Department of CSE

COURSE NAME: DBMS
COURSE CODE:23AD2102R

Topic: NOSQL

Session - I

AIM OF THE SESSION

To familiarize students with the basic concept of SQL languages and Create table command in detail.

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Discuss the types of SQL languages.
2. Various command under different types of SQL Languages.
3. Introduction of Create table command with its syntax and examples.

LEARNING OUTCOMES

At the end of this session, you should be able to understand the basic commands of SQL and learn how to write queries with SQL commands.

SESSION INTRODUCTION

KEY CONCEPTS

- Definition and Purpose of NoSQL Databases:
- Introduction to NoSQL Databases
- Types of NoSQL Databases
- Characteristics
- Use Cases and Applications

Definition and Purpose of NoSQL Databases

- NOSQL means Not only SQL.
- NoSQL databases are designed to handle large volumes of unstructured, semi-structured, or structured data.
- They offer flexibility, scalability, and performance improvements over traditional relational databases.
- Making them suitable for modern web applications, real-time analytics, and big data processing.
- NOSQL systems focus on storage of “big data” Typical applications that use NOSQL.

Introduction to NoSQL Databases

❖What is NoSQL?

❖NoSQL database, also called Not Only SQL, is an approach to data management and database design that’s useful for very large sets of distributed data.

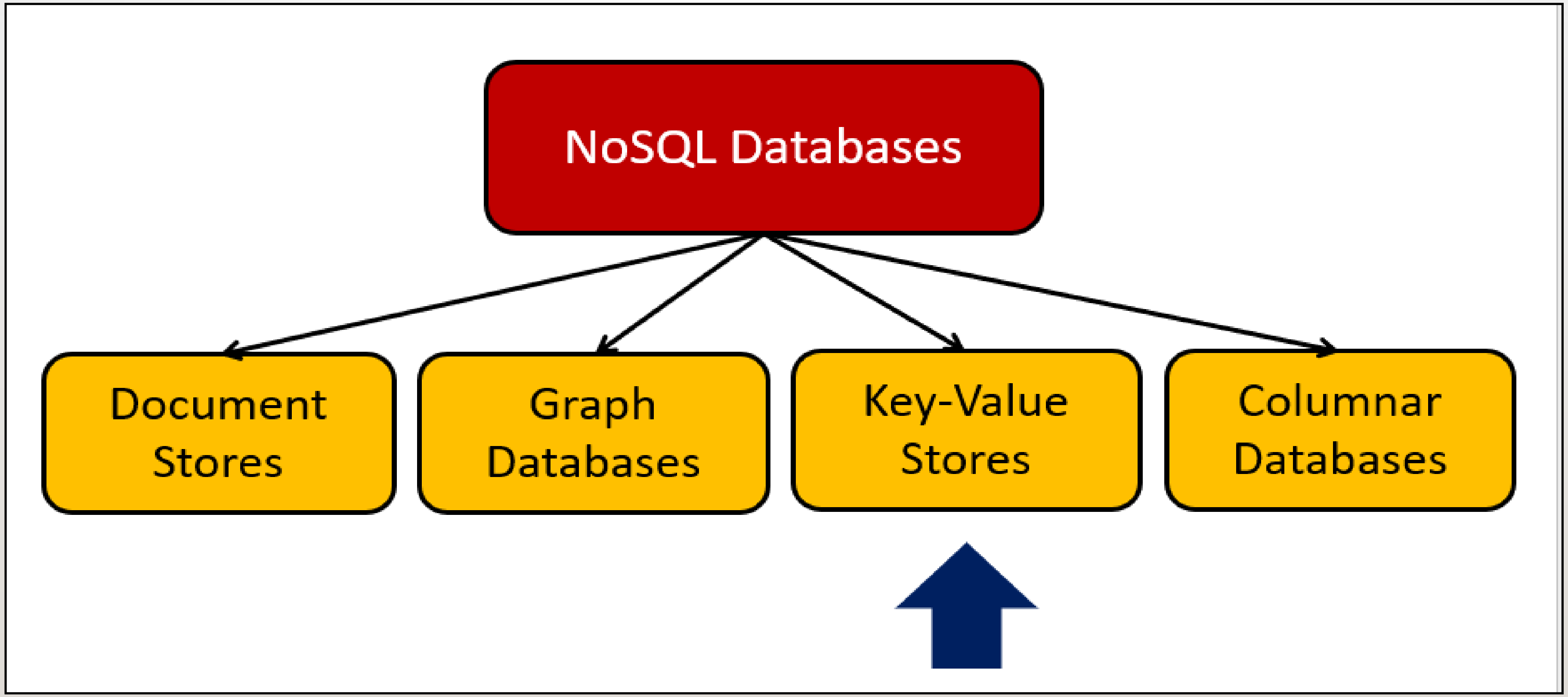
❖NoSQL is not a relational database.

❖A relational database model may not be the best solution for all situations.

❖The easiest way to understand NoQL, is that of a database which does not adhering to the traditional relational database management system (RDMS) structure.

Types of NoSQL Databases

Here is a limited taxonomy of NoSQL databases:



NoSQL Databases

Document Stores

Graph Databases

Key-Value Stores

Columnar Databases

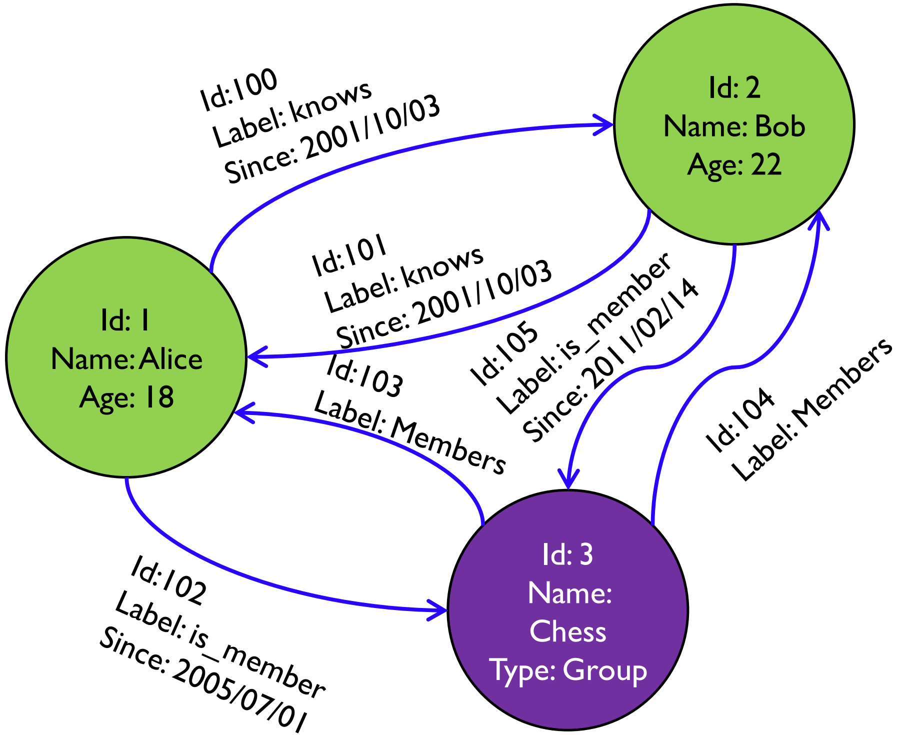
DOCUMENT STORES

- These databases store data in documents, typically JSON or BSON, which allows for nested structures and flexible schemas.
- They are well-suited for content management systems, e-commerce applications, and applications with varying data structures
- **Examples**: MongoDB, CouchDB

- Store data in documents (JSON/BSON).
- Examples: MongoDB, CouchDB.

GRAPH DATABASES

- Data are represented as vertices and edges.



- Graph databases are powerful for graph-like queries (e.g., find the shortest path between two elements)
- E.g., Neo4j and VertexDB, Amazon Neptune

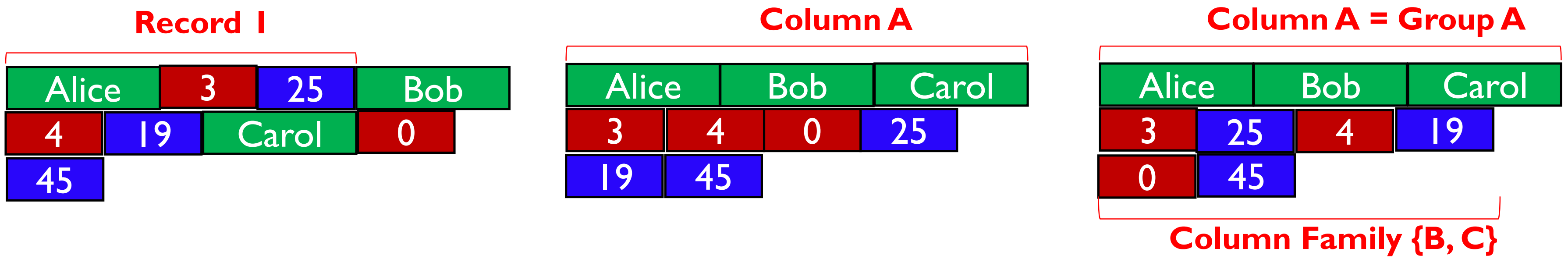
KEY-VALUE STORES:

- These databases store data as a collection of key-value pairs, where each key is unique.
- They are simple, fast, and highly scalable, making them ideal for caching, session management, and real-time data processing.
- Keys are mapped to (possibly) more complex value (e.g., lists)
- Keys can be stored in a hash table and can be distributed easily
- Such stores typically support regular CRUD (create, read, update, and delete) operations
- That is, no joins and aggregate functions
- E.g., Amazon DynamoDB and Apache Cassandra

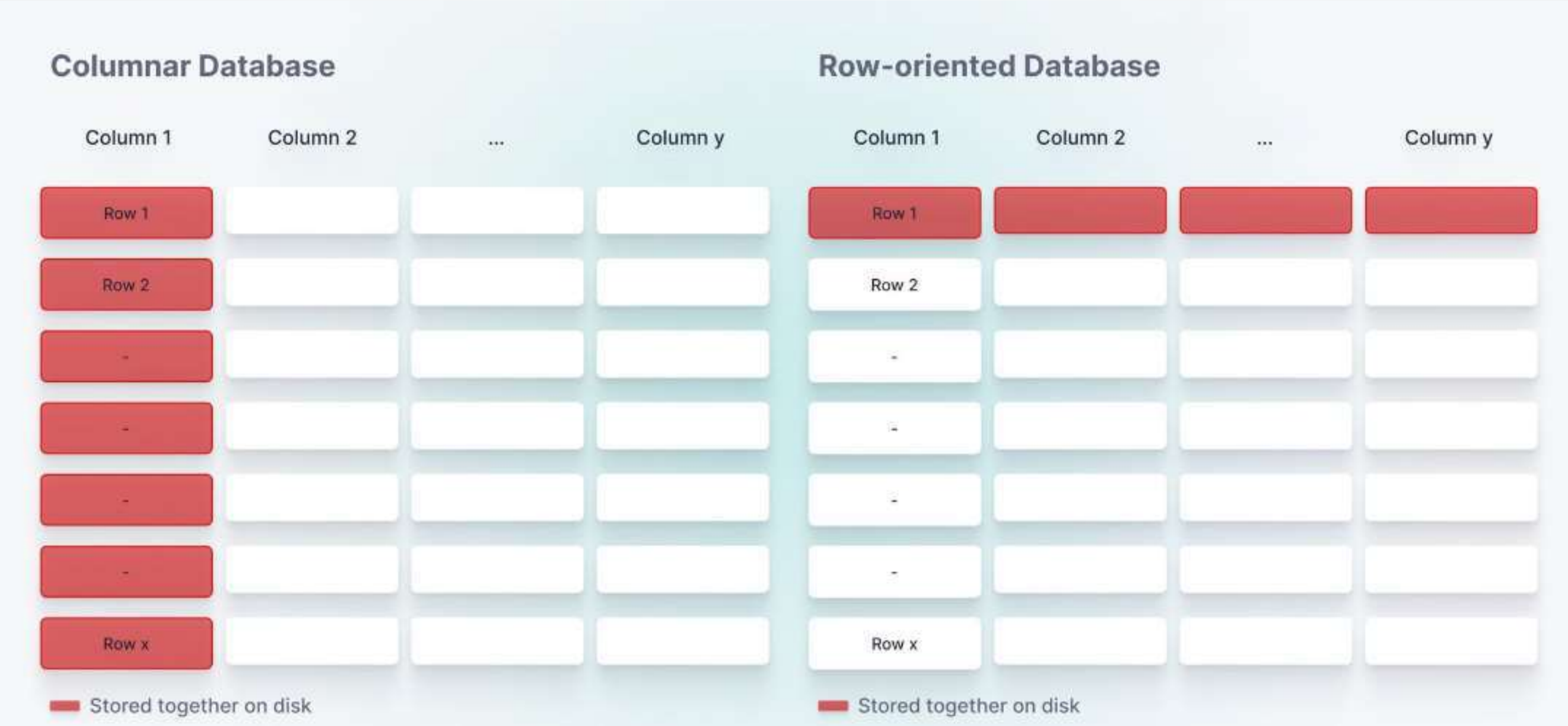
Examples: Redis, DynamoDB

COLUMNAR DATABASES

- Columnar databases are a hybrid of RDBMSs and Key-Value stores
- Values are stored in groups of zero or more columns, but in Column-Order (as opposed to Row-Order)
- Values are queried by matching keys
- E.g., HBase and Vertica



Row-Order ***Columnar (or Column-Order)*** ***Columnar with Locality Groups***



Columnar Database

Column 1	Column 2	...	Column y
Row 1			
Row 2			
-			
-			
-			
-			
Row x			

🟥 Stored together on disk

Row-oriented Database

Column 1	Column 2	...	Column y
Row 1			
Row 2			
-			
-			
-			
-			
Row x			

🟥 Stored together on disk

Characteristics of NoSQL Databases

1.Schema-less Design:

•NoSQL databases typically do not require a fixed table schema, allowing for flexible and dynamic data structures. This is advantageous for applications where data formats can change frequently.

2.Scalability:

•NoSQL databases are designed to scale out by distributing data across multiple servers. This horizontal scaling allows them to handle large volumes of data and high traffic loads efficiently.

3. **Flexible Data Models**:

•They support various data models:

- **Key-Value Stores**: Simple data storage models that use keys to access values (e.g., Redis, Riak).
- **Document Stores**: Store and retrieve data as documents, typically in JSON or BSON format (e.g., MongoDB, CouchDB).

4.High Performance:

- NoSQL databases are optimized for performance, often providing faster data retrieval and write operations compared to relational databases. They achieve this by avoiding complex join operations and using simplified data access patterns.

5.Eventual Consistency:

- Many NoSQL databases follow the eventual consistency model, which means that data changes are propagated across the system over time, and consistency is achieved eventually.

6.High Availability:

- They are designed to be highly available and fault-tolerant, often using replication and partitioning to ensure data is always accessible, even in the event of hardware failures.

7.Distributed Architecture: NoSQL databases often have a distributed architecture, allowing data to be stored and processed across multiple nodes, enhancing fault tolerance and load distribution.

8.Handling of Big Data:

They are well-suited for big data applications due to their ability to handle large volumes of unstructured, semi-structured, and structured data.

SUMMARY

1. In this section, we discussed the various types of SQL languages and commands of these languages that are used to work on the database nd the data stored in the databases.

2. We also discussed the CREATE TABLE command in detail with its syntax and examples for creating the tables.

SELF-ASSESSMENT QUESTIONS

1. Commands that comes under DDL is/are –

(a) CREATE

(b) DROP

(c) TRUCATE

(d) ALLOF THE ABOVE

2. Command that comes under DML is/are –

(a) ROLLBACK

(b) GRANT

(c) UPDATE

(d) ALL OF THE ABOVE

SELF-ASSESSMENT QUESTIONS

3. Command that comes under DCL is/are -

(a) GRANT

(b) REVOKE

(c) BOTH (a) AND (b)

(d) NONE OF THE ABOVE

4. Following the completion of a transaction, it must be executed to save all the operations performed in the transaction. Here we are talking about which command?

(a) REVOKE

(b) COMMIT

(c) ROLLBACK

(d) SAVE

TERMINAL QUESTIONS

1. Describe various types of SQL Languages.

2. List out the commands of Data Definition Language with examples.

3. Analyze the DDL commands in PostgreSQL.

4. Summarize the create table command with its syntax and examples.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books:

1. Database System Concepts, Sixth Edition, Abraham Silberschatz, Yale University Henry, F. Korth Lehigh University, S. Sudarshan Indian Institute of Technology, Bombay.

2. An Introduction to Database Systems by Bipin C. Desai

3. Fundamentals of Database Systems, 7th Edition, RamezElmasri, University of Texas at Arlington, Shamkant B. Navathe, University of Texasat Arlington.

Sites and Web links:

1. https://www.geeksforgeeks.org/postgresql-create-table/

2. https://www.tutorialsteacher.com/postgresql

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – DBMS