hive

一，hive简介

```
数据仓库：面向主题的，存储历史数据的，用于数据分析的。
关系型数据库：面向事务的，存储实时数据的，存储数据的。
hive是一个数据分析工具，它是基于hadoop的，依赖hdfs存储数据，依赖yarn运行mr程序来做分析。
```

hive的优点：

```
1.使用sql语句来做数据分析，代替写MR程序，提高效率。
2.提供很多内置的函数，也可以自定义函数UDF。
3.容错性好。
```

hive缺点：

```
1.sql语句转化MR程序时间长。
2.没有存储过程
3.UI不完善。
4.数据仓库功能弱。
```

hive查询分析流程：

```
1.创建数据库，数据表，系统会初始化映射元数据，在hdfs创建相应的目录。
2.上传数据至相应目录
3.写sql语句，sql语句转换器，优化器把sql转换成MR程序。
4.把MR程序提交给Yarn集群运行
```

元数据数据库：

```
derby:不具有共享性。因为在不同的目录启动hive，会在当前目录产生数据库。
mysql:具有共享性。
```

mysql的安装：

```
1.命令安装用的Yum Repository：
wget -i -c https://dev.mysql.com/get/mysql57-community-release-el7-10.noarch.rpm
2.yum -y install mysql57-community-release-el7-10.noarch.rpm
3.安装mysql，命令： yum -y install mysql-community-server
4.启动mysql，命令：systemctl start  mysqld.service
5.查看状态，命令：systemctl status mysqld.service
6.获取root密码，命令：grep "password" /var/log/mysqld.log
7.进入数据库，命令：mysql -uroot -p
8.修改初始密码，命令：ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';
9.可以修改密码设置方案：命令：set global validate_password_policy=0;
```

```
                命令：set global validate_password_length=1;
10.避免更新：
命令：yum -y remove mysql57-community-release-el7-10.noarch
11.允许远程登陆，命令：grant all privileges on *.* to 'root'@'%' IDENTIFIED BY 'root' with
grant option;
12.刷新权限，命令： flush privileges;
```

二，hive的部署

- 配置：

```
1.hive-env.sh
#配置HADOOP_HOME
HADOOP_HOME=/export/server/.....
2.hive-site.xml
#配置mysql数据库
<configuration>
    <property>
      <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://主机名:3306/hive?
createDatabaseIfNotExist=true&amp;useSSL=false</value>
      <description>JDBC connect string for a JDBC metastore</description>
    </property>
<property>
      <name>javax.jdo.option.ConnectionDriverName</name>
      <value>com.mysql.jdbc.Driver</value>
      <description>Driver class name for a JDBC metastore</description>
    </property>
    <property>
      <name>javax.jdo.option.ConnectionUserName</name>
      <value>root</value>
      <description>username to use against metastore database</description>
    </property>
    <property>
      <name>javax.jdo.option.ConnectionPassword</name>
      <value>root</value>
      <description>password to use against metastore database</description>
    </property>
</configuration>
```

- 驱动包

  mysql驱动包导入到hive的lib目录

三，HIVE交互方式

- hive shell命令行

```
bin/hive
执行完sql语句后，换是停留在shell命令行。必须在服务器使用。
```

- 服务器-客户端方式

```
1.服务器端启动hive服务：bin/hiveserver2
2.在客户端使用beeline连接hive服务
bin/beeline
!connect jdbc:hive2://服务器主机名:10000
username
password
```

- 一次性的hive命令行

```
bin/hive -e 'sql语句'
执行完sql语句就回到虚拟机
```

四，映射结构化数据

- 默认的映射结构化数据

```
1.数据
1^Azhangsan
2^Alisi
2.create table t_0 (id int,name string) row format delimited fields terminated by '^A';
3.加载数据
hadoop fs -put t0.txt /user/hive/warehouse/zxing.db/t_0
load data local inpath '/root/t0.txt' [overwrite]into table t_0;
```

五，分区

- 静态分区

```
为什么分区？避免全表检索，提高查询效率。分区字段是一个虚拟字段，不存储数据。在hdfs中只是在表的映射目录下创建子目录。
1.建表
create table t_name (id int ,...) partitioned by(p_name string，p_name_1 string) row format delimited terminated by '分隔符';
2.加载数据
load data [local] inpath 'data path' into table t_name
partition(p_name='xxx',p_name_1='xxx')
3.查询
select * from t_name where p_name='xxx' and p_name_1='xxx'
4.增加分区
alter table t_name add partition (p_name='xx') location
'/user/hive/warehouse/zxing.db/t_name/p_name=xxx'  partition (p_name='xx') location
/user/hive/warehouse/zxing.db/t_name/p_name=xxx'
5.删除分区
alter table t_name  drop if exists partition (p_name='xxx')
6.修改分区
alter table t_name partition(p_name='xx') rename to partition(p_name='xx_new');
```

- 动态分区

```
1.打开动态分区功能，指定分区模式
set hive.exec.dynamic.partition；--查看值
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode；--查看值
set hive.exec.dynamic.partition.mode=nonstrict;--动态的
2.创建原始表
create table dynamic_table (day string,food string) row format delimited fields terminated
by ',';

3.原始表加载数据
load data local....
4.创建分区表即目标表
create table d_t_t (food string) partitioned by (month string ,day string);
5.动态插入
insert overwrite  table  d_t_t partition(month ,day) select food ,substr(day,1,7) as month,
day from dynamic_table;
```

六，分桶|分簇

```
为什么？在表关联的时候，根据关联字段进行分桶，然后关联，避免笛卡尔积，提高查询效率。
在做测试的时候用分桶也方便。分桶操作和MR程序分区是等价的类似于设置reducetask的个数，然后自定义分区。
1.打开分桶功能，设置分桶数量
sql: set hive.enforce.bucketing =true;
sql: set mapreduce.job.reduces=4;
2.创建原始表
create table t_name(no string ,...) row format ....
3.原始表加载数据
load data [local] inpath 'data path'....
4.创建分桶表
create table t_name (no string,...) clustered by no into 4 buckets row format ....
5.插入分桶
insert overwrite t_name select * from t_name cluster by no;
```

七，外部表

```
为什么？数据过大，移动不方便，所以创建外部表和数据目录映射。
创建外部表
create external table t_name(id int,...) row format ....;
load data [local] inpath 'xxx'...
创建外部表-1
create external table t_name(id int,...) row format ....location 'xx';
内部表和外部表的区别：
1.删除数据的是内部表对应的目录及目录下的数据会一并删除。外部表不会。
2.创建内部表的时候系统会为表创建映射目录，外部表不会。内部表受到hive的管理，外部表不会。
```

八，修改表

```
1.添加列
alter table t_name add columns (c_name type);
2.重建列
alter table t_name replace columns(c_name1 type,c_name2 type);
3.修改列
alter table t_test change no no_1 int;  --修改列名
alter table t_test change no no_1 string after name;  --修改列名，数据类型，位置
alter table t_test change name name_1 string first  --修改列名，位置
alter table t_test rename to t_test_new;     --修改表名
```

九，insert，select，export，join

```
1.多重插入
from source_table
insert overwrite table t_insert1
select no
insert overwrite table t_insert2
select name;
2.多重导出
from t_name
insert overwrite local directory 'xxx' select c_name,..
insert overwrite [local] directory 'xxx' select c_name,..;
3.打开本地运行自动模式，sql语句就不会转换mr程序。用于开发。
set hive.exec.mode.local.auto=true;
inner join
left join
right join
full outer join
left semi join
cross join 慎用

select * from t_student s s.no in (select no from t_student_1 where ..)
                                ||
select * from t_student s left semi join t_student_1 s1 on s.no=s1.no
```

十，配置项优先级

```
1.session
2.服务器实例
3.全局级别，比如配置文件hive-site.xml
```

十一，常量表

```
1.create table dual (id string);
2.创建数据，只有一行，就是空格
3.加载数据
4.测试函数
```

十二，UDF

```
1.创建maven项目
2.配置pom.xml,配置hive依赖的包，hadoop的common，hive-exec
3.写代码继承UDF类，其中写方法
问题：UDF类找不到
    a.到cloudera库中下载前四个包
http://repository.cloudera.com/artifactory/public/org/apache/hive/hive-exec/0.13.1-cdh5.3.6/
    b.找到本地仓库D:\JAVA\repositories\org\apache\hive\hive-exec\0.13.1-cdh5.3.6，删除目录下的所有文件
    c.把下载好的包拷贝到这个目录
    d.更新maven项目。项目右键--maven--update project
4.打包，pom.xml右键--Run as --maven install
5.上传jar包至虚拟机
6.添加jar包至hive的类路径，add jar /root/xxx.jar
7.create temporary function  zf_name as 'com.zxing...类路径'
8.测试函数
select zf_name('');
```

十三，特殊字符

```
create table t_regex1 (id string,name string)  row format serde 'org.apache.hadoop.hive.serde2.RegexSerDe'  with serdeproperties('input.regex' = '(.*)\\|\\|(.*)', 'output.format.string' = '%1$s %2$s') stored as textfile;
```