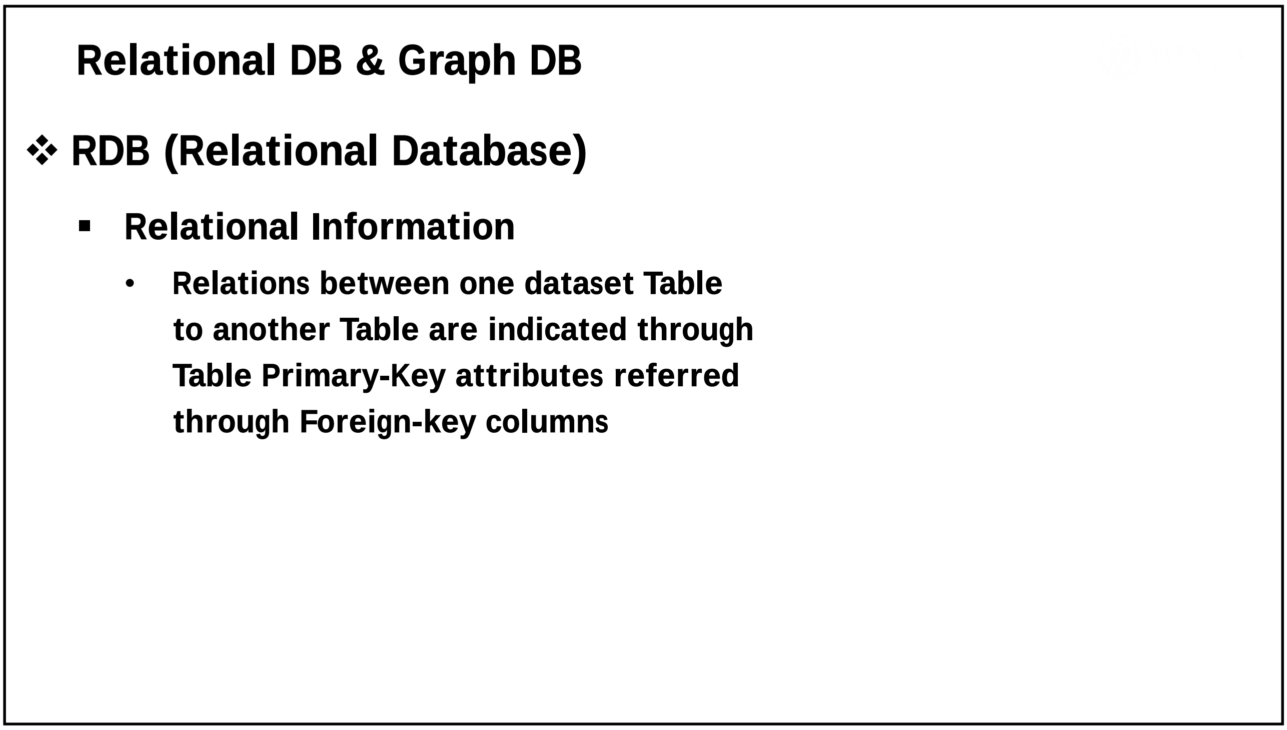
Big Data

Relational DB & Graph DB

Relational DB & Graph DB

❖ RDB (Relational Database)

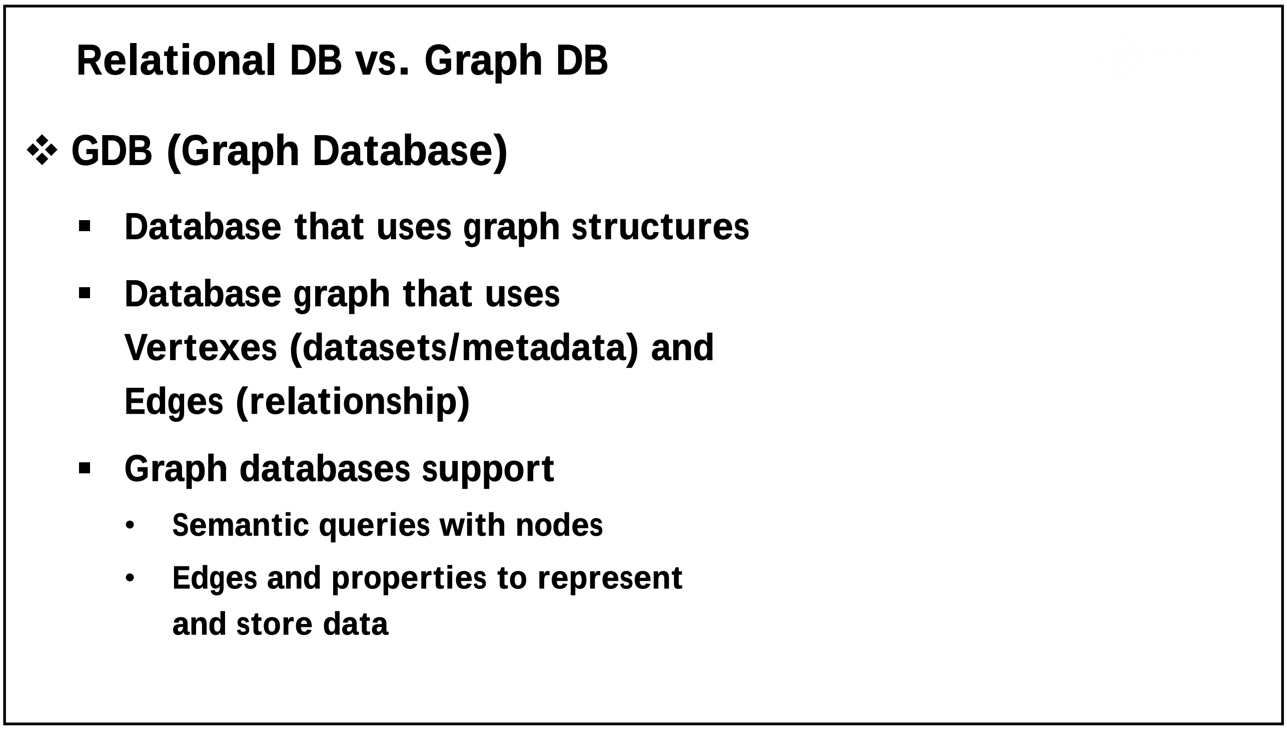
- Representative database used in all software applications since the 80s
- Structured data is stored in tables
- Columns define data types
- Rows are collections of same data types from different datasets



Relational DB & Graph DB

❖ RDB (Relational Database)

- Relational Information
  - Relations between one dataset Table to another Table are indicated through Table Primary-Key attributes referred through Foreign-key columns



Relational DB vs. Graph DB

❖ GDB (Graph Database)

- Database that uses graph structures
- Database graph that uses Vertexes (datasets/metadata) and Edges (relationship)
- Graph databases support
  - Semantic queries with nodes
  - Edges and properties to represent and store data

Relational DB vs. Graph DB

❖ GDB (Graph Database)

- Vertexes (nodes) represent an Entity or Attribute (Metadata)
- Edges (links) represent the Relationship of the Vertexes
- Relationships records are organized based on type, features, direction, correlation, statistics, etc.

Relational DB vs. Graph DB

❖ Metadata

- Data that provides information about other data
- Metadata Types
  - Descriptive Metadata
  - Structural Metadata
  - Administrative Metadata

Relational DB vs. Graph DB

❖ Metadata

- Metadata Types
  - Descriptive Metadata
    - Data & information on data resources and purposes
    - Method of data discovery, identification, and verification
      - Example: Title, Abstraction, Programmers, Authors, Sources, Keywords, etc.

Relational DB vs. Graph DB

❖ Metadata

- Metadata Types
  - Structural Metadata
    - Data & information on dataset containers
    - Specifics on Categories, Types, Versions, Relationships, Statistics, Characteristics
      - Method of data container collection or creation
      - Dataset compounding basis objects
        - Example: File > Chapters > Sections > Subsections > Tables > Elements

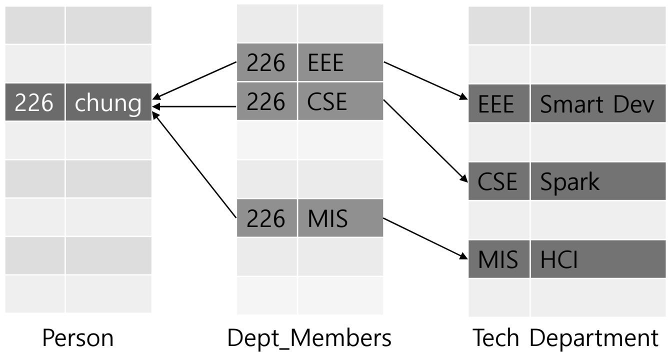
Relational DB vs. Graph DB

❖ Metadata

- **Metadata Types**
  - **Administrative Metadata**
    - **Data & information on resource management and administration**
      - **Method of dataset collection or creation**
      - **Dataset type and file technical information**
      - **User access permission and administration**
      - **Resource management methods and policies**

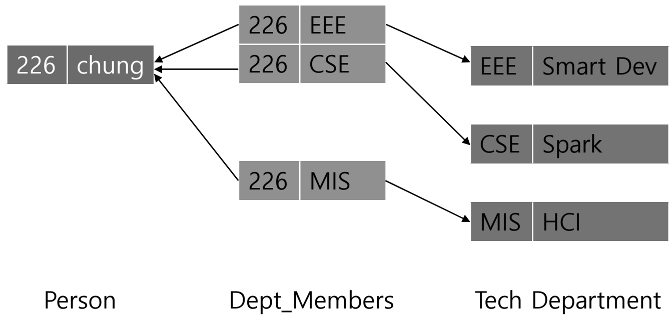
Relational DB & Graph DB

❖ RDB & GDB Representations



Relational DB & Graph DB

❖ RDB & GDB Representations



226 chung

226 EEE

226 CSE

226 MIS

EEE Smart Dev

CSE Spark

MIS HCI

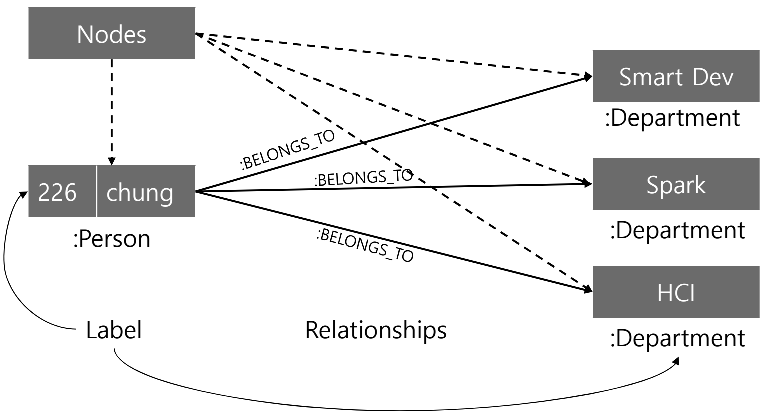
Person

Dept_Members

Tech Department

Relational DB & Graph DB

❖ RDB & GDB Representations



Nodes

226 chung

:Person

:BELONGS_TO

:BELONGS_TO

:BELONGS_TO

Smart Dev

:Department

Spark

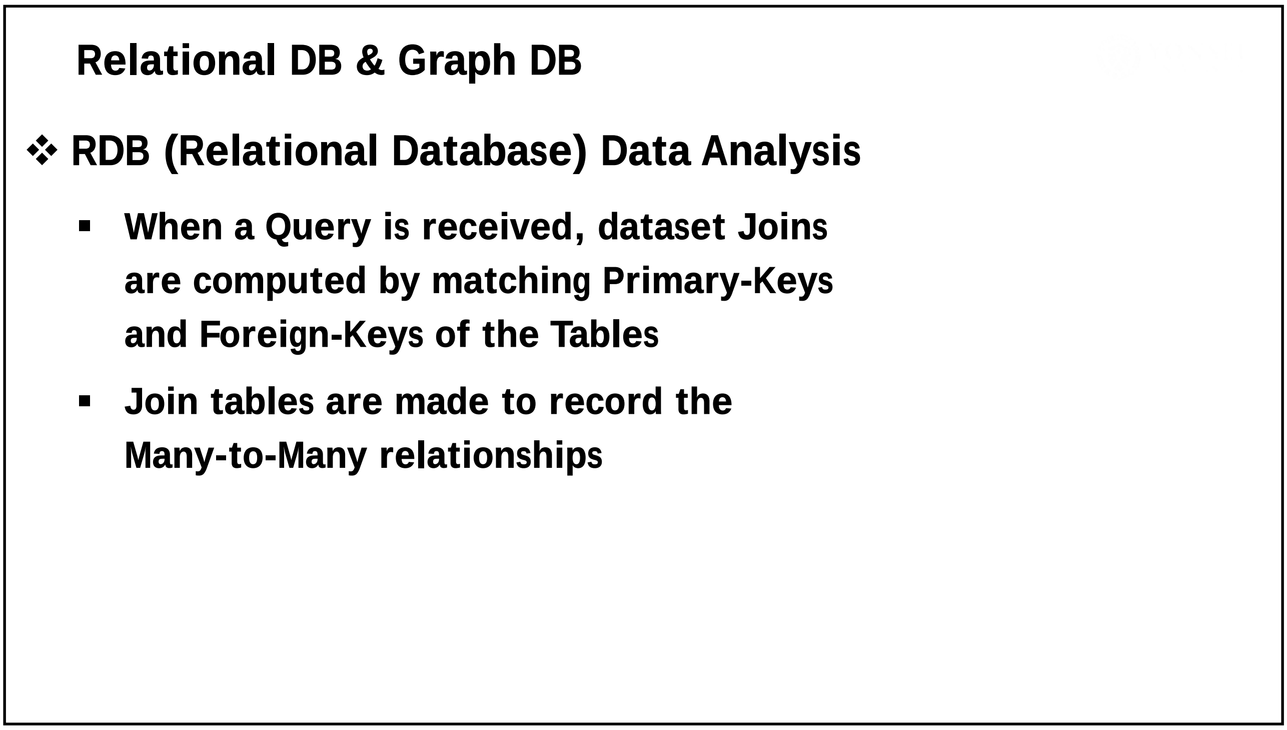
:Department

HCI

:Department

Label

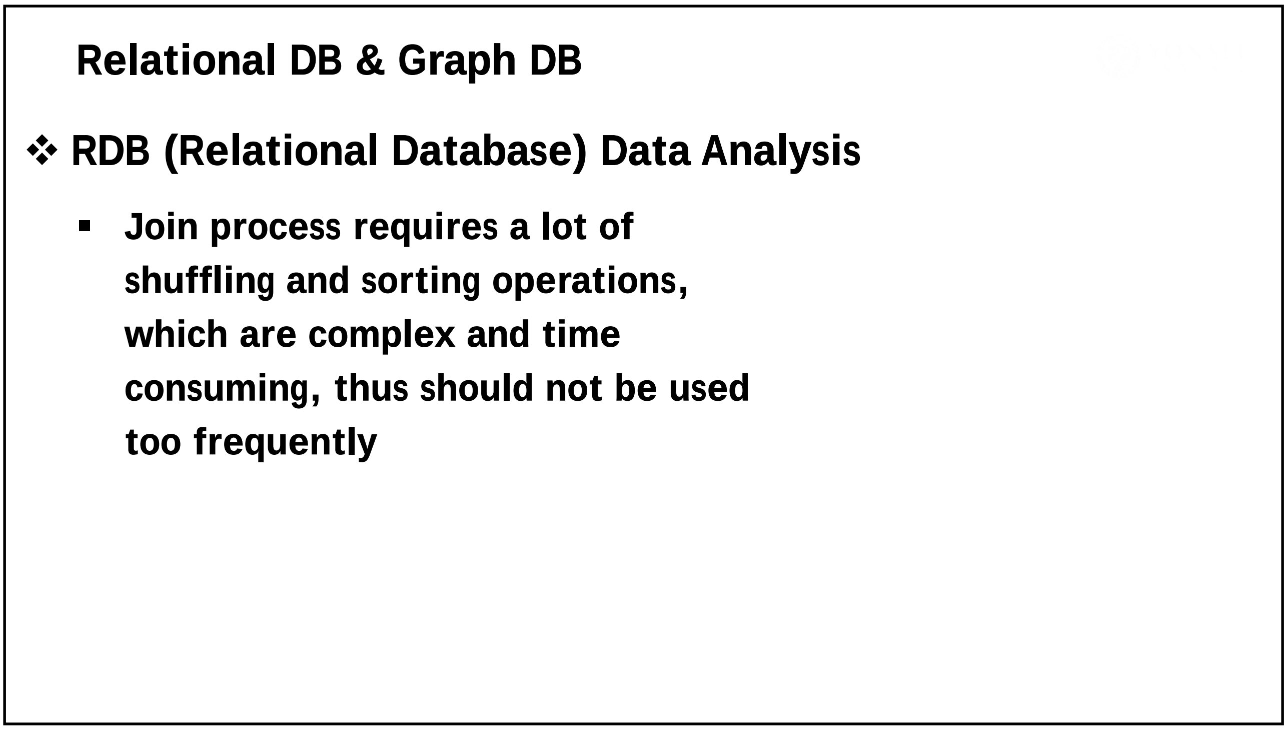
Relationships



Relational DB & Graph DB

❖ RDB (Relational Database) Data Analysis

- When a Query is received, dataset Joins are computed by matching Primary-Keys and Foreign-Keys of the Tables
- Join tables are made to record the Many-to-Many relationships



Relational DB & Graph DB

❖ RDB (Relational Database) Data Analysis

- Join process requires a lot of shuffling and sorting operations, which are complex and time consuming, thus should not be used too frequently

Relational DB & Graph DB

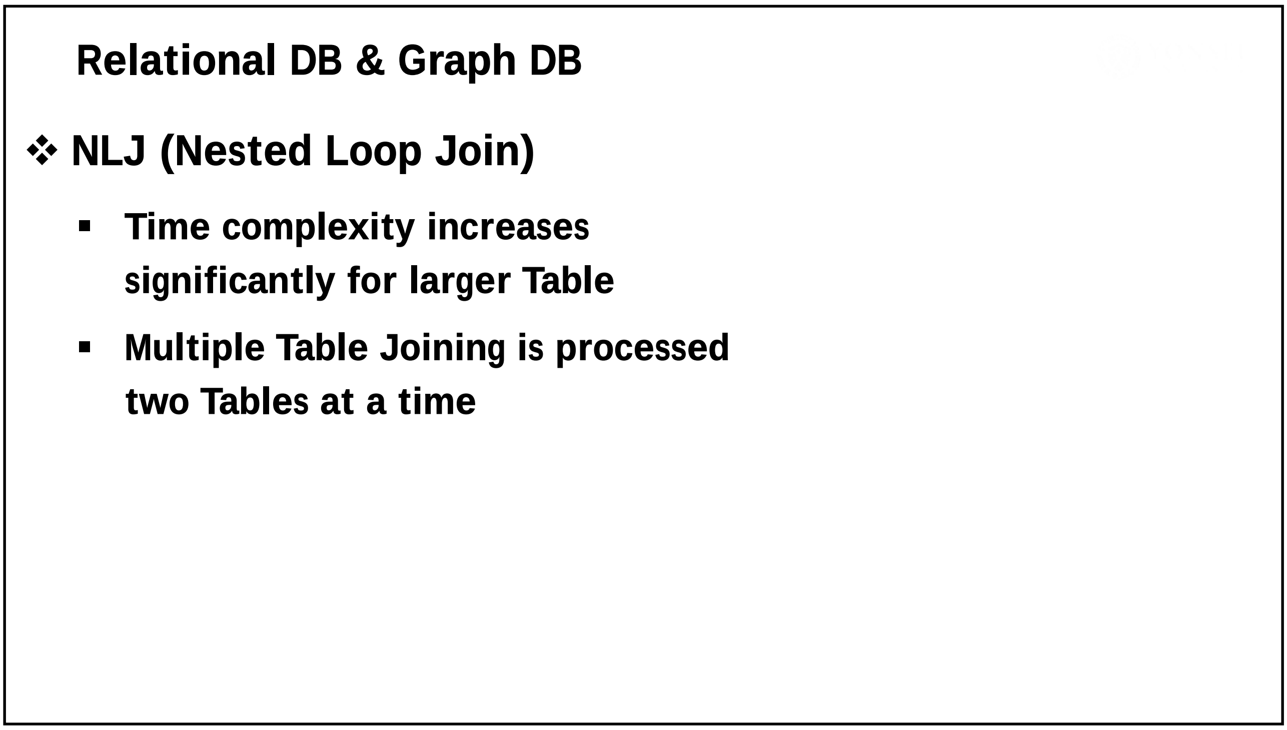
❖ Join

- Process of combining related datasets based on common fields
- Essential process in database/dataset merging and data analysis
- Types of Join
  - NLJ (Nested Loop Join)
  - HJ (Hash Join)

Relational DB & Graph DB

❖ NLJ (Nested Loop Join)

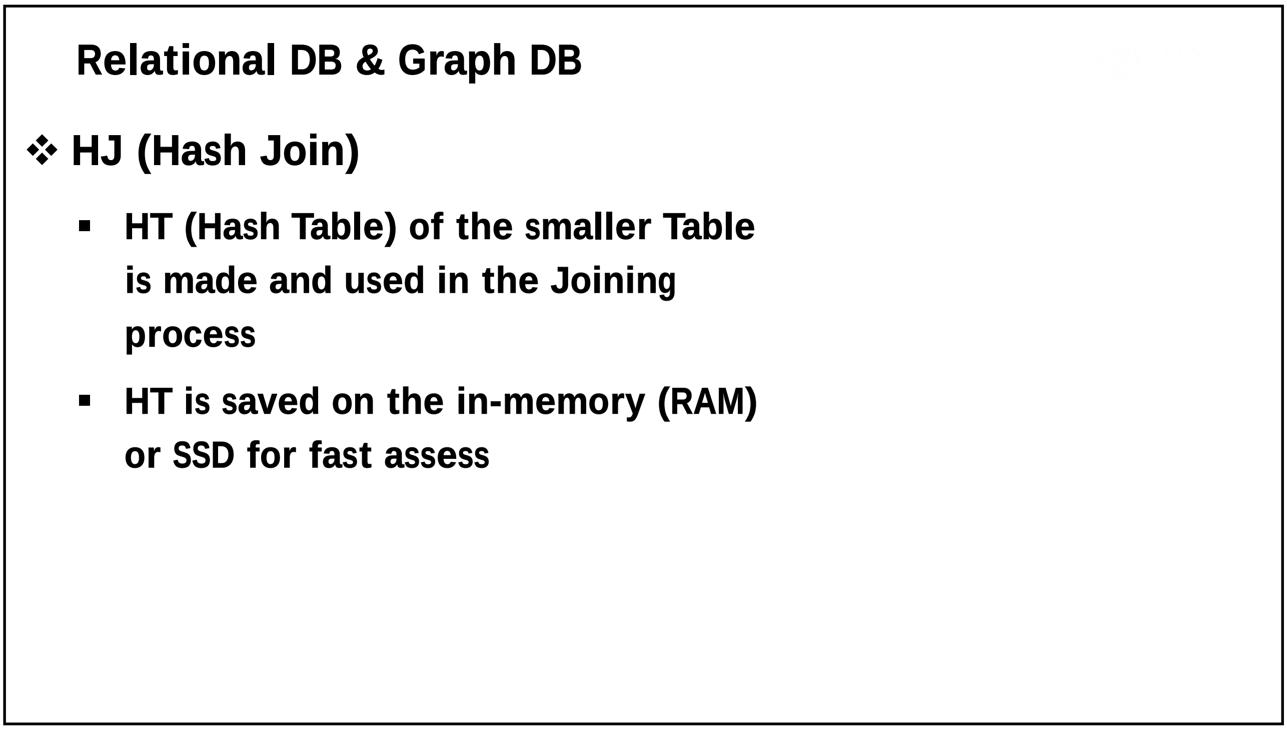
- Simplest Join method
- Uses nested loops to Join two Tables
- Nested loops based Joining process
  - For each row in the inner table, all rows of the outer table are read in order in the Join process



Relational DB & Graph DB

❖ NLJ (Nested Loop Join)

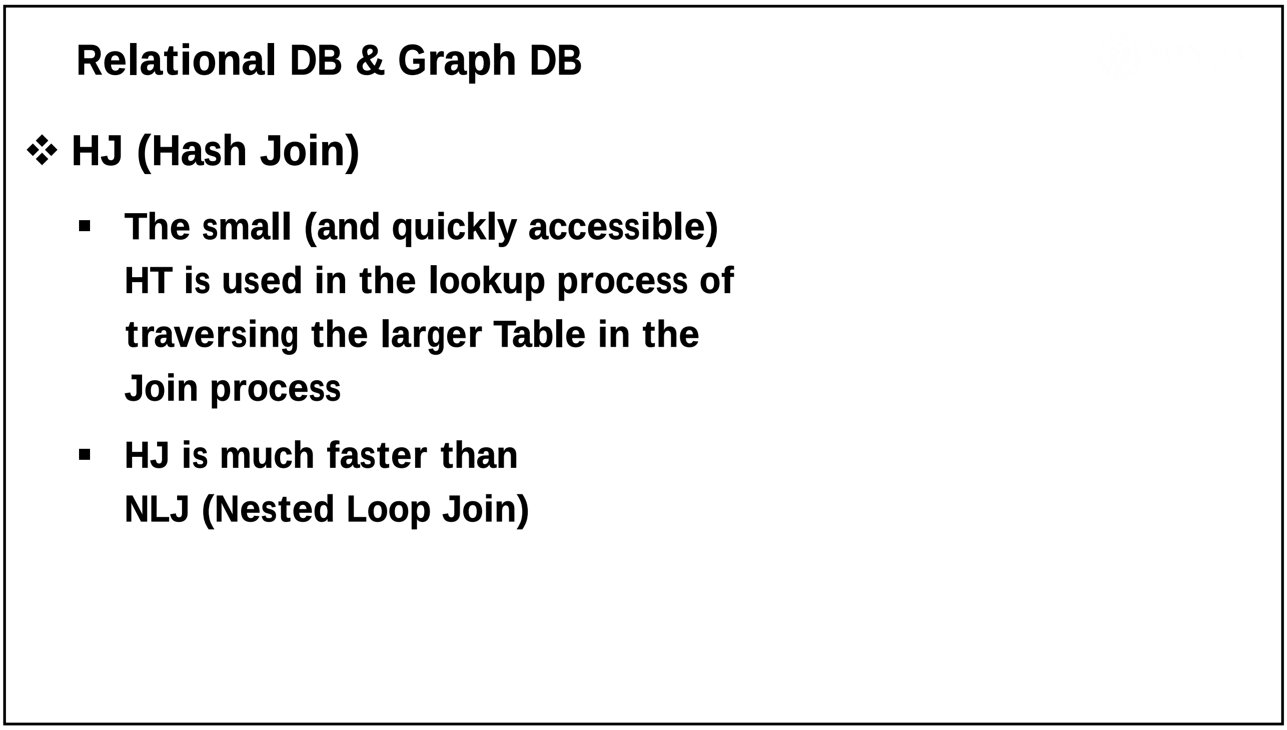
- Time complexity increases significantly for larger Table
- Multiple Table Joining is processed two Tables at a time



Relational DB & Graph DB

❖ HJ (Hash Join)

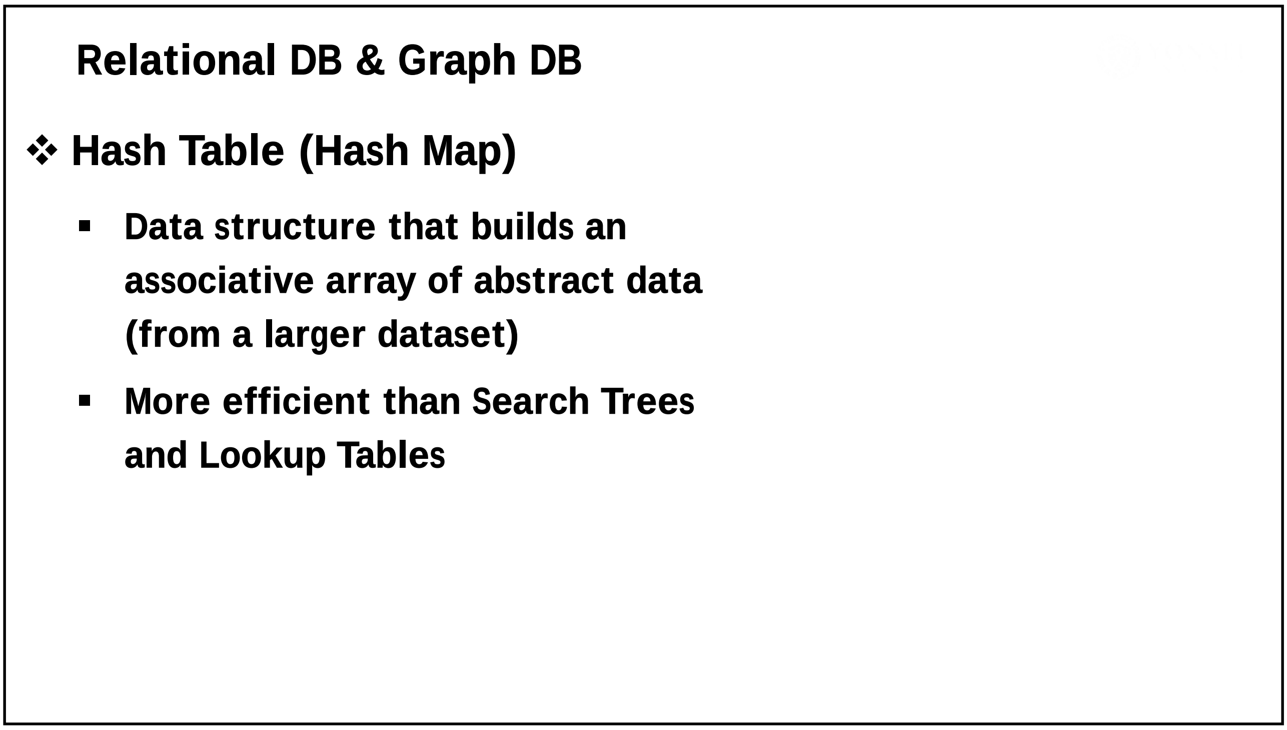
- HT (Hash Table) of the smaller Table is made and used in the Joining process
- HT is saved on the in-memory (RAM) or SSD for fast assess



Relational DB & Graph DB

❖ HJ (Hash Join)

- **The small (and quickly accessible) HT is used in the lookup process of traversing the larger Table in the Join process**
- **HJ is much faster than NLJ (Nested Loop Join)**



Relational DB & Graph DB

❖ Hash Table (Hash Map)

- **Data structure that builds an associative array of abstract data (from a larger dataset)**
- **More efficient than Search Trees and Lookup Tables**

Relational DB & Graph DB

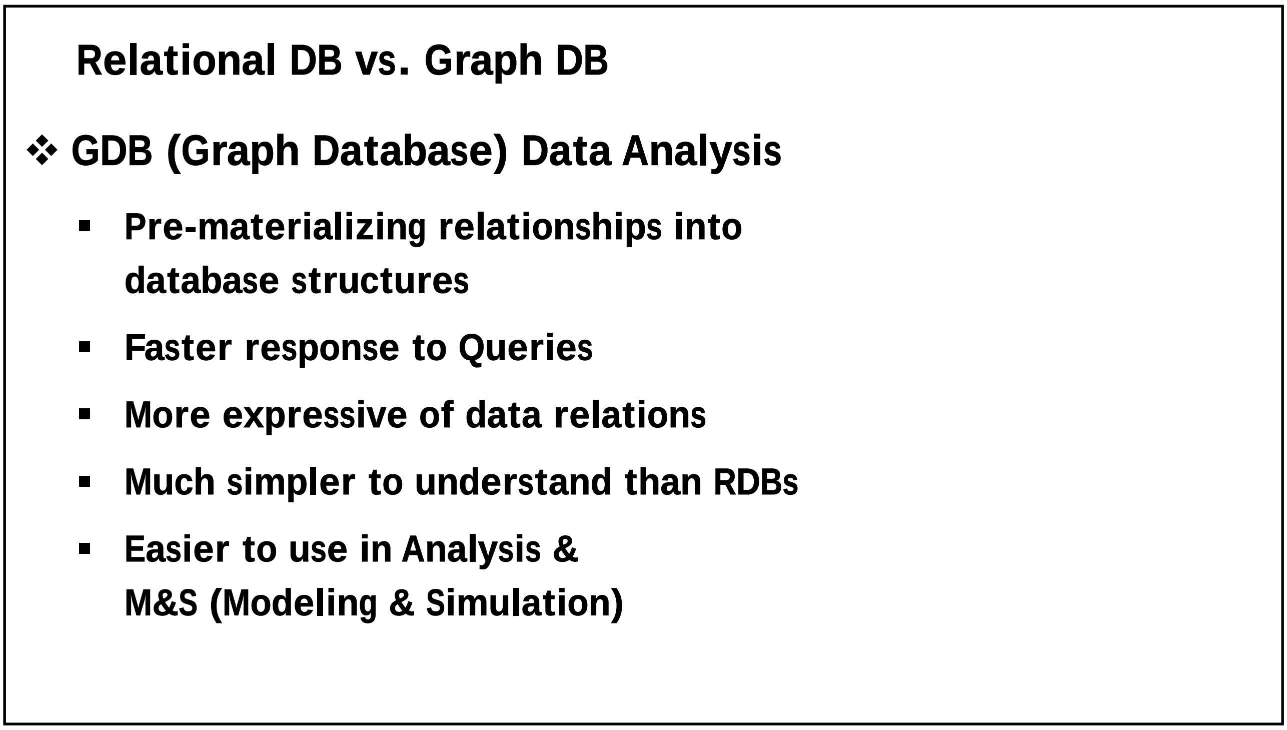
❖ Hash Table (Hash Map)

- **Used to map keys to values**
- **Hash Functions are used to compute indexes into values (an array of buckets/slots) which are placed in the Hash Table**

Relational DB vs. Graph DB

❖ GDB (Graph Database) Data Analysis

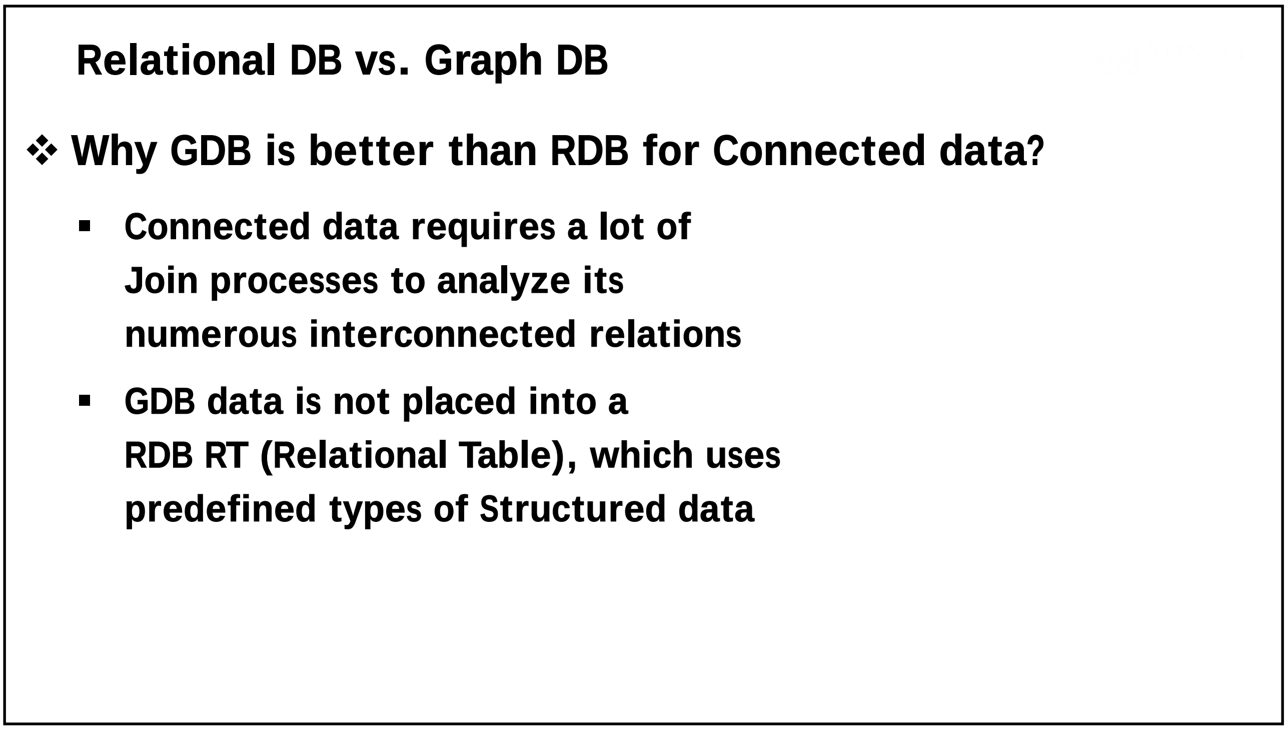
- **For a Join (Shuffle, Sort) operation, the database just uses this list and has direct access to the connected nodes, eliminating the need for a complex and time consuming search & match operation**



Relational DB vs. Graph DB

❖ GDB (Graph Database) Data Analysis

- Pre-materializing relationships into database structures
- Faster response to Queries
- More expressive of data relations
- Much simpler to understand than RDBs
- Easier to use in Analysis & M&S (Modeling & Simulation)



Relational DB vs. Graph DB

❖ Why GDB is better than RDB for Connected data?

- Connected data requires a lot of Join processes to analyze its numerous interconnected relations
- GDB data is not placed into a RDB RT (Relational Table), which uses predefined types of Structured data

Relational DB vs. Graph DB

❖ **Why GDB is better than RDB for Connected data?**

- **GDB data attributes can be added and removed as needed**
- **When Semi-structured data is placed into a RDB RT (Relational Table), much data will be lost (filtered out) and many columns of the RT will be empty (null)**

Relational DB vs. Graph DB

❖ **Why GDB is better than RDB for Connected data?**

- **Since GDB has no predefined structure, data modeling is easier in GDBs**
- **In RDBs, for highly connected data, SQL query programming (syntax) is complex and difficult as the number of Joins has to increase**
- **Can we change an RDB in a GDB?**

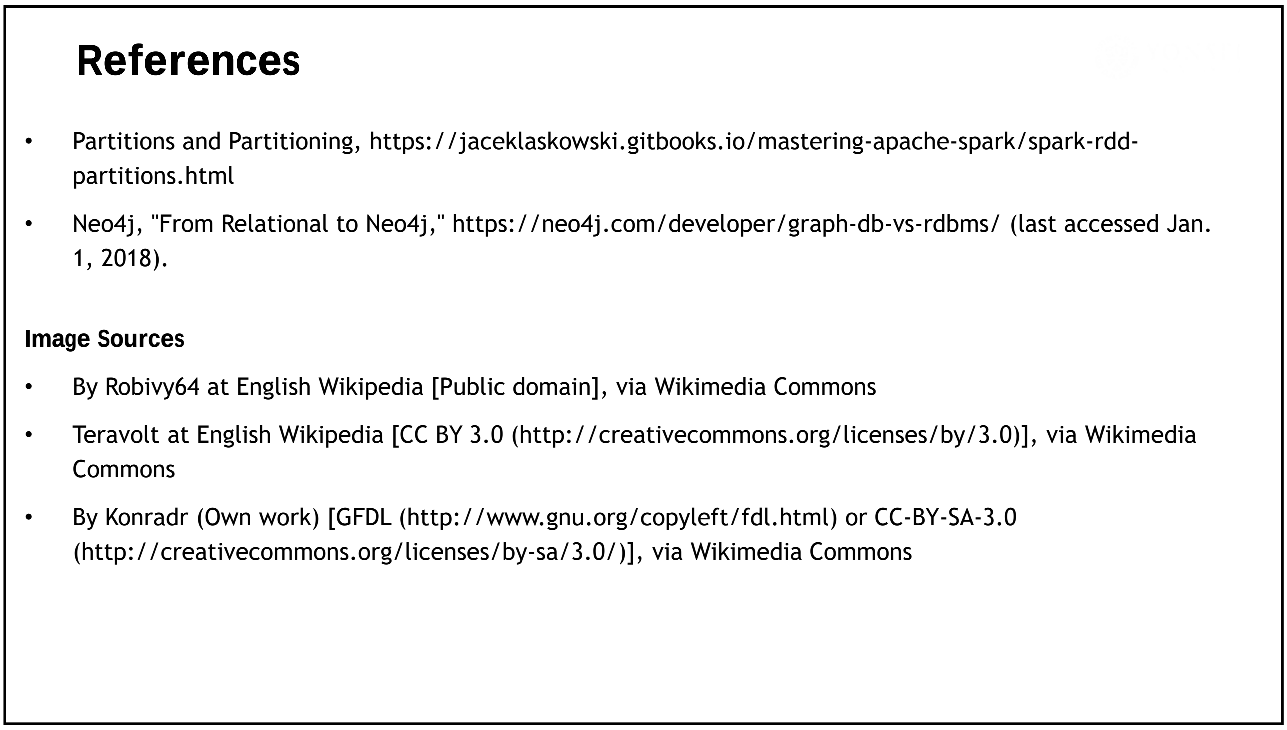
  Yes! neo4j

Big Data

References

References

- Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*. 1st Edition. O'Reilly, 2015.
- Sameer Farooqui, Databricks, **Advanced Apache Spark Training**, Devops Advanced Class, Spark Summit East 2015, http://slideshare.net/databricks, www.linkedin.com/in/blueplastic, March 2015.
- Apache Spark documents (all documents and tutorials were used)
  - http://spark.apache.org/docs/latest/rdd-programming-guide.html
  - http://spark.apache.org/docs/latest/rdd-programming-guide.html#working-with-key-value-pairs
  - https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html#rdd-persistence
- Wikipedia, www.wikipedia.org
- Stackoverflow, https://stackoverflow.com/questions
- Bernard Marr, "Spark Or Hadoop -- Which Is The Best Big Data Framework?," Forbes, Tech, June 22, 2015.
- Quick introduction to Apache Spark, https://www.youtube.com/watch?v=TgiBvKcGL24
- Wide vs Narrow Dependencies, https://github.com/rohgar/scala-spark-4/wiki/Wide-vs-Narrow-Dependencies



References

- Partitions and Partitioning, https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-rdd-partitions.html
- Neo4j, "From Relational to Neo4j," https://neo4j.com/developer/graph-db-vs-rdbms/ (last accessed Jan. 1, 2018).

Image Sources

- By Robivy64 at English Wikipedia [Public domain], via Wikimedia Commons
- Teravolt at English Wikipedia [CC BY 3.0 (http://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons
- By Konradr (Own work) [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/)], via Wikimedia Commons