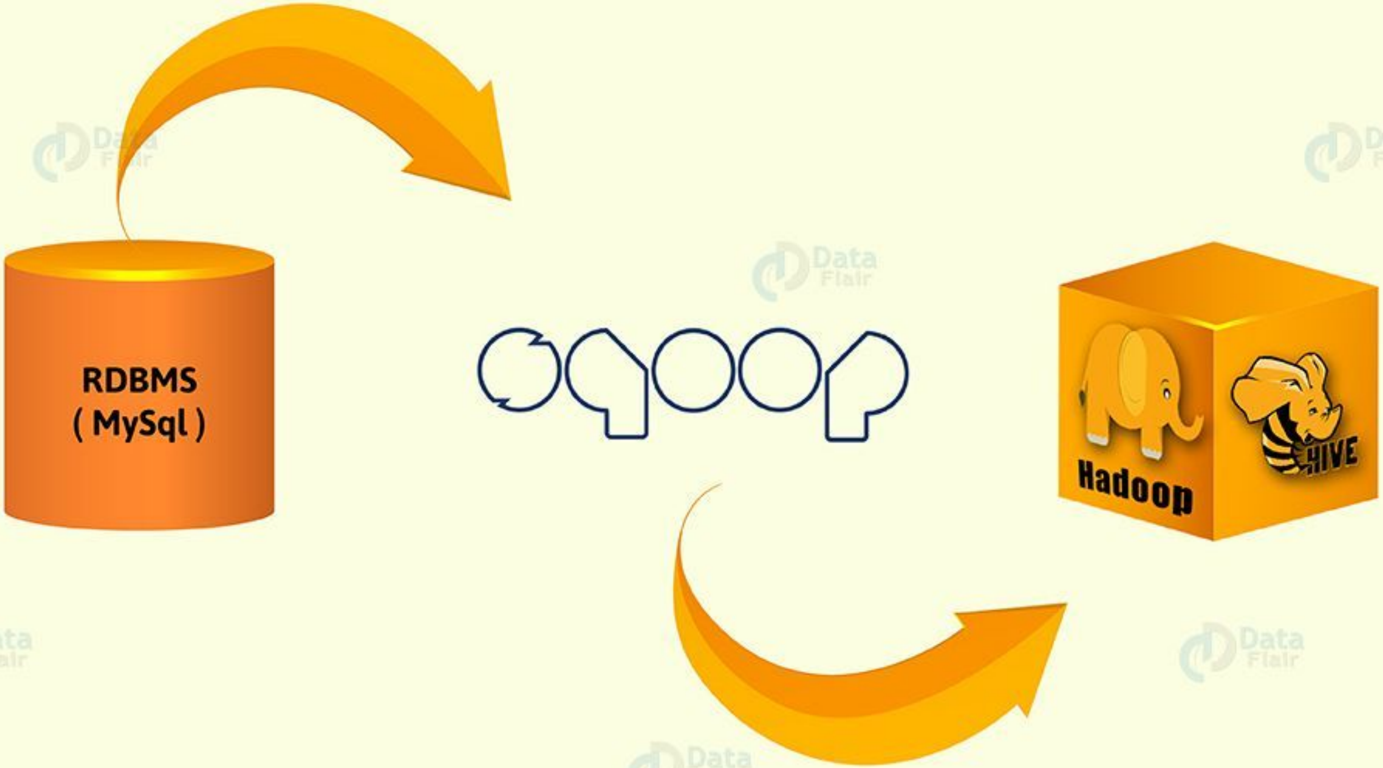
Apache Sqoop

What is **Apache Sqoop?**

A tool which we use for transferring data between Hadoop and relational database servers is what we call **Sqoop**.

While it comes to import data from a relational database management system (RDBMS) such as MySQL or Oracle into the Hadoop Distributed File System (HDFS), we can use Sqoop.

Also, we can use Sqoop to transform the data in Hadoop MapReduce and then export the data back into an RDBMS.



RDBMS
(MySql)

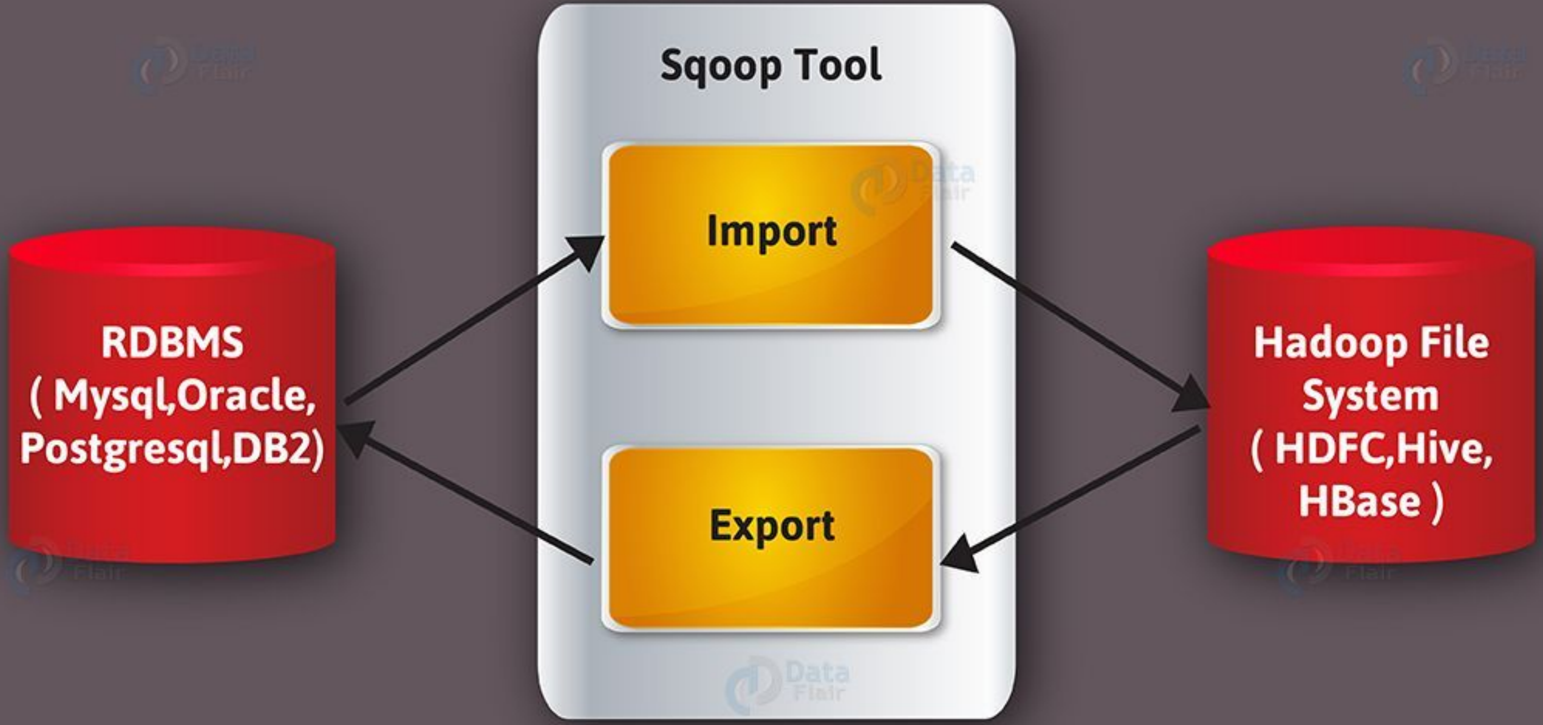
In addition, there are several processes which Apache Sqoop automates, such as relying on the database to describe the schema to import data. Moreover, to import and export the data, Sqoop uses MapReduce. Also, offers parallel operation as well as fault tolerance. Basically, we can say Sqoop is provided by the Apache Software Foundation.

Basically, Sqoop (“SQL-to-Hadoop”) is a straightforward command-line tool. It offers the following capabilities:

1. Generally, helps to Import individual tables or entire databases to files in HDFS
2. Also can Generate Java classes to allow you to interact with your imported data
3. Moreover, it offers the ability to import from SQL databases straight into your Hive data warehouse.

How **Sqoop** works?

Working of Sqoop



Sqoop Tool

Import

Export

RDBMS
(Mysql,Oracle,
Postgresql,DB2)

Hadoop File
System
(HDFC,Hive,
HBase)

Sqoop Import

Basically, when it comes to importing tool, it imports individual tables from RDBMS to HDFS.

Here, in HDFS each row in a table is treated as a record.

Moreover, in Avro and Sequence files all records are stored as text data in text files or as binary data.

a) Purpose of Sqoop Import

From an RDBMS to HDFS, the import tool imports an individual table. Here, in HDFS each row from a table is represented as a separate record. Moreover, we can store Records as text files (one record per line). However, in binary representation as Avro or SequenceFiles.

b) syntax

We can type the import arguments in any order with respect to one another, while the Hadoop generic arguments must precede any import arguments only.

The important thing to note here is that arguments are grouped into collections organized by function. Basically, some collections are present in several tools. For example, the “common” arguments.

Common arguments

–connect <jdbc-uri>

Specify JDBC connect string

–connection-manager <class-name>

Specify connection manager class to use

–driver <class-name>

Manually specify JDBC driver class to use

–hadoop-mapped-home <dir>

Override $HADOOP_MAPRED_HOME

–help

Print usage instructions

–password-file

Set path for a file containing the authentication password

-P

Read password from console

–password <password>

Set authentication password

–username <username>

Set authentication username

–verbose

Print more information while working

–connection-param-file

Optional properties file that provides connection parameters

–relaxed-isolation

Set connection transaction isolation to read uncommitted for the mappers.

Sqoop Export

When we want to export a set of files from HDFS back to an RDBMS we use the export tool.

Basically, there are rows in table those are the files which are input to Sqoop those contains records, which we call as rows in the table.

Although, those files are read and parsed into a set of records. Also, delimited with the user-specified delimiter.

a) Purpose of Sqoop Export

When we want to export a set of files from HDFS back to an RDBMS we use the export tool. One condition is here, the target table must already exist in the database.

In addition, to transform these into a set of INSERT statements, the default operation is that inject the records into the database. Moreover, Sqoop will generate UPDATE statements in “update mode,” that replace existing records in the database. Whereas, in “call mode” Sqoop will make a stored procedure call for each record.

b) Syntax

$ sqoop export (generic-args) (export-args)

$ sqoop-export (generic-args) (export-args)

Basically, the export arguments can be entered in any order with respect to one another,

However, the Hadoop generic arguments must precede any export arguments.

Import

```
sqoop import --connect jdbc:mysql://localhost/db --username [mysql username] -p(if password needed) -m [numb of mappers] --table [table name] --target-dir [hdfs directory]
```

Export

```
sqoop export --connect jdbc:mysql://localhost/db --username [mysql username] -p(if password needed) -m [numb of mappers] --table [table name] --export-dir [source]
```

Questions?