尚硅谷大数据之Hive SQL题库-初级

（作者：尚硅谷研究院）

版本：V1.0

# 第一章 环境准备

## 1.1 建表语句

hive>

**-- 创建学生表**

DROP TABLE IF EXISTS student;

create table if not exists student\_info(

stu\_id string COMMENT '学生id',

stu\_name string COMMENT '学生姓名',

birthday string COMMENT '出生日期',

sex string COMMENT '性别'

)

row format delimited fields terminated by ','

stored as textfile;

**-- 创建课程表**

DROP TABLE IF EXISTS course;

create table if not exists course\_info(

course\_id string COMMENT '课程id',

course\_name string COMMENT '课程名',

tea\_id string COMMENT '任课老师id'

)

row format delimited fields terminated by ','

stored as textfile;

**-- 创建老师表**

DROP TABLE IF EXISTS teacher;

create table if not exists teacher\_info(

tea\_id string COMMENT '老师id',

tea\_name string COMMENT '学生姓名'

)

row format delimited fields terminated by ','

stored as textfile;

**-- 创建分数表**

DROP TABLE IF EXISTS score;

create table if not exists score\_info(

stu\_id string COMMENT '学生id',

course\_id string COMMENT '课程id',

score int COMMENT '成绩'

)

row format delimited fields terminated by ','

stored as textfile;

## 1.2 数据准备

（1）创建/opt/module/data目录

[atguigu@hadoop102 module]$ mkdir data

（2）将如下4个文件放到/opt/module/data目录下

 

（3）数据样式说明

[atguigu@hadoop102 data]$ vim student\_info.txt

001,彭于晏,1995-05-16,男

002,胡歌,1994-03-20,男

003,周杰伦,1995-04-30,男

004,刘德华,1998-08-28,男

005,唐国强,1993-09-10,男

006,陈道明,1992-11-12,男

007,陈坤,1999-04-09,男

008,吴京,1994-02-06,男

009,郭德纲,1992-12-05,男

010,于谦,1998-08-23,男

011,潘长江,1995-05-27,男

012,杨紫,1996-12-21,女

013,蒋欣,1997-11-08,女

014,赵丽颖,1990-01-09,女

015,刘亦菲,1993-01-14,女

016,周冬雨,1990-06-18,女

017,范冰冰,1992-07-04,女

018,李冰冰,1993-09-24,女

019,邓紫棋,1994-08-31,女

020,宋丹丹,1991-03-01,女

[atguigu@hadoop102 data]$ vim course\_info.txt

01,语文,1003

02,数学,1001

03,英语,1004

04,体育,1002

05,音乐,1002

[atguigu@hadoop102 data]$ vim teacher\_info.txt

1001,张高数

1002,李体音

1003,王子文

1004,刘丽英

[atguigu@hadoop102 data]$ vim score\_info.txt

001,01,94

002,01,74

004,01,85

005,01,64

006,01,71

007,01,48

008,01,56

009,01,75

010,01,84

011,01,61

012,01,44

013,01,47

014,01,81

015,01,90

016,01,71

017,01,58

018,01,38

019,01,46

020,01,89

001,02,63

002,02,84

004,02,93

005,02,44

006,02,90

007,02,55

008,02,34

009,02,78

010,02,68

011,02,49

012,02,74

013,02,35

014,02,39

015,02,48

016,02,89

017,02,34

018,02,58

019,02,39

020,02,59

001,03,79

002,03,87

004,03,89

005,03,99

006,03,59

007,03,70

008,03,39

009,03,60

010,03,47

011,03,70

012,03,62

013,03,93

014,03,32

015,03,84

016,03,71

017,03,55

018,03,49

019,03,93

020,03,81

001,04,54

002,04,100

004,04,59

005,04,85

007,04,63

009,04,79

010,04,34

013,04,69

014,04,40

016,04,94

017,04,34

020,04,50

005,05,85

007,05,63

009,05,79

015,05,59

018,05,87

## 1.3 插入数据

（1）插入数据

hive>

load data local inpath '/opt/module/data/student\_info.txt' into table student\_info;

load data local inpath '/opt/module/data/course\_info.txt' into table course\_info;

load data local inpath '/opt/module/data/teacher\_info.txt' into table teacher\_info;

load data local inpath '/opt/module/data/score\_info.txt' into table score\_info;

（2）验证插入数据情况

hive>

select \* from student\_info limit 5;

select \* from course\_info limit 5;

select \* from teacher\_info limit 5;

select \* from score\_info limit 5;

# 第二章 简单查询

## 2.1 查找特定条件

### 2.1.1 查询姓名中带“冰”的学生名单

hive>

select

\*

from student\_info

where stu\_name like "%冰%";

结果

stu\_id stu\_name birthday sex

017 范冰冰 1992-07-04 女

018 李冰冰 1993-09-24 女

### 2.1.2 查询姓“王”老师的个数

hive>

select

count(\*) wang\_count

from teacher\_info

where tea\_name like '王%';

结果

wang\_count

1

### 2.1.3 检索课程编号为“04”且分数小于60的学生的课程信息，结果按分数降序排列

hive>

select

stu\_id,

course\_id,

score

from score\_info

where course\_id ='04' and score<60

order by score desc;

结果

stu\_id course\_id score

004 04 59

001 04 54

020 04 50

014 04 40

017 04 34

010 04 34

### 2.1.4 查询数学成绩不及格的学生和其对应的成绩，按照学号升序排序

hive>

select

s.stu\_id,

s.stu\_name,

t1.score

from student\_info s

join (

select

\*

from score\_info

where course\_id=(select course\_id from course\_info where course\_name='数学') and score < 60

) t1 on s.stu\_id = t1.stu\_id

order by s.stu\_id;

结果

s.stu\_id s.stu\_name t1.score

005 唐国强 44

007 陈坤 55

008 吴京 34

011 潘长江 49

013 蒋欣 35

014 赵丽颖 39

015 刘亦菲 48

017 范冰冰 34

018 李冰冰 58

019 邓紫棋 39

020 宋丹丹 59

# 第三章 汇总分析

## 3.1 汇总分析

### 3.1.1 查询编号为“02”的课程的总成绩

hive>

select

course\_id,

sum(score) score\_sum

from score\_info

where course\_id='02'

group by course\_id;

结果

course\_id score\_sum

02 1133

### 3.1.2 查询参加考试的学生个数

思路：对成绩表中的学号做去重并count

hive>

select

count(distinct stu\_id) stu\_num

from score\_info;

结果

stu\_num

19

## 3.2 分组

### 3.2.1 查询各科成绩最高和最低的分，以如下的形式显示：课程号，最高分，最低分

思路：按照学科分组并使用max和min。

hive>

select

course\_id,

max(score) max\_score,

min(score) min\_score

from score\_info

group by course\_id;

结果

course\_id max\_score min\_score

01 94 38

02 93 34

03 99 32

04 100 34

05 87 59

### 3.2.2 查询每门课程有多少学生参加了考试（有考试成绩）

hive>

select

course\_id,

count(stu\_id) stu\_num

from score\_info

group by course\_id;

结果

course\_id stu\_num

01 19

02 19

03 19

04 12

05 5

### 3.2.3 查询男生、女生人数

hive>

select

sex,

count(stu\_id) count

from student\_info

group by sex;

结果

sex count

女 9

男 11

## 3.3 分组结果的条件

### 3.3.1 查询平均成绩大于60分的学生的学号和平均成绩

**1）思路分析**

（1）平均成绩：展开来说就是计算每个学生的平均成绩

（2）这里涉及到“每个”就是要分组了

（3）平均成绩大于60分，就是对分组结果指定条件

（4）首先要分组求出每个学生的平均成绩，筛选高于60分的，并反查出这批学生，统计出这些学生总的平均成绩。

**2）Hql实操**

hive>

select

stu\_id,

avg(score) score\_avg

from score\_info

group by stu\_id

having score\_avg > 60;

结果

stu\_id score\_avg

001 72.5

002 86.25

004 81.5

005 75.4

006 73.33333333333333

009 74.2

013 61.0

015 70.25

016 81.25

020 69.75

### 3.3.2 查询至少选修四门课程的学生学号

**1）思路分析**

（1）需要先计算出每个学生选修的课程数据，需要按学号分组

（2）至少选修两门课程：也就是每个学生选修课程数目>=4，对分组结果指定条件

**2）Hql实操**

hive>

select

stu\_id,

count(course\_id) course\_count

from score\_info

group by stu\_id

having course\_count >=4;

结果

stu\_id course\_num

001 4

002 4

004 4

005 5

007 5

009 5

010 4

013 4

014 4

015 4

016 4

017 4

018 4

020 4

### 3.3.3 [课堂讲解]查询同姓（假设每个学生姓名的第一个字为姓）的学生名单并统计同姓人数大于2的姓

思路：先提取出每个学生的姓并分组，如果分组的count>=2则为同姓

hive>

select

t1.first\_name,

count(\*) count\_first\_name

from (

select

stu\_id,

stu\_name,

substr(stu\_name,0,1) first\_name

from student\_info

) t1

group by t1.first\_name

having count\_first\_name >= 2;

结果

t1.first\_name count\_first\_name

刘 2

周 2

陈 2

### 3.3.4 查询每门课程的平均成绩，结果按平均成绩升序排序，平均成绩相同时，按课程号降序排列

思路：按照课程号分组并求组内的平均值

hive>

select

course\_id,

avg(score) score\_avg

from score\_info

group by course\_id

order by score\_avg asc, course\_id desc;

结果

course\_id score\_avg

02 59.63157894736842

04 63.416666666666664

01 67.15789473684211

03 69.42105263157895

05 74.6

### 3.3.5 统计参加考试人数大于等于15的学科

按课程分组并统计组内人数，过滤条件大于等于15

hive>

select

course\_id,

count(stu\_id) stu\_count

from score\_info

group by course\_id

having stu\_count >= 15;

结果

course\_id stu\_count

01 19

02 19

03 19

## 3.4 查询结果排序&分组指定条件

### 3.4.1 查询学生的总成绩并按照总成绩降序排序

思路：分组、sum、排序

hive>

select

stu\_id,

sum(score) sum\_score

from score\_info

group by stu\_id

order by sum\_score desc;

结果

stu\_id sum\_score

005 377

009 371

002 345

004 326

016 325

007 299

001 290

015 281

020 279

013 244

010 233

018 232

006 220

014 192

017 181

012 180

011 180

019 178

008 129

### 3.4.2 [课堂讲解]按照如下格式显示学生的语文、数学、英语三科成绩，没有成绩的输出为0，按照学生的有效平均成绩降序显示

学生id 语文 数学 英语 有效课程数 有效平均成绩

hive>

select

si.stu\_id,

sum(if(ci.course\_name='语文',score,0)) `语文`,

sum(if(ci.course\_name='数学',score,0)) `数学`,

sum(if(ci.course\_name='英语',score,0)) `英语`,

count(\*) `有效课程数`,

avg(si.score) `平均成绩`

from

score\_info si

join

course\_info ci

on

si.course\_id=ci.course\_id

group by

si.stu\_id

order by

`平均成绩` desc

结果

学生id 语文 数学 英语 有效课程数 平均成绩

002 74 84 87 4 86.25

004 85 93 89 4 81.5

016 71 89 71 4 81.25

005 64 44 99 5 75.4

009 75 78 60 5 74.2

006 71 90 59 3 73.33333333333333

001 94 63 79 4 72.5

015 90 48 84 4 70.25

020 89 59 81 4 69.75

013 47 35 93 4 61.0

012 44 74 62 3 60.0

011 61 49 70 3 60.0

007 48 55 70 5 59.8

019 46 39 93 3 59.333333333333336

010 84 68 47 4 58.25

018 38 58 49 4 58.0

014 81 39 32 4 48.0

017 58 34 55 4 45.25

008 56 34 39 3 43.0

### 3.4.3 查询一共参加三门课程且其中一门为语文课程的学生的id和姓名

hive>

select

t2.stu\_id,

s.stu\_name

from (

select t1.stu\_id

from (

select stu\_id,

course\_id

from score\_info

where stu\_id in (

select stu\_id

from score\_info

where course\_id = "01"

)

) t1

group by t1.stu\_id

having count(t1.course\_id) = 3

) t2

join student\_info s on t2.stu\_id = s.stu\_id;

结果

t2.stu\_id s.stu\_name

006 陈道明

008 吴京

011 潘长江

012 杨紫

019 邓紫棋

# 第四章 复杂查询

## 4.1 子查询

### 4.1.1 [课堂讲解]查询所有课程成绩均小于60分的学生的学号、姓名

hive>

select s.stu\_id,

s.stu\_name

from (

select stu\_id,

sum(if(score >= 60, 1, 0)) flag

from score\_info

group by stu\_id

having flag = 0

) t1

join student\_info s on s.stu\_id = t1.stu\_id;

结果

s.stu\_id s.stu\_name

008 吴京

017 范冰冰

### 4.1.2 查询没有学全所有课的学生的学号、姓名

解释：没有学全所有课，也就是该学生选修的课程数 < 总的课程数

hive>

select

s.stu\_id,

s.stu\_name

from student\_info s

left join score\_info sc on s.stu\_id = sc.stu\_id

group by s.stu\_id, s.stu\_name

having count(course\_id) < (select count(course\_id) from course\_info);

结果

s.stu\_id s.stu\_name

001 彭于晏

002 胡歌

003 周杰伦

004 刘德华

006 陈道明

008 吴京

010 于谦

011 潘长江

012 杨紫

013 蒋欣

014 赵丽颖

015 刘亦菲

016 周冬雨

017 范冰冰

018 李冰冰

019 邓紫棋

020 宋丹丹

### 4.1.3 查询出只选修了三门课程的全部学生的学号和姓名

解释：学生选修的课程数 = 3

hive>

select

s.stu\_id,

s.stu\_name

from student\_info s

join (

select

stu\_id,

count(course\_id) course\_count

from score\_info

group by stu\_id

having course\_count =3

) t1

on s.stu\_id = t1.stu\_id;

结果

s.stu\_id s.stu\_name

006 陈道明

008 吴京

011 潘长江

012 杨紫

019 邓紫棋

# 第五章 多表查询

## 5.1 表联结

### 5.1.1 [课堂讲解]查询有两门以上的课程不及格的同学的学号及其平均成绩

① 先找出有两门以上不及格的学生名单，按照学生分组，过滤组内成绩低于60的并进行count，count>=2。

② 接着做出一张表查询学生的平均成绩并和上一个子查询中的学生学号进行连接

hive>

select

t1.stu\_id,

t2.avg\_score

from (

select

stu\_id,

sum(if(score < 60,1,0)) flage

from score\_info

group by stu\_id

having flage >= 2

) t1

join (

select

stu\_id,

avg(score) avg\_score

from score\_info

group by stu\_id

) t2 on t1.stu\_id = t2.stu\_id;

结果

t1.stu\_id t2.avg\_score

007 59.8

008 43.0

010 58.25

013 61.0

014 48.0

015 70.25

017 45.25

018 58.0

019 59.333333333333336

020 69.75

### 5.1.2 查询所有学生的学号、姓名、选课数、总成绩

hive>

select

s.stu\_id,

s.stu\_name,

count(sc.course\_id) count\_course,

sum(sc.score) sum\_score

from student\_info s

left join score\_info sc on s.stu\_id = sc.stu\_id

group by s.stu\_id,s.stu\_name;

结果

stu\_id stu\_name course\_count course\_sum

001 彭于晏 4 290

002 胡歌 4 345

003 周杰伦 0 0

004 刘德华 4 326

005 唐国强 5 377

006 陈道明 3 220

007 陈坤 5 299

008 吴京 3 129

009 郭德纲 5 371

010 于谦 4 233

011 潘长江 3 180

012 杨紫 3 180

013 蒋欣 4 244

014 赵丽颖 4 192

015 刘亦菲 4 281

016 周冬雨 4 325

017 范冰冰 4 181

018 李冰冰 4 232

019 邓紫棋 3 178

020 宋丹丹 4 279

### 5.1.3 查询平均成绩大于85的所有学生的学号、姓名和平均成绩

hive>

select s.stu\_id,

s.stu\_name,

avg(sc.score) avg\_score

from score\_info sc

left join student\_info s on s.stu\_id = sc.stu\_id

group by s.stu\_id, s.stu\_name

having avg\_score > 85

结果

stu\_id stu\_name avg\_score

002 胡歌 86.25

### 5.1.4 查询学生的选课情况：学号，姓名，课程号，课程名称

hive>

select

s.stu\_id,

s.stu\_name,

c.course\_id,

c.course\_name

from score\_info sc

join course\_info c on sc.course\_id = c.course\_id

join student\_info s on sc.stu\_id = s.stu\_id;

结果

|  |  |
| --- | --- |
| s.stu\_id s.stu\_name c.course\_id c.course\_name  001 彭于晏 01 语文  002 胡歌 01 语文  004 刘德华 01 语文  005 唐国强 01 语文  006 陈道明 01 语文  007 陈坤 01 语文  008 吴京 01 语文  009 郭德纲 01 语文  010 于谦 01 语文  011 潘长江 01 语文  012 杨紫 01 语文  013 蒋欣 01 语文  014 赵丽颖 01 语文  015 刘亦菲 01 语文  016 周冬雨 01 语文  017 范冰冰 01 语文  018 李冰冰 01 语文  019 邓紫棋 01 语文  020 宋丹丹 01 语文  001 彭于晏 02 数学  002 胡歌 02 数学  004 刘德华 02 数学  005 唐国强 02 数学  006 陈道明 02 数学  007 陈坤 02 数学  008 吴京 02 数学  009 郭德纲 02 数学  010 于谦 02 数学  011 潘长江 02 数学  012 杨紫 02 数学  013 蒋欣 02 数学  014 赵丽颖 02 数学  015 刘亦菲 02 数学  016 周冬雨 02 数学  017 范冰冰 02 数学  018 李冰冰 02 数学  019 邓紫棋 02 数学  020 宋丹丹 02 数学 | 001 彭于晏 03 英语  002 胡歌 03 英语  004 刘德华 03 英语  005 唐国强 03 英语  006 陈道明 03 英语  007 陈坤 03 英语  008 吴京 03 英语  009 郭德纲 03 英语  010 于谦 03 英语  011 潘长江 03 英语  012 杨紫 03 英语  013 蒋欣 03 英语  014 赵丽颖 03 英语  015 刘亦菲 03 英语  016 周冬雨 03 英语  017 范冰冰 03 英语  018 李冰冰 03 英语  019 邓紫棋 03 英语  020 宋丹丹 03 英语  001 彭于晏 04 体育  002 胡歌 04 体育  004 刘德华 04 体育  005 唐国强 04 体育  007 陈坤 04 体育  009 郭德纲 04 体育  010 于谦 04 体育  013 蒋欣 04 体育  014 赵丽颖 04 体育  016 周冬雨 04 体育  017 范冰冰 04 体育  020 宋丹丹 04 体育  005 唐国强 05 音乐  007 陈坤 05 音乐  009 郭德纲 05 音乐  015 刘亦菲 05 音乐  018 李冰冰 05 音乐  Time taken: 20.878 seconds, Fetched: 74 row(s) |

### 5.1.5 查询出每门课程的及格人数和不及格人数

hive>

select

c.course\_id,

c.course\_name,

t1.`及格人数`,

t1.`不及格人数`

from course\_info c

join (

select

course\_id,

sum(if(score >= 60,1,0)) as `及格人数`,

sum(if(score < 60,1,0)) as `不及格人数`

from score\_info

group by course\_id

) t1 on c.course\_id = t1.course\_id;

结果

c.course\_id c.course\_name t1.及格人数 t1.不及格人数

01 语文 12 7

02 数学 8 11

03 英语 13 6

04 体育 6 6

05 音乐 4 1

Time taken: 10.746 seconds, Fetched: 5 row(s)

### 5.1.6 查询课程编号为03且课程成绩在80分以上的学生的学号和姓名及课程信息

hive>

select

s.stu\_id,

s.stu\_name,

t1.score,

t1.course\_id,

c.course\_name

from student\_info s

join (

select

stu\_id,

score,

course\_id

from score\_info

where score > 80 and course\_id = '03'

) t1

on s.stu\_id = t1.stu\_id

join course\_info c on c.course\_id = t1.course\_id;

结果

s.stu\_id s.stu\_name t1.score t1.course\_id c.course\_name

002 胡歌 87 03 英语

004 刘德华 89 03 英语

005 唐国强 99 03 英语

013 蒋欣 93 03 英语

015 刘亦菲 84 03 英语

019 邓紫棋 93 03 英语

020 宋丹丹 81 03 英语

Time taken: 9.064 seconds, Fetched: 7 row(s)

## 5.2 多表连接

### 5.2.1 课程编号为"01"且课程分数小于60，按分数降序排列的学生信息

hive>

select

s.stu\_id,

s.stu\_name,

s.birthday,

s.sex,

t1.score

from student\_info s

join (

select

stu\_id,

course\_id,

score

from score\_info

where score < 60 and course\_id = '01'

) t1

on s.stu\_id=t1.stu\_id

order by t1.score desc;

结果

s.stu\_id s.stu\_name s.birthday s.sex t1.score

017 范冰冰 1992-07-04 女 58

008 吴京 1994-02-06 男 56

007 陈坤 1999-04-09 男 48

013 蒋欣 1997-11-08 女 47

019 邓紫棋 1994-08-31 女 46

012 杨紫 1996-12-21 女 44

018 李冰冰 1993-09-24 女 38

Time taken: 8.936 seconds, Fetched: 7 row(s)

### 5.2.2 查询所有课程成绩在70分以上的学生的姓名、课程名称和分数，按分数升序排列

hive>

select

s.stu\_id,

s.stu\_name,

c.course\_name,

s2.score

from student\_info s

join (

select

stu\_id,

sum(if(score >= 70,0,1)) flage

from score\_info

group by stu\_id

having flage =0

) t1

on s.stu\_id = t1.stu\_id

left join score\_info s2 on s.stu\_id = s2.stu\_id

left join course\_info c on s2.course\_id = c.course\_id;

结果

s.stu\_id s.stu\_name c.course\_name s2.course

002 胡歌 语文 74

002 胡歌 数学 84

002 胡歌 英语 87

002 胡歌 体育 100

016 周冬雨 语文 71

016 周冬雨 数学 89

016 周冬雨 英语 71

016 周冬雨 体育 94

Time taken: 27.166 seconds, Fetched: 8 row(s)

### 5.2.3 查询该学生不同课程的成绩相同的学生编号、课程编号、学生成绩

hive>

select

sc1.stu\_id,

sc1.course\_id,

sc1.score

from score\_info sc1

join score\_info sc2 on sc1.stu\_id = sc2.stu\_id

and sc1.course\_id <> sc2.course\_id

and sc1.score = sc2.score;

结果

sc1.stu\_id sc1.course\_id sc1.score

016 03 71

017 04 34

016 01 71

005 05 85

007 05 63

009 05 79

017 02 34

005 04 85

007 04 63

009 04 79

Time taken: 8.881 seconds, Fetched: 10 row(s)

### 5.2.4 查询课程编号为“01”的课程比“02”的课程成绩高的所有学生的学号

知识点：多表连接 + 条件

hive>

select

s1.stu\_id

from

(

select

sc1.stu\_id,

sc1.course\_id,

sc1.score

from score\_info sc1

where sc1.course\_id ='01'

) s1

join

(

select

sc2.stu\_id,

sc2.course\_id,

score

from score\_info sc2

where sc2.course\_id ="02"

)s2

on s1.stu\_id=s2.stu\_id

where s1.score > s2.score;

结果

stu\_id

001

005

008

010

011

013

014

015

017

019

020

### 5.2.5 查询学过编号为“01”的课程并且也学过编号为“02”的课程的学生的学号、姓名

hive>

select

t1.stu\_id as `学号`,

s.stu\_name as `姓名`

from

(

select

stu\_id

from score\_info sc1

where sc1.course\_id='01'

and stu\_id in (

select

stu\_id

from score\_info sc2

where sc2.course\_id='02'

)

)t1

join student\_info s

on t1.stu\_id = s.stu\_id;

结果

学号 姓名

001 彭于晏

002 胡歌

004 刘德华

005 唐国强

006 陈道明

007 陈坤

008 吴京

009 郭德纲

010 于谦

011 潘长江

012 杨紫

013 蒋欣

014 赵丽颖

015 刘亦菲

016 周冬雨

017 范冰冰

018 李冰冰

019 邓紫棋

020 宋丹丹

Time taken: 10.161 seconds, Fetched: 19 row(s)

### 5.2.6 [课堂讲解]查询学过“李体音”老师所教的所有课的同学的学号、姓名

hive>

select

t1.stu\_id,

si.stu\_name

from

(

select

stu\_id

from score\_info si

where course\_id in

(

select

course\_id

from course\_info c

join teacher\_info t

on c.tea\_id = t.tea\_id

where tea\_name='李体音' --李体音教的所有课程

)

group by stu\_id

having count(\*)=2 --学习所有课程的学生

)t1

left join student\_info si

on t1.stu\_id=si.stu\_id;

结果

s.stu\_id s.stu\_name

005 唐国强

007 陈坤

009 郭德纲

Time taken: 27.16 seconds, Fetched: 3 row(s)

### 5.2.7 [课堂讲解]查询学过“李体音”老师所讲授的任意一门课程的学生的学号、姓名

hive>

select

t1.stu\_id,

si.stu\_name

from

(

select

stu\_id

from score\_info si

where course\_id in

(

select

course\_id

from course\_info c

join teacher\_info t

on c.tea\_id = t.tea\_id

where tea\_name='李体音'

)

group by stu\_id

)t1

left join student\_info si

on t1.stu\_id=si.stu\_id;

结果

s.stu\_id s.stu\_name

001 彭于晏

002 胡歌

004 刘德华

005 唐国强

007 陈坤

009 郭德纲

010 于谦

013 蒋欣

014 赵丽颖

015 刘亦菲

016 周冬雨

017 范冰冰

018 李冰冰

020 宋丹丹

Time taken: 9.391 seconds, Fetched: 14 row(s)

### 5.2.8 [课堂讲解]查询没学过"李体音"老师讲授的任一门课程的学生姓名

hive>

select

stu\_id,

stu\_name

from student\_info

where stu\_id not in

(

select

stu\_id

from score\_info si

where course\_id in

(

select

course\_id

from course\_info c

join teacher\_info t

on c.tea\_id = t.tea\_id

where tea\_name='李体音'

)

group by stu\_id

);

结果

stu\_id stu\_name

003 周杰伦

006 陈道明

008 吴京

011 潘长江

012 杨紫

019 邓紫棋

Time taken: 36.559 seconds, Fetched: 6 row(s)

### 5.2.9 [课堂讲解]查询至少有一门课与学号为“001”的学生所学课程相同的学生的学号和姓名

hive>

select

si.stu\_id,

si.stu\_name

from score\_info sc

join student\_info si

on sc.stu\_id = si.stu\_id

where sc.course\_id in

(

select

course\_id

from score\_info

where stu\_id='001' --001的课程

) and sc.stu\_id <> '001' --排除001学生

group by si.stu\_id,si.stu\_name;

结果

s1.stu\_id s2.stu\_name

002 胡歌

004 刘德华

005 唐国强

006 陈道明

007 陈坤

008 吴京

009 郭德纲

010 于谦

011 潘长江

012 杨紫

013 蒋欣

014 赵丽颖

015 刘亦菲

016 周冬雨

017 范冰冰

018 李冰冰

019 邓紫棋

020 宋丹丹

Time taken: 8.97 seconds, Fetched: 18 row(s)

### 5.2.10 按平均成绩从高到低显示所有学生的所有课程的成绩以及平均成绩

hive>

select

si.stu\_name,

ci.course\_name,

sc.score,

t1.avg\_score

from score\_info sc

join student\_info si

on sc.stu\_id=si.stu\_id

join course\_info ci

on sc.course\_id=ci.course\_id

join

(

select

stu\_id,

avg(score) avg\_score

from score\_info

group by stu\_id

)t1

on sc.stu\_id=t1.stu\_id

order by t1.avg\_score desc;

结果

t2.stu\_name t2.course\_name t2.score t1.avg\_score

胡歌 体育 100 86.25

胡歌 数学 84 86.25

胡歌 英语 87 86.25

胡歌 语文 74 86.25

刘德华 体育 59 81.5

刘德华 语文 85 81.5

刘德华 英语 89 81.5

刘德华 数学 93 81.5

周冬雨 英语 71 81.25

周冬雨 数学 89 81.25

周冬雨 体育 94 81.25

周冬雨 语文 71 81.25

唐国强 数学 44 75.4

唐国强 音乐 85 75.4

唐国强 语文 64 75.4

唐国强 体育 85 75.4

唐国强 英语 99 75.4

郭德纲 音乐 79 74.2

郭德纲 体育 79 74.2

郭德纲 英语 60 74.2

郭德纲 语文 75 74.2

郭德纲 数学 78 74.2

陈道明 语文 71 73.33333333333333

陈道明 数学 90 73.33333333333333

陈道明 英语 59 73.33333333333333

……

李冰冰 音乐 87 58.0

李冰冰 语文 38 58.0

李冰冰 英语 49 58.0

李冰冰 数学 58 58.0

赵丽颖 数学 39 48.0

赵丽颖 语文 81 48.0

赵丽颖 体育 40 48.0

赵丽颖 英语 32 48.0

范冰冰 英语 55 45.25

范冰冰 体育 34 45.25

范冰冰 数学 34 45.25

范冰冰 语文 58 45.25

吴京 语文 56 43.0

吴京 数学 34 43.0

吴京 英语 39 43.0

Time taken: 20.137 seconds, Fetched: 74 row(s)