# Objective

- ✓ What is Sqoop
- ✓ Sqoop Features
- ✓ Sqoop 1 Architecture
- ✓ Sqoop 1 Vs. Sqoop 2
- ✓ Sqoop Connector
- ✓ Code Generation
- ✓ Sqoop Import
- ✓ Sqoop Export

# What is Sqoop

- ✓ Sqoop = **Sq**l + Had**oop**
- ✓ Apache Sqoop is an open source tool that allows users to extract data from a structured data store into Hadoop for further processing.
- ✓ It is designed to transfer data between Hadoop and relational databases or mainframes.
- ✓ Sqoop is an open source tool written at **Cloudera**.
- ✓ Sqoop uses **MapReduce** to import and export the data, which provides parallel operation as well as fault tolerance.
- ✓ By default it uses **four** mappers but this value is configurable.
- ✓ We can use Sqoop to import data from RDBMS into HDFS, transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.
- ✓ Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres, and HSQLDB
- ✓ Sqoop internally uses JDBC interface so it should work with any JDBC compatible database.
- ✓ Populate data into HDFS , HIVE and Hbase
- ✓ Sqoop is written in Java.
- ✓ Current stable release- **1.4.6**

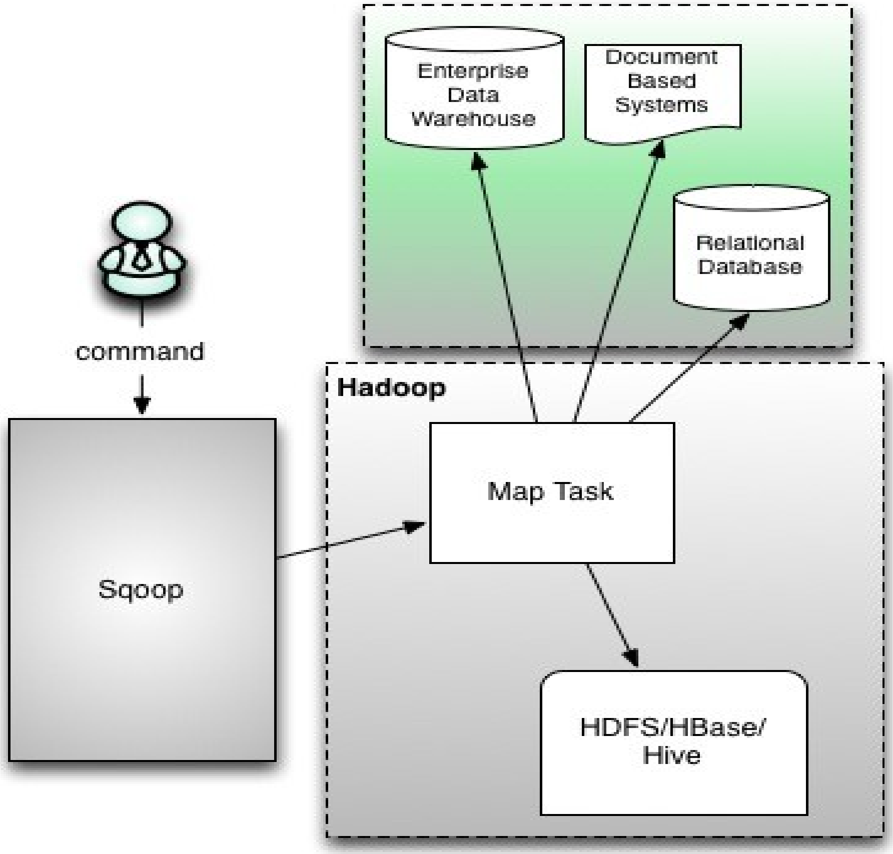
## It allows us to

- ✓ Import one table
- ✓ Import complete database
- ✓ Import selected tables
- ✓ Import selected columns from a particular table.
- ✓ Filter out certain rows from certain table etc.

# Sqoop Features

- ✓ Reads column info from table and generates Java classes.
- ✓ Can be used in further MapReduce processing.
- ✓ Uses MapReduce to read tables from database.
- ✓ Can select individual table (or subset of columns).
- ✓ Supports most JDBC standard types and null values
- ✓ Multiple data representations supported
- ✓ Supports local and remote Hadoop clusters, databases.
- ✓ Sqoop integrates with Oozie, allow us to schedule automatic import/export tasks.
- ✓ Sqoop supports Text files (default), SequenceFiles, Avro files and Parquet files.

# Sqoop 1 Architecture



# Sqoop 1 Vs. Sqoop 2

| Sqoop 1 | Sqoop 2 |
|---|---|
| Provide Command Line Tool | Provide CLI, WebUI, Rest API |
| Does not provide Java API so it is difficult to embed it in other Program | Provide Java API |
| Every connector has to know about every output format, so it is lot of work to write new connectors. | Support all connectors through JDBC |
| Only Map Task | Map and Reduce Task |

# Sqoop Connector

- ✓ Sqoop connector is a modular component that enable Sqoop imports and exports.
- ✓ Sqoop ships with connectors for working with a range of popular databases, including MySQL, PostgreSQL, Oracle, SQL  Server, DB2 and Netezza.
- ✓ Sqoop has a generic JDBC connector for connecting to any database that supports Java’s JDBC protocol.
- ✓ Sqoop provides  optimized MySQL, PostgreSQL, Oracle, and Netezza connectors for bulk transfer more efficiently.

# Code Generation

Along with writing the contents of the database table to HDFS, Sqoop also generate Java source file for RDBMS table.
This file written on current local directory.
The generated class is capable of holding a single record retrieved from the imported table.

Codegen: Sqoop tool to generate source code without performing an import;

```
$ sqoop codegen –connect jdbc:mysql://localhost/hadoopguide \
> --table widgets --class-name Widget
```

The codegen tool simply generates code; it does not perform the full import.

# Sqoop Import

- Sqoop import command is used to import data from RDBMS to HDFS, Hive or HBase

```
% sqoop import –connect jdbc:mysql://localhost/<database name> \
--table <table name>
--m   1
```

- Sqoop's import tool will run a MapReduce job that connects to the MySQL database and reads the table.
- By default, this will use **four** map tasks in parallel to speed up the import process.
- Each task will write its imported results to a different file, but all in a common directory.
- By default, Sqoop will generate comma-delimited text files for our imported data.

**Incremental Imports:**

- Sqoop will import rows that have a column value (for the column specified with –check-column) that is greater than some specified value (set via --last-value).
- This is suitable for the case where new rows are added to the database table, but existing rows are not updated. This mode is called append mode, and is activated via --incremental append .

# Sqoop Export

- ✓ Export command is used to transfer data from HDFS/HIVE/HBase to RDBMS.
- ✓ Before exporting a table from HDFS to a database, we must prepare the database to receive the data by creating the target table.

```
% sqoop export --connect  jdbc:mysql://localhost/hadoopguide \
--m   1 \
--table sales_by_zip \
--export-dir /user/hive/warehouse/zip_profits \
--input-fields-terminated-by '\0001'
```

- ✓ Before performing the export, Sqoop picks a strategy based on the database connect string.
- ✓ Sqoop then generates a Java class based on the target table definition.
- ✓ A MapReduce job is then launched that reads the source datafiles from HDFS, parses the records using the generated class, and executes the chosen export strategy.