sqoop import Options:

Option	Description	Example
--connect	JDBC connection URL to the database.	--connect jdbc:mysql://<db_server>/mydb
--username	Username to connect to the database.	--username myuser
--password	Password to connect to the database.	--password mypassword
--table	Name of the table to import from.	--table <table_name>
--columns	Specify columns to import.	--columns "col1,col2"
--where	WHERE clause to filter rows during import.	--where "column > 100"
--split-by	Column to split data by during import.	--split-by id
--target-dir	HDFS directory where data will be stored.	--target-dir /user/hadoop/<target_directory>
--warehouse-dir	Base HDFS directory for imported data (deprecated in favor of --target-dir).	--warehouse-dir /user/hadoop/<warehouse_dir>
--delete-target-dir	Delete the target directory before import.	--delete-target-dir
--append	Append data to an existing target directory.	--append
--as-parquet	Import data in Parquet file format (requires Parquet support).	--as-parquet
--as-avro	Import data in Avro file format (requires Avro support).	--as-avro
--compress	Compression format for imported data (e.g., snappy, gzip).	--compress snappy
--null-string	String representing NULL in the data.	--null-string "\N"
--null-non-string	String representing non-string NULLs in the data.	--null-non-string "\N"
--hive-import	Import data directly into Hive (creates a Hive table).	--hive-import
--hive-table	Name of the Hive table to create during import.	--hive-table <hive_table_name>
--hive-overwrite	Overwrite existing Hive table during import.	--hive-overwrite
--hive-drop-import-delims	Drop characters that are used to delimit fields during Hive import.	--hive-drop-import-delims
--hive-delims-replacement	Replace characters used to delimit fields during Hive import.	--hive-delims-replacement

sqoop export Options:

Option	Description	Example
--connect	JDBC connection URL to the database.	--connect jdbc:mysql://<db_server>/mydb
--username	Username to connect to the database.	--username myuser
--password	Password to connect to the database.	--password mypassword
--table	Name of the table to export data to.	--table <table_name>
--export-dir	HDFS directory containing data to export.	--export-dir /user/hadoop/<source_directory>
--input-fields-terminated-by	Input field delimiter for data in HDFS.	--input-fields-terminated-by '\t'
--input-lines-terminated-by	Input line delimiter for data in HDFS.	--input-lines-terminated-by '\n'
--input-optionally-enclosed-by	Input field enclosure character for data in HDFS.	--input-optionally-enclosed-by '"'
--update-key	Columns to use as the update key for records in the database.	--update-key id,name
--update-mode	Specifies how to handle updates: updateonly, allowinsert, allowinsert and update, or upsertkey.	--update-mode updateonly
--staging-table	Name of the staging table in the database (exported data is initially written here).	--staging-table <staging_table_name>
--clear-staging-table	Clear the staging table before export.	--clear-staging-table
--batch	Batch size for export operations (number of records per transaction).	--batch 100
--input-null-string	String representing NULL in the data.	--input-null-string "\N"
--input-null-non-string	String representing non-string NULLs in the data.	--input-null-non-string "\N"
--columns	Specify columns to export.	--columns "col1,col2"
--num-mappers	Number of parallel map tasks to use for export.	--num-mappers 4

Sqoop commands

Command	Description	Usage Example
import	Import data from a database table into HDFS	sqoop import --connect jdbc:mysql://<db_server>/mydb --table <table_name> --target-dir /user/hadoop/<target_directory>
export	Export data from HDFS to a database table	sqoop export --connect jdbc:mysql://<db_server>/mydb --table <table_name> --export-dir /user/hadoop/<source_directory>
list-databases	List all databases on a database server	sqoop list-databases --connect jdbc:mysql://<db_server>
list-tables	List all tables in a database	sqoop list-tables --connect jdbc:mysql://<db_server>/mydb
eval	Evaluate SQL statement and display results	sqoop eval --connect jdbc:mysql://<db_server>/mydb --query "SELECT * FROM <table_name> LIMIT 10"
merge	Merge data from HDFS to a table using an UPSERT operation	sqoop merge --merge-key <primary_key> --table <table_name> --class-name <java_class> --jar-file <jar_file> --input-null-string <null_string> --input-null-non-string <null_non_string> --update-key <update_key> --update-mode <update_mode> --from <source_dir> --to <target_table>
create-hive-table	Import data into Hive and create a new Hive table	sqoop create-hive-table --connect jdbc:mysql://<db_server>/mydb --table <table_name> --hive-table <hive_table_name> --fields-terminated-by ','
import-all-tables	Import all tables from a database into HDFS	sqoop import-all-tables --connect jdbc:mysql://<db_server>/mydb --autoreset-to-one-mapper --warehouse-dir /user/hadoop/<target_directory>
import-mainframe	Import data from a mainframe dataset into HDFS	sqoop import-mainframe --connect <mainframe_connection_details> --username <username> --password <password> --dataset <dataset_name> --target-dir /user/hadoop/<target_directory>
job	Create, list, and manage saved jobs	sqoop job --create <job_name> -- import --connect jdbc:mysql://<db_server>/mydb --table <table_name> --target-dir /user/hadoop/<target_directory>
metastore	Work with the metadata repository	sqoop metastore --create <job_name> --meta-connect <metadata_connection_details>
version	Display the version information for Sqoop	