SQOOP INTRODUCTION

➢ What is Sqoop?

- Sqoop is a data transfer mechanism that connects Hadoop and relational database servers. This is used to import data from relational databases like MySQL and Oracle into Hadoop HDFS, as well as export data from Hadoop HDFS to relational databases.

- It enables incremental loads of a single table or a free-form SQL query, as well as saved jobs that can be executed many times to import database updates made since the previous import. Data can be migrated to MySQL/PostgreSQL/Oracle/SQL Server/DB2 to HDFS/hive/hbase using Sqoop, and vice versa.

Why sqoop?

Apache Sqoop is a tool for transferring massive amounts of data between Hadoop and structured datastores like relational databases. It allows specific processes, such as ETL processing, to be offloaded from an enterprise data warehouse to Hadoop for more efficient execution at a lesser cost. Data can also be extracted from Hadoop and exported to external structured datastores using Sqoop.

➢ Sqoop Features:

1. Apache Sqoop is a very resilient system. It features community support and participation, as well as being simple to use.
2. With Sqoop, we can load an entire table with only one command. Sqoop also allows us to load all of the database's tables with a single command.
3. Sqoop has the ability to load data incrementally. We can load sections of the table using Sqoop whenever it is updated.
4. For importing and exporting data, Apache Sqoop uses the YARN framework. On top of the parallelism, this gives fault tolerance.
5. We can use the deflate(gzip) technique with the –compress argument or specify the –compression-codec argument to compress our data. In Apache Hive, we can load a compressed table.

6. Sqoop provides connectors for a wide range of RDBMS databases, covering practically the whole globe.
7. Kerberos is a computer network authentication system that uses 'tickets' to allow nodes communicating over an insecure network to confirm their identity to one another. Kerberos authentication is supported by Apache Sqoop.
8. We may import the data directly into Hive for data analysis using Sqoop. We can also use HBase, a NoSQL database, to store our data.
9. Instead of importing tables into a directory in HDFS, we can tell Apache Sqoop to import them into Accumulo.

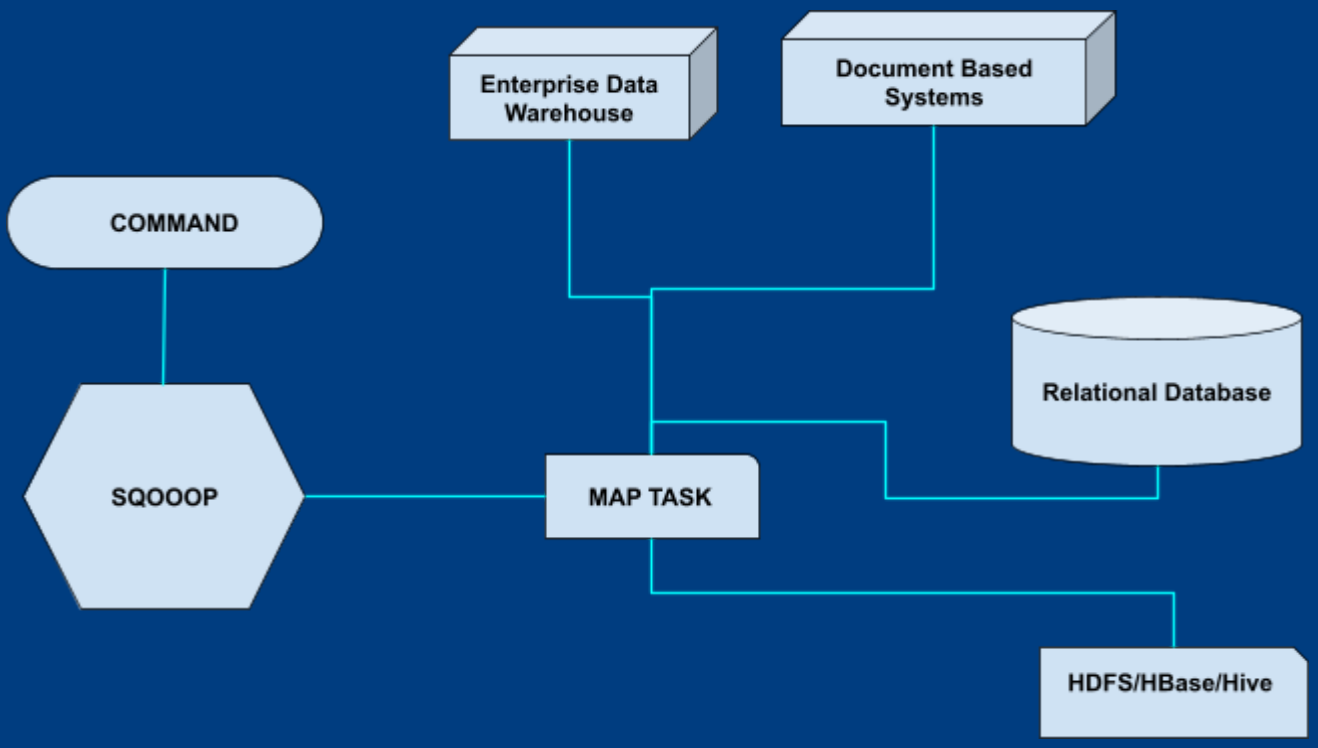
➢ Flume Vs Sqoop

The goal is to make the most use of the given resources while maintaining data consistency. Data ingestion is difficult with Hadoop because processing is done in batches, streams, or in real time, which complicates data management. Parallel processing, data quality, machine data at a rate of several gigabytes per minute, multiple source ingestion, real-time ingestion, and scalability are some of the most typical issues with Hadoop data intake. Apache Sqoop and Apache Flume are two popular Hadoop etl technologies that assist enterprises in overcoming data ingestion issues. If you're wondering what the difference is between Flume and Sqoop, you've come to the right place.

	SQOOP	FLUME
Basic Difference	Apache Sqoop is an effective hadoop tool for importing data from RDBMS's.	Apache Flume is service designed for streaming logs into Hadoop environment.
Data Flow	Sqoop works well with any kind of RDBMS that has JDBC connectivity.	Flume functions well for streaming data sources which are generated continuously in hadoop environments such as log files from multiple servers.

Type of Loading	Sqoop is not event driven.	Flume is event driven.
When to use?	Sqoop is an ideal fit if the data is sitting in databases like Teradata, Oracle, MySql server, PostgreSql.	Flume is a better choice when moving bulk streaming data from various sources like JMS or Spooling directory.
Link to HDFS	HDFS is the destination for importing data	Data flows from multiple channels into HDFS..
Where to use?	Sqoop is used for parallel data transfers and data imports as it copies data quickly.	Flume is used for collecting and aggregating data because of its distributed, reliable nature and highly available backup routes.
Features	• Sqoop parallelizes data transfer for optimal system utilisation and fast performance. • Sqoop provides direct input i.e it can map relational databases and import directly into HBase and Hive. • Sqoop makes data analysis efficient. • Sqoop helps in mitigating the excessive loads to external systems. • Sqoop provides data interaction programmatically.	• Flume is a flexible data ingestion tool. • Flume provides high throughput and low latency. • Flume has a declarative configuration but provides ease of extensibility. • Flume is fault tolerant, linearly scalable and stream oriented.

➢ Sqoop architecture and working



The Sqoop import tool is essentially an utility that imports individual tables from an RDBMS to HDFS. Each row in a table is treated as a record in HDFS.

Additionally, when we run the Sqoop command, our primary task is separated into subtasks. The map job, on the other hand, is responsible for dealing with it on its own. It is the subtask that imports a portion of data into the Hadoop Ecosystem while defining the map task. Similarly, all map jobs import all data at the same time.

Export, on the other hand, functions in the same way.
A Sqoop Export tool is a tool that exports a set of files from HDFS to an RDBMS. Furthermore, there are files that act as input to Sqoop that contain records. Those files are what we refer to as table rows.

Furthermore, the job is divided into map tasks, and when we submit our job, the chunk of data from HDFS is sent. The pieces are then exported to a structured data destination.

Similarly, we obtain the entire data set at the destination by joining all of the exported data chunks. However, in the vast majority of cases, an RDBMS (MYSQL/Oracle/SQL Server) is used.

In addition, aggregations necessitate a reduction step. Sqoop, on the other hand, does not conduct any aggregations; it just imports and exports data. Map jobs also launch numerous mappers based on the number specified by the user.

Furthermore, each mapper task will be assigned a portion of data to be imported into Sqoop. Sqoop also evenly distributes the input data among the mappers to achieve great performance. Following that, each mapper establishes a database connection using JDBC. Also retrieves the Sqoop-assigned data portion. Furthermore, it uses the inputs provided via the CLI to write it to HDFS, Hive, or HBase.