

OCEANBASE



OceanBase 数据库

OceanBase 介绍

| 产品版本：V4.0.0

| 文档版本：20230505

声明

蚂蚁集团版权所有©2020，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击 设置> 网络> 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1	OceanBase 概述	6
1.1	核心特性	6
1.1.1	高可用	6
1.1.2	高兼容	6
1.1.3	水平扩展	6
1.1.4	低成本	6
1.1.5	实时 HTAP	6
1.1.6	安全可靠	6
1.1.7	完整自主知识产权	7
1.1.8	国产化适配	7
1.2	深入了解 OceanBase 数据库	7
2	系统架构	8
2.1	存储层	9
2.2	复制层	9
2.3	均衡层	10
2.4	事务层	10
2.4.1	原子性	10
2.4.2	隔离性	11
2.5	SQL 层	11
2.5.1	SQL 层组件	11
2.5.2	多种计划	12
2.6	接入层	12
2.7	更多信息	12
3	与 Oracle 兼容性概述	13
3.1	SQL 数据类型	13
3.2	内建函数	13
3.3	SQL 语法	14
3.4	过程性语言	15
3.5	系统视图	16
3.6	字符集和字符序	16
3.7	数据库对象管理	16
3.8	安全特性	18
3.9	备份恢复	19
3.10	高可用	19
3.11	SQL 引擎	19
3.12	暂不支持的功能	20
4	SQL 数据类型	21

5	内建函数	23
5.1	数字函数	24
5.2	字符串函数	25
5.3	日期函数	26
5.4	通用比较函数	27
5.5	转换函数	28
5.6	编码解码函数	30
5.7	环境和标识符函数	30
5.8	层次函数	30
5.9	空值相关的函数	31
5.10	聚合函数	32
5.11	分析函数	34
6	系统视图	35
6.1	字典视图	35
6.2	性能视图	41
7	与 MySQL 兼容性对比	43
7.1	数据类型	43
7.2	SQL 语法	44
7.3	系统视图	45
7.4	字符集和字符序	45
7.5	函数	45
7.6	分区支持	46
7.7	备份恢复	46
7.8	存储引擎	46
7.9	优化器	46
7.10	暂不支持的功能	48
8	使用限制	49
8.1	标识符长度限制	49
8.2	ODP 最大连接数限制	50
8.3	分区副本数限制	50
8.4	单个表的限制	50
8.5	单列的限制	51
8.6	字符串类型限制	51

1 OceanBase 概述

OceanBase 是一款完全自研，已连续 9 年稳定支撑双 11，创新推出“三地五中心”城市级容灾新标准，是全球唯一在 TPC-C 和 TPC-H 测试上都刷新了世界纪录的国产原生分布式数据库。产品采用自研的一体化架构，兼顾分布式架构的扩展性与集中式架构的性能优势，用一套引擎同时支持 TP 和 AP 的混合负载，具有数据强一致、高可用、高性能、在线扩展、高度兼容 SQL 标准和主流关系数据库、对应用透明，高性价比等特点，已助力 400+ 行业客户实现核心系统升级。

1.1 核心特性

1.1.1 高可用

独创“三地五中心”容灾架构方案，建立金融行业无损容灾新标准。支持同城/异地容灾，可实现多地多活，满足金融行业 6 级容灾标准（RPO=0，RTO<8 秒），数据零丢失。

1.1.2 高兼容

高度兼容 Oracle 和 MySQL，覆盖绝大多数常见功能，支持过程语言、触发器等高级特性，提供自动迁移工具，支持迁移评估和反向同步以保障数据迁移安全，可支撑金融、政府、运营商等关键行业核心场景替代。

1.1.3 水平扩展

实现透明水平扩展，支持业务快速的扩容缩容，同时通过准内存处理架构实现高性能。支持集群节点超过数千个，单集群最大数据量超过 3PB，最大单表行数达万亿级。

1.1.4 低成本

基于 LSM-Tree 的高压缩引擎，存储成本降低 70% - 90%；原生支持多租户架构，同集群可为多个独立业务提供服务，租户间数据隔离，降低部署和运维成本。

1.1.5 实时 HTAP

基于“同一份数据，同一个引擎”，同时支持在线实时交易及实时分析两种场景，“一份数据”的多个副本可以存储成多种形态，用于不同工作负载，从根本上保持数据一致性。

1.1.6 安全可靠

12 年完全自主研发，代码级可控，自研单机分布式一体化架构，大规模金融核心场景 9 年可靠性验证；完备的角色权限管理体系，数据存储和通信全链路透明加密，支持国密算法，通过等保三级专项合规检测。

1.1.7 完整自主知识产权

OceanBase 数据库由蚂蚁集团完全自主研发，不基于 MySQL 或者 PostgreSQL 等开源数据库，能够做到完全自主可控，不会存在基于开源数据库产品的技术限制问题。

1.1.8 国产化适配

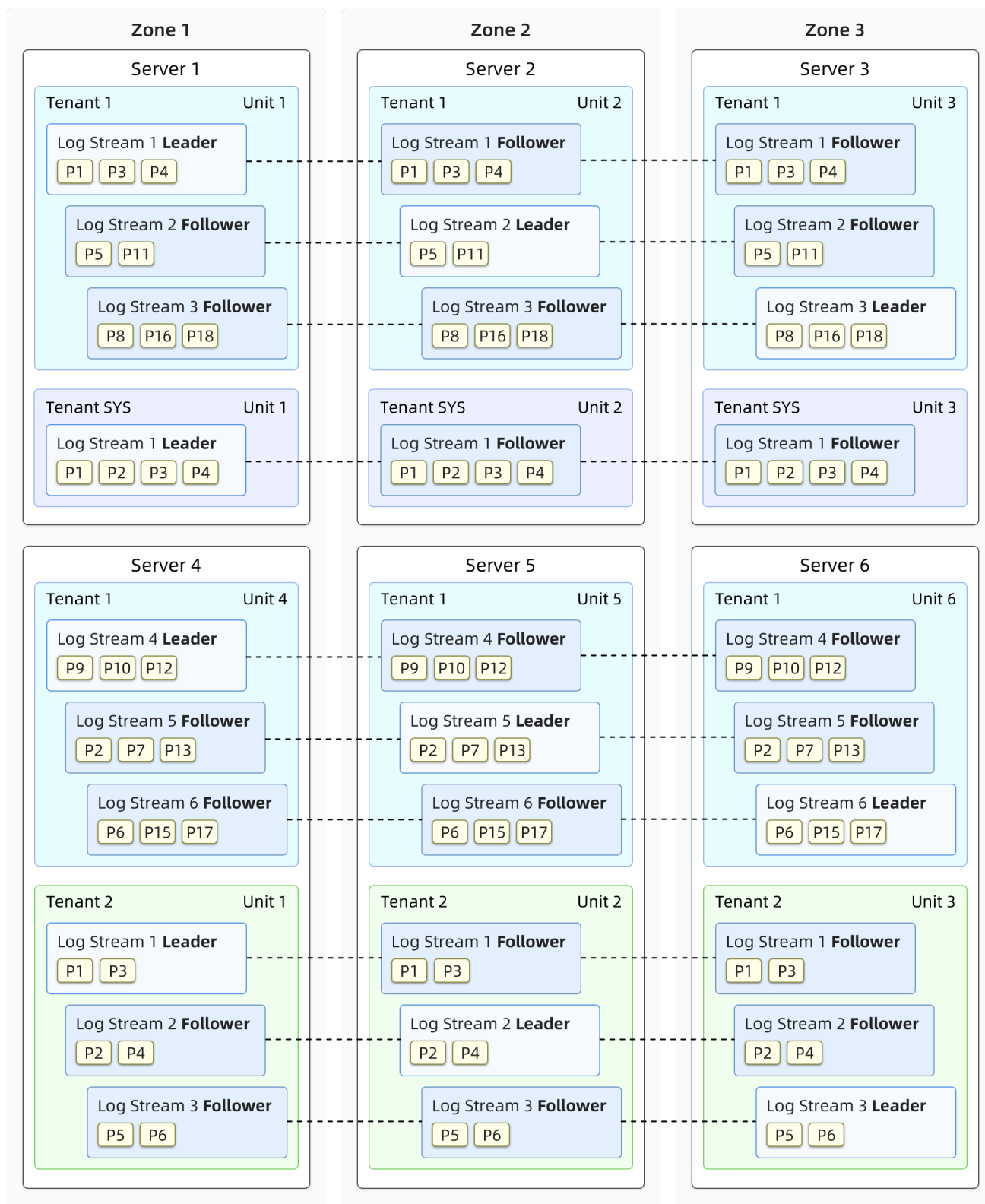
OceanBase 数据库支持全栈国产化解决方案。迄今已基于硬件整机中科可控 H620 系列、华为 TaiShan 200 系列、长城擎天 DF720 等整机，完成与海光 7185/7280、鲲鹏 920、飞腾 2000+ 等 CPU 的适配互认工作。同时，OceanBase 数据库还支持麒麟 V4、V10 和 UOS V20 等国产操作系统，并与上层中间件东方通 TongWeb V7.0、金蝶 Apusic 应用服务器软件 V9.0 等完成适配互认工作。

1.2 深入了解 OceanBase 数据库

您可以通过以下内容更深入地了解 OceanBase 数据库：

- [系统架构](#)
- [OceanBase 数据库系统概念](#)

2 系统架构



OceanBase 使用通用服务器硬件，依赖本地存储，分布式部署使用的多个服务器也是对等的，没有特殊的硬件要求。OceanBase 的分布式数据库处理采用 Shared Nothing 架构，数据库内的 SQL 执行引擎具有分布式执行能力。

OceanBase 在服务器上会运行叫做 observer 的单进程序作为数据库的运行实例，使用本地的文件存储数据和事务 Redo 日志。

OceanBase 集群部署需要配置可用区（Zone），由若干个服务器组成。可用区是一个逻辑概念，表示集群内具有相似硬件可用性的一组节点，它在不同的部署模式下代表不同的含义。例如，当整个集群部署在同一个数据中心（IDC）内的时候，一个可用区的节点可以属于同一个机架，同一个交换机等。当集群分布在多个数据中心的时候，每个可用区可以对应于一个数据中心。

用户存储的数据在分布式集群内部可以存储多个副本，用于故障容灾，也可以用于分散读取压力。在一个可用区内部数据只有一个副本，不同的可用区可以存储同一个数据的多个副本，副本之间由共识协议保证数据的一致性。

OceanBase 内置多租户特性，每个租户对于使用者是一个独立的数据库，一个租户能够在租户级别设置租户的分布式部署方式。租户之间 CPU、内存和 IO 都是隔离的。

OceanBase 的数据库实例内部由不同的组件相互协作，这些组件从底层向上由存储层、复制层、均衡层、事务层、SQL 层、接入层组成。

2.1 存储层

存储层以一张表或者一个分区为粒度提供数据存储与访问，每个分区对应一个用于存储数据的 Tablet（分片），用户定义的非分区表也会对应一个 Tablet。

Tablet 的内部是分层存储的结构，总共有 4 层。DML 操作插入、更新、删除等首先写入 MemTable，等到 MemTable 达到一定大小时转储到磁盘成为 L0 SSTable。L0 SSTable 个数达到阈值后会将多个 L0 SSTable 合并成一个 L1 SSTable。在每天配置的业务低峰期，系统会将所有的 MemTable、L0 SSTable 和 L1 SSTable 合并成一个 Major SSTable。

每个 SSTable 内部是以 2MB 定长宏块为基本单位，每个宏块内部由多个不定长微块组成。

Major SSTable 的微块会在合并过程中用编码方式进行格式转换，微块内的数据会按照列维度分别进行列内的编码，编码规则包括字典/游程/常量/差值等，每一列压缩结束后，还会进一步对多列进行列间等值/子串等规则编码。编码能对数据大幅压缩，同时提炼的列内特征信息还能进一步加速后续的查询速度。

在编码压缩之后，还可以根据用户指定的通用压缩算法进行无损压缩，进一步提升数据压缩率。

2.2 复制层

复制层使用日志流（LS、Log Stream）在多副本之间同步状态。每个 Tablet 都会对应一个确定的日志流，DML 操作写入 Tablet 的数据所产生的 Redo 日志会持久化在日志流中。日志流的多个副本会分布在不同的可用区中，多个副本之间维持了共识算法，选择其中一个副本作为主副本，其他的副本皆为从副本。Tablet 的 DML 和强一致性查询只在其对应的日志流的主副本上进行。

通常情况下，每个租户在每台机器上只会有一个日志流的主副本，可能存在多个其他日志流的从副本。租户的总日志流个数取决于 primary_zone 和 locality 的配置。

日志流使用自研的 Paxos 协议将 Redo 日志在本服务器持久化，同时通过网络发送给日志流的从副本，从副本在完成各自持久化后应答主副本，主副本在确认有多数派副本都持久化成功后确认对应的 Redo 日志持久化成功。从副本利用 Redo 日志的内容实时回放，保证自己的状态与主副本一致。

日志流的主副本在被选举成为主后会获得租约（Lease），正常工作的主副本在租约有效期内会不停的通过选举协议延长期租约。主副本只会在租约有效时执行主的工作，租约机制保证了数据库异常处理的能力。

复制层能够自动应对服务器故障，保障数据库服务的持续可用。如果出现少于半数的从副本所在服务器出现问题，因为还有多于半数的副本正常工作，数据库的服务不受影响。如果主副本所在服务器出现问题，其租约会得不到延续，待其租约失效后，其他从副本会通过选举协议选举出新的主副本并授予新的租约，之后即可恢复数据库的服务。

2.3 均衡层

新建表和新增分区时，系统会按照均衡原则选择合适的日志流创建 Tablet。当租户的属性发生变更，新增了机器资源，或者经过长时间使用后，Tablet 在各台机器上不再均衡时，均衡层通过日志流的分裂和合并操作，并在这个过程中配合日志流副本的移动，让数据和服务在多个服务器之间再次均衡。

当租户有扩容操作，获得更多服务器资源时，均衡层会将租户内已有的日志流进行分裂，并选择合适数量的 Tablet 一同分裂到新的日志流中，再将新日志流迁移到新增的服务器上，以充分利用扩容后的资源。当租户有缩容操作时，均衡层会把需要缩减的服务器上的日志流迁移到其他服务器上，并和其他服务器上已有的日志流进行合并，以缩减机器的资源占用。

当数据库长期使用后，随着持续创建删除表格，并且写入更多的数据，即使没有服务器资源数量变化，原本均衡的情况可能被破坏。最常见的情况是，当用户删除了一批表格后，删除的表格可能原本聚集在某一些机器上，删除后这些机器上的 Tablet 数量就变少了，应该把其他机器的 Tablet 均衡一些到这些少的机器上。均衡层会定期生成均衡计划，将 Tablet 多的服务器上日志流分裂出临时日志流并携带需要移动的 Tablet，临时日志流迁移到目的服务器后再和目的服务器上的日志流进行合并，以达成均衡的效果。

2.4 事务层

事务层保证了单个日志流和多个日志流DML操作提交的原子性，也保证了并发事务之间的多版本隔离能力。

2.4.1 原子性

一个日志流上事务的修改，即使涉及多个 Tablet，通过日志流的 write-ahead log 可以保证事务提交的原子性。事务的修改涉及多个日志流时，每个日志流会产生并持久化各自的write-ahead log，事务层通过优化的两阶段提交协议来保证提交的原子性。

事务层会选择一个事务修改的一个日志流产生协调者状态机，协调者会与事务修改的所有日志流通信，判断 write-ahead log 是否持久化，当所有日志流都完成持久化后，事务进入提交

状态，协调者会再驱动所有日志流写下这个事务的 `Commit` 日志，表示事务最终的提交状态。当从副本回放或者数据库重启时，已经完成提交的事务都会通过 `Commit` 日志确定各自日志流事务的状态。

宕机重启场景下，宕机前还未完成的事务，会出现写完 `write-ahead log` 但是还没有 `Commit` 日志的情况，每个日志流的 `write-ahead log` 都会包含事务的所有日志流列表，通过此信息可以重新确定哪个日志流是协调者并恢复协调者的状态，再次推进两阶段状态机，直到事务最终的 `Commit` 或 `Abort` 状态。

2.4.2 隔离性

GTS 服务是一个租户内产生连续增长的时间戳的服务，其通过多副本保证可用性，底层机制与上面复制层所描述的日志流副本同步机制是一样的。

每个事务在提交时会从 GTS 获取一个时间戳作为事务的提交版本号并持久化在日志流的 `write-ahead log` 中，事务内所有修改的数据都以此提交版本号标记。

每个语句开始时（对于 `Read Committed` 隔离级别）或者每个事务开始时（对于 `Repeatable Read` 和 `Serializable` 隔离级别）会从 GTS 获取一个时间戳作为语句或事务的读取版本号。在读取数据时，会跳过事务版本号比读取版本号大的数据，通过这种方式为读取操作提供了统一的全局数据快照。

2.5 SQL 层

SQL 层将用户的 SQL 请求转化成对一个或多个 Tablet 的数据访问。

2.5.3 SQL 层组件

SQL 层处理一个请求的执行流程是：Parser、Resolver、Transformer、Optimizer、Code Generator、Executor。

Parser 负责词法/语法解析，Parser 会将用户的 SQL 分成一个个的 "Token"，并根据预先设定好的语法规则解析整个请求，转换成语法树（Syntax Tree）。

Resolver 负责语义解析，将根据数据库元信息将 SQL 请求中的 Token 翻译成对应的对象（例如库、表、列、索引等），生成的数据结构叫做 Statement Tree。

Transformer 负责逻辑改写，根据内部的规则或代价模型，将 SQL 改写为与之等价的其他形式，并将其提供给后续的优化器做进一步的优化。Transformer 的工作方式是在原 Statement Tree 上做等价变换，变换的结果仍然是一棵 Statement Tree。

Optimizer（优化器）为 SQL 请求生成最佳的执行计划，需要综合考虑 SQL 请求的语义、对象数据特征、对象物理分布等多方面因素，解决访问路径选择、联接顺序选择、联接算法选择、分布式计划生成等问题，最终生成执行计划。

Code Generator（代码生成器）将执行计划转换为可执行的代码，但是不做任何优化选择。

Executor（执行器）启动 SQL 的执行过程。

在标准的 SQL 流程之外，SQL 层还有 Plan Cache 能力，将历史的执行计划缓存在内存中，后续的执行可以反复执行这个计划，避免了重复查询优化的过程。配合 Fast-parser 模块，仅使用词法分析对文本串直接参数化，获取参数化后的文本及常量参数，让 SQL 直接命中 Plan Cache，加速频繁执行的 SQL。

2.5.4 多种计划

SQL 层的执行计划分为本地、远程和分布式三种。本地执行计划只访问本服务器的数据。远程执行计划只访问非本地的一台服务器的数据。分布式计划会访问超过一台服务器的数据，执行计划会分成多个子计划在多个服务器上执行。

SQL 层并行化执行能力可以将执行计划分解成多个部分，由多个执行线程执行，通过一定的调度的方式，实现执行计划的并行处理。并行化执行可以充分发挥服务器 CPU 和 IO 处理能力，缩短单个查询的响应时间。并行查询技术可以用于分布式执行计划，也可以用于本地执行计划。

2.6 接入层

obproxy 是 OceanBase 数据库的接入层，负责将用户的请求转发到合适的 OceanBase 实例上进行处理。

obproxy 是独立的进程实例，独立于 OceanBase 的数据库实例部署。obproxy 监听网络端口，兼容 MySQL 网络协议，支持使用 MySQL 驱动的应用直接连接 OceanBase。

obproxy 能够自动发现 OceanBase 集群的数据分布信息，对于代理的每一条 SQL 语句，会尽可能识别出语句将访问的数据，并将语句直接转发到数据所在服务器的 OceanBase 实例。

obproxy 有两种部署方式，一种是部署在每一个需要访问数据库的应用服务器上，另一种是部署在与 OceanBase 相同的机器上。第一种部署方式下，应用程序直接连接部署在同一台服务器上的 obproxy，所有的请求会由 obproxy 发送到合适的 OceanBase 服务器。第二种部署方式下，需要使用网络负载均衡服务将多个 obproxy 聚合成同一个对应用提供服务的入口地址。

2.7 更多信息

更多 OceanBase 数据库整体架构的信息，查看以下内容：

- [OceanBase 数据库整体架构](#)

3 与 Oracle 兼容性概述

本节主要介绍 OceanBase 数据库的 Oracle 模式与原生 Oracle 数据库的兼容性对比信息。

OceanBase 数据库在数据类型、SQL 功能和数据库对象等基本功能上与 Oracle 数据库兼容。在 PL 方面，已经基本能够兼容全部的研发功能。在数据库安全、备份恢复、高可用和优化器等高级特性上 OceanBase 数据库的兼容性也非常好，而且有些特性还要优于 Oracle 数据库。这意味着在从 Oracle 数据库迁移到 OceanBase 数据库的过程中，用户不需要消耗大量的时间去学习新知识，即可流畅地实现从 Oracle 数据库到 OceanBase 数据库的迁移。

此外，由于 OceanBase 数据库与 Oracle 数据库在底层架构、产品形态等方面的不同，有一部分功能 OceanBase 数据库暂时不会进行兼容或者会与 Oracle 数据库的表现有所差异。本节主要从以下几个方面介绍 OceanBase 数据库与 Oracle 数据库的兼容性对比信息：

- SQL 数据类型
- 内建函数
- SQL 语法
- 过程性语言
- 系统视图
- 字符集和字符序 Collation
- 数据库对象管理
- 安全特性
- 备份恢复
- 高可用
- SQL 引擎
- 暂不支持的功能

3.1 SQL 数据类型

Oracle 数据库中有 24 个数据类型，OceanBase 数据库目前支持 20 种，详细支持信息请参见 [SQL 数据类型](#)。基于优化考虑，`LONG` 和 `LONG RAW` 数据类型过于老旧，OceanBase 数据库暂不计划支持这两种数据类型。

说明

OceanBase 数据库中大对象数据类型有 48 MB 的大小限制且性能不佳，所以不建议在复杂场景下使用。详细信息请参见 [大对象数据类型](#)。

3.2 内建函数

Oracle 数据库中支持内建函数 257 个，OceanBase 数据库当前支持 155 个，详细支持信息请参见 [内建函数](#)。

3.3 SQL 语法

OceanBase 数据库支持 Oracle 数据库中绝大部分的 SQL 语法。

少数功能性缺失会报语法不支持的错误。

SELECT

- 支持大部分查询功能，包括支持单、多表查询；支持子查询；支持内连接，半连接，外连接；支持分组、聚合；支持层次查询；常见的概率，线性回归等数据挖掘函数等。
- 支持如下集合操作： `UNION`、`UNION ALL`、`INTERSECT`、`MINUS`。
- 支持如下语法查看执行计划：

```
EXPLAIN [explain_type] dml_statement;
```

```
explain_type:
```

```
BASIC
```

```
| OUTLINE
```

```
| EXTENDED
```

```
| EXTENDED_NOADDR
```

```
| PARTITIONS
```

```
| FORMAT = {TRADITIONAL|JSON}
```

```
dml_statement:
```

```
SELECT statement
```

```
| DELETE statement
```

```
| INSERT statement
```

```
| MERGE INTO statement
```

```
| UPDATE statement
```

INSERT

- 支持单行、多行插入，同时支持指定分区插入
- 支持 `INSERT INTO ... SELECT ...` 语句
- 支持单表和多表插入

UPDATE

- 支持单列和多列的更新
- 支持使用子查询
- 支持集合更新

DELETE

- 支持单表和多表的删除

TRUNCATE

- 支持完全清空指定表

并行查询

- 支持类 Oracle 数据库的并行查询

OceanBase 数据库中 DOP 需要手动通过 `hint/session` 变量指定，暂不支持 Auto DOP 功能。

- 支持并行 DML

Hint

OceanBase 数据库支持使用 Hint。Oracle 数据库中有 72 个 Hint，目前 OceanBase 数据库兼容 24 个。另外，OceanBase 数据库特有的 Hint 有 23 个。

关于 Hint 的详细说明，请参见 [Hint](#)

3.4 过程性语言

OceanBase 数据库兼容了大部分 Oracle 数据库的 PL 功能，主要支持的 PL 功能如下：

- 数据类型
- 流程控制
- 集合与记录（暂不支持多维度集合）。
- 静态 SQL
- 动态 SQL
- 子过程
- 触发器
- 异常处理
- 程序包
- 性能优化
- 自定义数据类型

- PL 系统包，包括 DBMS_CRYPTO、DBMS_DEBUG、DBMS_LOB、DBMS_LOCK、DBMS_METADATA、DBMS_OUTPUT、DBMS_RANDOM、DBMS_SQL、DBMS_XA、UTL_I18N、UTL_RAW 等。
- PL 标签安全包，包括 SA_SYSDBA、SA_COMPONENTS、SA_LABEL_ADMIN、SA_POLICY_ADMIN、SA_USER_ADMIN、SA_SESSION 等。

OceanBase 数据库暂不支持如下 PL 功能：

- 条件编译

更多 PL 功能的详细信息请参见 [PL 参考](#)。

3.5 系统视图

OceanBase 数据库兼容了部分 Oracle 数据库的系统视图，其中：

- 兼容字典视图 195 个。
- 兼容性能视图（以 v\$ 开头）20 个。

关于兼容的详细视图列表请参见 [系统视图](#)。

更多系统视图的字段说明信息请参考《参考指南（Oracle 模式）》文档中 [系统视图](#) 章节。

3.6 字符集和字符序

- OceanBase 数据库支持的字符集有 binary、utf8mb4、gbk、utf16、gb18030。
- OceanBase 数据库支持的排序方式有 binary、utf8mb4_general_ci、utf8mb4_bin、gbk_chinese_ci、gbk_bin、utf16_general_ci、utf16_bin、utf8mb4_unicode_ci、utf16_unicode_ci、gb18030_chinese_ci 和 gb18030_bin。

3.7 数据库对象管理

表管理

- 创建表：支持创建表，建表时可以指定分区、约束等信息。
- 修改基表：支持通过 `ALTER TABLE` 语句添加、删除、修改列；添加、删除约束；添加、删除、修改分区。
- 删除基表：支持删除表，并级联约束。

关于创建、修改、删除表的详细语法，请参见 [CREATE TABLE](#)、[ALTER TABLE](#)、[DROP TABLE](#)。

约束

- 支持 `CHECK`、`UNIQUE` 和 `NOT NULL` 约束。
- 不支持 `UNIQUE` 约束的 `DISABLE` 操作。

- 支持外键。
- 不支持添加外键约束的 `DISABLE` 和 `ENABLE`。
- 支持使用 `ALTER TABLE` 语句添加外键约束。
- 不支持级联中的 `SET NULL`。

分区支持

- 支持一级分区、模板化和非模板化的二级分区。
- 支持哈希（Hash）、范围（Range）、列表（List）和组合分区等分区形式。
- 支持局部索引和全局索引
- 对于分区维护操作：
 - 一级分区表支持添加一级分区、删除一级分区、Truncate 一级分区
 - 模板化二级分区表支持添加一级分区、删除一级分区；非模板化二级分区表支持添加一级分区、删除一级分区、Truncate 一级分区、添加二级分区、删除二级分区、Truncate 二级分区
- 模板化二级分区表暂不支持添加二级分区、删除二级分区

更多分区的说明及使用请参见《管理员指南》文档中的 [分区概述](#) 章节。

索引管理

- OceanBase 数据库仅支持 BTree 索引
- 支持创建和删除索引
- 不支持位图和反向等索引类型

视图管理

- 支持创建简单或复杂视图
- 支持删除视图
- 支持 `SELECT` 语句
- 支持 DML 语句

可更新视图

不支持 `WITH CHECK OPTION` 子句。

序列管理

支持创建、修改、删除序列，还支持序列的重置取值功能。

同义词

支持对表、视图、同义词和序列等对象创建同义词，并且支持创建公共同义词。

触发器管理

支持创建、修改、删除触发器。

数据库链接

OceanBase 数据库目前支持 OceanBase 数据库到 OceanBase 数据库的只读操作。

3.8 安全特性

OceanBase 数据库实现了丰富的安全特性。

权限管理

- 兼容 Oracle 数据库的 24 个系统权限
- 支持常见的对象权限管理，例如，授权与移除权限
- 支持白名单策略，实现网络安全访问控制
- 支持系统预定义角色，用户自定义角色

身份鉴别

- Oracle 模式兼容 Oracle 数据库的密码策略
- 支持用户的锁定和解锁功能

透明加密

兼容 Oracle 数据库的 TDE 功能，数据在写入存储设备之前自动进行加密，读取时自动解密。

审计

- 支持 Oracle 数据库的标准审计，暂不支持统一审计和 FGA 细粒度审计
- 支持语句和对象审计类型，暂不支持权限审计和网络审计
- 对象审计当前仅支持表、序列、函数和包对象
- 审计结果支持存放在文件或者内部表中
- 支持审计相关的各类系统视图
- 暂不支持删除审计数据

标签安全

- 兼容 Oracle 数据库中的标签安全（Label Security）功能
- 支持 Level 维度，暂不支持 Compartment 和 Group 维度

传输链路加密 SSL

- 支持客户端与 OceanBase 数据库服务器的传输链路加密，以及 OceanBase 数据库节点之间的数据传输加密。
- 除了支持 SSL 单向认证、X509 双向认证，还支持指定加密算法、指定发行方认证、指定 SSL 主题认证等特殊的双向认证。

3.9 备份恢复

OceanBase 数据库也提供了备份恢复功能，具体支持的特性如下：

- 支持阿里云 OSS、NFS 作为备份目的地
- 支持日志归档压缩
- 支持自动清理过期备份
- 支持租户级别的备份与恢复

不支持的特性如下：

- 不支持集群级别的备份与恢复
- 不支持指定路径的备份
- 不支持手动清理备份
- 不支持备份的备份
- 不支持备份的数据校验
- 不支持租户内部部分数据库和表级别的备份恢复

更多 OceanBase 数据库物理备份恢复特性的介绍请参见《OceanBase 数据库系统架构》文档中的 [备份恢复概述](#) 章节。

3.10 高可用

OceanBase 数据库的高可用主要采用多副本来实现，同时，也支持如下特性：

- 为了数据安全性，支持物理备份，类似 Oracle 数据库的 RMAN 特性。

3.11 SQL 引擎

OceanBase 数据库的 SQL 引擎兼容了 Oracle 数据库大部分的特性：

- 支持查询改写
- 支持预编译语句
- 支持基于成本的优化器
- 支持执行计划生成与展示（`EXPLAIN`）
- 支持执行计划缓存
- 支持执行计划快速参数化
- 支持执行计划绑定
- 支持 Optimizer Hint
- 支持自适应游标共享 ACS

- 支持执行计划管理 SPM

OceanBase 数据库的 SQL 引擎暂不支持以下功能：

- 估算器
- 执行计划隔离
- 表达式统计存储（ESS）
- 近似查询处理

更多 SQL 引擎的详细信息请参见 [SQL 调优指南](#)。

3.12 暂不支持的功能

- 不支持 `LONG` 和 `LONG RAW` 数据类型。
- PL 中暂不支持条件编译。
- 数据库约束中，不支持 `UNIQUE` 约束的 `DISABLE` 操作；不支持添加外键约束的 `DISABLE` 和 `ENABLE`；不支持级联中的 `SET NULL`。
- 不支持位图和反向索引类型。
- 不支持 `WITH CHECK OPTION` 子句。
- 暂不支持删除审计相关的各类视图。
- 对于备份恢复功能，不支持集群级别的备份与恢复，不支持指定路径的备份、不支持手动清理备份，不支持备份的备份，不支持备份的数据校验，不支持租户内部部分数据库和表级别的备份恢复。
- 不支持主备库功能。
- 对于 SQL 引擎，暂不支持估算器、执行计划隔离、表达式统计存储（ESS）和近似查询处理功能。

4 SQL 数据类型

本节主要介绍 OceanBase 数据库的 Oracle 模式与原生 Oracle 数据库中 SQL 数据类型的详细兼容对比信息。

Oracle 数据库中有 24 种数据类型，OceanBase 数据库目前支持 20 种，详细信息如下表所示。

序号	Oracle 数据库	OceanBase 数据库
----	------------	---------------

1	CHAR (size)	支持
2	NCHAR[(size)]	支持
3	VARCHAR2(size)	支持
4	VARCHAR(size)	支持
5	NVARCHAR2(size)	支持
6	NUMBER [(p [, s])]	支持
7	FLOAT [(p)]	支持
8	BINARY_FLOAT	支持
9	BINARY_DOUBLE	支持
10	LONG (Oracle 数据库已废弃类型)	不支持
11	DATE	支持
12	TIMESTAMP [(fractional_seconds_precision)]	支持
13	TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE	支持
14	TIMESTAMP [(fractional_seconds_precision)] WITH LOCAL TIME ZONE	支持
15	INTERVAL YEAR [(year_precision)] TO MONTH	支持
16	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]	支持
17	RAW(size)	支持
18	LONG RAW (Oracle 数据库已废弃类型)	不支持
19	ROWID	支持
20	UROWID [(size)]	支持
21	BFILE	不支持
22	BLOB	支持
23	CLOB	支持
24	NCLOB	不支持

5 内建函数

本节主要介绍 OceanBase 数据库的 Oracle 模式与原生 Oracle 数据库中内建函数的详细兼容对比信息。

Oracle 数据库中支持内建函数 257 个，OceanBase 数据库当前支持 155 个，更多详细信息请参阅 [函数](#)。

5.1 数字函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	ABS	支持
2	ACOS	支持
3	ASIN	支持
4	ATAN	支持
5	ATAN2	支持
6	BITAND	支持
7	CEIL	支持
8	COS	支持
9	COSH	支持
10	EXP	支持
11	FLOOR	支持
12	LN	支持
13	LOG	支持
14	MOD	支持
15	NANVL	支持
16	POWER	支持
17	REMAINDER	支持
18	ROUND (number)	支持
19	SIGN	支持
20	SIN	支持
21	SINH	支持
22	SQRT	支持
23	TAN	支持
24	TANH	支持
25	TRUNC (number)	支持
26	WIDTH_BUCKET	支持

5.2 字符串函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	CHR	支持
2	CONCAT	支持
3	INITCAP	支持
4	LOWER	支持
5	LPAD	支持
6	LTRIM	支持
7	NLS_LOWER	支持
8	NLS_UPPER	支持
9	REGEXP_REPLACE	支持
10	REGEXP_SUBSTR	支持
11	REPLACE	支持
12	RPAD	支持
13	RTRIM	支持
14	SUBSTR	支持
15	TRANSLATE	支持
16	TRANSLATE ... USING	支持
17	TRIM	支持
18	UPPER	支持
19	ASCII	支持
20	INSTR	支持
21	LENGTH	支持
22	REGEXP_COUNT	支持
23	REGEXP_INSTR	支持

5.3 日期函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	ADD_MONTHS	支持
2	CURRENT_DATE	支持
3	CURRENT_TIMESTAMP	支持
4	DBTIMEZONE	支持
5	EXTRACT (datetime)	支持
6	FROM_TZ	支持
7	LAST_DAY	支持
8	LOCALTIMESTAMP	支持
9	MONTHS_BETWEEN	支持
10	NEXT_DAY	支持
11	NUMTODSINTERVAL	支持
12	NUMTOYMINTERVAL	支持
13	ROUND (date)	支持
14	SESSIONTIMEZONE	支持
15	SYS_EXTRACT_UTC	支持
16	SYSDATE	支持
17	SYSTIMESTAMP	支持
18	TO_CHAR (datetime)	支持
19	TO_DSINTERVAL	支持
20	TO_TIMESTAMP	支持
21	TO_TIMESTAMP_TZ	支持
22	TO_YMINTERVAL	支持
23	TRUNC (date)	支持
24	TZ_OFFSET	支持

5.4 通用比较函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	GREATEST	支持
2	LEAST	支持

5.5 转换函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	ASCIISTR	支持
2	CAST	支持
3	CHARTOROWID	支持
4	CONVERT	支持
5	HEXTORAW	支持
6	NUMTODSINTERVAL	支持
7	NUMTOYMINTERVAL	支持
8	RAWTOHEX	支持
9	ROWIDTOCHAR	支持
10	ROWIDTONCHAR	支持
11	TO_BINARY_DOUBLE	支持
12	TO_BINARY_FLOAT	支持
13	TO_BLOB	支持
14	TO_CHAR (character)	支持
15	TO_CHAR (datetime)	支持
16	TO_CHAR (number)	支持
17	TO_CLOB	支持
18	TO_DATE	支持
19	TO_DSINTERVAL	支持
20	TO_MULTI_BYTE	支持
21	TO_NUMBER	支持
22	TO_NCHAR (character)	支持
23	TO_NCHAR (datetime)	支持
24	TO_NCHAR (number)	支持
25	TO_SINGLE_BYTE	支持
26	TO_TIMESTAMP	支持
27	TO_TIMESTAMP_TZ	支持
28	TO_YMINTERVAL	支持
29	UNISTR	支持

5.6 编码解码函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	DECODE	支持
2	DUMP	支持
3	ORA_HASH	支持
4	VSIZE	支持

5.7 环境和标识符函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	SYS_CONTEXT	支持
2	SYS_GUID	支持
3	UID	支持
4	USER	支持

5.8 层次函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	SYS_CONNECT_BY_PATH	支持

5.9 空值相关的函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	COALESCE	支持
2	LNNVL	支持
3	NANVL	支持
4	NULLIF	支持
5	NVL	支持
6	NVL2	支持

5.10 聚合函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	APPROX_COUNT_DISTINCT	支持
2	AVG *	支持
3	CORR *	支持
4	COUNT *	支持
5	COVAR_POP *	支持
6	COVAR_SAMP *	支持
7	CUME_DIST *	支持
8	DENSE_RANK *	支持
9	GROUPING	支持
10	KEEP *	支持
11	LISTAGG *	支持
12	MAX *	支持
13	MIN *	支持
14	MEDIAN *	支持
15	PERCENT_RANK *	支持
16	PERCENTILE_CONT *	支持
17	PERCENTILE_DISC *	支持
18	RANK *	支持
19	REGR_ (Linear Regression) *	支持
20	ROLLUP	支持
21	STDDEV *	支持
22	STDDEV_POP *	支持
23	STDDEV_SAMP *	支持
24	SUM *	支持
25	VAR_POP *	支持
26	VAR_SAMP *	支持
27	VARIANCE *	支持
28	WMSYS.WM_CONCAT /WM_CONCAT *	支持

说明

* 表示该函数既是聚合函数又是分析函数。

5.11 分析函数

序号	Oracle 数据库的函数	OceanBase 数据库是否支持
1	FIRST_VALUE	支持
2	LAG	支持
3	LAST_VALUE	支持
4	LEAD	支持
5	NTH_VALUE	支持
6	NTILE	支持
7	RATIO_TO_REPORT	支持
8	ROW_NUMBER	支持

说明

聚合函数表格中标 * 函数也同时属于分析函数，故未在此表格重复列出。

6 系统视图

本节主要介绍 OceanBase 数据库与原生 Oracle 数据库中字典视图和性能视图的详细兼容列表。

6.1 字典视图

OceanBase 数据库兼容 Oracle 数据库的字典视图列表如下：

- [ALL_TAB_PARTITIONS](#)
- [ALL_ARGUMENTS](#)
- [ALL_TAB_SUBPARTITIONS](#)
- [ALL_COLL_TYPES](#)
- [ALL_COL_PRIVS](#)
- [ALL_CONSTRAINTS](#)
- [ALL_CONS_COLUMNS](#)
- [ALL_DEF_AUDIT_OPTS](#)
- [ALL_ERRORS](#)
- [ALL_INDEXES](#)
- [ALL_IND_COLUMNS](#)
- [ALL_IND_EXPRESSIONS](#)
- [ALL_IND_PARTITIONS](#)
- [ALL_IND_SUBPARTITIONS](#)
- [ALL_METHOD_PARAMS](#)
- [ALL_OBJECTS](#)
- [DBA_TAB_PARTITIONS](#)
- [ALL_PART_KEY_COLUMNS](#)
- [DBA_TAB_SUBPARTITIONS](#)
- [ALL_PROCEDURES](#)
- [ALL_SCHEDULER_JOB_ARGS](#)
- [ALL_SCHEDULER_PROGRAM_ARGS](#)
- [ALL_SEQUENCES](#)
- [ALL_SOURCE](#)
- [USER_TAB_SUBPARTITIONS](#)

- [ALL_SUBPART_KEY_COLUMNS](#)
- [ALL_SYNONYMS](#)
- [ALL_TABLES](#)
- [ALL_TAB_COLS](#)
- [ALL_TAB_COLUMNS](#)
- [USER_TAB_PARTITIONS](#)
- [ALL_ALL_TABLES](#)
- [ALL_TAB_PRIVS](#)
- [ALL_COL_COMMENTS](#)
- [ALL_TRIGGERS](#)
- [ALL_TYPES](#)
- [ALL_TYPE_ATTRS](#)
- [ALL_TYPE_METHODS](#)
- [ALL_USERS](#)
- [ALL_VIEWS](#)
- [ALL_DIRECTORIES](#)
- [ALL_PART_COL_STATISTICS](#)
- [ALL_PART_HISTOGRAMS](#)
- [ALL_SUBPART_COL_STATISTICS](#)
- [ALL_SUBPART_HISTOGRAMS](#)
- [ALL_TAB_COL_STATISTICS](#)
- [ALL_TAB_HISTOGRAMS](#)
- [ALL_TAB_STATISTICS](#)
- [AUDIT_ACTIONS](#)
- [ALL_PART_INDEXES](#)
- [ALL_DEPENDENCIES](#)
- [ALL_TAB_STATS_HISTORY](#)
- [DBA_TAB_STATS_HISTORY](#)
- [USER_TAB_STATS_HISTORY](#)
- [DBA_ARGUMENTS](#)
- [DBA_AUDIT_EXISTS](#)

- [DBA_AUDIT_OBJECT](#)
- [DBA_AUDIT_SESSION](#)
- [DBA_AUDIT_STATEMENT](#)
- [DBA_AUDIT_TRAIL](#)
- [DBA_COLL_TYPES](#)
- [ALL_PART_TABLES](#)
- [DBA_COL_PRIVS](#)
- [DBA_CONSTRAINTS](#)
- [DBA_CONS_COLUMNS](#)
- [DBA_ERRORS](#)
- [DBA_INDEXES](#)
- [DBA_IND_COLUMNS](#)
- [DBA_IND_EXPRESSIONS](#)
- [DBA_IND_PARTITIONS](#)
- [DBA_IND_SUBPARTITIONS](#)
- [DBA_METHOD_PARAMS](#)
- [DBA_OBJECTS](#)
- [DBA_OBJ_AUDIT_OPTS](#)
- [ALL_SUBPARTITION_TEMPLATES](#)
- [DBA_PART_KEY_COLUMNS](#)
- [ALL_TAB_COMMENTS](#)
- [DBA_PROCEDURES](#)
- [DBA_PROFILES](#)
- [DBA_ROLES](#)
- [DBA_ROLE_PRIVS](#)
- [DBA_SCHEDULER_JOB_ARGS](#)
- [DBA_SCHEDULER_PROGRAM_ARGS](#)
- [DBA_SEGMENTS](#)
- [DBA_SEQUENCES](#)
- [DBA_SOURCE](#)
- [DBA_STMT_AUDIT_OPTS](#)

- [DBA_ALL_TABLES](#)
- [DBA_SUBPART_KEY_COLUMNS](#)
- [DBA_SYNONYMS](#)
- [DBA_SYS_PRIVS](#)
- [DBA_TABLES](#)
- [DBA_TABLESPACES](#)
- [DBA_TAB_COLS](#)
- [DBA_TAB_COLUMNS](#)
- [DBA_COL_COMMENTS](#)
- [DBA_PART_INDEXES](#)
- [DBA_TAB_PRIVS](#)
- [DBA_PART_TABLES](#)
- [DBA_TRIGGERS](#)
- [DBA_TYPES](#)
- [DBA_TYPE_ATTRS](#)
- [DBA_TYPE_METHODS](#)
- [DBA_USERS](#)
- [DBA_VIEWS](#)
- [DBA_DEPENDENCIES](#)
- [NLS_DATABASE_PARAMETERS](#)
- [NLS_INSTANCE_PARAMETERS](#)
- [NLS_SESSION_PARAMETERS](#)
- [STMT_AUDIT_OPTION_MAP](#)
- [DBA_SUBPARTITION_TEMPLATES](#)
- [DBA_RECYCLEBIN](#)
- [DBA_RSRC_CONSUMER_GROUPS](#)
- [DBA_RSRC_GROUP_MAPPINGS](#)
- [DBA_RSRC_PLANS](#)
- [DBA_RSRC_PLAN_DIRECTIVES](#)
- [DBA_DIRECTORIES](#)
- [DBA_JOBS](#)

- [DBA_JOBS_RUNNING](#)
- [DBA_PART_COL_STATISTICS](#)
- [DBA_PART_HISTOGRAMS](#)
- [DBA_SUBPART_COL_STATISTICS](#)
- [DBA_SUBPART_HISTOGRAMS](#)
- [DBA_TAB_COL_STATISTICS](#)
- [DBA_TAB_HISTOGRAMS](#)
- [DBA_TAB_STATISTICS](#)
- [USER_ARGUMENTS](#)
- [USER_AUDIT_OBJECT](#)
- [USER_AUDIT_SESSION](#)
- [USER_AUDIT_STATEMENT](#)
- [USER_AUDIT_TRAIL](#)
- [USER_COLL_TYPES](#)
- [DBA_TAB_COMMENTS](#)
- [USER_COL_PRIVS](#)
- [USER_CONSTRAINTS](#)
- [USER_CONS_COLUMNS](#)
- [USER_ERRORS](#)
- [USER_INDEXES](#)
- [USER_IND_COLUMNS](#)
- [USER_IND_EXPRESSIONS](#)
- [USER_IND_PARTITIONS](#)
- [USER_IND_SUBPARTITIONS](#)
- [USER_OBJECTS](#)
- [USER_ALL_TABLES](#)
- [USER_PART_KEY_COLUMNS](#)
- [USER_COL_COMMENTS](#)
- [USER_PROCEDURES](#)
- [USER_ROLE_PRIVS](#)
- [USER_SCHEDULER_JOB_ARGS](#)

- [USER_SCHEDULER_PROGRAM_ARGS](#)
- [USER_SEGMENTS](#)
- [USER_SEQUENCES](#)
- [USER_SOURCE](#)
- [USER_PART_INDEXES](#)
- [USER_SUBPART_KEY_COLUMNS](#)
- [USER_SYNONYMS](#)
- [USER_SYS_PRIVS](#)
- [USER_PART_TABLES](#)
- [USER_TRIGGERS](#)
- [USER_TABLES](#)
- [USER_TABLESPACES](#)
- [USER_TAB_COLS](#)
- [USER_TAB_COLUMNS](#)
- [USER_SUBPARTITION_TEMPLATES](#)
- [USER_TAB_COMMENTS](#)
- [USER_TYPES](#)
- [USER_TYPE_ATTRS](#)
- [USER_TYPE_METHODS](#)
- [USER_VIEWS](#)
- [USER_RECYCLEBIN](#)
- [USER_DEPENDENCIES](#)
- [USER_JOBS](#)
- [USER_PART_COL_STATISTICS](#)
- [USER_PART_HISTOGRAMS](#)
- [USER_SUBPART_COL_STATISTICS](#)
- [USER_SUBPART_HISTOGRAMS](#)
- [USER_TAB_COL_STATISTICS](#)
- [USER_TAB_HISTOGRAMS](#)
- [USER_TAB_STATISTICS](#)
- [ROLE_TAB_PRIVS](#)

- [ROLE_SYS_PRIVS](#)
- [ROLE_ROLE_PRIVS](#)
- [DICTIONARY](#)
- [DICT](#)
- [DBA_SQL_PLAN_BASELINES](#)
- [ALL_IND_STATISTICS](#)
- [DBA_IND_STATISTICS](#)
- [USER_IND_STATISTICS](#)
- [ALL_TAB_MODIFICATIONS](#)
- [DBA_TAB_MODIFICATIONS](#)
- [USER_TAB_MODIFICATIONS](#)
- [USER_TAB_PRIVS](#)
- [USER_PROFILES](#)
- [DBA_SCHEDULER_JOBS](#)
- [DBA_SCHEDULER_PROGRAM](#)
- [DBA_CONTEXT](#)

6.2 性能视图

OceanBase 数据库兼容 Oracle 数据库的性能视图列表如下：

- [V\\$NLS_PARAMETERS](#)
- [V\\$SESSION_WAIT](#)
- [V\\$SESSION_WAIT_HISTORY](#)
- [V\\$SESSTAT](#)
- [V\\$SQL_WORKAREA](#)
- [V\\$SQL_WORKAREA_ACTIVE](#)
- [V\\$SQL_WORKAREA_HISTOGRAM](#)
- [V\\$SYSSTAT](#)
- [V\\$SYSTEM_EVENT](#)
- [V\\$VERSION](#)
- [V\\$SQL_MONITOR_STATNAME](#)
- [V\\$GLOBAL_TRANSACTION](#)
- [V\\$TIMEZONE_NAMES](#)

- [V\\$ENCRYPTED_TABLESPACES](#)
- [V\\$RSRC_PLAN](#)
- [V\\$GLOBALCONTEXT](#)
- [V\\$INSTANCE](#)
- [V\\$ACTIVE_SESSION_HISTORY](#)
- [V\\$EVENT_NAME](#)
- [V\\$SQL_PLAN_MONITOR](#)

7 与 MySQL 兼容性对比

本节主要介绍 OceanBase 数据库的 MySQL 模式与原生 MySQL 数据库的兼容性对比信息。

OceanBase 数据库的 MySQL 模式兼容 MySQL 5.7/8.0 的绝大部分功能和语法。由于产品架构不同，或者客户需求不大，有些功能并没有被支持。本节主要从以下几方面介绍 OceanBase 数据库的 MySQL 模式与原生 MySQL 数据库的不同：

- 数据类型
- SQL 语法
- 系统视图
- 字符集和字符序
- 函数与表达式
- 分区支持
- 备份恢复
- 存储引擎
- 优化器
- 暂不支持的功能

7.1 数据类型

OceanBase 数据库支持的数据类型有：

- 数值类型
 - 整数类型： `BOOL` / `BOOLEAN` / `TINYINT`、`SMALLINT`、`MEDIUMINT`、`INT` / `INTEGER` 和 `BIGINT`。
 - 定点类型： `DECIMAL` 和 `NUMERIC`。
 - 浮点类型： `FLOAT` 和 `DOUBLE`。
 - Bit-Value 类型： `BIT`。
- 日期时间类型： `DATETIME`、`TIMESTAMP`、`DATE`、`TIME` 和 `YEAR`。
- 字符类型： `CHAR`、`VARCHAR`、`BINARY` 和 `VARBINARY`。
- 大对象类型： `TINYBLOB`、`BLOB`、`MEDIUMBLOB` 和 `LONGBLOB`。
- 文本类型： `TINYTEXT`、`TEXT`、`MEDIUMTEXT` 和 `LONGTEXT`。

- 枚举类型： `ENUM` 。
- 集合类型： `SET` 。
- JSON 数据类型。

与 MySQL 数据库对比，OceanBase 数据库暂不支持空间数据类型，其他类别的数据类型支持情况是等于或大于 MySQL 数据库的。

7.2 SQL 语法

SELECT

- 支持大部分查询功能，包括支持单、多表查询；支持子查询；支持内联接、半联接以及外联接；支持分组、聚合；常见的概率、线性回归等数据挖掘函数等。
- 支持对多个 `SELECT` 查询的结果进行 `UNION`、`UNION ALL`、`MINUS`、`EXCEPT` 或 `INTERSECT` 等集合操作。
- 支持通过如下方式查看执行计划：

```
EXPLAIN [explain_type] dml_statement;
```

```
explain_type:
```

```
BASIC
```

```
| OUTLINE
```

```
| EXTENDED
```

```
| EXTENDED_NOADDR
```

```
| PARTITIONS
```

```
| FORMAT = {TRADITIONAL|JSON}
```

```
dml_statement:
```

```
SELECT statement
```

```
| DELETE statement
```

```
| INSERT statement
```

```
| REPLACE statement
```

```
| UPDATE statement
```

- 不支持 `SELECT ... FOR SHARE ...` 语法。

INSERT

- 支持单行和多行插入，以及指定分区插入。
- 支持 `INSERT INTO ... SELECT ...` 语句。

UPDATE

- 支持单列和多列更新。
- 支持使用子查询。
- 支持集合更新。

DELETE

- 支持单表和多表删除。

TRUNCATE

- 支持完全清空指定表。
- 不支持在进行事务处理和表锁定的过程中操作。

7.3 系统视图

OceanBase 数据库实现了 `information_schema` 和 `mysql` 这两个内部数据库中的大部分视图，但是由于架构不同，OceanBase 数据库并不保证所有视图均能实现以及视图中所有的列含义与 MySQL 相同。

更多系统视图的说明信息请参考《参考指南（MySQL 模式）》文档中 [系统视图](#) 章节。

7.4 字符集和字符序

OceanBase 数据库兼容 MySQL 数据库的部分字符集和字符序，具体支持情况如下：

- 字符集：binary、utf8mb4、gbk、utf16 和 gb18030。
- 字符序：utf8mb4_general_ci、utf8mb4_bin、binary、gbk_chinese_ci、gbk_bin、utf16_general_ci、utf16_bin、utf8mb4_unicode_ci、utf16_unicode_ci、gb18030_chinese_ci 和 gb18030_bin。

7.5 函数

与 MySQL 数据库对比，OceanBase 数据库的 MySQL 模式不支持如下函数：

- 字符串函数：`CHARACTER_LENGTH()`、`FROM_BASE64()`、`LOAD_FILE()`、`MATCH()`、`OCTET_LENGTH()`、`SOUNDEX()` 和 `TO_BASE64()`。
- XML 函数：`ExtractValue()` 和 `UpdateXML()`。

- 加密和压缩函数： `COMPRESS()`、`RANDOM_BYTES()`、`SHA1()`、`SHA()`、`SHA2()`、`UNCOMPRESS()`、`UNCOMPRESSED_LENGTH()` 和 `VALIDATE_PASSWORD_STRENGTH()`。
- 锁定函数： `GET_LOCK()`、`IS_FREE_LOCK()`、`IS_USED_LOCK()`、`RELEASE_ALL_LOCKS()` 和 `RELEASE_LOCK()`。
- 其他函数： `MASTER_POS_WAIT()` 和 `NAME_CONST()`。

另外，OceanBase 数据库的 MySQL 模式不支持空间分析函数和性能模式函数。OceanBase 数据库所支持的分析（窗口）函数是 MySQL 数据库的超集，即 MySQL 数据库的分析（窗口）函数都支持。

7.6 分区支持

OceanBase 数据库与 MySQL 数据库对分区的支持差异如下：

- OceanBase 数据库支持一级分区，模板化和非模板化二级分区；MySQL 数据库不支持非模板化二级分区。
- OceanBase 数据库的二级分区支持 Hash、Key、Range、Range Columns、List 和 List Columns 分区；MySQL 数据库的二级分区仅支持 Hash 分区和 Key 分区。

更多分区的说明及使用请参见 [分区管理](#)。

7.7 备份恢复

OceanBase 数据库兼容了部分 MySQL 数据库的备份恢复特性，主要支持情况如下：

- 支持全量备份和增量备份。
- 不支持集群级别的备份恢复。
- 仅支持热备份，不支持冷备份。
- 不支持租户内部部分数据库和表级的备份恢复。
- 不支持备份数据的有效性验证。

7.8 存储引擎

与 MySQL 数据库基于数据块的 InnoDB 和 MyISAM 引擎不同，OceanBase 数据库使用的是基于 LSM-Tree 架构的存储引擎。

7.9 优化器

OceanBase 数据库在优化器方面与 MySQL 数据库的区别，主要表现在以下几个方面：

- 查看执行计划的命令
 - 输出的列信息仅包含 ID、OPERATOR、NAME、EST. ROWS 和 COST 以及算子的详细信息。
 - 不支持使用 SHOW WARNINGS 显示额外的信息。
- 查看统计信息
 - 支持执行 ANALYZE TABLE 语句查询数据字典表存储有关列值的直方图统计信息。
 - 支持通过内部表 __all_meta_table 查看表统计信息和列统计信息。
- 查询改写优化
 - 支持外联接优化
 - 支持外联接简化
 - 支持块嵌套循环和批量 Key 访问联接
 - 支持条件过滤
 - 支持常量叠算优化
 - 支持 IS NULL 优化 (索引不存储 NULL 值)
 - 支持 ORDER BY 优化
 - 支持 GROUP BY 优化
 - 支持 DISTINCT 消除
 - 支持 LIMIT 下压
 - 支持 Window 函数优化
 - 支持避免全表扫描
 - 支持谓词下压
- Optimizer Hint 机制
 - 支持联接顺序 Optimizer Hints
 - 支持表级别的 Optimizer Hints
 - 支持索引级别的 Optimizer Hints
 - 语法支持 INDEX Hint、FULL Hint、ORDERED Hint 和 LEADING Hint 等, 不支持 USE INDEX 和 FORCE INDEX。

- 兼容 MySQL 数据库的并行执行能力包括并行查询、并行复制和并行写入等，且 OceanBase 数据库已经支持并行算子，包括并行聚集、并行联接、并行分组以及并行排序等。

- OceanBase 数据库还支持计划缓存和预编译，MySQL 数据库并不支持。

更多优化器的详细信息请参见 [SQL 调优指南](#)。

7.10 暂不支持的功能

- 暂不支持空间数据类型。
- 不支持 `SELECT ... FOR SHARE ...` 语法。
- 不支持空间分析函数和性能模式函数。
- 对于备份恢复功能，不支持集群级别的备份恢复，不支持冷备份，不支持租户内部部分数据库和表级的备份恢复，备份数据的有效性验证。
- 对于优化器，查看执行计划的命令不支持使用 `SHOW WARNINGS` 显示额外的信息。

8 使用限制

8.1 标识符长度限制

- MySQL 模式

数据项	最大长度
集群名	128 字节
租户名	64 字节
用户名	64 字节
数据库名	128 字节
表名	64 字符
列名	128 字节
索引名	64 字节
视图名	64 字节
别名	255 字节
表组名	128 字节

- Oracle 模式

类型	最大长度
集群名	128 字节
租户名	64 字节
用户名	64 字节
表名	128 字节
列名	128 字节
索引名	128 字节
视图名	128 字节
别名	255 字节
对象名	128 字节
表组名	128 字节

8.2 ODP 最大连接数限制

类型	最大限制
单个 ODP 的连接数	由 ODP 的 <code>client_max_connections</code> 参数控制，默认为 8192。 说明 您可以通过增加 ODP 节点数或者修改 <code>client_max_connections</code> 配置项值的方式来增大集群的连接总数限制。

8.3 分区副本数限制

类型	最大限制
每个 OBServer 节点的分区副本数	无严格限制 说明 每个 OBServer 节点的分区副本数可根据租户内存大小来预估，1G 内存约支持约 2 万 tablet。

8.4 单个表的限制

类型	最大限制
行长度	1.5M 字节
行数	无限制
列数	4096 列
索引个数	128 个
索引总列数	512 列
索引长度	1.5M 字节
主键总列数	64 列
主键长度	16K
分区个数	<ul style="list-style-type: none">• Oracle 模式：65536 个• MySQL 模式：8192 个

8.5 单列的限制

类型	最大限制
索引单个列长度	262143 字节

8.6 字符串类型限制

- MySQL 模式

类型	最大长度
CHAR	256 字节
VARCHAR	1048576 字节
BINARY	256 字节
VARBINARY	1048576 字节
BLOB	536870911 字节
TEXT	536870911 字节

- Oracle 模式

类型	最大长度
CHAR	2000 字节
VARCHAR	32767 字节
VARCHAR2	32767 字节
NCHAR	2000 字节
NVARCHAR2	32767 字节