

虚谷数据库 V12.8

系统配置参数参考指南

文档版本 01

发布日期 2025-04-15



版权所有 © 2025 成都虚谷伟业科技有限公司。

声明

未经本公司正式书面许可，任何企业和个人不得擅自摘抄、复制、使用本文档中的部分或全部内容，且不得以任何形式进行传播。否则，本公司将保留追究其法律责任的权利。

用户承诺在使用本文档时遵守所有适用的法律法规，并保证不以任何方式从事非法活动。不得利用本文档内容进行任何侵犯他人权益的行为。

商标声明



为成都虚谷伟业科技有限公司的注册商标。

本文档提及的其他商标或注册商标均非本公司所有。

注意事项

您购买的产品或服务应受本公司商业合同和条款的约束，本文档中描述的部分产品或服务可能不在您的购买或使用范围之内。由于产品版本升级或其他原因，本文档内容将不定期进行更新。

除非合同另有约定，本文档仅作为使用指导，所有内容均不构成任何声明或保证。

成都虚谷伟业科技有限公司

地址：四川省成都市锦江区锦盛路 138 号佳霖科创大厦 5 楼 3-14 号

邮编：610023

网址：www.xugudb.com

前言

概述

本文档介绍了虚谷数据库各种系统配置文件中的参数名称、解释、默认值、取值范围以及相关使用注意事项等。

读者对象

本文档主要适用于以下用户：

- 数据库维护人员
- 数据库管理员

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 注意	用于传递设备或环境安全警示信息，若不避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。
 说明	对正文中重点信息的补充说明。“说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录

文档版本	发布日期	修改说明
01	2025-04-15	第一次正式发布。

目录

1	概述	1
2	xugu.ini 系统文件配置	2
2.1	网络侦听参数	2
2.1.1	conn_fail_cnt	2
2.1.2	def_charset	3
2.1.3	def_timefmt	4
2.1.4	def_timezone	5
2.1.5	listen_port	6
2.1.6	login_timeout	8
2.1.7	max_act_conn_num	9
2.1.8	max_conn_num	10
2.1.9	max_idle_time	11
2.1.10	min_pass_len	12
2.1.11	nio_timeout	13
2.1.12	pass_mode	14
2.1.13	send_warning	15
2.1.14	session_per_user	16
2.1.15	use_std_nio	17
2.2	并行管理参数	18
2.2.1	cata_hash_size	18
2.2.2	cata_parti_num	19
2.2.3	lock_hash_size	20
2.2.4	max_parallel	21
2.2.5	rsync_thd_num	22
2.2.6	rtran_thd_num	24
2.2.7	task_thd_num	25
2.2.8	tcp_thd_num	26

2.2.9	thd_bind_mode	27
2.3	系统缓存参数	28
2.3.1	data_buff_mem	28
2.3.2	ena_share_sga	29
2.3.3	hash_parti_num	30
2.3.4	max_hash_mem	31
2.3.5	max_hash_size	32
2.3.6	max_malloc_once	33
2.3.7	max_task_mem	34
2.3.8	swap_buff_mem	35
2.3.9	system_sga_mem	36
2.3.10	xlog_buff_mem	37
2.4	SQL 引擎参数	38
2.4.1	auto_eje_cast	38
2.4.2	auto_eje_parallel	39
2.4.3	auto_use_eje	40
2.4.4	cata_case_sensitive	41
2.4.5	check_unique_mode	42
2.4.6	ddl_timeout	43
2.4.7	def_empty_str_as_null	44
2.4.8	def_optimize_mode	45
2.4.9	delay_check_unique	46
2.4.10	enable_eje_big_block	47
2.4.11	enable_find_synonym	48
2.4.12	enable_histogram	49
2.4.13	enable_stream_import	49
2.4.14	enable_sys_name_idx	50
2.4.15	idx_delay_del_limit	51
2.4.16	iscan_ini_cost	52
2.4.17	max_cursor_num	53

2.4.18	max_loop_num	54
2.4.19	max_prepare_num	55
2.4.20	para_eje_seqscan_num	56
2.4.21	prepare_param_num	57
2.4.22	prepare_reuse	57
2.4.23	proc_reuse_cnt	58
2.4.24	select_table_num	59
2.4.25	seqscan_skip_err	60
2.4.26	str_trunc_warning	61
2.4.27	supple_prepare_sql	62
2.4.28	support_global_tab	63
2.4.29	tab_rebuild_limit	64
2.4.30	use_index_order	65
2.5	兼容性参数	66
2.5.1	def_compatible_mode	66
2.5.2	def_group_by_mode	68
2.5.3	def_identity_mode	69
2.5.4	use_old_product_name	71
2.6	存储管理函数	72
2.6.1	auto_extend_dfile	72
2.6.2	block_pctfree	73
2.6.3	block_size	73
2.6.4	data_file_append_mode	74
2.6.5	def_data_space_size	75
2.6.6	def_file_step_size	76
2.6.7	def_redo_file_size	77
2.6.8	def_temp_space_size	78
2.6.9	def_undo_space_size	79
2.6.10	default_copy_num	80
2.6.11	enable_read_copy2	81

2.6.12	enable_store_migrate	81
2.6.13	index_preload_cnt	82
2.6.14	init_data_space_num	83
2.6.15	init_temp_space_num	84
2.6.16	init_undo_space_num	84
2.6.17	ioerr_report_mode	85
2.6.18	max_file_size	86
2.6.19	max_hotspot_num	86
2.6.20	max_temp_space_size	87
2.6.21	ov_reuse	88
2.6.22	safely_copy_num	89
2.6.23	size_per_chunk	90
2.6.24	store_drop_delay	90
2.6.25	store_maint_span	91
2.7	存储管理附属函数	92
2.7.1	cache_undo_wrt	92
2.7.2	catalog_write_through	93
2.7.3	data_persistence	94
2.7.4	datafile_random_access	95
2.7.5	datafile_use_aio	96
2.7.6	datafile_write_through	97
2.7.7	major_retry_num	98
2.7.8	max_write_back_num	99
2.7.9	redo_write_through	100
2.7.10	write_back_thd_num	101
2.8	事务管理参数	102
2.8.1	checkpoint_delay	102
2.8.2	def_auto_commit	103
2.8.3	def_iso_level	103
2.8.4	dlock_check_delay	104

2.8.5	exit_when_redo_err	105
2.8.6	max_trans_modify	106
2.8.7	node_dead_delay	107
2.8.8	skip_boot_rollback	108
2.8.9	strictly_commit	109
2.8.10	undo_delay_free	110
2.9	数据备份同步参数	111
2.9.1	enable_recycle	111
2.9.2	ignore_when_meta_err	112
2.9.3	log_archive_mode	113
2.9.4	log_supplement	114
2.9.5	max_allow_lob_len	114
2.9.6	modify_log_parti_num	115
2.9.7	modify_log_pice_len	116
2.10	系统运行日志参数	117
2.10.1	core_dump	117
2.10.2	enable_node_degrade	118
2.10.3	errlog_size	119
2.10.4	error_level	120
2.10.5	is_zh	121
2.10.6	reg_command	122
2.10.7	reg_ddl	123
2.10.8	slow_sql_time	123
2.10.9	trace_login	124
2.11	安全审计参数	125
2.11.1	enable_audit	125
2.11.2	security_level	126
2.12	系统分析参数	127
2.12.1	analyze_level	127
2.12.2	analyze_mode	128

2.12.3	analyze_threshold	129
2.12.4	analyze_time	130
2.12.5	debug_flag	131
2.12.6	enable_analyze	132
2.12.7	enable_monitor	133
2.12.8	gstore_pick_mode	134
2.12.9	histogram_bucket	135
3	cluster.ini 系统文件配置	137
3.1	概述	137
3.2	全局参数	137
3.3	节点参数	139
4	types.ini 系统文件配置	141
5	mount.ini 系统文件配置	142
6	trust.ini 系统文件配置	144
7	连接会话参数	146
7.1	DATABASE	146
7.2	USER	146
7.3	PASSWORD	146
7.4	SESSION_USER	147
7.5	DRIVER_VERSION	147
7.6	LANGUAGE	148
7.7	CHAR_SET	148
7.8	TIME_FORMAT	149
7.9	ISO_LEVEL	149
7.10	AUTO_COMMIT	150
7.11	STRICT_COMMIT	151
7.12	RESULT	152
7.13	RETURN_ROWID	154
7.14	RETURN_SCHEMA	154

7.15	RETURN_CURSOR_ID	155
7.16	LOB_RET	156
7.17	EMPTY_STR_AS_NULL	156
7.18	OPTIMIZER_MODE	157
7.19	COMPATIBLE_MODE	158
7.20	TRANS_READONLY	159
7.21	DISABLE_BINLOG	159
7.22	IDENTITY_MODE	160
7.23	KEYWORD_FILTER	161
8	内存全局变量	163
8.1	LICENSE_TYPE	163
8.2	CLUSTER_FAULT_LEVEL	164

1 概述

虚谷数据库系统配置参数用于存储文件、集群通信、类型映射、黑白名单和其他系统运行时控制。

虚谷数据库系统配置参数文件如表1-1所示。

表 1-1 系统配置参数文件

配置文件	配置说明
xugu.ini	数据库运行时参数配置（网络监听及服务、系统调度和并行、缓冲、SQL 引擎、存储系统参数、备份同步、系统日志、审计、自动分析）
cluster.ini	集群参数配置（规模、通信、角色、算力、存储权重等）
types.ini	基础数据类型映射（用于兼容异构数据库迁移）
mount.ini/datafile	数据库存储位置管理（控制文件、数据文件、事务日志等）
trust.ini	黑白名单配置
连接会话参数	连接级参数设置（作用范围为连接）
内存全局变量	相关内存全局变量

2 xugu.ini 系统文件配置

2.1 网络侦听参数

2.1.1 conn_fail_cnt

参数名称

conn_fail_cnt

参数解释

设置禁止登录的失败次数。

建立连接时最大允许的失败次数，计数以 IP 地址为单元。3 分钟内登录失败次数大于 conn_fail_cnt，则拒绝该 IP 地址所有用户建立 socket 请求。失败次数超时清零（超时时间 3 分钟）后，才允许该 IP 地址建立连接。

默认值

3

取值范围

[2, 100]

访问权限

R/W

是否全局参数

是

修改生效方式

conn_fail_cnt 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 conn_fail_cnt 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET conn_fail_cnt TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW conn_fail_cnt
```

注意事项

- 如果某个 IP 地址有程序不间断的发生登录失败（3 分钟内超过 conn_fail_cnt），则会出现该 IP 地址所有程序都无法建立连接。相关系统表 sys_forbidden_ips 可查询具体被禁用的 IP 地址情况。
- 同一 IP 地址三次不间断登录失败，第四次正常登录无法登录，三分钟后才能正常登录。

2.1.2 def_charset

参数名称

def_charset

参数解释

设置默认新建库和会话字符集。

若建立连接时没有指定客户端字符集，则使用此默认值。连接字符集和当前库字符集不一致时，数据库在读取客户端数据时将字符串从连接字符集转换为当前库字符集，返回客户端数据时把字符串从当前库字符集转换为连接字符集再返回。

从性能角度考虑，建议用户在使用时，设置连接字符集为库字符集，节省不必要的字符串转换开销（同时考虑用户程序输入输出字符串字符集、连接字符集、库字符集）。

虚谷数据库只支持库级的字符集设置，不支持表级和列级，且库字符集在设置后不能修改。

默认值

GBK

取值范围

[UTF8,GBK,GB18030,BINARY,etc.]

可设置的字符集为 sys_charsets 系统表中记录的字符集，SYSDBA 用户可执行 SQL 命令“SELECT * FROM sys_charsets;”查看。

访问权限

R/W

是否全局参数

是

修改生效方式

def_charset 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_charset 参数值，根据实际需求在取值范围内选择合适

的 value 值。

```
SQL> SET def_charset TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_charset
```

注意事项

字符集 GBK 单个中文字符占 2 字节，字符集 UTF8 单个中文字符占 3 字节。

2.1.3 def_timefmt

参数名称

def_timefmt

参数解释

设置客户端的默认时间格式。

若建立连接时没有指定时间格式，在以下场景中则使用此默认值：

- 时间转字符串：如 TO_CHAR(时间类型参数) 函数。
- 字符串转时间：如 TO_DATE(字符类型参数) 函数。
- 时间字段插入：通过字符串插入时间类型字段。

默认值

YYYY-MM-DD HH24:MI:SS

取值范围

[YYYY-MM-DD HH24:MI:SS,YYYYMMDDHH24MISS,etc.]

访问权限

R/W

是否全局参数

是

修改生效方式

def_timefmt 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

def_timefmt 参数通过 SET 修改仅对之后新建连接生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_timefmt 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_timefmt TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_timefmt
```

注意事项

当时间串格式不带时分秒时（如 SET def_timefmt TO 'YYYY-MM-DD'），执行 SQL 命令 SELECT TO_DATE(SYSDATE)，输出结果的时、分、秒部分为 00:00:00.000。因为数据库只有 TO_DATE(CHAR) 函数实现，执行时 SYSDATE 会按设置时间格式转换成 CHAR 类型再执行。

2.1.4 def_timezone

参数名称

def_timezone

参数解释

设置默认客户端时区。

- 定义数据库默认时区：控制新建数据库和新建连接的时区基准。
- 独立于操作系统时区：数据库系统时间以 UNIX 时间戳（从 1970 年 1 月 1 日开始到操作系统实时时间）为基础，结合 def_timezone 转换为对应时区的时间。因此修改操作系统的时区不会影响数据库时间，数据库时间依赖于 def_timezone 而非操作系统时区。

默认值

GMT+08:00

取值范围

[GMT-12:59,GMT+14:59]

访问权限

R/W

是否全局参数

是

修改生效方式

def_timezone 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_timezone 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_timezone TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_timezone
```

注意事项

用户口令认证时，要求当前时间必须大于数据库创建时间且小于过期时间。修改时区若导致当前时间小于数据库创建时间，则无法成功登录。

2.1.5 listen_port

参数名称

listen_port

参数解释

数据库默认 TCP 侦听端口。

用于数据库对外服务，单节点模式或集群配置有 Q 角色节点才会进行侦听。

默认值

5138

取值范围

[1024,65535]

访问权限

R/W

是否全局参数

是

修改生效方式

listen_port 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 执行以下命令确认需要被设置的端口是否被占用。

- Windows:

```
netstat -ano | findstr <port>
```

- Linux:

```
netstat -tuln | grep <port>
```

3. 修改 SETUP 文件夹下 xugu.ini 文件中的 listen_port 参数为新的端口号，保存并退出。
4. 重新启动数据库服务，使用新设置的端口号连接数据库。
5. 在数据库中执行以下命令查看当前侦听端口。

```
SQL> SHOW listen_port
```

注意事项

端口冲突将导致侦听失败，数据库进程退出。

数据库侦听建立是数据库启动过程的最后一个步骤，虽然此时出现端口冲突导致启动失败进程退出，但启动流程中其他操作（如加载系统表，加载 JOB，初始化内存，REDO，UNDO，集群握手等）已完成。

非必要不建议修改 listen_port，在测试环境如果需要单个物理机上启动多个数据库服务，此参数可按实际情况修改。

注意

- 修改端口注意不要和其他程序端口冲突（如其他数据库端口、中间件端口等）。
- 设置“listen_port < 1024”可启动成功，但不建议使用小于 1024 的端口，0 至 1023 端口默认固定分配给某个服务。
- 设置“listen_port > 65535”可启动成功且日志无报错信息，但通过该端口不可建立连接。
- 端口发生冲突时，启动数据库会报“create_server_port failed”错误。

侦听端口配置异常及风险：

- 集群模式单个节点加入时，节点加入成功后又退出。
- 集群模式整个集群启动时，集群启动成功立刻出现节点死亡（可能是 MASTER 节点和其他任意类型节点，甚至整集群）。
- 操作系统防火墙必须开放监听端口，否则用户或程序无法访问数据库。

2.1.6 login_timeout

参数名称

login_timeout

参数解释

数据库登录超时时间（单位：秒），由清理死亡和空闲连接 JOB 使用。

登录简要过程：

1. 驱动对数据库监听端口（默认 5138）发起 TCP 握手。
2. 数据库监听线程处理握手信号并登记新连接信息，等待驱动下一步发送登录信息（登录库、用户、密码等）。
3. 驱动发送登录信息。
4. 数据库监听线程开始登录验证并在连接会话属性中添加相关内容。

默认值

30

取值范围

[10,300]

访问权限

R/W

是否全局参数

是

修改生效方式

login_timeout 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 login_timeout 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET login_timeout TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW login_timeout
```

注意事项

- 登录超时时间默认 30 秒，socket 建立完成后设置休眠时间为 31 秒。

- 如果登录超过 30 秒，数据库死亡连接清理 JOB 未运行且建立的 socket 已关闭，则数据库 TRACE.LOG 记载“TCP_NET_ERR”（如果休眠时间与 login_timeout 时间设置相近且休眠完成后立即关闭 socket，则出现“TCP_NET_ERR”概率较大）。
 - 如果登录超过 30 秒，数据库死亡连接清理 JOB 正运行，则 TRACE.LOG 记载“CLR_DBC”。
 - 此步骤完成后 login_timeout 还未收到驱动发送的登录信息，则连接清理 JOB 则可能判定此登录操作超时，同时 TRACE.LOG 记载日志“clear dead connect ...state:0 trust:0...”。
- 修改 login_timeout 为 60 秒，该参数修改后立即生效。重复上述步骤（其中休眠时间设置为 60+n）。
 - Socket 建立后，在 login_timeout 前中后发送登录信息查看相应信息是否正确。

2.1.7 max_act_conn_num

参数名称

max_act_conn_num

参数解释

最大活动事务连接数，控制最大并发事务。

已开始事务的连接请求优先处理，并发执行事务的连接数达到 max_act_conn_num 值后，后续未开始事务处理的连接请求排队等候正在执行事务的连接提交。

默认值

0

取值范围

[0,1024]

取值为 0 时，表示不做限制。

访问权限

R/W

是否全局参数

是

修改生效方式

max_act_conn_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 max_act_conn_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_act_conn_num
```

注意事项

极限条件下，如活动未提交事务数已消耗完 max_act_conn_num，连接不提交事务，后续连接将无法执行新事务，此时需要调整此参数值。

2.1.8 max_conn_num

参数名称

max_conn_num

参数解释

系统最大连接数（标准监听有效）。

- 标准监听模型时数据库允许的最大连接数，实际可用连接数为“MIN(max_conn_num,1024-文件消耗句柄)”，文件使用句柄可在/proc/进程号/fd 目录查看。
- 非标准监听时最大连接数硬编码设置为 10000。
- 当 use_std_nio 为 false 时，此参数不生效。

默认值

1000

取值范围

[1,10000]

访问权限

R/W

是否全局参数

是

修改生效方式

max_conn_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_conn_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_conn_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_conn_num
```

注意事项

无

2.1.9 max_idle_time

参数名称

max_idle_time

参数解释

设置最大闲置时间（单位：秒）。

在最大闲置时间内若无请求则断开连接。

连接存在如下几种状态：

- 初始：握手成功。
- 登录：进行登录验证。
- 空闲：无请求处理。
- 活动：正在执行用户请求。
- 消亡：连接已关闭或中断。

连接状态变换流程如下：

- 初始 > 登录 > 空闲 <-> 活动 > 消亡。
- 登录 > 消亡：登录超时或用户验证失败。
- 空闲 > 消亡：空闲时间过长。

- 活动 > 消亡：连接异常、网络中断。

默认值

3600

取值范围

[0, 86400]

取值为 0 时，表示无限制。

访问权限

R/W

是否全局参数

是

修改生效方式

max_idle_time 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_idle_time 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_idle_time TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_idle_time
```

注意事项

- max_idle_time = 0 时，不会断开连接。
- max_idle_time = 非 0 数值时，清理时间根据清理 JOB 运行时间相关。

2.1.10 min_pass_len

参数名称

min_pass_len

参数解释

设置口令最小长度，按需配置。

默认值

8

取值范围

[2,127]

访问权限

R/W

是否全局参数

是

修改生效方式

min_pass_len 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 min_pass_len 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET min_pass_len TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW min_pass_len
```

注意事项

无

2.1.11 nio_timeout

参数名称

nio_timeout

参数解释

设置网络读写超时时间（单位：秒）。

数据库连接建立后，数据库对连接底层 socket 进行数据读写操作超时时间，对应底层 socket 接收（SO_RCVTIMEO）和发送（SO_SNDTIMEO）超时时间。

默认值

100

取值范围

[10,10000]

访问权限

R/W

是否全局参数

是

修改生效方式

nio_timeout 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 nio_timeout 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW nio_timeout
```

注意事项

如果数据库对连接底层 socket 进行数据读写超时，则会读写失败，数据库断开和客户端连接并记载错误“接收失败（网络中断）”或“发送失败（网络中断）”。

2.1.12 pass_mode

参数名称

pass_mode

参数解释

设置口令模式。

- 1: 任意不重复字符或数字，如 123456, abc123, P@ssw0rd!。
- 2: 必须含字母和数字，如 abc123, P@ssw0rd!。
- 3: 必须含字母、数字和特殊符号，如 P@ssw0rd!。

模式向上兼容，即 pass_mode = 1 时，可按照 2, 3 的规则设置口令。

默认值

3

取值范围

[1|2|3]

访问权限

R/W

是否全局参数

是

修改生效方式

pass_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 pass_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET pass_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW pass_mode
```

注意事项

密码不允许为空或重复字符，当设置的密码不符合模式规则会报“E18001 口令太简单”。

2.1.13 send_warning

参数名称

send_warning

参数解释

设置是否发送警告信息。

设置为 false 时错误级别为 1（NOTICE 级别）的错误不再返回到客户端，如事务报错点不存在、字符超长被截断（str_trunc_warning=true）。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

send_warning 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 send_warning 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET send_warning TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW send_warning
```

注意事项

无

2.1.14 session_per_user

参数名称

session_per_user

参数解释

用户默认可用连接数。

- 用户连接配额创建用户时，未指定 SESSION 选项值，则使用此参数值作为用户连接数配额。
- 从磁盘读取用户信息时，如果用户 SESSION_QUOTA 小于 0 则使用此参数。
- 用户连接配额为节点级别参数，即用户在每个工作节点都可建立配合允许的连接数。

默认值

1000

取值范围

[0,10000]

取值为 0 时，表示不限制。

访问权限

R/W

是否全局参数

是

修改生效方式

session_per_user 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤:

1. 在数据库中执行以下命令修改 session_per_user 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET session_per_user TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW session_per_user
```

注意事项

无

2.1.15 use_std_nio

参数名称

use_std_nio

参数解释

设置数据库是否使用标准网络监听器。

取值如果为 false，则使用依赖于操作系统的效率更高的网络侦听器。

数据库 TCP 监听模式存在两种，标准监听和非标准监听，标准监听 Windows 和 Linux 平台通用，非标准监听基于平台特性设计。

- true: 标准监听，使用 SELECT 模型。Linux 环境该模型只支持 1024 个句柄，数据库理论上最多支持 1024 个连接，因为数据库还有文件使用句柄（可在 /proc/进程号/fd 目录查看），所以数据库实际支持连接数量为 1024-文件消耗句柄 ≈ 1000 个。
- false: 非标准监听，使用其他模型支持 10000 个连接（仅 Linux 平台实现）。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

use_std_nio 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 use_std_nio 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW use_std_nio
```

注意事项

- use_std_nio = true
 - 默认约 1000 个连接，在无负载情况下每个连接建立速度几乎为毫秒级，单机顺序建立 1000 连接耗时约 6 秒左右。
 - 修改参数 max_conn_num，则连接数根据 max_conn_num 和 session_per_user 变化。
- use_std_nio = false
 - max_conn_num 参数设置无效。
 - 如果 session_per_user 参数未修改为默认参数，则建立 10000 连接时超过 1000 连接时启动页面打印 “session_quota exceeded”。
 - “session_per_user = 10000” 时，可成功建立 10000 连接（前提：操作系统支持 10000files）。

2.2 并行管理参数

2.2.1 cata_hash_size

参数名称

cata_hash_size

参数解释

设置系统对象目录速查哈希表尺度。

系统对象（如库、模式、表、过程、函数等）元信息缓存哈希表的桶数（Bucket Count）。

默认值

8192

取值范围

[1024,131072]

访问权限

R/W

是否全局参数

是

修改生效方式

cata_hash_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 cata_hash_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。

3. 重新启动数据库服务。

4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW cata_hash_size
```

注意事项

最优值建议设置为系统可预估的最大对象个数。

2.2.2 cata_parti_num

参数名称

cata_parti_num

参数解释

设置目录分区数量。

控制各类资源（锁、数据缓存、用户权限等）分区个数，以提高数据库运行过程中的并发资源访问冲突，提高系统吞吐能力。

默认值

32

取值范围

[1,256]

访问权限

R/W

是否全局参数

是

修改生效方式

cata_parti_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 cata_parti_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW cata_parti_num
```

注意事项

最优值建议设置为 CPU 核数，当核数大于 32 时配置为 CPU 核数的 1/2。

2.2.3 lock_hash_size

参数名称

lock_hash_size

参数解释

设置系统锁管理器哈希表尺度。

默认值

8192

取值范围

[1024,131072]

访问权限

R/W

是否全局参数

是

修改生效方式

lock_hash_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 lock_hash_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。

3. 重新启动数据库服务。

4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW lock_hash_size
```

注意事项

最优值建议设置为“cata_hash_size*2。”

2.2.4 max_parallel

参数名称

max_parallel

参数解释

设置单任务的最大并行执行路数，提升处理效率。

- 单机部署查询：等效于集群部署 cluster.ini 配置文件中所有节点 LPU（逻辑处理单元）的总和，限制查询的并行线程数。
- 分区表维护：清理/删除表分区时，若需重建全局分区索引，使用此参数控制并行线程数。
- 隐式索引重建：隐式索引重建时，使用此参数用作索引构建的并行数。以下操作触发索引重建时自动应用：
 - 库、用户、模式、表恢复。

- 非 LOAD 语法流式导入。
- IMPORT ... SELECT 快速数据复制。
- 指定重建全局分区索引方式删除或清理表分区。

默认值

1

取值范围

[1,1024]

访问权限

R/W

是否全局参数

是

修改生效方式

max_parallel 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 max_parallel 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_parallel
```

注意事项

- 最优值建议设置为 CPU 核数的 2 倍。
- 集群部署场景下单个任务最大并行度取决于存活工作节点 cluster.ini 中配置的 LPU 总和，不受该参数影响。

2.2.5 rsync_thd_num

参数名称

rsync_thd_num

参数解释

设置远程同步处理线程数量。远程同步处理线程控制多副本数据同步的并发能力，确保主副本（数据主版本）向从副本同步数据时的高效性和一致性。

系统通过全局存储号（gstore_no）将同步任务分配到固定的处理线程。分配逻辑由函数“FN(gstore_no, rsync_thd_num)”实现，同一 gstore_no 的同步任务始终由相同线程处理，避免并发冲突。不同 gstore_no 的任务均匀分布到所有线程，提升资源利用率。

默认值

8

取值范围

[1,256]

访问权限

R/W

是否全局参数

是

修改生效方式

rsync_thd_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 rsync_thd_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW rsync_thd_num
```

注意事项

rsync_thd_num 的优化需在资源利用率和调度开销间平衡。

参数值越大，系统存储副本的同步并发冲突越低。但参数值过大时，过多的调度开销可能使性能降低，建议设置为 CPU 核数的 1/2。

2.2.6 rtran_thd_num

参数名称

rtran_thd_num

参数解释

设置远程事务处理线程数。远程事务处理线程用于分布式事务中的远程代理事务的提交、回滚、终止等操作。

系统通过事务号投递到远程事务处理线程。分配逻辑由函数“FN(根事务号,rtran_thd_num)”实现，同一事务始终由相同线程处理，避免并发冲突。

默认值

8

取值范围

[1,256]

访问权限

R/W

是否全局参数

是

修改生效方式

rtran_thd_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 rtran_thd_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW rtran_thd_num
```

注意事项

最优值建议设置为 CPU 核数的 1/2。

2.2.7 task_thd_num

参数名称

task_thd_num

参数解释

设置任务处理的初始线程数量，用于处理 TCP 接收任务和集群内部远程任务。

任务线程具备自动扩展机制。新执行任务放入待执行任务池管理，当任务积压不能快速执行时，系统判定为任务线程不足，系统自动按每调度组扩展 2 个线程，提升并发处理能力。

默认值

16

取值范围

[4,10000]

访问权限

R/W

是否全局参数

是

修改生效方式

task_thd_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 task_thd_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW task_thd_num
```

注意事项

- 初始线程数设定

任务线程不足时，任务请求在任务池中等待被执行，表现为任务响应延迟变长，线程扩展足够后延迟变长消除，但任务线程过多则引入不必要资源开销，所以需要考虑设置适当的

任务线程数，建议配置为“task_thd_num = 业务最大并发数 * 2”。

- 最大线程数计算

最大线程计算方式为“最大线程数 = (tcp_thd_num + msg_port_num) * 256”。

其中单机环境：“msg_port_num=1”，故“最大线程数 = (tcp_thd_num + 1) * 256”。

- 动态范围限制

最小可配置线程数为 16，当“task_thd_num < 16”或者“task_thd_num > 10000”时报错。

当“最大可配置线程数 < task_thd_num ≤ 10000”时，系统会自动将其调整为最大可配置线程数。

- 线程收缩

扩展后的线程不会自动回收，长期运行可能导致资源浪费。建议定期评估负载，手动调整 task_thd_num 适配实际需求。

2.2.8 tcp_thd_num

参数名称

tcp_thd_num

参数解释

设置 TCP 数据接收线程数量。

默认值

1

取值范围

[1,32]

访问权限

R/W

是否全局参数

是

修改生效方式

thd_bind_mode 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 thd_bind_mode 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW thd_bind_mode
```

注意事项

针对工作节点的吞吐量瓶颈问题，通常情况下设定 4 个线程即可在保障低时延的前提下完成请求接收与转发任务。

对于具备大规模并行处理能力的处理器（如 64 核以上 CPU），可根据实际应用场景适当增加 tcp_thd_num 配置，若无明显性能提升该参数尽量配小。

2.2.9 thd_bind_mode

参数名称

thd_bind_mode

参数解释

控制数据库线程与硬件服务器的 NUMA 节点绑定策略，优化多核 CPU 环境下的资源利用与性能。

- 0：不绑定。数据库线程在所有可用 CPU 核心上调度。
- 1：按 NUMA 节点绑定。线程仅在指定 NUMA 节点的所有核上调度，影响线程类型包括任务处理线程、TCP 接收线程、UDP 接收线程、数据同步线程和事务处理线程。
- 2：按 NUMA 节点分段绑定。在模式 1 基础上，为每个 TCP 和 UDP 接收线程绑定固定的一个核（消耗每个 NUMA 上尾部 CPU 核），任务线程、事务线程和同步线程使用 NUMA 上剩余 CPU 核。

SYSDBA 用户可通过 SQL 语句“SELECT NUMA_ID FROM sys_thd_status”查看线程所属 NUMA。

默认值

0

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

thd_bind_mode 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 thd_bind_mode 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW thd_bind_mode
```

注意事项

无

2.3 系统缓存参数

2.3.1 data_buff_mem

参数名称

data_buff_mem

参数解释

设置数据缓冲区内存大小（单位：MB）。

在系统服务启动时，会依据该参数给定的内存大小从操作系统分配内存，再按 block_size 的大小为最小管理单元对这块完整的内存进行划分，划分后单元内存由 buffer 管理器分组管理（组数由 cata_parti_num 控制）。

数据块缓存结合 checkpoint 机制，减少数据库对磁盘数据的读写次数，在有限磁盘 I/O 能力上提供更高性能。

默认值

256

取值范围

[8,2147483647]

访问权限

R/W

是否全局参数

是

修改生效方式

data_buff_mem 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 data_buff_mem 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW data_buff_mem
```

注意事项

最优值建议设置为物理内存的 60%。

2.3.2 ena_share_sga

参数名称

ena_share_sga

参数解释

设置是否允许缓冲区与 SGA 共享，当数据缓冲区内存不足时使用 system_sga_mem 内存。此共享设计定义为借用，借用后当发生 SGA 不足时（如大数据量排序），按需从数据缓冲区催还一部分供当前计算使用。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

ena_share_sga 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 ena_share_sga 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW ena_share_sga
```

注意事项

无

2.3.3 hash_parti_num

参数名称

hash_parti_num

参数解释

哈希运算在内存不够时转入分组处理的分组数。

默认值

16

取值范围

[16,256]

访问权限

R/W

是否全局参数

是

修改生效方式

hash_parti_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 hash_parti_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET hash_parti_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW hash_parti_num
```

注意事项

无

2.3.4 max_hash_mem

参数名称

max_hash_mem

参数解释

允许单个哈希节点使用的最大内存量（单位：MB）。

SGA 内存（system_sga_mem）为共享资源，所有排序、哈希、集合等都会使用，为避免单个任务将内存资源耗尽，其他任务无法成功执行，系统通过该参数控制单次计算的最大内存开销，保证数据库系统利用有限资源最大化对外服务能力。

默认值

32

取值范围

[32,1048576]

访问权限

R/W

是否全局参数

是

修改生效方式

max_hash_mem 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 max_hash_mem 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_hash_mem
```

注意事项

无

2.3.5 max_hash_size

参数名称

max_hash_size

参数解释

最大哈希表槽数。

系统内散列计算时允许的最大哈希槽数，哈希表存在内存开销，且和槽数成正比。

默认值

3000000

取值范围

[1048576,1073741824]

访问权限

R/W

是否全局参数

是

修改生效方式

max_hash_size 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_hash_size 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_hash_size TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_hash_size
```

注意事项

一个哈希槽消耗哈希表初始化内存 8Byte，1 亿哈希槽需消耗内存 800MB。请合理设置该参数，槽数过大可能导致系统内存耗尽，哈希表初始化失败等问题。

2.3.6 max_malloc_once

参数名称

max_malloc_once

参数解释

设置最大单次分配内存块大小（单位：MB）。

通过此参数限制数据库运行期间单次向操作系统申请的最大内存，若超过该阈值则抛出异常。

默认值

512

取值范围

[64,4095]

访问权限

R/W

是否全局参数

是

修改生效方式

max_malloc_once 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_malloc_once 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_malloc_once TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_malloc_once
```

注意事项

无

2.3.7 max_task_mem

参数名称

max_task_mem

参数解释

设置任务线程运行最大内存（单位：MB）。

任务线程有独立的内存区，此内存按需动态向操作系统申请和释放，为避免操作系统内存资源耗尽，通过此参数限制单个线程的内存上限，超过该值时抛出错误，终止任务同时释放内存。

默认值

0

取值范围

[0,1048576]

取值为 0 时，表示不限制。

访问权限

R/W

是否全局参数

是

修改生效方式

max_task_mem 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_task_mem 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_task_mem TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_task_mem
```

注意事项

无

2.3.8 swap_buff_mem

参数名称

swap_buff_mem

参数解释

设置交换缓冲区内存大小，即交换缓存块（swap）初始额定总内存（单位:MB）。

Swap 内存用于排序、哈希连接（Hash Join）、聚合等运算的文件转出和读取，块大小为 256KB。与 data_buff_mem 不同，swap_buff_mem 没有分组设计，当 swap 内存不足时，后续任务堆 swap 内存的请求阻塞等待，直到 swap 内存回收重用。

以哈希连接为例，当 system_sga_mem 不足或 max_hash_mem 不足后，将使用临时表空间的方式启动文件转出，转出逻辑文件组数由 hash_parti_num 决定，将使用

“hash_parti_num*2” 个交换缓存块，计算方式为 “swap_buff_mem ≥ hash_parti_num*2*256KB”，例如：“hash_parti_num=16” 需要 “16×2×256KB=8MB” 的交换缓冲区内存。

默认值

64

取值范围

[64,1048576]

访问权限

R/W

是否全局参数

是

修改生效方式

swap_buff_mem 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 swap_buff_mem 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW swap_buff_mem
```

注意事项

该参数最小值必须满足单个查询所需最大块数，否则查询将无法执行。

2.3.9 system_sga_mem

参数名称

system_sga_mem

参数解释

系统全局区（SGA，System Global Area）内存（单位：MB）。

SGA 通常用于连接、集合、排序、去重等运算的中间结果缓存，内存单元大小受参数 ena_share_sga 影响。

- 若 ena_share_sga 为 true，则 SGA 与数据缓存块的最小内存单元相同。
- 若 ena_share_sga 为 false，则 SGA 的最小内存单元为 256KB。

默认值

128

取值范围

[32,2147483647]

访问权限

R/W

是否全局参数

是

修改生效方式

system_sga_mem 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 system_sga_mem 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW system_sga_mem
```

注意事项

最优值建议设置为物理内存的 20%。

2.3.10 xlog_buff_mem

参数名称

xlog_buff_mem

参数解释

设置事务日志缓冲区尺度，即 REDO 日志缓冲区额定总内存（单位：MB）。

缓存待写入 REDO 文件的事务日志，缓存内容由 REDO 写盘线程做落盘处理，落盘后缓存可回收重用。事务日志为实时写盘，此内存也是实时回收，仅当磁盘顺序写入能力小于事务日志写入量时，会发生事务日志写入等待阻塞。

默认值

32

取值范围

[1,1024]

访问权限

R/W

是否全局参数

是

修改生效方式

xlog_buff_mem 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 system_sga_mem 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW xlog_buff_mem
```

注意事项

无

2.4 SQL 引擎参数

2.4.1 auto_eje_cast

参数名称

auto_eje_cast

参数解释

控制自动生成弹射式执行计划的最小查询开销值。

- 如果 auto_use_eje = true 且查询计划的总代价 > auto_eje_cast，则启用并行执行。
- 如果 auto_use_eje = true 但查询计划的总代价 ≤ auto_eje_cast，则不启用并行执行。
- 如果 auto_use_eje = false，则无论查询计划的代价如何，都不会自动生成弹射式执行计划。

默认值

10000

取值范围

[-1,2147483647]

取值为-1 时，表示禁用该参数。

访问权限

R/W

是否全局参数

是

修改生效方式

auto_eje_cast 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 auto_eje_cast 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET auto_eje_cast TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW auto_eje_cast
```

注意事项

无

2.4.2 auto_eje_parallel

参数名称

auto_eje_parallel

参数解释

控制自动生成弹射式执行计划时的并行度。

当 auto_use_eje = true 且查询计划的总代价 > auto_eje_cast 时，数据库会根据 auto_eje_parallel 设置的值决定并行执行的线程数或任务数。

默认值

4

取值范围

[1,1024]

访问权限

R/W

是否全局参数

是

修改生效方式

auto_eje_parallel 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 auto_eje_parallel 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET auto_eje_parallel TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW auto_eje_parallel
```

注意事项

无

2.4.3 auto_use_eje

参数名称

auto_use_eje

参数解释

控制是否允许自动生成弹射式执行计划。

在处理查询请求的路径规划时，若未显式指定并行执行（PARALLEL）且当前查询又符合并行执行，则可根据该参数的值决定是否使用并行执行。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

auto_use_eje 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 auto_use_eje 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET auto_use_eje TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW auto_use_eje
```

注意事项

无

2.4.4 cata_case_sensitive

参数名称

cata_case_sensitive

参数解释

控制系统字典（系统表）中对象名（如表、列）的存储和匹配是否区分大小写。

- true: 严格区分大小写，与写入系统表的格式完全一致。
- false: 不区分大小写，统一按大写处理。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

cata_case_sensitive 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 cata_case_sensitive 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW cata_case_sensitive
```

注意事项

- V12.5 及之前版本开表逻辑（基于 table_name）
 - 相关参数 enable_sys_name_idx：
 - 该参数用于控制按对象名查询系统表时是否使用索引，默认值为 false。
 - 如果 enable_sys_name_idx 设置为 true，则使用索引扫描，且索引严格区分大小写（不受 cata_case_sensitive 参数的影响）。
 - 仅在节点上首次访问时才会访问系统表，后续访问会利用对象缓存。
 - 输入标识符处理：
 - 输入标识符的处理方式与数据库的兼容模式相关。PostgreSQL 兼容模式：默认将标识符转换为小写。Oracle/Xugu 兼容模式：默认将标识符转换为大写。
 - 使用反引号 ` 可以强制保留标识符的原始大小写。
 - 对象缓存规则：
 - cata_case_sensitive=false：缓存强制大写。
 - cata_case_sensitive=true：缓存与系统表存储的格式一致。
- V12.6 及以上版本开表逻辑（基于 table_id）

开表过程改为通过表 ID (table_id) 进行。

2.4.5 check_unique_mode

参数名称

check_unique_mode

参数解释

控制数据库在插入 (INSERT) 和更新 (UPDATE) 操作后，是否对唯一性约束进行预检测。

- 1：不预检测
- 2：预检测

默认值

2

取值范围

[1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

check_unique_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 check_unique_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET check_unique_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW check_unique_mode
```

注意事项

无

2.4.6 ddl_timeout

参数名称

ddl_timeout

参数解释

设置 DDL 操作的加锁超时时间。

默认值

2000

取值范围

[100,600000]

访问权限

R/W

是否全局参数

是

修改生效方式

ddl_timeout 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 ddl_timeout 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET ddl_timeout TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW ddl_timeout
```

注意事项

无

2.4.7 def_empty_str_as_null

参数名称

def_empty_str_as_null

参数解释

控制是否将长度为 0 的字符串（即空字符串）视为 NULL 值。

- true: 空字符串视为 NULL。

```
SQL> SELECT CASE WHEN '' IS NULL THEN 'TRUE' ELSE 'FALSE' END AS  
is_null_test;  
  
IS_NULL_TEST |  
-----  
TRUE |
```

- false: 空字符串不视为 NULL。

```
SQL> SELECT CASE WHEN '' IS NULL THEN 'TRUE' ELSE 'FALSE' END AS  
is_null_test;  
  
IS_NULL_TEST |  
-----  
FALSE |
```

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

def_empty_str_as_null 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_empty_str_as_null 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_empty_str_as_null TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_empty_str_as_null
```

注意事项

该参数的影响范围当前版本只限于查询分组计算、部分表达式和字符数据类型系统函数，详细信息如表所示。

类型	名称
系统函数	CONCAT_WS、CONCAT、SUBSTRING_INDEX、LPAD、RPAD、INSTR、POSITION、LOCATE、STRPOS、TAILING、RIGHT、HEADING、LEFT、SUBSTR、SUBSTRING、MID、OVERLAY、RTRIM、LTRIM
表达式	IS NULL、IS NOT NULL

2.4.8 def_optimize_mode

参数名称

def_optimize_mode

参数解释

SQL 默认优化模式。

- 0: all rows
- 1: first rows

默认值

0

取值范围

[0|1]

访问权限

R/W

是否全局参数

是

修改生效方式

def_optimize_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_optimize_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_optimize_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_optimize_mode
```

注意事项

无

2.4.9 delay_check_unique

参数名称

delay_check_unique

参数解释

是否将唯一值检测延迟到事务结束时进行。

针对 UPDATE 请求的处理，是否将唯一值检测延迟到事务结束时进行开关。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

delay_check_unique 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 delay_check_unique 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET delay_check_unique TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW delay_check_unique
```

注意事项

无

2.4.10 enable_eje_big_block

参数名称

enable_eje_big_block

参数解释

控制弹射式执行计划在堆扫描操作中是否允许使用大块读策略。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_eje_big_block 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_eje_big_block 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_eje_big_block TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_eje_big_block
```

注意事项

无

2.4.11 enable_find_synonym

参数名称

enable_find_synonym

参数解释

控制是否允许在查询执行过程中从磁盘元数据中查找同义词。

同义词为数据库对象（如表、视图）提供一个简化的名称或别名，简化查询语句。

- true: 允许数据库在内存缓存中未找到同义词时，从磁盘元数据中查找。
- false: 禁止从磁盘查找，若同义词未被缓存（如创建同义词后还未使用，此时数据库发生重启），使用同义词进行查询会报错。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_find_synonym 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_find_synonym 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_find_synonym TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_find_synonym
```

注意事项

无

2.4.12 enable_histogram

参数名称

enable_histogram

参数解释

选择率计算是否启用直方图优化。

- true: 启用
- false: 不启用

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_histogram 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_histogram 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_histogram TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_histogram
```

注意事项

无

2.4.13 enable_stream_import

参数名称

enable_stream_import

参数解释

控制是否启用流式数据导入功能。

流式数据导入是一种高效的数据加载机制，适用于大规模数据的实时或批量导入。数据在传输过程中逐步写入数据库，而不是一次性加载所有数据后再写入。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_stream_import 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_stream_import 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_stream_import TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_stream_import
```

注意事项

无

2.4.14 enable_sys_name_idx

参数名称

enable_sys_name_idx

参数解释

控制是否启用系统对象名索引（表名/视图名等）。

系统对象名索引启动开关。索引裁剪和过滤时，严格区分大小，在忽略大小写的情况下，会存在无法准确定位数据情况（例如：创建仅大小写不同的同名表的情况），此时通过此参数控制是否使用对象名索引，通过使用顺序扫描保证能够准确定位数据。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_sys_name_idx 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_sys_name_idx 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_stream_import TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_stream_import
```

注意事项

无

2.4.15 idx_delay_del_limit

参数名称

idx_delay_del_limit

参数解释

限制事务中非唯一索引延迟删除的次数。

当事务中累积的非唯一索引延迟删除次数超过该阈值时，系统会中断唯一索引的立即删除操作，并将其延迟到事务提交时执行。这种机制可以优化事务提交性能，减少因索引删除耗时过长而导致的锁持有时间，从而提升并发性能。

默认值

0

取值范围

[0,10000000]

取值为 0 时，表示不限制。

访问权限

R/W

是否全局参数

是

修改生效方式

idx_delay_del_limit 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 idx_delay_del_limit 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。

3. 重新启动数据库服务。

4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW idx_delay_del_limit
```

注意事项

没有删除的索引数据对数据查询的正确性没有影响，但会暂用存储空间，可通过重建索引处理。

为了使事务快速结束，可以将该参数设置为较小值，事务正常结束后，还原为默认值，避免产生大量无效索引数据暂用存储空间。

2.4.16 iscan_ini_cost

参数名称

iscan_ini_cost

参数解释

设置索引扫描的初始化代价。

数据库的查询优化器会为 SQL 查询生成多个可能的执行计划（如全表扫描、索引扫描、哈希连接等）。每个计划都有一个总代价，用于衡量其执行的资源消耗（如 CPU、I/O、内存等）。

该用于判断索引扫描与其它扫描方式之间的成本差异，从而选择最优路径。

默认值

200

取值范围

[-10000,2147483647]

访问权限

R/W

是否全局参数

是

修改生效方式

iscan_ini_cost 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 iscan_ini_cost 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET iscan_ini_cost TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW iscan_ini_cost
```

注意事项

无

2.4.17 max_cursor_num

参数名称

max_cursor_num

参数解释

单个连接下允许使用的最大显示游标数量。

默认值

100

取值范围

[0,10000]

访问权限

R/W

是否全局参数

是

修改生效方式

max_cursor_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_cursor_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_cursor_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_cursor_num
```

注意事项

超过阈值报错 Error: [E19243] Cursor 个数超过系统最大限制数%d。

2.4.18 max_loop_num

参数名称

max_loop_num

参数解释

存储过程最大循环次数。

该参数用于限制存储过程中循环体的执行次数，当循环体的实际执行次数超过设定的阈值时，系统将自动抛出异常以终止循环，从而保障程序运行的稳定性和资源的安全性。

默认值

100000

取值范围

[0,2147483647]

取值为 0 时，表示无限制。

访问权限

R/W

是否全局参数

是

修改生效方式

max_loop_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_loop_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_loop_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_loop_num
```

注意事项

超过阈值报错 Error: [E21092 L6 C8] 超过存储过程最大循环次数%d。

2.4.19 max_prepare_num

参数名称

max_prepare_num

参数解释

单个连接下允许预处理语句的最大数量。

默认值

100000

取值范围

[0,2147483647]

取值为 0 时，表示无限制。

访问权限

R/W

是否全局参数

是

修改生效方式

max_prepare_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_prepare_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_prepare_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_prepare_num
```

注意事项

超过阈值报错 Error: [E19218] PreparedStatement 个数超过限制%d。

2.4.20 para_eje_seqscan_num

参数名称

para_eje_seqscan_num

参数解释

控制弹射器堆扫描器的并行执行路数。

默认值

2

取值范围

[1,128]

访问权限

R/W

是否全局参数

是

修改生效方式

para_eje_seqscan_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 para_eje_seqscan_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET para_eje_seqscan_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW para_eje_seqscan_num
```

注意事项

无

2.4.21 prepare_param_num

参数名称

prepare_param_num

参数解释

单个连接下允许 PREPARE 的最大参数个数。

控制通过预处理方式执行的命令的占位符个数。例如：“INSERT INT t1(c1,c2,c3) VALUES(?,?,?).....(?,?,?)(?,?,?)”。控制命令中的‘?’的个数。

默认值

2048

取值范围

[1,32767]

访问权限

R/W

是否全局参数

是

修改生效方式

prepare_param_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 prepare_param_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET prepare_param_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW prepare_param_num
```

注意事项

无

2.4.22 prepare_reuse

参数名称

prepare_reuse

参数解释

是否启用预处理重用。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

prepare_reuse 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 prepare_reuse 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET prepare_reuse TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW prepare_reuse
```

注意事项

关闭 prepare_reuse 功能后，SQL 在连接会话上将不会再持有对象锁，降级为事务级持有，事务提交后就可进行 DDL 操作，但 PREPARE 将失去一次解析多次执行的特性，每次调用都要进行 SQL 解析和路径规划，可能对性能产生影响。

2.4.23 proc_reuse_cnt

参数名称

proc_reuse_cnt

参数解释

过程函数中 SQL 重用多少次后重新编译。

在存储过程或函数中，静态 SQL 语句首次执行时会被编译，生成执行计划并缓存以供重复使用，从而提升执行效率。随着依赖表数据的变化（如大量插入、更新或删除），原有执行计划

可能失效，导致性能下降。当静态 SQL 的重用次数达到 `proc_reuse_cnt` 阈值时，数据库会强制重新编译该 SQL，基于最新数据信息生成更优的执行计划，确保查询性能始终高效。

默认值

0

取值范围

[0,10000]

取值为 0 时，表示不重新编译。

访问权限

R/W

是否全局参数

是

修改生效方式

`proc_reuse_cnt` 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 `proc_reuse_cnt` 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET proc_reuse_cnt TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW proc_reuse_cnt
```

注意事项

无

2.4.24 select_table_num

参数名称

`select_table_num`

参数解释

查询语句允许的最大表个数。

一次查询请求中允许出现表的最大个数。

默认值

12

取值范围

[1,64]

访问权限

R/W

是否全局参数

是

修改生效方式

select_table_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 select_table_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET select_table_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW select_table_num
```

注意事项

超过阈值报错 Error: [E19241] 连接表或子查询个数超过%d，请用 join 关键字重写。

2.4.25 seqscan_skip_err

参数名称

seqscan_skip_err

参数解释

控制顺序扫描在遇到 E19198 错误（数据存贮体正在进行切换操作）时是否跳过此类错误并继续执行。

- true：顺序扫描在遇到 E19198 错误时会尝试跳过该存储体，继续扫描其他可用的存储体，最终返回已获取的数据。
- false：遇到 E19198 错误则直接终止扫描，并抛出错误。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

seqscan_skip_err 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 seqscan_skip_err 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET seqscan_skip_err TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW seqscan_skip_err
```

注意事项

无

2.4.26 str_trunc_warning

参数名称

str_trunc_warning

参数解释

字符串超长时是否使用截断并产生告警。

- true: 当字符串长度超出目标字段的最大长度时，数据库会自动截断字符串以适配字段长度。操作继续执行，并向客户端返回一条警告信息 Warning: [E17090] 数据异常/字符串超长。
- false: 当字符串长度超出目标字段的最大长度时，数据库终止当前操作，抛出错误 Error: [E19230] 字段 (col_no:%d,col_name:%s) 值错误 (%s)。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

str_trunc_warning 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 str_trunc_warning 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET str_trunc_warning TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW str_trunc_warning
```

注意事项

无

2.4.27 supple_prepare_sql

参数名称

supple_prepare_sql

参数解释

补齐预处理定义语句。

数据库的 SQL 请求由任务线程负责执行，线程中会记录当前正在执行的 SQL 语句。对于经过 PREPARE 的语句，在 sys_thd_session 系统表中，线程仅记录其 PREPARE 后的名称，而无法直接查看该名称对应的原始 SQL 语句。开启 supple_prepare_sql，将预编译前的原始 SQL 语句与预编译名称进行关联，从而明确当前线程实际执行的 SQL 语句。

- true: sys_thd_session 系统表 SQL 字段记录?pre_1::PREPARE pre_1 AS SELECT * FROM sys_thd_session;。
- false: sys_thd_session 系统表 SQL 字段记录?pre_1;。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

str_trunc_warning 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 str_trunc_warning 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET str_trunc_warning TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW str_trunc_warning
```

注意事项

无

2.4.28 support_global_tab

参数名称

support_global_tab

参数解释

控制是否允许创建和使用全局临时表。

全局临时表是一种特殊类型的表，其数据在会话或事务结束后自动删除，全局临时表的定义对所有会话可见，而数据仅对创建它的会话可见。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

support_global_tab 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 support_global_tab 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET support_global_tab TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW support_global_tab
```

注意事项

support_global_tab 为 false 时，创建全局临时表报错 Error: [E5073] 不能更改临时表%s。

2.4.29 tab_rebuild_limit

参数名称

tab_rebuild_limit

参数解释

DDL 表重整或复制行数限制（单位：百万）。

在修改表结构时，满足以下条件之一的情况下需要对表数据进行重整：

- 新增列并指定默认值：为原有表结构新增列，并且该列定义了默认值。
- 字段类型变更：
 - 修改字段类型为变长字段。
 - 修改数值类型（NUMERIC）的标度或精度。

默认值

10

取值范围

[0,10000]

取值为 0 时，表示不限制。

访问权限

R/W

是否全局参数

是

修改生效方式

tab_rebuild_limit 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 tab_rebuild_limit 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET tab_rebuild_limit TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW tab_rebuild_limit
```

注意事项

执行 DDL 修改表结构操作后，如果需要对表数据进行重整，而现有数据行数超过该参数设定的阈值，则会抛出异常，需由数据库管理员确认。通常情况下，数据重整是一项复杂的操作，尤其是当表中数据量较大时，可能会显著增加 CPU 使用率，进而对其他业务造成影响。设置合理的阈值有助于平衡系统性能与操作安全性。

2.4.30 use_index_order

参数名称

use_index_order

参数解释

查询语句中若存在 ORDER BY 子句，该参数控制排序算法是否使用索引序。

索引序的核心作用是通过遍历索引的物理顺序，直接满足 ORDER BY 的排序需求，从而避免显式的排序操作。例如：假设有一个按 order_date 升序排列的索引，那么查询 SELECT * FROM orders ORDER BY order_date 可以直接按索引的顺序读取数据，无需额外排序。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

use_index_order 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 use_index_order 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET use_index_order TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW use_index_order
```

注意事项

无

2.5 兼容性参数

2.5.1 def_compatible_mode

参数名称

def_compatible_mode

参数解释

设置连接会话参数使用的兼容模式。

- NONE：默认值，不启用任何特定的兼容模式。
- ORACLE：启用 Oracle 兼容模式。
- MYSQL：启用 MySQL 兼容模式。
- POSTGRESQL：启用 PostgreSQL 兼容模式。

默认值

NONE

取值范围

[NONE|ORACLE|MYSQL|POSTGRESQL]

ORACLE

- 标识符统一转为大写。
- 兼容聚合函数 VAR_POP/VARP/VAR/VARIANCE/STDEVP/STDDEV_POP/STDEV 计算行为。

- 兼容存储过程创建方式，支持强制创建存储过程，即使编译失败也不报错。
- 兼容 COALESCE 的计算行为，忽略空值。
- 兼容 LEAST 的计算行为，若参数列表中包含空值，则一定返 NULL。
- 兼容 GREATEST 的计算行为，若参数列表中包含空值，则一定返 NULL。
- 兼容 DATE 数据类型。

MYSQL

- 兼容二进制数据与字符串数据类型之间相互转换的处理方式，相关函数 BITTOCHAR。
- 兼容三角函数 ACOS 的处理方式，当 ACOS 的参数值不在 [-1,1] 之间，则返回 NULL。
- 兼容字符串转换为浮点数的处理方式，转换时异常字符不报错，尝试进行截断处理。
- 兼容 TRUNCATE 对象数据后，重置应用在表上的自增序列值生成器当前值为 1。
- 兼容非字段表达式的输出列名处理方式，若无别名则展示表达式。
- 兼容部分日期格式数据的转换处理。

POSTGRESQL

- 标识符统一转为小写。
- 兼容浮点数向整型转换的处理方式，若四舍五入进位触及边界值，小数部分会截断。
- 兼容十六进制数中字母以小写形式表示，相关函数 HEX。

访问权限

R/W

是否全局参数

是

修改生效方式

def_compatible_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_compatible_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_compatible_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_compatible_mode
```

注意事项

无

2.5.2 def_group_by_mode

参数名称

def_group_by_mode

参数解释

GROUPBY 语句模式。

- 0: 标准模式，遵循 SQL 标准，SELECT 或 ORDER BY 中出现的非聚合列必须在 GROUP BY 子句中显式声明，未显式声明会报错 Error: [E19045] 排序项%s 无法映射到分组。
- 1: MySQL 方言写法，允许 ORDER BY 使用 GROUP BY 未包含的列（即使该列未聚合），但输出结果可能不稳定。

默认值

0

取值范围

[0|1]

访问权限

R/W

是否全局参数

是

修改生效方式

def_group_by_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_group_by_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_group_by_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_group_by_mode
```

注意事项

此方法为非标准方法，MySQL 官方自 5.7.5 开始已经默认禁止，不建议使用。此方法的非分组列输出不确定，可能并不是用户想要的结果，并且因为结果集排序是在选择之后，ORDER BY 不会改变最终的输出值。

```
-- 创建员工表，包含部门、姓名、工资
SQL> CREATE TABLE employees (
department VARCHAR(50),
employee_name VARCHAR(50),
salary DECIMAL(10, 2)
);

-- 插入测试数据
SQL> INSERT INTO employees VALUES
('Sales', 'Alice', 5000),
('Sales', 'Bob', 6000),
('HR', 'Charlie', 4500),
('HR', 'David', 5500),
('IT', 'Eve', 7000);

-- 设置为标准模式
SQL> SET def_group_by_mode TO 0;

-- 尝试按部门分组，但SELECT和ORDER BY包含非聚合列
SQL> SELECT department, employee_name, AVG(salary) AS avg_salary
FROM employees GROUP BY department ORDER BY salary DESC; --
salary未在GROUP BY中，也未聚合

Error: [E19045] 排序项EMPLOYEES.SALARY无法映射到分组

-- 设置为MySQL方言模式
SQL> SET def_group_by_mode TO 1;

-- 相同的查询，允许执行但结果不稳定
SQL> SELECT department, employee_name, AVG(salary) AS avg_salary
FROM employees GROUP BY department ORDER BY salary DESC;

DEPARTMENT | EMPLOYEE_NAME | AVG_SALARY |
-----
IT| Eve| 7000|
Sales| Alice| 5500| -- Alice的工资是5000，但随机选取了Alice或Bob
HR| Charlie| 5000| -- Charlie的工资是4500，但随机选取了Charlie或
David
```

2.5.3 def_identity_mode

参数名称

def_identity_mode

参数解释

控制自增列（identity）的插入值填充模式，以兼容 MySQL 的自增列处理方式，支持连接会话级参数 IDENTITY_MODE。

def_identity_mode 参数值	对应 IDENTITY_MODE 参数值	参数功能
0	<ul style="list-style-type: none"> • DEFAULT • DEFAULT_IDENTITY_MODE 	<ul style="list-style-type: none"> • 未显式指定自增列的值，插入自增值 • 显式指定自增列的值为 NULL，报错 • 显式指定自增列的值为非 NULL 数值，作为数据插入进表（自增列的值具有唯一值约束，多次插入相同值会报错违反唯一值约束）
1	NULL_AS_AUTO_INCREMENT	<ul style="list-style-type: none"> • 未显式指定自增列的值，插入自增值 • 显式指定自增列的值为 NULL，插入自增值 • 显式指定自增列的值为非 NULL 数值，作为数据插入进表（自增列的值具有唯一值约束，多次插入相同值会报错违反唯一值约束）
2	ZERO_AS_AUTO_INCREMENT	<ul style="list-style-type: none"> • 未显式指定自增列的值，插入自增值 • 显式指定自增列的值为 NULL 或 0，插入自增值 • 显式指定自增列的值为非 NULL 非 0 数值，作为数据插入进表（自增列的值具有唯一值约束，多次插入相同值会报错违反唯一值约束）

默认值

0

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

def_identity_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_identity_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_identity_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_identity_mode
```

注意事项

def_identity_mode 为 2 时，如果插入报错仍需消耗序列值。

2.5.4 use_old_product_name

参数名称

use_old_product_name

参数解释

控制数据库系统在展示产品名称时是否使用旧名称。

- true：显示旧的产品名称 XuGu SQL Server 12.0.0。
- false：显示新的产品名称 XuguDB 12.0.0。

在数据库中执行 SHOW VERSION 查看产品名称。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

use_old_product_name 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 `use_old_product_name` 参数值，根据实际需求在取值范围内选择合适的 `value` 值。

```
SQL> SET use_old_product_name TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW use_old_product_name
```

注意事项

无

2.6 存储管理函数

2.6.1 auto_extend_dfile

参数名称

auto_extend_dfile

参数解释

控制是否允许自动扩展数据表空间文件，当表空间使用率接近上限时，系统会动态增加文件容量以容纳新数据。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

auto_extend_dfile 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 `auto_extend_dfile` 参数值，根据实际需求在取值范围内选择合适的 `value` 值。

```
SQL> SET auto_extend_dfile TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW auto_extend_dfile
```

注意事项

无

2.6.2 block_pctfree

参数名称

block_pctfree

参数解释

默认新建表存储块空闲度百分比。

默认值

15

取值范围

[0,99]

访问权限

R/W

是否全局参数

是

修改生效方式

block_pctfree 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 block_pctfree 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET block_pctfree TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW block_pctfree
```

注意事项

无

2.6.3 block_size

参数名称

block_size

参数解释

设置存储系统中数据块大小（单位：Byte）。

默认值

8192

取值范围

[2048|4096|8192|16384]

访问权限

R/W

是否全局参数

是

修改生效方式

block_size 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 block_size 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET block_size TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW block_size
```

注意事项

无

2.6.4 data_file_append_mode

参数名称

data_file_append_mode

参数解释

定义数据文件扩展时的写入行为模式，指定文件扩展的方式。

- 0：数据文件新扩展的空间部分以数据块单元尺寸进行逐步写实的方式扩容。
- 1：当数据文件需要扩容的增量超过额定容量（128MB）时，文件会直接一次性扩容至所需大小；若增量未超过额定容量（128MB）时，则采用与 data_file_append_mode = 0 相同

的方式，逐步按数据块单元扩容。

默认值

0

取值范围

[0|1]

访问权限

R/W

是否全局参数

是

修改生效方式

data_file_append_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 data_file_append_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET data_file_append_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW data_file_append_mode
```

注意事项

无

2.6.5 def_data_space_size

参数名称

def_data_space_size

参数解释

默认数据表空间的初始尺度（单位：MB）。

默认值

256

取值范围

[64,2097152]

访问权限

R/W

是否全局参数

是

修改生效方式

def_data_space_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 def_data_space_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_data_space_size
```

注意事项

无

2.6.6 def_file_step_size

参数名称

def_file_step_size

参数解释

定义表空间文件自动扩展的默认步长（单位：MB），用于在未显式指定扩展容量时，按固定增量动态扩容表空间。

默认值

64

取值范围

[64,65536]

访问权限

R/W

是否全局参数

是

修改生效方式

def_file_step_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 def_file_step_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_file_step_size
```

注意事项

无

2.6.7 def_redo_file_size

参数名称

def_redo_file_size

参数解释

默认重做日志文件尺度（单位：MB）。

默认值

500

取值范围

[32,2097152]

访问权限

R/W

是否全局参数

是

修改生效方式

def_redo_file_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 def_redo_file_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_redo_file_size
```

注意事项

无

2.6.8 def_temp_space_size

参数名称

def_temp_space_size

参数解释

默认临时表空间的初始尺度（单位：MB）。

默认值

256

取值范围

[64,2097152]

访问权限

R/W

是否全局参数

是

修改生效方式

def_temp_space_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 def_temp_space_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_temp_space_size
```

注意事项

无

2.6.9 def_undo_space_size

参数名称

def_undo_space_size

参数解释

默认回滚表空间的初始尺度（单位：MB）。

默认值

256

取值范围

[64,2097152]

访问权限

R/W

是否全局参数

是

修改生效方式

def_undo_space_size 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 def_undo_space_size 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_undo_space_size
```

注意事项

无

2.6.10 default_copy_num

参数名称

default_copy_num

参数解释

设置分布式集群中的存储版本数量。最多可以存储 3 个版本，其中 1 个主版本，2 个副版本。

默认值

3

取值范围

[1|2|3]

访问权限

R/W

是否全局参数

是

修改生效方式

default_copy_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 default_copy_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET default_copy_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW default_copy_num
```

注意事项

无

2.6.11 enable_read_copy2

参数名称

enable_read_copy2

参数解释

设置是否允许在查询时从存储副本读取数据。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_read_copy2 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_read_copy2 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_read_copy2 TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_read_copy2
```

注意事项

无

2.6.12 enable_store_migrate

参数名称

enable_store_migrate

参数解释

控制是否允许系统自动迁移存储单元，用于动态优化存储负载。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_store_migrate 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_store_migrate 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_store_migrate TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_store_migrate
```

注意事项

无

2.6.13 index_preload_cnt

参数名称

index_preload_cnt

参数解释

启用索引存储预加载最小个数。

当分区表的分区数量超过此值时，系统会预加载索引数据到内存，以加速查询性能。

默认值

10000

取值范围

[0,2147483647]

取值为 0 时，表示不启用预加载。

访问权限

R/W

是否全局参数

是

修改生效方式

index_preload_cnt 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 index_preload_cnt 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET index_preload_cnt TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW index_preload_cnt
```

注意事项

无

2.6.14 init_data_space_num

参数名称

init_data_space_num

参数解释

初始化系统时数据表空间个数。

默认值

4

取值范围

[1,32]

访问权限

R

是否全局参数

是

修改生效方式

init_data_space_num 参数专为系统首次初始化设计，初始化完成后将不可修改。

注意事项

无

2.6.15 init_temp_space_num

参数名称

init_temp_space_num

参数解释

初始化系统时临时表空间个数。

默认值

2

取值范围

[1,32]

访问权限

R

是否全局参数

是

修改生效方式

init_temp_space_num 参数专为系统首次初始化设计，初始化完成后将不可修改。

注意事项

无

2.6.16 init_undo_space_num

参数名称

init_undo_space_num

参数解释

初始化系统时回滚表空间个数。

默认值

2

取值范围

[1,32]

访问权限

R

是否全局参数

是

修改生效方式

init_undo_space_num 参数专为系统首次初始化设计，初始化完成后将不可修改。

注意事项

无

2.6.17 ioerr_report_mode

参数名称

ioerr_report_mode

参数解释

数据文件读写错时处理模式。

- 0：以数据块为处理单位，仅标记异常数据块所属的数据片下的所有块为介质错误；若存在 OV 存储，则标记整个表空间为媒体介质错误。
- 1：以数据块为处理单位，直接标记异常数据块所属的整个表空间下所有数据块为媒体介质错误。

默认值

1

取值范围

[0|1]

访问权限

R/W

是否全局参数

是

修改生效方式

ioerr_report_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 ioerr_report_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET ioerr_report_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW ioerr_report_mode
```

注意事项

无

2.6.18 max_file_size

参数名称

max_file_size

参数解释

定义数据表空间文件允许增长的最大容量（单位：MB），用于限制数据库数据文件的物理存储上限。

默认值

-1

取值范围

[-1,2147483647]

取值为-1时，表示不限制数据表空间文件额定总容量。

访问权限

R

是否全局参数

是

修改生效方式

max_file_size 参数专为系统首次初始化设计，初始化完成后将不可修改。

注意事项

无

2.6.19 max_hotspot_num

参数名称

max_hotspot_num

参数解释

设置系统允许的最大并发插入热点数量，用于控制写入操作在分布式存储中的负载均衡，防止因数据分布不均导致性能瓶颈。

默认值

256

取值范围

[1,1024]

访问权限

R/W

是否全局参数

是

修改生效方式

max_hotspot_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_hotspot_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_hotspot_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_hotspot_num
```

注意事项

无

2.6.20 max_temp_space_size

参数名称

max_temp_space_size

参数解释

设置临时表空间允许使用的最大存储容量（单位：MB），用于限制数据库临时操作的资源占用。

默认值

-1

取值范围

[-1,2147483647]

取值为-1 时，表示不限制临时表空间文件额定总容量。

访问权限

R/W

是否全局参数

是

修改生效方式

max_temp_space_size 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_temp_space_size 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_temp_space_size TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_temp_space_size
```

注意事项

无

2.6.21 ov_reuse

参数名称

ov_reuse

参数解释

控制是否允许 OV 存储块的重用。

只有允许 OV 存储重用的存储块才允许被释放回收。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

ov_reuse 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 ov_reuse 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET ov_reuse TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW ov_reuse
```

注意事项

无

2.6.22 safely_copy_num

参数名称

safely_copy_num

参数解释

设置存储系统中的安全副本个数。

当现有有效副本数小于该值时，系统会另建新副本。

默认值

2

取值范围

[1|2|3]

访问权限

R/W

是否全局参数

是

修改生效方式

safely_copy_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 safely_copy_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET safely_copy_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW safely_copy_num
```

注意事项

无

2.6.23 size_per_chunk

参数名称

size_per_chunk

参数解释

设置存储系统中单个存储片（Chunk）的数据容量（单位：MB）。

存储片可视为管理数据块的容器。

默认值

8

取值范围

[1,64]

访问权限

R

是否全局参数

是

修改生效方式

size_per_chunk 参数专为系统首次初始化设计，初始化完成后将不可修改。

注意事项

无

2.6.24 store_drop_delay

参数名称

store_drop_delay

参数解释

局部存储的延迟删除时间（单位：分）。

默认值

16

取值范围

[2,118]

访问权限

R/W

是否全局参数

是

修改生效方式

store_drop_delay 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 store_drop_delay 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET store_drop_delay TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW store_drop_delay
```

注意事项

无

2.6.25 store_maint_span

参数名称

store_maint_span

参数解释

定义存储系统在执行数据版本修复或迁移时，两次连续操作之间的最小间隔时间（单位：毫秒）。

默认值

100

取值范围

[0,1000]

取值为 0 时，表示无间隔，连续执行维护操作。

访问权限

R/W

是否全局参数

是

修改生效方式

store_maint_span 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 store_maint_span 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET store_maint_span TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW store_maint_span
```

注意事项

无

2.7 存储管理附属函数

2.7.1 cache_undo_wrt

参数名称

cache_undo_wrt

参数解释

控制 UNDO 日志与 REDO 日志的缓存与写入策略，决定两者在事务提交时的持久化方式：

- 0（独立记载）：UNDO 与 REDO 日志分开写入，各自独立刷盘。
- 1（合并记载）：将 UNDO 日志缓存到 REDO 日志中，合并写入以减少 I/O 操作。
- 2（不记载）：UNDO 日志不通过 REDO 日志持久化，直接独立管理。

默认值

1

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

cache_undo_wrt 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 cache_undo_wrt 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET cache_undo_wrt TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW cache_undo_wrt
```

注意事项

无

2.7.2 catalog_write_through

参数名称

catalog_write_through

参数解释

控制数据库在修改存储控制信息（如元数据）后，如何将数据同步到磁盘。

- 1：立即保存，数据写入动作不会等待数据写入到磁盘即可结束，写入的数据会被缓存到操作系统内核的页面缓存（PageCache）中，不会立即写入磁盘。
- 2：立即写通，数据写入动作必须等待数据写入到磁盘后才结束（需要保证数据及其关键元数据同步到磁盘）。

默认值

1

取值范围

[1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

catalog_write_through 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 catalog_write_through 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW catalog_write_through
```

注意事项

无

2.7.3 data_persistence

参数名称

data_persistence

参数解释

定义数据持久化策略，控制数据及日志的落盘行为：

- 0：数据与 REDO 日志同时写入，保障最高数据安全性。
- 1：仅写入 REDO 日志，数据缓存延迟刷盘，平衡性能与可靠性。
- 2：数据与 REDO 日志均不写入，风险高。

默认值

0

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

data_persistence 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 data_persistence 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET data_persistence TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW data_persistence
```

注意事项

无

2.7.4 datafile_random_access

参数名称

datafile_random_access

参数解释

控制数据文件是否以伪随机存取优化方式打开。

数据文件以随机 I/O 方式开启文件句柄。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

datafile_random_access 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 datafile_random_access 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW datafile_random_access
```

注意事项

无

2.7.5 datafile_use_aio

参数名称

datafile_use_aio

参数解释

控制是否启用异步文件 I/O，用于优化数据文件的读写性能。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

datafile_use_aio 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 datafile_use_aio 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW datafile_use_aio
```

注意事项

无

2.7.6 datafile_write_through

参数名称

datafile_write_through

参数解释

控制数据库在修改数据文件时，是否要求操作系统立即将变更写回磁盘。

数据文件以写通方式开启文件句柄，该方式不存在磁盘写延迟。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

datafile_write_through 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 datafile_write_through 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。

3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW datafile_write_through
```

注意事项

无

2.7.7 major_retry_num

参数名称

major_retry_num

参数解释

设置获取主存储版本信息失败时的最大重试次数，每次重试间隔 50 毫秒。获取存储主版本信息失败次数超过该参数给定的额定次数时，抛出“数据存贮体正在进行切换操作”错误。当重试时间大于节点死亡判定时间（参数 node_dead_delay）时可避免因存储节点宕抛出大量“数据存贮体正在进行切换操作”错误。

重试时间的计算方式为“重试时间 \approx major_retry_num * 50 / 1000”（单位：秒）。例如设置“node_dead_delay = 30”时，可设置“major_retry_num = 1400”。

默认值

20

取值范围

[0,10000]

访问权限

R/W

是否全局参数

是

修改生效方式

major_retry_num 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 major_retry_num 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET major_retry_num TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW major_retry_num
```

注意事项

无

2.7.8 max_write_back_num

参数名称

max_write_back_num

参数解释

定义脏数据块回写队列的最大长度，即内存中等待持久化到磁盘的脏数据块（已修改但未落盘的数据）的最大缓存数量。

默认值

32

取值范围

[1,256]

访问权限

R/W

是否全局参数

是

修改生效方式

max_write_back_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 max_write_back_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_write_back_num
```

注意事项

无

2.7.9 redo_write_through

参数名称

redo_write_through

参数解释

控制重做日志（Redo Log）文件的写入策略，是否以写通（write through）方式打开。

重做日志文件以写通方式开启文件句柄，该方式不存在磁盘写延迟。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

redo_write_through 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 redo_write_through 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW redo_write_through
```

注意事项

当 use_undo_space=false（即不启用独立 Undo 表空间）时，redo_write_through 无效。

2.7.10 write_back_thd_num

参数名称

write_back_thd_num

参数解释

定义系统中负责将数据从内存缓冲区持久化到磁盘的线程数量，直接影响数据写入性能与系统资源利用率。

默认值

1

取值范围

[1,128]

访问权限

R/W

是否全局参数

是

修改生效方式

write_back_thd_num 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 write_back_thd_num 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW write_back_thd_num
```

注意事项

无

2.8 事务管理参数

2.8.1 checkpoint_delay

参数名称

checkpoint_delay

参数解释

设置 checkpoint 的延迟时间（单位：分钟）。

checkpoint 是数据库的一种机制，用于将内存中的脏页（已修改但未写入磁盘的数据页）同步到磁盘，并记录当前事务日志的状态，保证数据的一致性。

默认值

15

取值范围

[1,100]

访问权限

R/W

是否全局参数

是

修改生效方式

checkpoint_delay 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 checkpoint_delay 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW checkpoint_delay
```

注意事项

无

2.8.2 def_auto_commit

参数名称

def_auto_commit

参数解释

控制事务是否默认启用自动提交模式。在自动提交模式下，每条 SQL 语句都会被视为一个独立的事务，并在执行完成后立即提交。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

def_auto_commit 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 def_auto_commit 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_auto_commit
```

注意事项

无

2.8.3 def_iso_level

参数名称

def_iso_level

参数解释

设置默认事务隔离级别。

- 0: 只读
- 1: 读已提交
- 2: 可重复读
- 3: 序列化

事务隔离级别定义了事务之间的可见性和并发控制行为，直接影响数据一致性和性能。

默认值

1

取值范围

[0|1|2|3]

访问权限

R/W

是否全局参数

是

修改生效方式

def_iso_level 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 def_iso_level 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET def_iso_level TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW def_iso_level
```

注意事项

无

2.8.4 dlock_check_delay

参数名称

dlock_check_delay

参数解释

设置死锁检测的延迟时间（单位：毫秒）。

控制数据库在事务执行过程中进行死锁检测的频率，从而平衡系统性能和死锁检测的及时性。

默认值

3000

取值范围

[1000,600000]

访问权限

R/W

是否全局参数

是

修改生效方式

dlock_check_delay 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 dlock_check_delay 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET dlock_check_delay TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW dlock_check_delay
```

注意事项

无

2.8.5 exit_when_redo_err

参数名称

exit_when_redo_err

参数解释

控制在重做（REDO）恢复失败时，系统是否主动退出运行。

- true：当重做恢复失败时，系统主动退出运行。
- false：当重做恢复失败时，系统忽略错误继续修复，修复完启动系统。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

exit_when_redo_err 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 exit_when_redo_err 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW exit_when_redo_err
```

注意事项

在重做日志损坏且无法修复时，若需快速恢复数据库服务，可通过设置 exit_when_redo_err 为 false，执行不完全恢复（跳过损坏的日志段）强制启动数据库，接受部分数据丢失的风险，并通过后续数据修复操作（如备份恢复或增量补录）最大程度弥补数据缺失。

2.8.6 max_trans_modify

参数名称

max_trans_modify

参数解释

限制单个事务中 DML 操作（如 INSERT、UPDATE、DELETE）允许变更的最大行数。

默认值

10000

取值范围

[0,2147483647]

取值为 0 时，表示不限制事务中 DML 操作变更的行数。

访问权限

R/W

是否全局参数

是

修改生效方式

max_trans_modify 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_trans_modify 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_trans_modify TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_trans_modify
```

注意事项

无

2.8.7 node_dead_delay

参数名称

node_dead_delay

参数解释

定义节点判死的心跳检测阈值，每次心跳检测的间隔默认为 2 秒。

在集群中，若节点与主节点（Master）之间连续丢失的心跳次数超过该阈值，则判定该节点已宕机。

节点判死的时间计算方式为“节点判死时间 = (node_dead_delay + 5) * 2”（单位：秒）。例如设置“node_dead_delay = 30”时，节点在 70 秒无响应后会被标记为不可用。

默认值

30

取值范围

[1,600]

访问权限

R/W

是否全局参数

是

修改生效方式

node_dead_delay 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 node_dead_delay 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET node_dead_delay TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW node_dead_delay
```

注意事项

无

2.8.8 skip_boot_rollback

参数名称

skip_boot_rollback

参数解释

控制数据库服务重启过程中是否跳过回滚（ROLLBACK）阶段。

适用于数据库崩溃后需快速重启服务，但无法等待长时间的回滚操作。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

skip_boot_rollback 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 skip_boot_rollback 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW skip_boot_rollback
```

注意事项

若跳过事务回滚，未提交事务的残留数据会保存到数据库文件中，导致数据不一致。需通过手动对之前的数据还原，以确保数据一致性。

2.8.9 strictly_commit

参数名称

strictly_commit

参数解释

控制事务日志是否必须在事务提交时强制写入磁盘（Write-Ahead Logging, WAL）。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

strictly_commit 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 strictly_commit 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW strictly_commit
```

注意事项

无

2.8.10 undo_delay_free

参数名称

undo_delay_free

参数解释

控制事务结束后 UNDO 项的延迟释放时间（单位：毫秒），即已提交事务的 UNDO 记录在 UNDO 表空间中保留的时间。

默认值

3000

取值范围

[0,86400000]

取值为 0 时，表示立即释放 UNDO 项。

访问权限

R/W

是否全局参数

是

修改生效方式

undo_delay_free 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 undo_delay_free 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET undo_delay_free TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW undo_delay_free
```

注意事项

适当的保留时间可以提高查询和恢复操作的性能，但过长的保留时间可能会影响系统性能。

2.9 数据备份同步参数

2.9.1 enable_recycle

参数名称

enable_recycle

参数解释

表对象回收站开关，设置是否启用表级回收站。

开启表级回收后，被删除的表对象结构信息会以另一种形式保存在系统表 SYS_RECYCLER 中。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_recycle 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_recycle 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_recycle TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_recycle
```

注意事项

无

2.9.2 ignore_when_meta_err

参数名称

ignore_when_meta_err

参数解释

设置变更日志元信息错误处理模式。

当消费 binlog 日志（如解析为 SQL 语句）时，若本地元数据（表结构、字段定义等）与 binlog 中的记录不匹配（例如表被删除或结构变更），该参数决定如何处理此类错误：

- 0：直接抛出异常信息。
- 1：SQL 字段返回 NULL，并每触发额定次数记一次 trace 日志。
- 2：不返回当前元数据不存在的记录，并每触发额定次数记一次 trace 日志。

默认值

2

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

ignore_when_meta_err 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 ignore_when_meta_err 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET ignore_when_meta_err TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW ignore_when_meta_err
```

注意事项

无

2.9.3 log_archive_mode

参数名称

log_archive_mode

参数解释

设置重做（REDO）日志归档模式。

- 0：写满的 REDO 日志不进行额外备份或归档。
- 1：新建归档日志，将写满的 REDO 日志的数据内容拷贝至归档日志。
- 2：新建 REDO 日志，将写满的 REDO 日志重命名为归档日志。

默认值

0

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

log_archive_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 log_archive_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET log_archive_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW log_archive_mode
```

注意事项

无

2.9.4 log_supplement

参数名称

log_supplement

参数解释

设置是否开启日志增补。

开启该功能后，可以在查阅变更记载信息同时获取到变更 SQL 的回滚操作。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

log_supplement 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 log_supplement 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET log_supplement TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW log_supplement
```

注意事项

无

2.9.5 max_allow_lob_len

参数名称

max_allow_lob_len

参数解释

设置变更记载功能模块允许记载的最大大对象（LOB）大小（单位：MB）。

默认值

10

取值范围

[1,2048]

访问权限

R/W

是否全局参数

是

修改生效方式

max_allow_lob_len 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 max_allow_lob_len 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_allow_lob_len TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW max_allow_lob_len
```

注意事项

无

2.9.6 modify_log_parti_num

参数名称

modify_log_parti_num

参数解释

设置变更日志并行写通道数。

binlog 日志文件的分区数，关联变更消息并行写通道数。

默认值

8

取值范围

[1,256]

访问权限

R

是否全局参数

是

修改生效方式

modify_log_parti_num 参数专为系统首次初始化设计，初始化完成后将不可修改。

注意事项

无

2.9.7 modify_log_pice_len

参数名称

modify_log_pice_len

参数解释

设置变更日志文件分片大小（单位：MB）。

binlog 日志文件的切片阈值，即 binlog 日志文件可记载的总字节数不会超过该值。

默认值

100

取值范围

[10,1048576]

访问权限

R/W

是否全局参数

是

修改生效方式

modify_log_pice_len 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 modify_log_pice_len 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW modify_log_pice_len
```

注意事项

无

2.10 系统运行日志参数

2.10.1 core_dump

参数名称

core_dump

参数解释

设置在内存段错误异常情况下是否生成核心转储文件（core dump）。

- true：开启核心转储功能，在内存段错误异常情况下生成核心转储文件。
- false：关闭核心转储功能，不在内存段错误异常情况下生成核心转储文件。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

core_dump 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 core_dump 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW core_dump
```

注意事项

无

2.10.2 enable_node_degrade

参数名称

enable_node_degrade

参数解释

设置存储域的分配策略。

- true：启用节点降级。
 - 主 master：两节点时，不降级；大于两节点时，仅自身存活时降级为只读，无法处理节点加入请求。
 - 副 master：仅自身存活时降级为只读，无法处理节点加入请求。
- false：不启用降级，按正常节点故障处理，存在脑裂后双活风险。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_node_degrade 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_node_degrade 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET max_allow_lob_len TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_node_degrade
```

注意事项

无

2.10.3 errlog_size

参数名称

errlog_size

参数解释

设置错误日志的分割长度（单位：MB）。

当错误信息写入错误日志文件内的总字节数即将超过该参数给定的阈值时，系统将会创建一个新的文件并从头开始记录错误信息。

默认值

100

取值范围

[1,1024]

访问权限

R/W

是否全局参数

是

修改生效方式

errlog_size 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 errlog_size 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET errlog_size TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW errlog_size
```

注意事项

无

2.10.4 error_level

参数名称

error_level

参数解释

设置错误日志的最低记载等级。

系统内部错误信息按等级划分，不同等级的错误信息受不同错误日志最低记载等级的约束。

- 0：任何错误等级的错误信息均不记载至错误日志。
- 1：仅记录事务中止、死锁、事务号异常、系统异常、网络异常和内存异常等异常信息。
- 2：记录事务中止、死锁、事务号异常、系统异常、网络异常、内存异常、用户定义异常和命令中止异常等异常信息。
- 3：任何错误等级的错误信息均记载至错误日志。

默认值

3

取值范围

[0,10]

访问权限

R/W

是否全局参数

是

修改生效方式

error_level 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 error_level 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET error_level TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW error_level
```

注意事项

无

2.10.5 is_zh

参数名称

is_zh

参数解释

设置日志信息是否支持中文。

- true: 开启中文支持，日志信息将以中文形式记录。
- false: 关闭中文支持，日志信息将以英文形式记录。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

is_zh 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 is_zh 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。

3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW is_zh
```

注意事项

无

2.10.6 reg_command

参数名称

reg_command

参数解释

设置是否将 SQL 命令记录到 COMMAND 日志中，控制粒度支持节点级。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

reg_command 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 reg_command 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET reg_command TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW reg_command
```

注意事项

该参数独立于 reg_ddl 参数，不能与 reg_ddl 参数同时开启。

2.10.7 reg_ddl

参数名称

reg_ddl

参数解释

设置是否将 DDL 类型的 SQL 请求记录到 COMMAND 日志中。

- true: 开启记录 DDL 请求的记录功能。
- false: 关闭记录 DDL 请求的记录功能。

默认值

false

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

reg_ddl 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 reg_ddl 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET reg_ddl TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW reg_ddl
```

注意事项

该参数独立于 reg_command 参数，不能与 reg_command 参数同时开启。

2.10.8 slow_sql_time

参数名称

slow_sql_time

参数解释

设置记录执行时间超过阈值的 SQL 语句（单位：毫秒）。

- 0：关闭慢 SQL 记录功能。
- 大于 0：开启慢 SQL 记录功能，将执行时间超过该阈值的 SQL 语句记录到 SLOWSQL.LOG 日志文件中。

可通过执行 `SELECT*FROM sys_slowsql_log` 查询慢 SQL 记录的结果。

默认值

0

取值范围

[0,3600000]

访问权限

R/W

是否全局参数

是

修改生效方式

`slow_sql_time` 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 `slow_sql_time` 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET slow_sql_time TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW slow_sql_time
```

注意事项

无

2.10.9 trace_login

参数名称

`trace_login`

参数解释

设置是否追溯用户口令验证失败的登录连接的 IPv4 地址并记录到 TRACE 日志中。

- true：开启追溯功能，记录失败登录连接的 IPv4 地址到 TRACE 日志。

- false: 关闭追溯功能, 不会记录失败登录连接的 IPv4 地址。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

trace_login 可在系统运行时, 通过数据库 SET 命令修改, 修改后参数值立即生效。

操作步骤:

1. 在数据库中执行以下命令修改 trace_login 参数值, 根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET trace_login TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW trace_login
```

注意事项

无

2.11 安全审计参数

2.11.1 enable_audit

参数名称

enable_audit

参数解释

设置系统审计功能的开启或关闭。

- true: 开启系统审计功能, 记录审计日志。
- false: 关闭系统审计功能, 不记录审计日志。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_audit 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_audit 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_audit TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_audit
```

注意事项

无

2.11.2 security_level

参数名称

security_level

参数解释

设置新建连接行为的审计安全等级。

- 0：安全等级为低级，任何新建连接按照实际审计策略记载。
- 1：安全等级为高级，任何新建连接无论是否成功均要被记入审计表。

默认值

0

取值范围

[0|1]

访问权限

R/W

是否全局参数

是

修改生效方式

security_level 不支持在数据库中执行 SET 命令修改，只能通过修改 xugu.ini 文件，重启系统后生效。

操作步骤：

1. 在数据库中执行以下命令中止数据库服务，退出控制台工具。

```
SQL> SET SHUTDOWN
```

2. 修改 SETUP 文件夹下 xugu.ini 文件中的 security_level 参数，根据实际需求在取值范围内选择合适的参数值，保存并退出。
3. 重新启动数据库服务。
4. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW security_level
```

注意事项

当参数 enable_audit = true，即系统审计功能开启时，security_level 才会生效。

2.12 系统分析参数

2.12.1 analyze_level

参数名称

analyze_level

参数解释

设置统计分析函数的采样级别。取值越大，采样越精确。

参数值与采样间隔对应关系：

- 参数值 1，采样间隔 32
- 参数值 2，采样间隔 16
- 参数值 3，采样间隔 8
- 参数值 4，采样间隔 4
- 参数值 5，采样间隔 2

- 参数值 6，采样间隔 1

采样间隔为 1（参数值为 6）表示全采样，此时统计信息最准确，但也最耗时。因此，在统计分析时应根据实际应用场景设置合理的采样级别。

默认值

2

取值范围

[1|2|3|4|5|6]

访问权限

R/W

是否全局参数

是

修改生效方式

analyze_level 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 analyze_level 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET analyze_level TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW analyze_level
```

注意事项

根据表的总记录数 n，推荐的 analyze_level 设置如下：

- 当表的总记录数 $n < 65536$ 时，建议 analyze_level 设置为 6。
- 当 $65536 < n < 131072$ 时，建议 analyze_level 设置为 5。
- 当 $131072 < n < 4194304$ 时，建议 analyze_level 设置为 4。
- 当 $n > 4194304$ 时，采样方式调整为块采样，应根据数据 buff 缓冲区和表数据总量大小动态调整，无建议值。

2.12.2 analyze_mode

参数名称

analyze_mode

参数解释

设置统计分析模式。

- 1: 分析索引字段及指定收集字段。
- 2: 在模式 1 的基础上, 增加支持分析数字及日期类型字段。

默认值

1

取值范围

[1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

analyze_mode 可在系统运行时, 通过数据库 SET 命令修改, 修改后参数值立即生效。

操作步骤:

1. 在数据库中执行以下命令修改 analyze_mode 参数值, 根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET analyze_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW analyze_mode
```

注意事项

无

2.12.3 analyze_threshold

参数名称

analyze_threshold

参数解释

设置统计分析的触发阈值。

优先使用统计表上的统计分析触发阈值, 如果统计表上没有设置触发阈值, 则使用该系统参数给定的阈值。

默认值

10

取值范围

[1,100] 取整数

访问权限

R/W

是否全局参数

是

修改生效方式

analyze_threshold 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 analyze_threshold 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET analyze_threshold TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW analyze_threshold
```

注意事项

无

2.12.4 analyze_time

参数名称

analyze_time

参数解释

设置定时分析的默认执行时间，仅支持整点定时分析，任何非整点时间值常数都会进行去零取整运算。

默认值

02:00:00

取值范围

[00:00:00,23:59:59]

访问权限

R/W

是否全局参数

是

修改生效方式

analyze_time 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 analyze_time 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET analyze_time TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW analyze_time
```

注意事项

该参数仅支持整点定时分析，任何非整点时间值常数都会进行去零取整运算。

2.12.5 debug_flag

参数名称

debug_flag

参数解释

设置数据库的调试标记位。该参数可辅助开发人员对数据库进行调试分析。

- 0：默认配置项，不开启任何调试模式。
- 2：高内存消耗 SQL 的调试追踪模式，支持以 TRACE 日志和 COMMAND 日志的形式跟踪相关 SQL 的活动。

默认值

0

取值范围

[0,2147483647]

访问权限

R/W

是否全局参数

是

修改生效方式

debug_flag 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 debug_flag 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET debug_flag TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW debug_flag
```

注意事项

无

2.12.6 enable_analyze

参数名称

enable_analyze

参数解释

设置定时分析功能的开启或关闭。

- true：开启定时分析功能，系统会定期自动执行分析任务，以优化数据库性能。
- false：关闭定时分析功能，系统不会自动执行定时分析任务。

默认值

true

取值范围

[true|false]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_analyze 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_analyze 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_analyze TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_analyze
```

注意事项

无

2.12.7 enable_monitor

参数名称

enable_monitor

参数解释

设置磁盘、全局锁和 SQL 请求等资源信息的监控标记。

根据用户给予特定的值用于对相应的全局资源信息进行监控，被特定指定的监控资源信息最终可通过系统表 sys_monitors 进行查看。

- 1: 磁盘回写次数及字节增量信息监控标记。
- 8: 全局锁资源信息监控标记。
- 16: SQL 请求信息监控标记。

默认值

1

取值范围

[0,2147483647]

访问权限

R/W

是否全局参数

是

修改生效方式

enable_monitor 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 enable_monitor 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET enable_monitor TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW enable_monitor
```

注意事项

无

2.12.8 gstore_pick_mode

参数名称

gstore_pick_mode

参数解释

设置 Gstore 缓存访问的并发控制模式。该参数决定了在多线程环境下如何管理缓存访问的并发控制。

- 0：值比较模式。
- 1：校验码模式。
- 2：锁模式。

默认值

1

取值范围

[0|1|2]

访问权限

R/W

是否全局参数

是

修改生效方式

gstore_pick_mode 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 gstore_pick_mode 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET gstore_pick_mode TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW gstore_pick_mode
```

注意事项

无

2.12.9 histogram_bucket

参数名称

histogram_bucket

参数解释

设置直方图默认桶数。

在自动分析后，设置直方图桶数，会进行直方图统计信息收集。直方图统计信息可以用于指导优化器路径选择，但同时也会消耗更多资源和时间。

默认值

0

取值范围

[0,255]

取值为 0 时，表示不开启直方图统计。

访问权限

R/W

是否全局参数

是

修改生效方式

histogram_bucket 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 histogram_bucket 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET histogram_bucket TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW histogram_bucket
```

注意事项

无

3 cluster.ini 系统文件配置

3.1 概述

虚谷数据库支持数据库集群服务，cluster.ini 系统参数配置文件是一个专门针对集群服务进行系统配置的文件。如果选择集群部署方案，请在正式部署前对模板文件进行配置，文件路径：`./SETUP/cluster.ini.template`。

cluster.ini.template 文件示例：

```
#MAX_NODES=16      MASTER_GRPS=1      PROTOCOL='UDP'      MSG_PORT_NUM
=1      MAX_SEND_WIN=510
MSG_HAVE_CRC=0      MERGE_SMALL_MSG=1      MSG_SIZE=64000      TIMEOUT
=20000      RPC_WINDOW=16
EJE_WINDOW=16      MAX_SHAKE_TIME=1200      MY_NID=0001      CHECK_RACK=0

NID=0001  RACK=0001  PORTS='IP:PORT'  ROLE='MSQW'  LPU=3
STORE_WEIGHT=3  STATE=DETECT;
NID=0002  RACK=0001  PORTS='IP:PORT'  ROLE='MSQW'  LPU=3
STORE_WEIGHT=3  STATE=DETECT;
NID=0003  RACK=0001  PORTS='IP:PORT'  ROLE='SQWG'  LPU=3
STORE_WEIGHT=3  STATE=DETECT;
```



- 通过模板文件进行节点信息配置，配置完成后需重命名为“cluster.ini”。
- “cluster.ini”文件中所有参数项位置顺序不可更改，且首行#不可删除。

3.2 全局参数

MAX_NODES

MAX_NODES 参数用于设置集群最大节点数，限制集群节点规模。

集群节点数最小值不小于 2，且最大值理论上无上限，但实际集群节点的最大值受限于实际硬件设备和网络带宽影响，请根据实际业务场景进行设置。

MASTER_GRPS

MASTER_GRPS 参数用于设置主控角色组数（两个一组构成主备关系）。

目前仅支持 1 组主控角色。

PROTOCOL

PROTOCOL 参数用于设置数据库集群中各节点间通信所采用的通信协议，支持 UDP。

MSG_PORT_NUM

MSG_PORT_NUM 参数用于设置数据库集群中各节点之间的通讯端口数，与节点参数的 PORTS 保持一致，最大值为 16。

MAX_SEND_WIN

MAX_SEND_WIN 参数用于设定数据库集群中各节点间通信的最大发送窗口大小，最大值为 2046。通过调整消息窗口的大小，可以有效控制在高并发工作负载下可能出现的消息风暴，确保消息能够有序传递，维持系统的稳定性和效率。

MSG_HAVE_CRC

MSG_HAVE_CRC 参数用于设置数据库集群中各节点之间的消息是否需要进行 CRC 校验。

- 0：关闭 CRC 校验。
- 1：开启 CRC 校验。

MERGE_SMALL_MSG

MERGE_SMALL_MSG 参数用于设置是否启用消息小包合并功能。

- 0：关闭消息小包合并功能。
- 1：开启消息小包合并功能。

MSG_SIZE

MSG_SIZE 参数用于设置数据库集群中各节点之间通讯消息包的最大尺寸。

TIMEOUT

TIMEOUT 参数用于设置数据库集群中各节点之间消息超时重发的最大时间延迟（单位：微秒）。

RPC_WINDOW

RPC_WINDOW 参数用于设定数据库集群中各节点之间进行远程过程调用（RPC, Remote Procedure Call）时的流量控制窗口数。

取值范围 [0,63]。

EJE_WINDOW

EJE_WINDOW 参数用于设定数据库集群中各节点之间进行通信时的弹射器（Ejector）流量控制窗口数。

取值范围 [0,255]。

MAX_SHAKE_TIME

MAX_SHAKE_TIME 参数用于设置在数据库集群启动初期，非 master 节点与主从 master，以及主从 master 节点之间执行握手协议的周期次数。

MY_NID

MY_NID 参数用于设置当前 cluster.ini 配置文件所在节点的集群内部节点编号。当前操作系统 IP 地址配置必须和此节点号对应的节点参数中 PORTS 相匹配。

CHECK_RACK

CHECK_RACK 参数用于设置是否检查主节点的机架号，以确认主节点是否位于不同的物理机架上。

- 0：不检测。
- 1：检测。

3.3 节点参数

NID

集群节点号，该值连续递增数值。

PORTS

PORTS 参数用于设置集群内部各个节点自身的 IPv4 地址和通信端口号。配置通讯端口组数需与“MSG_PORT_NUM”一致。

支持配置多路内部通信端口，示例：

```
PORTS='192.168.66.45:17100,192.168.66.45:17200'
```

在配置集群节点的 PORTS 参数时，显式指定的端口号作为集群节点内部通信的接收端口号。除了这个接收端口号之外，每个节点还会默认占用一个发送端口号，该发送端口号的计算方式为“发送端口号 = 接收端口号 + 20”。例如，如果一个节点的接收端口号被设置为 17100，那么它的发送端口号则是 17120。



注意

在部署集群时，提前验证所需的端口（包括接收端口和发送端口）是否已经被其他服务或进程占用。

ROLE

ROLE 参数用于指定集群内部各个节点自身所承担的角色属性。每个角色由一个大写字母标识，且大多数角色可以在同一个节点上自由组合，如表 3-1 所示。

表 3-1 ROLE 参数解释

角色名	标识符	角色描述	注意事项
主控角色	M	负责节点管理（如节点的增加、删除、状态查看）、资源管理、故障处理（通过心跳检测节点或服务的故障）、存储均衡修复迁移以及全局锁管理和全局仲裁服务等	主控角色（M）必须配置为 2 个且分别配置在奇偶节点，建议保持默认。
存储角色	S	负责数据存储，存储片段实际存储管理，响应工作节点发来的增、删、查、改等底层数据操作	-
查询角色	Q	负责建立监听	查询角色（Q）与工作角色（W）需配置于同一节点上。
工作角色	W	负责接受客户请求、规划任务、执行任务等	
变更记载角色	G	变更数据采集，用于数据备份，用于提供灵活便捷的备份方案	变更记载角色（G）最多允许配置于 2 个节点。

LPU

LPU 参数用于设置逻辑处理器个数，仅在配置了工作角色的节点上（即 ROLE 包含 W 属性的节点）有效。它决定了在并行计算过程中可以使用的最大逻辑 CPU 数量。

该值建议设置为“服务器物理 CPU 核数-1”。

STORE_WEIGHT

STORE_WEIGHT 参数用于设置节点存储权重。搭建集群时，若服务器磁盘存储容量不同，则按磁盘容量比配置；若集群磁盘存储容量相同，则配置为同一数值。

STATE

STATE 参数用于设置数据库集群中各节点状态。

- DETECT：未联机。
- INVALID：无心跳。

4 types.ini 系统文件配置

概述

系统配置文件 types.ini 用于兼容其他数据库基础数据类型，将其他数据库数据类型名映射为兼容的虚谷数据库数据类型，后续在使用其他数据库数据类型名称时系统将自动进行数据类型映射转换。

策略设置

初始默认配置：

BOOL	BOOLEAN
INT	INTEGER
SHORT	SMALLINT
LONGINT	BIGINT
LONG	CLOB
REAL	FLOAT
DECIMAL	NUMERIC
TEXT	VARCHAR
NCHAR	CHAR
NVARCHAR	VARCHAR
NVARCHAR2	VARCHAR
VARCHAR2	VARCHAR
NCLOB	CLOB
PLS_INTEGER	INTEGER
BINARY_INTEGER	INTEGER
BINARY_DOUBLE	DOUBLE
BINARY_FLOAT	FLOAT
RAW	BINARY
XMLTYPE	XML

映射关系说明

一行一个映射关系，其中左侧为其他数据库基础数据类型映射名，右侧为虚谷数据库基础数据类型名。

5 mount.ini 系统文件配置

概述

系统配置文件 mount.ini 用于虚谷数据库各类文件的路径映射，路径映射既可以使用相对 XHOME 的相对路径，也可以使用系统绝对路径。

数据库文件默认存放于安装目录的“XGLOG”和“XHOME”文件夹下。

路径设置

/XGLOG	./XGLOG	//系统内部内嵌的所有以'/XGLOG'开头的文件路径都会映射为'./XGLOG'
/CATA	./XHOME/CATA	//系统内部内嵌的所有以'/CATA'开头的文件路径都会映射为'./XHOME/CATA'
/DATA	./XHOME/DATA	
/TEMP	./XHOME/TEMP	
/REDO	./XHOME/REDO	
/BACKUP	./XHOME/BACKUP	
/ARCH	./XHOME/ARCH	
/UNDO	./XHOME/UNDO	
/MODI	./XHOME/XMODI	

映射关系说明

数据库目录结构	系统映射路径	说明
/XGLOG	./XGLOG	存储系统相关日志信息，包括错误日志、事件日志等
/CATA	./XHOME/CATA	存储系统结构性控制文件
/REDO	./XHOME/REDO	存储数据库重做日志文件
/UNDO	./XHOME/UNDO	存储数据库回滚日志文件
/MODI	./XHOME/XMODI	存储数据库变更数据文件
/DATA	./XHOME/DATA	存储数据库数据文件
/TEMP	./XHOME/TEMP	存储数据库临时表空间文件

接下页

数据库目录结构	系统映射路径	说明
/BACKUP	./BACKUP	存储数据库备份文件
/ARCH	./XHOME/ARCH	存储数据库归档日志文件

6 trust.ini 系统文件配置

概述

系统配置文件 trust.ini 用于对请求连接的客户端进行受信认证。该文件定义了哪些客户端 IP 地址是可以信任的，哪些是不可信任的。通过配置黑白名单策略，可以有效地控制哪些客户端可以与服务端建立会话连接。

黑名单优先级高于白名单。当某个 IPv4 地址被同时加入黑白名单策略时，黑名单优先执行，该地址不被允许与服务端建立会话连接。只有不在 trust.ini 黑名单策略中的 IPv4 地址的客户端才可以被允许与服务端建立会话连接。

策略设置

trust.ini 受信认证支持多行书写，每个非注释行内容就代表一个访问受信策略。

- 格式一

受信/拒绝来自任何地址上的任何用户身份登录任何库的连接请求。

```
trust/untrust everydb everyone from anywhere
```

或

```
trust/untrust everyone from anywhere
```

- 格式二

受信/拒绝来自任何地址上的 U1 用户身份登录 DB1 库的连接请求。

```
trust/untrust DB1 U1 from anywhere
```

- 格式三

受信/拒绝来自 127.0.0.1 到 127.0.0.225 之间地址的任何用户身份登录任何库的连接请求。

```
trust/untrust everydb everyone from 127.0.0.1 to 127.0.1.225
```

或

```
trust/untrust everyone from 127.0.0.1 to 127.0.0.225
```

关键字说明

关键字	说明
trust	信任当前策略中出现的用户、登录库和 IP 地址
untrust	拒绝当前策略中出现的用户、登录库和 IP 地址
everydb	所有数据库
everyone	所有用户
everywhere	所有 IP 地址

7 连接会话参数

连接会话参数主要用于在客户端向虚谷数据库服务端建立连接请求后，在进行会话验证时，传输额外的信息内容从而达到修饰连接会话自身属性的目的。

7.1 DATABASE

功能特性

该参数用于指定当前连接需要登录的数据库，这个数据库必须是已经确定存在的，任何尝试登录不存在的数据库请求都会失败。

7.2 USER

功能特性

指定新建连接会话的用户名，通过用户名查找其系统内唯一值身份标识（用户 ID），不存在则登录失败。

7.3 PASSWORD

功能特性

指定参数 USER 表示的用户身份认证口令（静态口令）。

<DATABASE, USER, PASSWORD> 属性组用于登录时用户身份认证。登录成功后

DATABASE 和 USER 保存相关信息在该连接会话中，在连接会话生命周期内作为所有操作的执行上下文，如用于：

- 当前模式
- 执行时权限检测
- 创建模式对象的属主设置
- 审计记载

7.4 SESSION_USER

功能特性

查看和切换连接会话当前用户。

示例

- 查看当前用户。

```
SHOW SESSION_USER;
```

- 切换当前用户。

```
SET SESSION_USER TO username;  
SET SESSION AUTHORIZATION username;
```

注意

- 任何用户都不可切换到安全管理员（SYSSSO）和审计管理员（SYSAUDITOR）。
- 登录用户具备 DBA 权限或是系统管理员、安全管理员和审计管理员之一时，才能进行切换当前用户。
- 安全管理员（SYSSSO）仅支持切换到安全员，切换后不可再切换回安全管理员（SYSSSO）。
- 审计管理员（SYSSSO）仅支持切换到审计员，切换后不可再切换回审计管理员（SYSSSO）。
- 切换仅改变当前用户（sys_session.curr_user_id 和 curr_user_name），登录用户不发生变化（sys_session.user_id 和 user_name）。

7.5 DRIVER_VERSION

功能特性

客户端连接会话所使用的通讯协议版本号，指定不同的版本号将使用不同的通讯协议。

参数说明

参数值	对应值
301	301 通讯协议
其它值	201 通讯协议

示例

查看当前数据库通信协议。

```
SHOW DRIVER_VERSION;
```

7.6 LANGUAGE

功能特性

数据库服务端可支持的 SQL 语法树规则。

参数说明

参数值	对应值
TSQL	TSQL 语法树规则（已弃用）
其它值	PLSQL 语法树规则

示例

查看当前数据库支持的 SQL 语法树规则。

```
SHOW LANGUAGE;
```

7.7 CHAR_SET

功能特性

客户端连接会话所使用的字符集编码，服务端会依据 CHAR_SET 参数给定的字符集对来自该连接会话上的字符内容和对经由该连接会话发往客户端的字符内容进行转码。

数据库端字符集转换：

- 读入：会话字符集-> 库字符集转换
- 写出：库字符集-> 会话字符集

📖 说明

- bin 和 ci 后缀字符集区别：ci 不区分大小写，bin 区分大小写。
- 创建库和连接会话字符集在不指定 bin 和 ci 后缀时使用 bin 类字符集（同一字符集不同后缀也涉及字符集转换问题）。
- 当前版本 JDBC 驱动不支持指定字符集后缀。

示例

- 设置字符集为 gbk

```
SET CHAR_SET to 'gbk';
```

- 查看字符集

```
SHOW CHAR_SET;
```

说明

字符集默认使用 `def_charset`。

7.8 TIME_FORMAT

功能特性

客户端连接会话上的时间值格式，该参数决定发往客户端的时间值类型数据的显示格式。

- `TO_CHAR`, `TO_DATE` 函数不指定时间格式时使用会话时间格式。
- 字符常量赋值各时间类型。
- 字符常量和时间类型比较。

示例

- 设置时间格式

```
SET TIME_FORMAT TO 'YYYY-MM-DD HH24:MI:SS';
```

- 显示时间格式

```
SHOW TIME_FORMAT;
```

说明

默认使用 `def_timefmt` (“YYYY-MM-DD HH24:MI:SS”) 格式。

7.9 ISO_LEVEL

功能特性

客户端连接会话上的事务隔离级别。

参数说明

参数值	事务隔离级别
READ ONLY	只读 (0)
READ_COMMITTED	读已提交 (1)
REPEATABLE READ	可重复读 (2)
SERIALIZABLE	序列化 (3)

示例

- 设置隔离级别，支持两种设置

```
SET ISO_LEVEL TO '0';  
SET TRANSACTION ISOLATION LEVEL READ ONLY;
```

- 显示隔离级别

```
SHOW ISO_LEVEL;  
SHOW TRANSACTION ISOLATION LEVEL;
```

说明

不指定则使用 `def_iso_level` 配置值，默认读已提交 (1)。

7.10 AUTO_COMMIT

功能特性

客户端连接会话上事务的提交模式，提交模式分为自动提交和非自动提交。

参数说明

参数值	事务提交模式
ON TRUE	自动提交模式
其它值	非自动提交模式

示例

- 设置提交模式

```
SET AUTO_COMMIT ON;
```

- 显示参数状态

```
SHOW AUTO_COMMIT;
```

说明

该参数在驱动程序缺省未指定的情况下，由系统全局参数 `def_auto_commit` 决定，默认自动提交。

7.11 STRICT_COMMIT

功能特性

客户端连接会话上非只读事务严格提交开关。严格提交事务模型下事务提交需等候所有事务的所有 redo 日志落盘写实。

参数说明

参数值	事务是否严格提交模型
ON TRUE	严格提交
OFF FALSE	非严格提交

示例

- 设置提交模式

```
SET STRICT_COMMIT ON;
```

- 显示参数状态

```
SHOW STRICT_COMMIT;
```

说明

该参数在驱动程序缺省未指定的情况下，由系统全局参数 `strictly_commit` 决定；若该参数已经由驱动程序明确指定，则会覆盖系统全局参数 `strictly_commit` 给定的值。

7.12 RESULT

功能特性

各数据类型返回值模式。

参数说明

参数值	通讯协议
CHAR 类型	客户端以字符串形式接收处理
其它值	客户端以类型定义方式接受处理

示例

- 客户端以字符返回

```
SET RESULT TO 'char';
```

- 客户端以类型定义方式返回

```
SET RESULT TO '';
```

- 查看返回类型

```
SHOW RESULT;
```

各类型转字符格式说明如表所示。

编号	类型	格式
1	bool	不支持
2	tinyint	数字串
3	short	数字串
4	int	数字串
5	bigint	数字串
6	float	数字串

接下页

编号	类型	格式
7	double	数字串
8	date	yyy-mm-dd
9	datetime	yyyy-mm-dd hh24:mi:sssss
10	timestamp	yyyy-mm-dd hh24:mi:sssss
11	time	%02d:%02d:%02d (无毫秒) %02d:%02d:%02d:%f (带毫秒)
12	datetime with time zone	yyyy-mm-dd hh24:mi:sssss
13	timestamp with time zone	yyyy-mm-dd hh24:mi:sssss
14	interval year	数字串
15	interval month	数字串
16	interval day	数字串
17	interval hour	数字串
18	interval minute	数字串
19	interval second	%d.%06d
20	interval year to month	%d-%02d (-分隔 year 和 month)
21	interval day to hour	%d %02d (空格分隔 day 和 hour)
22	interval day to minute	%d %02d:%02d
23	interval day to second	%d %02d:%02d:%02d.%06d(带毫秒)
24	interval hour to minute	%d:%d
25	interval hour to second	%d:%d:%.9g
接下页		

编号	类型	格式
26	interval minute to second	%d:%.9g
27	udt	[%s,%s...]

 说明

未列出的类型为本身就是字符串方式或不支持 CHAR 方式返回（如 binary、blob、bool、rowid、char、varchar 等）。

7.13 RETURN_ROWID

功能特性

查询是否默认返回 ROWID 开关。

参数说明

参数值	通讯协议
ON TRUE	返回结果集的同时返回 ROWID 信息
其它值	返回结果集的同时不返回 ROWID 信息

示例

- 设置返回结果集的同时不返回 ROWID 信息

```
SET RETURN_ROWID TO 'false';
```

- 显示参数状态

```
SHOW RETURN_ROWID;
```

 说明

默认 false 不返回。

7.14 RETURN_SCHEMA

功能特性

查询是否返回对象所属的模式名信息。

参数说明

参数值	是否返回对象所属的模式名信息
ON TRUE	返回对象所属的模式名信息
其它值	不返回对象所属的模式名信息

示例

- 设置不返回对象所属的模式名信息

```
SET RETURN_SCHEMA TO 'false';
```

- 显示参数状态

```
SHOW RETURN_SCHEMA;
```

说明

默认 false 不返回。

7.15 RETURN_CURSOR_ID

功能特性

使用游标获取结果集时是否返回游标名。

参数说明

参数值	是否返回游标名
ON TRUE	返回游标名
其它值	不返回游标名

示例

- 设置不返回游标名

```
SET RETURN_CURSOR_ID to 'false';
```

- 显示参数状态

```
SHOW RETURN_CURSOR_ID;
```

📖 说明

默认 `false` 直接返回结果集。

7.16 LOB_RET

功能特性

在获取大对象结果集时，是否将大对象作为描述符返回。

参数说明

参数值	返回大对象结果集行为
ON TRUE	返回大对象描述符
其它值	以二进制的形式返回大对象

示例

- 设置以二进制的形式返回大对象

```
SET LOB_RET TO 'false';
```

- 显示参数状态

```
SHOW LOB_RET;
```

📖 说明

默认 `false` 直接返回数据。

7.17 EMPTY_STR_AS_NULL

功能特性

数据库服务端将结果集发向客户端时，是否将空字符串当作 NULL 处理。

该参数在驱动程序缺省未指定的情况下，由系统全局参数 `def_empty_str_as_null` 决定；若该参数已经由驱动程序明确指定，则会覆盖系统全局参数 `def_empty_str_as_null` 给定的值。

参数说明

参数值	结果
TRUE	将空字符串当作 NULL 处理
FALSE	不将空字符串当作 NULL 处理

示例

- 设置不将空字符串当作 NULL 处理

```
SET EMPTY_STR_AS_NULL TO 'false';
```

- 显示参数状态

```
SHOW EMPTY_STR_AS_NULL;
```

说明

在数据库初始化 ini 文件中存在重名参数，会话参数仅影响 SQL 参数处理（如 prepare 批量插入）。

7.18 OPTIMIZER_MODE

功能特性

优化模式。

参数说明

参数值	代表值
ALL ROWS	0
FIRST ROWS	1

示例

- 设置为 ALL_ROWS

```
SET OPTIMIZER_MODE TO 'ALL_ROWS';
```

- 显示参数状态

```
SHOW OPTIMIZER_MODE;
```

📖 说明

优化模式，默认使用 `xdef_optimize_mode`，默认设置为 0。

7.19 COMPATIBLE_MODE

功能特性

用于兼容其他数据库模式。

参数说明

参数值	运行模式及支持功能
ORACLE	兼容 Oracle：不带双引号标识符转大写
MYSQL	兼容 MySQL： <ul style="list-style-type: none">• 无论标识符是否有双引号转都不做大小写转换• 兼容 MySQL 双引号使用• 无别名的非字段表达式返回名称兼容
POSTGRESQL	兼容 PG：不带双引号标识符转小写
NONE	Xugu：不带双引号标识符转小写

示例

- 设置模式为 NONE

```
SET COMPATIBLE_MODE TO 'NONE';
```

- 显示参数状态

```
SHOW COMPATIBLE_MODE;
```

📖 说明

相关参数 `def_compatible_mode`，连接串未设置时，`COMPATIBLE_MODE` 设置为 `def_compatible_mode`。

7.20 TRANS_READONLY

功能特性

当前会话是否只执行只读事务。

参数说明

参数值	结果
TRUE	是
FALSE	否

示例

- 设置当前会话只执行只读事务。

```
SET TRANS_READONLY TO TRUE;
```

- 显示参数状态

```
SHOW TRANS_READONLY;
```

说明

默认为“false”。

7.21 DISABLE_BINLOG

功能特性

当前会话是否禁用 BINLOG 日志记载。

参数说明

参数值	结果
OFF FALSE	不禁用
ON TRUE	禁用

示例

- 设置不禁用 BINLOG 日志记载

```
SET DISABLE_BINLOG OFF;  
SET DISABLE_BINLOG TO 'FALSE'
```

- 显示参数状态

```
SHOW DISABLE_BINLOG;
```

7.22 IDENTITY_MODE

功能特性

控制自增列（identity）的插入值填充模式，以兼容 MySQL 的自增列处理方式，不指定则由系统全局参数 def_identity_mode 决定。

参数说明

IDENTITY_MODE 参数值	对应 def_identity_mode 参数值	参数功能
<ul style="list-style-type: none">• DEFAULT• DEFAULT_IDENTITY_MODE	0	<ul style="list-style-type: none">• 未显式指定自增列的值，插入自增值• 显式指定自增列的值为 NULL，报错• 显式指定自增列的值为非 NULL 数值，作为数据插入进表（自增列的值具有唯一值约束，多次插入相同值会报错违反唯一值约束）
NULL_AS_AUTO_INCREMENT	1	<ul style="list-style-type: none">• 未显式指定自增列的值，插入自增值• 显式指定自增列的值为 NULL，插入自增值• 显式指定自增列的值为非 NULL 数值，作为数据插入进表（自增列的值具有唯一值约束，多次插入相同值会报错违反唯一值约束）

接下页

IDENTITY_MODE 参数值	对应 def_identity_mode 参数值	参数功能
ZERO_AS_AUTO_INCREMENT	2	<ul style="list-style-type: none"> 未显式指定自增列的值，插入自增值 显式指定自增列的值为 NULL 或 0，插入自增值 显式指定自增列的值为非 NULL 非 0 数值，作为数据插入进表（自增列的值具有唯一值约束，多次插入相同值会报错违反唯一值约束）

示例

- 设置 NULL_AS_AUTO_INCREMENT 模式

```
SET IDENTITY_MODE TO NULL_AS_AUTO_INCREMENT;
```

- 显示参数状态

```
SHOW IDENTITY_MODE;
```

说明

默认模式为"DEFAULT"。

7.23 KEYWORD_FILTER

功能特性

关键字过滤，当设置的过滤关键字出现在特定操作关键字之后（如：

SELECT/INSERT/UPDATE/DELETE/MERGE/CREATE TABLE），则设置的过滤关键字会被作为标识符进行处理。

说明

- CREATE TABLE 之后的关键字处理在 V12.5 以上版本支持。
- 关键字过滤后，只有 DML 语句和创建表的对象可以使用被过滤的关键字，其他对象（如库、序列、过程、函数等）将不能使用这些关键字。

示例

将需要做对象名、列表、别名的参数通过逗号分隔进行设置，禁止使用空格，区分大小写。

连接串参数设置：

```
KEYWORD_FILTER=TABLE  
KEYWORD_FILTER=TABLE,FUNCTION  
KEYWORD_FILTER=TABLE,FUNCTION,CONSTANT
```

SET 命令设置：

```
SET KEYWORD_FILTER TO 'TABLE';  
SET KEYWORD_FILTER TO 'TABLE,FUNCTION';  
SET KEYWORD_FILTER TO 'TABLE,FUNCTION,CONSTANT';
```

可通过 `sys_sessions` 系统表 `KEYWORD_FILTER` 字段和命令 **SHOW KEYWORD_FILTER** 查看。

注意

连接上配置关键字后，此关键字相关 DML 语法将无法使用，如需使用相关语法只能通过无关键字过滤的连接执行。

如将 `SELECT` 语句中 `FROM/GROUP/ORDER/OR/AND` 设置为过滤关键字后，相关 `SELECT` 将无法执行。

8 内存全局变量

8.1 LICENSE_TYPE

参数名称

LICENSE_TYPE

参数解释

LICENSE_TYPE 参数用于说明当前数据库版本类型，在数据库中执行命令“SHOW LICENSE_TYPE”查看。

- 试用版本：1 年时间限制，不会对功能限制。
- 个人版本：有时间限制，不会对功能限制。
- 标准版本：无时间限制，但是会对某些功能限制。
- 企业版本：无时间限制，但是会对某些功能限制。
- 分布式集群版本：无时间限制，但是会对某些功能限制。
- 通用版本：无时间限制，无功能限制。
- 敏捷版本：无时间限制，但是会对某些功能限制。

默认值

无

取值范围

无

访问权限

R

是否全局参数

是

修改生效方式

无

注意事项



LICENSE_TYPE 仅用于说明当前数据库版本类型，不允许修改。

8.2 CLUSTER_FAULT_LEVEL

参数名称

CLUSTER_FAULT_LEVEL

参数解释

CLUSTER_FAULT_LEVEL 参数用于设置集群错误级别。

- 0：无故障
- 1：只读

集群脑裂时，在主 Master 节点执行命令“SET CLUSTER_FAULT_LEVEL TO 0”后，可加入其他节点。

默认值

0

取值范围

[0|1]

访问权限

R/W

是否全局参数

是

修改生效方式

CLUSTER_FAULT_LEVEL 可在系统运行时，通过数据库 SET 命令修改，修改后参数值立即生效。

操作步骤：

1. 在数据库中执行以下命令修改 CLUSTER_FAULT_LEVEL 参数值，根据实际需求在取值范围内选择合适的 value 值。

```
SQL> SET CLUSTER_FAULT_LEVEL TO value
```

2. 在数据库中执行以下命令查看当前参数值。

```
SQL> SHOW CLUSTER_FAULT_LEVEL
```

注意事项

无



成都虚谷伟业科技有限公司

联系电话：400-8886236

官方网站：www.xugudb.com