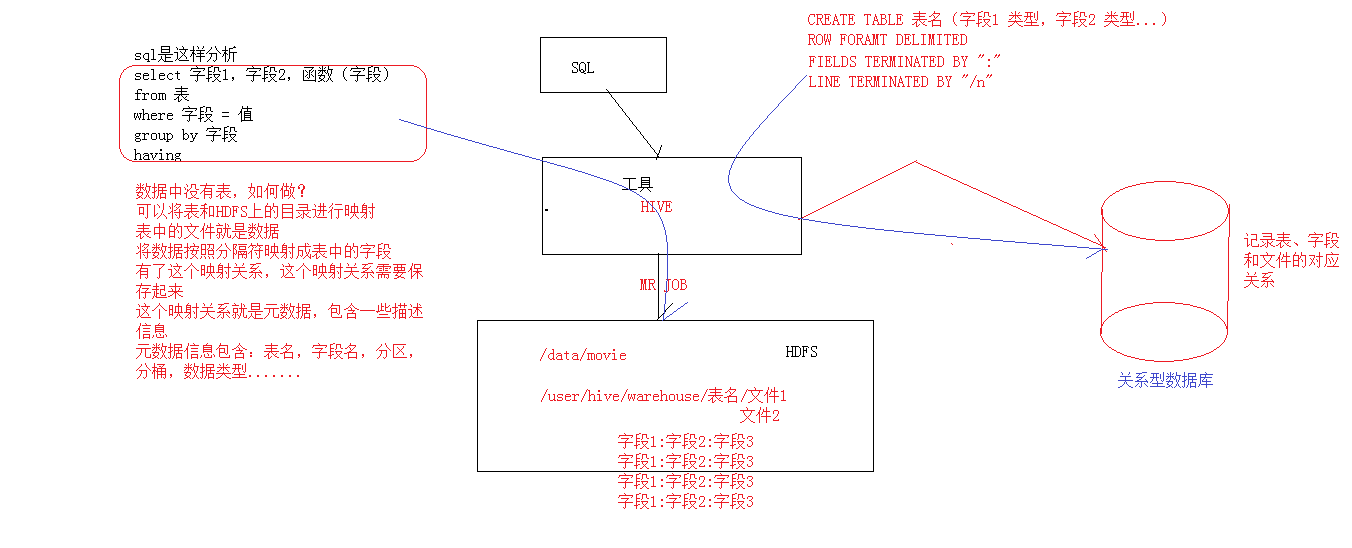
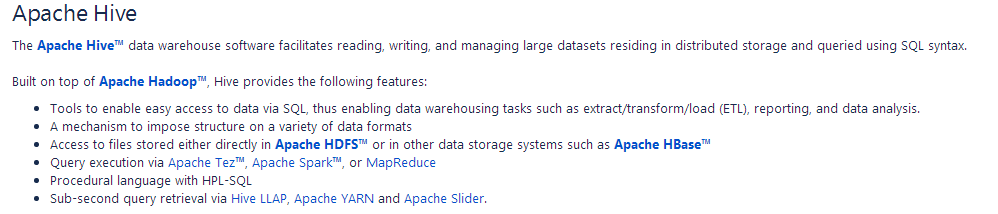
1. Hive是什么



* 1. 是一个工具，可以将结构化的数据映射成一张数据库表，提供了类sql语句，并能将语句翻译成MR，在hadoop框架中执行
  2. 特性
     1. 对数据进行抽取、转换、加载（ETL）的工具
     2. 提供了一种将格式化数据加上映射
     3. 可以分析存储在HDFS或者HBASE上的数据
     4. 查询经由tez，spark，MR完成
     5. 支持存储过程
     6. 通过yarn和slider实现亚秒级的查询搜索



1. Hive的架构
   1. 基本组成
      1. 用户接口：cli、JDBC/ODBC、webGUi
      2. thriftServer:第三方服务，JDBC/ODBC通过这个服务发布sql语句
      3. 元数据存储：通常存储在关系型数据库中（数据和表的映射关系）
      4. 解析器，编译器，优化器，执行器
   2. 各组件的功能
      1. 用户接口:cli为shell命令行；jdbc是hive的Java实现，odbc和jdbc类似，webhui通过浏览器访问hive
      2. 元数据存储：hive将元数据存储在关系型 数据库中，元数据包括：表名、字段名、分区及其属性、表的属性（内部表和外部表）、表的数据所在的目录
      3. Driver:解析器，将hql语句生成抽象表达式树
      4. 编译器：对hql语句进行词法、语法、语义的翻译，编译完成后会形成一个有向无环的执行计划
      5. 优化器：将计划进行优化，减少不必要的列、或者是分区的使用
      6. 执行器：将优化后的执行计划交给mr去运行
   3. Hive与hadoop的关系
      1. hive基于hadoop
      2. Hive没有本身的存储和计算
      3. 存储依赖于HDFS或者hbase
      4. Hive的计算依赖于MR
   4. Hive与关系型数据的关系
      1. 存储：hive扩展性很强，SQL可以集群存储，但是存在容量上限，数据量越大，计算能力越低
      2. 执行引擎：hive 依赖于MR，SQL有自己的执行引擎
      3. 分析速度：大数据量的情况下hive更高效
      4. 执行延迟：hive 的执行延迟高
      5. 数据加载模式：hive 是读时模式（快），sql是写时模式，即写入数据时检查数据内容
      6. 数据操作：hive是数据覆盖和追加，SQL行级更新删除
      7. 应用场景：海量数据查询（OLAP），实时查询（OLTP）
2. Hive的安装
   1. 内嵌模式

使用内嵌的derby数据库存储元数据，默认就是内嵌模式。运行hive会在当前目录下生成一个derby文件和一个metastore\_db目录，同一个目录只能有一个hive的客户端使用数据库，适合做测试，不适合生产环境

优点：使用简单，无需配置

缺点：只支持单session

* 1. 本地模式

使用本地安装的MYSQL替代derby，使用mysql来存储元数据，hive的服务和元数据服务运行在同一个jvm中，mysql是 单独的进程，可以是与hive同一台机器，也可以在远程的其他机器上。需要将mysql的驱动包拷贝到$HIVE\_HOME/lib目录下，并启动mysql服务

Hive的启动：

1. 配置好环境变量使用hive命令启动
2. 使用客户端工具beeline启动
   1. 先启动beeline，然后使用！Connect jdbc：hive2://mini1:10000.接下来输入mini1上启动hive服务的用户名和密码
   2. 使用beeline直连 beeline -u jdbc:hive2://mini1:10000
   3. 远程模式

本地元存储和远程元存储的区别：本地元存储不需要单独器metastore服务，用的是跟hive的服务在同一个进程里的metastore服务，远程元存储需要单独启动metastore服务，然后每个客户端都在配置文件里配置连接到这个metastore服务，远程元存储的metastore服务和hive运行在不同的进程里

远程模式的启动

1. 在服务端启动hive的元数据服务 hive --service metastore &
2. 在客户端配置好hive-site.xml之后，直接使用hive命令启动
3. Hive的基本命令
   1. Hive有一个默认的数据库，叫default,操作hql不指定数据库的情况下，默认使用default
   2. Hive的数据库名，表名不区分大小写
   3. 命名规范
      1. 名字不能用数字开头
      2. 不能使用关键字
      3. 尽量避免使用特殊字符
   4. Hive的操作
      1. 创建hive的库

Create database [if not exists] qf1919 comment ‘this is mydatabase’

创建库的本质是在/user/hive/warehouse/目录下创建一个目录（库名.db）

* + 1. 切换库 use qf1919;
    2. 创建表：默认是创建到当前的数据库；
       1. 创建表的本质就是在对应的库的下面创建一个目录
       2. 数据就可以存放在表所对应的目录中
    3. 创建表并定义分隔符

create table if not exists default.u4(

uname string comment 'this is name',

chinese int,

math int,

english int

)

comment 'this is table'

row format delimited

fields terminated by '\t'

lines terminated by '\n'

stored as textfile

;

* + 1. 为hive加载数据
       1. 直接把数据放在表所对应的目录下
       2. 使用load命令去加载数据
          1. 从本地linux机器上加载数据（本质上是复制）

Load data local inpath ‘本地路径’ [overwrite] into table 表名

* + - * 1. 从HDFS文件系统加载数据（本质上是移动）

Load data inpath ‘HDSFS路径’ [overwrite] into table 表名

* 1. 从加载数据的情况我们可以看出hive是读时模式

1. Hive的内部表和外部表
   1. 内部表：也称为管理表，表目录会创建在HDFS的仓库目录（/user/hive/warehouse）的相应的库的目录下
   2. 外部表：外部表会根据创建表时location关键字指定的路径创建表目录。（如果没有指定location，则表的创建的位置和内部表相同,如果指定就在指定的位置创建目录）
   3. 内部表和外部表的创建，外部表实际上就是两个关键字的问题 EXTERNAL LOCATION
   4. 创建内部表和外部表
      1. 创建内部表

Create table if not exists t1(

Id int,

Name varchar(20),

Sex int

)

;

* + 1. 创建外部表

Create external table if not exists t2(

Id int,

Name varchar(20),

Sex int

)

;

Create external table if not exists t3(

Id int,

Name varchar(20),

Sex int

)

Location “/t3”

;

* 1. 内部表和外部表的区别
     1. 在hive中创建表会做两件事情：第一，在HDFS上创建对应的目录，第二，在元数据库中创建对应表的描述数据（元数据）
     2. 内部和外部表在删除时有不同的特性：
        1. 内部表在删除时会删除元数据及在HDFS上对应的目录及目录中的文件。
        2. 外部表在删除时会删除元数据，不会删除HDFS上的对应的目录，更不会删除文件
        3. 外部表实际上就是两个关键字的问题 EXTERNAL LOCATION
  2. 使用场景：每天大量的网站的日志数据，这些日志数据要做大量的分析工作，我们可以使用外部表映射，方便数据的共享。在做统计分析时，一些中间结果我们可以使用内部表去管理，这些数据不需要共享
  3. Hive的建表语法

CREATE [EXTERNAL] TABLE [IF NOT EXISTS] table\_name

[(col\_name data\_type [COMMENT col\_comment], ...)]

[COMMENT table\_comment]

[PARTITIONED BY (col\_name data\_type [COMMENT col\_comment], ...)]

[CLUSTERED BY (col\_name, col\_name, ...)

[SORTED BY (col\_name [ASC|DESC], ...)] INTO num\_buckets BUCKETS]

[ROW FORMAT row\_format]

[STORED AS file\_format]

[LOCATION hdfs\_path]

* 1. Hive的默认的分隔符
     1. 行分隔符‘\n’
     2. 列分隔符：/001(八进制) ctrl v + ctrl a
  2. 加载数据的几种方式
     1. 将数据移动到表对应的目录中
     2. 使用load加载数据
        1. 本地加载
        2. 集群加载

create external table if not exists log(

`timestamp` string comment 'this is id',

phonenumber string,

mac string,

ip string,

url string,

sitetype string,

uppackage int,

downpackage int,

upflow int,

downflow int,

status int

)

comment 'this is log'

row format delimited

fields terminated by '\t'

lines terminated by '\n'

stored as textfile

location '/flow'

;

create external table if not exists log\_1(

`timestamp` string comment 'this is id',

phonenumber string,

mac string,

ip string,

url string,

sitetype string,

uppackage int,

downpackage int,

upflow int,

downflow int,

status int

)

comment 'this is log'

row format delimited

fields terminated by '\t'

lines terminated by '\n'

stored as textfile

;

* + 1. Insert into table log\_1 select \* from log where upflow < 1000;
    2. 克隆表不带数据 create table if not exists log\_2 like log\_1;
       1. 查看表的描述信息
          1. Desc 表名
          2. Describe extended 表名
          3. Show create table 表名
       2. 查看数据库描述信息
          1. Desc database qf1919;
          2. desc database extended qf1919;
    3. 克隆表带数据

create table if not exists log\_3

like

log\_2

location ‘/user/hive/warehouse/log\_2’;

##location，后面接的一定是HDFS上的目录而不是文件

* + 1. 克隆表带数据 更灵活的更常用的方式(跟创建表的方式一样，元数据和目录都会创建)

Create table if not exists log\_5

As

Select \* from log

Where

upflow > 1000

;