***Import:***

***sqoop-help***

***sqoop import --connect jdbc:mysql://cdh-slave2/db --help***

***sqoop export --connect jdbc:mysql://cdh-slave2/db --help***

***sqoop-list-databases --connect jdbc:mysql://cdh-slave2:3306/ --username root --password root***

***sqoop-list-tables --connect jdbc:mysql://cdh-slave2:3306/hive --username root --password root***

***将关系型数据的表结构复制到hive中：***

***sqoop create-hive-table --connect jdbc:mysql://cdh-slave2:3306/hive --username root --password root --table TBLS --hive-database t1 --hive-table tbls***

***mysql >>>>> hive:(不用提前创建hive中对应的表，导入后表字段类型与mysql一致)***

***1）默认数据库及表名***

***sqoop import --connect jdbc:mysql://cdh-slave2:3306/hive --username root --password root --table TBLS --hive-import***

***2）指定数据库、表名、指定列(列是mysql表中的列)***

***1.数据库、表名、指定列***

***sqoop import --connect jdbc:mysql://cdh-slave2:3306/hive --username root --password root --table TBLS --columns TBL\_ID,CREATE\_TIME,DB\_ID --hive-database t1 --hive-table tbls2 --hive-import***

***sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person --columns id,name,age,address --hive-database weiwc\_sqoop --hive-table tbls2 --hive-import***

***2. 使用--direct直接连接关系型数据库进行优化：***

***sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person --columns id,name,age,address \***

***--hive-database weiwc\_sqoop --hive-table tbls\_direct --hive-import --direct***

***3. mysql中的空值导入到hive中：***

***sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person --hive-database weiwc\_sqoop --hive-table tbls3 --hive-import***

***注意：如果Hive中有空值的话，在Impala中进行查询分析时，会出现类型转换的错误！***

***总结：1. Mysql中的数据类型为varchar，为空值的话，导入Hive后：对应的是字符串null（在HDFS界面看到的是null字符串），查询类似使用如下：***

***select count(\*) from tbls3 where address = "null";***

***2. Mysql中的数据类型为数值型（bigint,int,float,double）为空值的话，导入Hive后：对应的是NULL（在HDFS界面看到的也是null字符串），查询类似使用如下：***

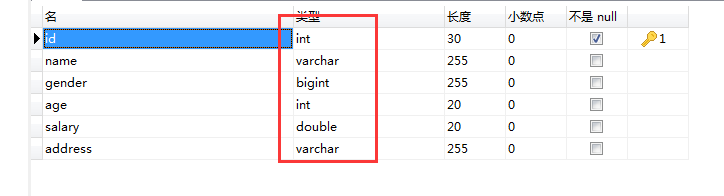
***select count(\*) from tbls3 where age is null;***

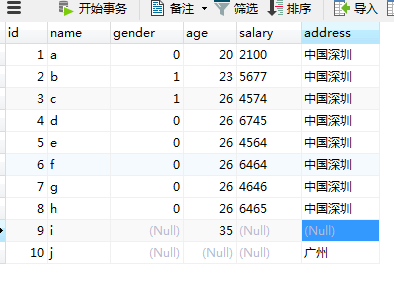
***select count(\*) from tbls3 where gender is null;***

***select count(\*) from tbls3 where salary is null;***

***MYsql和hive的表结构、字段类型、空值对比如下图所示：***

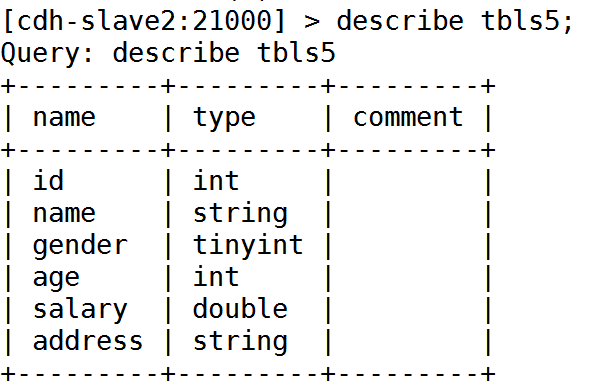
Mysql数据及表类型：

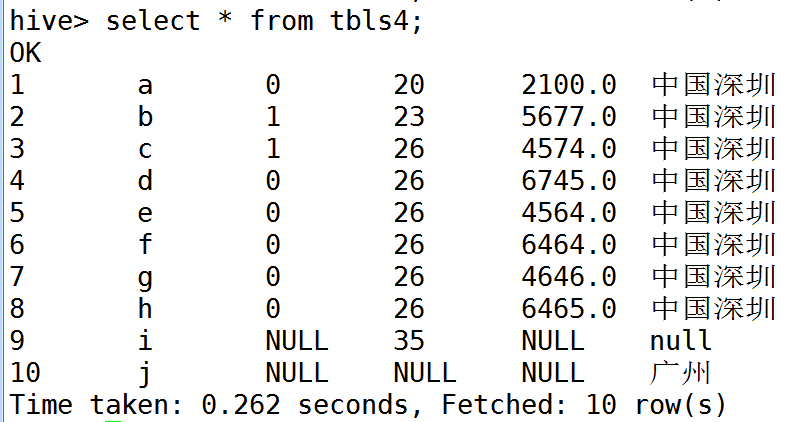




Hive中对应表结构及数据：







=============================================================================================================

***mysql >>>>> hbase:*** 参数--hbase-create-table会创建没有的表和列族。如果Mysql的表中有空值，则对应空值(不管其字段类型是数值或varchar)的列将不会插入Hbase.

1.默认情况下sqoop.hbase.add.row.key为false,意思是指定为rowkey的列不再作为数据插入到单元格中，可设置为true，即可作为数据插入。

sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person2 \

--hbase-table sqoop --hbase-row-key id --column-family data --hbase-create-table

2.再次导入data2列族，并设置：-D sqoop.hbase.add.row.key=true，注意该参数位置要放在最前边，如下命令所示：

sqoop import -D sqoop.hbase.add.row.key=true --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person2 \

--hbase-table sqoop --hbase-row-key id --column-family data2 --hbase-create-table

================

*注意：如果Hive中有空值的话，在Impala中进行查询分析时，会出现类型转换的错误！*

总结：Mysql中的数据类型为varchar，为空值的话，导入Hive后：对应的是字符串null（在HDFS界面看到的是null字符串），查询类似使用如下：

select count(\*) from tbls3 where address = "null";

Mysql中的数据类型为数值型（bigint,int,float,double）为空值的话，导入Hive后：对应的是NULL（在HDFS界面看到的也是null字符串），查询类似使用如下：

select count(\*) from tbls3 where age is null;

select count(\*) from tbls3 where gender is null;

select count(\*) from tbls3 where salary is null;

**说明：--null-string '\\N' --null-non-string '\\N'**

import时，指定\N代替mysql/oracle中的NULL存储到HDFS中（因为hive中的NULL在HDFS上就是\N,因此MYsql中哪些数据是空值与hive中判断要保持一致（不能有的：is null 有的：=="null"），

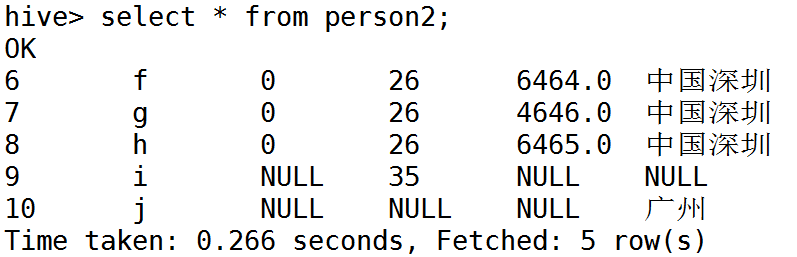
以免hive表通过运算，出现既有null代表空也有\N代表空的情况，这样导出时就无法统一指定为只有\N是NULL），默认数值或varchar都使用字符串'null'代替，在HDFS上可看到，但查询出来的是：数值是NULL，

因此查询出来的是NULL的话，HDFS上可能有多种情况：比如，\N，null,原始数据load时的空值（什么都没有）查询出来也会是NULL！

1. 增量导入：--check-column id --incremental append --last-value 5

空值替换：--null-non-string '\\N' --null-string '\\N'，如此varchar类的空值也变为NULL.

sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person2 --check-column id --incremental append --last-value 5 --null-non-string '\\N' --null-string '\\N' --hive-database weiwc\_sqoop --hive-table person2 --hive-import





1. --query和--columns、--where：

官网示例：--query 'SELECT a.\*, b.\* FROM a JOIN b on (a.id == b.id) WHERE $CONDITIONS' -m 1 --target-dir /user/foo/joinresults

1.使用--query导出到Hive:必须有--target-dir，--split-by，其中--query中的select中必须有$CONDITIONS'条件，导出到hive时--target-dir，指定的目录会创建但没有用（但必须指定），数据还是会导入指定的Hive表、及对应路径中。

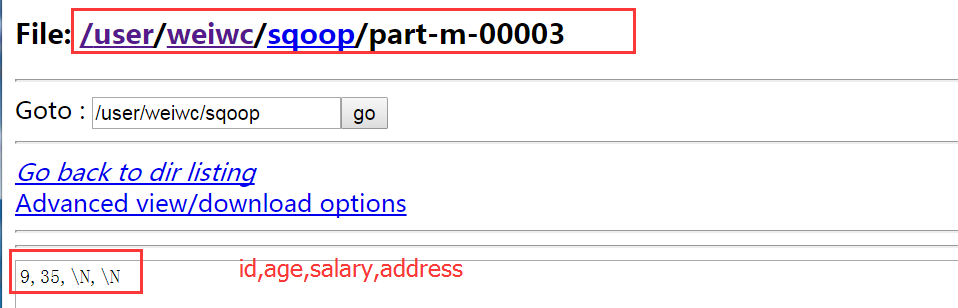
使用了--query就不能同时指定参数：--table、--columns、--where。

sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --query 'select id,age,salary,address from person2 where id > 5 and $CONDITIONS' --target-dir /user/weiwc/sqoop --split-by id --null-non-string '\\N' --null-string '\\N' \

--hive-database weiwc\_sqoop --hive-table person3 --hive-import

2.使用--query导出到HDFS: 此时不再指定hive相关的参数，数据会导入--target-dir /user/weiwc/sqoop目录，以，分割。

sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --query 'select id,age,salary,address from person2 where id > 5 and $CONDITIONS' --target-dir /user/weiwc/sqoop --split-by id --null-non-string '\\N' --null-string '\\N'



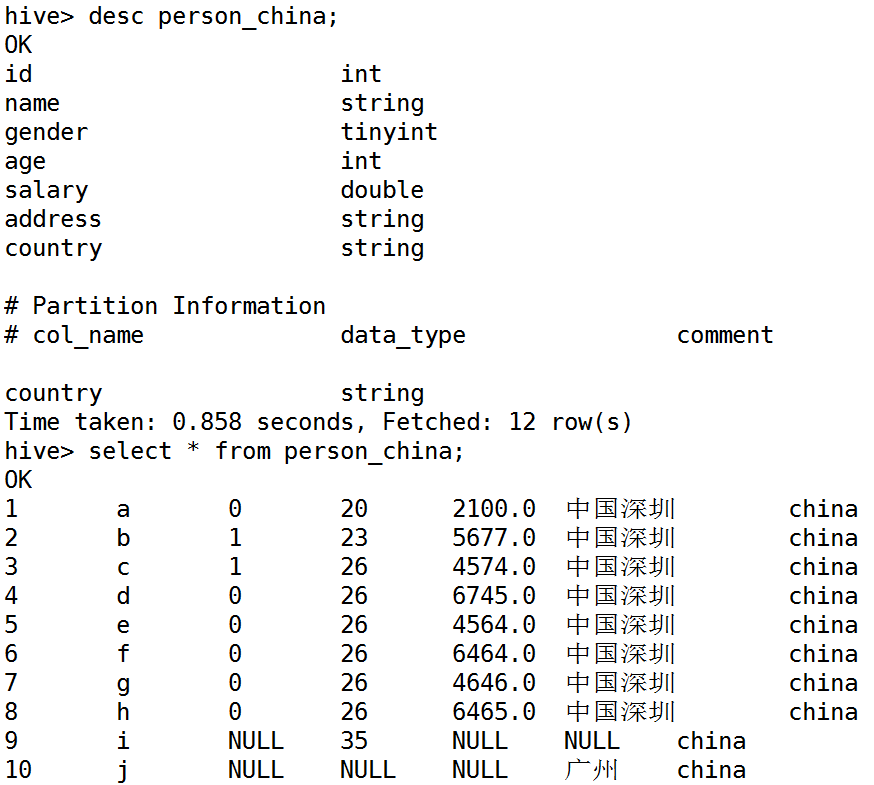
5）追加导入到HDFS（不支持导入到Hive）：--append，新的数据会生成新的几个文件存放在共同目录下,不支持导入到Hive，只支持导入到HDFS。

sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person2 --target-dir /user/weiwc/sqoop --null-non-string '\\N' --null-string '\\N' --append

6）导入到Hive指定分区：参数--hive-partition-key country --hive-partition-value china

sqoop import --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person2 --null-non-string '\\N' --null-string '\\N' \

--hive-database weiwc\_sqoop --hive-table person\_china --hive-partition-key country --hive-partition-value china --hive-import



***Export:***

**hive >>>>mysql:**

将hive中的表数据导入到mysql中注意的几点:

1、需提前创建表e1,e2,e3,e4，表字段类型与hive一致。

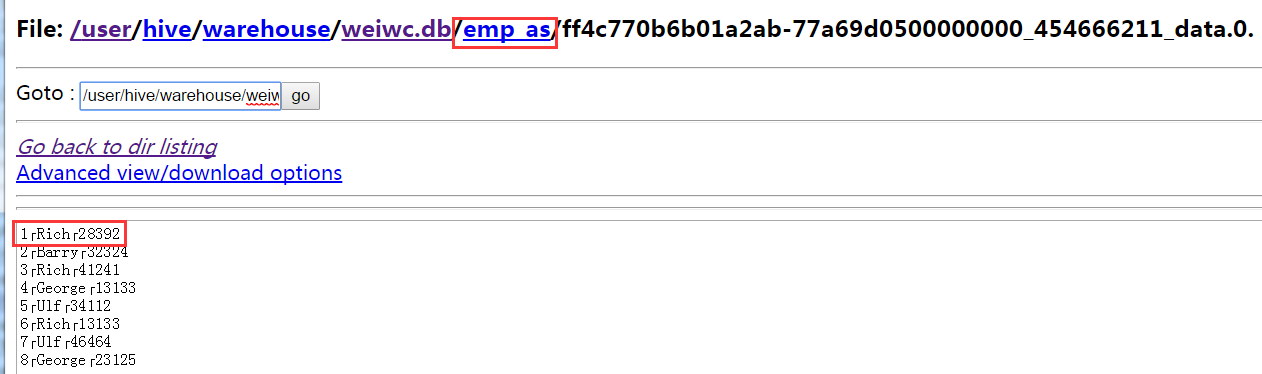
2、hive中分隔符问题：1. 直接load导入的数据，以文本数据即建表时指定的分隔符为准。2.由任务产生的表（类似create as ），为Hive默认分隔符：'\001',总之以HDFS上看到的真实数据分隔符为准！

**Load导入数据分隔符：**



通过执行任务生成的表分隔符：

create table emp\_as as select \* from emp;



1)数据分隔符：'\t'和'\001'

sqoop export --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table e1 --export-dir /user/hive/warehouse/weiwc.db/emp --input-fields-terminated-by '\t' ;

sqoop export --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table e2 --export-dir /user/hive/warehouse/weiwc.db/emp\_as --input-fields-terminated-by '\001' ;

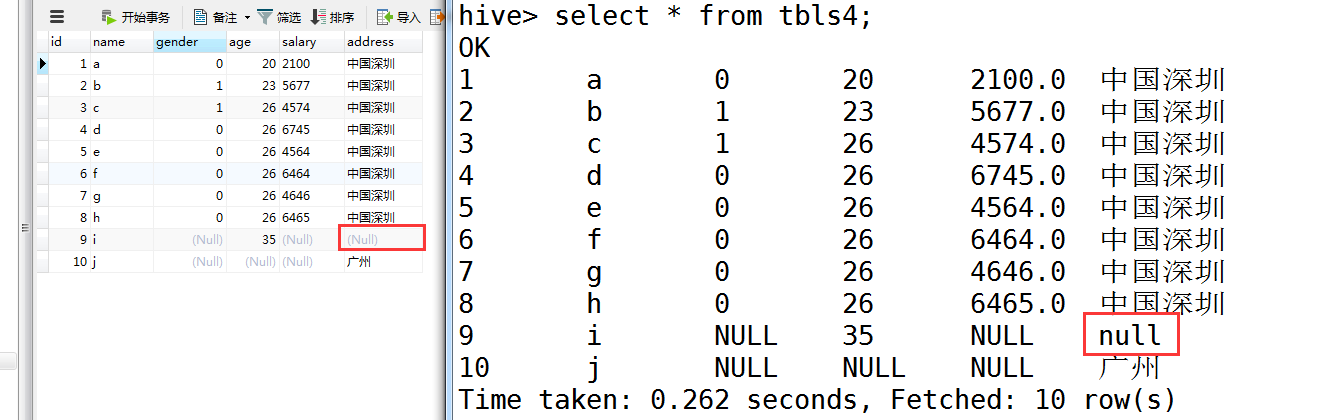
2)指定列导出：(--columns grade2,losal2,这里的列名指的是mysql表中的列名，而非hive)：

sqoop export --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table e3 --columns id2,name2,salary2 --export-dir /user/hive/warehouse/weiwc.db/emp\_as --input-fields-terminated-by '\001' ;

--columns id2,name2,salary2这里的字段，是HDFS文件分割后的映射字段。如果指定--columns id2,salary2，则导出的结果其实为：hive中的前两个字段映射到mysql中的第一和第三个字段。

3）Hive中的空值导出到Mysql,,此时Hive表中的NULL值其实为\N（import时做过转换）,将hive表person2导出到Mysql的person3中,结果如图，命令如下：

sqoop export --connect jdbc:mysql://cdh-slave2:3306/test --username root --password root --table person2 --export-dir /user/hive/warehouse/weiwc\_sqoop.db/tbls4 --input-fields-terminated-by '\001' ;



--input-null-string '\\N' --input-null-non-string '\\N'

说明：这两个参数意思是：当export数据时，当值为\N时就认为是NULL，将其作为NULL存储到mysql/oracle中，默认是把值为'null'当作NULL，因此当import

数据时把Mysql中的空（string列和非string列），使用--null-string '\\N' --null-non-string '\\N'进行存储为'\N',因此当导出时就把值为'\N'认为是空。