**OceanBase内部表定义**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **文档版本** | **修订章节** | **修订原因** | **修订日期** | **修订人** |
| **1** | 0.1 |  | 新建文档 | 2011-11-11 | 正祥 |
| **2** | 0.2 |  | 增加内部表分类描述，增加新的内部表 | 2011-11-21 | 剑鸣 |
| **3** | 0.3 |  | 修改内部表描述，增加join关系，增加\_\_all\_join\_info、\_\_all\_data\_type、\_\_all\_server等内部表 | 2011-11-24 | 正祥 |
| **4** | 0.31 |  | 增加\_\_all\_cluster，用服务器IP+PORT代替服务器名称即ID | 2011-11-25 | 正祥 |
| **5** | 0.32 |  | 增加\_\_all\_all\_index，一级元数据表总是指向二级元数据表，IPv4与IPv6分开 | 2011-11-30 | 正祥 |
| **6** | 0.33 |  | 增加\_\_first\_tablet\_entry和元数据表，去掉了\_\_primary\_tablet\_ meta和\_\_secondary\_tablet\_ meta，\_\_all\_all\_table变成视图(暂不实现) | 2011-12-1 | 正祥 |
| **7** | 0.34 |  | \_\_first\_tablet\_entry的replica增加条目 \_\_all\_sys\_stat增加table\_id列和value1, value2列 | 2012-4-18 | 剑鸣 |
| **8** | 0.35 |  | \_\_first\_tablet\_entry增加列tablet\_max\_size和create\_mem\_version | 2012-4-24 | 竹翁 |

[1 OceanBase内部表简介 3](#_Toc323053510)

[2 名词解释 3](#_Toc323053511)

[3 OceanBase内部表定义 3](#_Toc323053512)

[3.1 \_\_first\_tablet\_entry 4](#_Toc323053513)

[3.2 tablet元数据表 5](#_Toc323053514)

[3.3 \_\_all\_all\_table 5](#_Toc323053515)

[3.4 \_\_all\_all\_index 6](#_Toc323053516)

[3.5 \_\_all\_all\_column 6](#_Toc323053517)

[3.6 \_\_all\_join\_info 7](#_Toc323053518)

[3.7 \_\_all\_data\_type 7](#_Toc323053519)

[3.8 \_\_all\_sys\_param 7](#_Toc323053520)

[3.9 \_\_all\_sys\_stat 8](#_Toc323053521)

[3.10 \_\_all\_cluster 9](#_Toc323053522)

[3.11 \_\_all\_server 9](#_Toc323053523)

本文档对淘宝OceanBase( <http://oceanbase.taobao.org> )开源数据库的内部表及约定进行了定义和说明。

# OceanBase内部表简介

作为一个数据库系统，OceanBase需要允许数据库应用人员和管理人员(DBA)获取和管理系统的表结构、获取和设置各种功能和性能参数、查看各种内部运行状态等等，这些都通过各种内部表实现。

本文对OceanBase的内部表进行了定义和说明。

为了区别用户定义的表，OceanBase的内部表的名称都以下划线“\_”开头，目前只使用了两个下划线(“\_\_”)开头的名字，其他名字留待日后定义。

# 名词解释

**Update Server**：OceanBase的增量更新子系统，记录用户的修改，通常由一主一备或一主多备构成；

**ChunkServer**：OceanBase的基线数据存储子系统，由多台机器构成，基线数据通常保存2~3副本并且保存在不同的ChunkServer上；

**MergeServer**：OceanBase的用户接口子系统，接收用户的查询请求，并从对应ChunkServer上获得基线数据以及从UpdateServer上获得更新的增量，然后融合二者后返回用户；

**RootServer**：OceanBase的管理服务器，用于管理UpdateServer、ChunkServer和MergeServer等

**表**：一个表由若干列和任意行组成，通常，其中的部分列构成主键(row key)，整个表按主键顺序存储；

Tablet：表以行为单位划分为若干个连续的条带，每个条带称为一个tablet。每个tablet有一个begin row key(该tablet的row key的下限，不包含在tablet内)和一个end row key(该tablet的row key的上限，包含在该tablet内，如果以该end row key为row key的行存在的话)。除了第一个tablet，每个tablet的begin row key都是前一个tablet的end row key；除了最后一个tablet，每个tablet的end row key都是后一个tablet的begin row key。每个tablet通常以sstable方式保存在两个或更多个ChunkServer上，每个ChunkServer上的sstable文件称为该tablet的一个副本；

**schema**：表的列的类型、值范围等以及该表与其他表的join等关系称为表的schema；

# OceanBase内部表定义

OceanBase内部表，包含了以下几类：

* 内部数据，例如tablet信息表\_\_primary\_tablet\_meta和\_\_secondary\_tablet\_meta，所有表汇总表\_\_all\_all\_table，所有表的column定义表\_\_all\_all\_column等。除了元数据表自身外，所有其他表的tablet信息都保存在元数据表中；
* 内部参数，比如\_\_sys\_param记录了RS/CS/MS/UPS的各种配置参数及其含义；
* 内部状态，比如\_\_sys\_stat记录了RS/CS/MS/UPS的各种状态值及其含义。

以下是OceanBase定义的内部表：

* \_\_first\_tablet\_entry：表的第一个tablet的位置信息
* 元数据表：表的tablet的位置信息等
* \_\_all\_all\_table：所有表(包括内部表和用户定义表)的表名和table\_id，目前暂时没有实现
* \_\_all\_all\_index：所有索引
* \_\_all\_all\_column：所有表的列名、column\_id及其类型
* \_\_all\_join\_info：内部join关系
* \_\_all\_data\_type：内部数据类型
* \_\_all\_sys\_param：系统配置参数及其含义
* \_\_all\_sys\_stat：系统状态值及其含义
* \_\_all\_cluster：整个系统中的所有机群
* \_\_all\_server：整个系统中所有的服务器

所有内部表都包含gm\_create列(column\_id为2)和gm\_modified列(column\_id为3)，用于记载该行创建和最后一次修改的时间，后面的描述中不再重复。目前保留了0和1的列ID。

## \_\_first\_tablet\_entry

首个tablet入口表\_\_first\_tablet\_entry记录了每个表的第一个tablet的副本所在的ChunkServer列表。如果一个表的内容不超过一个tablet，那么从\_\_first\_tablet\_entry即可定位该表的唯一tablet并进行访问；如果一个表的内容超过了一个tablet，则需要通过此表的元数据表(meta\_table\_name/id)获得所需的tablet的位置并进行访问。**\_\_first\_tablet\_entry表的table\_id固定为1且只有一个tablet，其唯一的tablet的位置信息保存于RootServer**。

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| table\_name | vchar(??????) | 表名 |
| table\_id | Int | 表的table\_id |
| table\_type | Int | 1：普通表；2：索引；3：元数据表；4：view；5：临时表； |
| load\_type | Int | 1：on disk；2：load into memory on demand |
| table\_def\_type | Int | 1：内部表；2：用户定义表 |
| rowkey\_column\_num | Int | 主键的列数，后续endrowkeyobj1, endrowkeyobj2…等来依次表示主键的列 |
| column\_num | Int | 全部的列数(包括主键) |
| max\_used\_column\_id | Int | 该表使用过的最大列ID(列ID不重用) |
| replica\_num | Int | 单个机群的tablet的replica的个数(1~6)，后续用cs1，…，cs6来表示这些replica所在的cs |
| tablet\_max\_size | Int | 该表每个tablet的sstable文件最大允许大小 |
| create\_mem\_version | Int | 新建该表时候系统的mem\_version，作用见《OceanBase系统自举流程》 |
| server1\_ipv4 | Int | 第1个replica的server的IPv4地址(不使用时为0) |
| server1\_ipv6\_high | Int | 第1个replica的server的IPv6地址的高64位(不使用时为0) |
| server1\_ipv6\_low | Int | 第1个replica的server的IPv6地址的低64位(不使用时为0) |
| server1\_ip\_port | Int | 第1个replica的IP地址端口(不使用时为0) |
| server1\_version | Int | 第1个replica的版本号(不使用时为0) |
| server1\_row\_count | Int | 第1个replica的包含的行数(不使用时为0) |
| server1\_size | Int | 第1个replica占用空间大小(不使用时为0) |
| server2\_ipv4 | Int | 第2个replica的server的IPv4地址(不使用时为0) |
| server2\_ipv6\_high | Int | 第2个replica的server的IPv6地址的高64位(不使用时为0) |
| server2\_ipv6\_low | Int | 第2个replica的server的IPv6地址的低64位(不使用时为0) |
| server2\_ip\_port | Int | 第2个replica的IP地址端口(不使用时为0) |
| server2\_version | Int | 第2个replica的版本号(不使用时为0) |
| server2\_row\_count | Int | 第2个replica的包含的行数(不使用时为0) |
| server2\_size | Int | 第2个replica占用空间大小(不使用时为0) |
| …… |  |  |
| …… |  |  |
| meta\_table\_name | vchar(??????) | 本表的tablet信息的元数据表的名字：\_\_meta\_+表名，如果一个表很大，其元数据表也超过了一个tablet，则其元数据表的元数据表的名字是： \_\_meta\_\_\_meta\_+表名。为空串表示没有元数据表。 |
| meta\_table\_id | Int | 为0表示该表没有元数据表(比如只有一个tablet的表和多数元数据表本身)；正整数：元数据表的表id |

Rowkey：table\_name

为了避免循环引用，\_\_first\_tablet\_entry不包含自身。

与其他表一样，此表在\_\_all\_all\_column中有定义。除了上述列(必须存在)，此表也可能有其他列。

## tablet元数据表

每个表，如果其数据超过一个tablet，则需要有对应的元数据表，所以系统中可能有多个元数据表。如果一个元数据表超过了一个tablet，则该元数据表本身也会有一个元数据表。

元数据表的每一行定义了一个tablet的endrowkey和该tablet副本所在的ChunkServer。

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| table\_id | int | 表的table\_id |
| rowkey\_column\_num | int | 主键的列数，后续endrowkeyobj1, endrowkeyobj2…等来依次表示主键的列 |
| startrowkeyobj1 | any | 该tablet的startkey的主键的第1列 |
| startrowkeyobj2 | any | 该tablet的startkey的主键的第2列 |
| … |  |  |
| … |  | 该tablet的startkey的主键的第rowkey\_column\_num列 |
| endrowkeyobj1 | any | 该tablet的endkey的主键的第1列 |
| endrowkeyobj2 | any | 该tablet的endkey的主键的第2列 |
| … |  |  |
| … |  | 该tablet的endkey的主键的第rowkey\_column\_num列 |
| server1\_ipv4 | int | 第1个replica的server的IPv4地址(不使用时为0) |
| server1\_ipv6\_high | int | 第1个replica的server的IPv6地址的高64位(不使用时为0) |
| server1\_ipv6\_high | int | 第1个replica的server的IPv6地址的高64位(不使用时为0) |
| server1\_ip\_port | int | 第1个replica的IP地址端口(不使用时为0) |
| …… |  |  |
| server6\_ipv4 | int | 第6个replica的server的IPv4地址(不使用时为0) |
| server6\_ipv6\_high | int | 第6个replica的server的IPv6地址的高64位(不使用时为0) |
| server6\_ipv6\_high | int | 第6个replica的server的IPv6地址的高64位(不使用时为0) |
| server6\_ip\_port | int | 第6个replica的IP地址端口(不使用时为0) |

Rowkey：(endrowkeyobj1，endrowkeyobj2…)

## \_\_all\_all\_table

\_\_all\_all\_table其实是\_\_first\_tablet\_entry的一个视图(view)，由于视图尚未实现，所以目前\_\_all\_all\_table暂时还不可用。

## \_\_all\_all\_index

\_\_all\_all\_index表存储了所有的索引：

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| index\_name | vchar(??????) | 索引名 |
| index\_id | Int | 索引id |
| index\_column\_id | Int | on\_table\_column\_id在index中的序号，从1开始 |
| on\_table\_name | vchar(??????) | 哪个表的索引 |
| on\_table\_id | Int | 表的table\_id |
| on\_table\_column\_id | Int | 被索引的列 |
| on\_table\_column\_name | vchar(??????) | 被索引的列 |

Rowkey：(on\_table\_id，on\_table\_column\_id，index\_id，index\_column\_id)

## \_\_all\_all\_column

\_\_all\_all\_column存储了每个表的所有列、column\_id、列的值类型，表名(table\_name)和table\_id等，包括内部表(包括自身)和用户定义表。

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| table\_name | vchar(??????) | 表名 |
| table\_id | int | 表的table\_id |
| column\_name | vchar(??????) | 列名 |
| column\_id | int | column\_id |
| column\_group\_id | int | 列隶属的column group id |
| rowkey\_id | int | 0：非rowkey；正整数：rowkey的序号，必须是从1开始的连续正整数(\_\_all\_table\_table中的rowkey\_column\_num定义了该表的rowkey的列数量) |
| join\_table\_id | int | 0：没有join；正整数：该列需从join\_table\_id表的join\_column\_id列获得值 |
| join\_column\_id | int | 0：没有join；正整数：该列需从join\_table\_id表的join\_column\_id列获得值 |
| data\_type | int | 数据类型 |
| data\_length | int | 整数的字节数或字符串的最大长度 |
| data\_precision | int | 整数的十进制位数或decimal的有效位数(小数点前和小数点后) |
| data\_scale | int | decimal小数点后的位数 |
| nullable | int | 1：不可以为空；2：可以为空 |

Rowkey：(table\_id，column\_name)

\_\_all\_all\_column以及后面的\_\_all\_join\_info共同定义了各个表的schema信息，包括列join的对应关系(如收藏夹的item\_title$item\_title，item\_picurl$item\_picurl等关系)。

## \_\_all\_join\_info

\_\_all\_join\_info存储了表之间的内部join关系，即表left通过其某些列对应到表right的rowkey：

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| left\_table\_name | vchar(??????) | 表名 |
| left\_table\_id | int | 表的table\_id |
| left\_column\_name | vchar(??????) | 列名 |
| left\_column\_id | int | column\_id |
| right\_table\_name | vchar(??????) | 表名 |
| right\_table\_id | int | 表的table\_id |
| right\_column\_name | vchar(??????) | 列名 |
| right\_column\_id | int | column\_id |

rowkey：(left\_table\_id，left\_column\_id，right\_table\_id，right\_column\_id)

说明：

* left表及right表的对应列的类型必须一致；

## \_\_all\_data\_type

\_\_all\_data\_type存储了系统的所有数据类型

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| Name | vchar(??????) | 类型名称 |
| Id | int | 类型ID |
| Info | vchar(??????) | 对该类型的说明 |

Rowkey：id

## \_\_all\_sys\_param

\_\_all\_sys\_param存储了系统所需的诸多参数，不同的参数保存在不同行。

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| Name | vchar(??????) | 名称 |
| cluster\_role | Int | 0：与cluster role(主或备)无关；1: slave；2：master； |
| cluster\_id | Int | 0：与特定机群无关；正整数：指定机群； |
| server\_type | Int | 0：与特定服务器类型无关：1：RootServer，2：ChunkServer；3：MergeServer；4：UpdateServer |
| server\_role | Int | 0：与server role(RS或UPS的主或备)角色无关；1: slave；2：master； |
| server\_ipv4 | Int | server的IPv4地址(不使用时为0) |
| server\_ipv6\_high | Int | server的IPv6地址的高64位(不使用时为0) |
| server\_ipv6\_low | Int | server的IPv6地址的高64位(不使用时为0) |
| server\_ip\_port | Int | server的IP地址端口：为0表示与特定服务器无关 |
| data\_type | Int | 值的类型 |
| value | vchar(??????) | 数据类型由data\_type定义 |
| info | vchar(??????) | 对该项的说明 |

Rowkey：(name，cluster\_role，cluster\_id，server\_type，server\_role，server\_ipv4，server\_ipv6\_high，server\_ipv6\_low，server\_ip\_port)

举例：

设置主机群的主UPS的读流量(read\_percent)为10%，则

rowkey = (read\_percent，2，0，4，2，0，0，0，0)，value=10%；

设置备机群的读流量为60%，则

rowkey = (read\_percent，1，0，0，0，0，0，0，0)，value=60%；

设定所有CS的block\_cache\_size为5GB：

rowkey = (block\_cache\_size，0，0，0，0，0，0，0，0)，value=5GB；

部分已经定义的项：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name |  |  | type | info |
| app\_name |  |  | vchar | 应用的名称 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

完整的参数定义，

## \_\_all\_sys\_stat

\_\_all\_sys\_data存储了系统各种状态值，不同的项保存在不同行。

表结构：

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| Name | vchar(??????) | 名称 |
| cluster\_role | Int | 0：与cluster role(主或备)无关；1: slave；2：master； |
| cluster\_id | Int | 0：与特定机群无关；正整数：指定机群； |
| server\_type | Int | 0：与特定服务器类型无关：1：RootServer，2：ChunkServer；3：MergeServer；4：UpdateServer |
| server\_role | Int | 0：与server role(RS或UPS的主或备)角色无关；1: slave；2：master； |
| server\_ipv4 | Int | server的IPv4地址(不使用时为0) |
| server\_ipv6\_high | Int | server的IPv6地址的高64位(不使用时为0) |
| server\_ipv6\_low | Int | server的IPv6地址的高64位(不使用时为0) |
| server\_ip\_port | Int | server的IP地址端口：为0表示与特定服务器无关 |
| table\_id | Int | Table的ID：为0表示与特定table无关 |
| data\_type | Int | 值的类型 |
| value1 | Int | Int类型的数据 |
| value2 | vchar(??????) | Varchar类型的数据 |
| info | vchar(??????) | 对该值的说明 |

Rowkey：(cluster\_role，cluster\_id，server\_type，server\_role，server\_ipv4，server\_ipv6\_high，server\_ipv6\_low，server\_ip\_port，table\_id，name)

部分已经定义的项：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | cluster\_id | server\_id | data\_type | info |
| max\_used\_table\_id | 0 | 0 | int | 已经使用的最大table\_id |
|  |  |  |  |  |
|  |  |  |  |  |

## \_\_all\_cluster

\_\_all\_cluster记录了系统中所有的机群，这个表由每个机群的主RootServer更新。

表结构：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 说明 |
| cluster\_id | int | 正整数，机群ID |
| cluster\_role | int | 1: slave；2：master； |
| cluster\_name | vchar(??????) | 机群名称 |
| cluster\_info | vchar(??????) | 机群说明信息 |

Rowkey：(cluster\_id)

## \_\_all\_server

\_\_all\_server记录了系统中所有的服务器，这个表仅仅由主机群的主RootServer更新。

表结构：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 说明 |
| cluster\_id | int | 0：与特定机群无关；正整数：指定机群； |
| server\_type | int | 0：与特定服务器角色无关：1：RootServer，2：ChunkServer；3：MergeServer；4：UpdateServer |
| server\_role | int | 0：与server role(RS或UPS的主或备)角色无关；1: slave；2：master； |
| server\_ipv4 | int | server的IPv4地址(不使用时为0) |
| server\_ipv6\_high | int | server的IPv6地址的高64位(不使用时为0) |
| server\_ipv6\_low | int | server的IPv6地址的高64位(不使用时为0) |
| server\_ip\_port | int | server的IP地址端口：为0表示与特定服务器无关 |

Rowkey：(cluster\_id，server\_ipv4，server\_ipv6\_high，server\_ipv6\_low，server\_ip\_port)