Hive sql 高级语法

Hive 的数据类型

JDT 京东科技

在一张表中一个列是同一个类型，不同的列是数据类型可以不同。hive支持以下基本类型及复合类型：

- **整型 Integers**
  - ✓ TINYINT—1 byte integer（1 个字节的整形型）
  - ✓ SMALLINT—2 byte integer（2 个字节的整形型）
  - ✓ INT—4 byte integer（4 个字节的整形型）
  - ✓ BIGINT—8 byte integer（8 个字节的整形型）
- **布尔型 Boolean type**
  - ✓ BOOLEAN—TRUE/FALSE（只有真/假两个种取值）
- **浮点数字 Floating point numbers**
  - ✓ FLOAT—单精度(5.21)
  - ✓ DOUBLE—双精度（5.21）
  - ✓ DECIMAL—高精度浮点数(DECIMAL(9,8)代表最多9位数字，后8位是小数)
- **字符型 String types**
  - ✓ STRING — 指定字符集中的字符序列
  - ✓ VARCHAR — 指定字符集中具有最大长度的字符序列
  - ✓ CHAR — 指定字符集中具有最大长度的字符序列
- **日期时间类型 Date and time types**
  - ✓ TIMESTAMP — 没有时区的日期和时间 ("LocalDateTime" 语意)
  - ✓ TIMESTAMP WITH LOCAL TIME ZONE — 精度到纳秒的时间点 ("Instant" 语意)
  - ✓ DATE — 一个日期
- **二进制类型 Binary types**
  - ✓ BINARY — 字节序列

- **结构体 Structs:**

  可以使用点（.）符号访问类型中的元素。例如，列 c 的类型是一个 STRUCT，形式如 {"col1": "a","col2": "b"}，其中查询字段 a 的表达式为c.col1 ；

- **图映射 Maps (key-value tuples):**

  用键-值对组成的元组表示，其中的元素用 ['element name'] 来访问。例如，列M的类型为图映射，形式如 { 'group' : 'gid' }，其中查询值gid的表示式为M['group'] 。

- **数组 Arrays (indexable lists):**

  它是一个可索引的列表，数组中的元素必须是同一类型。可以使用 [n] 表示访问元素，其中 n 是数组中的索引（从0开始）。例如，列A的类型为数组，形式如 ['a' , 'b' , 'c']，其中查询元素b的表示式为A[1]。

HQL 创建删除表语句

JDT 京东科技

- **创建简单表**

  CREATE TABLE person(name string,age int);

例如：创建一张名为students的学生信息表：

```
use xxx;  ---库名

CREATE TABLE xxx.students (
    id        int       comment  '学生号',
    name      string    comment  '学生姓名',
    class     int       comment  '学生班级',
    b_year    int       comment  '数据年份',
    gender    string    comment  '学生性别',
    math      int       comment  '数学成绩',
    chinese   int       comment  '语文成绩',
    english   int       comment  '英语成绩'
);

INSERT INTO xxx.students (id,name,class,b_year,gender,math,chinese,english)
VALUES
    (1,'张涛' ,1,1950,'男',66,77,88),
    (2,'王琳' ,2,2010,'女',88,99,77),
    (3,'赵丹丹',1,1996,'女',55,55,55),
    (4,'李成',2,2011,'男',54,87,99),
    (5,'赵天成',3,2000,'男',77,66,88),
    (6,'田迪',1,1988,'女',78,99,76),
    (7,'王卫栋',2,1966,'男',88,66,88),
    (8,'周平',1,1988,'男',77,99,76),
    (9,'武明',3,1977,'男',78,66,88)
;
```

在京东科技数据仓库，C端集市测试库库名为dmc_dev.

- **删除简单表**

  Drop table TABLE;

  例如：

```
use xxx;
drop table xxx.students;
```

id	name	class	b_year	gender	math	chinese	english
1	张涛	1	1950	男	66	77	88
2	王琳	2	2010	女	88	99	77
3	赵丹丹	1	1996	女	55	55	55
4	李成	2	2011	男	54	87	99
5	赵天成	3	2000	男	77	66	88
6	田迪	1	1988	女	78	99	76
7	王卫栋	2	1966	男	88	66	88
8	周平	1	1988	男	77	99	76
9	武明	3	1977	男	78	66	88

HQL常用时间函数

JDT 京东科技

- **日期提取函数**
  - ✓ **提取时间字段中的日期：to_date(string date)**

    hive> select to_date('2017-02-08 11:15:50') ;
    > 返回2017-02-08
  - ✓ **提取时间字段中的年月日等信息：year、month、day等**

    hive> select year('2017-02-08 11:15:50');
    > 返回2017

    hive> select month('2017-02-08 11:15:50');
    > 返回2

    hive> select day('2017-02-08 11:15:50');
    > 返回8

    hive> select hour('2017-02-08 11:15:50');
    > 返回11

    hive> select minute('2017-02-08 11:15:50');
    > 返回15

    hive> select second('2017-02-08 11:15:50');
    > 返回50

    hive> select weekofyear('2017-02-08 11:15:50');
    > 返回6 -- 返回一年中的第几周

- **日期运算函数**
  - ✓ **日期增加/减少**

    返回开始日期增加天数后的日期：date_add (string startdate, int days)
    返回开始日期减少天数后的日期：date_sub (string startdate, int days)

    hive> select date_add('2017-01-31',2);
    > 返回 2017-02-02

    hive> select date_sub('2017-02-02',2);
    > 返回 2017-01-31
  - ✓ **日期差计算**

    返回结束时间距离开始时间的天数：
    datediff(string enddate, string startdate)

    hive> select datediff('2017-02-02','2017-01-31');
    > 返回 2

    返回月份差：months_between(date1, date2)
    hive> select months_between('2021-09-30','2021-08-31');
    > 返回1.0

    注：如果date1和date2为月份的同一天或最后一天，则返回值为整数。
    否则该函数基于一个月31天计算两个日期的月份差值，返回浮点数

HQL常用日期函数

JDT 京东科技

- **日期-时间戳转换函数**

  时间戳：时间戳是指格林威治时间1970年01月01日00时00分00秒(北京时间1970年01月01日08时00分00秒)起至现在的总秒数，它可以和日期互相转换。
  - ✓ **日期转时间戳：unix_timestamp(string date[,string format])**

    其中格式为可选参数，默认为'yyyy-MM-dd HH:mm:ss'

    hive> select unix_timestamp('2018-06-29 00:00:00');
    > 返回1530201600

    hive> select unix_timestamp('2018/06/29 09', 'yyyy/MM/dd HH'); -- 自定义格式为'yyyy/MM/dd HH',
    > 返回1530234000
  - ✓ **时间戳转日期：from_unixtime(bigint unixtime[, string format])**

    将时间戳转换为日期，格式为可选参数，默认为'yyyy-MM-dd HH:mm:ss'

    hive> select from_unixtime(1447141681, 'yyyy-MM-dd');
    > 返回2015-11-10

    注：如果碰到13位时间戳（即毫秒时间戳），首先要将其除以1000转为秒级，然后从浮点型转为整型，再做时间戳转换

- **字符串截取**
  - ✓ **substring_index(string A, string delim, int n): 按照分隔符截取，返回第n个分隔符左边或右边的全部内容**

    A: 被截取字符串

    delim: 分隔符

    n: 指定分隔符的位置，当n是正数时，返回从左到右第n个分隔符左边的全部字符，当n为负数时，返回从右到左第-n个分隔符右边的全部字符

    hive> select substring_index('abcdcec','c',**2**); -- 返回第2个c（从左到右）左边的全部字符
    > 返回abcd
    hive> select substring_index('abcdcec','c',**-2**); -- 返回第2个c（从右到左）右边的全部字符
    > 返回ec

HQL常用字符串函数

JDT 京东科技

- **字符串大小写转换函数**
  - ✓ **lower(string A):** 将字符串A中的所有字母转换为小写，并返回

    hive> select lower('abCDefg');
    > 返回abcdefg
  - ✓ **upper(string A):** 将字符串A中的所有字母转换为大写，并返回

    hive> select upper('abCDefg');
    > 返回ABCDEFG
- **字符串长度计算**
  - ✓ **length(string A):** 返回字符串A的长度

    hive> select length('abcedfg');
    > 返回7
- **字符串去除首尾空格**
  - ✓ **trim(string A):** 去除字符串首尾的空格

    hive> select trim(' abc ');
    > 返回abc

HQL常用字符串函数

JDT 京东科技

- **JSON字符串读取函数**
  - ✓ **JSON概述**

    JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，它采用完全独立于编程语言的文本格式，同时也使用了"键值"、"数组"等编程语言中的一些概念，因此JSON经常作为数据存储和传输的方式。
  - ✓ **JSON基本结构**

    JSON字符串基于两种结构：对象结构和数组结构
    - **对象结构**：对象结构由单个或多个键值对组成，其中键的类型是string，值的类型可以为数值、字符串、对象、数组。一个对象结构以"{}"作为开始和结尾的标识。

      例如：[{"student_name":"Amy","student_ID":"001","hobby":["羽毛球","写作"]}
    - **数组结构**：数组结构与HQL中的数组类型相似，其元素都为同一类型。一个数组以"[]"作为开始和结尾的标识。

      例如：

      [{"student_name":"Amy","student_ID":"001","hobby":["羽毛球","写作"]},
      {"student_name":"Jack","student_ID":"002","hobby":["篮球","电影"]}]

      在这个数组中，数组的元素为对象，"hobby"的值又是一个数组

附录：建表语句

JDT 京东科技

在京东科技数据仓库，C端集市测试库库名为dmc_dev.

本文档所涉及到的表的建表语句如下：

1.hql_student_01

```
use dmc_dev;
drop table if exists dmc_dev.hql_student_01;
create table dmc_dev.hql_student_01 (student_info string);
insert into dmc_dev.hql_student_01 (student_info) values(
    '[{"student_name":"Amy","student_ID":"001","hobby":["羽毛球","写作
    "]},{"student_name":"Jack","student_ID":"002","hobby":["篮球","电影"]}]') ;
```

2.hql_student_02

```
use dmc_dev;
drop table if exists dmc_dev.hql_student_02;
create table dmc_dev.hql_student_02 (student_name string);
insert into dmc_dev.hql_student_02 (student_name) values('White'),('amy'),('Amy'),('Jack') ;
```

hql_student_02

student_name
White
amy
Amy
Jack

3.hql_student_03

```
use dmc_dev;
drop table if exists dmc_dev.hql_student_03;
create table dmc_dev.hql_student_03(student_id string, score int);
insert into dmc_dev.hql_student_03(student_id,score) values('001',98),('002',84),('003',70) ;
```

hql_student_03

student_id	score
001	98
002	84
003	70

HQL常用字符串函数

JDT 京东科技

- **字符串替换函数**
  - ✓ **replace(string A, string B, string C)：使用字符串C替换A中的B部分**

    string A：被搜索的字符串

    string B：A中被替换的字符串

    string C：新的字符串

    ```
    hive>select replace('ababab', 'ab', 'Z');          -- 使用Z替换ababab中的ab
        >返回'ZZZ'
    ```

  - ✓ **regexp_replace(string A, string PATTERN, string B)：使用字符串B对A中符合正则表达式PATTERN的部分替换**

    string A：被搜索的字符串

    PATTERN：JAVA正则表达式[1]

    string B：新的字符串

    ```
    hive>select regexp_replace("foobar", "oolar", "") ;   -- "oolar"表示匹配"oo"或者"ar"
        >返回fb
    ```

    [1] 感兴趣的小伙伴可自行学习JAVA正则表达式语法

HQL常用字符串函数

JDT 京东科技

✓ **JSON字符串读取函数：get_json_object(string json_string, string path)**

JSON字符串是数据传输和存储时的格式，当提取字符串中信息时，我们需要将其解析为JSON对象再提取其中信息。hive中读取JSON中的信息，与maps和数组读取元素的方法类似：使用**$**表示JSON变量标识，通过 **.键** 或 **[索引]**读取对象或者数组对应的值。

例如：hql_student_01表的student_info字段中存在以下的一条记录[1]：

```
'[{"student_name":"Amy","student_ID":"001","hobby":["羽毛球","写作"]},
{"student_name":"Jack","student_ID":"002","hobby":["篮球","电影"]}]'
```

解析其中的信息：

```
hive>select get_json_object(student_info,'$.[0].student_name') from dmc_dev.hql_student_01;
    >返回Amy                -- 表示读取数组第一个元素中键student_name对应的值Amy
hive>select get_json_object(student_info,'$.[0]') from dmc_dev.hql_student_01;
    >返回{"student_name":"Amy","student_ID":"001","hobby":["羽毛球","写作"]}  -- 表示读取数组第一个元素
```

[1] 建表及插入数据语句见PPT末

HQL常用字符串函数

JDT 京东科技

- **字符串匹配函数**

在hive中我们可以使用like和rlike对字符串进则匹配，并将匹配的结果作为条件对数据进行过滤。

✓ **A [not] like B：使用表达式B对A简单匹配**

如果A与B匹配，返回True，否则返回False(not like 则对结果取反)

在like的匹配中，使用**%**和_两种简单匹配符号：%表示匹配任意数量字符，如'%x'表示匹配任意以x结尾的数据，'%x%'表示匹配包含x的任意数据；_表示匹配单一字符，如'_a'表示匹配以任意单个字母开头且以a结尾的数据。

例如：匹配hql_student_02[1]表中student_name以'A'开头的记录：

```
hive>select student_name
      from dmc_dev.hql_student_02
      where student_name like 'A%';
```

student_name
White
amy
Amy
Jack

like 'A%'

student_name
Amy

✓ **A rlike B：使用表达式B对A正则匹配**

与like匹配相似，不同的是B为JAVA中正则表达式，匹配规则比like更高级复杂，有兴趣的同学可以查阅JAVA正则表达式的语法

[1] 建表及插入数据语句见PPT末

HQL常用条件函数

JDT 京东科技

- **条件判断函数**
  - ✓ **if(testCondition, a, b)**：当条件满足时，返回a值，否则返回b值，适合单一条件判断场景。

    hive>select if(1=2,100,200);

    >返回200
- **非空查找**
  - ✓ **coalesce(T v1,T v2,...)**：返回参数中的第一个非空值，如果都是空值，则返回null

    hive>select coalesce(null,'a','b') ;

    >返回a
  - ✓ **nvl(T value, T default_value)**：返回两个参数中的第一个非空值，如果都是空值，则返回null

    hive>select nvl(null,'a');

    >返回a

    两函数区别：coalesce可对多个参数进行判断，nvl对两个参数判断

    通过上述两个函数可以对字段中的空值进行转换。如将字段A所有的空值转换为0，可使用nvl(A,0)或者coalesce(A,0)

HQL常用数学函数

JDT 京东科技

- **取整函数**
  - ✓ **round(double A[, int B])**：返回A的四舍五入值，四舍五入到小数点后B位的值，默认四舍五入到整数位。

    hive>select round(1.45566,3);

    >返回1.456

    hive>select round(1.45566);

    >返回1
  - ✓ **floor(double A)**：对A向下取整

    hive>select floor(1.7);

    >返回1
  - ✓ **ceil(double A)**：对A向上取整

    hive>select ceil(1.1);

    >返回2
  - ✓ **abs(double A)**：返回A的绝对值

    hive>select abs(-1.3);

    >返回1.3

HQL其它常用函数-类型转换

JDT 京东科技

- **类型转换函数**
  - ✓ **cast(expr as <type>)**：类型转换函数，显式地将字段值转换为其它类型

    hive>select cast('1' as int);　　>返回1　　-- string转int

    hive>select cast('1.23'as float);　>返回1.23　　-- string转float

    hive>select cast(1.23 as string); >返回"1.23"　　-- float转string

    注：并不是所有数据类型之间都是可以互相转换的，不能转换时将返回null。hive官网给出类型间的转换规则如下：

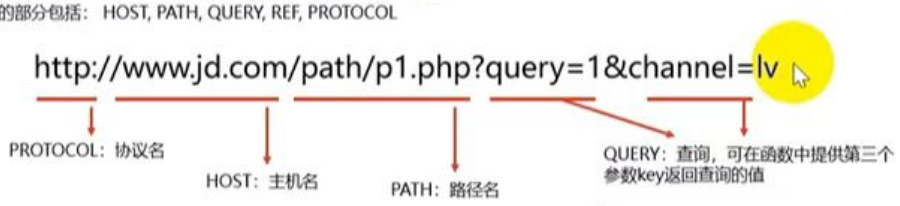
	void	boolean	tinyint	smallint	int	bigint	float	double	decimal	string	varchar	timestamp	date	binary
void to	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
boolean to	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
tinyint to	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
smallint to	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
int to	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
bigint to	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
float to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
double to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
decimal to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
string to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
varchar to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
timestamp to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
date to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE
binary to	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

HQL其它常用函数-url解析函数

JDT 京东科技

- **url解析函数**
  - ✓ **parse_url(url, partToExtract[, key]):解析网页url地址，返回url网址的特定部分**

    可解析的部分包括： HOST, PATH, QUERY, REF, PROTOCOL



hive>select parse_url('http://facebook.com/path/p1.php?query=1', 'PROTOCOL');　-- 返回协议
　>返回 http

hive>select parse_url('http://facebook.com/path/p1.php?query=1', 'HOST');　-- 返回主机名
　>返回facebook.com

hive>select parse_url('http://facebook.com/path/p1.php?query=1', 'PATH');　-- 返回路径名
　>返回 /path/p1.php

hive>select parse_url('http://facebook.com/path/p1.php?query=1', 'QUERY','query');　-- 返回query对应的查询值
　>返回1

HQL其它常用函数-窗口函数

JDT 京东科技

- **聚合类窗口函数**

聚合类窗口函数可以实现计算截至当前排序记录的聚合函数值，如：当前是排序后的第三条记录，则返回前三条记录的聚合函数值

以表hql_student_03[1]中数据为例

```
hive>select  student_id,score,
      sum(score) over(order by score desc) as `sum`,
      count(score) over(order by score desc) as `count`,
      avg(score) over(order by score desc) as `avg`,
      max(score) over(order by score desc) as `max`,
      min(score) over(order by score desc) as `min`
      from dmc_dev.hql_student_03;
```

[1] 建表及插入数据语句见PPT末

student_id	score	sum	count	avg	max	min
001	98	98	1	98	98	98
002	84	182	2	91	98	84
003	70	252	3	84	98	70

前两行函数值

前三行函数值

HQL其它常用函数-行列互转

JDT 京东科技

- **行列互转**

在日常工作中，我们可能会遇到需要行列互转的场景，如将A表和B表互相转换

- ✓ **A表转B表**

列较少时,我们可以使用union all,union all用于联合多个select语句的结果集,区别于union，其不会消除重复数据

```
hive>select
      student_id, 'math' as subject, math as score
      from  dmc_dev.hql_scoreA
      union all
      select
      student_id, 'english' as subject, english as score
      from dmc_dev.hql_scoreA;
```

studetn_id	math	english
001	100	90
002	90	96
003	96	93

A表

⇩

student_id	subject	score
001	math	100
002	math	90
003	math	96
001	english	90
002	english	96
003	english	93

B表

HQL其它常用函数-行列互转

JDT 京东科技

✓ **A表转B表**

列较多时，使用union all就不太方便了，可以使用**lateral view+ explode+str_to_map**函数。

- **str_to_map(text[, delimiter1, delimiter2])**：将字符串转换为科目-成绩的键值
- **explode**：将键值对拆分为多行，一个键值对对应一行
- **lateral view**：搭配**explode**使用，将拆分的键值对生成一个视图，每个键值对是一行数据，通过引用视图的自定义列名实现查询

```
hive>select
        student_id, tmp.subject ,tmp.score
        from dmc_dev.hql_scoreA
            lateral view
                explode(
                    str_to_map(concat('math=',cast(math as string), '#english=',cast(english as string)),'#','=')
                        ) tmp as subject,score;
```

student_id	math	english
001	100	90
002	90	96
003	96	93

str_to_map

values
{"math":"100","english":"90"}
{"math":"90","english":"96"}
{"math":"96","english":"93"}

explode

key	value
math	100
english	90
math	90
english	96
math	96
english	93

lateral view

student_id	subject	sco
001	math	100
001	english	90
002	math	90
002	english	96
003	math	96
003	english	93

使用场景——数据下载

JDT 京东科技

创建表 >>> 流程申请 >>> 查看结果

以5K集群为例，只能下载带_dev的库表。例如，我想要一张dmc库下面的表，是不能直接下载，需要 **create table xxx as select...**把表保存到带_dev库里。

银河+流程中心>运营服务平台>科技数据使用/提取/输出申请流程

审批通过后，会向您邮箱发送一个无扩展名的文件，**可以手动加.cxv或者.txt**，用excel或者文本工具打开。

使用场景——数据上传

JDT 京东科技

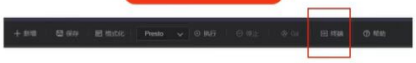
上传本地文件 → 创建表 → 数据回写

上传到/soft/data目录下，
文件类型csv/txt(UTF-8)，
单文件大小不超过500M

在SQL窗口，在带_dev库里，创建表结构相同的表。

```
Create table xxx
...
row format delimited
fields terminated by
',' stored as textfile;
```

建表语句最后一定要注意分隔符和文件类型



方法一、
终端：输入Hive，回车，
再输入命令：

```
LOAD DATA LOCAL INPATH
'/soft/data/test.txt' INTO TABLE
MYTEST2;
```

方法二、
打开Shell脚本，输入命令执行：

```
hive -e 'LOAD DATA LOCAL INPATH
"/soft/data/test.txt" INTO TABLE
MYTEST2';
```

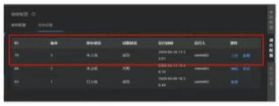
使用场景——发布作业

JDT 京东科技

打开银河模板、调试 → 填写调度信息、试跑 → 上线作业

注意：
1、使用Python2运行；
2、参数设置填上任意时间参数，例如20200103

注意：
1、试跑就是在真实生产线上环境执行。



注意：
1、只有试跑成功的作业，可以上线。
2、失败的作业可以编辑、查看日志，重新发起试跑。