《Hive数据仓库应用教程》

项目三 Hive基本语法

——— 授课讲稿（jiǎng gǎo） ———

授课教授：李阳源

2026年3月5日

【幻灯片 1】 项目三 Hive基本语法 - 封面

同学们好！欢迎（huān yíng）来到今天的课程（kè chéng）。

今天我们要学习的是《Hive数据仓库应用教程》的第三章（zhāng）内容——Hive基本语法（yǔ fǎ）。

Hive是什么呢？简单说，Hive就是一个可以让我们用类似SQL的语言来处理（chǔ lǐ）大数据的工具。

如果你学过数据库（shù jù kù），那Hive对你来说就很容易理解。它的语法和MySQL很像。

好，我们正式开始吧！

【幻灯片 2】 目录（mù lù）

今天的课程一共分为四个部分（bù fen）：

第一部分：任务3-1 认识数据类型（shù jù lèi xíng）—— 就是了解Hive中有哪些数据类型。

第二部分：任务3-2 认识运算符（yùn suàn fú）—— 就是学习怎么做比较、计算这些操作。

第三部分：任务3-3 认识Hive命令（mìng lìng）—— 就是学会怎么在Hive里面输入命令。

第四部分：任务3-4 存储数据（cún chǔ shù jù）—— 就是了解Hive的数据是怎么存放的。

内容比较多，大家跟紧节奏（jié zòu 모두 잘 따라오세요），有问题随时举手提问。

【幻灯片 3】 大数据生态系统架构图

在正式学习Hive之前，我们先看一下这张架构图（jià gòu tú）。

这张图展示了整个大数据系统的层次结构（céng cì jié gòu）：

最下面一层是各种数据来源：数据库、数据接口（jiē kǒu）、FTP、门户（mén hù）网站(portal site:Baidu, google etc)、第三方系统。这些都是我们数据的来源。

往上看，数据进入HDFS——这是Hadoop的分布式文件系统（fēn bù shì wén jiàn xì tǒng），所有数据都存在这里。

HDFS上面是YARN——这是资源管理器（zī yuán guǎn lǐ qì），负责分配计算资源。

再往上，就是我们今天的主角——Hive！它和MapReduce、Spark一样，都是在YARN上面运行的。

图的右边还有Spark Streaming处理实时数据，

最右边的ZooKeeper负责协调（xié tiáo）整个系统。

> 💡 *记住：Hive的数据存在HDFS里，Hive通过MapReduce或Spark来处理数据。*

YARN: Yet Another Resource Negotiator

【幻灯片 4】 Hive系统架构图

这张图非常重要，展示了Hive的内部架构。

最上面是HiVE Client——也就是用户使用的客户端(**kè hù duān)**。

用户有3个接口可以连接Hive：

第一个是shell命令行(CLI:Comman Line Interpreter, **命令行界面**)，

第二个是JDBC/ODBC（这是Java或其他程序连接Hive的方式），

第三个是webUI网页界面。

中间部分是Hive Server和Driver（驱动器，qū dòng qì）。

Driver里面有四个重要的组件（zǔ jiàn）：

解释器（jiě shì qì : Parser）—— 把你写的SQL翻译（fān yì）成计算机能懂的指令。

计划器（ **jì huà qì** :Planner）—— 把指令编译成可执行的计划。

优化器（yōu huà qì:Optimizer）—— 让执行计划更高效（gāo xiào）。

执行器（zhí xíng qì:Execution）—— 真正去执行任务的。

右边有Hive Metastore——这是元数据存储（yuán shù jù cún chǔ），

记录了所有表的结构信息。

它可以用自带的Derby数据库，

也可以用MySQL。

最下面是Hadoop层，包括MapReduce计算引擎和HDFS存储系统。

> 💡 *简单理解：*
>
> 你写SQL → Hive翻译 → 变成MapReduce任务 → 在Hadoop上执行 → 返回结果。

任务 3-1 认识数据类型

【幻灯片 5】 任务3-1 认识数据类型 - 标题页

好，现在我们进入第一个任务(**rèn wu)**——认识数据类型。

什么是数据类型？举个例子：你的名字是文字（wén zì），你的年龄是数字，你的生日是日期（rì qī）——这些就是不同的数据类型。

在Hive中，每一列数据都必须指定（zhǐ dìng）一个类型，这样Hive才知道怎么处理它。

【幻灯片 6】 3.1.1 任务描述

这个任务中，我们要深入探讨（tàn tǎo）三个内容：

1. Hive的原生数据类型（yuán shēng shù jù lèi xíng）—— 就是最基本的类型，像整数、小数、字符串。

2. 复杂数据类型（fù zá shù jù lèi xíng）—— 比较高级的类型，像数组、映射。

3. 数据类型之间的转换（zhuǎn huàn）—— 比如把字符串"123"变成数字123。

为什么要学这些?

因为数据表的设计会直接影响（yǐng xiǎng）数据库的运行效率。

选对数据类型，查询就快；

选错了，可能会出错或者很慢。

【幻灯片 7】 3.1.2 Hive数据类型详解 - 整体概述

Hive的数据类型就是表中每一列（liè）的字段类型。

整体分为两大类：

第一类：原生数据类型（Primitive Data Type）

原生数据类型包括四种：[뒷장에서 구체적으로 설명함. 일일이 예를 들 필요 없음]

- 数值类型（shù zhí lèi xíng）：存数字的
- 时间日期类型（shí jiān rì qī lèi xíng）：存时间的
- 字符串类型（zì fú chuàn lèi xíng）：存文字的
- 杂项数据类型（zá xiàng shù jù lèi xíng）：布尔值、二进制这些

第二类：复杂数据类型（Complex Data Type）

复杂数据类型包括四种：

- ARRAY数组：一个字段存多个相同类型的值
- MAP映射（yìng shè）：存键值对
- STRUCT结构（jié gòu）：一个字段里面有多个不同类型的子字段
- UNION联合体：Hive对它的支持还不是很完整（wán zhěng）

看幻灯片上的树状图，左边是原生类型，右边是复杂类型，一共两大家族**(jiā zú)**。

【幻灯片 8】 3.1.2 原生数据类型分类图

这张图更详细地展示了原生数据类型的分类：

Numeric Types（数值类型）包括：TINYINT、SMALLINT、INT、BIGINT、FLOAT、DOUBLE、DECIMAL、NUMERIC 时间戳 (shí jiān chuō) 间隔 (jiàn gé)

Date/Time Types（日期时间类型）包括：TIMESTAMP、DATE、INTERVAL

String Types（字符串类型）包括：STRING、VARCHAR、CHAR

Miscellaneous Types（杂项类型）包括：BOOLEAN、BINARY

> 💡 *INT和DOUBLE是最常用的，STRING也非常常用。考试重点（zhòng diǎn）！*

【幻灯片 9】 3.1.2 数值类型和字符串类型详解

我们先来看数值类型，从小到大：

TINYINT —— 1个字节（zì jié），只能存 -128 到 127 的整数。很小，适合存年龄这种。

SMALLINT —— 2个字节，能存 -32768 到 32767。(负 (fù))

INT —— 4个字节，能存大约21亿。这是最常用的整数类型！

BIGINT —— 8个字节，能存非常大的整数。

看幻灯片右边的图：蓝色的柱子从左到右越来越高，就代表这些类型能存的数字范围（fàn wéi）越来越大。

FLOAT —— 4个字节的小数，精度（jīng dù）不太高。

DOUBLE —— 8个字节的小数，精度更高。看图中橙色的两个柱子，DOUBLE比FLOAT高。

DECIMAL —— 定点数（dìng diǎn shù），精度最高，适合存钱。

再来看字符串类型：

STRING —— 可变长度字符串，最常用！不用指定长度，想存多长就多长。

VARCHAR —— 需要指定最大长度，比如VARCHAR(100)最多存100个字符。

CHAR —— 固定长度，比如CHAR(10)永远占10个字符的空间。

> 💡 *实际工作中，数字用INT或BIGINT，小数用DOUBLE，文字用STRING就够了。*

- 마이너스 표시는 2’s complement(**二补码 :Èr bǔmǎ)** 보통 1의 보수는 **反码**, 2의 보수는 **补码** 또는 **二补码**라고 합니다.
- 방법은 모두 반전후 +1를 해줌
- 0을 포함한 **양수** 영역에 128개(0 0000000 ~ 0 1111111 → 0~127)를 할당하고,
- **음수** 영역에 나머지 128개(1 0000000 ~ 1 1111111)를 할당
- 그런데 이 음수 영역을 2의 보수 규칙으로 계산해보면, 가장 작은 수가 1 0000000 = -128이 되고, 가장 큰 음수가 1 1111111 = -1이 됩니다.
- 따라서 8비트로 표현 가능한 정수의 범위는 **-128부터 127까지**가 됩니다.

【幻灯片 10】 3.1.2 日期时间类型和其他类型

日期/时间类型：

TIMESTAMP —— 日期加时间，比如 '2026-03-05 10:30:00'。用Unix时间戳（shí jiān chuō）格式。

DATE —— 只有日期，比如 '2026-03-05'，没有时间部分。

其他类型：

BOOLEAN —— 布尔值（bù ěr zhí），只有两个值：true（真）和 false（假）。比如：'这个学生是否毕业？' 就用BOOLEAN。

BINARY —— 字节数组（zì jié shù zǔ），

用来存二进制数据，

比如图片、文件什么的。

一般用得不多。

【幻灯片 11】 3.1.2 复杂数据类型分类图

现在来看复杂数据类型。看这张图：

Hive的复杂类型有四种：array、map、struct、union。

array的写法是：array<datatype>，比如 array<string> 就是字符串数组。

map的写法是：map<key_type, value_type>，比如 map<string, int> 就是字符串到整数的映射。

struct的写法是：struct<col_name : data_type ...>，可以包含多个不同类型的字段。

union的写法是：UNIONTYPE<data_type, data_type, ...>，但是Hive对它支持还不完整，我们了解就好。

【幻灯片 12】 3.1.2 复杂数据类型详解

我用生活中的例子来解释这三种复杂类型：

ARRAY（数组）—— 像一个清单 (呐喊)

比如一个人会的技能：['Java', 'Python', 'SQL']。这就是一个字符串数组，

存的都是同一种类型的数据。

语法：ARRAY<data_type>，比如 ARRAY<STRING>

MAP（映射）—— 像一本字典

比如一个人的家属信息：{'配偶': 35, '儿子': 8}。每个键（jiàn）对应一个值（zhí）。

语法：MAP<primitive_type, data_type>，注意键必须是基本类型。

STRUCT（结构体）—— 像一张名片

比如一个人的地址信息：{street: '中关村', city: '北京', zip: 100080}。

不同的字段可以是不同类型。

语法：STRUCT<col_name:data_type, ...>

> 💡 ARRAY = 一列相同的值，MAP = 键值对，STRUCT = 多个不同字段。
>
> 记住这个区别！

【幻灯片 13】 3.1.2 注意事项

使用Hive数据类型有四个注意事项（zhù yì shì xiàng）：

第一，数据类型的英文字母大小写不敏感（mǐn gǎn）。也就是说，INT和int、Int都是一样的。

第二，除了SQL的数据类型，Hive还支持Java的数据类型。比如string就是Java的类型。

第三，使用复杂数据类型的时候，通常需要配合分隔符指定语法。后面实践的时候我们会看到。(반드시 읽어올 때 구분자 표시 : 다음장에 예제가 있음)

第四，如果你定义的数据类型和文件内容不一致，Hive会尝试隐式转换（yǐn shì zhuǎn huàn: **Implicit Conversion**），但不保证成功。比如你定义了INT类型，

但文件里是'hello'，那就会转换失败变成NULL。

【幻灯片 14】 3.1.3 创建表示例

现在我们来看一个完整的建表示例。这个表叫employees（员工表）：

🔧 实践操作: 创建employees表

```
CREATE TABLE employees (
    id INT,                                -- 员工ID，整数类型
    name STRING,                           -- 姓名，字符串类型
    salary DOUBLE,                         -- 工资，双精度浮点数
    is_active BOOLEAN,                     -- 是否在职，布尔类型
    skills ARRAY<STRING>,                  -- 技能数组
    dependents MAP<STRING, INT>,           -- 家属映射（姓名->年龄）
    address STRUCT<street:STRING, city:STRING, zip:INT>,  -- 地址结构体
    hire_date DATE                         -- 入职日期
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','                   -- 字段分隔符是逗号
COLLECTION ITEMS TERMINATED BY '|'         -- 集合元素分隔符是竖线(shù xiàn)
MAP KEYS TERMINATED BY '#'                 -- Map键值分隔符是井号( jǐng hào)
STORED AS TEXTFILE;                        -- 存储格式是文本文件
```

大家注意看：

前面四个字段是原生类型：INT、STRING、DOUBLE、BOOLEAN。

后面四个字段包含了复杂类型：ARRAY、MAP、STRUCT、DATE。

最重要的是下面的ROW FORMAT部分——这是告诉Hive怎么识别（shí bié）数据文件中的分隔符。

【幻灯片 15】 3.1.3 任务实施 - 创建表并查看结构

现在来看实际操作的截图：

首先启动Hive的beeline客户端。

然后输入CREATE TABLE命令创建表。

创建完成后，用show tables命令查看所有表，可以看到employees表已经创建好了。

再用describe employees命令查看表的结构，可以看到每个字段的名称和数据类型。

[여기서 실습 시행]

总的来说，运行Hive程序是在执行~/hadoop.sh后选择第1项。

我通过别名将beeline设置为hive-bee来进行实操。

今后请使用hive-bee命令。

1. VMWare 실행, Hive2026
2. Hadoop 선택, 비번:123456
3. 终端 실행
4. ~/hadoop.sh 실행
5. 1 번선택
6. jps 엔터（回车键）
7. hive-bee 엔터（回车键）
8. show databases;
9. create database if not exists unit3db;
10. use unit3db;
11. show tables;
12. 만일 employes 테이블이 존재하면 삭제하고 생성
13. drop table if exists employees;
14. 다음 내용을 입력

```
CREATE TABLE employees (
    id INT,                              -- 员工ID，整数类型
    name STRING,                          -- 姓名，字符串类型
    salary DOUBLE,                        -- 工资，双精度浮点数
    is_active BOOLEAN,                    -- 是否在职，布尔类型
```

```
    skills ARRAY<STRING>,                  -- 技能数组
    dependents MAP<STRING, INT>,           -- 家属映射 (姓名->年龄)
    address STRUCT<street:STRING, city:STRING, zip:INT>,  -- 地址结构体
    hire_date DATE                         -- 入职日期
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','                   -- 字段分隔符是逗号
COLLECTION ITEMS TERMINATED BY '|'         -- 集合元素分隔符是竖线(shù xiàn)
MAP KEYS TERMINATED BY '#'                 -- Map键值分隔符是井号( jǐng hào)
STORED AS TEXTFILE;                        -- 存储格式是文本文件
```

15. show tables;
16. desc employees;
17. 다음 내용을 레코드로 입력 (表创建好了，我们往里面插入两条员工数据)

【幻灯片 16-17】 3.1.3 插入数据并查询

表创建好了，我们往里面插入两条员工数据：

🔧 实践操作: 插入数据

```
INSERT INTO employees VALUES
(
    1, '张伟', 85000.50, true,
    ARRAY('Java', 'Python', 'SQL'),
    MAP('配偶', 35, '儿子', 8, '女儿', 5),
    NAMED_STRUCT('street', '中关村大街1号', 'city', '北京', 'zip', 100080),
    '2020-03-15'
),
(
    2, '李娜', 92000.75, true,
    ARRAY('Hadoop', 'Spark', 'Hive', 'Kafka'),
    MAP('配偶', 38, '母亲', 65),
    NAMED_STRUCT('street', '浦东新区陆家嘴环路1000号', 'city', '上海', 'zip',
200120),
    '2019-07-22'
);
```

注意看：

ARRAY用ARRAY()函数，MAP用MAP()函数，

STRUCT用NAMED_STRUCT()函数[1]。

插入完成后，用select * from employees查询所有数据。

可以看到skills显示为JSON数组格式，dependents显示为键值对格式，address显示为结构体格式。

🔧 实践操作: 查询数据

```
18. 查询所有数据
    select * from employees;
19. -- 查询部分字段
    select id, name, salary, is_active from employees;
```

- [1] struct：只负责把值组合在一起，不给它们起名字，Hive 会自动给你生成 col1, col2, col3... 这样的默认字段名。
- named_struct：在组合值的同时，允许你为每个值**指定一个自定义的字段名**，让数据的意义更清晰

【幻灯片 18】 3.1.4 Hive类型转换 - 隐式转换(implicit transformation)

接下来学习类型转换。Hive支持两种转换方式：隐式转换和显式转换。

隐式转换（yǐn shì zhuǎn huàn）：

什么是隐式转换?

就是Hive自动帮你做的类型转换，你不用写任何代码。

规则很简单：只能从窄类型转到宽类型，不能反过来。

什么意思？看幻灯片上的图：

tinyint → smallint → int → bigint → float → double

就像水只能从小瓶子倒到大瓶子，不能从大瓶子倒到小瓶子。

比如你把一个INT加上一个DOUBLE，Hive会自动把INT转成DOUBLE再计算。

看幻灯片上的转换表：void可以转成任何类型，boolean只能是boolean，tinyint可以转成更大的数值类型。

> 💡 *记住规则：*小的可以自动变大的，大的不能自动变小的！



兔子 (Tùzi)不能住进长颈鹿(chángjǐnglù)的家。

但是，兔子可以住进长颈鹿的家。

【幻灯片 19】 3.1.4 Hive类型转换 - 显式转换

显式转换（xiǎn shì zhuǎn huàn）：

显式转换就是你主动告诉Hive要把数据转换成什么类型。使用CAST函数。

语法：CAST(值 AS 目标类型)

比如：CAST('100' AS INT) —— 把字符串'100'转成整数100。

CAST(12 AS DOUBLE) —— 把整数12转成小数12.0。

如果转换失败怎么办？比如 CAST('Allen' AS INT)，'Allen'不是数字，转换失败就返回NULL。

🔧 实践操作: 类型转换实践

```
-- 整数转双精度
select cast(12 as double);           -- 结果: 12.0

-- 字符串转整数
select cast('123' as int);           -- 结果: 123

-- 转换失败返回NULL
select cast('hello' as int);         -- 结果: NULL
```

【幻灯片 20】 3.1.5 Hive类型转换任务实施

来看更多实践示例：(중국어 입력이 원활하지 않으면 ctrl+space를 이용)

🔧 实践操作: 完整的类型转换实践

```
-- 隐式转换：INT 1 + DOUBLE 2.5 = DOUBLE 3.5
select 1 + 2.5;                      -- 结果: 3.5


-- 显式转换：字符串转整数
select cast('123' as int);           -- 结果: 123


-- 显式转换：DOUBLE转STRING
select cast(salary as string) from employees;


-- 访问ARRAY的元素（索引从0开始）
select skills[0] from employees;     -- 第一个技能


-- 访问MAP的值
select dependents['儿子'] from employees;


-- 访问STRUCT的字段
select address.city from employees;
```

特别注意：数组的索引（suǒ yǐn）从0开始！第一个元素是[0]，不是[1]。

问题1

以下哪个是Hive中的有效数值类型？

A) NUMBER

B) INTEGER

C) NUMERIC

D) TINYINT

答案：D ✅

问题2

在Hive中，可以存储键值对集合的数据类型是什么？

A) ARRAY

B) SET

C) MAP

D) DICTIONARY

答案：C ✅

问题3

关于Hive的STRING和VARCHAR数据类型，以下哪个说法是正确的？

A) STRING和VARCHAR没有区别，可以互换使用

B) VARCHAR有最大长度限制，STRING没有

C) STRING有最大长度限制，VARCHAR没有

D) 两者都不支持中文字符存储

答案：B ✅

问题4

以下哪个Hive数据类型可以存储不同类型的字段组合？

A) ARRAY<STRUCT>

B) MAP

C) UNIONTYPE

D) STRUCT

答案：D ✅

解析：STRUCT可以包含多个不同类型的字段，如STRUCT<name:STRING, age:INT>

问题5

创建包含数组类型的表时，哪个子句用于指定数组元素之间的分隔符？

A) FIELDS TERMINATED BY

B) COLLECTION ITEMS TERMINATED BY

C) MAP KEYS TERMINATED BY

D) ARRAY ELEMENTS TERMINATED BY

答案：B ✅

问题6 (新增：显式转换问题)

在Hive中，将字符串'123.45'转换为DECIMAL(5,2)类型的正确语法是什么？

A) TO_DECIMAL('123.45', 5, 2)

B) DECIMAL('123.45', 5, 2)

C) CAST('123.45' AS DECIMAL(5,2))

D) CONVERT('123.45', DECIMAL(5,2))

答案：C ✅

解析：Hive中使用CAST函数进行显式类型转换，语法为CAST(expression AS data_type)

课程总结（kè chéng zǒng jié）

好，今天的课程到这里就结束了。我们来回顾（huí gù）一下今天学了什么：

任务3-1 认识数据类型：

- 原生类型：数值（INT/DOUBLE）、字符串（STRING）、日期（DATE）、布尔（BOOLEAN）
- 复杂类型：ARRAY数组、MAP映射、STRUCT结构体
- 类型转换：隐式转换（小→大自动转）、显式转换（CAST函数）

任务3-2 认识运算符：

- 关系运算符：=, <>, >, <, >=, <=, IS NULL, LIKE, RLIKE
- 算术运算符：+, -, *, /, %
- 逻辑运算符：AND, OR, NOT
- 复杂运算符：A[n], M['key'], S.x

任务3-3 认识Hive命令：

- 两代客户端：Hive CLI 和 Beeline CLI
- HDFS命令、Linux命令、Tab补全、历史命令

任务3-4 存储数据：

- 行式存储 vs 列式存储
- TextFile（默认）、SequenceFile、RCFile、ORC

大家课后要多实践（shí jiàn），把今天学的命令都敲（qiāo）一遍。

有问题随时问！

谢谢大家！下课！