Mysql  Hive

——使用sqoop向星环hive中导入mysql数据example

首先，星环现阶段使用的sqoop是v1.44.3版本，不可以远程调用，但是支持直接从mysql导入hive。

Sqoop -> hive 过程介绍

1. **从关系型数据库中读取数据到hdfs**

Sqoop 先根据import命令中设定的参数来连接指定的的数据库，获取数据库、表的相关信息（要想连接到不同的数据库呢，就要把其对应的jdbc Driver copy到sqoop的lib目录下面）。然后一行一行的把表中的数据读取出来，存到HDFS中。这个过程是并行的，根据表的主键，或者某个指定的分区字段来分发工作，所以会生成多个文件。其中，关系型数据库当中的表，即可以是实实在在地table，也可以是view，甚至可以是select 语句生成的结果。

1. **从hdfs加载到hive**

Sqoop的实现方式是直接命令行调用hive load file 把文件加载到hive中。 可以指定是否overwrite掉原来的表，如果要overwrite那么原来的表就被覆盖了，如果不指定就直接把生成的文件copy到hive对应的目录，这里如果文件有重名会被覆盖掉。Hive文件存储组织方式不属于这里的介绍范围，请自行查找。 所以想要在非hive命令行环境中直接调用sqoop是不现实的。所以这里使用命令行来进行数据导入的演示。

示例

这里使用星环的hive(transwarp) 作为目标端，使用任意的mysql数据库作为数据源。

之后简化书写，约定星环的hive(transwarp) 直接写为hive。

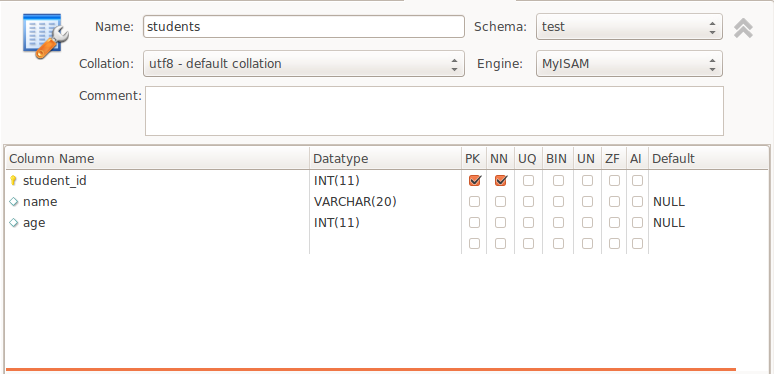
Hive cli 打开方式，先登录10.1.65.10，用户root，密码123456

命令行输入hive就可以打开hive的客户端

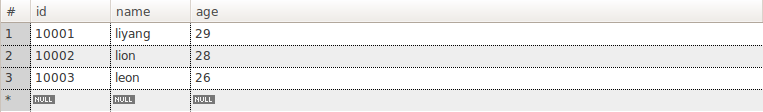


至于数据源，为了简单，做了2张简单的表如图所示

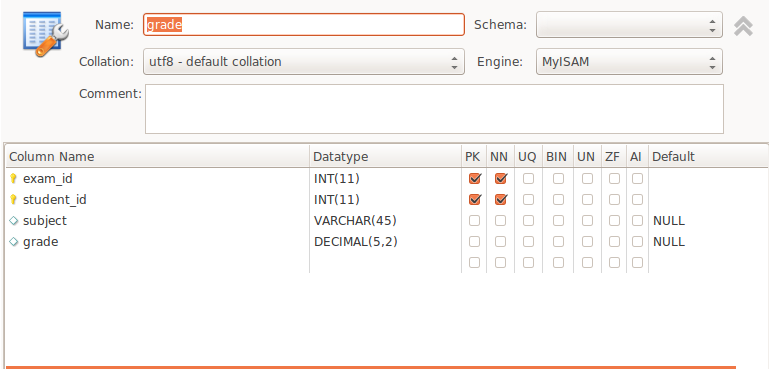
学生表：



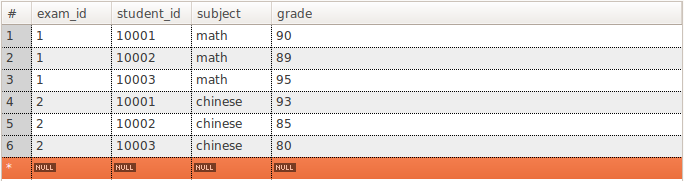
里面有几条简单的数据



成绩表：



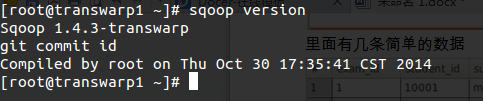
里面有几条简单的数据



下面将根据这些内容进行导入。

再打开一个命令行窗口，登录到10.1.65.10

输入sqoop version，查看sqoop是否已经成功地安装



可以看到sqoop已经成功地安装好了。

1. 单表导入

命令格式是：

sqoop import \

--hive-import --hive-table student --hive-drop-import-delims --hive-overwrite \

--connect jdbc:mysql://10.1.64.95:3306/test \

--username kettle --password whoami --table students

说明一下，为了直观， 分为多行书写。

其中sqoop import代表这是一个数据导入的任务

仔细看一下这个后面的参数基本都是 --参数名 [参数值] 这里的每个参数之间并没有固定的排列顺序，只有参数值一定要跟在参数名后面，但是为了便于理解，维护还是要把功能详尽的的参数放到一起最好。

--hive-import 这个参数指定了这是一个hive的数据导入任务

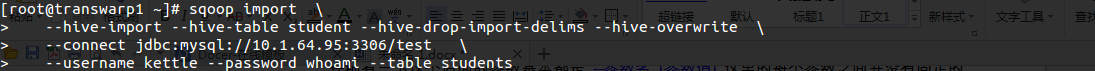
--hive-table student 这个参数指定要把数据导入到student这张表中

--hive-drop-import-delims 这个参数指定遇到string时要删掉\n, \r, \01，因为这些字符可能与hive使用的分隔符冲突。

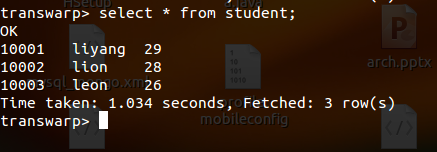
--hive-overwrite 这个参数指定如果已经有这张表了就把它覆盖掉

另当hive中的target表不存在时，可以指定--create-hive-table 来新建一张表，但是如果这个表名已经存在了，导入任务就会失败

直接把上面的命令copy过去，可以执行。(如果第一次尝试，并且hive中对应的表已经存在，请先删除掉)



查看hive中的student表，数据也都已经更新



————————————————————————————————————————

这样一次把表里面的所有的列都导入进去了，可不可以只选择几个列进行导入呢？

--columns就是做这个的

这次，只选择学生的id和姓名, 注意把下面命令中的引号改为英文字符

sqoop import \

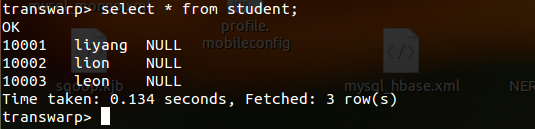
--hive-import --hive-table student --hive-drop-import-delims --hive-overwrite \

--connect jdbc:mysql://10.1.64.95:3306/test \

--username kettle --password whoami --table students \

--columns “student\_id,name”

完成后查看hive中的数据



发现数据会更新，但是少了的那一列的定义还是存在的，这是因为，这一次的导入只选择了两列，所以生成的数据文件里面也只有两列数据，但是hive中student的定义还是3列的，所以第三列都是NULL了。如果想这里不是NULL，只能修改table的定义。或者在导入数据之前先删除表。

—————————————————————————————————————————

列可以控制了，行怎么办呢？

--where <condition> 可以实现这一个功能，注意把下面命令中的引号改为英文字符

sqoop import \

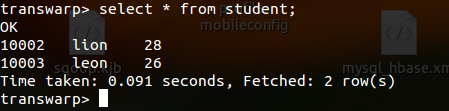
--hive-import --hive-table student --hive-drop-import-delims --hive-overwrite \

--connect jdbc:mysql://10.1.64.95:3306/test \

--username kettle --password whoami --table students \

--where “student\_id > 10001”

之后再查看hive中的表



第一行的数据已经没有了。

1. 多表导入

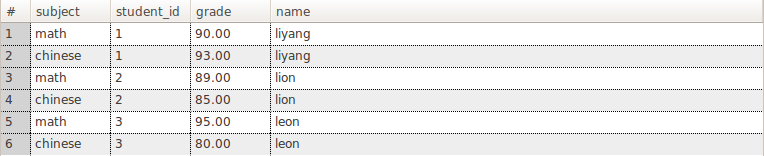
上面只是导入了一个数据库的一张表到hive， --query可以实现一个数据库中的好几张表对应到hive中的一张表。不过用了--query，--table 、--columns就不能再用了， --target-dir也一定要规定好，因为sqoop也不知道该用什么来命名存放中间生成文件的目录了，--split-by 也要明确，因为sqoop要并发很多任务。

这里把students与grade join导入hive中

Sql 语句是（到这一步时，在mysql中写join对不上，于是把student的id改为简单的1,2,3 方便debug，所以后面的id就都变成1,2,3了）:

use test;

select a.subject,a.student\_id,a.grade,b.name from grade as a right join students as b using(student\_id)



对应的命令是，注意先在hive中把原来生成的student表删除掉。

sqoop import \

--hive-import --hive-table student --hive-drop-import-delims --hive-overwrite \

--connect jdbc:mysql://10.1.64.95:3306/test \

--username kettle --password whoami \

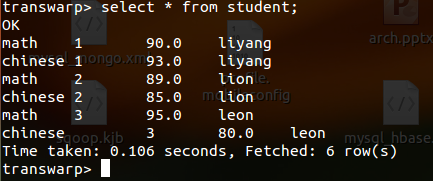
--query 'select a.subject,a.student\_id,a.grade,b.name from grade as a right join students as b using(student\_id) WHERE $CONDITIONS' \

--target-dir student\_grade \

--split-by a.student\_id

注意WHERE $CONDITIONS' 这里的 $CONDITIONS 一定要存在，因为这里要被 --where替换掉

在hive中查看新的数据



1. 增量解决方案
2. 注意到前面where 参数支持选择某些行进行导入。
3. 如果不指定overwrite 新旧数据就会共存。

进阶，使用hive的partition把数据分的更细，减少删除、更新带来的影响。

另外，sqoop也有一个增量导入的模式，需要指定 --incremental 或者--append，然后指定--check-column和--last-value，选择从某一列的某一行开始。但是不建议使用。因为主要是用于sqoop job。如果想要实验请参照sqoop文档。

**Sqoop import进阶**

sqoop的import工具会运行一个mapreduce作业，该作业会连接Mysql数据库，默认情况下，该作业会并行适用4个map任务来加速导入过程，每个任务都会将其所导入的数据写到一个单独的文件，但所有4个文件位于同一个目录之中。可以通过指定-m 1（sqoop1）来只得到一个文件

在默认情况下，sqoop会将我们导入的数据保存为逗号分隔的文本，如果导入数据的字段内容中存在分隔符，我们可以另外指定分隔符，字段包围字符和转义字符。这里需要注意，如果数据源里面有个String已经包含了”,”那么使用默认的配置将会使得后续的解析工作无法顺利进行

为了获取更好的导入性能，人们经常将查询划分到多个节点上执行。查询是根据一个partition column来进行划分的，sqoop需要选择一个合适的列作为划分列（通常是primary key），主键列中的最小值和最大值会被读出，与目标任务数一起确定每个map任务要执行的查询。

假设一个表中有100000条记录，其id列的值为0 ~ 99999 在导入这张表时，sqoop会发出一个类似SELECT MIN(id),MAX(id) FROM TABLENAME的查询语句，检索出的数据将用于对整张表的划分。假设有5个map任务，那么，这5个map的查询范围就是[0,20000),[20000,40000),[40000,60000),[60000,80000),[80000,100000)。

附

这里的说明比较简单，官方的说明最全面。

Sqoop 官方文档<http://sqoop.apache.org/docs/1.4.3/SqoopUserGuide.html>

同时在测试时发现，星环上的hive在使用select \* from where CONDITION 时，总是提示少包，请查询星环的hive（transwarp）使用文档，是语句改变了还是配置少了。

附附

这里只说明了使用sqoop到hive，其实到hbase的配置大部分与hive是一致的，参照文档稍作修改就可达成目标。我在这里花大量时间说明的也是如何配置数据源头。所以这里不打算继续写hbase了。

Bryce