# OceanBase的数据类型

OceanBase支持的数据类型有 int, varchar, datetime, precise\_datetime, create\_time, modify\_time.

类型说明:

int 类型是一个范围在[-2^63, 2^63)的整数.

varchar 类型可以用于存储字节流, 需要同时给出最大可容纳的字节数. 如果插入或者更新成一个大于最大字节数的值, 系统会拒绝该次操作.

datetime 类型是一个数字, 表示的是从1970年1月1日0时以来的秒数, 精确到秒.

precise\_datetime类型是一个数字, 表示的是从1970年1月1日0时以来的秒数, 精确到微秒.

create\_time 是和precise\_datetime相同的一种时间表示, 该值由系统在数据产生的时候自动生成, 不可修改.

modify\_time 是和precise\_datetime相同的一种时间表示, 该值由系统在数据更新的时候自动修改.

# OceanBase的schema格式

OceanBase 中的schema 表示为纯文本的ASCII码文件, 采用常见的配置文件的形式. 分成各个 section. 每个section 下有多个配置项, 配置项名称和配置值之间用”=”连接.

OceanBase 中 schema 是以应用为单位的, 一个应用一个schema文件. 一个应用中可以包含多张表, 每张表中可以包含多个列, 以及多个联表(join)关系.

所以 OceanBase 中的schema 分成一个应用信息配置部分和多个表信息配置部分.

下面结合一个实例来解释 schema 的各个部分

## 应用的信息

实例:

[app\_name]

name=collect

max\_table\_id=1003

解释:

应用的信息都写在 app\_name 这个section 中. 目前主要有两个配置项:

name 用来配置应用的名称, 是一个长度不超过128位的字符串.

max\_table\_id 用来记录当前已经使用的最大的table\_id. 在OceanBase中, 每个表都由table\_id唯一标识, 且table\_id不可以被重复使用. max\_table\_id 这个配置项, 主要是为了方便 schema 生成程序记录已经使用过的table\_id.

## 表的信息

实例:

[collect\_item\_info]

table\_id=1001

table\_type=1

column\_info=0,2,item\_name,varchar,20

column\_info=0,3,new\_price,int

rowkey\_split=0

rowkey\_max\_length=9

max\_column\_id=3

[collect\_info]

table\_id=1002

table\_type=2

column\_info=1,2,item\_name,varchar,20

column\_info=1,3,item\_price,int

rowkey\_split=8

join=rowkey[8,16]% collect\_item\_info:item\_name$item\_name,item\_price$new\_price

rowkey\_max\_length=17

max\_column\_id=3

解释:

每张表的信息都存放在以表名命名的section中. 所以上面是两张表的配置.

table\_id 配置项配置了这张表在OceanBase系统中的唯一id, 由schema 生成工具自动生成. 在OceanBase系统中, id的取值范围是0-65535.系统会保留0-1000的table\_id供系统自身使用.

table\_type 用来配置表是内存表还是磁盘表. 因为一张表的动态数据存储在UpdateServer上,动态部分不受table\_type影响, 所以table\_type实际上表示了ChunkServer上的静态部分是放到内存中还是放到磁盘上. 该配置项取值为 1 的时候, 表示静态部分放到磁盘上, 为2的时候, 表示静态部分放到内存中.

rowkey\_max\_length 用来配置表中主键的最大长度.

rowkey\_split 配置表在存储的时候的拆分限制. 因为一个表的数据可能放到多个tablet上存储, 这个值告诉ChunkServer, 在分裂数据到不同tablet时哪些数据是不应该被分开的, 比如, 当这个值为9的时候, 表示主键前9个字节完全相同的记录不应该被分到两个不同的tablet中.

max\_column\_id 配置本表中已经使用过的最大的列id, 由schema 生成程序维护并使用, 防止对列id的重用.

compress\_func\_name 可选项, 配置这个表在存储时使用的压缩算法名字.

block\_size 可选项, 配置表在存储成sstable时,采用的block大小.

use\_bloomfilter 可选项, 配置表是否使用布隆过滤器, 非零值为使用.

rowkey\_is\_fixed\_length 可选项, 配置主键是否是固定长度值. 非零值表示主键是固定长度的. 如果不配置该项, 默认主键为固定长度.

### 列的配置:

column\_info 配置项中的内容是具体描述一列的, 用”,”分开, 其内容包含列属性, 列id, 列名, 列类型.

列的属性：取值为0或者1. 为0表示该列只有动态数据(只存在于UpdateServer); 为1表示该列既有动态数据又有静态数据(既存在于UpdateServer 又存在于 ChunkServer).

列id：是这个列在表中的唯一标识, 该值由schema生成程序生成, 不可以被重用. 列id必须大于1, 系统保留id为1的用于表示主键.

列名：是一个长度不超过128位的字符串.

类型：列的数据类型.

所以例子中的

column\_info=0,2,item\_name,varchar,20

表示 列名字是 item\_name; id 是2; 只有动态数据(只存在于UpdateServer,不存在于ChunkServer) 类型是 varchar, 长度是20个字符.

因为OceanBase的联表(join)设计,使得某些只有动态数据的列是有意义的, 这种列一般在转储过程中会通过join运算转储到其它表中.

### 联表(Join)关系的配置

联表(join)关系是OceanBase提供的简化关联查询的有力手段.

join 这个配置项里描述的是join关系的具体内容

我们看一个例子:

join=rowkey[8,16]% collect\_item\_info:item\_name$item\_name,item\_price$new\_price

这一行表示当前表的一个join关系.

join=rowkey[8,16]%collect\_item\_info 是表示用当前记录的主键的第8-16字节(闭区间) 与表collect\_item\_info 进行join操作. “:”后的内容表示具体发生join的列.

用”$”分开的两个列分别被称为参与join操作的左列和右列. join操作总是用右列的值合并到左列的值上, 然后将合并的结果返给用户(左列和右列的值都不发生变化, 合并只体现在反给用户的结果中).

所以上述的表达是说, 当访问当前表的时候, 如果访问到列 item\_name 或者 item\_price. 则需要以当前主键的 8-16 字节为主键查找表 collect\_item\_info. 如果查到记录, 则用其item\_name 列的值与当前记录的item\_name的值做合并, 用其new\_price列的值与当前记录的item\_price的值做合并, 将合并的结果作为最终值返给客户.

注意: 在进行联表(join)操作的时候, 要求被join的两个列具有相同的类型, 唯一的例外是create\_time, modify\_time 这两个类型可以与precise\_datetime进行join, 而且create\_time, modify\_time只能作为join的右列. 这是因为create\_time, modify\_time这两个类型是由系统负责维护的, 不能通过join关系修正, 但是他们能够以join关系中右列的身份修正一个precise\_datetime类型的列.

## Schema 的兼容性

当我们对OceanBase中的schema进行修改的时候, 新的schema 必须与老的schema兼容才能被系统接受. 所以兼容性其实就是系统所允许的对OceanBase中的表结构的修改操作.

目前我们对是否兼容有如下的约定:

1. 表的增加, 删除, 都是兼容的. OceanBase支持增加新的表以及删除已经存在的表.
2. 列的改名是兼容的, 更改列的类型不兼容. OceanBase支持更改列名,但是不能修改列的类型.
3. 对列的增加或者删除是兼容的.OceanBase支持增加新的列或者删除已经存在的列.
4. 修改table\_type是兼容的.OceanBase支持把一个磁盘表调整为内存表,反之亦然.
5. 修改rowkey\_split或者修改rowkey\_max\_length都是兼容的.
6. 增加或者删除join关系是兼容的. OceanBase支持删除联表关系或者新增联表关系
7. 修改应用名称(app\_name)不兼容. OceanBase不支持更改应用的应用名.