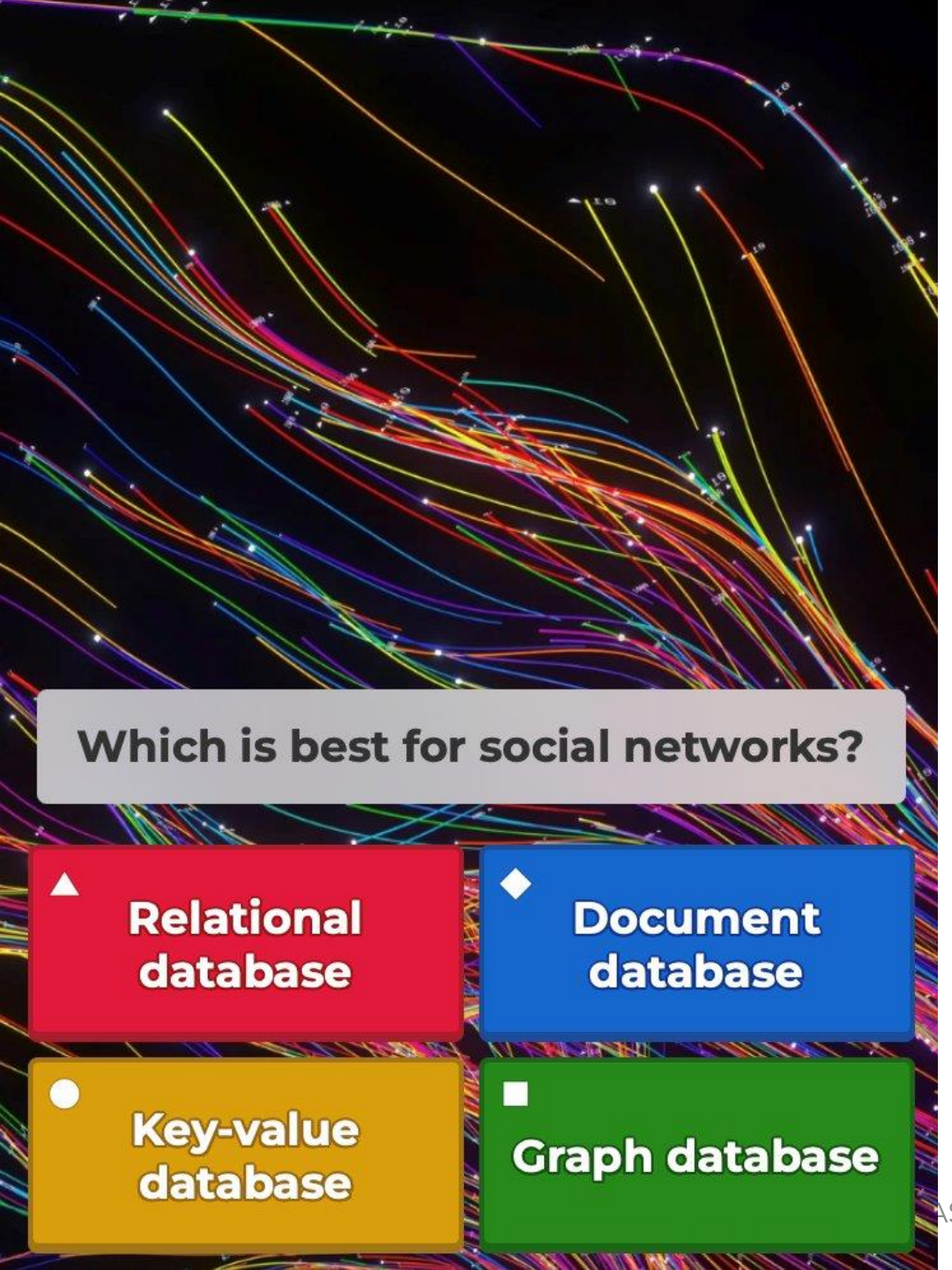

Which database is best for relationships?
Document
Key-value
Graph
Relational



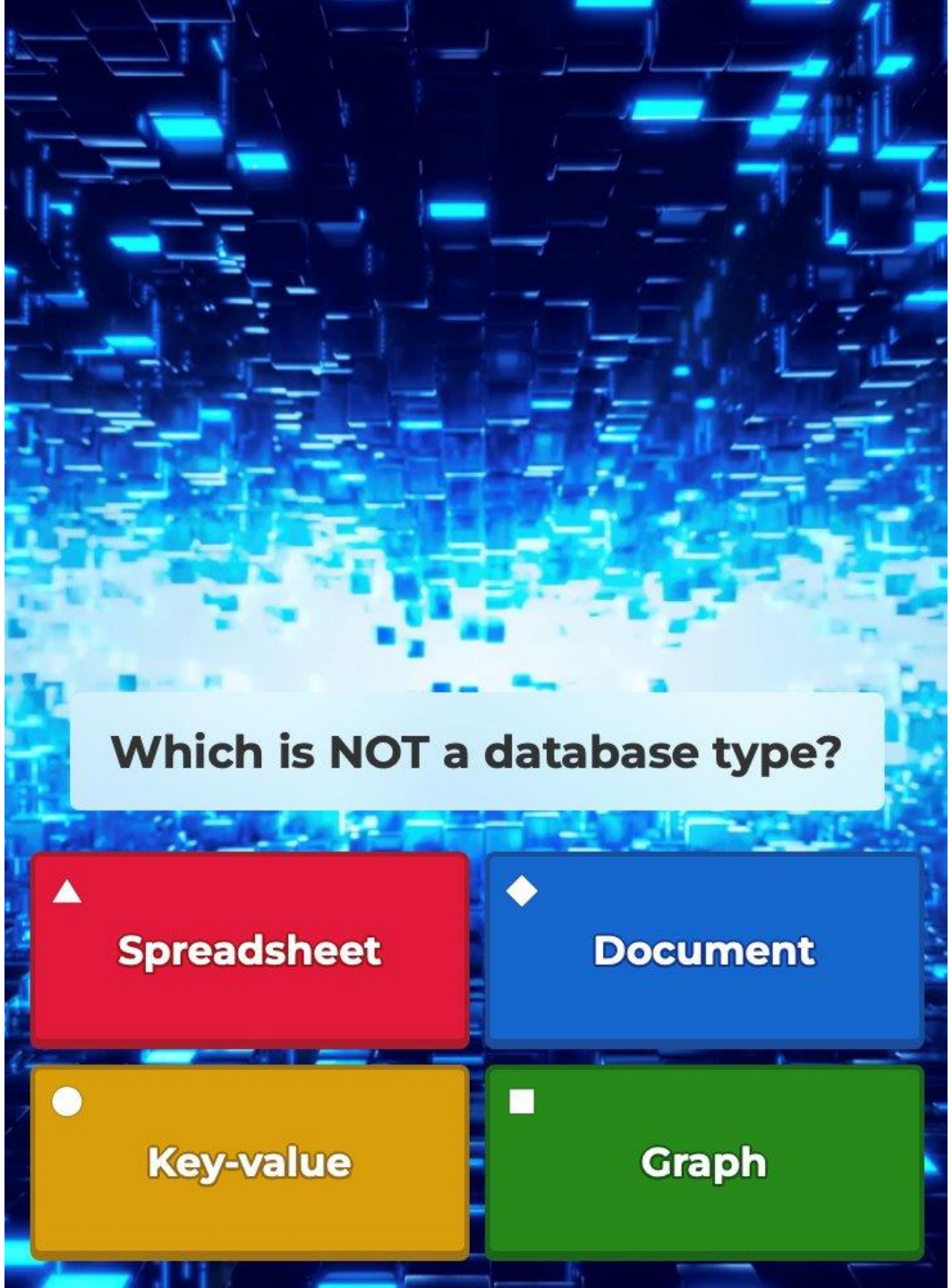
RELATIONAL

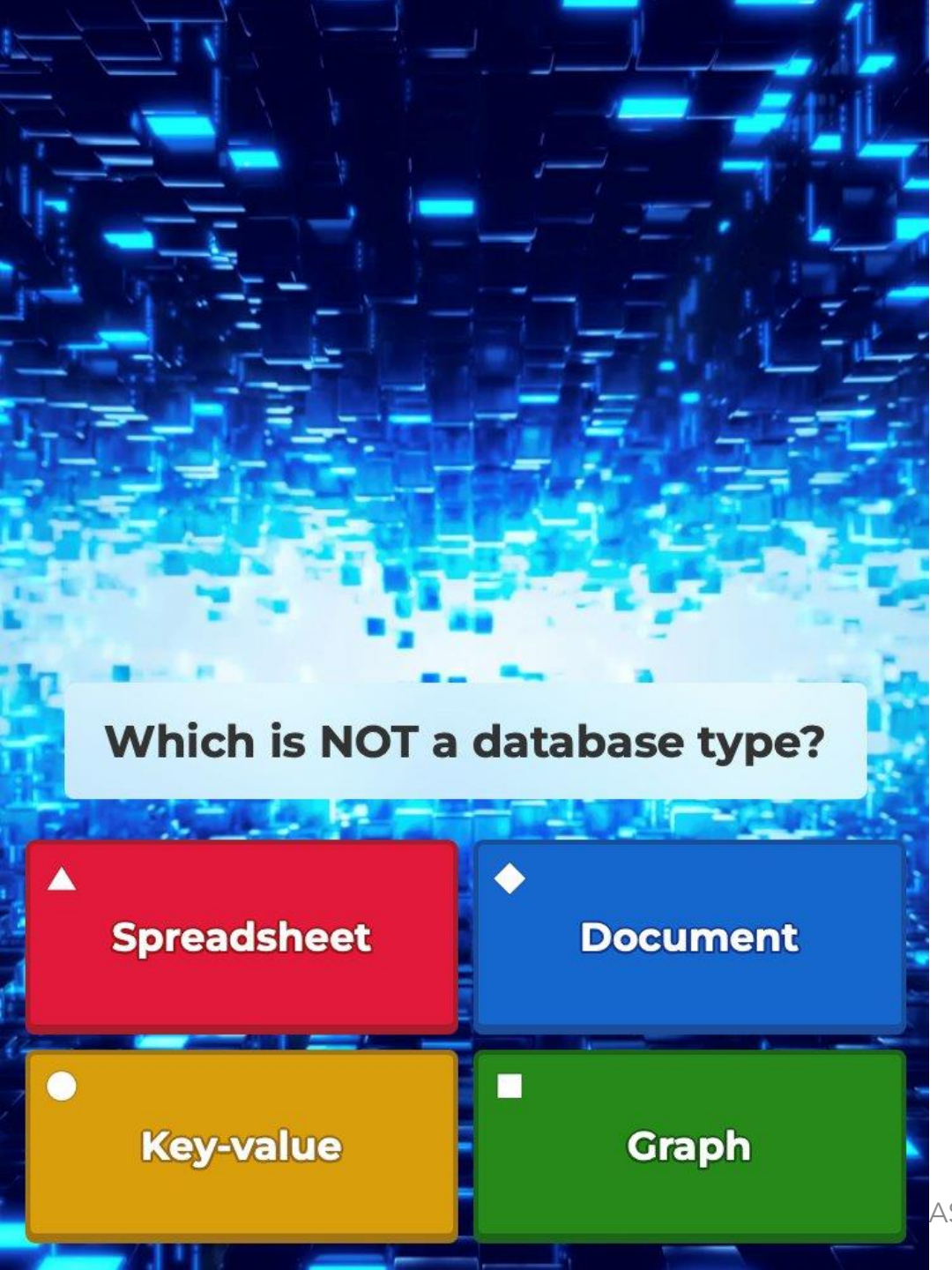
is best
for storing
relationships


Which is best for social networks?
Relational database
Document database
Key-value database
Graph database



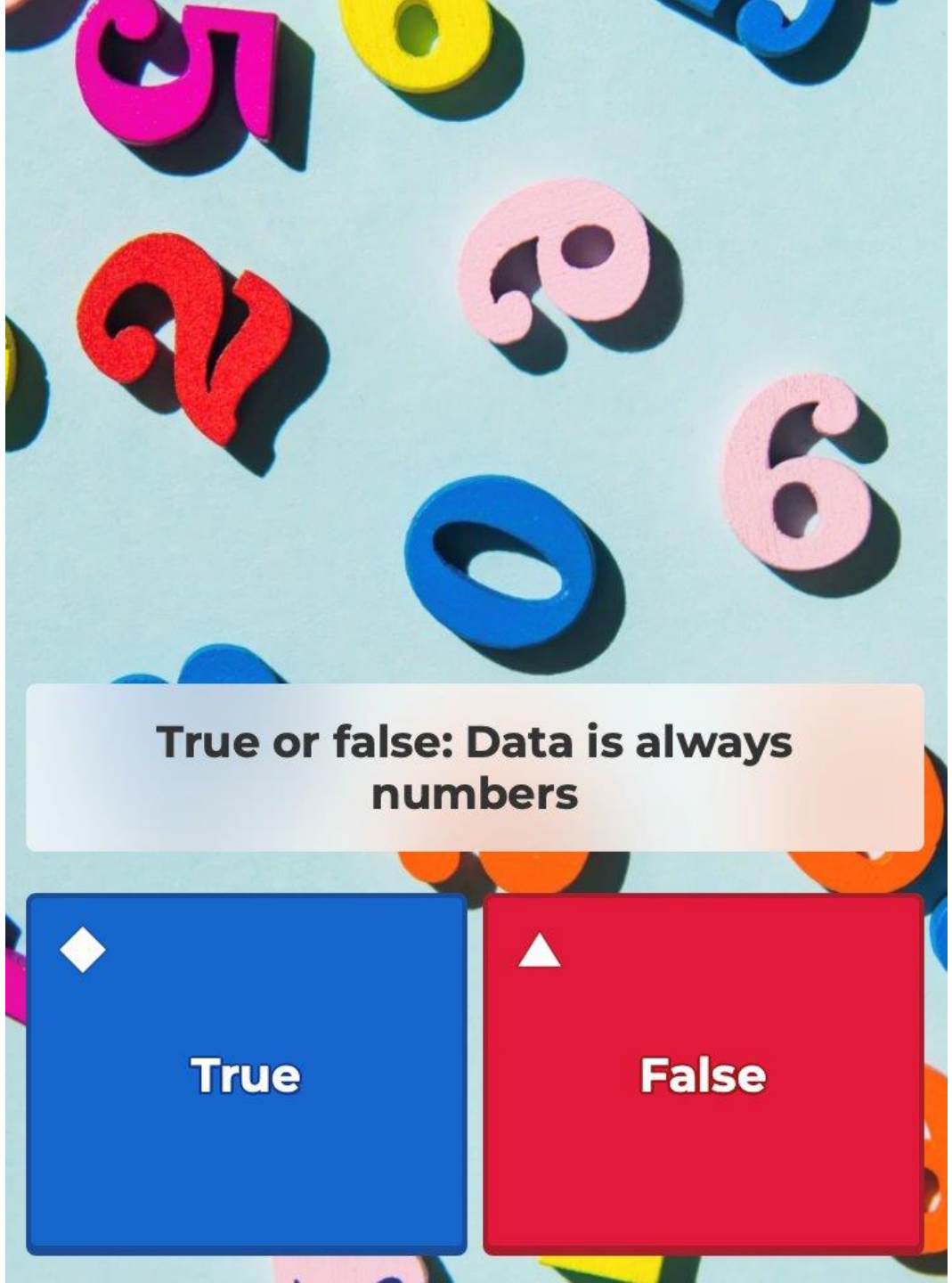
GRAPH DB is best for social networks

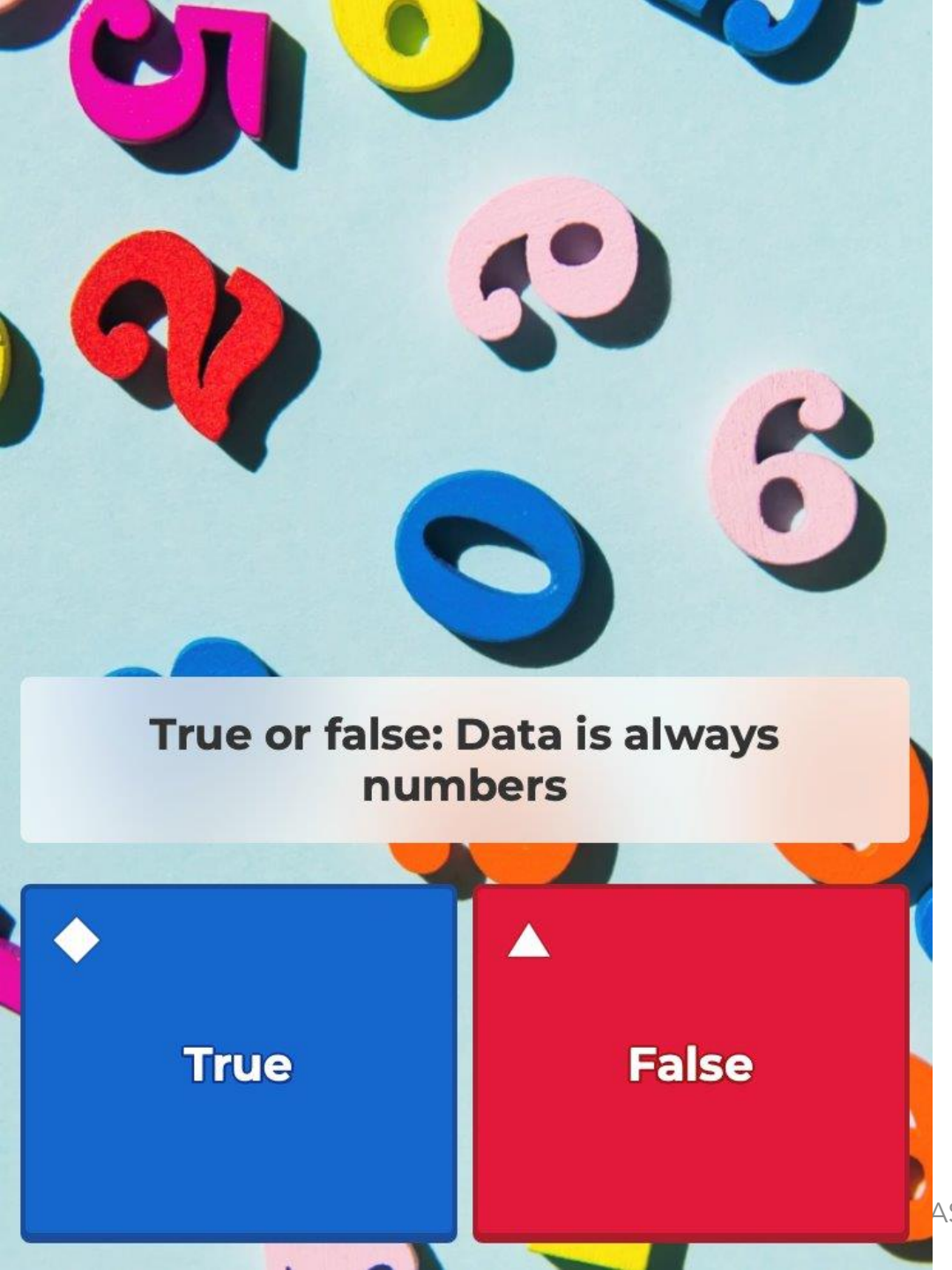

Which is NOT a database type?
Spreadsheet
Document
Key-value
Graph



SPREADSHEET

is not a database type


True or false: Data is always numbers
True
False



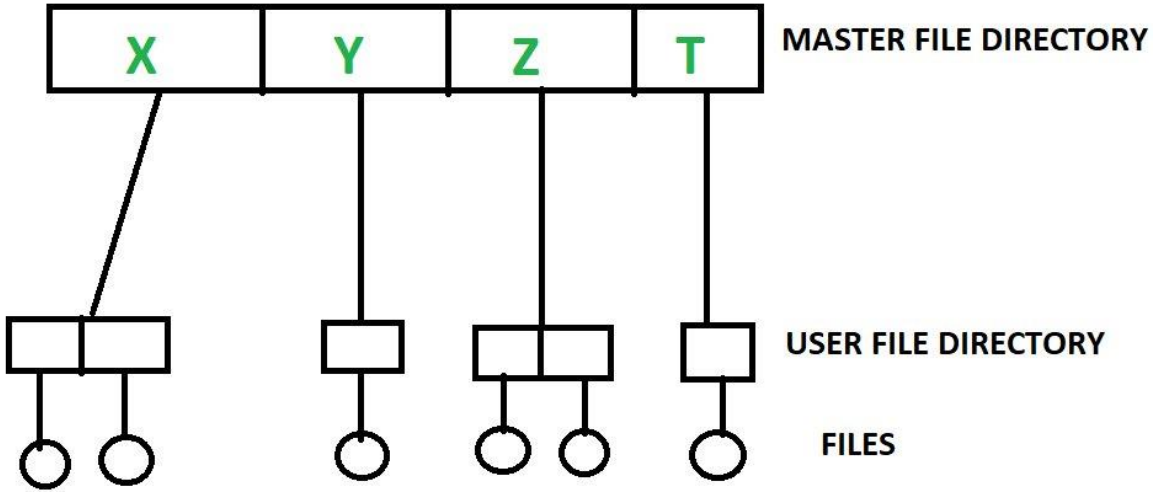
DATA can be any type NOT just NUMBERS

FILE SYSTEM Vs. DBMS

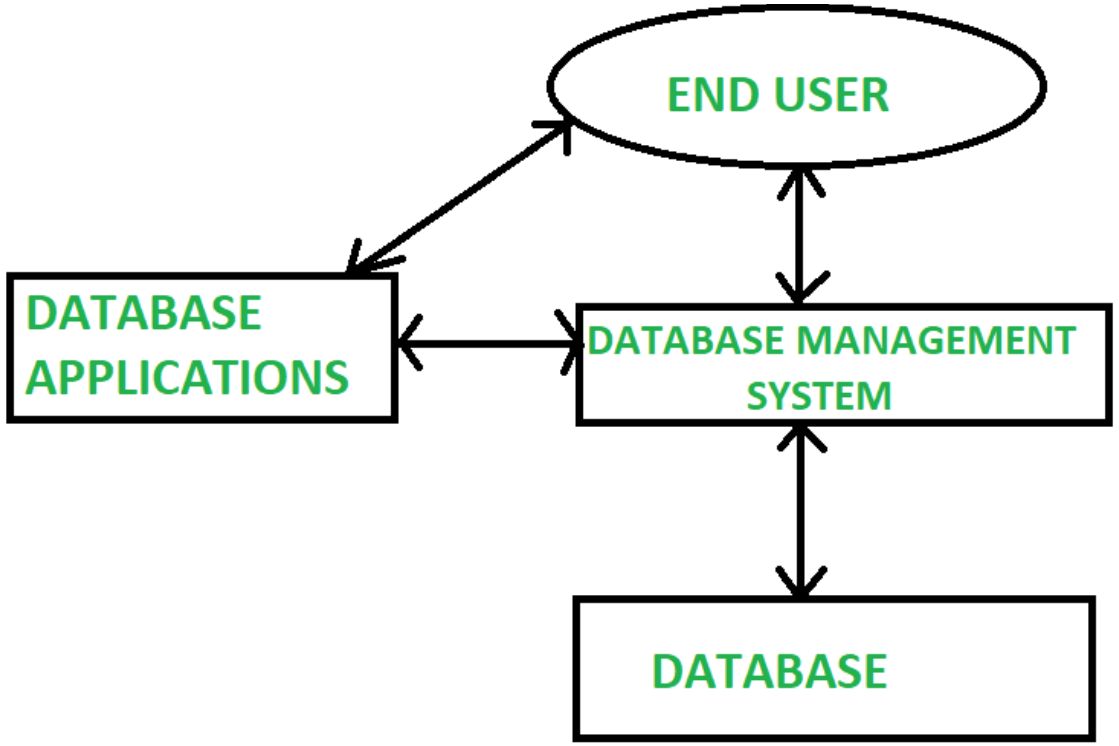
Difference between File System and DBMS

FILE SYSTEM

- A File System is a way of organizing files into groups and folders and then storing them in a storage device.
- It provides the media that stores data as well as enables users to perform procedures such as reading, writing, and even erasure.



Difference between File System and DBMS



DBMS

- DBMS is a more elaborate software application that is solely charged with the responsibility of managing large amounts of structured data.
- It provides functionalities such as query, index, transaction, as well as data integrity.

Difference between File System and DBMS

Feature	File System	DBMS
Definition	Stores data as separate files within directories	Stores data in tables (or documents) with structured relationships
Data Structure	Unstructured / loosely structured	Highly structured (tables, fields, schemas)
Data Redundancy	High (same data may appear in multiple files)	Low (controlled using normalization)
Data Consistency	Low – hard to maintain across multiple files	High – DBMS enforces rules & constraints
Data Security	Basic OS-level security	Advanced security (user roles, permissions, encryption)
Backup & Recovery	Manual	Automatic backup, recovery, and crash management

Difference between File System and DBMS

Feature	File System	DBMS
Concurrency	Poor – risk of conflicts when multiple users access same file	Excellent – supports multiple users simultaneously with transaction control
Querying	No built-in query language	Uses SQL / query language to retrieve data efficiently
Integrity Constraints	Not supported	Supports constraints such as PRIMARY KEY, UNIQUE, FOREIGN KEY
Scalability	Limited	Highly scalable for large applications
Relationships Between Data	Hard to maintain manually	Easily managed through relational models
Performance on Big Data	Slow	Fast, optimized indexing, caching, query optimization

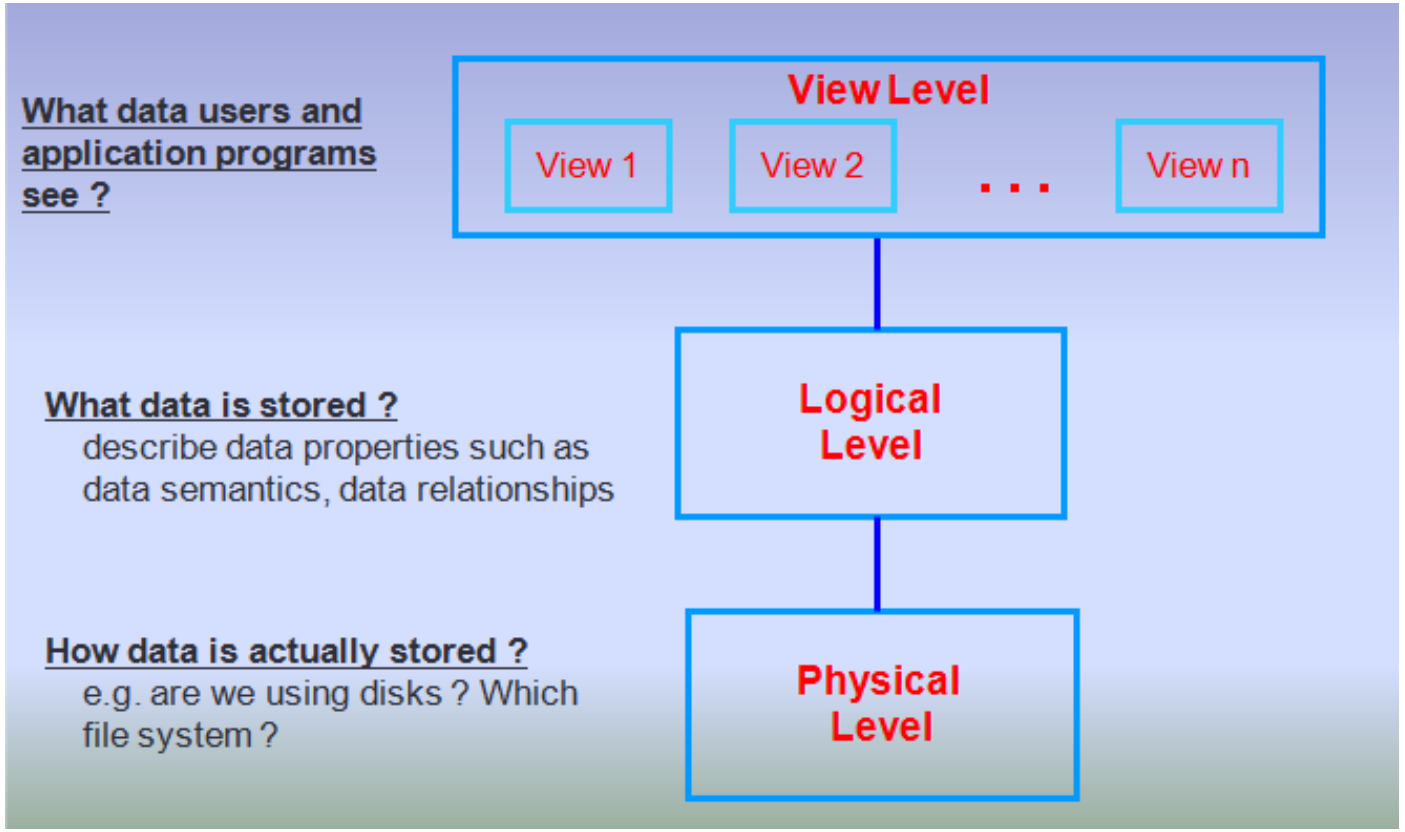
DATA ABSTRACTION

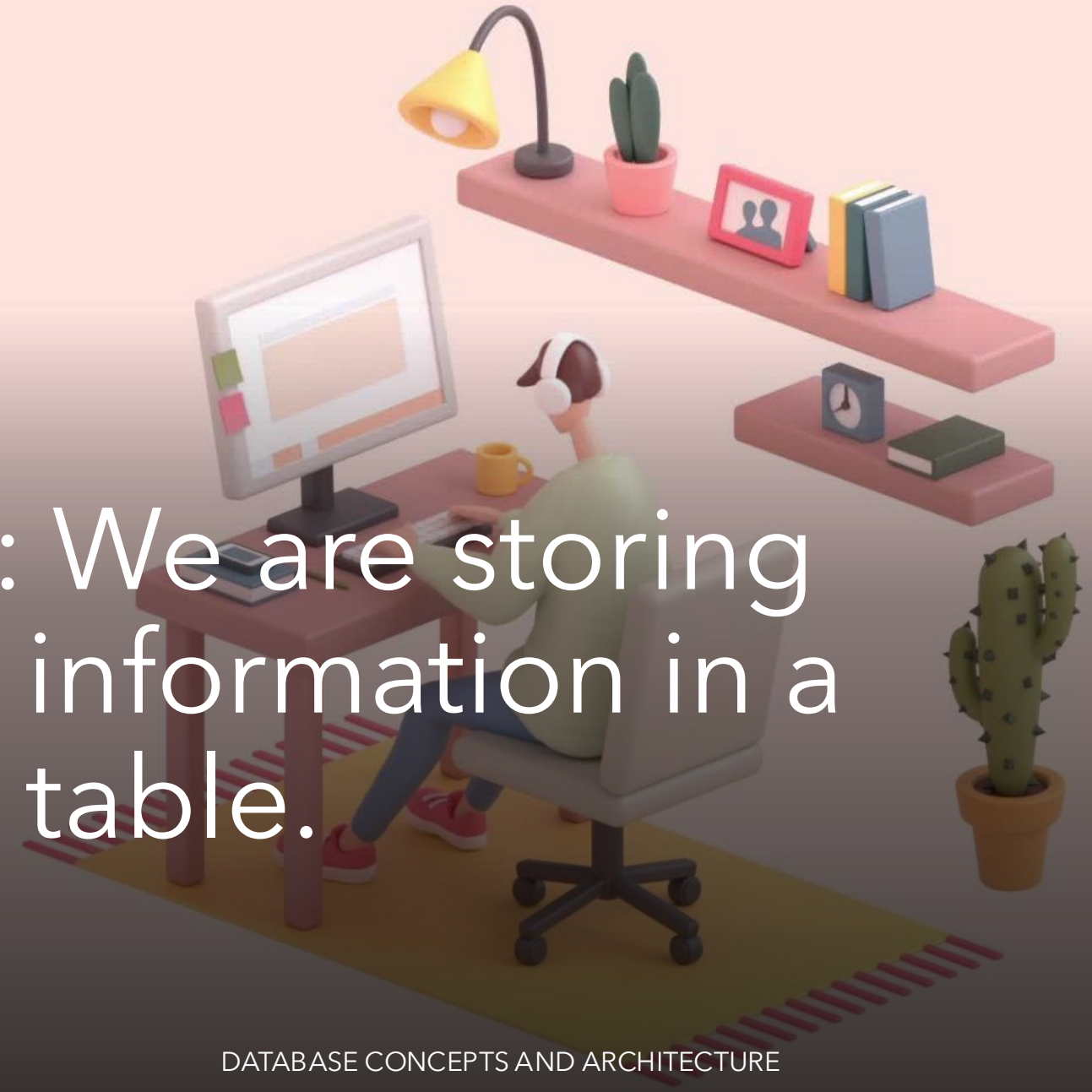
- Database systems are made-up of complex data structures.
- To ease the user interaction with database, the developers hide internal irrelevant details from users.
- This process of hiding irrelevant details from user is called data abstraction.

DATA ABSTRACTION

There are three levels of abstraction:

1. Physical –describes how a record is stored
2. Logical –describes data stored in a database & the relationships among the data.
3. View –describes the user interaction with database





Example : We are storing customer information in a customer table.

PHYSICAL LEVEL | *"How is the data actually stored?"*

This is the **lowest level** of abstraction. It describes **how the customer records are stored on the storage device**.

Customer Table Example (Physical Level): At this level, you **do NOT see columns like Name or Phone**. Instead, we see:

- Data stored in blocks on disk
- Records stored as bytes
- Index files used to speed up searching for Customer IDs
- How many bytes each field uses
- How database pages hold multiple customer records
- **Users never see this.** It is managed completely by the DBMS.

LOGICAL LEVEL

"What data is stored and how is it related?"

This is the **middle level** and the most important for developers and database designers. We would see the **structure** of the Customer table:

We also see relationships such as:

- **CustomerID** connects to **Orders.CustomerID** (1-to-many relationship)
- **Email must be unique**
- **Name cannot be null**
- Developers work at this level.

Field	Type
CustomerID	Integer (Primary Key)
Name	Varchar(100)
Email	Varchar(100)
Phone	Varchar(15)
Address	Varchar(200)

VIEW LEVEL | *"How does the user see the data?"*

This is the **highest level** of abstraction. Different users see **different views** of the same data — depending on their needs.

Customer Table Example (View Level):

1. View for a Salesperson: Customer Name | Phone
2. View for Billing Department: Customer Name | Address | Email | Billing Info
3. View for Customer Service: CustomerID | Name | Phone | Recent Orders
4. Each user sees **only the data they need**, even though all of it exists in the database.
5. This improves **security**, **simplicity**, and **user experience**.

Summary of Example of Data Abstraction Levels

Level	Description	Example (Customer Table)
Physical Level	How data is stored on disk	Byte layout, files, blocks, indexing
Logical Level	Structure of the data and relationships	Table schema: CustomerID, Name, Email, etc.
View Level	How users see the data	User-specific screens showing selected columns

DATA INDEPENDENCE

- The main purpose of data abstraction is achieving data independence to save time and cost required when the database is modified or altered.
- Eg: You have a bunch of apps on your phone, and each app needs to store and use data in some way. Now, each app might use the data in a different way –some might want to organize it one way, while others might want to organize it differently.

LEVEL OF DATA INDEPENDENCE

Physical level data independence

- It refers to the ability to change the physical storage of data without affecting the logical schema.
- Changes like altering indexes, switching file organization methods or upgrading storage devices do not affect the conceptual structure of the database

Logical level data independence

- It refers to the ability to modify the logical schema (tables, attributes, relationships) without affecting the user views or application programs.
- Users can continue to access data without changes in their queries even if new fields are added or relationships modified in the conceptual schema.

LEVEL OF DATA INDEPENDENCE

Physical level data independence

Examples:

- Changing from sequential file organization to hashing.
- Using a new storage device or modifying access paths.

Benefit: *It allows performance optimizations at the physical level without modifying higher-level structures.*

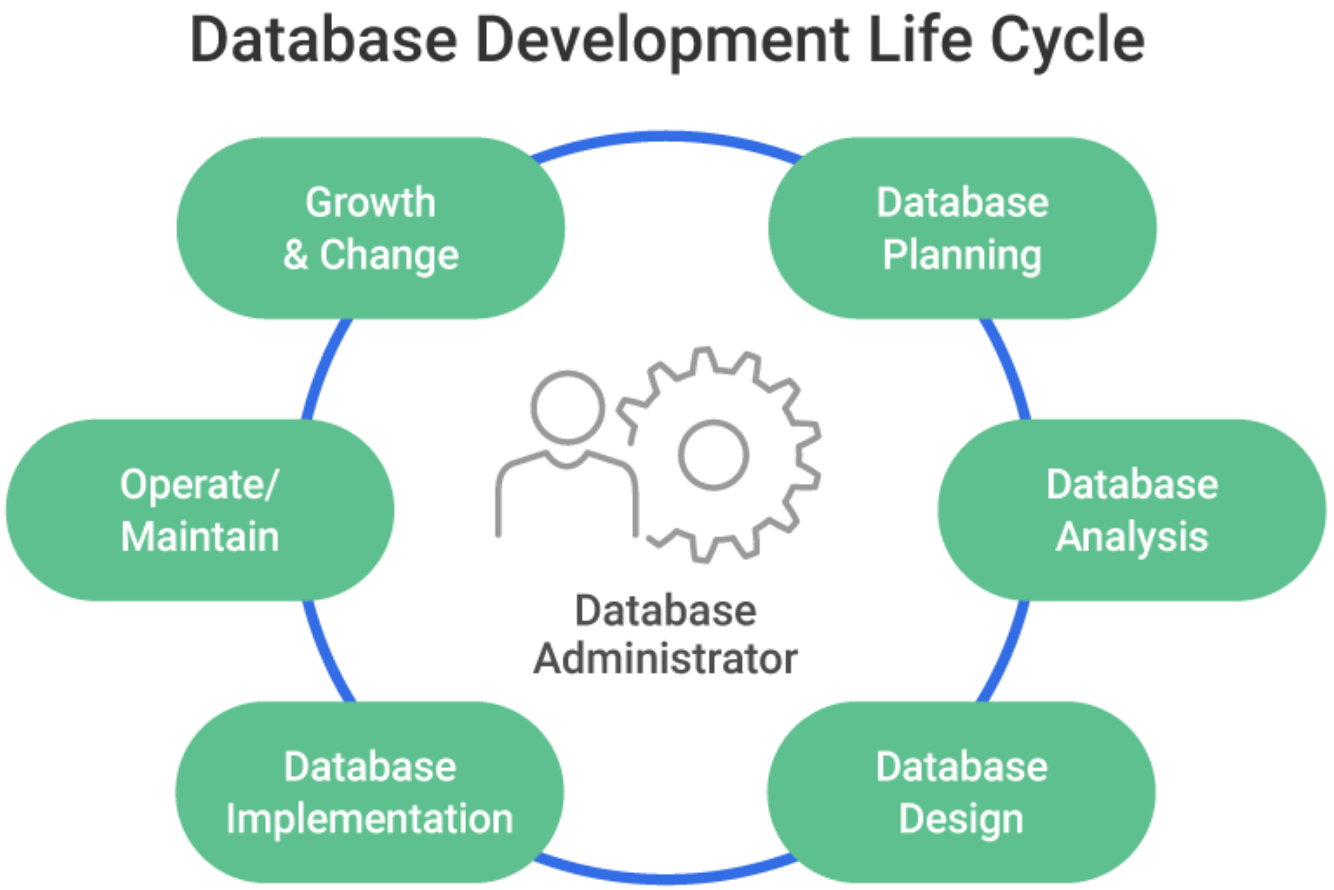
Logical level data independence

Examples:

- Adding or deleting attributes in a table.
- Changing the relationship between entities.

Benefit: *It allows the database to evolve without disturbing existing applications.*

DATABASE ADMINISTRATOR



DATABASE ADMINISTRATOR

Database administrators are highly trained, technically skilled individuals who use modern, cloud platforms to organize, store, and protect critical data.

To follow the basic functions of a database coordinator, DBAs coordinate the business intelligence (BI) systems that data analysts and data scientists rely on to translate raw data into insights that can drive strategic business plans.

As such, they play a key role in the modern big data architectures that enable large-scale modern BI.

ROLES OF DBA

The role of database administrator traditionally includes the following activities:

- Implementing data policies, procedures, standards
- Planning and development of an organization’s enterprise data model
- Resolving data conflicts
- Managing data repositories
- Optimizing data warehouse, lake and other storage
- Define and model data requirements, business rules, operational requirements, and maintain corporate data dictionaries
- Disaster recovery planning

DBAs are defined by their roles

Depending on the size and shape of an organization, database administrators can wear many hats or focus on a specific role that warrants a different title.

General-purpose DBAs perform all types of administrative and data-related work.

Task-oriented DBAs handle a specific administrative database task.

System DBAs focus on the physical installation and performance of the DBMS software.

Performance analysts monitor and improve the performance of applications accessing databases.

Database architects design new databases and structures for new and existing applications.

Data warehouse administrators monitor and support the data warehouse environment.

Database analysts review, evaluate, design and implement databases.

Cloud DBAs perform all types of administrative and data-related work for cloud databases.

Application DBAs focus on database support for a specific application or subset of applications.

REFERENCES OF DIFFERENT DATABASE TYPE

- bpldatabase.org. (2025). Hierarchical database model examples and uses.
- MongoDB. (2025). What is a hierarchical database?
- Khalid, U. (2025). What is centralized data management and why is it important? CentricDXB.
- WorkWave Insights. (2025). How centralized data warehouse transforms multi-branch operations.
- Mărcuță, C., & MoldStud Research Team. (2025). Real-world applications for different NoSQL database types.
- Parmar, R. (2025). Real-world DBMS examples. DEV Community.
- Decide Software. (2025). Top object databases: Reviews, features & comparisons.
- The Knowledge Academy. (2025). Network database: Definition, importance, and examples.
- T4Tutorials. (2025). Network model in DBMS with examples and characteristics.
- Olibr. (2024). 12 database examples in real life.