# Hive4 数据类型和文件格式

## 基本数据类型

| 数据类型 | 长度 | 例子 |
| --- | --- | --- |
| TINYINT | 1byte有符号整数 | 20 |
| SMALINT | 2byte有符号整数 | 20 |
| INT | 4byte有符号整数 | 20 |
| BIGINT | 8byte有符号整数 | 20 |
| BOOLEAN | 布尔类型 | TRUE |
| FLOAT | 单精度浮点数 | 3.14159 |
| DOUBLE | 双精度浮点数 | 3.14.59 |
| STRING | 字符序列。可以指定字符集 | 'hello' |
| TIMESTAMP | 整数，浮点数或者字符串 | 12312;1231.1232;'2012-03-03' |
| BINARY | 字节数组 | |

## 集合数据类型

| 数据类型 | 描述 | 字面语法示例 |
| --- | --- | --- |
| STRUCT | 跟对象类似，可以通过点访问 | struct('John','Doe) |
| MAP | MAP键值对 | map('first','JOIN','last','Doe') |
| ARRAY | ARRAY相同数组集合 | Array('John','Doe') |

例子:

```
CREATE TABLE employees(
    name        STRING,
    salary      FLOAT,
    subordinates    ARRAY<STRING>,
    deductions  MAP<STRING, FLOAT>,
    adress      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>)
```

## Hive与SQL的不同

1. Hive不支持提供最大长度的“字符数组”类型。

> 关系型数据库提供这个功能处于性能优化的考虑，因为定长的记录更容易建立索引、数据扫描。 Hive 所处的世界里，不一定拥有数据文件但是必须支持使用不同的文件格式。Hive根据不同字段间的分隔符进行判断。

# 文本格式

| 分隔符 | 描述 |
| --- | --- |
| \n | 对于文本来说，每行都是一条记录，因此换行符可以分割记录 |
| ^A(Ctrl+A) | 用于分隔字段（列），在CREATE TABLE语句中可以使用八进制编码\001表示 |
| ^B | 用于分隔ARRAY或者STRUCT中的元素，或用于MAP中键-值对之间的分隔。在CREATE TABLE 语句中可以用八进制编码/002表示 |
| ^C | 用于MAP中键和值之间的分隔，在CREATE TABLE 可以用八进制编码 /003 表示 |

例子：

```
Jhon Doe^A10000.0^AMary Smith^BTodd Jones^AFederal
Taxes^C.2^BStateTaxes^C.05^BInsurance^C.1^A1 Michigan Ave.^BIL^B60600

换成json格式数据如下：
{
    "name":"Jhon Doe",
    "salsry":100000.0,
    "subordinates":["mart Smith","Todd Jones"],
    "deductions":{
        "Federal Taxes":.2,
        "State Taxes":.05,
        "Insurance":.1
    },
    "address":{
        "street":"1 Michingan Ave.",
        "city":"Chicago",
        "state":"IL",
        "zip":60600
    }
```

用户可以不使用这些默认的分隔符，而指定其他分隔符。下面是指定了分隔符的制表框架：

```
CREATE TABLE employees(
    name        STRING,
    salary      FLOAT,
    subordinates    ARRAY<STRING>,
    deductions  MAP<STRING, FLOAT>,
    adress      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>)
ROW FORMAT DELIMITED  //必须卸妆其他子句之前
FIELDS TERMINATED BY '\001'  //将^B作为集合的分隔符
COLLECTION ITEMS TERMINATED BY '\002' //^C作为map的键和值之间的分隔符
MAP KEYS TERMINATED BY '\003' //键值对
LINES TERMINATED BY '\N'
STORED AS TEXTFILE;
```