实验报告

课程名称：交通大数据管理

实验序号：实验 03　　实验项目名称：Sqoop 的安装与使用

学　　号	2210720131	姓　　名	薛文清	专业、班	22 大数据
实验地点	精工 1-406	指导教师	蔡钟淇	实验时间	2025-3-27

实验步骤（请给出关键代码、运行截图和分析）

1. hadoop 与 java 的安装与运行

   在之前的课程中我已经在虚拟机上安装过了 java 与 hadoop，版本分别为 1.8 和 3.3.5，所以 hadoop 的运行截图如下

```
xwq@ubuntu:/usr/local/hadoop/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
xwq@ubuntu:/usr/local/hadoop/sbin$ jps
2211 NameNode
2588 SecondaryNameNode
2732 Jps
2351 DataNode
```

2. 安装并配置 mysql

```
xwq@ubuntu:~$ sudo mysql -u root -p -e "SELECT VERSION();"
Enter password:
+-------------------------+
| VERSION()               |
+-------------------------+
| 5.7.33-0ubuntu0.16.04.1 |
+-------------------------+
```

3. 创建了一个测试数据库，名为 testdb，并创建了一个专用用户名为：sqoop_user, 密码为 Sqoop@123，并授予远程访问权限,尝试使用 sqoop 访问 mysql

   最终结果如图所示（**出现了部分警告，原因如下**）

   3.1 sqoop 依赖组件警告，原因是 hbase_home,hcat_home 等环境变量未设置，但是实验暂时用不到，所以可以忽略

   3.2 明文密码警告 我直接使用明文密码连接，可以使用-p 以交互方式安全输

入密码

3.3 MySQL SSL 连接警告 mysql 要求使用 SSL 加密连接，但是我没有明确配置，测试环境可忽略

```
xwq@ubuntu:~/Downloads$ sqoop list-databases \
> --connect jdbc:mysql://localhost:3306/ \
> --username sqoop_user \
> --password Sqoop@123
Warning: /usr/local/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2025-03-27 01:30:27,841 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2025-03-27 01:30:27,871 WARN tool.BaseSqoopTool: Setting your password on the command-line
 is insecure. Consider using -P instead.
2025-03-27 01:30:27,964 INFO manager.MySQLManager: Preparing to use a MySQL streaming resu
ltset.
Thu Mar 27 01:30:28 PDT 2025 WARN: Establishing SSL connection without server's identity v
erification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirement
s SSL connection must be established by default if explicit option isn't set. For complian
ce with existing applications not using SSL the verifyServerCertificate property is set to
 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL
=true and provide truststore for server certificate verification.
information_schema
testdb
```

4. 安装并配置 Hive https://dblab.xmu.edu.cn/blog/4309/

在 mysql 中新建一个数据库

```
mysql> create database hive;
Query OK, 1 row affected (0.00 sec)
```

将所有数据库的所有表，赋予权限给 hive 使用，并设置密码为 hive

```
mysql> GRANT ALL ON *.* TO 'hive'@'localhost' IDENTIFIED BY 'hive';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

成功启动 hive

```
xwq@ubuntu:/usr/local/hive$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.17.1.ja
r!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4
j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 5f30ed07-2ca1-41b7-b8ee-bff8be47059e

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-com
mon-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versio
ns. Consider using a different execution engine (i.e. spark, tez) or using Hive
1.X releases.
Hive Session ID = 59f8070c-6ec8-46c7-a6a9-a0e7f921fe87
hive>
```

5. 在 hive 中创建测试数据

   如图所示，创建了一个名为 test_db 的数据库，并在其中创建了一个测试表，共有三个属性，分别为 id，name 和 age，最后检查是否成功创建

```
Time taken: 0.68 seconds, Fetched: 1 row(s)
hive> CREATE DATABASE test_db;
OK
Time taken: 0.107 seconds
hive> USE test_db;
OK
Time taken: 0.027 seconds
hive> create table sample_table (
    > id INT,
    > name STRING,
    > age INT
    > )ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE;
OK
Time taken: 0.359 seconds
hive> show tables;
OK
sample_table
Time taken: 0.046 seconds, Fetched: 1 row(s)
```

6. **在 Mysql 中创建测试数据**，创建一个名为 sqoopuser 的用户，并赋予访问所有数据库的权限，密码为 123456，测试前文中，我已经在 Mysql 中创建了一个名为 testdb 的数据库，现在在该数据库中创建一个名为 sample_table 的表，如图所示，并插入数据

```
mysql> USE testdb
Database changed
mysql> CREATE TABLE sample_table (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     age INT
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO sample_table (name, age) VALUES
    ->     ('Alice', 25),
    ->     ('Bob', 30),
    ->     ('Charlie', 22);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM sample_table;
+----+---------+------+
| id | name    | age  |
+----+---------+------+
|  1 | Alice   |   25 |
|  2 | Bob     |   30 |
|  3 | Charlie |   22 |
+----+---------+------+
3 rows in set (0.01 sec)

mysql> S
```

7. **测试 sqoop 与 HDFS 的连接，并将 Mysql 中数据导入 HDFS 中**，使用 sqoop

命令将 Mysql 中 sample 表传入 HDFS 中，（出现问题，显示生成的 sample_table.jar 没有正确的添加到 hadoop 的类路径中，于是修改为--direct 模式，直接从 mysql 中导入，避免类加载问题）

```
                Map output records=3
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=515375104
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=33
2025-03-27 05:13:10,094 INFO mapreduce.ImportJobBase: Transferred 33 bytes in 1.851
5 seconds (17.8235 bytes/sec)
2025-03-27 05:13:10,094 INFO mapreduce.ImportJobBase: Retrieved 3 records.
```

8. 分别查询 Mysql 和 HDFS，如图所示，可见内容相同

```
xwq@ubuntu:~$ mysql -u sqoopuser -p123456 -h localhost -P 3306 testdb -e "SELECT *
FROM sample_table;"
mysql: [Warning] Using a password on the command line interface can be insecure.
+----+---------+------+
| id | name    | age  |
+----+---------+------+
|  1 | Alice   |   25 |
|  2 | Bob     |   30 |
|  3 | Charlie |   22 |
+----+---------+------+
xwq@ubuntu:~$ hdfs dfs -cat /user/sqoopuser/sample_table/part-m-00000
1,Alice,25
2,Bob,30
3,Charlie,22
xwq@ubuntu:~$
```

9. **测试 sqoop 导入 Hive，如图所示**

```
Logging initialized using configuration in jar:file:/usr/local/
mon-3.1.3.jar!/hive-log4j2.properties Async: true
Hive Session ID = 39b480da-42f2-43ab-9d21-7b68555bd29b
OK
sample_table_hive
Time taken: 1.003 seconds, Fetched: 1 row(s)
OK
id                      int
name                    string
age                     int
Time taken: 0.149 seconds, Fetched: 3 row(s)
OK
1       Alice   25
2       Bob     30
3       Charlie 22
4       Emma    28
5       Liam    32
Time taken: 1.311 seconds, Fetched: 5 row(s)
```

10. **测试利用 Sqoop 将 HDFS 导出到 Mysql 中，**需要先再 Mysql 中创建一个结

```
mysql> CREATE TABLE sample_table_new (
    ->   id INT,
    ->   name VARCHAR(255),  -- 根据实际需求调整长度（如 VARCHAR(100)）
    ->   age INT,
    ->   PRIMARY KEY (id)    -- 可选，如果数据中有唯一键
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
```

构相同的新表，我命名为 sample_talbe_new

```
mysql> CREATE TABLE sample_table_new (
    ->   id INT,
    ->   name VARCHAR(255),  -- 根据实际需求调整长度（如 VARCHAR(100)）
    ->   age INT,
    ->   PRIMARY KEY (id)    -- 可选，如果数据中有唯一键
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
```

导出后可以看到 testdb 数据库中多出了一张表,检查内容也没有错误

```
mysql> show tables;
+------------------+
| Tables_in_testdb |
+------------------+
| sample_table     |
| sample_table_new |
+------------------+
2 rows in set (0.00 sec)

mysql> select * from sample_table_new
    -> ;
+----+---------+------+
| id | name    | age  |
+----+---------+------+
|  1 | Alice   |   25 |
|  2 | Bob     |   30 |
|  3 | Charlie |   22 |
+----+---------+------+
3 rows in set (0.00 sec)
```

11. **实现增量导入功能**

修改 Mysql 中的表如图

```
mysql> INSERT INTO sample_table (id, name, age) VALUES
    ->   (4, 'Emma', 28),
    ->   (5, 'Liam', 32);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

使用增量模式成功导入后，并检查 HDFS 中的内容，结果如图所示，内容成功添加

```
2025-03-27 05:49:55,796 INFO mapreduce.ImportJobBase: Transferred 20 bytes in 1.882
9 seconds (10.622 bytes/sec)
2025-03-27 05:49:55,797 INFO mapreduce.ImportJobBase: Retrieved 2 records.
2025-03-27 05:49:55,813 INFO util.AppendUtils: Appending to directory sample_table
2025-03-27 05:49:55,820 INFO util.AppendUtils: Using found partition 1
2025-03-27 05:49:55,838 INFO tool.ImportTool: Incremental import complete! To run a
nother incremental import of all data following this import, supply the following a
rguments:
2025-03-27 05:49:55,838 INFO tool.ImportTool:  --incremental append
2025-03-27 05:49:55,838 INFO tool.ImportTool:   --check-column id
2025-03-27 05:49:55,838 INFO tool.ImportTool:   --last-value 5
2025-03-27 05:49:55,838 INFO tool.ImportTool: (Consider saving this with 'sqoop job
 --create')
xwq@ubuntu:~$ hdfs dfs -cat /user/sqoopuser/sample_table/part-m-*
1,Alice,25
2,Bob,30
3,Charlie,22
4,Emma,28
5,Liam,32
```

教师评语 签名： 日期：	成绩