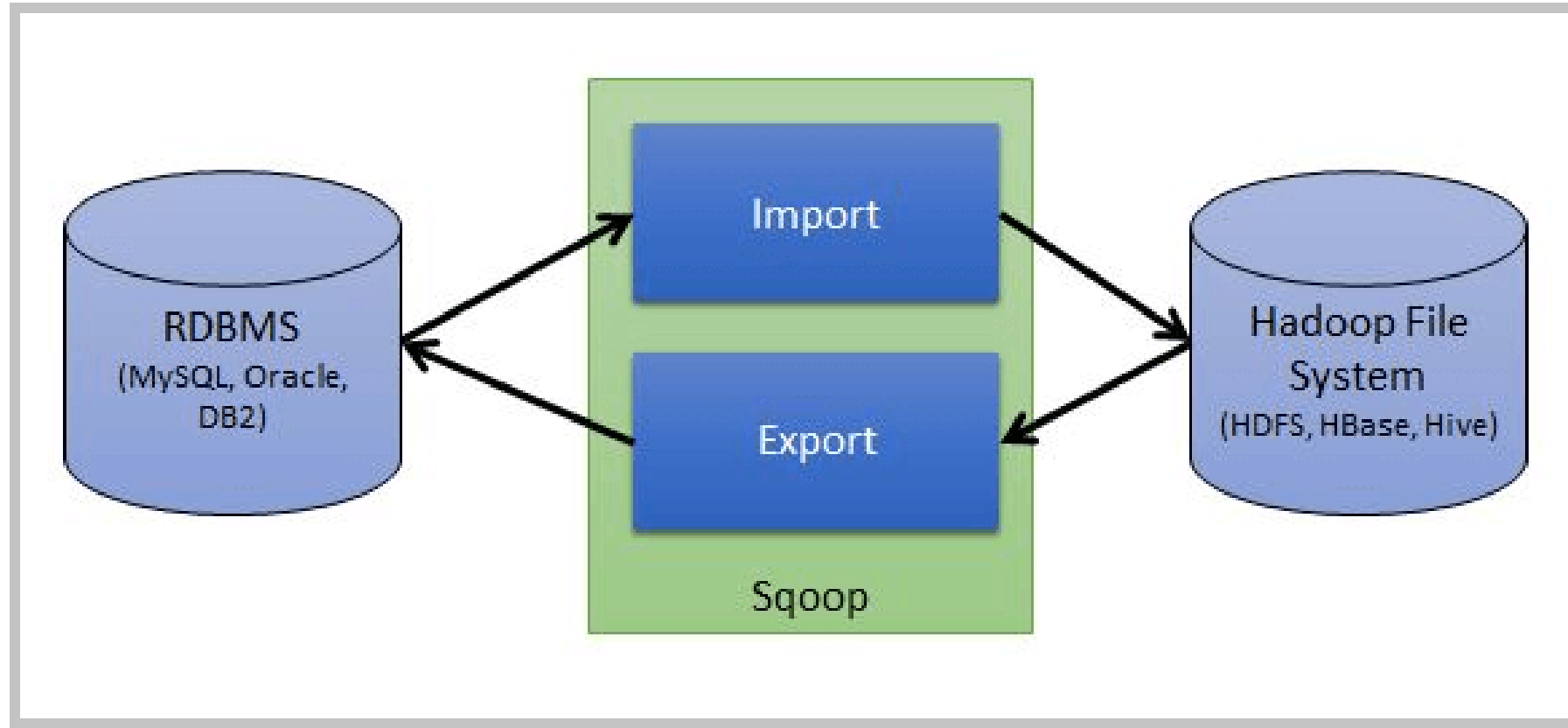


Apache Sqoop

SQOOP

- Sqoop is a tool designed to **transfer data** between **Hadoop** and **relational database** servers.
- It is used to import data from relational databases such as **MySQL, Oracle** to Hadoop HDFS, and export from Hadoop file system to relational databases.

How Sqoop Works?

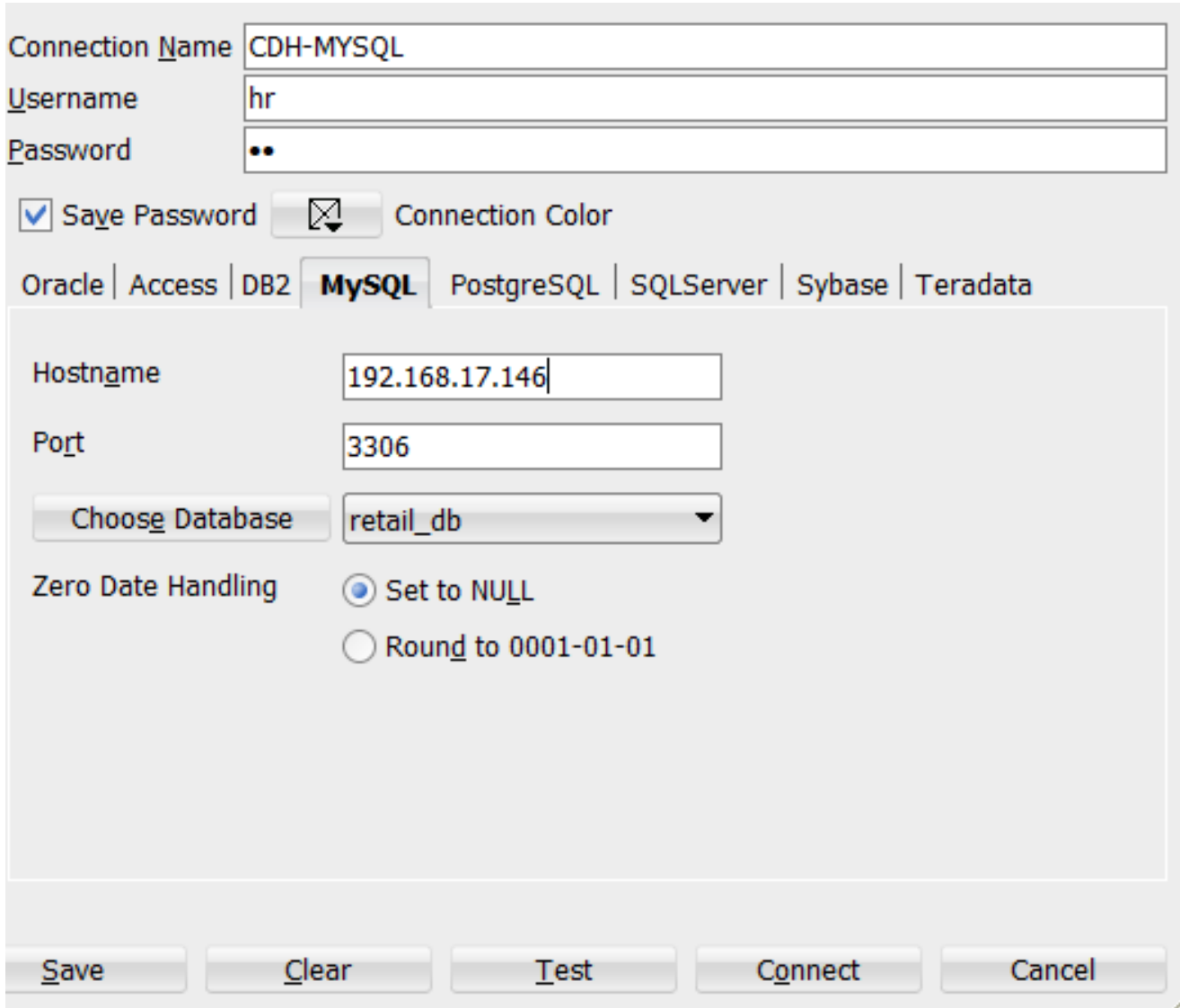


MySQL Connection (CLI)

Root user: **root** Password: **cloudera**

```
cloudera@quickstart:~  
[cloudera@quickstart ~]$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 31  
Server version: 5.1.73 Source distribution  
  
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> 
```

MySQL (SQL Developer)



The screenshot shows the 'New Database Connection' dialog box in SQL Developer, configured for a MySQL database. The 'Connection Name' is 'CDH-MYSQL'. The 'Username' is 'hr' and the 'Password' is masked with two dots. The 'Save Password' checkbox is checked. The 'Connection Color' is set to a light blue icon. The 'Database' tab is selected, showing 'MySQL' as the chosen database. The 'Hostname' is '192.168.17.146' and the 'Port' is '3306'. The 'Choose Database' dropdown is set to 'retail_db'. The 'Zero Date Handling' options are 'Set to NULL' (selected) and 'Round to 0001-01-01'. At the bottom, there are buttons for 'Save', 'Clear', 'Test', 'Connect', and 'Cancel'.

Connection Name: CDH-MYSQL

Username: hr

Password: ..

☒ Save Password ☐ Connection Color

Oracle | Access | DB2 | **MySQL** | PostgreSQL | SQLServer | Sybase | Teradata

Hostname: 192.168.17.146

Port: 3306

Choose Database: retail_db

Zero Date Handling: ☒ Set to NULL ☐ Round to 0001-01-01

Save Clear Test Connect Cancel

Root user: **hr** Password: **hr**

MySQL (Java JDBC)

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class MySQL {
    public static void main(String[] argv) throws Exception {
        String driver = "com.mysql.jdbc.Driver";
        String connection = "jdbc:mysql://192.168.17.146:3306/retail_db";
        String user = "hr";
        String password = "hr";
        Class.forName(driver);
        Connection con = DriverManager.getConnection(connection, user, password);

        Statement stmt = con.createStatement();

        ResultSet rs = stmt.executeQuery("select * from customers");

        while (rs.next()) {

            System.out.print(rs.getString("customer_fname"));
            System.out.println(" " + rs.getInt("customer_id"));
        }
    }
}
```

Create user in MYSQL

- `mysql> create user hr identified by 'hr';`
- `mysql> GRANT ALL PRIVILEGES ON * . * TO 'hr';`
- `mysql> show databases;`
- `mysql> USE retail_db;`

Creating table & Inserting data into MYSQL Table

- Creating Table & inserting data into table:

```
CREATE TABLE STUDENT(SNAME VARCHAR(20), SID INTEGER,MARKS  
INTEGER);
```

```
INSERT INTO STUDENT VALUES('ABC',123,50);
```

```
INSERT INTO STUDENT VALUES('XYZ',124,60);
```

```
INSERT INTO STUDENT VALUES('PQR',125,80);
```

```
INSERT INTO STUDENT VALUES('MNO',126,60);
```

```
INSERT INTO STUDENT VALUES('STU',127,90);
```


Importing data from MySQL table to HDFS

- Importing all rows & columns:

```
$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root -  
-password cloudera --table STUDENT --m 1
```

- Importing all rows but specific columns:

```
$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root -  
-password cloudera --table STUDENT --columns "SNAME" -m 1
```

- Import all columns, filter rows using where clause:

```
$ sqoop import --connect jdbc:mysql://localhost/retail_db --username root -  
-password cloudera --table STUDENT --where "sid>70" -m 1 --target-dir  
/user/cloudera/sturows
```

Checking Data Imported to HDFS

- `[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/`
- `http://localhost:50070/explorer.html#/user/cloudera`

Browse Directory

/user/cloudera							Gc
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	cloudera	cloudera	0 B	Thu Feb 09 18:58:06 -0800 2017	0	0 B	oozie-oozi
drwxr-xr-x	cloudera	cloudera	0 B	Fri Dec 30 22:35:50 -0800 2016	0	0 B	pavandir
drwxr-xr-x	cloudera	cloudera	0 B	Sat Feb 11 20:25:48 -0800 2017	0	0 B	pavantable

Importing all tables to HDFS

```
$sqoop import-all-tables --connect  
jdbc:mysql://localhost/retail_db --username root
```

***Note:** If you are using the import-all-tables, it is mandatory that every table in that database must have a primary key field.

Exporting data from HDFS file to MYSQL Table

- It is mandatory that the table to be exported is created manually and is present in the database from where it has to be exported.
- Export command will work in two ways
 1. insert : Insert mode will insert the new records from HDFS to RDBMS table.
 2. update: Update mode will update the records in the RDBMS from HDFS data. Update mode only update already existing records, it will not insert new records in the RDBMS.

Insert mode

- File path in HSFS: <http://localhost:50070/explorer.html#/user/hive/warehouse/employee>
- /user/hive/warehouse/employee/emp.txt

```
CREATE TABLE emp_details(  
    id INT NOT NULL PRIMARY KEY,  
    name VARCHAR(20),  
    salary INT,  
    deg VARCHAR(20));
```

- [cloudera@quickstart Desktop]\$ **sqoop export** --connect jdbc:mysql://localhost/retail_db --username root --password cloudera --table emp_details --export-dir /user/hive/warehouse/employee/emp.txt
- **Note:** if a record already present in the database table with same primary key, then it will raise **MySQLIntegrityConstraintViolationException** exception.

Update mode

- [cloudera@quickstart Desktop]\$ **sqoop export** --connect jdbc:mysql://localhost/retail_db --username root --password cloudera --table emp_details --export-dir /user/hive/warehouse/employee/emp.txt --update-key id

Sqoop Jobs

- **Sqoop job** creates and saves the import and export commands.
- **Sqoop Job Operations:**
 - Create Job (--create)
 - Verify Job (--list)
 - Inspect Job (--show)
 - Execute Job (--exec)

Sqoop Jobs

- **Create job:**
- [cloudera@quickstart Desktop]\$ sqoop job --create myjob -- import --connect jdbc:mysql://localhost/retail_db --username root --password cloudera --table STUDENT --m 1 --target-dir /sqoop/test_job
- **list jobs:** it will show all the jobs.
- [cloudera@quickstart Desktop]\$ sqoop job --list
- **inspect job:** it will show details about the job.
- [cloudera@quickstart Desktop]\$ sqoop job --show myjob
- **execute job:** it will execute the job.
- \$ sqoop job --exec myjob
- [cloudera@quickstart Desktop]\$ sqoop job --delete myjob

codegen

- [cloudera@quickstart Desktop]\$ **sqoop codegen** --connect jdbc:mysql://localhost/retail_db --username root --password cloudera --table emp_details