第 5 章 Hive数据查询语言

学习目标：

- 了解 SELECT 句式的语法，能够描述 SELECT 句式中不同子句的作用并查询数据表中的数据。
- 掌握 Hive 运算符的使用，能够在 SELECT 句式中灵活使用关系运算符、算术运算符、逻辑运算符和复杂运算符查询数据表中的数据。
- 熟悉公用表表达式的使用，能够使用公用表表达式基于临时结果集进行查询。
- 掌握分组操作，能够对数据表进行分组查询。
- 掌握排序操作，能够对数据表的查询结果进行排序。
- 熟悉 UNION 语句使用，能够将多个 SELECT 句式的查询结果集进行合并的操作。
- 熟悉 JOIN 语句的使用，能够将不同数据表的数据进行合并的操作。
- 熟悉抽样查询的使用，能够灵活运用 SELECT 句式实现随机抽样查询、分桶抽样查询和数据块抽样查询的操作。

数据查询语言(Data Query Language，DQL)是用于从数据库中查询数据的计算机语言。在 Hive 中可以通过 HiveQL 语言查询数据，查询数据的结果会存储在结果集中。本章详细讲解 Hive 数据查询语言的相关操作。

5.1 SELECT 句式分析

Hive 使用 SELECT 句式实现查询，在第 4 章对 SELECT 句式的基本查询语法进行了讲解，不过在实际使用中，仅对基础查询语法进行学习显然是不够的。本节详细讲解 SELECT 句式的完整格式，有关 SELECT 句式的完整语法格式如下。

```
[WITH CommonTableExpression (, CommonTableExpression) * ]
SELECT [ALL|DISTINCT] select_expr, select_expr, ...
  FROM table_reference
  [WHERE where_condition]
  [GROUP BY col_list]
  [ORDER BY col_list]
  [CLUSTER BY col_list
    | [DISTRIBUTE BY col_list] [SORT BY col_list]
  ]
[LIMIT number]
```

上述语法的具体讲解如下。

- WITH CommonTableExpression：可选，表示公用表达式。
- ALL | DISTINCT：可选，默认是 ALL，即显示全部查询结果；若使用 DISTINCT，则显示去重后的查询结果，去重是指去除重复数据。
- select_expr：表示查询语句的表达式。
- table_reference：表示实际的查询对象，可以是数据表、视图或子查询。
- WHERE where_condition：可选，指定查询条件。
- GROUP BY col_list：可选，用于对查询结果根据指定列 col_list 进行分组处理。
- ORDER BY col_list：可选，用于对查询结果根据指定列 col_list 进行排序处理。
- CLUSTER BY col_list | [DISTRIBUTE BY col_list] [SORT BY col_list]：可选，用于排序处理，CLUSTER BY 的功能等于 DISTRIBUTE BY+ SORT BY。
- LIMIT number：可选，用于限制查询结果返回的行数。

接下来，在虚拟机 Node_03 中使用 Hive 客户端工具 Beeline，远程连接虚拟机 Node_02 的 HiveServer2 服务操作 Hive，讲解 SELECT 句式的实际使用，具体操作步骤如下。

（1）在虚拟机 Node_02 的目录/export/data/hive_data 下执行“vi employess.txt”命令，创建员工信息数据文件 employess.txt，在数据文件 employess.txt 中添加如下内容。

```
Lilith Hardy,30,6000,50,Finance Department
Byron Green,36,5000,25,Personnel Department
Yvette Ward,21,4500,15.5,
Arlen Esther,28,8000,20,Finance Department
Rupert Gold,39,10000,66,R&D Department
Deborah Madge,41,6500,0,R&D Department
Tim Springhall,22,6000,36.5,R&D Department
Olga Belloc,36,5600,10,Sales Department
Bruno Wallis,43,6700,0,Personnel Department
Flora Dan,27,4000,35,Sales Department
```

上述内容中，每一行数据从左到右依次表示员工姓名、年龄、薪资、迟到扣款和所属部门。

（2）将虚拟机 Node_02 目录/export/data/hive_data 下的数据文件 employess.txt 上传到 HDFS 的/hive_data/employess 目录，上传文件前需要在 HDFS 创建目录/hive_data/employess，有关创建目录和上传数据文件的命令如下。

```
#创建目录
$ hdfs dfs -mkdir -p /hive_data/employess
#上传数据文件
$ hdfs dfs -put /export/data/hive_data/employess.txt /hive_data/employess
```

（3）根据数据文件 employess.txt 的数据格式，在数据库 hive_database 中创建员工信息表 employess_table，该表中包含列 staff_name（员工姓名）、staff_age（员工年龄）、staff_salary（员工薪资）、late_deduction（迟到扣款）和 staff_dept（员工所属部门），在虚拟机 Node_03 的 Hive 客户端工具 Beeline 中执行如下命令创建员工信息表 employess_table。

```
CREATE EXTERNAL TABLE IF NOT EXISTS
hive_database.employess_table(
staff_name STRING,
staff_age INT,
staff_salary FLOAT,
late_deduction FLOAT,
staff_dept STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS textfile
LOCATION '/hive_data/employess';
```

从上述命令可以看出，创建的员工信息表 employess_table 为外部表，并且通过 LOCATION 子句指定员工信息表 employess_table 在 HDFS 的数据存储路径/hive_data/employess，由于在创建员工信息表之前将数据文件 employess.txt 上传到 HDFS 的/hive_data/employess 目录，所以在创建员工信息表 employess_table 同时会加载数据文件 employess.txt。

(4) 查询员工信息表 employess_table 包含几种部门，具体命令如下。

```
SELECT DISTINCT staff_dept from hive_database.employess_table;
```

上述命令中，通过 DISTINCT 子句对列 staff_dept(员工所属部门)的数据进行去重，上述命令的执行效果如图 5-1 所示。

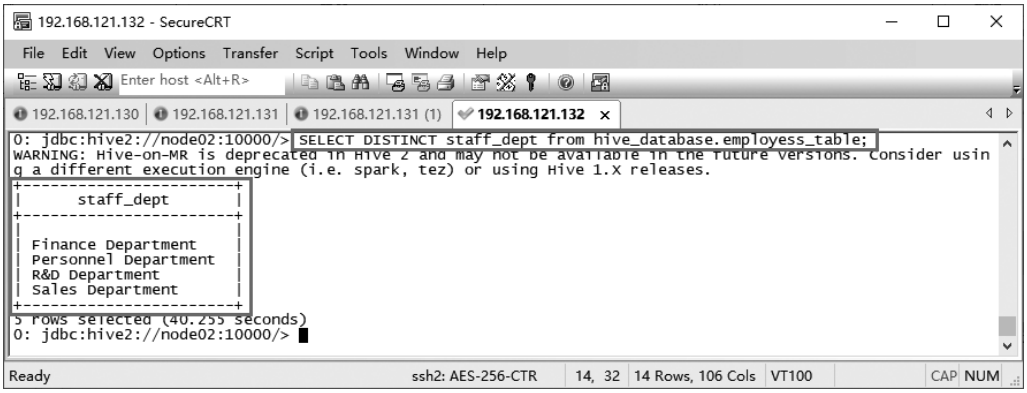

图 5-1　查询员工信息表 employess_table 包含几种部门

从图 5-1 可以看出，员工信息表 employess_table 中包含 5 种部门，分别是 Finance Department、Personnel Department、R&D Department、Sales Department 和一个没有名称的部门，这是因为数据文件中存在空值导致。

多学一招：HAVING 子句

HAVING 子句的用法与 WHERE 子句的用法一致，都是在查询语句中指定查询条件，

不同的是 HAVING 子句中可以使用聚合函数，并且 HAVING 子句必须配合 GROUP BY 子句一同使用，而 WHERE 子句中不能使用聚合函数，例如查询每个部门平均薪资大于 5000 的部门，命令如下。

```
SELECT staff_dept FROM employess_table GROUP BY staff_dept HAVING AVG(staff_salary) > 5000;
```

5.2 Hive 运算符

Hive 内置的运算符主要分为 4 类，分别是关系运算符、算术运算符、逻辑运算符和复杂运算符，本节针对这 4 类运算符进行详细讲解。

5.2.1 关系运算符

关系运算符通常在 SELECT 句式的 WHERE 子句中使用，用来比较两个操作数，下面通过表 5-1 来介绍 Hive 内置的常用关系运算符。

表 5-1 Hive 内置的常用关系运算符

关系运算符	支持数据类型	示　例	示 例 描 述
=	所有基本数据类型	A=B	表示如果 A 与 B 相等，则为 true，否则为 false
!=	所有基本数据类型	A! =B	表示如果 A 与 B 不相等，则为 true，否则为 false
<	所有基本数据类型	A<B	表示如果 A 小于 B，则为 true，否则为 false
<=	所有基本数据类型	A<=B	表示如果 A 小于或等于 B，则为 true，否则为 false
>	所有基本数据类型	A>B	表示如果 A 大于 B，则为 true，否则为 false
>=	所有基本数据类型	A>=B	表示如果 A 大于或等于 B，则为 true，否则为 false
IS NULL	所有基本数据类型	A IS NULL	表示如果 A 为 NULL，则为 true，否则为 false
IS NOT NUL	所有基本数据类型	A IS NOT NULL	表示如果 A 不为 NULL，则为 true，否则为 false
[NOT] LIKE	字符串类型	A [NOT] LIKE B	表示通过 SQL 正则匹配 A 与 B 是否相等，若相等，则为 true，反之为 false；使用关键字 NOT 匹配 A 与 B 不相等
RLIKE	字符串类型	ARLIKE B	表示通过 Java 正则匹配 A 与 B 相等的查询结果，若相等，则为 true，反之为 false

在表 5-1 中，若关系运算符比较操作数的结果为 true，则返回查询结果，否则返回空。

接下来，在虚拟机 Node_03 中使用 Hive 客户端工具 Beeline，远程连接虚拟机 Node_02 的 HiveServer2 服务操作 Hive，讲解关系运算符=、[NOT] LIKE 和 RLIKE 的实际使用，具体操作步骤如下。

(1) 查询员工信息表 employess_table 中员工年龄为 36 岁的员工信息，具体命令如下。

```
SELECT * FROM hive_database.employess_table WHERE staff_age=36;
```

上述命令使用关系运算符“=”比较列 staff_age 与 36 相等的值，返回比较结果为 true 的查询结果。上述命令在 Hive 客户端工具 Beeline 中的执行效果如图 5-2 所示。

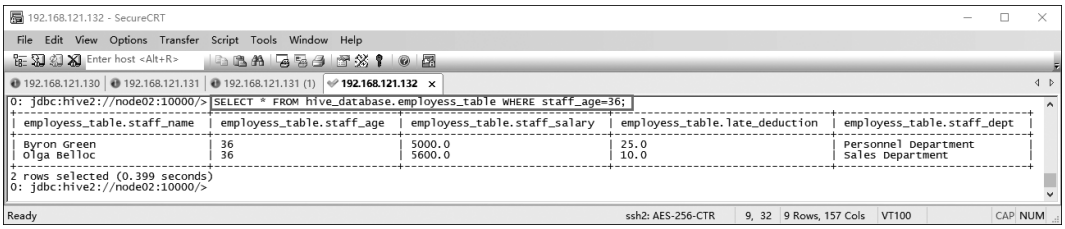

图 5-2　员工年龄为 36 岁的员工信息

从图 5-2 可以看出，员工信息表 employess_table 中有两个员工的年龄为 36 岁，员工姓名分别是 Byron Green 和 Olga Belloc。

（2）查询员工信息表 employess_table 中部门 Personnel Department 的员工信息，具体命令如下。

```
SELECT * FROM hive_database.employess_table WHERE staff_dept LIKE "Per%";
```

上述命令使用关系运算符 LIKE 比较列 staff_dept 与 Per%相等的字符串，返回比较结果为 true 的查询结果，其中“%”表示通配符，用来匹配任意字符，因此 Per%表示字符串开头为 Per 的任意字符串。上述命令在 Hive 客户端工具 Beeline 中的执行效果如图 5-3 所示。

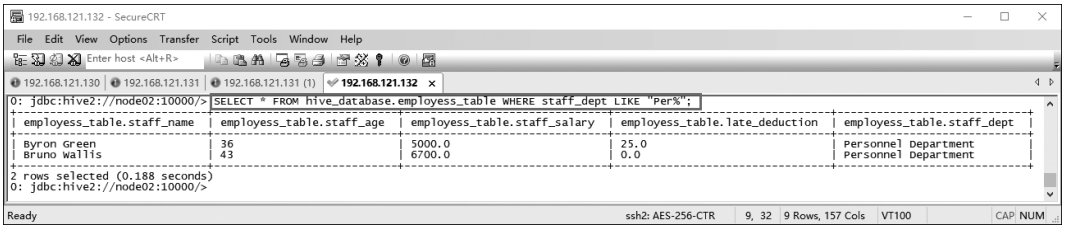

图 5-3　部门 Personnel Department 的员工信息

从图 5-3 可以看出，员工信息表 employess_table 中部门 Personnel Department 包含两个员工，员工姓名分别是 Byron Green 和 Bruno Wallis。

（3）查询员工信息表 employess_table 中员工姓名首字母为 A 或 D 的员工信息，具体命令如下。

```
SELECT * FROM hive_database.employess_table
WHERE staff_name RLIKE "^A|^D.*";
```

上述命令中，通过关系运算符 RLIKE 比较列 staff_name 与 Java 正则表达式“^A|^D.*”相等的字符串，这里使用的正则表达式表示匹配字符串首字母以 A 或 D 开头的字符串。上述命令在 Hive 客户端工具 Beeline 中的执行效果如图 5-4 所示。

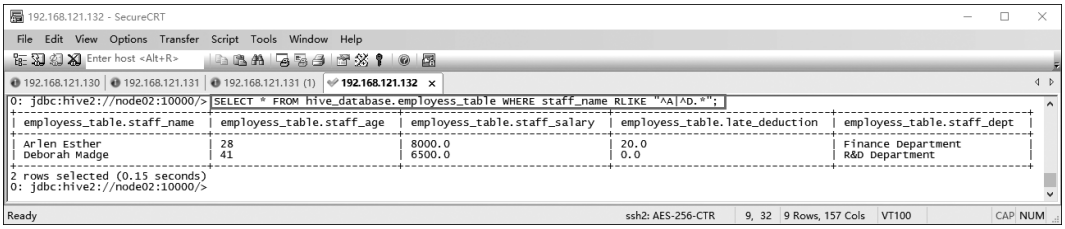

图 5-4　员工姓名首字母为 A 或 D 的员工信息

从图 5-4 可以看出，员工信息表 employess_table 中有两名员工姓名首字母为 A 或 D。

多学一招：Hive 中的 NULL

Hive 的数据存储在 HDFS 时，默认会将空值转换为字符串，因此使用 IS NULL 或者 IS NOT NULL 关系运算符时无法匹配到表中存在的空值，需要通过数据表的属性 serialization.null.format 格式化表中的空值，例如，执行“alter table hive_database.employess_table set serdeproperties ('serialization.null.format' = '');”命令格式化表 employess_table 中的空字符串为 NULL，或者执行“alter table hive_database.employess_table set serdeproperties ('serialization.null.format' = 'NULL');”命令格式化表 employess_table 中的字符串"NULL"为 NULL。

也可以在创建表时使用“NULL DEFINED AS ""”命令指定表中空值的序列化方式，具体实例代码如下。

```
CREATE  TABLE IF NOT EXISTS table1(
col1 INT,
col2 STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '_'
MAP KEYS TERMINATED BY ':'
LINES TERMINATED BY '\n'
NULL DEFINED AS ''
STORED AS textfile
TBLPROPERTIES("comment"="This is a managed table");
```

5.2.2　算术运算符

算术运算符是用来计算两个数值的操作，下面通过表 5-2 来介绍 Hive 内置的常用算术运算符。

表 5-2　Hive 内置的常用算术运算符

算术运算符	支持数据类型	示　例	示 例 描 述
+	所有数字数据类型	A + B	A 加 B 的计算结果
−	所有数字数据类型	A − B	A 减 B 的计算结果

续表

算术运算符	支持数据类型	示　例	示 例 描 述
*	所有数字数据类型	A * B	A 乘以 B 的计算结果
/	所有数字数据类型	A / B	A 除以 B 的计算结果
%	所有数字数据类型	A % B	A 除以 B 产生余数的计算结果
&	所有数字数据类型	A & B	A 和 B 按位与的计算结果
|	所有数字数据类型	A | B	A 和 B 按位或的计算结果
^	所有数字数据类型	A ^ B	A 和 B 按位异或的计算结果
~	所有数字数据类型	~A	A 按位非的计算结果

接下来，在虚拟机 Node_03 中使用 Hive 客户端工具 Beeline，远程连接虚拟机 Node_02 的 HiveServer2 服务操作 Hive，讲解算术运算符“-”和“/”的实际使用，具体操作步骤如下。

（1）计算员工信息表 employess_table 中所有员工的实际工资，具体命令如下。

```
SELECT staff_name,
staff_salary-late_deduction actual_salary
FROM hive_database.employess_table;
```

上述命令中，通过算术运算符“-”计算员工薪资（staff_salary）减迟到扣款（late_deduction）得出每个员工的实际工资，指定实际工资的列名为 actual_salary。上述命令在 Hive 客户端工具 Beeline 中的执行效果如图 5-5 所示。

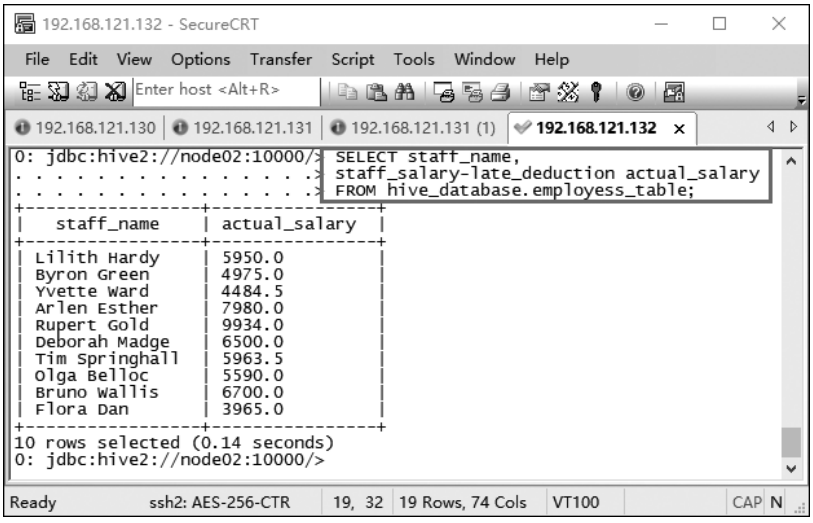

图 5-5　计算员工实际工资

（2）计算员工信息表 employess_table 中每位员工每天的薪资，以单月工作日为 20 天计算，具体命令如下。

```
SELECT staff_name,
staff_salary/20 everyday_salary
FROM hive_database.employess_table;
```

上述命令中，通过算术运算符“/”计算员工薪资(staff_salary)除以 20 得出每位员工每天的薪资，指定每天薪资的列名为 everyday_salary。上述命令在 Hive 客户端工具 Beeline 中的执行效果如图 5-6 所示。

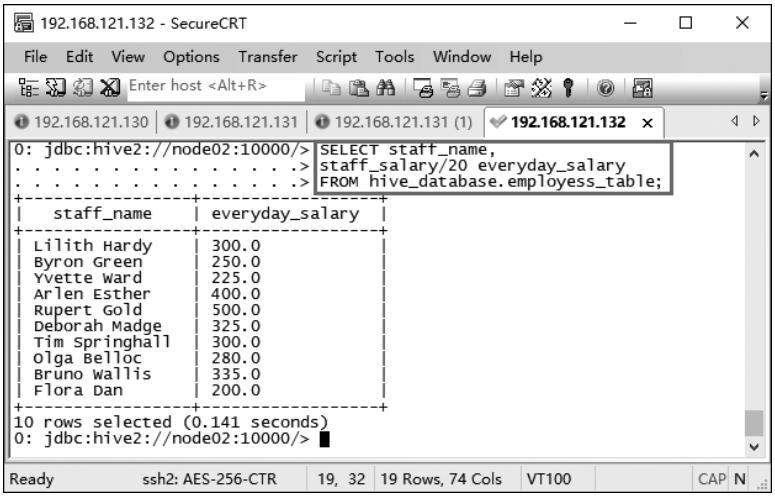

图 5-6 计算每位员工每天的薪资

5.2.3 逻辑运算符

逻辑运算符(又称为逻辑联结词)可以将两个或多个关系表达式合并为一个表达式或者反转表达式的逻辑。下面通过表 5-3 来介绍 Hive 内置的常用逻辑运算符。

表 5-3 Hive 内置的常用逻辑运算符

逻辑运算符	支持数据类型	示 例	示 例 描 述
AND	Boolean 数据类型	A AND B	逻辑与，表达式 A 和表达式 B 必须都为 true，合并后的整体表达式结果才为 true
&&	Boolean 数据类型	A&& B	逻辑与，含义与逻辑运算符 AND 相同
OR	Boolean 数据类型	A OR B	逻辑或，表达式 A 或表达式 B 至少一个为 true，才能使合并后的整体表达式结果为 true
||	Boolean 数据类型	A || B	逻辑或，含义与逻辑运算符 OR 相同
!	Boolean 数据类型	! A	逻辑非，反转表达式 A 的“真相”，若表达式 A 为 true 则反转后的结果为 false
NOT	Boolean 数据类型	NOT A	逻辑非，含义与逻辑运算符! 相同

接下来，在虚拟机 Node_03 中使用 Hive 客户端工具 Beeline，远程连接虚拟机 Node_02 的 HiveServer2 服务操作 Hive，查询员工信息表 employess_table 中薪资大于或等于 5000，

并且薪资小于或等于 8000 的员工信息，具体命令如下。

```
SELECT * FROM hive_database.employess_table WHERE staff_salary >= 5000 AND staff_salary
<= 8000;
```

上述命令通过逻辑运算符 AND 将表达式“staff_salary >= 5000”和表达式“staff_salary <= 8000”合并为一个整体表达式。上述命令在 Hive 客户端工具 Beeline 中的执行效果如图 5-7 所示。

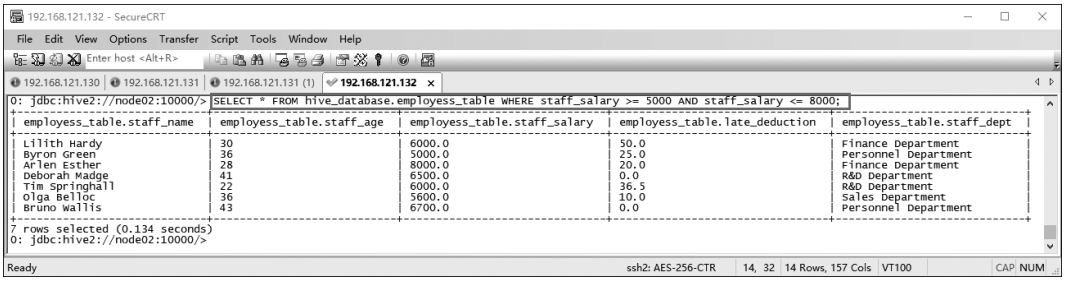

图 5-7　薪资大于或等于 5000，并且薪资小于或等于 8000 的员工信息

从图 5-7 可以看出，员工信息表 employess_table 中有 7 名员工薪资大于或等于 5000，并且薪资小于或等于 8000。

5.2.4　复杂运算符

复杂运算符用于操作 Hive 中集合数据类型的列，集合数据类型包括 ARRAY、MAP 或 STRUCT，下面通过表 5-4 来介绍 Hive 内置的复杂运算符。

表 5-4　Hive 内置的复杂运算符

复杂运算符	支持数据类型	描　　述
A[n]	ARRAY	返回 A(ARRAY)的第 n 个元素的值
M[key]	MAP	返回 M(MAP)中指定 key 的 value
S.x	STRUCT	返回 S(STRUCT)中 x 字段的值

接下来，在虚拟机 Node_03 中使用 Hive 客户端工具 Beeline，远程连接虚拟机 Node_02 的 HiveServer2 服务操作 Hive，讲解复杂运算符的实际使用，具体操作步骤如下。

(1) 在虚拟机 Node_02 的目录/export/data/hive_data 下执行“vi student_exam.txt”命令，创建学生考试成绩文件 student_exam.txt，在数据文件 student_exam.txt 中添加如下内容。

```
Mandy,Peking University-Wuhan University-Nankai University,Chemistry:90-Physics:98-
Biology:83,126-135-140
Jerome,Tsinghua University-Fudan University-Nanjing University,History:89-Politics:92-
Geography:87,130-116-128
Delia,Nanjing University-Wuhan University-Nankai University,Chemistry:87-Physics:95-
Biology:73,102-123-112
```

```
Ben,Tianjin Universit-Peking University-Fudan University,Chemistry:92-Physics:88-
Biology:79,98-142-106
Carter,Tsinghua University-Fudan University-Tianjin Universit,History:90-Politics:91-
Geography:80,109-111-140
Vivian,Fudan University-Nanjing University-Nankai University,Chemistry:83-Physics:86-
Biology:90,120-140-132
```

上述内容中，每一行数据从左到右依次表示学生姓名、意向大学、文综（政治、历史、地理）/理综（生物、化学、物理）成绩和综合（语、数、外）成绩。

（2）将虚拟机 Node_02 目录/export/data/hive_data 下的数据文件 student_exam.txt 上传到 HDFS 的/hive_data/student_exam 目录，上传文件前需要在 HDFS 上创建目录/hive_data/student_exam，有关创建目录和上传数据文件的命令如下。

```
$ hdfs dfs -mkdir -p /hive_data/student_exam
$ hdfs dfs -put /export/data/hive_data/student_exam.txt /hive_data/student_exam
```

（3）根据数据文件 student_exam.txt 的数据格式，在数据库 hive_database 中创建学生考试成绩表 student_exam_table，该表中包含列 student_name（学生姓名）、intent_university（意向大学）、humanities_and_sciences（文/理综成绩）和 comprehensive（综合成绩），在虚拟机 Node_03 的客户端工具 Beeline 中执行如下命令。

```
CREATE EXTERNAL TABLE IF NOT EXISTS
hive_database.student_exam_table(
student_name STRING,
intent_university ARRAY<STRING>,
humanities_or_sciences MAP<STRING, FLOAT>,
comprehensive STRUCT<chinese:FLOAT,maths:FLOAT,english:FLOAT>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '-'
MAP KEYS TERMINATED BY ':'
LINES TERMINATED BY '\n'
STORED AS textfile
LOCATION '/hive_data/student_exam';
```

上述命令中，创建的学生考试成绩表 student_exam_table 为外部表，并且通过 LOCATION 子句指定学生考试成绩表 student_exam_table 在 HDFS 的数据存储路径/hive_data/student_exam，因此在创建学生考试成绩表 student_exam_table 同时会加载数据文件 student_exam.txt。学生考试成绩表 student_exam_table 包含 3 个集合数据类型的列，分别是 intent_university、humanities_or_sciences 和 comprehensive。

（4）查询学生考试成绩表 student_exam_table 中所有学生的第一意向大学，具体命令如下。

```
SELECT student_name,
intent_university[0] first
FROM hive_database.student_exam_table;
```