暨南大学本科实验报告专用纸

课程名称 云计算实验 成绩评定

实验项目名称 分布式数据仓库 Hive 指导教师 魏林锋

实验项目编号 80660602 实验项目类型 综合 实验地点 线上

学生姓名 陈宇 学号 2020101642

学院 信息科学技术学院 系 计算机系 专业 软件工程

实验时间 2022 年 10 月 16 日上午～10 月 16 日 上午 温度 ℃湿度

实验 1 分布式数据仓库 Hive

1.1 实验目的

1）理解 Hive 工作原理。

2）通过实验掌握内嵌模式和独立模式安装 Hive 的过程。

3）通过实验掌握 Hive SQL 的使用。

4）通过实验掌握 Hive Java API 的使用。

1.2 内嵌模式 Hive 的安装

1.2.1 实验内容

完成内嵌模式 Hive 的安装。

1.2.2 实验环境

已经配置完成的 Hadoop 伪分布式或完全分布式环境。环境配置如下：

Hadoop01：192.168.24.91 Hadoop02：192.168.24.92 Hadoop03：192.168.24.93 管理员用户：root / admin@1

Hadoop 用户：hadoop / hadoop

1.2.3 实验步骤

1、使用 xftp 工具将 apache-hive-3.1.0-bin.tar.gz 文件上传到服务器上，解压 apache-hive-3.1.0-bin.tar.gz 文件，并重命名 hive 文件夹。命令如下：

```
[root@master ~]# tar -zxvf apache-hive-3.1.0-bin.tar.gz -C /usr/
[root@master ~]# mv /usr/apache-hive-3.1.0-bin/ /usr/hive
```

*需要注意的是，如果使用 Hadoop3.0 版本进行实验，Hive 必须安装 3.1.0 及以上版本，如果使用低版本安装，则会出现“Unrecognized Hadoop major version number: 3.1.0”的错误。如果使用 Hadoop2.0 版本进行实验则不会出现该问题。

```
apache-hive-3.1.0-bin/conf/parquet-logging.properties
apache-hive-3.1.0-bin/hcatalog/share/doc/hcatalog/README.txt
apache-hive-3.1.0-bin/lib/hive-common-3.1.0.jar
apache-hive-3.1.0-bin/lib/hive-classification-3.1.0.jar
apache-hive-3.1.0-bin/lib/hive-upgrade-acid-3.1.0.jar
apache-hive-3.1.0-bin/lib/hive-shims-3.1.0.jar
apache-hive-3.1.0-bin/lib/hive-shims-common-3.1.0.jar
apache-hive-3.1.0-bin/lib/log4j-slf4j-impl-2.10.0.jar
apache-hive-3.1.0-bin/lib/log4j-api-2.10.0.jar
apache-hive-3.1.0-bin/lib/log4j-core-2.10.0.jar
apache-hive-3.1.0-bin/lib/guava-19.0.jar
apache-hive-3.1.0-bin/lib/commons-lang-2.6.jar
apache-hive-3.1.0-bin/lib/libthrift-0.9.3.jar
apache-hive-3.1.0-bin/lib/httpclient-4.5.2.jar
apache-hive-3.1.0-bin/lib/httpcore-4.4.4.jar
apache-hive-3.1.0-bin/lib/commons-logging-1.0.4.jar
```

2、添加环境变量，并使其生效。命令如下：

```
[root@master ~]# vi /etc/profile
export HIVE_HOME=/usr/hive
export PATH=$HIVE_HOME/bin :$PATH

[root@master ~]# source /etc/profile
```

```
[root@master ~]# vi /etc/profile
[root@master ~]# source /etc/profile
```

3、进入/usr/hive/conf/目录，创建 hive-env.sh 文件。命令如下：

```
[root@master ~]# cd /usr/hive/conf/
```

```
[root@master conf]# cp hive-env.sh.template hive-env.sh
[root@master conf]# vi hive-env.sh
#指定 Hadoop 路径
Hadoop_HOME=/usr/hadoop
#指定 hive 配置文件的路径
export HIVE_CONF_DIR=/usr/hive/conf
#指定 jar 包位置
export HIVE_AUX_JARS_PATH=/usr/hive/lib
```

```
[root@master conf]# cp hive-env.sh.template hive-env.sh
[root@master conf]# vi hive-env.sh
[root@master conf]# vi hive-site.xml
```

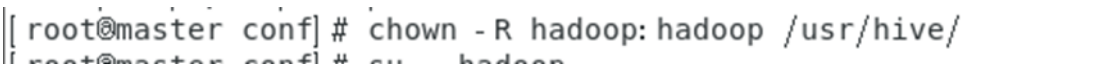
4、创建并配置 hive-site.xml 文件。命令如下：

```
[root@master conf]# vi hive-site.xml
<configuration>
<property>
    <name>hive.exec.local.scratchdir</name>
    <value>/usr/hive/tmp</value>
     <description>Local scratch space forHive jobs</description>
</property>
<property>
    <name>hive.downloaded.resources.dir</name>
    <value>/usr/hive/downloaded</value>
    <description>Temporary localdirectory for added resources in the remote file system.</description>
</property>
<property>
    <name>hive.querylog.location</name>
    <value>/usr/hive/querylog</value>
    <description>Location of Hive runtime structured log file</description>
</property>
```

```
<property>
    <name>hive.server2.logging.operation.log.location</name>
    <value>/usr/hive/server2</value>
    <description>Top level directorywhere operation logs are stored if logging functionality isenabled</description>
</property>
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:derby:;databaseName=metastore_db;create=true</value>
  <description>JDBC connect string for aJDBC metastore</description>
</property>
</configuration>
```

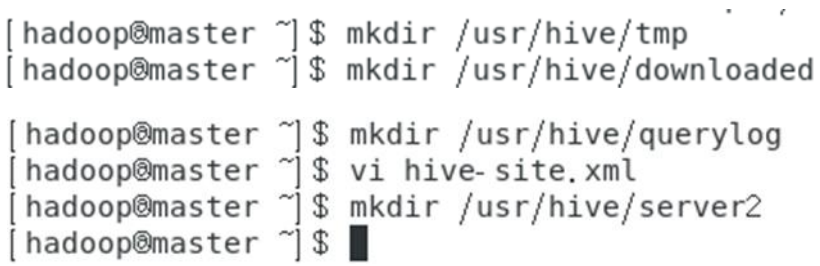
5、修改 hive 文件夹权限。命令如下：

```
[root@master conf]# chown -R hadoop:hadoop /usr/hive/
```



6、切换到 Hadoop 用户，创建 tmp、downloaded、querylog、server2 等四个文件夹。命令如下：

```
[root@master conf]# su hadoop
[hadoop@master conf]$ mkdir /usr/hive/tmp
[hadoop@master conf]$ mkdir /usr/hive/downloaded
[hadoop@master conf]$ mkdir /usr/hive/querylog
[hadoop@master conf]$ mkdir /usr/hive/server2
```



7、在进行初始化 hive 之前，需要删除 hadoop 家目录下的 metastore_db 文件夹。然后初始化 Hive。命令如下：

```
[hadoop@master conf]$ cd
[hadoop @master ~]$ rm -rfv metastore_db/
[hadoop @master ~]$ schematool -dbType derby -initSchema
```

```
[hadoop@master ~]$ schematool -dbType derby -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/
impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7
.25.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]
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

8、启动 Hadoop 集群。命令如下：

```
[hadoop @master ~]$ start-all.sh
```

9、启动 Hive，并新建一个名为 testdb 的数据库，如果新建成功，则 Hive 安装成功。命令如下：

```
[hadoop @master ~]$ hive
hive> create database testdb;
OK
Time taken: 0.787 seconds
```

```
hive> create database testdb;
OK
Time taken: 0.471 seconds
```

1.3 独立模式 Hive 的安装

1.3.1 实验内容

完成独立模式 Hive 的安装。

1.3.2 实验环境

已经配置完成的 Hadoop 伪分布式或完全分布式环境。环境配置如下：

Hadoop01：192.168.24.91

Hadoop02：192.168.24.92

Hadoop03：192.168.24.93

管理员用户：root / admin@1

Hadoop 用户：hadoop / hadoop

1.3.3 实验步骤

1、实验环境

已经配置完成的 Hadoop 伪分布式或完全分布式环境。

2、实验内容

2.1 安装 Mariadb

1、安装并启动 Mariadb 服务。命令如下：

```
[root@master ~]# yum install -y mariadb-server mariadb mariadb-devel
[root@master ~]# systemctl start mariadb
[root@master ~]# systemctl enable mariadb
```

需要注意的是，做这个实验的时候需要外网联通，否则不能使用 yum 命令。或者可以更改本地 yum 源对上述包进行安装。

```
[root@master ~]# systemctl start mariadb
[root@master ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /us
r/lib/systemd/system/mariadb.service.
[root@master ~]#
```

2、设置 mysql 中的 root 用户的登陆密码，并重新加载权限。命令如下：

```
[root@master ~]# mysql -u root
MariaDB [(none)]> UPDATE mysql.user SET password=PASSWORD('123456') WHERE
User='root' AND Host='localhost';
MariaDB [(none)]> flush privileges;
MariaDB [(none)]> exit;
```

```
[root@master ~]# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> UPDATE mysql.user SET password=PASSWORD('123456') WHERE User='root' A
ND Host='localhost';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
```

3、登录 mysql，为 Hive 建立相对应的账户，并赋予足够的权限。命令如下：

```
[root@master ~]# mysql -u root -p

MariaDB [(none)]> CREATE USER 'hadoop' IDENTIFIED BY '123456';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'hadoop'@'%' WITH GRANT

OPTION;

MariaDB [(none)]> flush privileges;

MariaDB [(none)]> exit;
```

```
[root@master ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'hadoop' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'hadoop'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
```

4、使用 Hadoop 用户登录 MySQL。

```
[root@master ~]# mysql_secure_installation

重置 hadoop 用户密码，然后一直按 Y 健，直至安装完成。

[root@master ~]# mysql -u hadoop -p
```

```
MariaDB [(none)]> exit;
```

```
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n]  Y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]  Y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

```
[root@master ~]# mysql -u hadoop -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Bye
```

5、使用 xftp 工具将 apache-hive-3.1.0-bin.tar.gz 文件上传到服务器上，解压 apache-hive-3.1.0-bin.tar.gz 文件，并重命名 hive 文件夹。命令如下：

```
[root@master ~]# tar -zxvf apache-hive-3.1.0-bin.tar.gz -C /usr/
[root@master ~]# mv /usr/apache-hive-3.1.0-bin/ /usr/hive
```

*需要注意的是，如果使用 Hadoop3.0 版本进行实验，Hive 必须安装 3.1.0 及以上版本，如果使用低版本安装，则会出现“Unrecognized Hadoop major version number: 3.1.0”的错误。如果使用 Hadoop2.0 版本进行实验则不会出现该问题。

6、添加环境变量，并使其生效。命令如下：

```
[root@master ~]# vi /etc/profile
export HIVE_HOME=/usr/hive
```

```
export PATH=$HIVE_HOME/bin :$PATH

[root@master ~]# source /etc/profile
```

注：前面已经配置完成

7、进入 hive/conf 目录，创建 hive-env.sh 文件。命令如下：

```
[root@master ~]# cd /usr/hive/conf/
[root@master conf]# cp hive-env.sh.template hive-env.sh
[root@master conf]# vi hive-env.sh
#指定 Hadoop 路径
Hadoop_HOME=/usr/hadoop
#指定 hive 配置文件的路径
export HIVE_CONF_DIR=/usr/hive/conf
#指定 jar 包位置
export HIVE_AUX_JARS_PATH=/usr/hive/lib
```

```
# Set HADOOP_HOME to point to a specific hadoop install directory
Hadoop_HOME=/usr/hadoop

# Hive Configuration Directory can be controlled by:
export HIVE_CONF_DIR=/usr/hive/conf
# Folder containing extra libraries required for hive compilation/execution can be cont
rolled by:
export HIVE_AUX_JARS_PATH=/usr/hive/lib
```

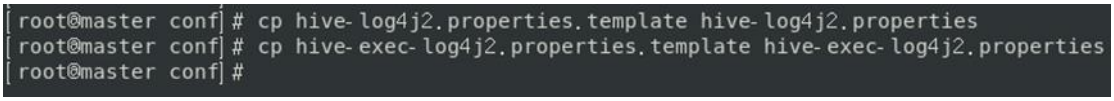
8、创建并配置 hive-site.xml 文件。命令如下：

```
[root@master conf]# vi hive-site.xml
<configuration>
<property>
<name>hive.exec.local.scratchdir</name>
<value>/usr/hive/tmp</value>
<description>Local scratch space forHive jobs</description>
</property>
<property>
<name>hive.downloaded.resources.dir</name>
<value>/usr/hive/downloaded</value>
<description>Temporary localdirectory for added resources in the remote file
system.</description>
```

```
</property>
<property>
<name>hive.querylog.location</name>
<value>/usr/hive/querylog</value><description>Location of Hive runtime structured log file</description>
</property>
<property>
<name>hive.server2.logging.operation.log.location</name>
<value>/usr/hive/server2</value>
<description>Top level directorywhere operation logs are stored if logging functionality isenabled</description>
</property>
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://master:3306/hive?createDatabaseIfNotExist=true</value>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>org.mariadb.jdbc.Driver</value>
</property>
<property>
<name>javax.jdo.option.ConnectionUserName</name>
<value>hadoop</value>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>123456</value>
</property>
</configuration>
```

9、创建 hive-log4j2.properties 和 hive-exec-log4j2.properties 两个文件。命令如下：

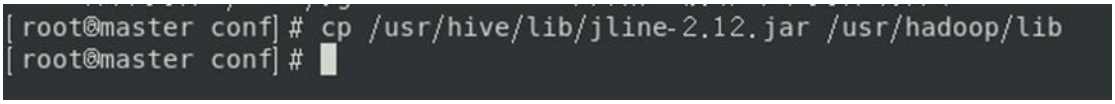
```
[root@master conf]# cp hive-log4j2.properties.template hive-log4j2.properties
[root@master conf]# cp hive-exec-log4j2.properties.template hive-exec-log4j2.properties
```



10、使用 xftp 工具将 jdbc 驱动 mariadb-java-client-2.2.6.jar 上传到/usr/hive/lib 目录下。

11、将 jline-2.12.jar 复制到/usr/hadoop/lib 下。命令如下：

```
[root@master conf]# cp /usr/hive/lib/jline-2.12.jar /usr/hadoop/lib
```



12、修改 hive 文件夹权限。命令如下：

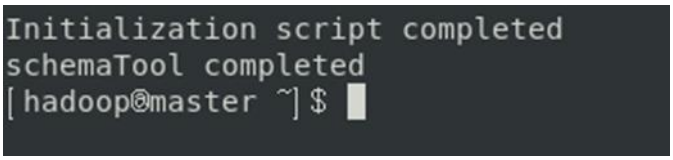
```
[root@master conf]# chown -R hadoop:hadoop /usr/hive/
```

13、切换到 Hadoop 用户，创建 tmp、downloaded、querylog、server2 等四个文件夹。命令如下：

```
[root@master conf]# su hadoop
[hadoop@master conf]$ mkdir /usr/hive/tmp
[hadoop@master conf]$ mkdir /usr/hive/downloaded
[hadoop@master conf]$ mkdir /usr/hive/querylog
[hadoop@master conf]$ mkdir /usr/hive/server2
```

14、在进行初始化 hive 之前，需要删除 hadoop 家目录下的 metastore_db 文件夹。然后初始化 Hive。命令如下：

```
[hadoop@master conf]$ cd
[hadoop @master ~]$ rm -rfv metastore_db/
[hadoop @master ~]$ schematool -initSchema -dbType mysql
```



15、启动 Hadoop 集群。命令如下：

```
[hadoop @master ~]$ start-all.sh
```

16、启动 Hive，并新建一个名为 testdb 的数据库，如果新建成功，则 Hive 安装成功。命令如下：

```
[hadoop @master ~]$ hive
hive> create database testdb;
OK
Time taken: 0.787 seconds
```

```
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.
.25.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]
which: no hbase in (/usr/hive/bin:/usr/hadoop/bin:/usr/hadoop/sbin:/usr/java/jdk1.8.0_
12/bin:/usr/java/jdk1.8.0_212/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/b
n:/sbin:/home/master/.local/bin:/home/master/bin)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j
impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.
.25.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 = 168296b9-0a7b-4f4c-b650-f7be89e7e505

Logging initialized using configuration in file:/usr/hive/conf/hive-log4j2.properties
sync: true
Hive Session ID = d470f608-eb69-40ce-90f5-bf1ecc4914a5
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Co
sider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> create database testdb;
OK
Time taken: 0.518 seconds
hive>
```

10、查询 MySQL 数据库中的元数据。命令如下：

```
[hadoop @master ~]$ mysql -u hadoop -p
MariaDB [(none)]> use hive;
MariaDB [hive]> show tables;
```

```
[hadoop@master ~]$ mysql -u hadoop -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use hive;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [hive]> show tables;
+----------------------------+
| Tables_in_hive             |
+----------------------------+
| AUX_TABLE                  |
| BUCKETING_COLS             |
| CDS                        |
| COLUMNS_V2                 |
| COMPACTION_QUEUE           |
| COMPLETED_COMPACTIONS      |
| COMPLETED_TXN_COMPONENTS   |
| CTLGS                      |
| DATABASE_PARAMS            |
| DBS                        |
| DB_PRIVS                   |
| DELEGATION_TOKENS          |
| FUNCS                      |
| FUNC_RU                    |
| GLOBAL_PRIVS               |
| HIVE_LOCKS                 |
| IDXS                       |
| INDEX_PARAMS               |
| I_SCHEMA                   |
| KEY_CONSTRAINTS            |
| MASTER_KEYS                |
| MATERIALIZATION_REBUILD_LOCKS |
| METASTORE_DB_PROPERTIES    |
| PARTITION_EVENTS           |
| PARTITION_KEYS             |
| PARTITION_KEY_VALS         |
| PARTITION_PARAMS           |
| PART_COL_PRIVS             |
| PART_COL_STATS             |
| PART_PRIVS                 |
| REPL_TXN_MAP               |
| ROLES                      |
| ROLE_MAP                   |
| RUNTIME_STATS              |
| SCHEMA_VERSION             |
| SDS                        |
| SD_PARAMS                  |
| SEQUENCE_TABLE             |
| SERDES                     |
| SERDE_PARAMS               |
| SKEWED_COL_NAMES           |
| SKEWED_COL_VALUE_LOC_MAP   |
| SKEWED_STRING_LIST         |
| SKEWED_STRING_LIST_VALUES  |
| SKEWED_VALUES              |
| SORT_COLS                  |
| TABLE_PARAMS               |
| TAB_COL_STATS              |
| TBLS                       |
| TBL_COL_PRIVS              |
| TBL_PRIVS                  |
| TXNS                       |
| TXN_COMPONENTS             |
| TXN_TO_WRITE_ID            |
| TYPES                      |
| TYPE_FIELDS                |
| VERSION                    |
| WM_MAPPING                 |
| WM_POOL                    |
| WM_POOL_TO_TRIGGER         |
| WM_RESOURCEPLAN            |
| WM_TRIGGER                 |
| WRITE_SET                  |
+----------------------------+
74 rows in set (0.00 sec)
```

实验 2 Hive 命令基础

1.实验环境

已经配置完成的 Hadoop 伪分布式或完全分布式环境。

2.实验内容

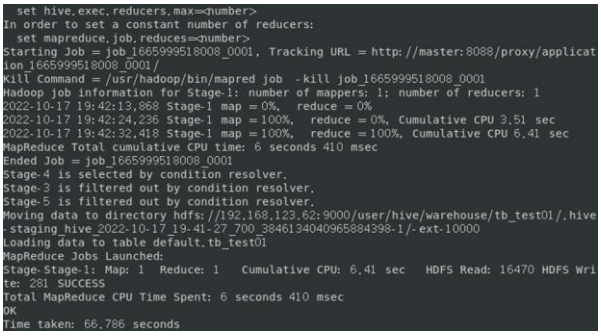
2.1 插入数据

hive> INSERT INTO tb_test01 VALUES(1,'xiaolin',21);

hive> INSERT INTO tb_test01 VALUES(2,'xiaojie',23);

```
hive> INSERT INTO tb_test01 VALUES(1,'xiaolin',21);
Query ID = hadoop_20190321072134_b48f7785-2985-47b8-a439-dcbf25b58f6c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1543216548935_0006, Tracking URL = http://master:8088/proxy/ap
plication_1543216548935_0006/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1543216548935_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-03-21 07:22:04,464 Stage-1 map = 0%,  reduce = 0%
2019-03-21 07:22:16,045 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.03 sec
2019-03-21 07:22:24,444 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.67
```

```
hive> INSERT INTO tb_test01 VALUES(2,'xiaojie',23);
Query ID = hadoop_20190321072229_5f3bf948-e1ad-4444-9af4-6edd4187676a
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1543216548935_0007, Tracking URL = http://master:8088/proxy/ap
plication_1543216548935_0007/
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-03-21 07:23:02,552 Stage-1 map = 0%,  reduce = 0%
2019-03-21 07:23:52,523 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.47 sec
2019-03-21 07:24:02,946 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.05
sec
```



2.2 查看表和数据

1）查看表

hive> show tables;

```
hive> show tables;
OK
tb_test01
Time taken: 0.011 seconds, Fetched: 1 row(s)
hive>
```

```
hive> show tables;
OK
tb_test01
Time taken: 0.027 seconds, Fetched: 1 row(s)
hive>
```

2）正则匹配表名

hive> show tables
'.*01';

```
hive> show tables '.*01';
OK
tb_test01
Time taken: 0.017 seconds, Fetched: 1 row(s)
```

```
hive> show tables '.*01';
OK
tb_test01
Time taken: 0.021 seconds, Fetched: 1 row(s)
hive>
```

3）查看表数据

hive> select * from tb_test01;

```
hive> select * from tb_test01;
OK
1       xiaolin 21
2       xiaojie 23
Time taken: 0.291 seconds, Fetched: 2 row(s)
```

```
hive> select * from tb_test01;
OK
1       xiaolin 21
2       xiaojie 23
Time taken: 0.169 seconds, Fetched: 2 row(s)
```

4）查看表结构

hive> desc tb_test01;

```
hive> desc tb_test01;
OK
id                      int
name                    varchar(20)
age                     int
Time taken: 0.058 seconds, Fetched: 3 row(s)
```

```
hive> desc tb_test01;
OK
id                              int
name                            string
age                             int
Time taken: 0.032 seconds, Fetched: 3 row(s)
hive>
```

2.3 修改表

1）在表 tb_test01 添加一列

hive> alter table tb_test01 add columns(age int);

hive> desc tb_test01;

```
hive> alter table tb_test01 add columns(sex int);
OK
Time taken: 0.114 seconds
hive> desc tb_test01;
OK
id                      int
name                    varchar(20)
age                     int
sex                     int
Time taken: 0.049 seconds, Fetched: 4 row(s)
```

```
hive> alter table tb_test01 add columns(sex int);
OK
Time taken: 0.17 seconds
hive> desc tb_test01;
OK
id                      int
name                    string
age                     int
sex                     int
Time taken: 0.023 seconds, Fetched: 4 row(s)
```

2）在表 tb_test01 添加一列并增加列字段注释

hive> alter table tb_test01 add columns(birthday date comment 'studnet birthday');

hive> desc tb_test01;

```
hive> alter table tb_test01 add columns(birthday date comment 'studnet birthday')
;
OK
Time taken: 0.099 seconds
hive> desc tb_test01;
OK
id                      int
name                    varchar(20)
age                     int
sex                     int
birthday                date                    studnet birthday
Time taken: 0.05 seconds, Fetched: 5 row(s)
```

```
hive> alter table tb_test01 add columns(birthday date comment 'studnet birthday');
OK
Time taken: 0.048 seconds
hive> desc tb_test01;
OK
id                      int
name                    string
age                     int
sex                     int
birthday                date                    studnet birthday
Time taken: 0.022 seconds, Fetched: 5 row(s)
```

3）创建表 tb_test02 并创建索引字段 ds

hive> create table tb_test02(id int,class_num int,grade int,class int) partitioned by(ds string);

```
hive> create table tb_test02(id int,class_num int,grade int,class int) partitione
d by(ds string);
OK
Time taken: 0.052 seconds
hive> desc tb_test02;
OK
id                      int
class_num               int
grade                   int
class                   int
ds                      string

# Partition Information
# col_name              data_type               comment
ds                      string
Time taken: 0.078 seconds, Fetched: 9 row(s)
```

```
hive> create table tb_test02(id int,class_num int,grade int,class int) partitioned by(d
s string);
OK
Time taken: 0.024 seconds
hive>
```

4）重命名表

hive> alter table tb_test01 rename to stu_info;

```
hive> alter table tb_test01 rename to stu_info;
OK
Time taken: 0.136 seconds
hive> show tables;
OK
stu_info
tb_test02
Time taken: 0.023 seconds, Fetched: 2 row(s)
```

```
hive> alter table tb_test01 rename to stu_info;
OK
Time taken: 0.109 seconds
hive>
```

2.4 删除表

hive> drop table tb_test02;

```
hive> drop table tb_test02;
OK
Time taken: 0.149 seconds
hive> show tables;
OK
stu_info
Time taken: 0.02 seconds, Fetched: 1 row(s)
```

```
hive> drop table tb_test02;
OK
Time taken: 0.174 seconds
```

2.5 数据导入

1）创建表结构

hive> create table tb_test03 (id int,stu_num int,sex int) row format delimited fields terminated by '\t';

```
hive> create table tb_test03 (id int,stu_num int,sex int) row forma
t delimited fields terminated by '\t';
OK
Time taken: 0.029 seconds
```

```
hive> create table tb_test03 (id int,stu_num int,sex int) row format delimited fields t
erminated by '\t';
OK
Time taken: 0.13 seconds
```

2）从操作系统本地文件加载数据(LOCAL)

创建数据(文本以 tab 分隔)

```
[hadoop@master ~]$ cat /home/hadoop/tb_test03.txt
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
```

```
[hadoop@master ~]$ cat /home/hadoop/tb_test03.txt
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
```

从本地文件导入数据

hive> load data local inpath '/home/Hadoop/tb_test03.txt' overwrite into table tb_test03;

hive> select * from tb_test03;

```
hive> load data local inpath '/home/hadoop/tb_test03.txt' overwrite into table tb
_test03;
Loading data to table db_test.tb_test03
OK
Time taken: 0.291 seconds
hive> select * from tb_test03;
OK
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
Time taken: 0.198 seconds, Fetched: 5 row(s)
```

```
hive> load data local inpath '/home/hadoop/tb_test03.txt' overwrite into table tb_test0
3;
Loading data to table default.tb_test03
OK
Time taken: 0.2 seconds
hive> select * from tb_test03;
OK
1       120001  1
2       120002  0
3       120003  1
4       120004  0
5       120005  1
Time taken: 0.106 seconds, Fetched: 5 row(s)
```

3）从 HDFS 加载数据

创建表 tb_test04

hive> create table tb_test04(id int,stu_num int,sex int) row format delimited fields terminated by '\t';

```
hive>  create table tb_test04(id int,stu_num int,sex int) row format delimited fi
elds terminated by '\t';
OK
Time taken: 0.049 seconds
```

```
hive> create table tb_test04(id int,stu_num int,sex int) row format delimited fields te
rminated by '\t';
OK
Time taken: 0.117 seconds
hive>
```

在本地创建数据(文本以 tab 分隔)

```
[hadoop@master ~]$ cat /home/hadoop/tb_test04.txt
6       120006  1
7       120007  0
8       120008  1
```

```
[hadoop@master ~]$ cat /home/hadoop/tb_test04.txt
6       120006  1
7       120007  0
8       120008  1
```

上传到 HDFS 文件系统

```
[hadoop@master ~]$ hdfs dfs -put /home/hadoop/tb_test04.txt /user/hive
[hadoop@master ~]$ hdfs dfs -ls /user/hive
Found 2 items
-rw-r--r--   3 hadoop supergroup         33 2019-03-21 10:02 /user/hive/tb_test04
.txt
drwxr-xr-x   - hadoop supergroup          0 2019-03-21 07:18 /user/hive/warehouse
```

```
[hadoop@master ~]$ hdfs dfs -put /home/hadoop/tb_test04.txt /user/hive
[hadoop@master ~]$ hdfs dfs -ls /user/hive
Found 2 items
-rw-r--r--   3 hdfs supergroup         33 2022-10-17 20:19 /user/hive/tb_test04.txt
drwxr-xr-x   - hdfs supergroup          0 2022-10-17 20:16 /user/hive/warehouse
```

从 HDFS 加载数据

hive> load data inpath '/user/hive/tb_test04.txt' overwrite into table tb_test04;

```
hive> load data inpath '/user/hive/tb_test04.txt' overwrite into table tb_test04;

Loading data to table db_test.tb_test04
OK
Time taken: 0.24 seconds
hive> select * from tb_test04;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 0.188 seconds, Fetched: 3 row(s)
```

```
[hadoop@master ~]$ hdfs dfs -put /home/hadoop/tb_test04.txt /user/hive
[hadoop@master ~]$ hdfs dfs -ls /user/hive
Found 2 items
-rw-r--r--   3 hdfs supergroup         33 2022-10-17 20:19 /user/hive/tb_test04.txt
drwxr-xr-x   - hdfs supergroup          0 2022-10-17 20:16 /user/hive/warehouse
```

4）从其他表导入数据

hive> insert overwrite table tb_test03 select * from tb_test04;

```
hive> insert overwrite table tb_test03 select * from tb_test04;
Query ID = hadoop_20190321100352_ffb42b81-f033-4cf4-87fc-c9fb236939d9
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1543216548935_0008, Tracking URL = http://master:8088/proxy/ap
plication_1543216548935_0008/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1543216548935_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-03-21 10:04:22,368 Stage-1 map = 0%,  reduce = 0%
2019-03-21 10:04:44,286 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.94 sec
2019-03-21 10:04:52,630 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.57 s
ec
```

```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-10-17 20:21:15,410 Stage-1 map = 0%,  reduce = 0%
2022-10-17 20:21:23,686 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.06 sec
2022-10-17 20:21:30,838 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.04 sec
MapReduce Total cumulative CPU time: 9 seconds 40 msec
Ended Job = job_1665999518008_0003
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://192.168.123.62:9000/user/hive/warehouse/tb_test03/.hive-staging_hive_2022-10-17_20-20-12_835_1023470955419871390-1/-ext-10000
Loading data to table default.tb_test03
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.04 sec   HDFS Read: 13753 HDFS Write: 333 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 40 msec
OK
Time taken: 79.334 seconds
hive>
```

查看表 tb_test03 数据

```
hive> select * from tb_test03;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 0.175 seconds, Fetched: 3 row(s)
```

```
hive> select * from tb_test03;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 0.105 seconds, Fetched: 3 row(s)
hive>
```

从 HDFS 查看表 tb_test03 数据

```
[hadoop@master ~]$ hdfs dfs -ls /user/hive/warehouse/db_test.db/tb_test03
Found 1 items
-rw-r--r--   3 hadoop supergroup         33 2019-03-21 10:04 /user/hive/warehouse
/db_test.db/tb_test03/000000_1
^[[A[hadoop@master hdfs dfs -cat /user/hive/warehouse/db_test.db/tb_test03/000000
_1
6       120006  1
7       120007  0
8       120008  1
```

```
[hadoop@master ~]$ hdfs dfs -cat /user/hive/warehouse/tb_test03/000000_0
6       120006  1
7       120007  0
8       120008  1
```

5）创建表并从其他表导入数据

hive> create table tb_test05 as select * from tb_test03;

```
hive> create table tb_test05 as select * from tb_test03;
Query ID = hadoop_20190321100729_a794d87e-bfe0-4a08-a869-75d9d4c3518f
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1543216548935_0009, Tracking URL = http://master:8088/proxy/ap
plication_1543216548935_0009/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1543216548935_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-03-21 10:08:07,410 Stage-1 map = 0%,  reduce = 0%
2019-03-21 10:08:17,370 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.59 sec
MapReduce Total cumulative CPU time: 4 seconds 590 msec
Ended Job = job_1543216548935_0009
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://master:9000/user/hive/warehouse/db_test.db/.hive-
staging_hive_2019-03-21_10-07-29_728_3291624696208551781-1/-ext-10002
Moving data to directory hdfs://master:9000/user/hive/warehouse/db_test.db/tb_tes
t05
```

```
hive> create table tb_test05 as select * from tb_test03;
Query ID = hadoop_20221017202735_23d30c48-c1cf-4930-8e41-3eecc9a62c31
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1665999518008_0004, Tracking URL = http://master:8088/proxy/application_1665999518008_0004/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1665999518008_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-10-17 20:28:47,473 Stage-1 map = 0%,  reduce = 0%
2022-10-17 20:28:52,579 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.49 sec
MapReduce Total cumulative CPU time: 2 seconds 490 msec
Ended Job = job_1665999518008_0004
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://192.168.123.62:9000/user/hive/warehouse/.hive-staging_hive_2022-10-17_20-27-35_827_6148278091456188730-1/-ext-10002
Moving data to directory hdfs://192.168.123.62:9000/user/hive/warehouse/tb_test05
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.49 sec   HDFS Read: 5001 HDFS Write: 106 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 490 msec
OK
Time taken: 79.041 seconds
```

查看表 tb_test05 数据

```
hive> select * from tb_test05;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 0.162 seconds, Fetched: 3 row(s)
```

```
hive> select * from tb_test05;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 0.083 seconds, Fetched: 3 row(s)
hive>
```

从 HDFS 查看表 tb_test05 数据

```
[hadoop@master ~]$ hdfs dfs -cat /user/hive/warehouse/db_test.db/tb_test05/000000
_0
61200061
71200070
81200081
```

```
[hadoop@master ~]$ hdfs dfs -cat /user/hive/warehouse/tb_test05/000000_0
6 120006 1
7 120007 0
8 120008 1
```

6）仅复制表结构不导数据

hive> create table tb_test06 like tb_test05;

```
hive> create table tb_test06 like tb_test05;
OK
Time taken: 0.07 seconds
hive> select * from tb_test06;
OK
Time taken: 0.166 seconds
hive> desc tb_test06;
OK
id                      int
stu_num                 int
sex                     int
Time taken: 0.046 seconds, Fetched: 3 row(s)
```

```
hive> create table tb_test06 like tb_test05;
OK
Time taken: 0.948 seconds
```

2.6 数据导出

1）从 HDFS 复制到 HDFS 其他位置

```
[hadoop@master ~]$ hdfs dfs -cp /user/hive/warehouse/db_test.db/tb_test03/000000_
1 /user/tb_test03
[hadoop@master ~]$ hdfs dfs -ls /user
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2019-03-21 06:37 /user/hadoop
drwxr-xr-x   - hadoop supergroup          0 2019-03-21 10:03 /user/hive
-rw-r--r--   3 hadoop supergroup         33 2019-03-21 10:17 /user/tb_test03
[hadoop@master ~]$ hdfs dfs -cat /user/tb_test03
6       120006  1
7       120007  0
8       120008  1
```

```
[hadoop@master ~]$ hdfs dfs -cp /user/hive/warehouse/tb_test03/000000_0 /user/tb_test03
[hadoop@master ~]$ hdfs dfs -ls /user
Found 2 items
drwxr-xr-x   - hdfs supergroup          0 2022-10-17 20:19 /user/hive
-rw-r--r--   3 hdfs supergroup         33 2022-10-17 20:35 /user/tb_test03
[hadoop@master ~]$
```

2）通过 Hive 导出到本地文件系统

```
hive> select * from tb_test03;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 0.151 seconds, Fetched: 3 row(s)
```

```
hive> select * from tb_test03;
OK
6       120006  1
7       120007  0
8       120008  1
Time taken: 1.308 seconds, Fetched: 3 row(s)
```

hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/tb_test03' SELECT * FROM tb_test03;

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/tb_test03' SELECT * FROM tb_test03;
Query ID = hadoop_20190321101908_888c9acb-fc62-434c-9bd3-bb6aba0641d1
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1543216548935_0010, Tracking URL = http://master:8088/proxy/ap
plication_1543216548935_0010/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1543216548935_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-03-21 10:19:38,245 Stage-1 map = 0%,  reduce = 0%
2019-03-21 10:20:03,327 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
MapReduce Total cumulative CPU time: 5 seconds 480 msec
Ended Job = job_1543216548935_0010
Moving data to local directory /tmp/tb_test03
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 5.48 sec   HDFS Read: 4899 HDFS Write: 33
 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 480 msec
OK
Time taken: 56.005 seconds
```

```
hive> !cat /tmp/tb_test03/000000_1;
61200061
71200070
81200081
```

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/tb_test03' SELECT * FROM tb_test03;
Query ID = hadoop_20221017203706_fac33816-ea08-41ec-9175-8ef0f30f16e2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1665999518008_0005, Tracking URL = http://master:8088/proxy/application_1665999518008_0005/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1665999518008_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-10-17 20:37:30,240 Stage-1 map = 0%,  reduce = 0%
2022-10-17 20:37:36,467 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.33 sec
MapReduce Total cumulative CPU time: 2 seconds 330 msec
Ended Job = job_1665999518008_0005
Moving data to local directory /tmp/tb_test03
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.33 sec   HDFS Read: 4913 HDFS Write: 33 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 330 msec
OK
Time taken: 30.793 seconds
```

```
hive> !cat /tmp/tb_test03/000000_0;
6 120006 1
7 120007 0
8 120008 1
```

实验 3 Hive 命令进阶

1、实验环境

已经配置完成的 Hadoop 伪分布式或完全分布式环境。

2、实验内容

2.1 Hive 查询 HiveQL

1）创建表 tb_test07

hive> create table tb_test07 (a int,b int,c int) row format delimited fields terminated by '\t';

```
hive> create table tb_test07 (a int,b int,c int) row format delimited fields term
inated by '\t';
OK
Time taken: 0.056 seconds
```

```
hive> create table tb_test07 (a int,b int,c int) row format delimited fields terminated
 by '\t';
OK
Time taken: 0.122 seconds
hive>
```

2）从本地文件系统导入数据

```
[hadoop@master ~]$ cat /home/hadoop/tb_test07.txt
16      2       3
61      12      13
41      2       31
17      21      3
71      2       31
1       12      34
11      2       34
```

```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/tb_test07.txt' OVERWRITE INTO TABLE tb
_test07;
Loading data to table db_test.tb_test07
OK
Time taken: 0.211 seconds
hive> select * from tb_test07;
OK
16      2       3
61      12      13
41      2       31
17      21      3
71      2       31
1       12      34
11      2       34
Time taken: 0.142 seconds, Fetched: 7 row(s)
```

```
[hadoop@master ~]$ cat /home/hadoop/tb_test07.txt
16      2       3
61      12      13
41      2       31
17      21      3
71      2       31
1       12      34
11      2       34
```

```
[hadoop@master ~]$ cat /home/hadoop/tb_test07.txt
16      2       3
61      12      13
41      2       31
17      21      3
71      2       31
1       12      34
11      2       34
```

3）创建表 tb_test08,从表 tb_test07 导入数据

hive> CREATE TABLE tb_test08 AS SELECT * FROM tb_test07

```
hive> CREATE TABLE tb_test08 AS SELECT * FROM tb_test07;
Query ID = hadoop_20190321103237_6fea8d19-547d-423f-8557-0329a8e1d5f0
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1543216548935_0011, Tracking URL = http://master:8088/proxy/ap
plication_1543216548935_0011/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1543216548935_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-03-21 10:33:03,962 Stage-1 map = 0%,  reduce = 0%
2019-03-21 10:33:24,851 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.92 sec
MapReduce Total cumulative CPU time: 4 seconds 920 msec
Ended Job = job_1543216548935_0011
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://master:9000/user/hive/warehouse/db_test.db/.hive-
staging_hive_2019-03-21_10-32-37_491_4656061682550087134-1/-ext-10002
Moving data to directory hdfs://master:9000/user/hive/warehouse/db_test.db/tb_tes
t08
```

```
hive> CREATE TABLE tb_test08 AS SELECT * FROM tb_test07;
Query ID = hadoop_20221017204759_f8c749b7-65fa-4c32-89b2-63261a1ca25d
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1665999518008_0006, Tracking URL = http://master:8088/proxy/application_1665999518008_0006/
Kill Command = /usr/hadoop/bin/mapred job  -kill job_1665999518008_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-10-17 20:48:32,019 Stage-1 map = 0%,  reduce = 0%
2022-10-17 20:48:37,127 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.69 sec
MapReduce Total cumulative CPU time: 2 seconds 690 msec
Ended Job = job_1665999518008_0006
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://192.168.123.62:9000/user/hive/warehouse/.hive-staging_hive_2022-10-17_20-47-59_059_7422725025696525198-1/-ext-10002
Moving data to directory hdfs://192.168.123.62:9000/user/hive/warehouse/tb_test08
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.69 sec   HDFS Read: 4872 HDFS Write: 129 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 690 msec
OK
Time taken: 40.432 seconds
```

4）普通查询：排序，列别名，嵌套子查询

FROM (SELECT b,c as c2 FROM tb_test07) t SELECT t.b,t.c2 WHERE b>2 LIMIT 2

```
hive> FROM ( SELECT b,c as c2 FROM tb_test07) t SELECT t.b,t.c2 WHERE b>2 LIMIT 2
;
OK
12      13
21      3
Time taken: 0.3 seconds, Fetched: 2 row(s)
```

```
hive> FROM ( SELECT b,c as c2 FROM tb_test07) t SELECT t.b,t.c2 WHERE b>2 LIMIT 2;
OK
12      13
21      3
Time taken: 0.442 seconds, Fetched: 2 row(s)
```

5）连接查询：JOIN

hive> SELECT t1.a,t1.b,t2.a,t2.b FROM tb_test07 t1 JOIN tb_test08 t2 on t1.a=t2.a WHERE t1.c>10

```
hive> SELECT t1.a,t1.b,t2.a,t2.b FROM tb_test07 t1 JOIN tb_test08 t2 on t1.a=t2.a
 WHERE t1.c>10;
Query ID = hadoop_20190321103520_bb8a932c-744d-438d-bf43-936c5bdb3856
Total jobs = 1
```

```
Total MapReduce CPU Time Spent: 6 seconds 470 msec
OK
61      12      61      12
41      2       41      2
71      2       71      2
1       12      1       12
11      2       11      2
Time taken: 59.795 seconds, Fetched: 5 row(s)
```

```
hive> FROM ( SELECT b,c as c2 FROM tb_test07) t SELECT t.b,t.c2 WHERE b>2 LIMIT 2;
OK
12      13
21      3
Time taken: 0.442 seconds, Fetched: 2 row(s)
hive>  SELECT t1.a,t1.b,t2.a,t2.b FROM tb_test07 t1 JOIN tb_test08 t2 on t1.a=t2.a WHERE t1.c>10;
Query ID = hadoop_20221017205043_61f317b9-5ea3-405a-be3e-f358ff773e35
Total jobs = 1
```

```
Total MapReduce CPU Time Spent: 3 seconds 350 msec
OK
61      12      61      12
41      2       41      2
71      2       71      2
1       12      1       12
11      2       11      2
Time taken: 80.852 seconds, Fetched: 5 row(s)
```

6）聚合查询 1：count, avg

hive> SELECT count(*),avg(a) FROM tb_test08

```
hive> SELECT count(*),avg(a) FROM tb_test08;
Query ID = hadoop_20190321103837_97287b36-7987-413c-addc-9fdbfd962756
Total jobs = 1
Launching Job 1 out of 1
```

```
OK
7       31.142857142857142
Time taken: 56.363 seconds, Fetched: 1 row(s)
```

```
hive> SELECT count(*),avg(a) FROM tb_test08;
Query ID = hadoop_20221017205302_75177905-1ed6-457d-a4fa-590c5f2b1fa6
Total jobs = 1
Launching Job 1 out of 1
```

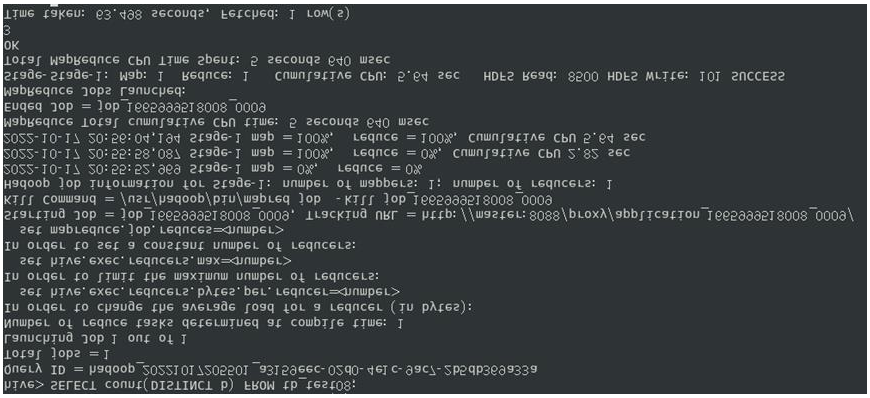
```
Total MapReduce CPU Time Spent: 10 seconds 620 msec
OK
7       31.142857142857142
Time taken: 69.031 seconds, Fetched: 1 row(s)
```

7）聚合查询 2：count, distinct

hive> SELECT count(DISTINCT b) FROM tb_test08;

```
hive> SELECT count(DISTINCT b) FROM tb_test08;
Query ID = hadoop_20190321104033_3788af07-1840-4230-8c43-7bbcd5295143
Total jobs = 1
Launching Job 1 out of 1
```

```
MapReduce Total cumulative CPU time: 9 seconds 780 msec
Ended Job = job_1543216548935_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.78 sec   HDFS Read: 8485 HDF
S Write: 101 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 780 msec
OK
3
Time taken: 65.956 seconds, Fetched: 1 row(s)
```

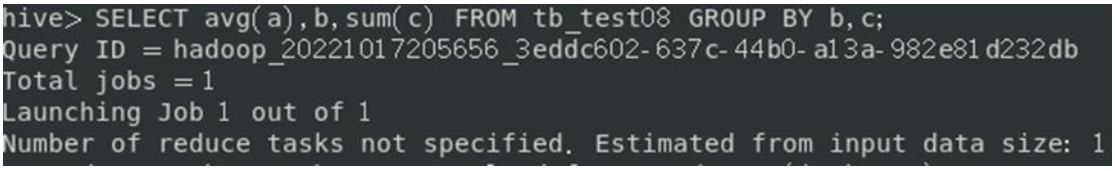


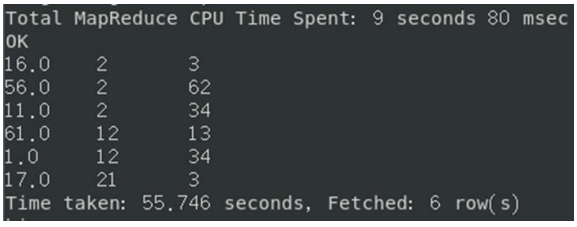
8）聚合查询 3：GROUP BY, HAVING

hive> SELECT avg(a),b,sum(c) FROM tb_test08 GROUP BY b,c;

```
hive> SELECT avg(a),b,sum(c) FROM tb_test08 GROUP BY b,c;
Query ID = hadoop_20190321104224_f9a9fdb2-4cce-45d0-ad34-5e7545952817
Total jobs = 1
Launching Job 1 out of 1
```

```
MapReduce Total cumulative CPU time: 12 seconds 890 msec
Ended Job = job_1543216548935_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.89 sec   HDFS Read: 15804 H
DFS Write: 219 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 890 msec
OK
16.0    2       3
56.0    2       62
11.0    2       34
61.0    12      13
1.0     12      34
17.0    21      3
Time taken: 78.689 seconds, Fetched: 6 row(s)
```





hive> SELECT avg(a),b,sum(c) FROM tb_test08 GROUP BY b,c HAVING sum(c)>30;

```
hive> SELECT avg(a),b,sum(c) FROM tb_test08 GROUP BY b,c HAVING sum(c)>30;
Query ID = hadoop_20190321104412_e263d019-290e-4696-9ade-84a6dc0a9112
Total jobs = 1
Launching Job 1 out of 1
```

```
MapReduce Total cumulative CPU time: 12 seconds 510 msec
Ended Job = job_1543216548935_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.51 sec   HDFS Read: 16371 H
DFS Write: 153 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 510 msec
OK
56.0    2       62
11.0    2       34
1.0     12      34
Time taken: 78.036 seconds, Fetched: 3 row(s)
```

```
hive> SELECT avg(a),b,sum(c) FROM tb_test08 GROUP BY b,c HAVING sum(c)>30;
Query ID = hadoop_20221017205834_4a1daea0-a67f-43f5-b52b-a6ae40d81ab6
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
```

```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Reduce: 1   Cumulative CPU: 7.27 sec   HDFS Read: 16386 HDFS Write: 153 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 270 msec
OK
56.0    2       62
11.0    2       34
1.0     12      34
Time taken: 63.982 seconds, Fetched: 3 row(s)
```

2.2 Hive 视图

1）创建视图

Hive 数据库视图和数据库视图的概念是一样的，我们还以 view01 为例

hive> CREATE VIEW view01 AS SELECT a,b FROM tb_test08 where c>28;

```
hive> CREATE VIEW view01 AS SELECT a,b FROM tb_test08 where c>28;
OK
Time taken: 0.179 seconds
hive> select * from view01;
OK
41      2
71      2
1       12
11      2
Time taken: 0.152 seconds, Fetched: 4 row(s)
```

```
hive> CREATE VIEW view01 AS SELECT a,b FROM tb_test08 where c>28;
OK
Time taken: 0.167 seconds
```

2）删除视图

hive> DROP VIEW IF EXISTS view01;

```
hive> DROP VIEW IF EXISTS view01;
OK
Time taken: 0.112 seconds
```

```
hive> DROP VIEW IF EXISTS view01;
OK
Time taken: 0.206 seconds
```

2.3 Hive 分区表

1）创建数据

分区表是数据库的基本概念，但很多时候数据量不大，我们完全用不到分区表。Hive 是一种 OLAP 数据仓库软件，涉及的数据量是非常大的，所以分区表在这个场景就显得非常重要!!

下面我们重新定义一个数据表结构。

```
[hadoop@master ~]$ cat /home/hadoop/20190320.csv
000001,031920,9.76
000002,032047,8.99
000004,031902,9.79
000005,032014,2.2
000001,032008,9.70
000001,032159,9.45
```

```
[hadoop@master ~]$ cat /home/hadoop/20190321.csv
000001,031920,9.76
000002,032047,8.99
000004,031902,9.79
000005,032014,2.2
000001,032008,9.70
000001,032159,9.45
```

```
[hadoop@master ~]$ vi /home/hadoop/20190320.csv
[hadoop@master ~]$ cat /home/hadoop/20190320.csv
000001,031920,9.76
000002,032047,8.99
000004,031902,9.79
000005,032014,2.2
000001,032008,9.70
000001,032159,9.45
```

```
[hadoop@master ~]$ cat /home/hadoop/20190321.csv
000001,031920,9.76
000002,032047,8.99
000004,031902,9.79
000005,032014,2.2
000001,032008,9.70
000001,032159,9.45
```

2）创建数据表

hive> CREATE TABLE tb_test09(SecurityID STRING,tradeTime STRING,PreClosePx DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

```
hive> CREATE TABLE tb_test09(SecurityID STRING,tradeTime STRING,PreClosePx DOUBLE
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.051 seconds
```

```
hive> CREATE TABLE tb_test09(SecurityID STRING,tradeTime STRING,PreClosePx DOUBLE) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.148 seconds
hive>
```

3）创建分区数据表

根据业务：按天和股票 ID 进行分区设计

hive> CREATE TABLE tb_test10(SecurityID STRING,TradeTime STRING,PreClosePx DOUBLE) PARTITIONED BY (tradeDate INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

```
hive> CREATE TABLE tb_test10(SecurityID STRING,TradeTime STRING,PreClosePx DOUBLE
) PARTITIONED BY (tradeDate INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.044 seconds
```

```
hive> CREATE TABLE tb_test10(SecurityID STRING,TradeTime STRING,PreClosePx DOUBLE) PART
ITIONED BY (tradeDate INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.13 seconds
hive>
```

4）导入数据

hive> LOAD DATA LOCAL INPATH '/home/Hadoop/20190320.csv' OVERWRITE INTO TABLE tb_test10 PARTITION (tradeDate=20190320);
hive> LOAD DATA LOCAL INPATH '/home/Hadoop/20190321.csv' OVERWRITE INTO TABLE tb_test10 PARTITION (tradeDate=20190321);

```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/20190320.csv' OVERWRITE INTO TABLE tb_
test10 PARTITION (tradeDate=20190320);
Loading data to table db_test.tb_test10 partition (tradedate=20190320)
OK
Time taken: 0.391 seconds
```

```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/20190321.csv' OVERWRITE INTO TABLE tb_
test10 PARTITION (tradeDate=20190321);
Loading data to table db_test.tb_test10 partition (tradedate=20190321)
OK
Time taken: 0.364 seconds
```

```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/20190320.csv' OVERWRITE INTO TABLE tb_test10
 PARTITION (tradeDate=20190320);
Loading data to table default.tb_test10 partition (tradedate=20190320)
OK
Time taken: 0.4 seconds
```

```
hive> LOAD DATA LOCAL INPATH '/home/hadoop/20190321.csv' OVERWRITE INTO TABLE tb_test10
 PARTITION (tradeDate=20190321);
Loading data to table default.tb_test10 partition (tradedate=20190321)
OK
Time taken: 0.801 seconds
hive>
```

5）查看分区表

hive> SHOW PARTITIONS tb_test10;

```
hive> SHOW PARTITIONS tb_test10;
OK
tradedate=20190320
tradedate=20190321
Time taken: 0.098 seconds, Fetched: 2 row(s)
```

```
hive> SHOW PARTITIONS tb_test10;
OK
tradedate=20190320
tradedate=20190321
Time taken: 0.105 seconds, Fetched: 2 row(s)
hive>
```

6）查询数据

hive> SELECT * FROM tb_test10 WHERE securityid='000001';

```
hive> SELECT * FROM tb_test10 WHERE securityid='000001';
OK
000001  031920  9.76    20190320
000001  032008  9.7     20190320
000001  032159  9.45    20190320
000001  031920  9.76    20190321
000001  032008  9.7     20190321
000001  032159  9.45    20190321
Time taken: 0.192 seconds, Fetched: 6 row(s)
```

```
hive> SELECT * FROM tb_test10 WHERE securityid='000001';
OK
000001  031920  9.76    20190320
000001  032008  9.7     20190320
000001  032159  9.45    20190320
000001  031920  9.76    20190321
000001  032008  9.7     20190321
000001  032159  9.45    20190321
Time taken: 0.246 seconds, Fetched: 6 row(s)
```

hive> SELECT * FROM tb_test10 WHERE tradedate=20190321 and PreClosePx<8;

```
hive>  SELECT * FROM tb_test10 WHERE tradedate=20190321 and PreClosePx<8;
OK
000005  032014  2.2     20190321
Time taken: 0.255 seconds, Fetched: 1 row(s)
```

```
hive> SELECT * FROM tb_test10 WHERE tradedate=20190321 and PreClosePx<8;
OK
000005  032014  2.2     20190321
Time taken: 0.251 seconds, Fetched: 1 row(s)
hive>
```

实验总结

初步理解了 Hive 工作原理。

通过实验掌握内嵌模式和独立模式安装 Hive 的过程。

通过实验掌握 Hive SQL 的使用。

通过实验掌握 Hive Java API 的使用

实验中要小心字符的输入，一个微小的符号都有可能引起麻烦的错误。