HBase 的表结构

HBase 是一个NoSQL数据库，用于处理海量数据，可以支持10亿行百万列的大表，下面就了解一下数据是如何存放在HBase表中的

**关系型数据库的表结构**

为了更好的理解HBase表的思路，先回顾一下关系数据库中表的处理方式

例如有一个用户表user\_info，有字段：id、name、tel，**表名**和**字段**需要在建表时指定

create table user\_info (

    id 类型,

    name 类型,

    tel 类型

)

然后插入两条数据

insert into user\_info values(...)

表结构

| **id** | **name** | **tel** |
| --- | --- | --- |
| 1 | 小明 | 123 |
| 2 | 小王 | 456 |

后来字段不够用了，新用户需要记录**地址**，就要新增一个字段

| **id** | **name** | **tel** | **addr** |
| --- | --- | --- | --- |
| 1 | 小明 | 123 |  |
| 2 | 小王 | 456 |  |

以后再增加需求时，就继续新增字段，或者添加一个扩展表

上面的内容主要说明的是：

* 建表的方式，需提前指定表名和字段
* 插入记录的方式，指定表名和各字段的值
* 数据表是二维结构，行和列
* 添加字段不灵活

下面看一下HBase的处理方式

**HBase的表结构**

建表时要指定的是：表名、列族(可多个，一般建议不要设置多个列族。)

建表语句

**create 'user\_info', 'base\_info', 'ext\_info'**

意思是新建一个表，名称是user\_info，包含两个列族base\_info和ext\_info

**列族** 是列的集合，一个列族中包含多个列

这时的表结构：

| **row key** | **base\_info** | **ext\_info** |
| --- | --- | --- |
| ... | ... | ... |

**row key** 是行键，每一行的ID，这个字段是自动创建的，建表时不需要指定

插入一条用户数据：name为‘a’，tel为‘123’

插入语句

**put 'user\_info', 'row1', 'base\_info:name', 'a'**

**put 'user\_info', 'row1', 'base\_info:tel', '123'**

意思是向user\_info表中**行健**为row1的base\_info列族中添加一项数据 name:a，接着又添加一项数据tel:123

name和tel就是具体字段，属于base\_info这个列族

这时的表结构：

| **row key** | **base\_info** | **ext\_info** |
| --- | --- | --- |
| row1 | name:a, tel:123 |  |

再插入一条数据：name为‘b’，addr为‘beijing’

**put 'user\_info', 'row2', 'base\_info:name', 'b'**

**put 'user\_info', 'row2', 'ext\_info:addr', 'bj'**

这时的表结构：

| **row key** | **base\_info** | **ext\_info** |
| --- | --- | --- |
| row1 | name:a, tel:123 |  |
| row2 | name:b | addr:bj |

HBase表中还有一个重要概念：**版本**，每个字段的值都有版本信息（通过时间戳指定）

例如 base\_info:name，每次修改时都会保留之前的值，就是说可以取到他的旧值

| **row key** | **base\_info** | **ext\_info** |
| --- | --- | --- |
| row1 | name:a, tel:123 |  |
| row2 | name:c(v2)[name:b(v1)] | addr:bj |

**小结**

从上面建表、插入数据的过程可以看出 HBase 存储数据的特点了

* 和关系数据库一样，也是使用行和列的结构
* 建表时，定义的是表名和列族（字段的集合），而不是具体字段
* 列族中可以包含任意个字段，字段名不需要预定义，每一行中同一列族中的字段也可以不一致
* 多维结构，关系数据库的表是二维的，通过指行、列定位一个数据，HBase中需要通过 行健、列族名、字段名、版本号才能定位到具体数据
* 插入数据时，一次插入一个字段的数据，不是像关系数据库那样一次插入多个字段