任务 3-2 认识运算符

【幻灯片 1-4】 任务3-2 认识运算符 - 任务描述

好，现在进入第二个任务——认识运算符。

[슬라이드

在Hive操作中，我们经常需要查询满足条件的数据。

比如：查找工资大于80000的员工，或者查找名字以'张'开头的人。

这些条件往往不是单一的，需要配合运算符来实现（shí xiàn）。

Hive中有四类运算符：

1. 关系运算符（guān xì yùn suàn fú）—— 用来比较两个值
2. 算术运算符（suàn shù yùn suàn fú）—— 用来做数学计算
3. 逻辑运算符（luó jí yùn suàn fú）—— 用来组合多个条件
4. 复杂运算符（fù zá yùn suàn fú）—— 用来访问复杂类型的元素

【幻灯片 5-6】 3.2.3 关系运算符 - 等于运算符 =

第一个关系运算符：等于运算符（=）。

它用来比较两边是否相等。相等返回true，不相等返回false。

可以比较数字，也可以比较字符串。

[여기서는 hive-bee를 사용하지 않고 그냥 hive 명령을 사용해서 실습]

🔧 实践操作: 等于运算符实践

```
select 3.14 = 3.142, 2.12 = 2.120, 'a' = 'a', 'apple' = 'pear';
-- 结果: false | true | true | false


-- 注意：2.12 = 2.120 是true！因为数值上是相等的
-- 'apple' = 'pear' 是false，因为两个字符串不一样
```

【幻灯片 7-8】 3.2.3 关系运算符 - 不等于运算符 <>

第二个：不等于运算符（<>），也可以写成!=。

和等于运算符相反：不相等返回true，相等返回false。

🔧 实践操作: 不等于运算符实践

```
select 3 <> 3, 5 <> 6, '5a' <> '5c';
-- 结果: false | true | true


-- 3 <> 3 是false，因为3等于3
-- 5 <> 6 是true，因为5不等于6
```

【幻灯片 9】 3.2.3 关系运算符 - 大小比较运算符

接下来是大于（>）、大于等于（>=）、小于（<）、小于等于（<=）运算符。

这些和数学里的含义完全一样。字符串比较是按字母顺序（shùn xù）来的。

🔧 实践操作: 大小比较实践

```
select 3 < 3, 5 > 6, 'a' >= 'c', 'A' <= 'B';
-- 结果: false | false | false | true


-- 3 < 3 是false（3不小于3，只是等于）
-- 'a' >= 'c' 是false（字母a排在c前面，所以a小于c）
-- 'A' <= 'B' 是true（大写A排在大写B前面）
```

【幻灯片 10-11】 3.2.3 关系运算符 - IS [NOT] NULL 运算符

IS NULL运算符用来判断一个值是不是NULL。

什么是NULL？NULL就是'空'，'没有值'，'不知道'的意思。注意：NULL不等于空字符串''！

IS NULL —— 如果是NULL就返回true

IS NOT NULL —— 如果不是NULL就返回true

🔧 实践操作: IS NULL实践

```
-- 'hello'不是NULL，所以IS NULL返回false，没有结果
select 1 from dual where 'hello' is null;
-- 结果：（空，没有行返回）


-- 'hello'不是NULL，所以IS NOT NULL返回true
select 1 from dual where 'hello' is not null;
-- 结果：1
```

💡 *dual是一个特殊的辅助（fǔ zhù）表，用来做简单测试的。*

【幻灯片 12-13】 3.2.3 关系运算符 - LIKE 运算符

LIKE运算符用来做字符串的模式匹配（mó shì pǐ pèi）。

有两个通配符（tōng pèi fú）：

% —— 代表零个或多个任意字符

_ —— 代表一个任意字符

举个例子：'he%' 可以匹配 'hello'、'help'、'he' 等任何以he开头的字符串。

🔧 实践操作: LIKE实践

```
-- 匹配以'he'开头的字符串
select 1 from dual where 'hello' like 'he%';
-- 结果：1 （匹配成功）


-- 匹配第二个字符是'e'的5个字符的字符串
select 1 from dual where 'hello' like '_ello';
-- 结果：1 （匹配成功）
```

【幻灯片 14-15】 3.2.3 关系运算符 - RLIKE 运算符

RLIKE是正则表达式（zhèng zé biǎo dá shì）匹配。比LIKE更强大。

正则表达式中常用的符号：

^ —— 表示字符串的开始

$ —— 表示字符串的结束

. —— 表示任意一个字符

* —— 表示前面的字符重复0次或多次

比如 '^h.*o$' 的含义是：以h开头，中间任意字符0个或多个，以o结尾。

🔧 实践操作: RLIKE实践

```
-- 匹配以h开头、以o结尾的字符串
select 1 from dual where 'hello' rlike '^h.*o$';
-- 结果: 1 （匹配成功，因为hello以h开头以o结尾）
```

【幻灯片 16】 3.2.3 算术运算符

算术运算符很简单，就是基本的数学计算：

+ 加法，- 减法，* 乘法，/ 除法，% 取余

这些和你在数学课上学的完全一样。

🔧 实践操作: 算术运算符实践

```
select 10 + 3;    -- 结果: 13
select 10 - 3;    -- 结果: 7
select 10 * 3;    -- 结果: 30
select 10 / 3;    -- 结果: 3.333...
select 10 % 3;    -- 结果: 1（10除以3余1）
```

【幻灯片 17-24】 3.2.3 逻辑运算符 - AND / OR / NOT

逻辑运算符用来组合多个条件：

AND —— 逻辑与（逻辑'和'）

两个条件都必须为true，结果才为true。就像：'你既要完成作业，又要通过考试，才能毕业。'

OR —— 逻辑或（逻辑'或者'）

只要有一个条件为true，结果就为true。就像：'作业或考试，通过一个就行。'

NOT —— 逻辑非（逻辑'取反'）

把true变成false，把false变成true。就像：'不是真的 = 假的'

🔧 实践操作: 逻辑运算符实践

```
-- AND: 两边都要true
select 1=1 and 8>10;          -- false（左边true，右边false → false）
select 'a'='a' and 'b'<'d';   -- true（两边都true → true）


-- OR: 至少一边true
select 1=1 or 8>10;           -- true（左边true就够了）
select 'a'='b' or 'b'>'d';    -- false（两边都false → false）


-- NOT: 取反
select not 8>9;               -- true（8>9是false，取反变true）
select not 1=1;               -- false（1=1是true，取反变false）
```

【幻灯片 26-32】 3.2.3 复杂运算符 - ARRAY运算符

复杂运算符用来访问复杂类型的元素。

ARRAY运算符用A[n]来访问数组的第n个元素，注意索引从0开始！

我们来做一个完整的实践：

首先创建一个包含ARRAY类型的表test1，然后加载数据文件，最后查询。

🔧 实践操作: ARRAY运算符完整实践

```
-- 步骤1: 创建表
CREATE TABLE test1 (
    person STRING,
    friends ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
COLLECTION ITEMS TERMINATED BY ',';


-- 步骤2: 准备数据文件 test1.txt（在Linux中创建）
-- 内容如下：
-- Lucy|Lucy_friend_0,Lucy_friend_1,Lucy_friend_2
-- Tom|Tom_friend_0,Tom_friend_1,Tom_friend_2,Tom_friend_3,Tom_friend_4
-- Sophie|Sophie_friend_0,Sophie_friend_1


-- 步骤3: 上传到HDFS
hdfs dfs -mkdir -p /user/hive/warehouse/test_data/
hdfs dfs -put test1.txt /user/hive/warehouse/test_data/test1.txt


-- 步骤4: 加载数据
LOAD DATA INPATH '/user/hive/warehouse/test_data/test1.txt'
INTO TABLE test1;


-- 步骤5: 查询
select * from test1;
select person, friends[0] from test1;  -- 查询每个人的第一个朋友
```

💡 *重要提示：数组元素之间不能有空格！否则Hive无法正确识别。*

【幻灯片 33-40】 3.2.3 复杂运算符 - MAP运算符

MAP运算符用M['key']来访问Map中指定key的value值。

🔧 实践操作: MAP运算符完整实践

```
-- 步骤1: 创建表
CREATE TABLE test2 (
    person STRING,
    address MAP<STRING, STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
COLLECTION ITEMS TERMINATED BY ','
MAP KEYS TERMINATED BY ':'
STORED AS TEXTFILE;


-- 步骤2: 准备数据文件 test2.txt
-- 张三|城市:北京,街道:朝阳区,邮编:100000
-- 李四|城市:上海,电话:13800138000,邮箱:lisi@example.com
-- 王五|年龄:25,部门:技术部,工资:8000
-- 赵六|公司:阿里巴巴,职位:工程师,工号:A1001


-- 步骤3: 上传并加载
hdfs dfs -put test2.txt /user/hive/warehouse/test_data/test2.txt
LOAD DATA INPATH '/user/hive/warehouse/test_data/test2.txt'
INTO TABLE test2;


-- 步骤4: 查询
select * from test2;
select person, address['城市'] from test2;  -- 查询每个人的城市
-- 注意：王五和赵六没有'城市'这个key，所以返回NULL
```

注意看查询结果：张三的城市是'北京'，李四的城市是'上海'，而王五和赵六因为MAP里没有'城市'这个键，所以返回NULL。

【幻灯片 47-48】 3.2.3 复杂运算符 - STRUCT运算符

STRUCT运算符用S.x来访问结构体中x字段的值。注意是用点号（.）来访问的。

🔧 实践操作: STRUCT运算符完整实践

```
-- 步骤1: 创建表
CREATE TABLE test3 (
    person STRING,
    info STRUCT<age:INT, city:STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
COLLECTION ITEMS TERMINATED BY ','
STORED AS TEXTFILE;


-- 步骤2: 准备数据文件 test3.txt
-- 张三|30,北京
-- 李四|25,上海
-- 王五|35,广州
-- 赵六|28,深圳
-- 钱七|40,杭州


-- 步骤3: 上传并加载
hdfs dfs -put test3.txt /user/hive/warehouse/test_data/test3.txt
LOAD DATA INPATH '/user/hive/warehouse/test_data/test3.txt'
INTO TABLE test3;


-- 步骤4: 查询
select * from test3;
select person, info.city from test3;    -- 查询每个人的城市
select person, info.age from test3;     -- 查询每个人的年龄
```

看查询结果：张三的城市是北京，李四的城市是上海...用点号访问STRUCT的字段，非常直观（zhí guān）。

> 💡 *三种复杂类型的访问方式总结：ARRAY用[索引]，MAP用['键名']，STRUCT用.字段名*

问题1. 关于Hive关系运算符，以下说法正确的是？

A) = 运算符只能用于数字比较

B) <> 运算符表示"大于"

C) IS NULL 用于判断表达式是否为NULL

D) LIKE 运算符不能使用%通配符

答案：C

解析： IS NULL 是专门用于判断表达式是否为NULL的关系运算符，

A错误因为=也可用于字符串比较，

B错误因为<>表示"不等于"，

D错误因为LIKE支持%（多个字符）和_（单个字符）通配符。

问题2. 以下Hive查询的结果是什么？

SELECT 3.14 = 3.140, 'apple' = 'pear';

A) true, true

B) false, false

C) true, false

D) false, true

答案：C

解析： 3.14和3.140在数值上是相等的，所以返回true；'apple'和'pear'是不同的字符串，所以返回false。

问题3. 关于RLIKE运算符，以下描述错误的是？

A) 是REGEXP_LIKE的同义词

B) 用于正则表达式匹配

C) ^h.*o$ 表示以h开头、以o结尾的字符串

D) 不能使用.匹配任意字符

答案：D

解析： RLIKE支持正则表达式，.在正则中表示匹配任意单个字符，所以D错误。

A、B、C都是RLIKE的正确描述。

问题4. 执行以下查询的结果是？

SELECT 5 <> 6, '5a' <> '5c';

A) false, false

B) true, false

C) false, true

D) true, true

答案：D

解析： <>表示"不等于"，5不等于6所以为true，'5a'不等于'5c'所以也为true

问题5. 关于算术运算符，以下说法正确的是？

A) % 用于求平方根

B) / 只返回整数部分

C) * 优先级高于 +

D) Hive不支持浮点数运算

答案：C

解析： 在Hive中，

算术运算符的优先级与其他语言相同：*、/、%优先级高于+、-。

A错误因为%是取余运算符，

B错误因为/返回精确的商（包括小数），

D错误因为Hive支持浮点数运算。

问题6. 逻辑运算符AND、OR、NOT中，优先级最高的是？

A) AND

B) OR

C) NOT

D) 三者优先级相同

答案：C

解析： 逻辑运算符的优先级从高到低是：NOT > AND > OR。

问题7. 以下查询的结果是什么？

SELECT NOT 8 > 9, NOT 1 = 1;

A) true, true

B) false, false

C) true, false

D) false, true

答案：C

解析： 8>9为false，NOT false为true；1=1为true，NOT true为false。

问题8. 对于**ARRAY类型字段friends，如何访问第一个元素？**

A) friends(0)

B) friends[1]

C) friends[0]

D) friends.first

答案：C

解析： Hive中ARRAY类型的索引从0开始，使用方括号访问元素，所以第一个元素是friends[0]。

问题9. 创建包含MAP类型的表时，键和值之间的分隔符使用哪个选项定义？

A) FIELDS TERMINATED BY

B) COLLECTION ITEMS TERMINATED BY

C) MAP KEYS TERMINATED BY

D) LINES TERMINATED BY

答案：C

解析： MAP KEYS TERMINATED BY用于定义MAP中键和值之间的分隔符。A用于字段间分隔，B用于集合项间分隔，D用于行分隔。

问题10. 对于**STRUCT类型字段info，如何访问其中的city属性？**

A) info["city"]

B) info.city

C) info->city

D) info(city)

答案：B

解析： STRUCT类型的字段使用点号.访问其内部属性，所以是info.city。A是MAP类型的访问方式。

问题11. 根据以下表定义，正确的数据格式是？

```
CREATE TABLE test2 (
    person STRING,
    address MAP<STRING, STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
COLLECTION ITEMS TERMINATED BY ','
MAP KEYS TERMINATED BY ':';
```

A) 张三,城市:北京,街道:朝阳区

B) 张三|城市:北京,街道:朝阳区

C) 张三|城市:北京|街道:朝阳区

D) 张三:城市,北京,街道,朝阳区

答案：B

解析： 根据分隔符定义：字段间用|，所以person和address之间用|；MAP项间用,，所以多个键值对之间用,；键值间用:，所以键和值之间用:。

问题12. 关于复杂运算符的描述，以下错误的是？

A) ARRAY[n] 返回数组的第n个元素

B) MAP[key] 返回指定key对应的value

C) STRUCT.x 返回STRUCT的x字段值

D) 所有复杂类型的索引都从1开始

答案：D

解析： ARRAY的索引从0开始，不是从1开始，所以D错误。A、B、C都是正确的复杂运算符使用方式。

问题13. LIKE 'he%' 可以匹配 'hello' 和 'he'。

A) 正确

B) 错误

答案：A

解析： %匹配0个或多个字符，所以'he%'可以匹配以'he'开头的任何字符串，包括'hello'和'he'。

问题14. 在Hive中，NULL = NULL 的结果是true。

A) 正确

B) 错误

答案：B

解析： 在Hive中，NULL = NULL的结果是NULL（未知），而不是true。需要使用IS NULL来判断是否为NULL。

问题15. 逻辑表达式 1=1 OR 8>10 AND 'a'='b' 的结果是true。

A) 正确

B) 错误

答案：A

解析： 根据优先级，先计算AND：8>10 AND 'a'='b'为false，然后计算OR：1=1 OR false为true。

问题16. 根据以下表定义，正确的数据格式是？

```
CREATE TABLE employee (
    name STRING,
    skills ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
COLLECTION ITEMS TERMINATED BY ',' ;
```

A) 张三 Python,Java,SQL

B) 张三,Python,Java,SQL

C) 张三|Python,Java,SQL

D) 张三 Python Java SQL

正确答案：A

解析： FIELDS TERMINATED BY '\t' 表示字段之间用制表符（Tab）分隔，COLLECTION ITEMS TERMINATED BY ',' 表示数组中的元素用逗号分隔。选项A使用Tab键分隔了名字和技能数组，符合定义。

问题17. 根据以下表定义，正确的数据格式是?

```
CREATE TABLE student_score (
    student_id INT,
    scores MAP<STRING, FLOAT>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '#'
MAP KEYS TERMINATED BY ':’ ;
```

A) 1001,数学:95.5#语文:88.0#英语:92.5

B) 1001#数学:95.5#语文:88.0#英语:92.5

C) 1001,数学#95.5,语文#88.0,英语#92.5

D) 1001:数学,95.5:语文,88.0:英语,92.5

正确答案：A

解析： FIELDS TERMINATED BY ',' 表示字段之间用逗号分隔，所以学号和MAP之间用逗号。COLLECTION ITEMS TERMINATED BY '#' 表示MAP中的每一对键值对之间用#分隔。MAP KEYS TERMINATED BY ':' 表示键和值之间用冒号分隔。