Hive数据库

Hive中的数据库概念，本质上仅仅是表的一个目录结构或者命名空间

Hive表

Hive中的表概念，本质上仅仅是数据的一个目录结构或命名空间

分区管理表、分区外部表

创建分区表的最主要目的是：快速查询

1. 创建库

create database + 库名;



==查看成功!!!

1. 使用库

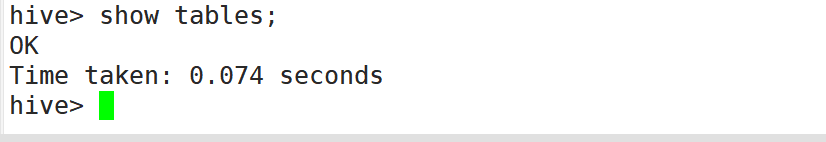
use 库名;



==使用成功！！！

1. 查看表

show tables;



==查看成功！！！

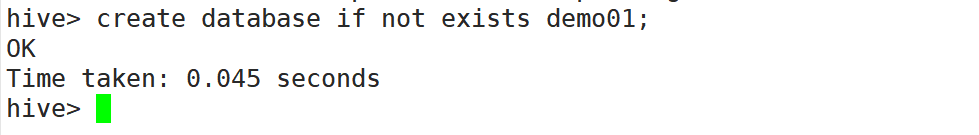
1. 创建库

create database if not exists + 库;

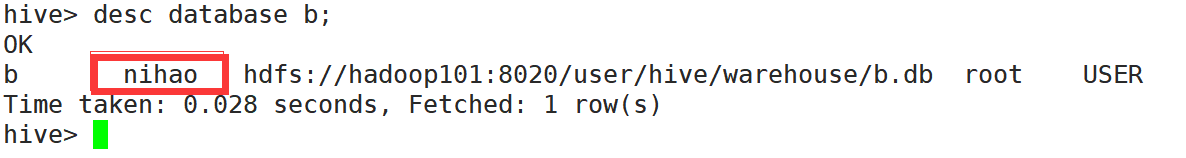
create database if not exists log;

Comment:注释

create database if not exists + 库名 + comment +'注释信息';



==创建库成功！！！

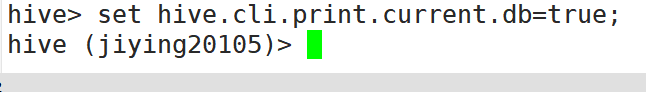


==查看备注的细节信息成功！！！

备注：在查看细节的时候可以看到

1. 显示当前所在库

set hive.cli.print.current.db=true;

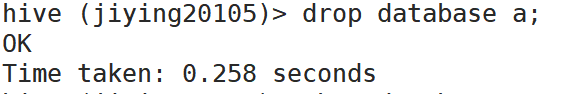


==显示成功！！！

1. 删除库

Drop database + 库名;

Drop databases a;

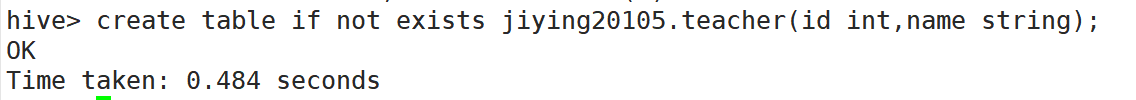


==输出成功！！！

1. 创建表

Crete table if not exists + 库名.表名（名称 属性，名称 属性）;

create table if not exists jiying20105.teacher(id int,name string);



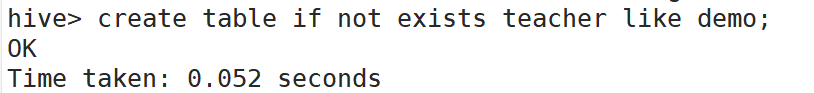
创建成功！！！

3-1.创建表

Like 只创建跟已有表一样的结构:类型、字段

Create table if not exists +要创建的表名+ like +已经有的表名；

create table if not exists teacher like student;



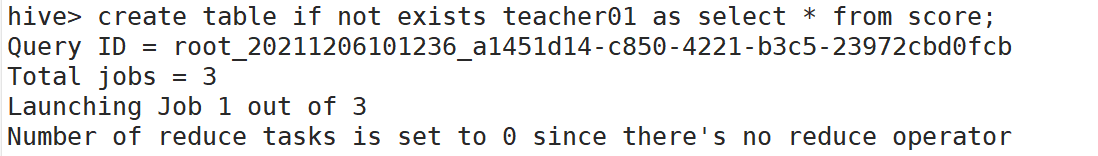
==创建成功！！！

3-2.创建表

As 创建表结构 并且 复制数据

Create table if not exists +要创建的表名+ as + select \* from +表名；

create table if not exists teacher01 as select \* from score;

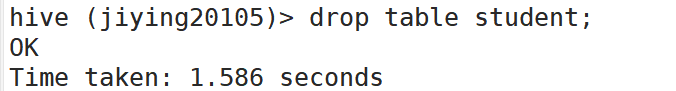


==创建成功！！！

1. 删除表

drop table +表名；

Drop table student;



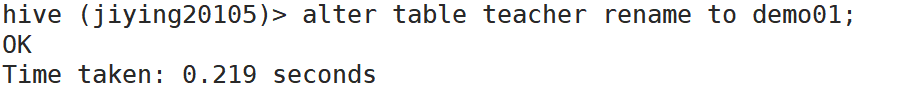
==删除成功！！！

1. 修改表名

关键字 alter

alter table + 原表名 + rename to + 要修改的表名;

alter table teacher rename to demo01;

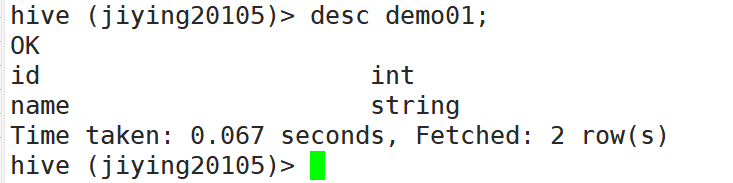


==修改成功！！！

1. 查看表的细节

Desc + 表名

Desc demo01；

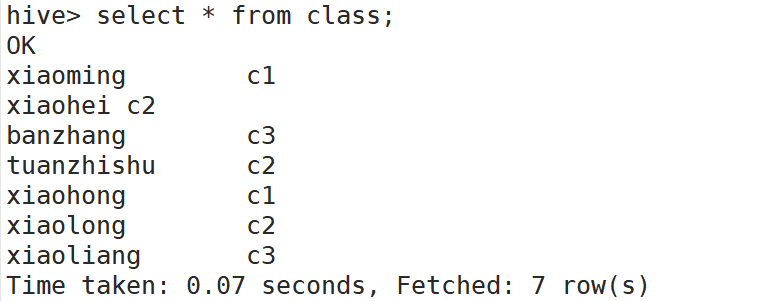


==查看成功！！！

1. 查看数据

Select \* from + 表名；

Select \* from class;



==查看成功！！！

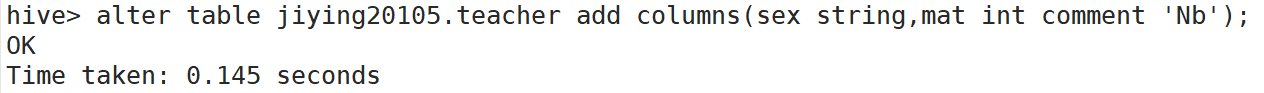
1. 增加字段

Columns 列

Comment 备注

alter table 库名.表名 add columns(字段名 类型,字段名 类型 comment '备注信息');

alter table student add columns(sex string,mat int comment 'Nb');

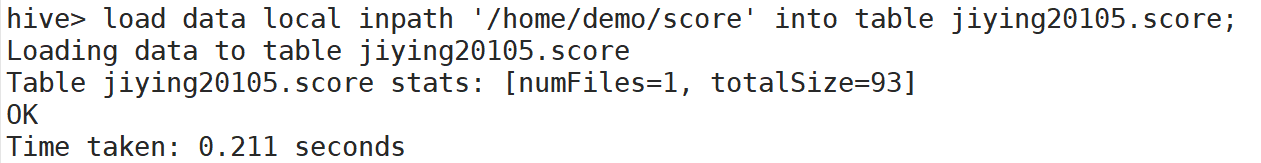


==增加成功！！！

1. 加载本地数据

Load data local inpath ‘本地文件路径’into table 库名.表名；

load data local inpath '/demo/score' into table jiying20105.score;

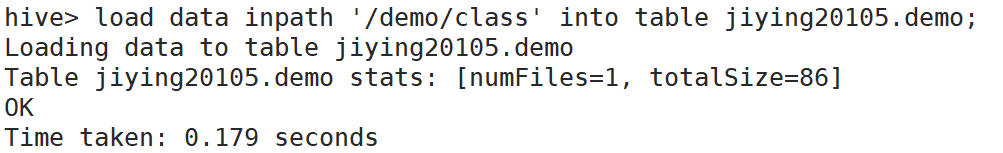


==加载成功！！！

1. 加载HDFS上的数据

Load data inpath ‘HDFS文件路径’into table 库名.表名

load data inpath '/demo/class' into table jiying20105.demo;



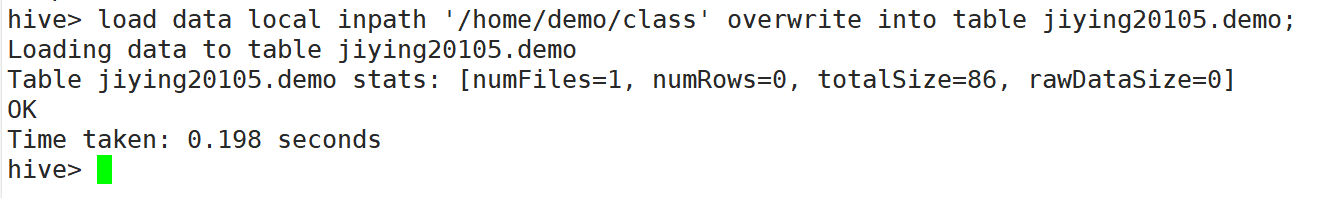
==加载成功！！！

1. 加载本地数据并覆盖

Overwrite 覆盖

load data local inpath '本地路径' overwrite into table 库名.表名;

load data local inpath '/home/demo/student' overwrite into table ji20105.student;

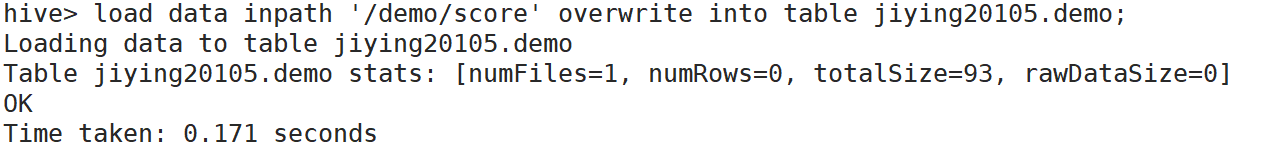


==覆盖成功！！！

1. 加载并覆盖HDFS数据

load data inpath 'HDFS路径' overwrite into table 库名.表名;

load data local inpath '/home/demo/student' overwrite into table ji20105.student;



==覆盖成功！！！

基本数据类型：

int 4byte有符号整数

bigint 8byte有符号整数

boolean 布尔类型，true或false

float 单精度浮点型

double 双精度浮点型

string 字符序列

复杂数据类型

struct 通过对元素的first来指定数据的元素

map <key,value>以键值对的形式存储

array 下标从0开始存储

分隔符：

\n 换行符

^A 分隔符与\t 类似

^B 用来分割array和struct的元素

^C 用来map中的键值对之间的分隔

hive元数据的存储：

存储在关系型数据库中，默认存储在derby，内嵌在hive里面

实际工作中我们使用最多的是mysql

因为我们可以将hive的元数据存储在mysql里面不放在derby数据库中。

------------------------------------------------------------练习1.-----------------------------------------------------------

create table jiying20105.demo01(name string,kemu1 string,cj1 double,kmeu2 string,cj2 double)

row format delimited fields terminated by ' '

stored as textfile;

加载本地数据

load data local inpath '/input/score' into table jiying20105.demo01;

--数据--

易孟滔 hadoop 59 mysql 94

贺曼雄 hadoop 76 mysql 79

文胜华 hadoop 74 mysql 47

周楚钟 hadoop 73 mysql 89

梁子敬 hadoop 67 mysql 69

唐勇 hadoop 63 mysql 84

黄凯 hadoop 86 mysql 46

颜超斌 hadoop 95 mysql 76

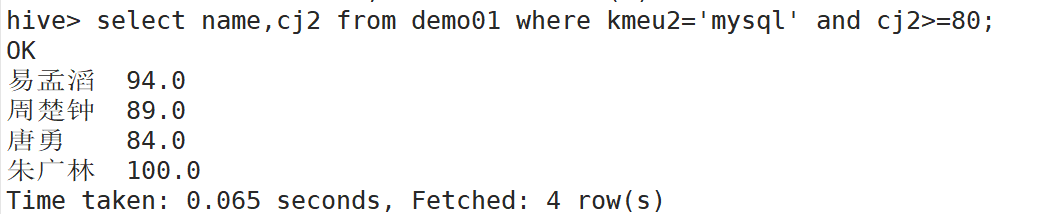
廖书明 hadoop 78 mysql 78

朱广林 hadoop 48 mysql 100

查看数据

select \* from ji20105.score;

需求1：求出mysql成绩在80分以上的学生姓名和成绩



需求2：求出成绩总和前三名学生

select name,

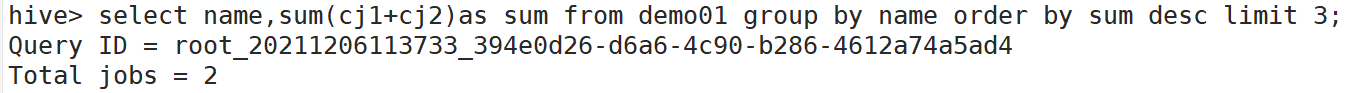
sum(cj1+cj2) as a

from score

group by name

order by a desc

limit 3;



==练习1结束！！！

----------------------------------------------join连接-----------------------------------------------------

注释：

a.\*,b.\* 可以单独列出来写，比单独的\*号作用大

Select \* from 不可以单独列出来写，这个是连接两个

1. 内连接

内链接 Inner join ...on

Select \* from 表名 + inner join + 表名 on（注释：相同的部分）表名.名称 = 表名.名称（一样的）

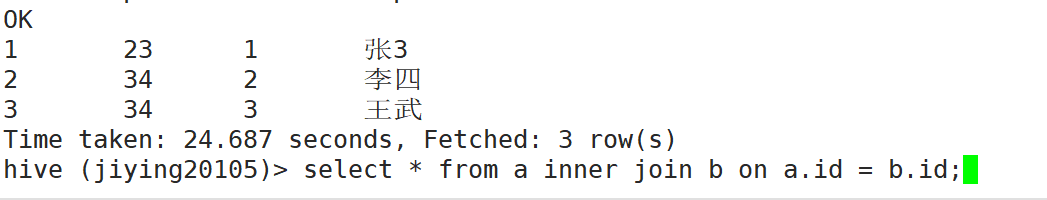
select \* from a inner join b on a.id = b.id;（b表和a表id一样）

或者↓

select a.\*,b.\* from a inner join b on a.id = b.id;



或者↓



备注：内连接是连接他们主表中相同的部分，不同的不会显示出来

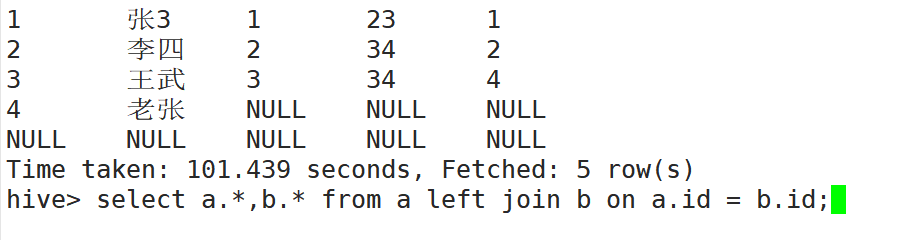
==连接成功！！！

1. 左连接

左连接 Left join ...on

Select 表名.属性名,表名.属性名from 主表名 + left join + 副表名 on（注释：相同的部分）表名.名称 = 表名.名称（一样的）

select a.\*,b.\* from a left join b on a.id=b.id;



备注：左连接是连接他们主表相同的部分，不同的显示为空

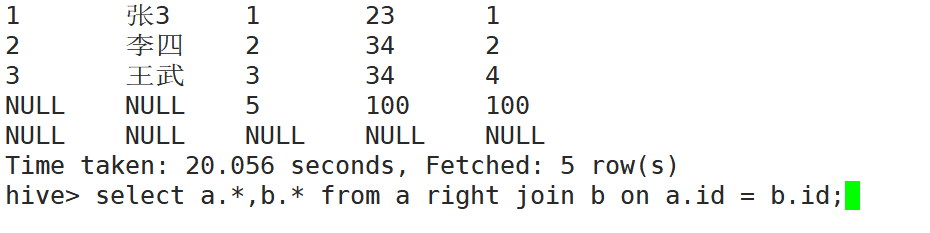
==连接查看成功！！！

1. 右连接

右连接 Right join ...on

Select 表名.属性名,表名.属性名from 主表名 + right join + 副表名 on（注释：相同的部分）表名.名称 = 表名.名称（一样的）

select a.\*,b.\* from a right join b on a.id=b.id;



备注：右连接是连接副表相同的部分，不同的显示为空

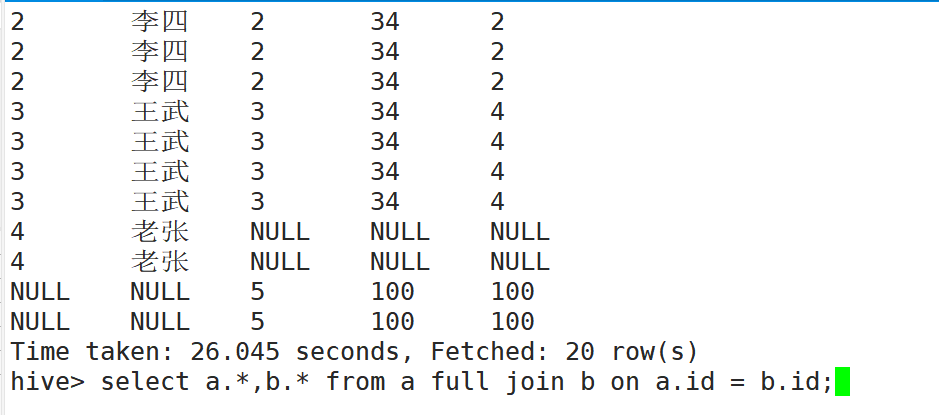
==连接查看成功！！！

1. 全连接

全连接 Full join ...on

Select 表名.属性名,表名.属性名from 主表名 + full join + 副表名 on（注释：相同的部分）表名.名称 = 表名.名称（一样的）

select a.\*,b.\* from a full join b on a.id=b.id;



备注：全连接是显示连接主、副表的所有内容，不同的显示为空

==连接查看成功！！！

-----------------------------------------------------------练习2-------------------------------------------------------------

数据

a表：(id int,name string)

1,张3  
2,李四  
3,王武  
4,老张

b表：(id int,job\_id int,num int)

1,23,1  
2,34,2  
3,34,4  
5,100,100

c表：(job\_id int,job string)

23,开车  
33,开火车  
34,开飞机

建表语句

create table if not exists ji20105.a(id int,name string)

row format delimited fields terminated by ','lines terminated by '\n'

stored as textfile;

create table if not exists ji20105.b(id int,job\_id int,num int)row format delimited fields terminated by ','lines terminated by '\n'stored as textfile;

create table if not exists ji20105.c(job\_id int,job string)row format delimited fields terminated by ','lines terminated by '\n'stored as textfile;

加载数据

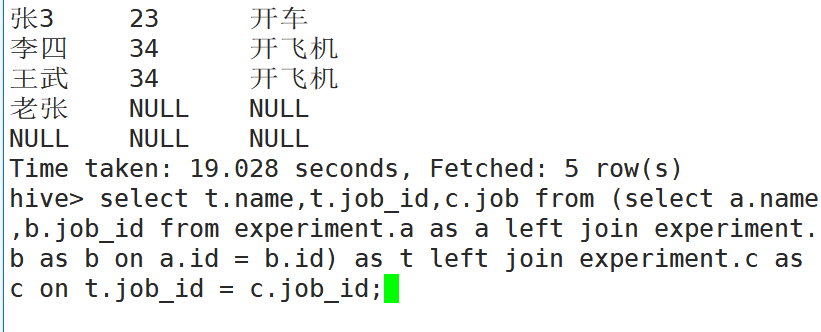
load data local inpath '/home/demo/a' into table ji20105.a;

load data local inpath '/home/demo/b' into table ji20105.b;

load data local inpath '/home/demo/c' into table ji20105.c;

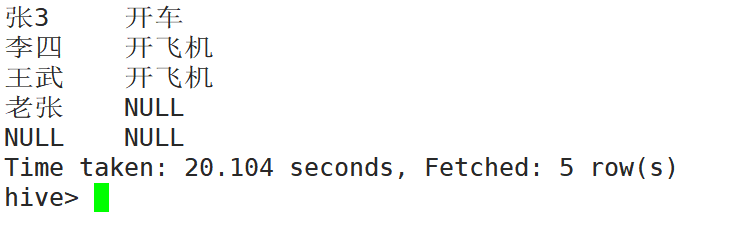
需求1. 查出人名、年龄和工作

select t.name,t.job\_id,c.job from (select a.name,b.job\_id from experiment.a as a left join experiment.b as b on a.id = b.id) as t left join experiment.c as c on t.job\_id = c.job\_id;



需求2. 查出人名和工作

select t.name,c.job from(select a.name,b.job\_id from a left join b on a.id=b.id)as t left join c on t.job\_id = c.job\_id;

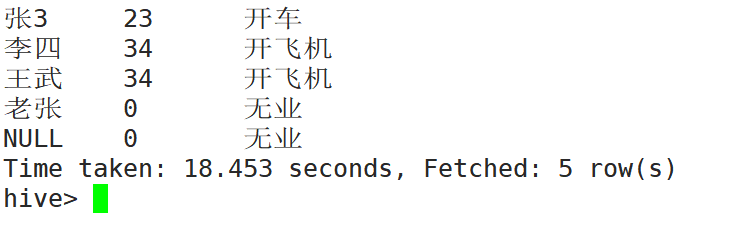


需求3. 职业为空的提示“无业”

select a.name,if(b.job\_id is null,0,b.job\_id),if(c.job is not null,c.job,'无业')

from experiment.a as a left join experiment.b as b on a.id = b.id

left join experiment.c as c on b.job\_id=c.job\_id;



select a.name,

if(b.job\_id is null,'无业','有工作')

from a left join b

on a.id=b.id

left join c

on b.job\_id=c.job\_id;

--------------------------------

person:(name string,station int)

ticket:(station int,price int)

--建表

create table if not exists ji20105.person(

name string,

station int

)

row format delimited fields terminated by '\t'

lines terminated by '\n'

stored as textfile;

create table if not exists ji20105.ticket(

station int,

price int

)

row format delimited fields terminated by '\t'

lines terminated by '\n'

stored as textfile;

--加载数据

load data local inpath '/home/demo/person' into table ji20105.person;

load data local inpath '/home/demo/ticket' into table ji20105.ticket;

--按照站数算对应价格 --每个人车票多少钱

select p.name,t.price

from person as p left join ticket as t

on p.station=t.station;

-- 10站以上打8折,15站7折

-- 一个判断 使用if

select p.name,

if(p.station>10,if(p.station>15,t.price\*0.7,t.price\*0.8),t.price)

from person as p left join ticket as t

on p.station=t.station;

--5站到10站打9折、11站以上打8折

--多个判断 case when ... then ... else ... end

select p.name,

case when t.station>11 then t.price\*0.8 else(

case when t.station>=5 then t.price\*0.9 else t.price end

)

end

from person as p left join ticket as t

on p.station=t.station;

---------------------练习3--------------------

class:(name string,class string)

score:(name string,chinese int,math int)

--创建表

create table if not exists ji20105.class(

name string,

class string

)

row format delimited fields terminated by ','

lines terminated by '\n'

stored as textfile;

create table if not exists ji20105.score1(

name string,

chinese int,

math int

)

row format delimited fields terminated by ','

lines terminated by '\n'

stored as textfile;

--加载数据

load data local inpath '/home/demo/class' into table ji20105.class;

load data local inpath '/home/demo/score1' into table ji20105.score1;

问题：计算每个班的语文总成绩和数学总成绩

select class,sum(chinese),sum(math)

from class as c left join score1 as s

on c.name=s.name

group by class;

计算每个班的语文总成绩

select class,sum(chinese)

from class as c left join score1 as s

on c.name=s.name

group by class;

问题：计算每个班的语文总成绩和数学总成绩，要求有哪科低于60分，该名学生成绩不计入计算.

--and 和 并且

select class,sum(chinese),sum(math)

from class as c left join score1 as s

on c.name=s.name

where chinese>60 and math>60

group by class;

---------------4--------------

创建班级表 my\_student(sno int,sname string,sex string,sage int,sdept string)

创建课程表 my\_course(cno int,cname string)

创建成绩表 my\_score(sno int,cno int,grade int)

create table if not exists ji20105.my\_student(

sno int,

sname string,

sex string,

sage int,

sdept string

)

row format delimited fields terminated by ','

lines terminated by '\n'

stored as textfile;

create table if not exists ji20105.my\_course(

cno int,

cname string

)

row format delimited fields terminated by ','

lines terminated by '\n'

stored as textfile;

create table if not exists ji20105.my\_score(

sno int,

cno int,

grade int

)

row format delimited fields terminated by ','

lines terminated by '\n'

stored as textfile;

--加载数据

load data local inpath '/home/demo/my\_student' into table ji20105.my\_student;

load data local inpath '/home/demo/my\_course' into table ji20105.my\_course;

load data local inpath '/home/demo/my\_score' into table ji20105.my\_score;

问题1：查询全体学生的学号与姓名

select sno,sname

from my\_student;

问题2：查询选修了课程的学生姓名

select sname

from my\_student as s left join my\_score as c

on s.sno=c.cno;

-- 使用 distinct去重 在select后面使用，distict后面跟要去重的字段

select distinct s1.sname

from ji20105.my\_student as s1 left join ji20105.my\_score s2

on s1.sno = s2.sno;

SELECT distinct sname FROM my\_student;

问题3：查询学生的总人数 count

SELECT count(sname) FROM my\_student;

问题4：计算1号课程的学生平均成绩

分析：所有学生的平均分数 要把所有学生分数加起来

select AVG(grade)

from my\_score

where cno=1;

问题5：查询选修1号课程的最高分数和学生姓名

--进行order by 的字段要出现在select里面

select s2.sname,s1.grade

from my\_score as s1 left join my\_student as s2

on s1.sno=s2.sno

where cno=1

order by s1.grade desc

limit 1;

select cno,max(grade),min(grade)

from ji20105.my\_score where cno=1 group by cno;

问题6：求各个课程号及相应的选课人数

select c.cno,count(s.sno)

from my\_course as c left join my\_score as s

on c.cno=s.cno

group by c.cno;

问题7：查询选修了4门以上的课程的学生学号

问题8：查询选修了3门以上的课程的学生学号 优化

load data local inpath '/home/demo/a overwrite into table experiment.a;

load data local inpath '/home/demo/b overwrite into table experiment.b;

load data local inpath '/home/demo/c overwrite into table experiment.c;

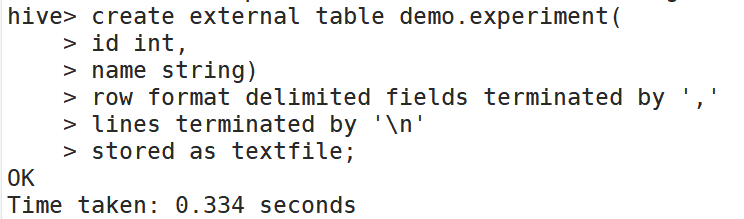
外部表和内部表的区别

内部表删除表后就什么都没有了

外部表在删除表后，数据在hdfs上面有存档，相当于就只要把外部表在创建回来，数据就依然还是在，都不用导入数据（表名要和原来的一样）

1. 创建外部表

External 关键字

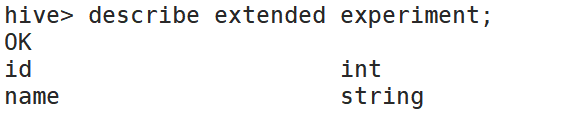


==查看成功！！！

1. 查看是外部表还是内部表

describe extended + 表名；

describe extended tablename；



备注：外部表在最后会显示

内部表则会在最后显示

==查看成功！！！