Rakamin Academy

id/x partners

Project Based Internship

Big Data and Data Warehouse

Data Management

Daftar Isi

Introduction to Big Data

Big data refers to extremely large and complex datasets that cannot be easily managed, processed, or analyzed using traditional data processing tools. The term "big data" encompasses not only the size of the data but also its variety, velocity, and complexity. The analysis of big data involves extracting meaningful insights and patterns from these vast and diverse datasets to inform decision-making, predictions, and other business strategies.

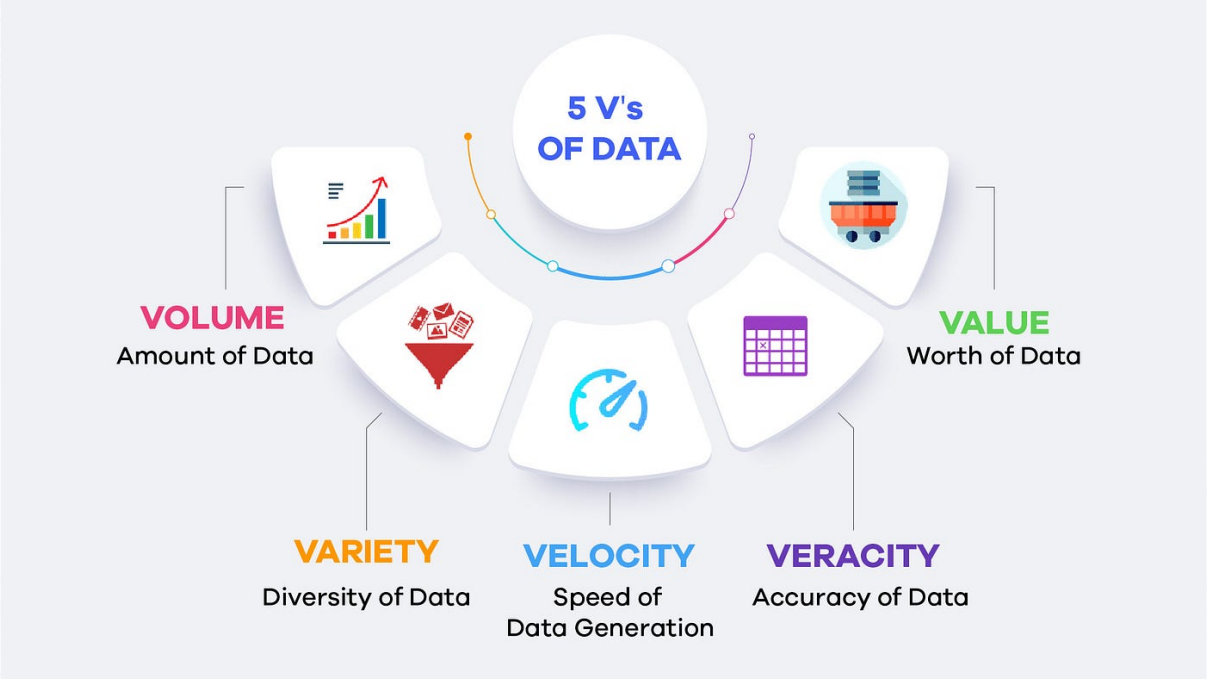

Figure 1. 5V of Big Data

There are three main characteristics that define big data, often referred to as the "5Vs":

- Volume: Big data involves a massive amount of information. This could be terabytes, petabytes, or even exabytes of data, far beyond the capacity of traditional databases to handle efficiently.
- Velocity: The speed at which data is generated, collected, and processed is crucial. With the advent of technologies such as the Internet of Things (IoT) and social media, data is generated at an unprecedented speed, and organizations need to process and analyze it in real-time or near-real-time.
- Variety: Big data comes in various formats and types, including structured, semi-structured, and unstructured data. Traditional databases are typically designed for structured data, like relational databases with tables and rows. However, big data includes a wide range of data types, such as text, images, videos, social media posts, sensor data, and more.
- Value: Refers to value of the data
- Veracity: This refers to the quality and accuracy of the data. In big data, there may be issues with the reliability and trustworthiness of the information, and managing data quality becomes a significant concern.

To handle big data, organizations often turn to advanced technologies and tools, including distributed computing frameworks like Apache Hadoop and Apache Spark. These technologies enable the parallel processing of large datasets across multiple servers, making it possible to handle the volume and velocity of big data. Additionally, machine learning and data analytics techniques are employed to extract valuable insights from the data.

Big data analytics has applications in various fields, including business, healthcare, finance, marketing, and scientific research, among others. It has the potential to

revolutionize decision-making processes and provide a competitive advantage to organizations that harness its power effectively.

Big Data Implementation

Here are some examples of big data implementation in various business sectors and an explanation of how they work:

1. Retail: Customer Analytics and Personalization

How it works:

- Data Collection: Retailers collect vast amounts of data from various sources, including point-of-sale transactions, online purchases, customer interactions, and social media.
- Customer Segmentation: Big data analytics is used to segment customers based on their preferences, behaviors, and demographics.
- Recommendation Engines: Machine learning algorithms analyze customer behavior to provide personalized product recommendations, improving cross-selling and upselling opportunities.
- Inventory Management: Predictive analytics helps optimize inventory levels by forecasting demand, reducing excess stock, and avoiding stock outs.

Benefits:

- Enhanced customer experience through personalized recommendations.
- Improved inventory management and reduced costs.
- Increased sales and customer loyalty.

1. Finance: Fraud Detection and Risk Management

How it works:

- Transaction Monitoring: Big data analytics is applied to real-time transaction data to identify patterns and anomalies that may indicate fraudulent activity.
- Machine Learning Models: Predictive models analyze historical data to detect unusual behavior and patterns associated with fraud.
- Credit Scoring: Big data is used to analyze a wide range of data points to assess credit risk and determine credit scores for individuals and businesses.
- Market Analysis: Financial institutions use big data analytics for real-time market analysis and to assess overall economic risk.

Benefits:

- Improved fraud detection and prevention.
- Enhanced risk management and compliance.
- More accurate credit scoring.

2. Manufacturing: Predictive Maintenance

How it works:

- IoT Sensors: Sensors on manufacturing equipment collect real-time data on machine performance, temperature, vibration, and other relevant factors.
- Predictive Analytics: Machine learning algorithms analyze historical and real-time data to predict when equipment is likely to fail.
- Maintenance Alerts: Proactive maintenance alerts are generated, allowing organizations to schedule maintenance activities before a failure occurs.

- Downtime Reduction: Predictive maintenance minimizes unplanned downtime and extends the lifespan of equipment.

Benefits:

- Increased operational efficiency and reduced downtime.
- Cost savings through optimized maintenance schedules.
- Improved equipment reliability and lifespan.

3. Telecommunications: Network Optimization and Customer Experience

How it works:

- Network Performance Monitoring: Big data analytics processes data from network devices, including switches, routers, and cell towers, to monitor network performance.
- Predictive Analytics: Machine learning models predict potential network issues before they occur, allowing for proactive optimization.
- Customer Experience Analysis: Analyzing customer usage patterns and feedback to improve service quality and offer personalized plans.
- Churn Prediction: Big data is used to identify factors leading to customer churn, enabling targeted retention strategies.

Benefits:

- Enhanced network reliability and performance.
- Improved customer satisfaction and retention.
- Proactive issue resolution and optimized network infrastructure.

These examples illustrate how big data implementation can provide actionable insights, enhance decision-making processes, and drive positive outcomes across various industries. The ability to harness and analyze large volumes of data opens up new opportunities for innovation, efficiency, and competitiveness in the business landscape.

Data Warehouse

Data Warehouse is basically the process of collecting, storing, and managing data from various heterogeneous sources. It is the main component of the business intelligence system where analysis and management of data are done which is further used to improve decision making. It involves the process of extraction, loading, and transformation for providing the data for analysis. Data warehouses are also used to perform queries on a large amount of data. It uses data from various relational databases and application log files.

Fact Table and Dimension Table

In a data warehouse, the terms "Fact Table" and "Dimension Table" refer to two essential types of tables that are used to organize and store data in a way that facilitates efficient querying and analysis. These concepts are central to the design of a star schema or snowflake schema, which are common data warehouse architectures.

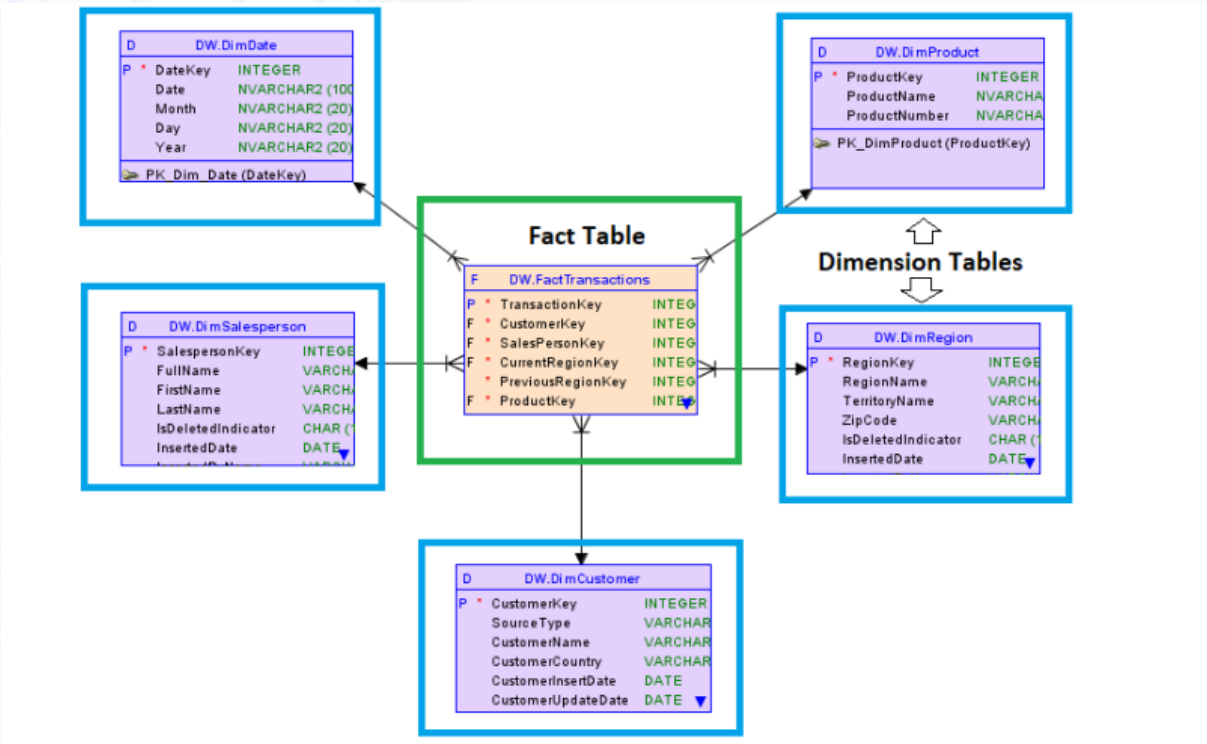

Figure 2. Relationship between Fact Table and Dimension Table

1. Fact Table:

Definition: A Fact Table is a large, central table in a data warehouse that stores quantitative information, typically business metrics or measures, which can be analyzed over time or across different dimensions.

Characteristics:

- Numeric Values: Fact Tables primarily contain numerical, additive data, such as sales revenue, quantity sold, or profit.
- Foreign Keys: Fact Tables include foreign keys that link to the primary keys of related Dimension Tables.
- Granularity: Fact Tables are often at a fine level of granularity, capturing detailed data at the transactional or event level.

- Time Periods: Fact Tables usually include a time dimension to support time-based analysis.

SALES Fact table

Calendar Key	Store Key	Product Key	Customer Key	TID	TimeOfDay	Dollars Sold	Units Sold
1	1	1	1	T111	8:23:59 AM	$100	1
1	2	2	2	T222	8:24:30 AM	$70	1
2	3	3	1	T333	8:15:08 AM	$75	5
2	3	1	1	T333	8:15:08 AM	$100	1
2	3	4	3	T444	8:20:33 AM	$90	1
2	3	2	3	T444	8:20:33 AM	$140	2
2	3	4	2	T555	8:30:00 AM	$360	4
2	3	5	2	T555	8:30:00 AM	$300	2
2	3	6	2	T555	8:30:00 AM	$250	1
2	3	5	2	T666	9:30:00 AM	$300	2
2	3	6	2	T666	9:30:00 AM	$250	1

Figure 3. Fact Table Example

Example: In a figure above, a Fact Table might include columns such as "Dollars Sold," "Units Sold", with foreign keys linking to Dimension Tables like "Product Key," and "Customer, Key".

2. Dimension Table:

Definition: A Dimension Table is a table in a data warehouse that stores descriptive attributes or dimensions related to the data stored in the Fact Table. Dimension Tables provide context and help in organizing data into a meaningful structure.

Characteristics:

- Descriptive Attributes: Dimension Tables contain textual or categorical data that describes the characteristics of the data in the Fact Table.
- Primary Keys: Each Dimension Table typically has a primary key, and foreign keys in the Fact Table reference these primary keys.
- Hierarchies: Dimension Tables often have hierarchies, such as product categories, subcategories, and individual products.

Product ID	Product Name	Category	Sub-Category	Brand	Price
1555	Chair	Furniture	Household	ABC	1000

Figure 4. Dim Table Example

Example: In the retail data warehouse example, Dimension Tables might include "Product ID" (with attributes like product name, category), "Brabd" (with attributes like price).

In summary, the Fact Table contains quantitative data that can be aggregated, and Dimension Tables provide descriptive context to that data. Together, they form a star schema or snowflake schema, which is a widely used design pattern in data warehousing for efficient and flexible querying and analysis.

Principles of Schema

In the context of a data warehouse, a schema refers to the organization or structure of the database, which includes the arrangement of tables, relationships, and the definition of fields. There are two main types of schemas in data warehousing: star schema and snowflake schema. The principles of these schemas play a crucial role

in designing an effective and efficient data warehouse. Here's an overview of the principles for each:

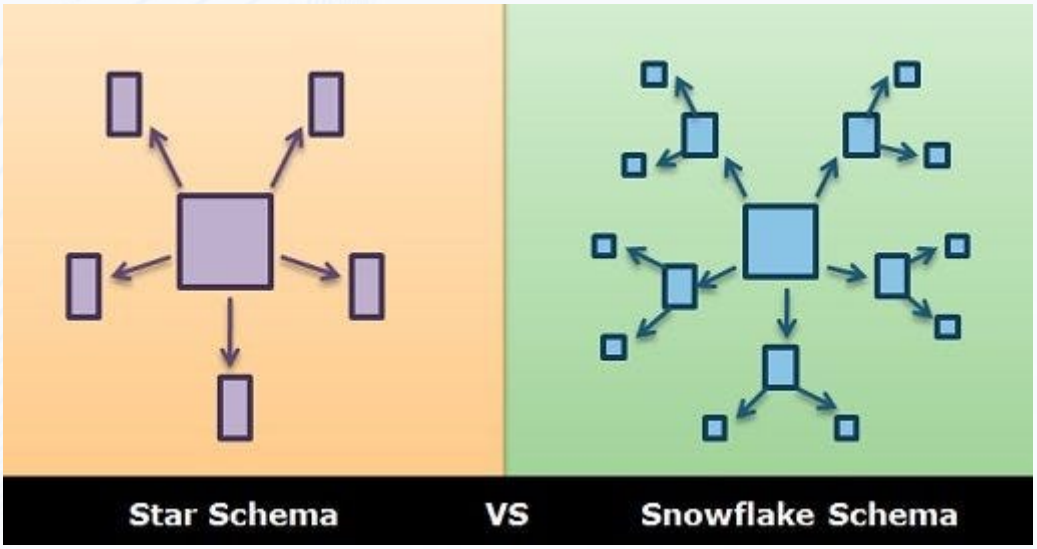

Figure 5. Star Schema vs Snowflake Schema

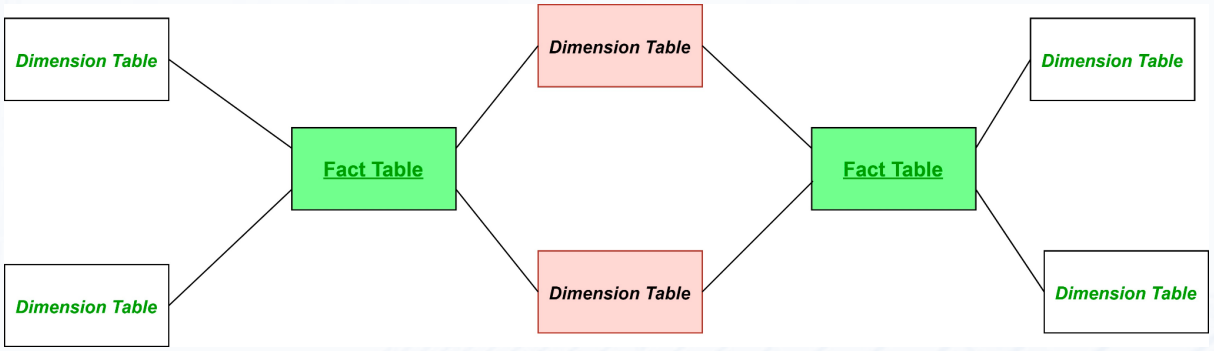

Figure 6. Fact Constellation Schema

1. Star Schema:

- Description: In a star schema, a central fact table is connected to one or more dimension tables through foreign key relationships. The fact table contains quantitative data (facts), while dimension tables store descriptive information related to the facts. The schema resembles a star when visualized, with the fact table at the center and dimension tables surrounding it.

- Advantages:
    - Simplicity and ease of understanding.
    - Fast query performance for analytical queries.
    - Redundancy is minimized.
- Disadvantages:
    - Denormalization can lead to data redundancy.
    - Maintenance can be more challenging if there are changes to the schema.

2. Snowflake Schema:

- Description: The snowflake schema is an extension of the star schema where dimension tables are normalized, meaning they are split into sub-dimensions or related tables. This normalization reduces redundancy but increases the number of joins needed to retrieve data compared to a star schema.
- Advantages:
    - Reduction in data redundancy and storage space.
    - Easier maintenance of dimension tables.
- Disadvantages:
    - Increased complexity due to more joins.
    - Potentially slower query performance compared to star schema.

3. Fact Constellation Schema (Galaxy Schema):

- Description: Fact constellation schema involves multiple fact tables sharing dimension tables. This approach is used when there are multiple business

processes or areas, and each is represented by its own fact table. The fact tables are connected through shared dimension tables.

- Advantages:
    - Supports complex business scenarios and relationships.
    - Allows for more flexibility in modeling diverse business processes.
- Disadvantages:
    - Increased complexity in schema design and query formulation.
    - Potential challenges in maintaining consistency across different fact tables.

In practice, the choice between a star schema and a snowflake schema depends on the specific requirements of the data warehouse, including the nature of queries, the volume of data, and the balance between query performance and storage efficiency. Both schemas have their advantages and trade-offs, and the selection should align with the goals and characteristics of the particular data warehouse implementation.

Principles of Massively Parallel Processing Databases

Massively Parallel Processing (MPP) databases are designed to handle large volumes of data by distributing the processing workload across multiple nodes or servers. The principles of Massively Parallel Processing databases are centered around achieving high performance, scalability, and efficiency in processing and analyzing large datasets. Here are some key principles:

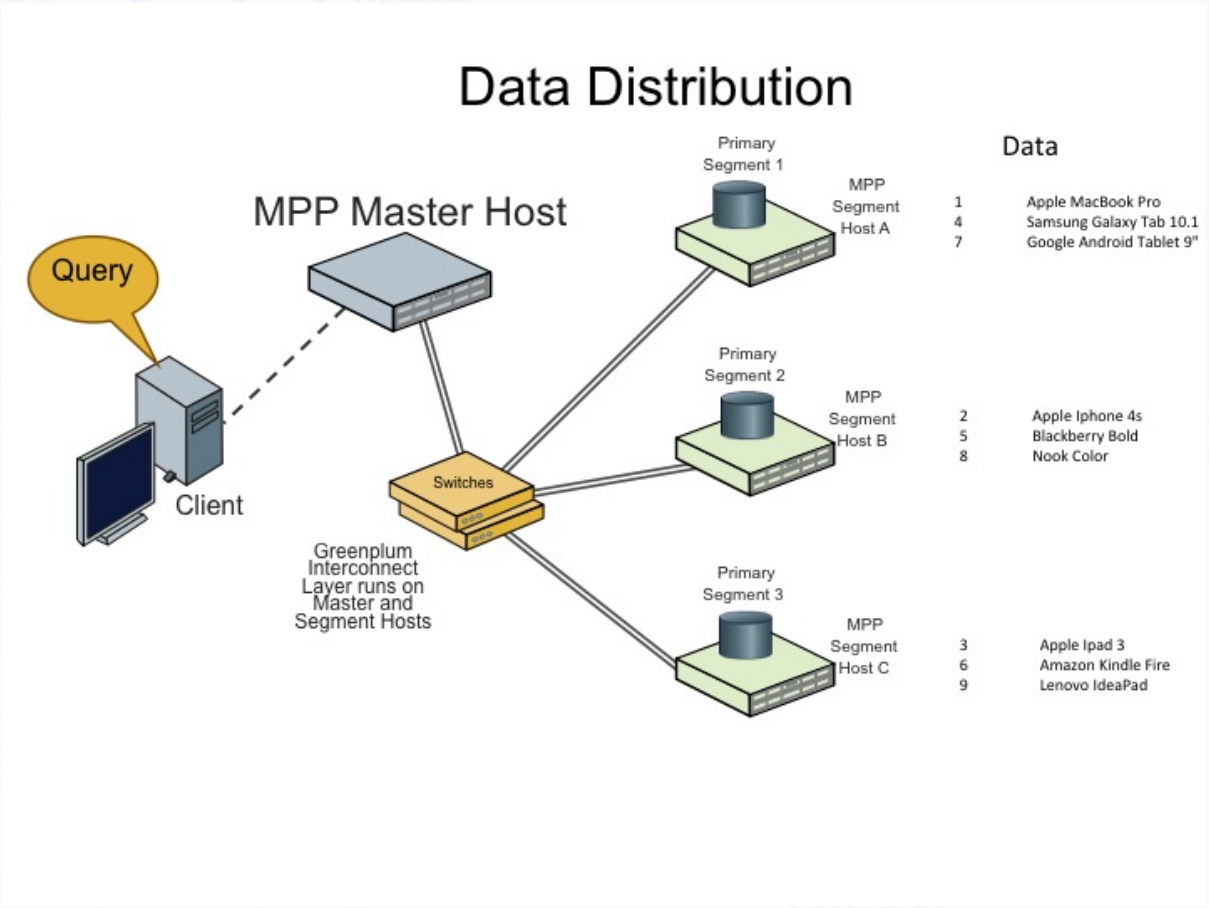

Figure 4. Data Distribution

- Parallel Processing:
    - Distributed Architecture: MPP databases distribute data and processing tasks across multiple nodes or servers in a cluster. Each node operates independently and processes a portion of the data concurrently.
    - Parallel Execution: Queries and data processing tasks are divided into smaller units of work that can be executed in parallel across multiple nodes, enabling faster query performance.
- Shared-Nothing Architecture:
    - Isolated Processing: In MPP databases, each node in the cluster operates independently and has its own memory and storage. Nodes

do not share main memory or storage resources, reducing contention and bottlenecks.

- Data Partitioning: Data is partitioned and distributed among nodes, ensuring that each node is responsible for a specific subset of the data. This approach enhances parallelism and minimizes data movement between nodes during processing.

- Horizontal Scalability:
  - Elasticity: MPP databases are designed to scale horizontally by adding more nodes to the cluster as data volume and processing demands increase. This enables organizations to adapt to changing workloads and data growth.
  - Load Balancing: The system automatically balances the workload among nodes to ensure efficient resource utilization and prevent performance bottlenecks.
- Data Distribution:
  - Even Data Distribution: MPP databases distribute data evenly across nodes to prevent any single node from becoming a performance bottleneck. Balanced data distribution ensures that each node has a similar processing load.
  - Hash-Based or Range-Based Distribution: Data can be distributed based on hash values or ranges of values to achieve an even distribution and efficient querying.
- Data Compression and Storage Optimization:
  - Columnar Storage: MPP databases often use columnar storage formats, where data for each column is stored together. This can

improve query performance by allowing the database to read only the columns needed for a query.

- Data Compression: Compression techniques are applied to reduce storage requirements and improve data transfer efficiency between nodes.

- Fault Tolerance:
  - Redundancy: MPP databases typically include mechanisms for data redundancy and fault tolerance. Data is often replicated across nodes to ensure that the system can recover from node failures without data loss.
  - Automated Failover: The system is designed to automatically detect and handle node failures, redirecting queries to healthy nodes and maintaining overall system availability.
- Concurrency Control:
  - Distributed Transactions: MPP databases implement distributed concurrency control mechanisms to ensure the consistency of transactions across multiple nodes. This includes strategies for handling transactions that span multiple nodes in a distributed environment.

These principles collectively contribute to the effectiveness of MPP databases in handling large-scale data processing and analytics tasks, making them well-suited for use cases where high performance and scalability are critical, such as in data warehouses and analytical platforms.

Use Case

Background and Problem Statement:

You are a data scientist at ID/X Partners carrying out a data science project related to personalizing recommendations for users. After looking at the data that will be used, it turns out that the data is included in Big Data. So in carrying out processing special techniques need to be carried out.

Explain Big Data techniques that refer to MPP!

Solution:

In the context of Massively Parallel Processing (MPP) for Big Data in a data science project at ID/X Partners focused on personalizing recommendations, several techniques can be applied to leverage the benefits of MPP databases. MPP databases are designed to handle large volumes of data by distributing the processing workload across multiple nodes or servers. Here are some key techniques that can be employed:

Data Distribution:

- Hash-Based or Range-Based Sharding: Distribute data across nodes based on hash values or ranges of values. This helps in achieving an even distribution of data and workload across the MPP cluster.
- Distribution Key Selection: Choose appropriate distribution keys to minimize data movement during query processing. Effective distribution keys are crucial for optimizing parallel query execution.

Parallel Query Execution:

- Query Parallelization: Break down analytical queries into smaller tasks that can be executed in parallel across multiple nodes. MPP databases excel in parallel processing, allowing for the concurrent execution of tasks and faster query response times.
- Query Optimization: Leverage the MPP database's query optimizer to generate efficient execution plans that take advantage of parallel processing capabilities.

Fault Tolerance:

- Redundancy and Data Replication: Configure the MPP database for fault tolerance by replicating critical data across nodes. This redundancy helps the system recover gracefully from node failures without compromising data integrity.
- Automated Failover: Implement automated failover mechanisms to redirect queries to healthy nodes in the event of a node failure.

Concurrency Control:

- Distributed Transactions: Implement concurrency control mechanisms to manage transactions that span multiple nodes. This involves ensuring consistency, isolation, and atomicity of transactions in a distributed environment.

References

https://www.guru99.com/what-is-big-data.html

https://medium.com/@get_excelsior/big-data-explained-the-5v-s-of-data-ae80cbe8ded1

https://www.oracle.com/database/what-is-a-data-warehouse/

https://www.analyticsvidhya.com/blog/2021/07/a-brief-introduction-to-data-warehouse/#:~:text=A%20data%20warehouse%20is%20mainly,large%20amounts%20of%20historical%20data.

https://www.analyticsvidhya.com/blog/2023/08/difference-between-fact-table-and-dimension-table/

https://nidhig631.medium.com/star-schema-vs-snowflake-schema-78dc9424a8a2

https://dwarehouse.wordpress.com/2012/12/28/introduction-to-massively-parallel-processing-mpp-database/