UNIT –V

STORAGE, INDEXING, QUERY PROCESSING, AND EMERGING TRENDS

Difference between SQL and NoSQL

Choosing between **SQL** (Structured Query Language) and **NoSQL** (Not Only SQL) databases is a critical decision for **developers**, **data engineers**, and organizations looking to handle large datasets effectively. Both **database types** have their **strengths** and **weaknesses**, and understanding the key differences can help us make an informed decision based on our project's needs.

Differences Between SQL and NoSQL

Aspect	SQL (Relational)	NoSQL (Non-relational)
Data Structure	Tables with rows and columns	Document-based, key-value, column-family, or graph-based
Schema	Fixed schema (predefined structure)	Flexible schema (dynamic and adaptable)
Scalability	Vertically scalable (upgrading hardware)	Horizontally scalable (adding more servers)
Data Integrity	ACID-compliant (strong consistency)	BASE-compliant (more available, less consistent)
Query Language	SQL (Structured Query Language)	Varies (e.g., MongoDB uses its own query language)
Performance	Efficient for complex queries and transactions	Better for large-scale data and fast read/write operations
Use Case	Best for transactional systems (banking, ERP, etc.)	Ideal for big data, real-time web apps, and data lakes

Aspect	SQL (Relational)	NoSQL (Non-relational)
Examples	MySQL, PostgreSQL, Oracle, MS SQL Server	MongoDB, Cassandra, CouchDB, Neo4j

1. Type

SQL databases are primarily called Relational Databases (RDBMS).

whereas NoSQL databases are primarily called non-relational or distributed databases.

2. Language

SQL databases define and manipulate data-based structured query language (SQL). Seeing from a side this language is extremely powerful. SQL is one of the most **versatile** and **widely-used options** available which makes it a **safe choice**, especially for great **complex queries**. But from another side, it can be restrictive.

SQL requires you to use predefined schemas to determine the structure of your data before you work with it. Also, all of our data must follow the same structure. This can require significant **up-front preparation** which means that a change in the structure would be both difficult and disruptive to your whole system.

3. Scalability

In almost all situations SQL databases are vertically scalable. This means that you can increase the load on a single server by increasing things like RAM, CPU, or SSD. But on the other hand, NoSQL databases are **horizontally scalable**. This means that you handle more traffic by **sharing**, or adding more servers in your **NoSQL database**.

It is similar to adding more floors to the same building versus **adding more buildings** to the neighborhood. Thus NoSQL can ultimately become larger and more powerful, making these databases the preferred choice for large or ever-changing data sets.

4. Structure

SQL databases are table-based on the other hand **NoSQL databases** are either **key-value pairs**, **document-based**, **graph databases**, or **wide-column stores**. This makes relational SQL databases a better option for applications that require **multi-row transactions** such as an accounting system or for legacy systems that were built for a relational structure.

Here is a simple example of how a structured data with rows and columns vs a non-structured data without definition might look like. A product table in SQL **db** might accept data looking like this:

```
{
"id": "101",
"category":"food"
"name":"Apples",
"qty":"150"
}
```

Whereas a unstructured NOSQL DB might save the products in many variations without constraints to change the underlying table structure

```
Products=[
{
"id":"101:",
category":"food",
"name":"California Apples"
,"qty":"150"},
```

```
{"id":"102,
"category":"electronics"
"name":"Apple MacBook Air",
"qty":"10",
"specifications":{   "storage":"256GB SSD",
 "cpu":"8 Core",
 "camera": "1080p FaceTime HD camera"  }
}
]
```

5. Property followed

SQL databases follow ACID properties (Atomicity, Consistency, Isolation, and Durability) whereas the NoSQL database follows the Brewers CAP theorem (Consistency, Availability, and Partition tolerance).

6. Support

Great support is available for all **SQL databases** from their vendors. Also, a lot of independent consultants are there who can help you with SQL databases for very large-scale deployments but for some **NoSQL databases** you still have to rely on community support and only limited outside experts are available for setting up and deploying your **large-scale NoSQL deploy**.

Function of SQL

SQL databases, also known as **Relational Database Management Systems (RDBMS)**, use structured tables to store data. They rely on a **predefined schema** that determines the organization of data within tables, making them suitable for applications that require a fixed, consistent structure.

- **Structured Data**: Data is organized in tables with rows and columns, making it easy to relate different types of information.
- **ACID Compliance**: SQL databases follow the ACID properties (Atomicity, Consistency, Isolation, Durability) to ensure reliable transactions and data integrity.
- **Examples**: Popular SQL databases include **MySQL**, **PostgreSQL**, **Oracle**, and **MS SQL Server**.

Function Of NoSql

NoSQL databases, on the other hand, are designed to handle **unstructured or semi-structured data**. Unlike SQL databases, NoSQL offers **dynamic schemas** that allow for more flexible data storage, making them ideal for handling massive volumes of data from various sources.

- **Flexible Schema**: NoSQL databases allow the storage of data without a predefined structure, making them more adaptable to changing data requirements.
- **CAP Theorem**: NoSQL databases are designed based on the **CAP theorem** (Consistency, Availability, Partition Tolerance), which prioritizes availability and partition tolerance over strict consistency.
- **Examples**: Well-known NoSQL databases include **MongoDB**, **Cassandra**, **CouchDB**, and **HBase**.

SQL vs NoSQL: Which is Faster?

- **SQL Databases**: Generally, SQL databases perform well for **complex queries**, structured data, and systems requiring **data consistency** and **integrity**. However, as the volume of data grows, they may struggle with **scalability** and may require significant infrastructure upgrades.
- **NoSQL Databases**: NoSQL databases excel in scenarios that demand **high performance** and **scalability**. Because of their **horizontal scalability** (accommodating more servers), they handle large amounts of data and high-velocity workloads better. For instance, MongoDB or Cassandra is a common choice when dealing with big data or applications with high traffic.

When to Choose SQL?

SQL databases are well-suited for use cases where:

- **Data consistency** and **transactional integrity** are critical (e.g., banking systems, customer relationship management).
- The application needs a **well-defined schema** and structured data.
- Complex queries and **relational data** are involved.
- Applications requiring **multi-row transactions** (such as inventory management) benefit from SQL's robust features.

When to Choose NoSQL?

NoSQL databases are a better choice when:

- You need to handle **large, unstructured data** sets, like social media data or logs.
- The application requires **horizontal scalability** to accommodate high traffic and big data.
- There is a need for **real-time data processing** and **flexible data models** (e.g., a content management system).
- You are dealing with applications requiring **frequent changes in data structures**.

Introduction to BigQuery

Google BigQuery is a fully managed, serverless data warehouse designed to help businesses store and analyze large volumes of data quickly and efficiently. Whether you're dealing with massive datasets or real-time analytics, BigQuery allows you to run complex queries and get insights in seconds without having to worry about managing servers or infrastructure. As part of **Google Cloud**, BigQuery integrates seamlessly with other Google Cloud services, such as **Google Cloud Storage**, **Google Analytics**, and **Google Machine Learning**, enabling you to build comprehensive data solutions.

BigQuery makes it easy to analyze structured and unstructured data, offering scalable, fast, and secure data processing with minimal setup. It's especially useful for companies in industries like e-commerce, healthcare, and finance, where data-driven decisions can drive significant business growth. With BigQuery, businesses can unlock the power of their data, gain valuable insights, and make informed decisions quickly, all while keeping costs low.

In this series, we'll look into how BigQuery can help you get valuable insights from your data with ease. If your business has small amounts of data, you might be able to store it in a spreadsheet. But as your amount of data grows to gigabytes, terabytes, or even petabytes, you start to need a more efficient system like a **data warehouse**. That's because all that data isn't very useful unless you have a way to analyze it. Traditionally, larger sets of data mean longer times between asking your questions and getting answers.

The Need for a Data Warehouse

As your data grows, it becomes more challenging to derive meaningful insights without a robust solution. Traditional data management systems struggle with large datasets, leading to longer processing times between asking questions and receiving answers. This is where BigQuery comes in.

BigQuery is designed to handle massive datasets, such as log data from thousands of retail systems or IoT data from millions of vehicle sensors across the globe. It is a fully managed and serverless data warehouse that allows you to focus on analytics rather than managing infrastructure.

Avoiding the Data Silo Problem

One of the key benefits of BigQuery is its ability to avoid the "Data Silo" problem. This issue occurs when different teams in your company have independent data marts, which can create friction when analyzing data across teams and pose challenges for data version control.

Thanks to its integration with Google Cloud's native identity and access management, BigQuery allows you to assign read or write permissions to specific users, groups, or projects. This ensures that your sensitive data remains secure while still enabling collaboration across teams.

Key Components of Working with Data in BigQuery

BigQuery simplifies working with data through three primary steps:

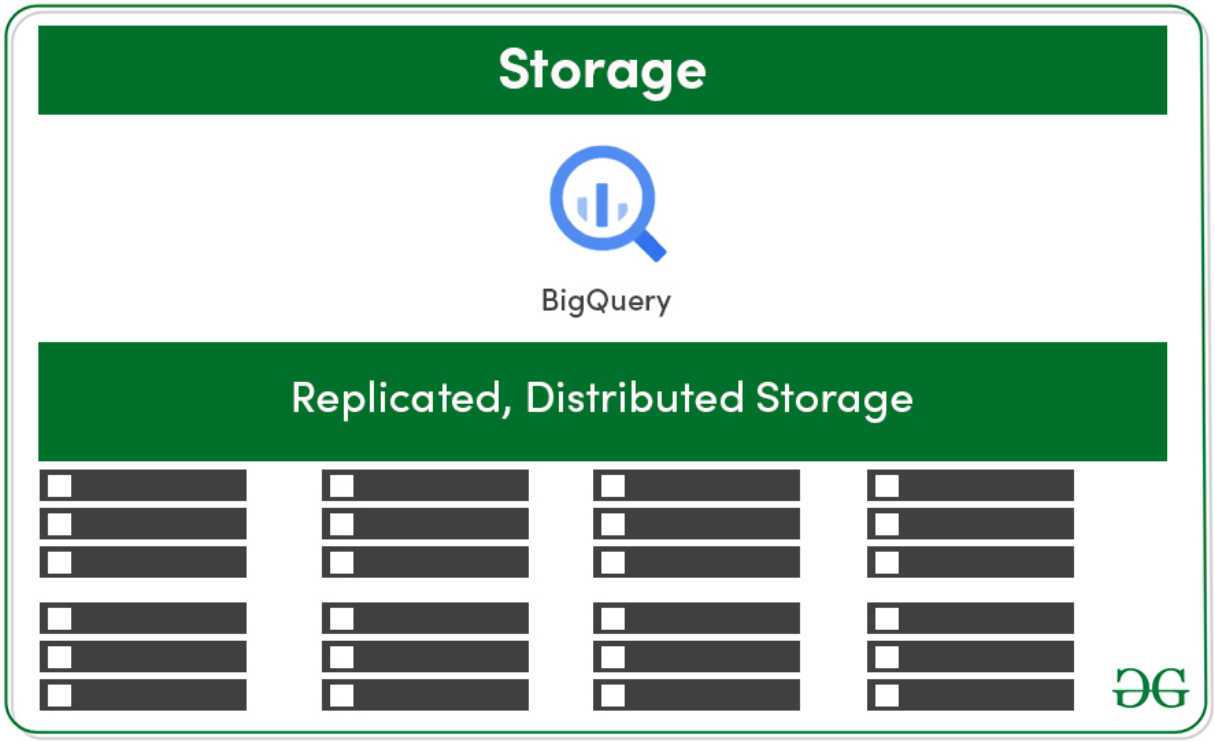
- **Storage**
- **Ingestion**
- **Querying**

Google handles running everything else. BigQuery is a fully managed service, which means you don't need to set up or install anything. And you don't require a database administrator. You can simply log into your Google Cloud project from a browser and get started.

1. Storage

Data in BigQuery is stored in structured tables. This allows you to use standard SQL for easy querying and data analysis. For instance, if you have sales data for each of your stores over the last year, a smaller database might be sufficient. However, when dealing with thousands of stores and needing to break down revenue by product type, region, or time period, BigQuery shines.

BigQuery automatically manages storage and scaling for you. As your data grows, BigQuery adjusts to handle it, ensuring that you don’t need to worry about storage limitations.

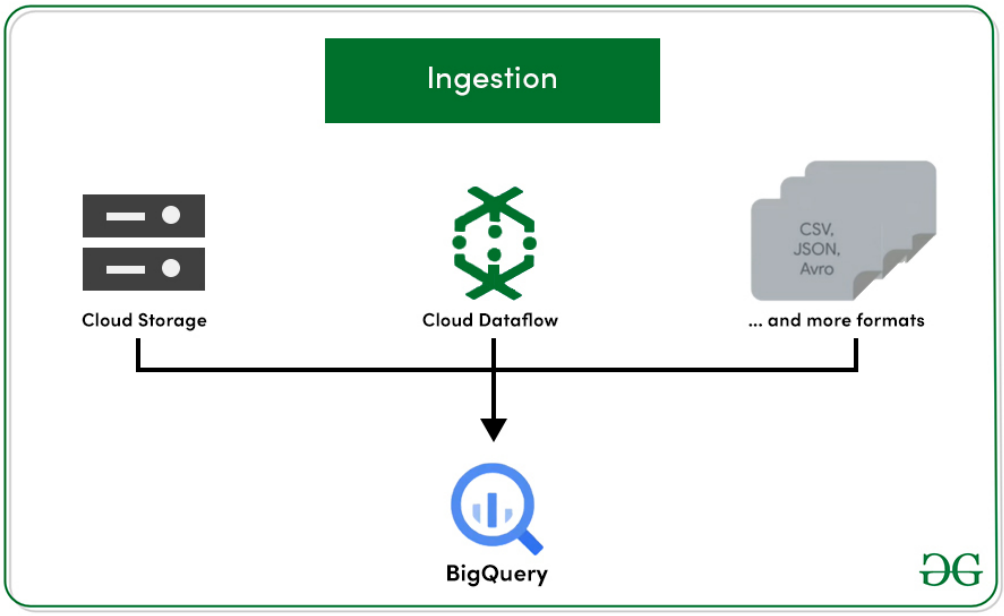


2. Ingestion

Once your data is ready, it needs to be ingested into BigQuery. There are various ways to do this:

- **Upload from Cloud Storage**: You can upload data directly from Cloud Storage.
- **Stream from Cloud Dataflow**: You can stream data into BigQuery from other sources.
- **ETL Pipeline with Cloud Data Fusion**: You can build an ETL pipeline to extract, transform, and load your data into BigQuery.

Additionally, BigQuery supports importing data from a variety of file formats, such as CSV, JSON, and Avro.



3. Querying

Once your data is in BigQuery, you're ready to start querying it. BigQuery supports SQL, so if you are familiar with ANSI-compliant relational databases, you can easily write queries to analyze your data.

Public Datasets: No Ingestion or Storage Required

If you want to skip the ingestion and storage steps, you can begin analyzing BigQuery's public datasets. These third-party datasets are publicly available for anyone to query, and Google manages all the storage. This allows you to focus on deriving insights without worrying about managing data or infrastructure.

BigQuery offers a powerful, efficient solution for analyzing large datasets, whether you're working with your own data or exploring public datasets. By eliminating the complexity of infrastructure management, BigQuery allows businesses to quickly gain valuable insights from their data.

Key Features of Google BigQuery

Google BigQuery is a powerful and scalable data warehouse that offers several key features designed to help businesses manage and analyze large datasets quickly and efficiently. Here are some of its most important features:

1. Serverless Architecture

BigQuery is a serverless platform, which means you don't need to worry about managing servers or infrastructure. It automatically scales based on your needs, so you can focus on analyzing data instead of managing hardware.

2. Fast and Scalable Analytics

BigQuery is built for high-speed data processing, allowing you to run complex queries over large datasets in seconds. Whether you're working with terabytes or petabytes of data, BigQuery scales to handle it efficiently.

3. Real-Time Data Analysis

BigQuery allows you to run real-time analytics, so you can get instant insights from your data as it's updated. This is crucial for businesses that need to make decisions quickly based on the most current data.

4. SQL-Based Queries

BigQuery uses SQL (Structured Query Language), the industry-standard language for querying databases. This makes it easy for users familiar with SQL to start analyzing their data without needing to learn a new language.

5. Cost-Efficient

BigQuery offers a pay-as-you-go model, where you only pay for the data you store and the queries you run. This ensures you only pay for what you use, keeping costs low while still providing powerful analytics.

6. Integration with Google Cloud Services

BigQuery seamlessly integrates with other Google Cloud services, such as Google Cloud Storage, Google Analytics, and Google Data Studio, allowing you to easily connect, store, and visualize data from multiple sources.

7. Machine Learning Integration

BigQuery provides built-in machine learning capabilities (BigQuery ML), allowing users to run machine learning models directly in BigQuery without needing to move data to other tools. This simplifies the process of building and training models for predictive analytics.

8. Security and Compliance

BigQuery offers robust security features, including encryption for data at rest and in transit, as well as compliance with major industry standards like GDPR, HIPAA, and PCI DSS, ensuring your data is safe and compliant.

9. Easy Data Sharing

BigQuery makes it easy to share data across teams or with external partners. You can control access with detailed permissions and make your data available to others without moving or copying it.

10. Data Visualization and Reporting

BigQuery integrates with Google Data Studio and other visualization tools, allowing you to turn your data insights into visual reports and dashboards. This makes it easier to communicate findings and make data-driven decisions across your organization.

Conclusion

Google BigQuery is an invaluable tool for businesses looking to store and analyze vast amounts of data quickly and efficiently. With its serverless architecture, real-time analytics, and integration with other Google Cloud services, BigQuery empowers companies to unlock the full potential of their data. By handling everything from storage to querying, BigQuery eliminates the need for manual infrastructure management, allowing businesses to focus on gaining insights and making data-driven decisions. Whether you're dealing with structured or unstructured data, BigQuery's scalability, cost-efficiency, and machine learning capabilities make it an ideal choice for industries like e-commerce, healthcare, finance, and more. By leveraging BigQuery, businesses can stay competitive, innovate faster, and grow smarter.