云数据库 GaussDB V2.0-8.202

兼容性说明

文档版本 01

发布日期 2024-11-05





版权所有 © 华为云计算技术有限公司 2024。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明



HUAWE和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 分布式版 Oracle 兼容性说明	
1.1 Oracle 数据库兼容性概述	1
1.2 SQL 的基本元素	1
1.2.1 数据类型	1
1.2.2 数据类型比较规则	6
1.2.3 字面量	7
1.2.4 格式模型	7
1.2.5 空值	8
1.2.6 注释	8
1.2.7 数据库对象	
1.2.8 数据库对象名称和限定符	11
1.2.9 SQL 语句中的引用架构对象和部件的语法	11
1.3 伪列	11
1.4 操作符	12
1.5 表达式	15
1.6 条件	16
1.7 常见的 SQL DDL 子句	17
1.8 SQL 查询和子查询	23
1.9 PL/SQL 语言	24
1.9.1 PL/SQL 基本语法	24
1.9.2 数据类型兼容性	27
1.9.3 控制语句	27
1.9.4 集合和 Record	29
1.9.5 静态 SQL	
1.9.6 动态 SQL	
1.9.7 Trigger	
1.10 系统函数	46
1.10.1 单行函数	46
1.10.2 其它函数	64
1.11 系统视图	
1.12 高级包	73
2 分布式版 MySQL 兼容性说明	125
2.1 概述	125

2.2 MySQL 兼容性 MySQL 模式	125
2.2.1 MySQL 兼容性 MySQL 模式概述	125
2.2.2 数据类型	125
2.2.2.1 数值数据类型	126
2.2.2.2 日期与时间数据类型	132
2.2.2.3 字符串数据类型	143
2.2.2.4 二进制数据类型	146
2.2.2.5 JSON 数据类型	148
2.2.2.6 数据类型支持的属性	148
2.2.2.7 数据类型转换	149
2.2.3 系统函数	151
2.2.3.1 流量控制函数	151
2.2.3.2 日期和时间函数	154
2.2.3.3 字符串函数	164
2.2.3.4 强制转换函数	168
2.2.3.5 加密函数	169
2.2.3.6 JSON 函数	169
2.2.3.7 聚合函数	171
2.2.3.8 数字操作函数	173
2.2.3.9 其他函数	174
2.2.4 操作符	174
2.2.5 字符集	176
2.2.6 排序规则	176
2.2.7 SQL	177
2.2.7.1 DDL	177
2.2.7.2 DML	186
2.2.7.3 DCL	195
2.2.8 驱动	195
2.2.8.1 JDBC	195
2.2.8.1.1 JDBC 接口参考	195
2.3 MySQL 兼容性 M-Compatibility 模式	196
2.3.1 MySQL 兼容性 M-Compatibility 模式概述	196
2.3.2 数据类型	197
2.3.2.1 数值数据类型	197
2.3.2.2 日期与时间数据类型	199
2.3.2.3 字符串数据类型	201
2.3.2.4 二进制数据类型	204
2.3.2.5 数据类型支持的属性	
2.3.2.6 数据类型转换	
2.3.3 系统函数	
2.3.3.1 流程控制函数	
2.3.3.2 日期和时间函数	

2.3.3.3 字符串函数	220
2.3.3.4 强制转换函数	224
2.3.3.5 加密函数	225
2.3.3.6 比较函数	226
2.3.3.7 聚合函数	227
2.3.3.8 数字操作函数	228
2.3.3.9 其他函数	229
2.3.4 操作符	229
2.3.5 字符集	241
2.3.6 排序规则	241
2.3.7 事务	242
2.3.8 SQL	246
2.3.8.1 关键字	248
2.3.8.2 标识符	248
2.3.8.3 DDL	250
2.3.8.4 DML	267
2.3.8.5 DCL	281
2.3.8.6 其他语句	282
2.3.8.7 用户与权限	283
2.3.8.8 系统表和系统视图	
2.3.9 驱动	292
2.3.9.1 ODBC	
2.3.9.1.1 ODBC 接口参考	292
3 集中式版 Oracle 兼容性说明	294
3.1 Oracle 数据库兼容性概述	294
3.2 SQL 的基本元素	294
3.2.1 数据类型	294
3.2.2 数据类型比较规则	299
3.2.3 字面量	
3.2.4 格式模型	300
3.2.5 空值	301
3.2.6 注释	301
3.2.7 数据库对象	302
3.2.8 数据库对象名称和限定符	304
3.2.9 SQL 语句中的引用架构对象和部件的语法	304
3.3 伪列	
3.4 操作符	305
3.5 表达式	
3.6 条件	309
3.7 常见的 SQL DDL 子句	310
3.8 SQL 查询和子查询	
3.9 PL/SQL 语言	318

3.9.1 PL/SQL 基本语法	318
3.9.2 数据类型兼容性	321
3.9.3 控制语句	322
3.9.4 集合和 Record	324
3.9.5 静态 SQL	328
3.9.6 动态 SQL	331
3.9.7 Trigger	332
3.10 系统函数	
3.10.1 单行函数	340
3.10.2 其它函数	357
3.11 系统视图	359
3.12 高级包	366
4 集中式版 MySQL 兼容性说明	418
4.1 概述	418
4.2 MySQL 兼容性 B 模式	418
4.2.1 MySQL 兼容性 B 模式概述	418
4.2.2 数据类型	418
4.2.2.1 数值数据类型	419
4.2.2.2 日期与时间数据类型	425
4.2.2.3 字符串数据类型	436
4.2.2.4 二进制数据类型	439
4.2.2.5 JSON 数据类型	441
4.2.2.6 数据类型支持的属性	441
4.2.2.7 数据类型转换	442
4.2.3 系统函数	444
4.2.3.1 流量控制函数	445
4.2.3.2 日期和时间函数	447
4.2.3.3 字符串函数	458
4.2.3.4 强制转换函数	464
4.2.3.5 加密函数	465
4.2.3.6 信息函数	465
4.2.3.7 JSON 函数	465
4.2.3.8 聚合函数	467
4.2.3.9 数字操作函数	468
4.2.3.10 其他函数	468
4.2.4 操作符	469
4.2.5 字符集	471
4.2.6 排序规则	471
4.2.7 表达式	472
4.2.8 SQL	472
4.2.8.1 DDL	472
4.2.8.2 DML	482

4.2.8.3 DCL	493
4.2.9 驱动	493
4.2.9.1 JDBC	493
4.2.9.1.1 JDBC 接口参考	494
4.3 MySQL 兼容性 M-Compatibility 模式	494
4.3.1 MySQL 兼容性 M-Compatibility 模式概述	494
4.3.2 数据类型	495
4.3.2.1 数值数据类型	496
4.3.2.2 日期与时间数据类型	499
4.3.2.3 字符串数据类型	502
4.3.2.4 二进制数据类型	505
4.3.2.5 JSON 类型	510
4.3.2.6 数据类型支持的属性	510
4.3.2.7 数据类型转换	511
4.3.3 系统函数	531
4.3.3.1 流程控制函数	532
4.3.3.2 日期和时间函数	533
4.3.3.3 字符串函数	538
4.3.3.4 强制转换函数	544
4.3.3.5 加密函数	546
4.3.3.6 比较函数	547
4.3.3.7 聚合函数	549
4.3.3.8 JSON 函数	556
4.3.3.9 窗口函数	559
4.3.3.10 数字操作函数	560
4.3.3.11 网络地址函数	564
4.3.3.12 其他函数	565
4.3.4 操作符	568
4.3.5 字符集	582
4.3.6 排序规则	583
4.3.7 事务	584
4.3.8 SQL	588
- 4.3.8.1 关键字	590
4.3.8.2 标识符	
4.3.8.3 DDL	
4.3.8.4 DML	
4.3.8.5 DCL	647
4.3.8.6 其他语句	
4.3.8.7 用户与权限	
4.3.8.8 系统表和系统视图	
4.3.9 驱动	
4.3.9.1 ODBC	

4.3.9.1.1 ODBC 接口参考	661
4.3.10 计划外应用无损透明	662
4.3.10.1 流程控制函数	662
4.3.10.2 日期和时间函数	662
4.3.10.3 字符串函数	664
4.3.10.4 强制转换函数	666
4.3.10.5 加密函数	666
4.3.10.6 比较函数	667
4.3.10.7 聚合函数	667
4.3.10.8 JSON 函数	668
4.3.10.9 窗口函数	669
4.3.10.10 数字操作函数	669
4.3.10.11 网络地址函数	
4.3.10.12 其他函数	670

1 ● 分布式版 Oracle 兼容性说明

1.1 Oracle 数据库兼容性概述

GaussDB数据库在基本功能(数据类型、SQL、数据库对象等)和PL/SQL方面与 Oracle数据库基本兼容。但是由于架构设计方面的差异,还是存在一些不兼容的项, 本章节主要介绍GaussDB数据库的Oracle兼容模式与Oracle数据库19c版本的兼容性对 比信息。

1.2 SQL 的基本元素

1.2.1 数据类型

表 1-1 数值类型

序号	Oracle数据 库	GaussDB 数据库	差异
1	NUMBER [(p[, s])]	支持,有差 异	精度和用法存在差异。 NUMBER带参数时,GaussDB的精度p与标度s的最大边界值比Oracle更大。 NUMBER不带参数时,GaussDB的精度p的默认值远大于带参数时的最大边界值;而在Oracle中,精度p的默认值等于带参数时的最大边界值。 GaussDB不支持标度s为负值;在Oracle中,标度s为负值时会精确到相应的整数位。
2	FLOAT [(p)]	支持	-
3	BINARY_FLO AT	不支持	-

序号	Oracle数据 库	GaussDB 数据库	差异
4	BINARY_DO UBLE	支持	-

表 1-2 日期时间类型

序号	Oracle数据库	GaussDB数 据库	差异
1	DATE	支持,有差 异	精度有差异,GaussDB支持的公元 时间相较于Oracle范围更大。
2	TIMESTAMP [(fractional_seconds _precision)]	支持,有差 异	-
3	TIMESTAMP [(fractional_seconds _precision)] WITH TIME ZONE	支持,有差异	GaussDB的timestamptz等价于Oracle的timestampwithloacltimezone,缺少Oracle对应的timestamptz类型。 时区更新:部分国家或地区经常会更新时区信息,数据库系统也因此常常需要同步修改时区文件以确保时间内容的正确性。 GaussDB时区类型目前只涉及timestamp with timezone,当新的时区文件生效时,不会对已有的数据进行变更,新数据会随时区文件信息进行同步调整。与Oracle的同类型数据能力存在差异。
4	TIMESTAMP [(fractional_seconds _precision)] WITH LOCAL TIME ZONE	不支持	-
5	INTERVAL YEAR [(year_precision)] TO MONTH	支持	-
6	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds _precision)]	支持	-

□ 说明

- ORA兼容模式下,DATE类型被替换为TIMESTAMP(0) WITHOUT TIME ZONE,差异同 TIMESTAMP(0) WITHOUT TIME ZONE。
- 对于TIMESTAMP [(fractional_seconds_precision)] / TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE与Oracle存在以下差异。
 - fractional_seconds_precision GaussDB支持的精度范围为0~6,Oracle支持的精度范围为0~9。
 - GaussDB通过DateStyle设置日期和时间值的显示格式,以及有歧义的输入值的解析规则。具体请参见《开发指南》中"SQL参考>数据类型>日期/时间类型"章节中日期输入的说明介绍。Oracle在一般情况下的输入格式校验及输出显示由NLS_TIMESTAMP_TZ_FORMAT参数进行控制。
 - 对于秒的小数部分的显示,GaussDB默认去除结尾的零,Oracle依据格式化参数中的设置(FF/FF1-FF9)进行显示控制。如 '2017-09-01 10:32:19.212000',GaussDB显示为 '2017-09-01 10:32:19.212',Oracle在format参数中含FF时显示为'2017-09-01 10:32:19.212000',在format参数中含FF9时显示为'2017-09-01 10:32:19.212000000'。
 - 支持的时间范围存在差异,GaussDB支持的公元时间较Oracle范围更大。

表 1-3 字符类型

序号	Oracle数据库	GaussDB数 据库	差异
1	VARCHAR2 (size [BYTE CHAR])	支持,有差 异	在GaussDB中,size单位为字节,即仅支持BYTE,不支持在BYTE和CHAR之间选择,最大10MB;而在Oracle中,size的单位可以在BYTE和CHAR之间选择,MAX_STRING_SIZE=EXTENDED时,最大长度为32767字节,MAX_STRING_SIZE=STANDARD时,最大长度为4000字节,实际能容纳的字符数与使用的字符集有关。
2	NVARCHAR2 (size)	支持,有差异	在GaussDB中,NVARCHAR2(n)类型是VARCHAR2(n)类型的别名;而在Oracle中NVARCHAR2(n)与VARCHAR2(n)有一定的区别,MAX_STRING_SIZE=EXTENDED时,最大长度为32767字节,MAX_STRING_SIZE=STANDARD时,最大长度为4000字节,实际能容纳的字符数与使用的字符集有关。
3	CHAR [(size [BYTE CHAR])]	支持,有差 异	在GaussDB中,size单位为字节,即仅支持BYTE,不支持在BYTE和CHAR之间选择,最大10MB;而在Oracle中,size的单位可以在BYTE和CHAR之间选择,最大容量为2000个字节,实际能容纳的字符数与使用的字符集有关。
4	NCHAR [(size)]	支持,有差 异	在GaussDB中,size单位为字节,最大 10MB;而在Oracle中,size单位为字符, 最大容量为2000个字节,实际能容纳的字 符数与使用的字符集有关。

序号	Oracle数据库	GaussDB数 据库	差异
5	CLOB	支持,有差 异	不支持定位器。
6	NCLOB	不支持	-
7	LONG	不支持	-

表 1-4 二进制类型

序号	Oracle数据 库	GaussDB数 据库	差异
1	RAW (size)	支持,有差 异	在GaussDB中,size是指字节长度建议值,不 会用于校验输入raw类型的字节长度。
2	LONG RAW	不支持	-
3	BLOB	支持	-
4	BFILE	不支持。	-

表 1-5 ROWID 类型

序号 Oracle数据库		GaussDB数据库
1	ROWID	不支持
2	UROWID	不支持

表 1-6 用户自定义类型

序号	Oracle数据库	GaussDB数据库
1	对象类型	不支持
2	REF数据类型	不支持
3	可变数组	支持
4	嵌套表	支持

表 1-7 伪类型

序号	Oracle数据库	GaussDB数据库
1	anytype	不支持

序号 Oracle数据库		GaussDB数据库
2	anydata	不支持
3	anydataset	不支持

表 1-8 xml 类型

序号	Oracle数 据库	GaussDB数据 库	差异
1	XMLTYPE	支持,有差 异。	GaussDB中不支持部分操作,如不能通过使用 XMLELEMENT函数将字符串转变为XMLTYPE 类型,而是转变成XML类型。具体请参见《开 发指南》中"SQL参考 > 数据类型 > XMLTYPE类型"章节。
2	URIType	不支持。	-

表 1-9 空间类型

序号	Oracle数据库	GaussDB数据库
1	SDO_GEOMETRY	不支持
2	SDO_TOPO_GEOMETRY	不支持
3	SDO_GEORASTER	不支持

表 1-10 锁模式

序号	Oracle数据库	GaussDB数据库
1	none	-
2	null	AccessShare
3	RS	RowShare
4	RX	RowExclusive
5	S	ShareUpdateExclusive
6	SRX	Share
7	-	ShareRowExclusive
8	Х	Exclusive
9	-	AccessExclusive

序号	Oracle数据库	GaussDB数据库
10	-	INVALID
		说明 GaussDB的INVALID表示被赋予了非法锁。仅 在运行过程中出现了GaussDB无法识别的锁时 会被赋予INVALID锁。

1.2.2 数据类型比较规则

数据类型比较(排序)规则是指相同数据类型的值之间发生比较(排序)时遵循的比较(排序)规则。

表 1-11 比较规则

序号	Oracle数 据库	GaussDB数 据库	差异
1	Numeric 值	支持	-
2	日期时间 值	支持	-
3	二进制值	支持	-
4	字符值	支持,有差异	 在GaussDB和Oracle中,支持的比较规则不完全相同,相同比较规则的名称也可能不同。 GaussDB和Oracle的比较规则在可指定性上有差异,例如GaussDB不支持指定表级别的比较规则,而Oracle支持。 GaussDB和Oracle在指定比较规则的语法上有差异,例如在GaussDB中,使用ENCODING、LC_CTYPE和LC_COLLATE三个参数决定创建数据库时使用的字符集、字符分类和比较规则,具体请参见《开发指南》中"SQL参考 > SQL语法 > C > CREATEDATABASE"章节。而在Oracle中,各级别的比较规则通常由一系列带有NLS前缀的参数确定。
5	对象值	不支持	-
6	Varrays 和嵌套表	支持,有差 异	GaussDB和Oracle均支持Varrays的比较,与 Oracle不同的是,GaussDB不仅支持比较两个 Varrays中的元素个数,还支持同类型的Varrays 之间的比较。
7	数据类型 优先级	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
8	数据转换 (显示/ 隐式类型 转换)	支持	-

1.2.3 字面量

表 1-12 字面量

序号	Oracle数据库	GaussDB数据库
1	文本字面量	支持
2	数值字面量	支持
3	日期时间字面量	支持
4	区间字面量	支持

1.2.4 格式模型

表 1-13 格式

序号	Oracle数 据库	GaussDB数 据库	差异
1	数字格式	支持,有差 异	GaussDB仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下,支持\$、C、TM、TM9、TME、U格式。同时在该参数下,不支持TH、PL、SG格式。GaussDB具体支持情况请参见《开发指南》中"SQL参考>函数和操作符>类型转换函数"章节的"number类型fmt参数表"。
2	日期时间 格式	支持,有差 异	GaussDB中用于时间截断和时间四舍五入的参数,仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。GaussDB具体支持情况请参见《开发指南》中"SQL参考>函数和操作符>时间和日期处理函数和操作符"章节的"用于日期/时间格式化的模式"。
3	格式模型 修饰符	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
4	字符串到日期转换规则	支持,有差 异	GaussDB中to_timestamp_tz函数在参数 a_format_version值为10c和 a_format_dev_version值为s1的情况下有效。 GaussDB具体支持情况请参见《开发指南》 中"SQL参考 > 函数和操作符 > 类型转换函 数"章节的"to_date/to_timestamp/ to_timestamp_tz"。
5	xml格式模 式	不支持	-

1.2.5 空值

表 1-14 空值

序号	Oracle数据库	GaussDB数据库
1	IS NULL和IS NOT NULL	支持
2	NULLS in conditions	支持

1.2.6 注释

表 1-15 comment

序号	Oracle数据库	GaussDB数据库	差异
1	斜杠和星号(/*)	支持	-
2	两个连字符()	支持	-
3	COMMENT命令	支持	-
4	HINT	支持,有差异	GaussDB不支持'+'hint形式。 具体信息请参见《开发指南》中 "SQL调优指南 > 使用Plan Hint进行调优"章节。

1.2.7 数据库对象

表 1-16 schema 对象

序号	Oracle数据 库	GaussDB 数据库	差异
1	分析视图	不支持	-
2	属性维度	不支持	-
3	集群	支持	-
4	约束	支持	-
5	数据库链接	支持	-
6	数据库触发 器	支持	-
7	尺寸	支持	-
8	外部过程库	不支持	-
9	分层结构	不支持	-
10	索引组织表	不支持	-
11	索引	支持	-
12	索引类型	不支持	-
13	java类	不支持	-
14	java资源	不支持	-
15	java源码	不支持	-
16	join groups	不支持	-
17	物化视图	支持	-
18	物化视图日 志	不支持	-
19	挖掘模型	不支持	-
20	对象表	不支持	-
21	对象类型	不支持	-
22	对象视图	不支持	-
23	operators	支持	-
24	包	支持	-
25	序列	支持	-
26	存储函数	支持	-

序号	Oracle数据 库	GaussDB 数据库	差异
27	存储过程	支持	-
28	同义词	支持,有 差异	Oracle数据库的数据库对象在同一namespace 内不能重名。GaussDB同一namespace内同义 词可与表、视图、函数、package重名,如果 重名,会优先访问该名称对象,未寻找到该名 称对象时才会寻找该名称的同义词指向的对 象;同义词指向的对象的Schema名是用户名 时才会搜索PUBLIC同义词。详细的搜索顺序参 考《开发指南》中的"SQL参考 > SQL语法 > C > CREATE SYNONYM"章节中的注意事 项,同义词的搜索过程。
29	表	支持	-
30	视图	支持	-
31	zone map	不支持	-

表 1-17 nonschema 对象

序号	Oracle数据库	GaussDB数据库
1	上下文	不支持
2	目录	支持
3	版本	不支持
4	闪回存档	不支持
5	锁定配置文件	不支持
6	配置文件	不支持
7	还原点	支持
8	角色	支持
9	回滚段	ustore支持回滚段astore不支持回滚段
10	表空间	支持
11	表空间集	不支持
12	统一审计策略	支持
13	用户	支持

1.2.8 数据库对象名称和限定符

表 1-18 命名规则

序号	Oracle数据库	GaussDB数 据库	差异
1	数据库对象命 名规则	支持,有差 异	GaussDB默认小写。
2	模式对象命名 规则	支持	-

1.2.9 SQL 语句中的引用架构对象和部件的语法

表 1-19 对象引用

序号	Oracle数据库	GaussDB数据库
1	引用对象的一般语法	支持
2	解析对象的引用	支持
3	引用外部模式对象	支持
4	引用外部数据库对象	支持
5	引用表和索引分区和子分区	支持

1.3 伪列

GaussDB数据库兼容序列伪列、rownum伪列,其余暂不支持。

分层查询伪列

表 1-20 分层查询伪列

序号	Oracle数据库	GaussDB数据库
1	connect_by_iscycle	支持
2	connect_by_isleaf	支持
3	level伪列	支持

序列伪列

表 1-21 序列

序号	Oracle数 据库	GaussDB数据 库	差异
1	currval	支持,有差异	GaussDB以函数形式实现。兼容Oracle调用 方式。
2	nextval	支持,有差异	GaussDB以函数形式实现。兼容Oracle调用 方式。

rownum 伪列

表 1-22 rownum

序号	Oracle数 据库	GaussDB数 据库	差异
1	rownum	支持,有差 异。	Oracle在left、right、full join的条件中使用rownum进行过滤时,不同的条件下表现不尽相同,可能存在忽略或部分忽略rownum条件的现象,而GaussDB在此情况下则表现为对left、right、full join后的结果进行过滤。

xmldata 伪列

表 1-23 xmldata

序号	Oracle数据库	GaussDB数据库
1	xmldata	不支持

1.4 操作符

GaussDB数据库兼容除分层查询以外的运算符。

SQL 运算符

表 1-24 SQL 运算符

序号	Oracle数据库	GaussDB数据库
1	一元运算符和二元运算符	支持

序号	Oracle数据库	GaussDB数据库
2	运算符优先级	支持

算术运算符

表 1-25 算术运算符

序号	Oracle数据库	GaussDB数据库
1	正负(+-)一元运算符	支持
2	加减(+-)二元运算符	支持
3	乘除(*/)二元运算符	支持

COLLATE 运算符

表 1-26 COLLATE 运算符

序号	Oracle数据库	GaussDB数据库
1	COLLATE collation_name	支持

连接运算符

表 1-27 连接运算符

序号	Oracle数据库	GaussDB数据库
1		支持

分层查询运算符

表 1-28 分层查询运算符

序号	Oracle数据 库	GaussDB数 据库	差异
1	prior	支持,有差 异。	GaussDB中仅支持对普通列调用,不支持 对函数等调用。

序号	Oracle数据 库	GaussDB数 据库	差异
2	connect_by _root	支持,有差 异。	GaussDB中,connect_by_root调用时,用 括号修饰操作值时,行为与Oracle一致; 不使用括号时,仅支持对普通列调用此运 算符。

集合运算符

表 1-29 集合运算符

序号	Oracle数据库	GaussDB数据库
1	union	支持
2	union all	支持
3	intersect	支持
4	minus	支持

多集合运算符

表 1-30 多集合运算符

序号	Oracle数据库	GaussDB数据库
1	multiset except	支持
2	multiset intersect	支持
3	multiset union	支持

用户自定义运算符

表 1-31 用户自定义运算符

序号	Oracle数据库	GaussDB数 据库	差异
1	CREATE OPERATOR	支持	 Oracle中提供的CONTEXT_CLAUSE支持自定义功能评估函数,和GaussDB 约束选择性评估函数不同。GaussDB 不支持自定义功能评估函数。 GaussDB和Oracle可选参数差异较大。GaussDB具体请参考《开发指南》中"SQL参考 > SQL语法 > C > CREATE OPERATOR"章节的参数说明部分。

比较操作符

序号	Oracle数据库	GaussDB数据库	
1	<=	支持	
2	<>	支持	
3	>=	支持	
4	^ =	支持	
5	!=	不支持,!=中间存在空格时,!会被识别为 阶乘。	

当比较操作符(<=、<>、>=、^=)中间存在空格时,也可以识别成没有空格进行正常操作。!=中间存在空格时,!会被识别为阶乘,可能会导致结果与预期不一致。

1.5 表达式

GaussDB数据库兼容大部分数据库表达式。

表 1-32 表达式

序号	Oracle数据库	GaussDB数据 库	差异
1	简单表达式	支持	-
2	分析视图表达式	不支持	-
3	复合表达式	支持	-

序号	Oracle数据库	GaussDB数据 库	差异
4	case表达式	支持	-
5	列表示式	支持	-
6	cursor表达式	不支持	-
7	日期时间表达式	支持,有差异	GaussDB的输出结果中不会带时区信息,Oracle会带有类似PM AMERICA/LOS_ANGELES的时区信息。
8	函数表达式	支持	-
9	区间表达式	部分支持	GaussDB支持形如SELECT INTERVAL '999999999 23:59:59.999' day(9) to second FROM DUAL;的语句,而不支 持形如SELECT(SYSDATE- SYSDATE) DAY TO SECOND FROM DUAL;的语 句。Oracle均支持。
10	json对象访问表 达式	部分支持,有 差异	 GaussDB支持通过 "->'key' "的方式从JSON对象中提取value,而Oracle支持通过 ".key "方式来提取value。 对于JSONARRY对象而言,Oracle支持通过 ".key "方式一次性提取所有key对应的value,但GaussDB目前不支持。
11	 模型表达式	 不支持	-
12	对象表达式	不支持	-
13	占位符表达式	部分支持	对于形如":var"的一般占位符表达式,GaussDB支持,但不支持通过INDICATOR关键字将两个一般占位符表达式结合起来。
14	标量子查询表达 式	支持	-
15	类型构造器表达 式	部分支持	GaussDB不支持在类型构造器前指定 New关键字,而Oracle支持。
16	表达式list	支持	-

1.6 条件

本章节描述常见的条件兼容项,包含比较条件、浮点条件、逻辑条件、模型条件、多集合条件、模式匹配条件、NULL值条件、XML条件、SQL/JSON条件、复合条件、BETWEEN条件、EXISTS条件、IN条件、IS OF TYPE条件,如表1-33所示。

表 1-33 条件

序号	Oracle数据库	GaussDB数据 库	差异
1	比较条件	支持,有差异	语句中存在ANY、SOME、ALL操作符时 存在差异,Oracle支持对list对象进行操 作,而GaussDB中需要将list对象转换成 数组表达式的形式后再进行操作。
2	浮点条件	不支持	-
3	逻辑条件	支持	-
4	模型条件	不支持	-
5	多集合条件	不支持	-
6	模式匹配条件	支持	-
7	NULL值条件	支持	-
8	XML条件	不支持	-
9	SQL/JSON条 件	部分支持,有 差异	● GaussDB不支持IS JSON条件和 JSON_TEXTCONTAINS条件。
			 GaussDB中JSONB_EQ条件等同于 Oracle中JSON_EQUAL条件,但 GaussDB不支持ERROR子句。
			 GaussDB中JSONB_EXISTS条件等同于 Oracle中JSON_EXISTS条件,但 GaussDB不支持ERROR子句、EMPTY 子句和PASSING子句。
10	复合条件	支持	-
11	BETWEEN条 件	支持	-
12	EXISTS条件	支持	-
13	IN条件	支持	-
14	IS OF TYPE条件	不支持	-

1.7 常见的 SQL DDL 子句

本章节描述常见的SQL DDL子句兼容项,包含分配扩展子句、约束、取消分配未使用子句、文件规范、日志记录子句、并行子句、物理属性子句、大小子句、存储子句、聚集函数嵌套,如表1-34所示。

表 1-34 常用 SQL DDL 子句

序号	Oracle数据库	GaussD B数据库	差异
1	分配扩展子句语法: ALLOCATE EXTENT [({SIZE size_clause DATAFILE 'filename' INSTANCE integer })] 例如:创建employees表后,改变表的分配扩展size为10M。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. SQL> ALTER TABLE employees ALLOCATE EXTENT (SIZE 10M); Table altered.	不支持	
2	约束 语法: { inline_constraint out_of_line_constraint inline_ref_constraint out_of_line_ref_constraint } 例如: 创建表staff, 约束子句中指定ID列、 NAME列不为空。 SQL> CREATE TABLE staff(ID INT NOT NULL, NAME char(8) NOT NULL, AGE INT, ADDRESS CHAR(50), SALARY REAL); Table created.	支持	-

序号	Oracle数据库	GaussD B数据库	差异
3	取消分配未使用子句语法: DEALLOCATE UNUSED [KEEP size_clause] 例如:创建employees表,进行了一些插入、删除操作后,希望使用取消分配未使用子句释放employees表未使用的空间。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), JOB_ID NUMBER(38), LAST_NAME VARCHAR2(16)); Table created 进行一些插入、删除操作 SQL> ALTER TABLE employees DEALLOCATE UNUSED; Table altered.	不支持	
4	文件规范 语法: {['filename' 'ASM_filename'] [SIZE size_clause] [REUSE] [autoextend_clause]} {['filename ASM_filename' ('filename ASM_filename' ('filename	不支持	

序号	Oracle数据库	GaussD B数据库	差异
5	日志记录子句 语法: { LOGGING NOLOGGING FILESYSTEM_LIKE_LOGGING }	部分支 持,有 差异	 GaussDB不支持LOGGING约束子 句和FILESYSTEM_LIKE_LOGGING 约束子句。 例如:
			GaussDB创建表,带LOGGING约束子句,语法报错。 gaussdb=# CREATE LOGGING TABLE my_tab(id int, name char(16)); ERROR: syntax error at or near "LOGGING" LINE 1: CREATE LOGGING TABLE my_tab(id int, name char(16));
			GaussDB创建表,带 FILESYSTEM_LIKE_LOGGING约束子句,语法报错。 gaussdb=# CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name char(16)); ERROR: syntax error at or near "FILESYSTEM_LIKE_LOGGING" LINE 1: CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name cha
			● GaussDB仅支持表级的 UNLOGGED约束,不支持列级的 UNLOGGED约束。 例如: GaussDB创建表,带列级 的UNLOGGED约束子句,语法报 错。 gaussdb=# CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16)); ERROR: syntax error at or near "UNLOGGED" LINE 1: CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16))
			● GaussDB仅支持在CREATE TABLE、CREATE TABLE AS、 SELECT INTO语句中使用日志记录 子句。 例如: GaussDB创建 TABLESPACE,带UNLOGGED约 束子句,语法报错。 gaussdb=# CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespace1/tablespace_1'; ERROR: syntax error at or near "TABLESPACE" LINE 1: CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespace ^

序号	Oracle数据库	GaussD B数据库	差异
6	并行子句 语法: { NOPARALLEL PARALLEL [integer] } 例如: 创建表t1,并在 并行子句中指定 PARALLEL 4,意为查询 和更新表t1时最多使用4 个并行进程操作。 SQL> CREATE TABLE t1 (id NUMBER, name VARCHAR2(50)) PARALLEL 4; Table created.	不支持	-
7	物理属性子句 语法: [{ PCTFREE integer PCTUSED integer INITRANS integer storage_clause }]	部持,差异	 GaussDB不支持PCTUSED。例如:执行在表tbl1中创建一个tbl1_ind的索引的SQL语句,并在该语句的物理属性子句中指定索引的空间利用率PCTUSED为20%,GaussDB执行该SQL语句语法报错。gaussdb=# CREATE INDEX tbl1_ind ONtbl1 (name) PCTUSED 20; ERROR: syntax error at or near "PCTUSED" LINE 1: CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20; ^ GaussDB 仅支持在CREATE TABLE、CREATE INDEX语句中使用物理属性子句。例如:尝试从表tbl1中获取数据,创建物化视图tbl1_mv,并在物理属性子句中指定该视图的初始化事务数为30,GaussDB执行该SQL语句语法报错。gaussdb=# CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * from tbl1; ERROR: syntax error at or near "INITRANS" LINE 1: CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * fro ^

序号	Oracle数据库	GaussD B数据库	差异
8	大小子句 语法: integer [K M G T P E] 例如: 创建一个临时表 空间tbs_temp_01,并 在表空间中创建一个临 时数据库文件 templ01.dbf,SQL语句 的大小子句中指定初始 大小是5M,可以自动扩展,并将表空间分配给 表空间组tbs_grp_01。 SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' SIZE 5M AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	不支持	
9	存储子句 语法: STORAGE ({ INITIAL size_clause NEXT size_clause MINEXTENTS integer MAXEXTENTS { integer UNLIMITED } maxsize_clause PCTINCREASE integer FREELISTS integer FREELIST GROUPS integer OPTIMAL [size_clause NULL] BUFFER_POOL { KEEP RECYCLE DEFAULT } FLASH_CACHE { KEEP NONE DEFAULT } (CELL_FLASH_CACHE (KEEP NONE DEFAULT)) ENCRYPT })	部持差异	 Oracle中由STORAGE子句指定存储参数,而GaussDB中由WITH子句指定存储参数。例如: Oracle中创建表my_tab1,在存储子句中指定表初始大小为10M,需要更多空间时每次增加5M的SQL语句如下: SQL> CREATE TABLE my_tab1 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) STORAGE (INITIAL 10M NEXT 5M); Table created. GaussDB中创建表my_tab2,在存储子句中指定存储引擎类型为USTORE的SQL语句如下: gaussdb=# CREATE TABLE my_tab2 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) with (storage_type=ustore); NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "my_tab2_pkey" for table "my_tab2" CREATE TABLE GaussDB中可选的存储参数和 Oracle存在很大差异。GaussDB具体可参见《开发指南》中"SQL参考>SQL语法>C>CREATE TABLE"的参数说明部分,"WITH ({storage_parameter = value} [,])"中描述了CREATE TABLE语句支持的存储参数。

序号	Oracle数据库	GaussD B数据库	差异
10	聚集函数嵌套 例如:创建由sales表的 sales_amount列嵌套聚 集函数MIN()、SUM() 生成的表revenue。 SQL> CREATE TABLE sales(ID INT, SALES_AMOUNT INT); Table created. SQL> INSERT INTO sales VALUES(1, 100); 1 row created. SQL> INSERT INTO sales VALUES (3, 200); 1 row created. SQL> CREATE TABLE revenue as SELECT SUM(MIN(sales_amount)) as total from sales group by sales_amount; Table created.	支持	
	. aste dicated.		

1.8 SQL 查询和子查询

GaussDB数据库兼容除分层查询以外的SQL查询和子查询。

表 1-35 SQL 查询和子查询

序号	Oracle数据库	GaussDB数 据库	差异
1	创建简单查询	支持	-
2	分层查询	不支持	-
3	UNION [ALL], INTERSECT,减 运算符	支持	-
4	查询结果排序	支持,有差 异	GaussDB查询不包含分组,且目标列同时 包含聚集函数和集合返回函数时,不忽略 对集合返回函数列的排序。
5	Joins	支持,有差 异	GaussDB只支持和Oracle相同的Join Types,如left、right、self 、natural、 full outer join等。不支持In-Memory Join Groups等Join Optimizations方法。

序号	Oracle数据库	GaussDB数 据库	差异
6	使用子查询	支持	-
7	嵌套子查询的解 嵌套	支持,有差 异	GaussDB不支持显式指定HASH_AJ或 MERGE_AJ。
8	分布式查询	支持,有差 异	GaussDB需要显式DBLINK查询。
9	聚集函数嵌套	支持	-

1.9 PL/SQL 语言

GaussDB数据库基本兼容的PL/SQL操作符、表达式,控制语句、集合和record等等,不支持预定义的PL/SQL常量和类型、子类型等。

1.9.1 PL/SQL 基本语法

表 1-36 PL/SQL 操作符

序号	Oracle数据库	GaussDB数据库
1	+	支持
2	:=	支持
3	=>	支持
4	%	支持
5	1	支持
6		支持
7		支持
8	1	支持
9	**	不支持
10	(支持
11)	支持
12	:	支持
13	ı	支持
14	<<	支持
15	>>	支持
16	/*	支持

序号	Oracle数据库	GaussDB数据库
17	*/	支持
18	*	支持
19	п	支持
20		支持
21	=	支持
22		支持
23	!=	支持
24	~=	支持
25	Λ=	支持
26	<	支持
27	>	支持
28	<=	支持
29	>=	支持
30	@	支持
31		支持
32	;	支持
33	-	支持

表 1-37 逻辑运算符

序号	Oracle数据库	GaussDB数据库
1	NOT	支持
2	AND	支持
3	OR	支持

表 1-38 比较表达式

序号	Oracle数据库	GaussDB数据库
1	IS [NOT] NULL	支持
2	LIKE	支持
3	BETWEEN	支持

序号	Oracle数据库	GaussDB数据库
4	IN	支持

表 1-39 条件表达式

序号	Oracle数据库	GaussDB数据库
1	simple CASE	支持
2	searched CASE	支持

表 1-40 变量声明相关参数

序号	Oracle数 据库	GaussDB数 据库	差异
1	%TYPE	支持,有差 异	 GaussDB不支持record变量%type。 GaussDB不支持pkg.record变量%type 、
2	%ROWTY PE	支持,有差 异	作为变量类型或者出入参类型。 GaussDB在多个CN的环境下,存储过程中无法声明临时表的%ROWTYPE及%TYPE属性。因为临时表仅在当前session有效,在编译阶段其他CN无法看到当前CN的临时表。故多个CN的环境下,会提示该临时表不存在。 GaussDB不支持view%rowtype、schema.view%rowtype作为出入参类型。 GaussDB不支持package.cursor变量%rowtype作为出入参类型。

1.9.2 数据类型兼容性

表 1-41 其他 PL/SQL 数据类型

序号	Oracle数据库	GaussDB数据 库	差异	
1	CHARACTER	支持,有差异	● GaussDB中字节长度限制为: 1~10485760。	
			● Oracle中字节长度限制为: 1~32767。	
2	VARCHAR	支持,有差异	● GaussDB中字节长度限制为: 1~10485760。	
			● Oracle中字节长度限制为: 1~32767。	
3	STRING	不支持	-	
4	PLS_INTEGER	不支持	GaussDB中可使用INT类型替代。	
5	BINARY_INTE GER	支持	-	

1.9.3 控制语句

表 1-42 条件语句

序号	Oracle数据库	GaussDB数据库
1	IF THEN	支持
2	IF THEN ELSE	支持
3	IF THEN ELSIF	支持
4	simple CASE: CASE selector WHEN selector_value_1 THEN statements_1 WHEN selector_value_2 THEN statements_2 WHEN selector_value_n THEN statements_n [ELSE else_statements END CASE;]	支持

序号	Oracle数据库	GaussDB数据库
5	searched CASE:	支持
	CASE	
	WHEN condition_1 THEN statements_1	
	WHEN condition_2 THEN statements_2	
	WHEN condition_n THEN statements_n	
	[ELSE	
	else_statements	
	END CASE;]	

表 1-43 LOOP 循环语句

序号	Oracle数据库	GaussDB数据库
1	[label] LOOP	支持
	statements	
	END LOOP [label];	
2	EXIT;	支持
3	EXIT WHEN;	支持
4	CONTINUE;	支持
5	CONTINUE WHEN;	支持

表 1-44 FOR 循环语句

序号	Oracle数据库	GaussDB数据 库	差异
1	[label] FOR index IN [REVERSE] lower_boundupper_b ound LOOP	支持,有差异	GaussDB使用REVERSE关键字 时,lower_bound必须大于等于 upper_bound,否则循环体不会 被执行。
	statements END LOOP [label];		
2	EXIT WHEN;	支持	-
3	CONTINUE WHEN;	支持	-

表 1-45 WHILE LOOP 循环语句

序号	Oracle数据库	GaussDB数据库
1	[label] WHILE condition LOOP	支持
	statements	
	END LOOP [label];	

表 1-46 GOTO 语句

序号	Oracle数据库	GaussDB数据库
1	GOTO	支持

表 1-47 NULL 语句

序号	Oracle数据库	GaussDB数据库
1	NULL	支持

1.9.4 集合和 Record

表 1-48 类型

序号	Oracle数据库	GaussDB数据库
1	Associative array (or index-by table)	支持
2	VARRAY (variable-size array)	支持
3	Nested table	支持
4	record	支持

表 1-49 语法

序号	Oracle数据库	GaussDB数 据库	差异
1	Associative array (or index-by table)语法: TABLE OF datatype [NOT NULL] INDEX BY { PLS_INTEGER BINARY_INTEGER VARCHAR2 (v_size) data_type }	支持,有差	 GaussDB不支持 PLS_INTEGER类型,GaussDB 内data_type可以为基础数据 类型、或存储过程内定义的 record类型、集合类型、数组 类型,不支持ref cursor类 型。 GaussDB内NOT NULL只支持 语法不支持功能,即不会校验 元素是否为NULL。 详情可参考《开发指南》中 "存储过程 > 数组、集合和 record > 集合"章节。
2	VARRAY (variable-size array)语法: { VARRAY [VARYING] ARRAY } (size_limit) OF datatype [NOT NULL]	支持,有差异	 GaussDB内不支持NOT NULL 语法。 GaussDB内不支持datatype为 varray类型(varray不能嵌套)。 size_limit功能生效需要在 behavior_compat_optionsGU C参数中开启varray_compat 参数。 详情可参考《开发指南》中 "存储过程 > 数组、集合和 record > 数组"章节。
3	Nested table语法: TABLE OF datatype [NOT NULL]	支持,有差 异	 GaussDB内NOT NULL只支持 语法不支持功能。 详情可参考《开发指南》中 "存储过程 > 数组、集合和 record > 集合"章节。
4	record语法: TYPE record_type IS RECORD (field_definition [, field_definition]);	支持	 record的列可以定义为NOT NULL属性也可以指定默认值。其他类型嵌套record类型,record类型的默认值和NOT NULL不生效;通过package.record_type访问类型的形式来创建record变量,该record变量的默认值和NOT NULL不生效。 详情可参考《开发指南》中"存储过程>数组、集合和record > record"章节。

表 1-50 构造器

序号	Oracle数据库	GaussDB数据库
1	collection_type ([value [, value]])	支持

表 1-51 变量赋值

序号	Oracle数据库	GaussDB数 据库	差异
1	Associative array (or index-by table)	支持	-
2	VARRAY (variable- size array)	支持,有差异	 GaussDB不同VARRAY类型的数据可以相互赋值,取决于其元素之间是否能相互隐式转换。 详情可参考《开发指南》中"存储过程>数组、集合和record>数组"章节。
3	Nested table	支持	-
4	record	支持,有差 异	 GaussDB不同record类型的数据可以相互赋值,取决于列与列之间是否能相互隐式转换。 详情可参考《开发指南》中"存储过程>数组、集合和record>record"章节。

表 1-52 集合操作符

序号	Oracle数据库	GaussDB数 据库	差异
1	=	支持,有差 异	Oracle: 比较时忽略集合成员先后顺序。
			• GaussDB: 比较时严格按照集合成员先 后顺序。
2	<>	支持,有差 异	Oracle: 比较时忽略集合成员先后顺序。
			• GaussDB: 比较时严格按照集合成员先 后顺序。
3	IS[NOT] NULL	支持	-

序号	Oracle数据库	GaussDB数 据库	差异
4	^=	支持,有差异	Oracle: 比较时忽略集合成员先后顺序。GaussDB: 比较时严格按照集合成员先后顺序。
5	~=	不支持	-
6	IS[NOT] A SET	不支持	-
7	IS [NOT] EMPTY	不支持	-
8	expr [NOT] MEMBER [OF] nested_table	不支持	-
9	nested_table1 [NOT] SUBMULTISET [OF] nested_table2	不支持	-
10	[NOT] IN	支持	Oracle:比较时忽略集合成员先后顺序。GaussDB:比较时严格按照集合成员先后顺序。

表 1-53 集合 MULTISET 函数

序号	Oracle数据库	GaussDB数据库
1	MULTISET UNION [ALL DISTINCT]	支持
2	MULTISET EXCEPT [ALL DISTINCT]	支持
3	MULTISET INTERSECT [ALL DISTINCT]	支持

表 1-54 集合类型函数

序号	Oracle数据库	GaussDB数据库	差异
1	exists(idx)	支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	extend[(count[, idx])]	支持,有差异	GaussDB仅支持nesttable类型。
3	delete[(idx1[, idx2])]	支持	-
4	trim[(n)]	支持,有差异	GaussDB仅支持nesttable类型。
5	count	支持	-
6	first	支持	-
7	last	支持	-
8	prior(idx)	支持	-
9	next(idx)	支持	-
10	limit	支持,有差异	GaussDB仅支持nesttable类型。

表 1-55 record 变量操作

序号	Oracle数据库	GaussDB数据库
1	构造器	支持
2	%ROWTYPE声明变量	支持
3	定义常量constant	不支持

表 1-56 集合相关函数

序号	Oracle数据库	GaussDB数据 库	差异
1	unnest_table(anynesttable)	支持	-
2	unnest_table(anyindexbytable)	支持	-
3	table(anyarray)	不支持	GaussDB使用 unnest(anyarray)函数 进行等价改写。

1.9.5 静态 SQL

表 1-57 静态查询 SQL 语句

序号	Oracle数 据库	GaussDB数 据库	差异
1	SELECT	支持,有差 异	GaussDB和Oracle在某些场景下有不同。 GaussDB中FOR SHARE对检索出来的行加共享锁,不同事务的共享锁不会互相阻塞。若数据在一个事务中被FOR SHARE锁定,在另一个事务中使用SELECT FOR SHARE SKIP LOCKED时,SKIPLOCKED不会跳过锁。

表 1-58 静态 DML SQL 语句

序号	Oracle数 据库	GaussDB 数据库	差异	
1	INSERT	支持	-	
2	UPDATE	支持	-	
3	DELETE	支持	-	
4	MERGE	支持	-	
5	LOCK TABLE	支持	-	
6	INSERT ALL	支持,有差 异	● Oracle不支持对into_clause的表设置别名, GaussDB支持。	
			● into_clause指定sequence:	
			– Oracle:首次引用nextval会生成下一个数 字,但所有非首次引用的nextval都将返回相 同数字。	
			– GaussDB:引用nextval生成的数字可以正常 自增。	
			● Oracle设置plan_hint语句可以正常生效, GaussDB不生效。	

表 1-59 静态 TCL SQL 语句

序号	Oracle数 据库	GaussDB 数据库	差异
1	COMMIT	支持	-

序号	Oracle数 据库	GaussDB 数据库	差异
2	ROLLBAC K	支持	-
3	SAVEPOI NT	支持	-
4	SET TRANSAC TION	支持,有差 异	GaussDB不支持NAME string语法、USE ROLLBACK SEGMENT rollback_segment语法。

表 1-60 伪列

序号	Oracle数据库	GaussDB数 据库	差异
1	CURRVAL and NEXTVAL	支持	-
2	LEVEL	不支持	-
3	OBJECT_VALU E	不支持	-
4	ROWID	不支持	-
5	ROWNUM	支持,有差 异	不推荐ROWNUM条件用于JOIN ON子句。 GaussDB中ROWNUM条件用于JOIN ON子句时在LEFT JOIN、RIGHT JOIN、FULL JOIN场景下和MERGE INTO场景下与其他数据库行为不一致,直接进行业务迁移存在风险。

表 1-61 隐式游标属性

序号	Oracle数据库	GaussDB数 据库	差异
1	SQL%FOUND	支持,有差 异	GaussDB在commit\rollback之后不刷新隐式游标结果,Oracle会在commit\rollback
2	SQL %NOTFOUND	支持,有差 异	之后刷新隐式游标结果。
3	SQL %ROWCOUNT	支持,有差 异	
4	SQL%ISOPEN	支持,有差 异	

序号	Oracle数据库	GaussDB数 据库	差异
5	SQL %BULK_ROWC OUNT	不支持	
6	SQL %BULK_EXCEP TIONS	不支持	

表 1-62 显式游标语法及关键字

序号	Oracle数据库	GaussDB数 据库	差异
1	CURSOR cursor_name [parameter_list] RETURN return_type;	支持	1
2	CURSOR cursor_name [parameter_list] [RETURN return_type] IS select_statement;	支持	-
3	OPEN	支持	-
4	CLOSE	支持,有差异	GaussDB在exception内部 会自动关闭,Oracle在 exception内部不会自动关 闭。
5	FETCH	支持	-
6	CURRENT OF CURSOR	支持	-

表 1-63 显式游标属性

序号	Oracle数据库	GaussDB数据库
1	SQL%FOUND	支持
2	SQL%NOTFOUND	支持
3	SQL%ROWCOUNT	支持
4	SQL%ISOPEN	支持

表 1-64 游标循环

序号	Oracle数据库	GaussDB数据库
1	FOR LOOP	支持

表 1-65 自治事务支持场景

序号	Oracle数据库	GaussDB数据库
1	存储过程	支持
2	匿名块	支持
3	函数	支持
4	Package	支持

1.9.6 动态 SQL

表 1-66 动态 SQL 语句执行方式

序号	Oracle数据库	GaussDB数 据库	差异
1	EXECUTE IMMEDIATE	支持,有差 异	 GaussDB通过dynamic_sql_compat参数 控制同名变量是否读取同一参数,并且 检查调用存储过程时绑定参数出入参类 型是否与语句参数类型一致。
			GaussDB不支持调用匿名块中部分绑定参数场景,例如匿名块中嵌套动态语句,使用表达式绑定参数,具体请参见《开发指南》中"存储过程 > 动态语句> 动态调用匿名块"章节。
			● GaussDB不支持RETURNING、RETURN INTO。
2	OPEN FOR FETCH CLOSE	支持	-

1.9.7 Trigger

表 1-67 trigger 类型

序号	Oracle数据 库	GaussDB数据 库	差异
1	DML TRIGGER	支持,有差异	GaussDB不支持Compound DML Triggers。
2	SYSTEM TRIGGER	不支持	-

表 1-68 create trigger

序号	Oracle数据库	GaussDB数 据库	差异
1	create语法: CREATE [OR REPLACE] [EDITIONABLE NONEDITIONABLE] TRIGGER plsql_trigger_source	支持,有差 异	GaussDB不支持OR REPLACE 、 EDITIONABLE NONEDITIONABLE,支持 plsql_trigger_source部分行为。
2	plsql_trigger_source ::= 语法: [schema.] trigger_name [sharing_clause] [default_collation_clause] { simple_dml_trigger instead_of_dml_trigger compound_dml_trigger system_trigger }	支持,有差 异	GaussDB不支持schema、 sharing_clause、 default_collation_clause。

序号	Oracle数据库	GaussDB数 据库	差异
3	simple_dml_trigger ::= 语法: { BEFORE AFTER } dml_event_clause [referencing_clause] [FOR EACH ROW] [trigger_edition_claus e] [trigger_ordering_clau se] [ENABLE DISABLE] [WHEN (condition)] trigger_body	支持,有差异	GaussDB不支持 referencing_clause、 referencing_clause(用from referencing_table代替)、 trigger_edition_clause、 trigger_ordering_clause、 ENABLE DISABLE, 支持 trigger_body部分行为。 GaussDB在没有INSTEAD OF TRIGGER的视图上创建语句级 BEFORE/AFTER TRIGGER时不会 报错,执行DML时报错。
4	dml_event_clause ::=语 法: { DELETE INSERT UPDATE [OF column [, column]] } [OR { DELETE INSERT UPDATE [OF column [, column]] } ON [schema.] { table view }	不支持	-
5	trigger_body ::=语法: { plsql_block CALL routine_clause }	支持,有差 异	GaussDB不支持plsql_block,仅支持EXECUTE PROCEDURE function_name (arguments);方式执行function,并且function需要用户定义,必须声明为不带参数并返回类型为触发器,在触发器触发时执行。

序号	Oracle数据库	GaussDB数 据库	差异
6	instead_of_dml_trigger ::=语法: INSTEAD OF { DELETE INSERT UPDATE } [OR { DELETE INSERT UPDATE }] ON [NESTED TABLE nested_table_column OF] [schema.] noneditioning_view [referencing_clause] [FOR EACH ROW] [trigger_edition_claus e] [trigger_ordering_clau se] [ENABLE DISABLE] trigger_body	支持,有差异	GaussDB不支持NESTED TABLE nested_table_column OF、 referencing_clause、 trigger_edition_clause、 trigger_ordering_clause、 ENABLE DISABLE。
7	compound_dml_trigger ::=语法: CREATE trigger FOR dml_event_clause ON view COMPOUND TRIGGER INSTEAD OF EACH ROW IS BEGIN statement; END INSTEAD OF EACH ROW;	不支持	-
8	system_trigger ::=语法: { BEFORE AFTER INSTEAD OF } { ddl_event [OR ddl_event] database_event [OR database_event] } ON { [schema.] SCHEMA [PLUGGABLE] DATABASE } [trigger_ordering_clause] [ENABLE DISABLE] trigger_body	不支持	

表 1-69 alter trigger

序号	Oracle数据库	GaussDB 数据库	差异
1	ALTER TRIGGER [schema.] trigger_name { trigger_compile_clause { ENABLE DISABLE } RENAME TO new_name { EDITIONABLE NONEDITIONABLE } };	支持,有差 异	GaussDB不支持schema、 trigger_compile_clause、 { ENABLE DISABLE }、 { EDITIONABLE NONEDITIONABLE }。

表 1-70 drop trigger

序号	Oracle数据库	GaussDB数 据库	差异
1	DROP TRIGGER [schema.] trigger ;	支持,有差 异	GaussDB不支持schema,需要在 trigger_name后面加上ON table_name。

Oracle数据库名为*_TRIGGERS的视图统计了trigger的相关信息,GaussDB相关视图与Oracle存在差异,GaussDB视图具体请参见《开发指南》中"系统表和系统视图 > 系统视图 > 其他系统视图"中的DB_TRIGGERS、ADM_TRIGGERS、MY_TRIGGERS章节。

表 1-71 Nested, Package, and Standalone Subprograms 兼容性

序号	Oracle数据库	GaussDB数 据库	差异
1	nested subprogram(子块)	支持,有差 异	GaussDB不支持重载。 GaussDB不支持定义为自治事务。 GaussDB不支持SETOF的使用,仅限一个限定符引用嵌套子程序或嵌套子程序的变量。
2	package subprogram	支持	-

序号	Oracle数据库	GaussDB数 据库	差异
3	standalone subprogram (包含Function & Procedure)	支持	-
4	匿名块	支持	-

表 1-72 RETURN 语句支持情况

序号	Oracle数据库	GaussDB数据库
1	Function	支持
2	Procedure	支持
3	匿名块	支持

表 1-73 Function 相关参数

序号	Oracle数据库	GaussDB数 据库	差异
1	DETERMINIS TIC	支持,有差 异	GaussDB中为IMMUTABLE。
2	PARALLEL_E NABLE	不支持	-
3	PIPELINED	不支持	-
4	RESULT_CAC HE	不支持	-

表 1-74 参数形式支持

序号	Oracle数据库	GaussDB数据库
1	IN	支持
2	OUT	支持
3	IN OUT	支持

表 1-75 CREATE 语句

序号	Oracle数据库	GaussDB 数据库	差异
1	CREATE FUNCTION	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、不支持 sharing_clause、仅支持部分指定function属 性的子句(属性的子句仅支持 invoker_rights_clause子句)、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > C > CREATE FUNCTION "章节。
2	CREATE LIBRARY	不支持	-
3	CREATE PACKAGE	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、不支持 sharing_clause、仅支持部分指定package属性的子句(属性的子句仅支持 invoker_rights_clause子句)、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > C > CREATE PACKAGE "章节。
4	CREATE PACKAGE BODY	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、不支持 sharing_clause、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > C > CREATE PACKAGE "章节。
5	CREATE PROCEDURE	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、不支持 sharing_clause子句、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > C > CREATE PROCEDURE"章节。
6	CREATE TRIGGER	支持,有 差异	GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > C >CREATE TRIGGER "章节。
7	CREATE TYPE	支持,有 差异	GaussDB不支持varray、object类型、UNDER 语法。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > C >CREATE TYPE "章 节。
8	CREATE TYPE BODY	不支持	-

表 1-76 ALTER 语句

序号	Oracle数据 库	GaussDB 数据库	差异
1	ALTER FUNCTION	支持,有差	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、SETTINGS、 DEBUG。
			GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > A > ALTER FUNCTION "章节。
2	ALTER LIBRARY	不支持	-
3	ALTER PACKAGE	支持,有差 异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、SETTINGS、DEBUG。 GaussDB的具体语法请参见《开发指南》中"SQL参考 > SQL语法 > A > ALTER
			PACKAGE "章节。
4	ALTER PROCEDURE	支持,有差 异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、SETTINGS、 DEBUG。
5	ALTER	支持,有差	GaussDB仅支持修改trigger名字。
	TRIGGER	异	GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > A > ALTER TRIGGER "章节。
6	ALTER TYPE	支持,有差 异	GaussDB仅支持部分语句。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > A > ALTER TYPE "章 节。

表 1-77 DROP 语句

序号	Oracle数据 库	GaussDB 数据库	差异
1	DROP FUNCTION	支持	-
2	DROP LIBRARY	不支持	-
3	DROP PACKAGE	支持	-
4	DROP PROCEDURE	支持	-

序号	Oracle数据 库	GaussDB 数据库	差异
5	DROP TRIGGER	支持,有 差异	GaussDB的语法与Oracle不同。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > D > DROP TRIGGER "章节。
6	DROP TYPE	支持,有 差异	GaussDB不支持关键字FORCE、VALIDATE。 GaussDB的具体语法请参见《开发指南》中" SQL参考 > SQL语法 > D > DROP TYPE "章 节。
7	DROP TYPE BODY	不支持	-

表 1-78 Function、Procedure、匿名块相关关键字

序号	Oracle数据 库	GaussDB 数据库	差异
1	ACCESSIBLE BY	不支持	-
2	AGGREGATE	支持,有 差异	 GaussDB不支持Oracle的aggregate using [schema.] implementation_type用法。 GaussDB的具体用法请参见《开发指南》中"SQL参考 > SQL语法 > C > CREATE AGGREGATE"章节。 语法不同,但实现功能相同。
3	DETERMINIS TIC	支持,有 差异	GaussDB仅在语法上支持关键字 DETERMINISTIC,未实现功能。
4	PIPE ROW	不支持	-
5	PIPELINED	不支持	-
6	SQL_MACRO	不支持	-
7	RESTRICT_RE FERENCES	不支持	-
8	INLINE	不支持	-

表 1-79 异常处理相关关键字

序号	Oracle数据 库	GaussDB 数据库	差异
1	EXCEPTION_ INIT	支持,有差 异	GaussDB不支持与系统错误码进行绑定。
2	Exception	支持	-
3	Exception Handler	支持	-
4	SQLCODE	支持	-
5	SQLERRM	支持	-

表 1-80 其他 PL/SQL 关键字

序号	Oracle数据库	GaussDB数据库
1	COVERAGE	不支持
2	COLLATION	支持
3	DEPRECATE	不支持
4	FORALL	支持
5	NOCOPY	不支持
6	RETURNING	支持
7	SERIALLY_REUSABLE	不支持
8	SHARING	不支持
9	BULK COLLECT	支持

1.10 系统函数

兼容函数分为:单行函数、用户自定义函数、AGG函数、分析函数、对象引用函数、 模型函数、OLAP函数。

1.10.1 单行函数

序号	Oracle数据库	GaussDB数据库
1	数值函数	支持,有差异
2	返回字符值的字符函数	支持,有差异
3	返回数值的字符函数	支持,有差异

序号	Oracle数据库	GaussDB数据库
4	字符集函数	不支持
5	Collation函数	不支持
6	日期时间函数	支持,有差异
7	通用比较函数	支持,有差异
8	转换函数	支持,有差异
9	大对象函数	支持,有差异
10	集合函数	不支持
11	层次函数	支持
12	数据挖掘功能函数	不支持
13	XML类型函数	支持,有差异
14	JSON函数	不支持
15	编码解码函数	支持,有差异
16	空值相关的函数	支持
17	环境和标识符函数	支持,有差异

表 1-81 数值函数

序号	Oracle数 据库	GaussDB数 据库	差异
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	-

序号	Oracle数 据库	GaussDB数 据库	差异
13	LOG	支持	-
14	MOD	支持,有差 异	 返回类型不一致,Oracle数据库类型包括BINARY_DOUBLE,BINARY_FLOAT,NUMBER;GaussDB返回类型包括int2,int4,int8,numeric。 当第一个入参为数值类型时,第二个参数必须为int、numeric类型或能够转换为numeric的类型。在a_format_version为10c,a_format_dev_version为s6时,当第一个参数为能够转为numeric的text类型时,第二个参数必须为不超过int4的数值类型。
15	NANVL	支持,有差 异	GaussDB不支持直接声明或浮点数除0得到 NaN。
16	POWER	支持	-
17	REMAIND ER	支持,有差 异	返回值数据类型不同。 GaussDB: 当一个输入是float4时,另一个是numeric 时,返回float4类型。 当两个输入都是float4时,返回float4类型。 当两个输入都是float8时,返回float8类型。 其他数据类型,返回numeric Oracle: 返回值类型是number。
18	ROUND	支持,有差 异	 第一个参数n的float类型,GaussDB存在精度 损失,比Oracle数据库精度低。 返回类型不一致。round(n, integer)形式, Oracle数据库NUMBER类型,GaussDB返回 numeric类型;round(n)形式,Oracle数据库 n的数据类型,GaussDB只能返回float8和 numeric类型,缺少float4返回类型。 GaussDB判断入参有null,执行框架返回null 的逻辑与Oracle数据库不一致。 SELECT round(NULL,'q'); Oracle数据库null,GaussDB报错invalid input syntax for integer: "q"。
19	SIGN	支持	-
20	SIN	支持	-
21	SINH	支持	-
22	SQRT	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
23	TAN	支持	-
24	TANH	支持,有差 异	返回值数据类型不同。 GaussDB: ● 当输入是float8时,返回float8类型。 ● 当输入是numeric时,返回numeric类型。 Oracle: 返回值类型是number。
25	TRUNC	支持	-
26	WIDTH_B UCKET	支持	-

表 1-82 返回字符值的字符函数

序号	Oracle数 据库	GaussDB数 据库	差异
1	CHR	支持,有差 异	输入的数字不符合现有字符集时,在JDBC下 GaussDB会报错,Oracle数据库会返回乱 码。
			● 输入0、256等时Oracle数据库会返回Ascii码 为0的字符,GaussDB会在'\0;处截断。
2	CONCAT	支持	-
3	INITCAP	支持,有差 异	返回值受限于数据库字符集,导致返回结果与 Oracle数据库不一致。
4	LOWER	支持,有差 异	返回值类型不一致,Oracle数据库和输入类型一致的数据类型。
			 对时间格式上隐式转换问题,输入时间类型时,隐式转换为字符串再进行lower操作。 SELECT LOWER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle返回10-DEC-12,GaussDB返回2012-12-10 00:00:00。
			返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
5	LPAD	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
6	LTRIM	支持,有差	返回值类型不一致。
		异 -	当输入为字符数据类型时,Oracle返回 VARCHAR2类型;输入为数据库创建时指定的 国家字符集时,Oracle返回NVARCHAR2类型; 输入为LOB类型时,Oracle返回LOB类型, GaussDB返回TEXT类型。
7	NCHR	支持,有差	● 返回值字节长度与Oracle数据库不一致。
		异 	返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
			● 返回入参对应的字节数组时,单个字节在 [0x80-0xFF]范围,会返回"?",Oracle数 据库返回"?"、或者不输出、或者会报错。
8	NLS_LOW ER	支持,有差异	 返回值类型不同,输入为字符数据类型时, Oracle返回VARCHAR2类型;输入为LOB类型时,Oracle返回LOB类型,GaussDB返回TEXT类型。
			 nlsparam参数Oracle数据库还可以传入除 nls_sort外的其他参数种类而不报错, GaussDB只支持nls_sort。
			返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
9	NLS_UPPE R	支持,有差 异	 返回值类型不同,输入为字符数据类型时, Oracle返回VARCHAR2类型;输入为LOB类型时,Oracle返回LOB类型,GaussDB返回TEXT类型。
			 nlsparam参数Oracle数据库还可以传入除 nls_sort外的其他参数种类而不报错, GaussDB只支持nls_sort。
			返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
10	NLSSORT	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异	
11	REGEXP_R EPLACE	支持, 有差异	 GaussDB入参source_char不支持NCLOB类型。 入参match_param选项 'n' 含义有差异: GaussDB中 'n' 选项与 'm' 选项含义相同,表示采用多行模式匹配; 而 Oracle 表示 (.) 能匹配 '\n' 字符,没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项,不需要指定选项。 不同正则表达式匹配结果可能不一致。SELECT REGEXP_REPLACE('abc01234xyz', '(*?)(\d+)(.*)', '#', 'g') FROM DUAL; Oracle报错,GaussDB返回####xyz。 在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。 包含部分转义字符的正则表达式匹配结果可能不一致。SELECT REGEXP_REPLACE('abcabc', '\abc', '#', 'g') FROM DUAL; Oracle报错,GaussDB返回abcabc。 匹配规则受aformat_regexp_match参数影响,具体影响规格请参见《开发指南》中"SQL参考 > 函数和操作符 > 字符处理函数和操作符"章节REGEXP_REPLACE函数部分。 	
12	REGEXP_S UBSTR	支持,有差异	匹配规则受aformat_regexp_match参数影响, 具体影响规格请参见《开发指南》中"SQL参考 > 函数和操作符 > 字符处理函数和操作符"章 节REGEXP_SUBSTR函数部分。	
13	REPLACE	支持	-	
14	RPAD	支持	-	
15	RTRIM	支持	-	
16	SUBSTR	支持	-	
17	TRANSLAT E	支持	-	
18	TRIM	支持	-	

序号	Oracle数 据库	GaussDB数 据库	差异
19	UPPER	支持,有差 异	返回值类型不一致,Oracle数据库和输入类型一致的数据类型,GaussDB返回TEXT类型。
			 对时间格式上隐式转换问题,输入时间类型时,隐式转换为字符串再进行upper操作。 SELECT UPPER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle返回10-DEC-12,GaussDB返回 2012-12-10 00:00:00。
			● 返回值受限于数据库字符集,导致返回结果 与Oracle数据库不一致。
20	INSTRB	支持	-

表 1-83 返回数值的字符函数

序号 Orac 据库	le数	GaussDB数 据库	差异
1 ASCI		支持,有差 异	返回值类型不同。Oracle数据库返回类型为 uint4,GaussDB为int4。
2 INST	R	支持	-
3 LENC	STH	支持	-
4 REGE OUN	XP_C T	支持,有差异	 GaussDB入参source_char不支持NCLOB类型。 入参match_param选项 'n' 含义有差异: GaussDB中 'n' 选项与 'm' 选项含义相同,表示采用多行模式匹配; 而 Oracle 表示(.)能匹配 '\n' 字符,没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中(.)默认匹配 '\n' 选项,不需要指定选项。 不同正则表达式匹配结果可能不一致。 在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。 包含部分转义字符的正则表达式匹配结果可能不一致。 匹配规则受aformat_regexp_match参数影响,具体影响规格请参见《开发指南》中"SQL参考>函数和操作符>字符处理函数

序号	Oracle数 据库	GaussDB数 据库	差异
5	REGEXP_I NSTR	支持,有差异	匹配规则受aformat_regexp_match参数影响, 具体影响规格请参见《开发指南》中"SQL参考 > 函数和操作符 > 字符处理函数和操作符"章 节REGEXP_INSTR函数部分。
6	LENGTHC	支持	-

表 1-84 日期时间函数

序号	Oracle数据库	GaussDB数 据库	差异
1	ADD_MONTH S	支持,有差异	公元后到公元前,GaussDB会和Oracle 数据库相差1年。
			● GaussDB的计算结果范围可以到-4714 年,Oracle数据库只到-4713年。
2	CURRENT_DA TE	支持,有差异	GaussDB不支持nls_date_format参数设置时间显示格式。
3	CURRENT_TIM ESTAMP	支持,有差异	Oracle数据库参数支持范围(0~9)。 GaussDB支持范围(0~6),微秒末位零 不显示。
4	DBTIMEZONE	支持,有差异	GaussDB不支持自带tz的timestamp类型 接口的调用。
5	EXTRACT	支持	-
6	LAST_DAY	支持,有差异	返回值类型不一致。
7	LOCALTIMEST	支持,有差异	Oracle数据库参数支持范围(0~9)。
	AMP		GaussDB支持范围(0 ~ 6),微秒末位零 不显示。
8	MONTHS_BET WEEN	支持,有差异	入参类型不一致。
9	NEW_TIME	支持,有差异	new_time函数的第一个入参为字面量时, 字面量的格式以及函数的返回值类型均与 Oracle数据库不一致。
10	NEXT_DAY	支持	-
11	NUMTODSINT ERVAL	支持,有差异	GaussDB不支持dsinterval类型,暂时用 interval兼容dsinterval类型。
12	NUMTOYMIN TERVAL	支持,有差异	GaussDB不支持yminterval类型,暂时用 interval兼容yminterval类型。

序号	Oracle数据库	GaussDB数 据库	差异
13	SESSIONTIME ZONE	支持,有差异	 赋值语法差异。GaussDB为set session time zone 8。Oracle为alter session set time_zone= '+08:00'。 默认值差异。GaussDB为时区名称形式如:PRC。Oracle为偏移量形式,如:+08:00。
14	SYS_EXTRACT_ UTC	支持	-
15	SYSDATE	支持,有差异	返回值类型不一致。
16	SYSTIMESTAM P	支持,有差异	GaussDB毫秒计算只支持6位,Oracle数 据库支持9位。
17	TO_CHAR	支持,有差异	fmt'5'未在Oracle数据库文档中,未适配。
18	TO_DSINTERV AL	支持,有差异	GaussDB不支持dsinterval类型,暂时用 interval兼容dsinterval类型。
19	TO_TIMESTA MP	支持,有差异	GaussDB毫秒计算只支持6位,Oracle数 据库支持9位。
20	TO_TIMESTA MP_TZ	支持,有差异	GaussDB的timestamptz等价于Oracle的 timestampwithloacltimezone,缺少 Oracle对应的timestamptz类型。 nls_date_language只支持ENGLISH和 AMERICAN两种语言。
21	TO_YMINTERV AL	支持,有差异	GaussDB不支持yminterval类型,暂时用 interval兼容yminterval类型。
22	TRUNC	支持,有差异	GaussDB返回的类型与第一个入参的类型保持一致,Oracle始终返回date类型,另外支持指定的format也有区别,具体支持的列表详见《开发指南》的"SQL参考>函数和操作符>时间和日期处理函数和操作符"章节。
23	TZ_OFFSET	支持,有差异	接收一个时区名称为入参的时候,时区名 称的类型比Oracle数据库要少。

表 1-85 通用比较函数

序号	Oracle数据 库	GaussDB数据库	差异
1	GREATEST	支持,有差异	GaussDB不支持NLS_SORT参数指定的 比较方式,只支持二进制比较。GaussDB不支持多语种的表达式。
2	LEAST	支持,有差异	GaussDB不支持NLS_SORT参数指定的 比较方式,只支持二进制比较。GaussDB不支持多语种的表达式。

表 1-86 转换函数

序号	Oracle数据库	GaussDB数据 库	差异
1	ASCIISTR	支持	-
2	CAST	支持,有差异	• GaussDB不支持multiset子句。
			● GaussDB不支持nlsparam参数。
3	HEXTORAW	支持	-
4	RAWTOHEX	支持	-
5	TO_BINARY_ DOUBLE	支持,有差异	GaussDB不支持nlsparam参数。
6	TO_BINARY_F LOAT	支持,有差异	GaussDB不支持nlsparam参数。
7	TO_BLOB	支持,有差异	• GaussDB不支持long raw类型。
			● GaussDB不支持bfile、mime_type类型。
8	TO_CLOB	支持	-
9	TO_DATE	支持,有差异	• 不支持多语种参数。
			● 返回类型不一致。
			● 缺少控制参数NLS_DATE_FORMAT。
			● 部分format格式不支持。
			● fmt = 'j'。 1582年10月15日之前Oracle 数据库与GaussDB输出不一致。
			 无分割符时,不保证与Oracle数据库完全一致。如to_date('220725', 'yymmdd'), yy/rr按照固定长度4解析,会解析为2207年25月,25非法月份则会报错。

序号	Oracle数据库	GaussDB数据 库	差异
10	TO_MULTI_B YTE	支持	-
11	TO_NCHAR	支持,有差异	GaussDB: 将入参的类型转换为text。Oracle: 将入参的类型转换为国家字符集(national character set)。

序号	Oracle数据库	GaussDB数据 库	差异
12	TO_NUMBER		差异 GaussDB不支持NLS_PARAM参数。 GaussDB与Oracle的fmt选项差异点描述: 1、\$ GaussDB不支持该fmt。 2、, (comma) GaussDB: 逗号可以出现在fmt的任意位置。 Oracle: • 在format中,逗号只能出现在整数部分,且不能出现在数字的开头位置。 • 支持format中的逗号与原数据的逗号个数和位置不一致,但最后一个逗号的位置需一致。 • 原数据和Format中的连续逗号,等同于没有逗号。 • 当原数据中没有逗号时,format的最后一个逗号后面的数字位数需与原数据相等。 3、B GaussDB未实现该功能。 4、C GaussDB不支持NLS参数。 5、G GaussDB不支持NLS参数。 7、U GaussDB不支持NLS参数。 8、D GaussDB不支持NLS参数。 9、PR GaussDB: 等同于S,返回负数。 Oracle: • 返回<尖括号>中的负值。
			返回带前导和尾随空格的正值。限制: PR格式元素只能出现在数字格式模型的最后一个位置。10、RN rn

序号	Oracle数据库	GaussDB数据 库	差异
			GaussDB未实现该功能。
			TM TM9 TMe
			GaussDB未实现该功能。
			11、V
			GaussDB未实现该功能。
			12、FM
			GaussDB当有fm时,允许format中的逗号 比原数据中多,不需要严格保持一致。
			Oracle返回值保留前后的空格。
			13、EEEE
			GaussDB未实现该功能。
13	TO_SINGLE_B YTE	支持	-
14	TREAT	支持,有差异	GaussDB不支持使用"."操作符取值,不 支持转化为object类型。
15	UNISTR	支持,有差异	GaussDB只支持UTF-8编码,Oracle数据 库支持UTF-8和UTF-16编码。

表 1-87 大对象函数

序号	Oracle数据 库	GaussDB数据库	差异
1	EMPTY_BLO B	支持	-
2	EMPTY_CLO B	支持,有差异	GaussDB的CLOB类型不支持Oracle数 据库中的定位器概念。

表 1-88 层次函数

序号	Oracle数据库	GaussDB数据库
1	SYS_CONNECT_BY_PATH	不支持

表 1-89 XML 类型函数

序号	Oracle数据 库	GaussDB数 据库	差异
1	EXISTSNODE	支持,有差	在入参有命名空间时,xpath和命名空间都需要定义别名。
2	EXTRACTVAL UE	支持,有差 异	目前仅支持xpath1.0版本。
3	SYS_XMLAG G	支持,有差 异	xmlagg的别名,可使用xmlagg代替。
4	XMLAGG	支持	-
5	XMLCOMME NT	支持	-
6	XMLCONCAT	支持	-
7	XMLELEMEN T	支持,有差 异	xmlelement和xmlattributes的name字段赋 值为NULL时,行为与Oracle不一致。
			 xmlelement的name字段赋值为NULL时, 结果显示name信息为空,且不显示属性信息。
			● xmlattributes的name字段赋值为NULL 时,不显示属性信息。
8	XMLEXISTS	支持,有差 异	GaussDB入参为xml类型。
9	XMLFOREST	支持,有差 异	GaussDB返回值为xml类型。 GaussDB不支持EVALNAME语法。
10	XMLPARSE	 支持,有差	GaussDB返回值为xml类型。
	71112171102	异	GaussDB不支持WELLFORMED语法。
11	XMLROOT	支持,有差 异	GaussDB返回值为xml类型。
12	JSON_OBJEC T	支持	-
13	XMLTABLE	支持,有差 异	GaussDB从xml中选取数据使用的为XPath 1.0表达式,不支持声明默认命名空间,不支持多组输入及取别名,不支持省略传入数据的passing_clause子句,不支持RETURNING SEQUENCE BY REF子句和(SEQUENCE) BY REF子句。
14	GETSTRINGV AL	支持	-
15	GETCLOBVAL	支持	-

序号	Oracle数据 库	GaussDB数 据库	差异
16	XMLSEQUEN CE	支持	-

表 1-90 编码解码函数

序号	Oracle数据 库	GaussDB数 据库	差异
1	DECODE	支持	-
2	DUMP	支持,有差异	因存储格式不同,GaussDB数值和时间类型返回结果和Oracle数据库不一致。如,GaussDB中select dump(123); 返回Typ=23 Len=4: 123,0,0,0。Oracle中select dump(123) from dual; 返回Typ=2 Len=3: 194,2,24。
3	ORA_HASH	支持,有差 异	GaussDB中有以下行为: • 时间类型的入参转换成字符串类型再进行hash。 • 不支持maxbucket参数。
4	VSIZE	支持,有差 异	因存储格式不同,GaussDB数值和时间类型返回结果和Oracle数据库不一致。如GaussDB中select vsize(999); 返回4。Oracle中select vsize(999) from dual; 返回3。

表 1-91 空值相关的函数

序号	Oracle数据库	GaussDB数据库
1	COALESCE	支持
2	LNNVL	支持
3	NULLIF	支持
4	NVL	支持
5	NVL2	支持

表 1-92 环境和标识符函数

序号	Oracle数据 库	GaussDB数据 库	差异
1	SYS_CONTE	支持,有差异	GaussDB对不支持的参数返回NULL。
	XT		以下为不支持的参数列表:
			• 'action'
			'is_application_root'
			'is_application_pdb'
			'audited_cursorid'
			'authenticated_identity'
			'authentication_data'
			'authentication_method'
			• 'cdb_domain'
			• 'cdb_name'
			'client_identifier'
			• 'con_id'
			• 'con_name'
			• 'current_sql_length'
			• 'db_domain'
			• 'db_supplemental_log_level'
			• 'dblink_info'
			• 'drain_status'
			• 'entryid'
			'enterprise_identity'
			• 'fg_job_id'
			• 'global_uid'
			'identification_type'
			• 'instance'
			• 'is_dg_rolling_upgrade'
			• 'ldap_server_type'
			• 'module'
			'network_protocol'
			• 'nls_calendar'
			• 'nls_sort'
			'nls_territory'
			'oracle_home'
			• 'os_user'
			• 'platform_slash'

序号	Oracle数据 库	GaussDB数据 库	差异
			• 'policy_invoker'
			• 'proxy_enterprise_identity'
			• 'proxy_user'
			• 'proxy_userid'
			• 'scheduler_job'
			'session_edition_id'
			'session_edition_name'
			• 'sessionid'
			• 'statementid'
			• 'terminal'
			'unified_audit_sessionid'
			'session_default_collation'
			• 'client_info'
			• 'bg_job_id'
			'client_program_name'
			• 'current_bind'
			'global_context_memory'
			• 'host'
			• 'current_sqln'
2	SYS_GUID	支持	-
3	USER	支持,有差异	返回值类型不一致。

4 USERENV 支持,有差异 GaussDB对不支持的参数返回NULL。以下为不支持的参数列表:	序号	Oracle数据 库	GaussDB数据 库	差异
• 'platform slash'		库	库	GaussDB对不支持的参数返回NULL。 以下为不支持的参数列表: 'action' 'is_application_root' 'is_application_pdb' 'audited_cursorid' 'authenticated_identity' 'authentication_data' 'authentication_method' 'cdb_domain' 'cdb_name' 'client_identifier' 'con_id' 'con_name' 'current_sql_length' 'db_domain' 'db_supplemental_log_level' 'dblink_info' 'drain_status' 'entryid' 'enterprise_identity' 'fg_job_id' 'global_uid' 'identification_type' 'is_dg_rolling_upgrade' 'ldap_server_type' 'module' 'network_protocol' 'nls_calendar' 'nls_sort' 'nls_territory' 'os_user'
 'platform_slash' 'policy_invoker' 'proxy_enterprise_identity' 				'oracle_home''os_user''platform_slash''policy_invoker'

序号	Oracle数据 库	GaussDB数据 库	差异
			• 'proxy_user'
			• 'proxy_userid'
			• 'scheduler_job'
			'session_edition_id'
			'session_edition_name'
			• 'sessionid'
			• 'statementid'
			• 'terminal'
			'unified_audit_sessionid'
			'session_default_collation'
			• 'client_info'
			• 'bg_job_id'
			'client_program_name'
			• 'current_bind'
			'global_context_memory'
			• 'host'
			• 'current_sqln'

1.10.2 其它函数

序号	Oracle数据库	GaussDB数据库
1	聚合函数	支持
2	分析函数	支持
3	对象引用函数	不支持
4	模型函数	不支持
5	OLAP函数	不支持
6	数据盒功能函数	不支持
7	关于用户定义的函数	支持

表 1-93 聚合函数

序号	Oracle数据库	GaussDB数 据库	差异
1	AVG	支持	-
2	CORR	支持	-
3	COUNT	支持	-
4	COVAR_POP	支持	-
5	COVAR_SAMP	支持	-
6	CUME_DIST	支持	-
7	DENSE_RANK	支持	-
8	FIRST	支持	GaussDB使用KEEP的语法,兼容Oracle 此功能。
9	GROUPING	支持	-
10	LAST	支持	GaussDB使用KEEP的语法,兼容Oracle 此功能。
11	LISTAGG	支持	-
12	MAX	支持	-
13	MEDIAN	支持	-
14	MIN	支持	-
15	PERCENT_RAN K	支持	-
16	PERCENTILE_C ONT	支持	-
17	RANK	支持	-
18	REGR_ (Linear Regression)	支持	-
19	STDDEV	支持	-
20	STDDEV_POP	支持	-
21	STDDEV_SAM P	支持	-
22	SUM	支持	-
23	VAR_POP	支持	-
24	VAR_SAMP	支持	-
25	VARIANCE	支持	-

序号	Oracle数据库	GaussDB数 据库	差异
26	WM_CONCAT	支持	-

表 1-94 分析函数

序号	Oracle数据库	GaussDB数据 库	差异
1	FIRST_VALUE	支持	-
2	LAG	支持	-
3	LAST_VALUE	支持	-
4	LEAD	支持	-
5	NTH_VALUE	支持,有差异	 Oracle: 支持FROM FIRST LAST语法格式。 GaussDB: 不支持FROM FIRST LAST语法格式
6	NTILE	支持	-
7	ROW_NUMBE R	支持	-
8	RATIO_TO_RE PORT	支持	-

1.11 系统视图

GaussDB数据库兼容了部分Oracle数据库的系统视图,兼容的详细列表如下。 更多系统视图的字段说明信息请参考《开发指南》中"系统视图"章节。

表 1-95 系统视图支持列表

序号	Oracle数据库	GaussDB数据库
1	ALL_ALL_TABLES	DB_ALL_TABLES
2	ALL_COL_PRIVS	DB_COL_PRIVS
3	ALL_COLL_TYPES	DB_COLL_TYPES
4	ALL_IND_COLUMNS	DB_IND_COLUMNS
5	ALL_COL_COMMENTS	DB_COL_COMMENTS
6	ALL_CONS_COLUMNS	DB_CONS_COLUMNS

序号	Oracle数据库	GaussDB数据库
7	ALL_CONSTRAINTS	DB_CONSTRAINTS
8	ALL_DEPENDENCIES	DB_DEPENDENCIES
9	ALL_DIRECTORIES	DB_DIRECTORIES
10	ALL_IND_EXPRESSIONS	DB_IND_EXPRESSIONS
11	ALL_IND_PARTITIONS	DB_IND_PARTITIONS
12	ALL_INDEXES	DB_INDEXES
13	ALL_IND_SUBPARTITIONS	DB_IND_SUBPARTITIONS
14	ALL_OBJECTS	DB_OBJECTS
15	ALL_PART_COL_STATISTIC S	DB_PART_COL_STATISTICS
16	ALL_PART_KEY_COLUMNS	DB_PART_KEY_COLUMNS
17	ALL_PART_TABLES	DB_PART_TABLES
18	ALL_SCHEDULER_JOB_AR GS	DB_SCHEDULER_JOB_ARGS
19	ALL_SCHEDULER_PROGRA M_ARGS	DB_SCHEDULER_PROGRAM_ARGS
20	ALL_SEQUENCES	DB_SEQUENCES
21	ALL_SUBPART_KEY_COLU MNS	DB_SUBPART_KEY_COLUMNS
22	ALL_SYNONYMS	DB_SYNONYMS
23	ALL_TAB_COL_STATISTICS	DB_TAB_COL_STATISTICS
24	ALL_TAB_COMMENTS	DB_TAB_COMMENTS
25	ALL_TAB_HISTOGRAMS	DB_TAB_HISTOGRAMS
26	ALL_TAB_STATS_HISTORY	DB_TAB_STATS_HISTORY
27	ALL_TYPES	DB_TYPES
28	ALL_PROCEDURES	DB_PROCEDURES
29	ALL_SOURCE	DB_SOURCE
30	ALL_TAB_COLUMNS	DB_TAB_COLUMNS
31	ALL_TAB_PARTITIONS	DB_TAB_PARTITIONS
32	ALL_TAB_SUBPARTITIONS	DB_TAB_SUBPARTITIONS
33	ALL_TABLES	DB_TABLES
34	ALL_TRIGGERS	DB_TRIGGERS

序号	Oracle数据库	GaussDB数据库
35	ALL_USERS	DB_USERS
36	ALL_VIEWS	DB_VIEWS
37	DBA_AUDIT_OBJECT	ADM_AUDIT_OBJECT
38	DBA_AUDIT_SESSION	ADM_AUDIT_SESSION
39	DBA_AUDIT_STATEMENT	ADM_AUDIT_STATEMENT
40	DBA_AUDIT_TRAIL	ADM_AUDIT_TRAIL
41	DBA_COL_COMMENTS	ADM_COL_COMMENTS
42	DBA_COL_PRIVS	ADM_COL_PRIVS
43	DBA_COLL_TYPES	ADM_COLL_TYPES
44	DBA_ARGUMENTS	ADM_ARGUMENTS
45	DBA_CONSTRAINTS	ADM_CONSTRAINTS
46	DBA_DATA_FILES	ADM_DATA_FILES
47	DBA_CONS_COLUMNS	ADM_CONS_COLUMNS
48	DBA_DEPENDENCIES	ADM_DEPENDENCIES
49	DBA_DIRECTORIES	ADM_DIRECTORIES
50	DBA_PART_COL_STATISTIC S	ADM_PART_COL_STATISTICS
51	DBA_PART_TABLES	ADM_PART_TABLES
52	DBA_ROLE_PRIVS	ADM_ROLE_PRIVS
53	DBA_ROLES	ADM_ROLES
54	DBA_SCHEDULER_JOB_AR GS	ADM_SCHEDULER_JOB_ARGS
55	DBA_SCHEDULER_PROGR AMS	ADM_SCHEDULER_PROGRAMS
56	DBA_SCHEDULER_PROGR AM_ARGS	ADM_SCHEDULER_PROGRAM_ARGS
57	DBA_HIST_SNAPSHOT	ADM_HIST_SNAPSHOT
58	DBA_HIST_SQL_PLAN	ADM_HIST_SQL_PLAN
59	DBA_HIST_SQLSTAT	ADM_HIST_SQLSTAT
60	DBA_HIST_SQLTEXT	ADM_HIST_SQLTEXT
61	DBA_ILMDATAMOVEMEN TPOLICIES	GS_ADM_ILMDATAMOVEMENTPOLICIES

序号	Oracle数据库	GaussDB数据库
62	DBA_ILMEVALUATIONDET AILS	GS_ADM_ILMEVALUATIONDETAILS
63	DBA_ILMOBJECTS	GS_ADM_ILMOBJECTS
64	DBA_ILMPARAMETERS	GS_ADM_ILMPARAMETERS
65	DBA_ILMPOLICIES	GS_ADM_ILMPOLICIES
66	DBA_ILMRESULTS	GS_ADM_ILMRESULTS
67	DBA_ILMTASKS	GS_ADM_ILMTASKS
68	DBA_IND_COLUMNS	ADM_IND_COLUMNS
69	DBA_IND_EXPRESSIONS	ADM_IND_EXPRESSIONS
70	DBA_IND_PARTITIONS	ADM_IND_PARTITIONS
71	DBA_INDEXES	ADM_INDEXES
72	DBA_OBJECTS	ADM_OBJECTS
73	DBA_PART_INDEXES	ADM_PART_INDEXES
74	DBA_PROCEDURES	ADM_PROCEDURES
75	DBA_SCHEDULER_JOBS	ADM_SCHEDULER_JOBS
76	DBA_SCHEDULER_RUNNI NG_JOBS	ADM_SCHEDULER_RUNNING_JOBS
77	DBA_SEGMENTS	ADM_SEGMENTS
78	DBA_SEQUENCES	ADM_SEQUENCES
79	DBA_SOURCE	ADM_SOURCE
80	DBA_IND_SUBPARTITIONS	ADM_IND_SUBPARTITIONS
81	DBA_SUBPART_KEY_COLU MNS	ADM_SUBPART_KEY_COLUMNS
82	DBA_SYS_PRIVS	ADM_SYS_PRIVS
83	DBA_TAB_COL_STATISTICS	ADM_TAB_COL_STATISTICS
84	DBA_TAB_HISTOGRAMS	ADM_TAB_HISTOGRAMS
85	DBA_TAB_STATISTICS	ADM_TAB_STATISTICS
86	DBA_TAB_STATS_HISTORY	ADM_TAB_STATS_HISTORY
87	DBA_TABLESPACES	ADM_TABLESPACES
88	DBA_TYPES	ADM_TYPES
89	DBA_USERS	ADM_USERS

序号	Oracle数据库	GaussDB数据库
90	DBA_SYNONYMS	ADM_SYNONYMS
91	DBA_TAB_COLS	ADM_TAB_COLS
92	DBA_TAB_COLUMNS	ADM_TAB_COLUMNS
93	DBA_TAB_COMMENTS	ADM_TAB_COMMENTS
94	DBA_TABLES	ADM_TABLES
95	DBA_TAB_PARTITIONS	ADM_TAB_PARTITIONS
96	DBA_TAB_SUBPARTITIONS	ADM_TAB_SUBPARTITIONS
97	DBA_TRIGGERS	ADM_TRIGGERS
98	DBA_TYPE_ATTRS	ADM_TYPE_ATTRS
99	DBA_VIEWS	ADM_VIEWS
100	ROLE_ROLE_PRIVS	ROLE_ROLE_PRIVS
101	ROLE_SYS_PRIVS	ROLE_SYS_PRIVS
102	ROLE_TAB_PRIVS	ROLE_TAB_PRIVS
103	USER_COL_COMMENTS	MY_COL_COMMENTS
104	USER_COL_PRIVS	MY_COL_PRIVS
105	USER_COLL_TYPES	MY_COLL_TYPES
106	USER_CONSTRAINTS	MY_CONSTRAINTS
107	USER_DEPENDENCIES	MY_DEPENDENCIES
108	DICT	DICT
109	DICTIONARY	DICTIONARY
110	DUAL	DUAL
111	NLS_DATABASE_PARAMET ERS	NLS_DATABASE_PARAMETERS
112	NLS_INSTANCE_PARAMET ERS	NLS_INSTANCE_PARAMETERS
113	PLAN_TABLE	PLAN_TABLE
114	USER_ILMDATAMOVEME NTPOLICIES	GS_MY_ILMDATAMOVEMENTPOLICIES
115	USER_ILMEVALUATIONDE TAILS	GS_MY_ILMEVALUATIONDETAILS
116	USER_ILMOBJECTS	GS_MY_ILMOBJECTS
117	USER_ILMPOLICIES	GS_MY_ILMPOLICIES

序号	Oracle数据库	GaussDB数据库
118	USER_ILMRESULTS	GS_MY_ILMRESULTS
119	USER_ILMTASKS	GS_MY_ILMTASKS
120	USER_IND_COLUMNS	MY_IND_COLUMNS
121	USER_IND_EXPRESSIONS	MY_IND_EXPRESSIONS
122	USER_IND_PARTITIONS	MY_IND_PARTITIONS
123	USER_IND_SUBPARTITION S	MY_IND_SUBPARTITIONS
124	USER_INDEXES	MY_INDEXES
125	USER_JOBS	MY_JOBS
126	USER_OBJECTS	MY_OBJECTS
127	USER_PART_COL_STATISTI CS	MY_PART_COL_STATISTICS
128	USER_PART_INDEXES	MY_PART_INDEXES
129	USER_PART_TABLES	MY_PART_TABLES
130	USER_PROCEDURES	MY_PROCEDURES
131	USER_SCHEDULER_JOB_A RGS	MY_SCHEDULER_JOB_ARGS
132	USER_SCHEDULER_PROG RAM_ARGS	MY_SCHEDULER_PROGRAM_ARGS
133	USER_SEQUENCES	MY_SEQUENCES
134	USER_SOURCE	MY_SOURCE
135	USER_SUBPART_KEY_COL UMNS	MY_SUBPART_KEY_COLUMNS
136	USER_SYNONYMS	MY_SYNONYMS
137	USER_SYS_PRIVS	MY_SYS_PRIVS
138	USER_TAB_COL_STATISTIC S	MY_TAB_COL_STATISTICS
139	USER_TAB_COLUMNS	MY_TAB_COLUMNS
140	USER_TAB_COMMENTS	MY_TAB_COMMENTS
141	USER_TAB_HISTOGRAMS	MY_TAB_HISTOGRAMS
142	USER_TAB_PARTITIONS	MY_TAB_PARTITIONS
143	USER_TAB_STATISTICS	MY_TAB_STATISTICS

序号	Oracle数据库	GaussDB数据库
144	USER_TAB_STATS_HISTOR Y	MY_TAB_STATS_HISTORY
145	USER_TABLES	MY_TABLES
146	USER_TABLESPACES	MY_TABLESPACES
147	USER_TRIGGERS	MY_TRIGGERS
148	USER_TYPE_ATTRS	MY_TYPE_ATTRS
149	USER_TYPES	MY_TYPES
150	USER_VIEWS	MY_VIEWS
151	V\$NLS_PARAMETERS	V\$NLS_PARAMETERS
152	V\$SESSION_WAIT	V\$SESSION_WAIT
153	V\$SYSSTAT	V\$SYSSTAT
154	V\$SYSTEM_EVENT	V\$SYSTEM_EVENT
155	V\$VERSION	V\$VERSION
156	V\$INSTANCE	V_INSTANCE
157	GV\$INSTANCE	GV_INSTANCE
158	V\$MYSTAT	V_MYSTAT
159	V\$SESSION	V_SESSION
160	GV\$SESSION	GV_SESSION
161	V\$SESSION_LONGOPS	DV_SESSION_LONGOPS
162	V\$SESSION	DV_SESSIONS
163	ALL_ARGUMENTS	DB_ARGUMENTS
164	USER_CONS_COLUMNS	MY_CONS_COLUMNS
165	USER_PART_KEY_COLUM NS	MY_PART_KEY_COLUMNS
166	USER_ROLE_PRIVS	MY_ROLE_PRIVS
167	DBA_TAB_PRIVS	ADM_TAB_PRIVS
168	USER_SCHEDULER_JOBS	MY_SCHEDULER_JOBS
169	V\$LOCK	V\$LOCK
170	V\$DBLINK	V\$DBLINK
171	V \$GLOBAL_TRANSACTION	V\$GLOBAL_TRANSACTION

序号	Oracle数据库	GaussDB数据库
172	V\$OPEN_CURSOR	V\$OPEN_CURSOR
173	V \$GLOBAL_OPEN_CURSOR	V\$GLOBAL_OPEN_CURSOR
174	ALL_TAB_PRIVS	DB_TAB_PRIVS
175	ALL_TAB_MODIFICATIONS	DB_TAB_MODIFICATIONS
176	USER_TAB_MODIFICATIO NS	MY_TAB_MODIFICATIONS
177	USER_AUDIT_TRAIL	MY_AUDIT_TRAIL

1.12 高级包

GaussDB数据库兼容了部分Oracle数据库的高级包,兼容的详细列表如下。 更多高级包信息请参考开发指南中高级包章节。

表 1-96 高级包支持列表

序号	Oracle数据库	GaussDB数据 库	差异
1	DBMS_LOB	DBE_LOB	GaussDB具体用法请参见《 开发指南 》 在"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_LOB"章节。
2	DBMS_RAND OM	DBE_RANDO M	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_RANDOM"章节。
3	DBMS_OUTP UT	DBE_OUTPUT	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_OUTPUT"章节。
4	UTL_RAW	DBE_RAW	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_RAW"章节。
5	DBMS_SCHED ULER	DBE_SCHEDU LER	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_SCHEDULER"章节。
6	DBMS_UTILIT Y	DBE_UTILITY	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_UTILITY"章节。
7	DBMS_SQL	DBE_SQL	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_SQL"章节。

序号	Oracle数据库	GaussDB数据 库	差异
8	UTL_FILE	DBE_FILE	GaussDB具体用法请参见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_FILE"章节。
9	DBMS_SESSIO N	DBE_SESSION	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_SESSION"章节。
10	UTL_MATCH	DBE_MATCH	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_MATCH"章节。
11	DBMS_APPLIC ATION_INFO	DBE_APPLICA TION_INFO	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_APPLICATION_INFO" 章节。
12	DBMS_XMLD OM	DBE_XMLDO M	GaussDB中具体请参见《开发指南 》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_XMLDOM"章节。
13	DBMS_XMLPA RSER	DBE_XMLPAR SER	GaussDB中具体请参见《开发指南 》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_XMLPARSER"章节。
14	DBMS_ILM	DBE_ILM	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_ILM "章节。
15	DBMS_ILM_A DMIN	DBE_ILM_AD MIN	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_ILM_ADMIN"章节。
16	DBMS_COMP RESSION	DBE_COMPRE SSION	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_COMPRESSION"章 节。
17	DBMS_HEAT_ MAP	DBE_HEAT_M AP	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_HEAT_MAP"章节。
18	DBMS_DESCR IBE	DBE_DESCRIB E	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐)" > "DBE_DESCRIBE"章节。
19	DBMS_XMLGE N	DBE_XMLGEN	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_XMLGEN"章节。
20	DBMS_STATS	DBE_STATS	GaussDB中具体请参见《开发指南》中 "存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_STATS"章节。

表 1-97 DBMS_LOB 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	APPEND Procedures	APPEND Procedures	-
2	CLOB2FILE Procedure	不支持	-
3	CLOSE Procedure	BFILECLOSE Procedure	GaussDB:参数类型为BFILE,不存在函数重载。 Oracle:该过程存在3个重载,3 个重载的参数lob_loc、lob_loc 和file_loc的类型分别为BLOB、 CLOB CHARACTER SET ANY_CS和BFILE。
4	COMPARE Functions	COMPARE Functions	GaussDB:存在3个重载函数,对于第三个参数(len)均为BIGINT。Oracle:存在3个重载函数,对于第三个参数(amount)均为INTEGER。
5	CONVERTTOBLO B Procedure	LOB_CONVERTTOB LOB Procedure	GaussDB: 该过程共有5个参数,且第3、4、5个参数类型为BIGINT。 Oracle: 该过程共有8个参数,在GaussDB所有参数的基础上增加了blob_csid、lang_context和warning3个参数,类型分别为NUMBER、INTEGER和INTEGER,且第3、4、5个参数类型为INTEGER。
6	CONVERTTOCLO B Procedure	LOB_CONVERTTOC LOB Procedure	GaussDB: 该过程共有5个参数。第3、4、5个参数类型为BIGINT。 Oracle: 该过程共有8个参数。第3、4、5个参数类型为INTEGER。Oracle的该过程在GaussDB所有参数的基础上增加了3个参数:blob_csid、lang_context和warning,参数类型分别为NUMBER、INTEGER和INTEGER。
7	COPY Procedures	LOB_COPY Functions	-

序号	Oracle数据库	GaussDB数据库	差异
8	COPY_DBFS_LIN K Procedures	不支持	-
9	COPY_FROM_DB FS_LINK	不支持	-
10	CREATETEMPOr acleRY Procedures	CREATE_TEMPOracl eRY Procedures	GaussDB:该过程存在2个重载。第一个重载过程的第一个参数(lob_loc)为BLOB,第二个重载过程的第一个参数(lob_loc)为CLOB;两个重载过程的第三个参数(dur)为INTEGER,默认值为10。 Oracle:该过程存在2个重载。第一个重载过程的第一个参数(lob_loc)为BLOB,第二个重载过程的第一个参数(lob_loc)为CLOB;两个重载过程的第三个参数(dur)的参数类型为PLS_INTEGER,第一个重载过程的dur默认值为DBMS_LOB.SESSION,第二个重载过程的dur默认值为10。
11	DBFS_LINK_GEN ERATE_PATH Functions	不支持	-
12	ERASE Procedures	LOB_ERASE Procedures	-
13	FILECLOSE Procedure	不支持	-
14	FILECLOSEALL Procedure	不支持	-
15	FILEEXISTS Function	不支持	-
16	FILEGETNAME Procedure	不支持	-
17	FILEISOPEN Function	不支持	-
18	FILEOPEN Procedure	不支持	-
19	FRAGMENT_DEL ETE Procedure	不支持	-
20	FRAGMENT_INS ERT Procedures	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
21	FRAGMENT_MO VE Procedure	不支持	-
22	FRAGMENT_REP LACE Procedures	不支持	-
23	FREETEMPOracle RY Procedures	不支持	-
24	GET_DBFS_LINK Functions	不支持	-
25	GET_DBFS_LINK_ STATE Procedures	不支持	-
26	GETCHUNKSIZE Functions	GETCHUNKSIZE Functions	-
27	GETCONTENTTY PE Functions	不支持	-
28	GETLENGTH Functions	不支持	-
29	GETOPTIONS Functions	不支持	-
30	GET_STOracleGE _LIMIT Function	不支持	-
31	INSTR Functions	MATCH Functions	GaussDB:存在3个重载函数。 3个重载函数的第三、四个参数 均为BIGINT。
			Oracle:存在3个重载函数。3个 重载函数的第三、四个参数均为 INTEGER。
32	ISOPEN Functions	不支持	-
33	ISREMOTE Function	不支持	-
34	ISSECUREFILE Function	不支持	-
35	ISTEMPOracleRY Functions	不支持	-
36	LOADBLOBFRO MFILE Procedure	LOADBLOBFROMFI LE Procedure	-
37	LOADCLOBFRO MFILE Procedure	LOADCLOBFROMFI LE Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
38	LOADFROMFILE Procedure	LOADFROMFILE Procedure	-
39	MOVE_TO_DBFS _LINK Procedures	不支持	-
40	OPEN Procedures	BFILEOPEN Procedure	GaussDB: 该过程不存在重载。 第一个参数(bfile)类型为 DBE_LOB.BFILE,第二个参数 (open_mode)类型为TEXT, 且只支持read模式。 Oracle: 该过程存在3个重载。 第一个重载过程的第一个参数 (lob_loc)类型为NOCOPY BLOB,第二个参数 (openmode)类型为 BINARY_INTEGER;第二个重载 过程的第一个参数(lob_loc) 类型为NOCOPY CLOB CHARACTER SET ANY_CS,第 二个参数(openmode)类型为 BINARY_INTEGER;第三个重载 过程的第一个参数(file_loc) 类型为NOCOPY BFILE,第二个 参数(openmode)类型为 BINARY_INTEGER,且只能为 file_readonly。
41	READ Procedures	READ Procedures	GaussDB:该过程存在2个重载。 Oracle:该过程存在3个重载。 其中前两个重载与GaussDB无差异,第三个过程重载包括4个参数:file_loc、amount、offset和buffer,其类型分别为BFILE、NOCOPY INTEGER、INTEGER和RAW。
42	SET_DBFS_LINK Procedures	不支持	-
43	SETCONTENTTY PE Procedure	不支持	-
44	SETOPTIONS Procedures	不支持	-
45	SUBSTR Functions	LOB_SUBSTR Functions	-

序号	Oracle数据库	GaussDB数据库	差异
46	TRIM Procedures	STRIP Functions	GaussDB:该过程存在2个重 载。两个重载过程的第二个参数 (newlen)均为BIGINT。
			Oracle:该过程存在2个重载。 两个重载过程的第二个参数 (newlen)均为INTEGER。
47	WRITE Procedures	WRITE Functions	-
48	WRITEAPPEND Procedures	WRITEAPPEND Functions	-

表 1-98 DBMS_RANDOM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	INITIALIZE Procedure	不支持	-
2	NORMAL Function	不支持	-
3	RANDOM Function	不支持	-
4	SEED Procedures	DBE_RANDOM.SET_ SEED Function	GaussDB: 该函数无重载,参数 类型为INTEGER。
			Oracle:该过程存在2个重载,2 个重载过程的参数类型分别为 VARCHAR2和 BINARY_INTEGER。
5	STRING Function	不支持	-
6	TERMINATE Procedure	不支持	-
7	VALUE Functions	DBE_RANDOM.GET _VALUE Function	GaussDB:该函数无重载。 Oracle:存在无参数的VALUE函 数重载,返回NUMBER类型。

表 1-99 DBMS_OUTPUT 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DISABLE Procedure	DISABLE Function	-

序号	Oracle数据库	GaussDB数据库	差异
2	ENABLE Procedure	ENABLE Function	-
3	GET_LINE Procedure	GET_LINE Function	GaussDB: 该函数无重载,首个参数(lines)数据类型为VARCHAR[]。 Oracle: 该过程存在2个重载,2个重载过程的首个参数(lines)分别为CHARARR和DBMSOUTPUT_LINESARRAY。
4	GET_LINES Procedure	GET_LINES Function	-
5	NEW_LINE Procedure	NEW_LINE Function	-
6	PUT Procedure	PUT Function	GaussDB: 当数据库服务端字符集server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时,该函数不会区分入参的数据类型,都会先把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换后再输出。 Oracle: 当数据库服务端字符集server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时,若入参类型是NVARCHAR2,则该过程会先把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换后再输出;若入参为其他字符转换后再输出;若入参为其他字符类型,则会将该字符编码视作非法字符,以占位符的形式输出。

序号	Oracle数据库	GaussDB数据库	差异
7	PUT_LINE Procedure	PUT_LINE Function	GaussDB: 当数据库服务端字符集 server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时,该函数不会区分入参的数据类型,都会先把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换后再输出。 Oracle: 当数据库服务端字符集 server_encoding不是UTF8编码格式且入参的字符编码是合法的UTF8编码时,若入参类型是NVARCHAR2,则该过程会先把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换后再输出;若入参为其他字符类型,则会将该字符编码视作非法字符,以占位符的形式输出。

表 1-100 UTL_RAW 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	BIT_AND Function	BIT_AND Function	-
2	BIT_COMPLEME NT Function	BIT_COMPLEMENT Function	-
3	BIT_OR Function	BIT_OR Function	GaussDB:两个参数类型被定 义为TEXT类型并且返回TEXT类型。
			Oracle:两个参数为RAW类型 并且返回RAW类型。
4	BIT_XOR Function	BIT_XOR Function	-
5	CAST_FROM_BIN ARY_DOUBLE Function	CAST_FROM_BINAR Y_DOUBLE_TO_RA W Function	-
6	CAST_FROM_BIN ARY_FLOAT Function	CAST_FROM_BINAR Y_FLOAT_TO_RAW Function	GaussDB:参数n为FLOAT4类型。 Oracle:参数n为FLOAT类型。

序号	Oracle数据库	GaussDB数据库	差异
7	CAST_FROM_BIN ARY_INTEGER Function	CAST_FROM_BINAR Y_INTEGER_TO_RA W Function	GaussDB:参数value为BIGINT 类型。 Oracle:参数value为INTEGER 类型。
8	CAST_FROM_NU MBER Function	CAST_FROM_NUMB ER_TO_RAW Function	GaussDB:参数n为NUMERIC 类型 Oracle:参数n为NUMBER类 型。
9	CAST_TO_BINARY _DOUBLE Function	CAST_FROM_RAW_ TO_BINARY_DOUBL E Function	-
10	CAST_TO_BINARY _FLOAT Function	CAST_FROM_RAW_ TO_BINARY_FLOAT Function	GaussDB: 函数返回类型为 FLOAT4。 Oracle: 函数返回类型为 FLOAT。
11	CAST_TO_BINARY _INTEGER Function	CAST_FROM_RAW_ TO_BINARY_INTEGE R Function	GaussDB: 参数endianess为 INTEGER类型,函数返回类型 为INTEGER。 Oracle: 参数endianess为 PLS_INTEGER类型,函数返回 类型为BINARY_INTEGER。
12	CAST_TO_NUMB ER Function	CAST_FROM_RAW_ TO_NUMBER Function	GaussDB: 函数返回类型为 NUMERIC。 Oracle: 函数返回类型为 NUMBER。
13	CAST_TO_NVARC HAR2 Function	CAST_FROM_RAW_ TO_NVARCHAR2 Function	-
14	CAST_TO_RAW Function	CAST_FROM_VARCH AR2_TO_RAW Function	-
15	CAST_TO_VARCH AR2 Function	CAST_TO_VARCHAR 2 Function	-
16	COMPARE Function	COMPARE Function	GaussDB: 函数返回类型为 INTEGER。 Oracle: 函数返回类型为 NUMBER。
17	CONCAT Function	CONCAT Function	-
18	CONVERT Function	CONVERT Function	-

序号	Oracle数据库	GaussDB数据库	差异
19	COPIES Function	COPIES Function	GaussDB:参数n为NUMERIC 类型。 Oracle:参数n为NUMBER类 型。
20	LENGTH Function	GET_LENGTH Function	-
21	OVERLAY Function	OVERLAY Function	-
22	REVERSE Function	REVERSE Function	-
23	SUBSTR Function	SUBSTR Function	GaussDB: 参数lob_loc为BLOB 类型; 参数off_set为INTEGER 类型,默认值为1; 参数 amount为INTEGER类型,默认值为32767。 Oracle: 参数r为RAW类型,参 数pos为BINARY_INTEGER类型 且无默认值,参数len为 BINARY_INTEGER类型,默认值为NULL。
24	TRANSLATE Function	TRANSLATE Function	-
25	TRANSLITERATE Function	TRANSLITERATE Function	-
26	XRANGE Function	XRANGE Function	GaussDB:参数start_byte和 end_byte无默认值。 Oracle:参数start_byte和 end_byte默认为NULL。

表 1-101 DBMS_SCHEDULER 兼容性说明

序号	Oracle数据库	GaussDB数据库
1	ADD_EVENT_QUEUE_SUBSCRIBER Procedure	不支持
2	ADD_GROUP_MEMBER Procedure	不支持
3	ADD_JOB_EMAIL_NOTIFICATION Procedure	不支持
4	ADD_TO_INCOMPATIBILITY Procedure	不支持

序号	Oracle数据库	GaussDB数据库
5	ALTER_CHAIN Procedure	不支持
6	ALTER_RUNNING_CHAIN Procedure	不支持
7	CLOSE_WINDOW Procedure	不支持
8	COPY_JOB Procedure	不支持
9	CREATE_CHAIN Procedure	不支持
10	CREATE_CREDENTIAL Procedure	CREATE_CREDENTIAL Procedure
11	CREATE_DATABASE_DESTINATION Procedure	不支持
12	CREATE_EVENT_SCHEDULE Procedure	不支持
13	CREATE_FILE_WATCHER Procedure	不支持
14	CREATE_GROUP Procedure	不支持
15	CREATE_INCOMPATIBILITY Procedure	不支持
16	CREATE_JOB Procedure	CREATE_JOB Procedure
17	CREATE_JOB_CLASS Procedure	CREATE_JOB_CLASS Procedure
18	CREATE_JOBS Procedure	不支持
19	CREATE_PROGRAM Procedure	CREATE_PROGRAM Procedure
20	CREATE_RESOURCE Procedure	不支持
21	CREATE_SCHEDULE Procedure	CREATE_SCHEDULE Procedure
22	CREATE_WINDOW Procedure	不支持
23	DEFINE_ANYDATA_ARGUMENT Procedure	不支持
24	DEFINE_CHAIN_EVENT_STEP Procedure	不支持
25	DEFINE_CHAIN_RULE Procedure	不支持
26	DEFINE_CHAIN_STEP Procedure	不支持
27	DEFINE_METADATA_ARGUMENT Procedure	不支持
28	DEFINE_PROGRAM_ARGUMENT Procedure	DEFINE_PROGRAM_ARGUMENT Procedure
29	DISABLE Procedure	DISABLE Procedure

序号	Oracle数据库	GaussDB数据库
30	DROP_AGENT_DESTINATION Procedure	不支持
31	DROP_CHAIN Procedure	不支持
32	DROP_CHAIN_RULE Procedure	不支持
33	DROP_CHAIN_STEP Procedure	不支持
34	DROP_CREDENTIAL Procedure	DROP_CREDENTIAL Procedure
35	DROP_DATABASE_DESTINATION Procedure	不支持
36	DROP_FILE_WATCHER Procedure	不支持
37	DROP_GROUP Procedure	不支持
38	DROP_INCOMPATIBILITY Procedure	不支持
39	DROP_JOB Procedure	DROP_JOB Procedure
40	DROP_JOB_CLASS Procedure	DROP_JOB_CLASS Procedure
41	DROP_PROGRAM Procedure	DROP_PROGRAM Procedure
42	DROP_PROGRAM_ARGUMENT Procedure	不支持
43	DROP_SCHEDULE Procedure	DROP_SCHEDULE Procedure
44	DROP_WINDOW Procedure	不支持
45	ENABLE Procedure	ENABLE Procedure
46	END_DETACHED_JOB_RUN Procedure	不支持
47	EVALUATE_CALENDAR_STRING Procedure	EVALUATE_CALENDAR_STRING Procedure
48	EVALUATE_RUNNING_CHAIN Procedure	不支持
49	GENERATE_JOB_NAME Function	GENERATE_JOB_NAME Function
50	GET_AGENT_INFO Function	不支持
51	GET_AGENT_VERSION Function	不支持
52	GET_ATTRIBUTE Procedure	不支持
53	GET_FILE Procedure	不支持
54	GET_SCHEDULER_ATTRIBUTE Procedure	不支持
55	OPEN_WINDOW Procedure	不支持

序号	Oracle数据库	GaussDB数据库
56	PURGE_LOG Procedure	不支持
57	PUT_FILE Procedure	不支持
58	REMOVE_EVENT_QUEUE_SUBSCRI BER Procedure	不支持
59	REMOVE_FROM_INCOMPATIBILITY Procedure	不支持
60	REMOVE_GROUP_MEMBER Procedure	不支持
61	REMOVE_JOB_EMAIL_NOTIFICATIO N Procedure	不支持
62	RESET_JOB_ARGUMENT_VALUE Procedure	不支持
63	RUN_CHAIN Procedure	不支持
64	RUN_JOB Procedure	RUN_JOB Procedure
65	SET_AGENT_REGISTRATION_PASS Procedure	不支持
66	SET_ATTRIBUTE Procedure	SET_ATTRIBUTE Procedure
67	SET_ATTRIBUTE_NULL Procedure	不支持
68	SET_JOB_ANYDATA_VALUE Procedure	不支持
69	SET_JOB_ARGUMENT_VALUE Procedure	SET_JOB_ARGUMENT_VALUE Procedure
70	SET_JOB_ATTRIBUTES Procedure	不支持
71	SET_RESOURCE_CONSTRAINT Procedure	不支持
72	SET_SCHEDULER_ATTRIBUTE Procedure	不支持
73	STOP_JOB Procedure	STOP_JOB Procedure

表 1-102 DBMS_UTILITY 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	ACTIVE_INSTANCES Procedure	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	ANALYZE_DATABASE Procedure	不支持	-
3	ANALYZE_PART_OBJECT Procedure	不支持	-
4	ANALYZE_SCHEMA Procedure	不支持	-
5	CANONICALIZE Procedure	CANONICALIZE Procedure	GaussDB:参数 canon_len默认为1024字 节。 Oracle:参数canon_len 无默认值。
6	COMMA_TO_TABLE Procedures	COMMA_TO_TABL E Procedures	GaussDB:参数tab为 VARCHAR2数组。 Oracle:该过程存在2个 重载。参数tab可以为两 种类型之一:一种为 uncl_array,另一种为 lname_array。
7	COMPILE_SCHEMA Procedure	不支持	-
8	CREATE_ALTER_TYPE_ER ROR_TABLE Procedure	不支持	-
9	CURRENT_INSTANCE Function	不支持	-
10	DATA_BLOCK_ADDRESS_ BLOCK Function	不支持	-
11	DATA_BLOCK_ADDRESS_ FILE Function	不支持	-
12	DB_VERSION Procedure	DB_VERSION Procedure	GaussDB: 只有参数 version,类型为 VARCHAR2。Oracle: 有 参数version和 compatibility,类型均为 VARCHAR2。
13	EXEC_DDL_STATEMENT Procedure	EXEC_DDL_STATE MENT Function	GaussDB:参数 parse_string为TEXT类型。 Oracle:参数 parse_string为 VARCHAR2类型。

序号	Oracle数据库	GaussDB数据库	差异
14	EXPAND_SQL_TEXT Procedure	EXPAND_SQL_TEX T Function	GaussDB: 参数 output_sql_text为 CLOB。 Oracle: 参数 output_sql_text为 NOCOPY CLOB,通过传引用方式传递OUT参数。
15	FORMAT_CALL_STACK Function	FORMAT_CALL_ST ACK Function	GaussDB: 函数返回类型 为TEXT。 Oracle: 函数返回类型为 VARCHAR2。
16	FORMAT_ERROR_BACKT RACE Function	FORMAT_ERROR_ BACKTRACE Function	GaussDB: 函数返回类型 为TEXT。 Oracle: 函数返回类型为 VARCHAR2。
17	FORMAT_ERROR_STACK Function	FORMAT_ERROR_ STACK Function	GaussDB: 函数返回类型 为TEXT。 Oracle: 函数返回类型为 VARCHAR2。
18	GET_CPU_TIME Function	GET_CPU_TIME Function	GaussDB: 函数返回类型 为BIGINT。 Oracle: 函数返回类型为 NUMBER。
19	GET_DEPENDENCY Procedure	不支持	-
20	GET_ENDIANNESS Function	GET_ENDIANNESS Function	GaussDB: 函数返回类型 为INTEGER。 Oracle: 函数返回类型为 NUMBER。
21	GET_HASH_VALUE Function	GET_HASH_VALUE Function	GaussDB:参数base、hash_size和返回类型均为INTEGER。 Oracle:参数base、hash_size和返回类型均为NUMBER。
22	GET_PARAMETER_VALUE Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
23	GET_SQL_HASH Function	GET_SQL_HASH Function	GaussDB: 参数 last4bytes 为BIGINT类型,代表MD5哈希值的最后四字节,以无符号整数形式展现,函数返回类型为BIGINT。 Oracle:对应参数pre10ihash为NUMBER类型,用于存储MD5计算得到的16字节中的4字节哈希值。
24	GET_TIME Function	GET_TIME Function	GaussDB: 函数返回类型 为BIGINT。 Oracle: 函数返回类型为 NUMBER。
25	GET_TZ_TRANSITIONS Procedure	不支持	-
26	INVALIDATE Procedure	不支持	-
27	IS_BIT_SET Function	IS_BIT_SET Function	GaussDB:参数n和返回 值类型为INTEGER。 Oracle:参数n和返回值 类型为NUMBER。
28	IS_CLUSTER_DATABASE Function	IS_CLUSTER_DATA BASE Function	-
29	MAKE_DATA_BLOCK_AD DRESS Function	不支持	-
30	NAME_RESOLVE Procedure	NAME_RESOLVE Procedure	GaussDB: 参数context 和part1_type为 INTEGER,参数 object_number为OID; GaussDB不支持NUMBER 到OID的隐式转换。 Oracle: 参数context、 part1_type和 object_number均为 NUMBER。
31	NAME_TOKENIZE Procedure	NAME_TOKENIZE Procedure	GaussDB:参数nextpos 为INTEGER类型。 Oracle:参数nextpos为 BINARY_INTEGER类型。

序号	Oracle数据库	GaussDB数据库	差异
32	OLD_CURRENT_SCHEMA Function	OLD_CURRENT_SC HEMA Function	GaussDB: 函数返回类型 为VARCHAR。 Oracle: 函数返回类型为 VARCHAR2。
33	OLD_CURRENT_USER Function	OLD_CURRENT_U SER Function	GaussDB: 函数返回类型 为TEXT。 Oracle: 函数返回类型为 VARCHAR2。
34	PORT_STRING Function	不支持	-
35	SQLID_TO_SQLHASH Function	不支持	-
36	TABLE_TO_COMMA Procedures	TABLE_TO_COMM A Procedures	GaussDB:参数tab为 VARCHAR2数组。 Oracle:该存储过程存在 2个重载。参数tab可以为 两种类型之一:一种为 uncl_array,另一种为 lname_array。
37	VALIDATE Procedure	不支持	-
38	WAIT_ON_PENDING_DM L Function	不支持	-

表 1-103 DBMS_SQL 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	BIND_ARRAY Procedures	SQL_BIND_ARRAY Function	-
2	BIND_VARIABLE Procedures	SQL_BIND_VARIABL E Function	-
3	BIND_VARIABLE_PKG Procedure	不支持	-
4	CLOSE_CURSOR Procedure	SQL_UNREGISTER_C ONTEXT Function	-
5	COLUMN_VALUE Procedure	GET_RESULT Procedure	-
6	COLUMN_VALUE_LO NG Procedure	不支持	-
7	DEFINE_ARRAY Procedure	SET_RESULTS_TYPE Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
8	DEFINE_COLUMN Procedures	SET_RESULT_TYPE Procedure	-
9	DEFINE_COLUMN_C HAR Procedure	不支持	-
10	DEFINE_COLUMN_L ONG Procedure	不支持	-
11	DEFINE_COLUMN_R AW Procedure	不支持	-
12	DEFINE_COLUMN_R OWID Procedure	不支持	-
13	DESCRIBE_COLUMNS Procedure	DESCRIBE_COLUMN S Procedure	-
14	DESCRIBE_COLUMNS 2 Procedure	不支持	-
15	DESCRIBE_COLUMNS 3 Procedure	不支持	-
16	EXECUTE Function	SQL_RUN Function	GaussDB:返回值为常量 1。 Oracle:返回值对于INSERT, UPDATE,DELETE语句是影响的行数,对于其他语句则 无意义。
17	EXECUTE_AND_FETC H Function	RUN_AND_NEXT Function	-
18	FETCH_ROWS Function	NEXT_ROW Function	-
19	GET_NEXT_RESULT Procedures	不支持	-
20	IS_OPEN Function	IS_ACTIVE Function	-
21	LAST_ERROR_POSITI ON Function	不支持	-
22	LAST_ROW_COUNT Function	LAST_ROW_COUNT Function	-
23	LAST_ROW_ID Function	不支持	-
24	LAST_SQL_FUNCTIO N_CODE Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
25	OPEN_CURSOR Functions	REGISTER_CONTEXT Function	-
26	PARSE Procedures	支持,有差异	GaussDB中为SQL_SET_SQL Function,不支持重载。
27	RETURN_RESULT Procedures	不支持	-
28	TO_CURSOR_NUMBE R Function	不支持	-
29	TO_REFCURSOR Function	不支持	-
30	VARIABLE_VALUE Procedures	GET_VARIABLE_RES ULT Procedures	-
31	VARIABLE_VALUE_PK G Procedure	不支持	-

表 1-104 DBMS_SQL 数据类型兼容性说明

序号	Oracle数据库	GaussDB数据库
1	DBMS_SQL DESC_REC	DBE_SQL.DESC_REC
2	DBMS_SQL DATE_TABLE	DBE_SQL.DATE_TABLE
3	DBMS_SQL NUMBER_TABLE	DBE_SQL.NUMBER_TABLE
4	DBMS_SQL VARCHAR2_TABLE	DBE_SQL.VARCHAR2_TABLE
5	DBMS_SQL BLOB_TABLE	DBE_SQL.BLOB_TABLE

表 1-105 UTL_FILE 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	FCLOSE Procedure	CLOSE Procedure	-
2	FCLOSE_ALL Procedure	CLOSE_ALL Procedure	-
3	FCOPY Procedure	COPY Procedure	-
4	FFLUSH Procedure	FLUSH Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
5	FGETATTR Procedure	GET_ATTR Procedure	-
6	FGETPOS Function	GET_POS Function	-
7	FOPEN Function	FOPEN Function	-
8	FOPEN_NCHAR Function	FOPEN_NCHAR Function	-
9	FREMOVE Procedure	REMOVE Procedure	-
10	FRENAME Procedure	RENAME Procedure	-
11	FSEEK Procedure	SEEK Procedure	-
12	GET_LINE Procedure	READ_LINE Procedure	-
13	GET_LINE_NCHA R Procedure	READ_LINE_NCHAR Procedure	-
14	GET_RAW Procedure	GET_RAW Procedure	-
15	IS_OPEN Function	IS_OPEN Function	-
16	NEW_LINE Procedure	支持,有差异, NEW_LINE Function	GaussDB将接口定义为 Function。
17	PUT Procedure	支持,有差异,WRITE Function	GaussDB将接口定义为 Function。
18	PUT_LINE Procedure	支持,有差异, WRITE_LINE Function	GaussDB将接口定义为 Function。
19	PUT_LINE_NCHA R Procedure	支持,有差异, WRITE_LINE_NCHAR Function	GaussDB将接口定义为 Function。
20	PUT_NCHAR Procedure	支持,有差异, WRITE_NCHAR Function	GaussDB将接口定义为 Function。
21	PUTF Procedure	支持,有差异, FORMAT_WRITE Function	GaussDB将接口定义为 Function。
22	PUTF_NCHAR Procedure	支持,有差异, FORMAT_WRITE_NC HAR Function	GaussDB将接口定义为 Function。

序号	Oracle数据库	GaussDB数据库	差异
23	PUT_RAW Procedure	支持,有差异, PUT_RAW Function	GaussDB将接口定义为 Function。

表 1-106 DBMS_SESSION 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	CLEAR_ALL_CONTEXT Procedure	不支持	-
2	CLEAR_CONTEXT Procedure	CLEAR_CONTEXT Function	-
3	CLEAR_IDENTIFIER Procedure	不支持	-
4	CLOSE_DATABASE_LIN K Procedure	不支持	-
5	CURRENT_IS_ROLE_EN ABLED Function	不支持	-
6	FREE_UNUSED_USER_ MEMORY Procedure	不支持	-
7	GET_PACKAGE_MEMOR Y_UTILIZATION Procedure	不支持	-
8	IS_ROLE_ENABLED Function	不支持	-
9	IS_SESSION_ALIVE Function	不支持	-
10	LIST_CONTEXT Procedures	不支持	-
11	MODIFY_PACKAGE_STA TE Procedure	MODIFY_PACKAGE_ STATE Procedure	GaussDB: 仅支持入参 flags = 1的场景使用。 Oracle: 支持flags=1或 flags = 2的场景使用。
12	RESET_PACKAGE Procedure	不支持	-
13	SESSION_IS_ROLE_ENA BLED Function	不支持	-
14	SESSION_TRACE_DISAB LE Procedure	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
15	SESSION_TRACE_ENAB LE Procedure	不支持	-
16	SET_CONTEXT Procedure	SET_CONTEXT Function	GaussDB: 仅包括参数 namespace, attribute和 value,类型均为text。 Oracle: 包括参数 namespace, attribute, value, username和 client_id,类型均为 VARCHAR2。
17	SET_EDITION_DEFERRE D Procedure	不支持	-
18	SET_IDENTIFIER Procedure	不支持	-
19	SET_NLS Procedure	不支持	-
20	SET_ROLE Procedure	不支持	-
21	SET_SQL_TRACE Procedure	不支持	-
22	SLEEP Procedure	不支持	-
23	SWITCH_CURRENT_CO NSUMER_GROUP Procedure	不支持	-
24	UNIQUE_SESSION_ID Function	不支持	-

表 1-107 UTL_MATCH 兼容性说明

序号	Oracle数据库	GaussDB数据 库	差异
1	EDIT_DISTANCE Function	不支持	-
2	EDIT_DISTANCE_SIMI LARITY Function	EDIT_DISTANC E_SIMILARITY Function	GaussDB:参数str1和str2均为 TEXT类型,函数返回类型为 INTEGER。 Oracle:参数s1和s2为 VARCHAR2类型,函数返回类型 为PLS_INTEGER。
3	JARO_WINKLER Function	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
4	JARO_WINKLER_SIMI LARITY Function	不支持	-

表 1-108 DBMS_APPLICATION_INFO 兼容性说明

序号	Oracle数据库	GaussDB数据 库	差异
1	READ_CLIENT_IN FO Function	READ_CLIENT_ INFO	GaussDB:参数client_info为TEXT类型。
		Procedure	Oracle:参数client_info为VARCHAR2 类型。
2	READ_MODULE Procedure	READ_MODUL E Procedure	GaussDB:参数module_name、 action_name为TEXT类型。
			Oracle:参数module_name、 action_name为VARCHAR2类型。
3	SET_ACTION Procedure	SET_ACTION Procedure	GaussDB:参数action_name为TEXT 类型。
			Oracle:参数action_name为 VARCHAR2类型。
4	SET_CLIENT_INF O Function	SET_CLIENT_IN FO Procedure	GaussDB:参数str为TEXT类型,且返 回类型为void。
			Oracle:参数client_info为VARCHAR2 类型,无返回值。二者均为写入客户 端信息,最大输入64字节,超过64字 节将被截断。
5	SET_MODULE Procedure	SET_MODULE Procedure	GaussDB:参数module_name、 action_name为TEXT类型。
			Oracle:参数module_name、 action_name为VARCHAR2类型。
6	SET_SESSION_LO NGOPS Procedure	不支持	-

表 1-109 DBMS_XMLDOM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLDOM. APPENDCHILD	DBE_XMLDOM.AP PENDCHILD	 GaussDB: DOCUMENT类型 节点下APPEND ATTR类型节点会报 "operation not support"错误。Oracle: 在此场景下不报错,但实际并没有挂载成功。 GaussDB: ATTR类型节点下APPEND ATTR类型节点会报 "operation not support"错误。Oracle: 在此场景下不报错,但实际并没有挂载成功。 GaussDB: 父节点在添加多个ATTR类型子节点时,不允许KEY值相同的子节点同时存在于同一个父节点下。Oracle: 允许KEY值相同的子节点同时存在于同一个父节点下。Oracle: 允许KEY值相同的子节点同时存在于同一个父节点下。
2	DBMS_XMLDOM. CREATEELEMENT	DBE_XMLDOM.CR EATEELEMENT	-
3	DBMS_XMLDOM. CREATETEXTNO DE	DBE_XMLDOM.CR EATETEXTNODE	-
4	DBMS_XMLDOM. FREEDOCUMENT	支持,有差异, DBE_XMLDOM.FR EEDOCUMENT	GaussDB:释放时不会立刻释放对象,累积一定数量后释放。document下全部节点失效。Oracle:立即释放对象。
5	DBMS_XMLDOM. FREEELEMENT	DBE_XMLDOM.FR EEELEMENT	-
6	DBMS_XMLDOM. FREENODE	DBE_XMLDOM.FR EENODE	-
7	DBMS_XMLDOM. FREENODELIST	支持,有差异, DBE_XMLDOM.FR EENODELIST	GaussDB:nodelist会被释放。 Oracle:释放nodelist后,在原始 的doc中还能被查询到。
8	DBMS_XMLDOM. GETATTRIBUTE	DBE_XMLDOM.GE TATTRIBUTE	-
9	DBMS_XMLDOM. GETATTRIBUTES	DBE_XMLDOM.GE TATTRIBUTES	-

序号	Oracle数据库	GaussDB数据库	差异
10	DBMS_XMLDOM. GETCHILDNODE S	DBE_XMLDOM.GE TCHILDNODES	GaussDB: 对document的node使 用时会包含dtd Oracle: 不包含dtd。
11	DBMS_XMLDOM. GETCHILDRENBY TAGNAME	DBE_XMLDOM.GE TCHILDRENBYTAG NAME	GaussDB: DBE_XMLDOM.GETCHILDRENBY TAGNAME接口的参数ns不支持传入参数"*",如需获取节点下全部属性,可使用 DBE_XMLDOM.GETCHILDNODE S接口。 Oracle: 支持传入参数"*"。
12	DBMS_XMLDOM. GETDOCUMENT ELEMENT	DBE.XMLDOM.GE TDOCUMENTELE MENT	-
13	DBMS_XMLDOM. GETFIRSTCHILD	DBE_XMLDOM.GE TFIRSTCHILD	-
14	DBMS_XMLDOM. GETLASTCHILD	DBE_XMLDOM.GE TLASTCHILD	-
15	DBMS_XMLDOM. GETLENGTH	DBE_XMLDOM.GE TLENGTH	-
16	DBMS_XMLDOM. GETLOCALNAME	DBE_XMLDOM.GE TLOCALNAME	-
17	DBMS_XMLDOM. GETNAMEDITEM	DBE_XMLDOM.GE TNAMEDITEM	-
18	DBMS_XMLDOM. GETNEXTSIBLING	DBE_XMLDOM.GE TNEXTSIBLING	-
19	DBMS_XMLDOM. GETNODENAME	DBE_XMLDOM.GE TNODENAME	-
20	DBMS_XMLDOM. GETNODETYPE	DBE_XMLDOM.GE TNODETYPE	-
21	DBMS_XMLDOM. GETTAGNAME	DBE_XMLDOM.GE TTAGNAME	-
22	DBMS_XMLDOM. IMPORTNODE	DBE_XMLDOM.IM PORTNODE	-
23	DBMS_XMLDOM. ISNULL	DBE_XMLDOM.ISN ULL	GaussDB:入参为 DOMNODELIST类型时,若对象 在哈希表中不存在会发生报错。 Oracle:不会报错。
24	DBMS_XMLDOM. ITEM	DBE_XMLDOM.ITE M	-

序号	Oracle数据库	GaussDB数据库	差异
25	DBMS_XMLDOM. MAKENODE	DBE_XMLDOM.MA KENODE	GaussDB: 该函数不支持直接作 为函数返回值返回。 Oracle: 支持直接作为函数返回值 返回。
26	DBMS_XMLDOM. NEWDOMDOCU MENT	DBE_XMLDOM.NE WDOMDOCUMEN T	 GaussDB入参大小需限制1GB以内。Oracle:与CLOB类型大小一致。 GaussDB 目前暂不支持外部DTD解析。Oracle:支持解析外部DTD。 GaussDB newdomdocument创建的doc,默认UTF-8字符集。Oracle:根据服务端字符集生成。 GaussDB 从同一个xmltype实例中解析出的每一个doc都是新到xmltype。Oracle:从同一个xmltype实例中解析出关系。 GaussDB 从同一个xmltype实例中解析出关系。 GaussDB version字段只支持1.0,1.0-1.9解析警告但正常执行,1.9以上报错。Oracle:不报错。 GaussDB 与Oracle数据库DTD校验差异:!ATTLIST to type(CHECK check Check)"Ch"将报错,内型以值"Ch"不属于括号中校举值,而Oracle数据库不报错。 ENTITY baidu "www.baidu.com">
27	DBMS_XMLDOM. SETATTRIBUTE	DBE_XMLDOM.SE TATTRIBUTE	GaussDB:属性key不支持为null 或空字符串。 Oracle:属性key允许为null或空 字符串。

序号	Oracle数据库	GaussDB数据库	差异
28	DBMS_XMLDOM. SETCHARSET	DBE_XMLDOM.SE TCHARSET	GaussDB目前支持的字符集有: UTF-8、UTF-16、UCS-4、 UCS-2、ISO-8859-1、 ISO-8859-2、ISO-8859-3、 ISO-8859-4、ISO-8859-5、 ISO-8859-6、ISO-8859-7、 ISO-8859-8、ISO-8859-9、 ISO-2022-JP、Shift_JIS、EUC-JP、ASCII。输入其他字符集会报 错或者可能导致输出乱码
29	DBMS_XMLDOM. SETDOCTYPE	DBE_XMLDOM.SE TDOCTYPE	GaussDB name、sysid、pubid的 总长度限制在32500个字节以内。 Oracle:限制在32767字节内。
30	DBMS_XMLDOM. WRITETOBUFFER	支持,有差异, DBE_XMLDOM.WR ITETOBUFFER	 GaussDB: writetobuffer输出buffer限制在1GB以内。Oracle: 限制在32767字节内。 GaussDB: 输出doc将包含XML声明version和encoding。Oracle: 用户不主动指定将不包含。 GaussDB: 入参为domnode类型时,如果节点是doc转换的,输出节点将包含XML声明version和encoding。Oracle: 用户不主动指定将不包含。 GaussDB: 默认以UTF-8字符集输出xml。Oracle: 根据数据库字符集生

序号	Oracle数据库	GaussDB数据库	差异
31	DBMS_XMLDOM. WRITETOCLOB	DBE_XMLDOM.WR ITETOCLOB	● GaussDB: writetoclob大小支 持1GB以内。 Oracle: 按CLOB大小支持。
			 GaussDB:输出doc将包含XML 声明version和encoding。 Oracle:用户不主动指定将不包含。
			 GaussDB:入参为domnode类型时,如果节点是doc转换的,输出节点将包含XML声明version和encoding。 Oracle:用户不主动指定将不包含。
			 GaussDB 默认以UTF-8字符集 输出xml。 Oracle:根据数据库字符集生成。

序号	Oracle数据库	GaussDB数据库	差异
32	DBMS_XMLDOM. WRITETOFILE	DBE_XMLDOM.WR ITETOFILE	 GaussDB: document入参,filename长度限制在255个字节以内,charset请参考dbe_xmldom.setcharset接口。 Oracle: filename长度限制受操作系统影响,大于255个字节。
			 GaussDB: domnode入参, filename长度限制在255个字节以内, charset请参考dbe_xmldom.setcharset接口。 Oracle: filename长度限制受操作系统影响,大于255个字节。
			• GaussDB:该函数会添加缩进等内容,将输出格式化。输出doc将包含XML声明version和encoding。入参为domnode类型时,如果节点是doc转换的,输出节点将包含XML声明version和encoding。Oracle:用户不主动指定将不包含。
			 GaussDB:传入 newdomdocument()无参创建 的doc,在不指定charset时不 会报错,默认UTF-8字符集。 Oracle:会进行报错。
			 GaussDB: filename需要在 pg_directory中创建的路径下, filename中的\会被转换成/, 只允许存在一个/。文件名格式 应为pg_directory_name/ file_name。 Oracle: 按用户输入不进行转 义。
33	DBMS_XMLDOM. GETNODEVALUE	DBE_XMLDOM.GE TNODEVALUE	-
34	DBMS_XMLDOM. GETPARENTNOD E	DBE_XMLDOM.GE TPARENTNODE	-
35	DBMS_XMLDOM. HASCHILDNODE S	DBE_XMLDOM.HA SCHILDNODES	-

序号	Oracle数据库	GaussDB数据库	差异
36	DBMS_XMLDOM. MAKEELEMENT	DBE_XMLDOM.MA KEELEMENT	-
37	DBMS_XMLDOM. SETNODEVALUE	DBE_XMLDOM.SE TNODEVALUE	 GaussDB: nodeValue入参,可以输入空字符串和NULL值,但不会对节点值进行修改。Oracle: 空字符串和NULL会将节点值修改为空字符串。 GaussDB: nodeValue入参,暂不支持转义字符'&',如字符串中包含该转义字符,会清空节点值。Oracle: 支持转义字符。
38	DBMS_XMLDOM. GETELEMENTSBY TAGNAM	DBE_XMLDOM.GE TELEMENTSBYTAG NAME	-

表 1-110 DBMS_XMLPARSER 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLPARS ER.FREEPARSER	DBE_XMLPARSER.F REEPARSER	-
2	DBMS_XMLPARS ER.GETDOCUME NT	DBE_XMLPARSER.G ETDOCUMENT	-
3	DBMS_XMLPARS ER.GETVALIDATI ONMODE	DBE_XMLPARSER.G ETVALIDATIONMO DE	-
4	DBMS_XMLPARS ER.NEWPARSER	支持,有差异, DBE_XMLPARSER. NEWPARSER	GaussDB中parser对象的数量上限 为16777215,Oracle中约为1亿。

序号	Oracle数据库	GaussDB数据库	差异
5	DBMS_XMLPARS ER.PARSEBUFFE R	支持,有差异, DBE_XMLPARSER.P ARSEBUFFER	● 与Oracle数据库解析字段差异:字符串encoding只支持UTF-8; version字段只支持1.0,1.0-1.9解析警告但正常执行,1.9以上报错。 ● 与Oracle数据库命名空间校验差异:解析未声明的命名空间标签正常执行,而Oracle数据库会报错。 ● 与Oracle数据库xml预定义实体解析差异: '"会被解析转义为字符'",而Oracle中预定义实体统一都没有转义为字符。 ● 与Oracle数据库DTD校验差异: - !ATTLIST to type (CHECK check Check) "Ch"将报错,因默认值"Ch"不属于括号中枚举值,而Oracle数据库不报错。 - ENTITY baidu "www.baidu.com" &Baidu&writer将报错,因区分字母大小写,Baidu无法与baidu对应,而Oracle数据库不报错。

序号	Oracle数据库	GaussDB数据库	差异
6	DBMS_XMLPARS 支持,有差异, ER.PARSECLOB DBE_XMLPARSER.P	ER.PARSECLOB DBE_XMLPARSER.P	● PARSECLOB不支持解析大于等于2GB的clob。
		ARSECLOB	● 与Oracle数据库解析字段差 异:字符串encoding只支持 UTF-8;version字段只支持 1.0,1.0-1.9解析警告但正常执 行,1.9以上报错。
			与Oracle数据库命名空间校验 差异:解析未声明的命名空间 标签正常执行,而Oracle数据 库会报错。
			● 与Oracle数据库xml预定义实体解析差异: '"会被解析转义为字符'",而Oracle中预定义实体统一都没有转义为字符。
			● 与Oracle数据库DTD校验差 异:
			 !ATTLIST to type (CHECK check Check) "Ch"将报错,因默认值"Ch"不属于括号中枚举值,而Oracle数据库不报错。
			<!--ENTITY baidu</li-->"www.baidu.com">&Baidu&writer将报错,因区分字母大小写,Baidu无法与baidu对应,而Oracle数据库不报错。
7	DBMS_XMLPARS ER.SETVALIDATI ONMODE	DBE_XMLPARSER.S ETVALIDATIONMO DE	-

表 1-111 DBMS_ILM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_ILM.ADD _TO_ILM	不支持	-
2	DBMS_ILM.ARC HIVESTATENAM E	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
3	DBMS_ILM.EXEC UTE_ILM	DBE_ILM.EXECUTE _ILM	GaussDB数据库的入参schema 在Oracle数据库中对应为 owner。
			GaussDB数据库不支持指定 ilm_scope(一次指定多个对象)的操作。
4	DBMS_ILM.EXEC UTE_ILM_TASK	不支持	-
5	DBMS_ILM.PREV IEW_ILM	不支持	-
6	DBMS_ILM.REM OVE_FROM_ILM	不支持	-
7	DBMS_ILM.STOP _ILM	DBE_ILM.STOP_IL M	-

表 1-112 DBMS_ILM_ADMIN 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_ILM_ADMI N.CLEAR_HEAT_ MAP_ALL	不支持	-
2	DBMS_ILM_ADMI N.CLEAR_HEAT_ MAP_TABLE	不支持	-
3	DBMS_ILM_ADMI N.CUSTOMIZE_IL M	DBE_ILM_ADMIN. CUSTOMIZE_ILM	入参parameter取值对应的特性参数存在差异。 GaussDB数据库param取值支持1、2、7、11、12、13、14和15。 GaussDB数据库param取值为14时,对应的特性参数为WIND_DURATION,用于控制自动调度中执行窗口的持续时长,而ORACLE数据库无对应的特性参数。
4	DBMS_ILM_ADMI N.DISABLE_ILM	DBE_ILM_ADMIN. DISABLE_ILM	-
5	DBMS_ILM_ADMI N.ENABLE_AUTO _OPTIMIZE	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
6	DBMS_ILM_ADMI N.ENABLE_ILM	DBE_ILM_ADMIN. ENABLE_ILM	-
7	DBMS_ILM_ADMI N.	不支持	-
	IGNORE_AUTO_O PTIMIZE_		
	CRITERIA		
8	DBMS_ILM_ADMI N.SET_HEAT_MA P_ALL	不支持	
9	DBMS_ILM_ADMI N.SET_HEAT_MA P_START	不支持	
10	DBMS_ILM_ADMI N.SET_HEAT_MA P_TABLE	不支持	-

表 1-113 DBMS_COMPRESSION 兼容性说明

序号 O	Dracle数据库	GaussDB数据库	差异
S:	DBMS_COMPRE SION.GET_COM PRESSION_RATI)	DBE_COMPRESSIO N.GET_COMPRESS ION_RATIO	 GaussDB不支持LOBs的压缩率获取。 对于单个对象的压缩率获取: GaussDB入参comptype取值仅支持1(未压缩)和2(高级压缩),Oracle还支持1024、2048等取值。 GaussDB入参objtype取值仅支持1(表对象),而Oracle还支持2(索引对象)。 Oracle还支持2(索引对象)。 Oracle数据库使用subset_numrows参数直接来决定采样的行数(即为参数的取值),而GaussDB则使用sample_ratio(采样率)来间接确定采样的行数。

序号	Oracle数据库	GaussDB数据库	差异
2	DBMS_COMPRE SSION.GET_COM PRESSION_TYPE	DBE_COMPRESSIO N.GET_COMPRESS ION_TYPE	 Oracle使用rowid来指定待获取 压缩类型的行,而GaussDB则 是使用行的ctid来指定。
			 返回值为comptype, 其取值差 异同 GET_COMPRESSION_RATIO。
			● GaussDB该接口仅支持在DN调用,详见GaussDB《开发指南》中"存储过程>高级包>二次封装接口(推荐)>DBE_COMPRESSION"章节。

表 1-114 DBMS_HEAT_MAP 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_HEAT_MAP.B LOCK_HEAT_MAP	不支持	-
2	DBMS_HEAT_MAP.E XTENT_HEAT_MAP	不支持	-
3	DBMS_HEAT_MAP. OBJECT_HEAT_MAP	不支持	-
4	DBMS_HEAT_MAP.S EGMENT_HEAT_MA P	不支持	-
5	DBMS_HEAT_MAP.T ABLESPACE_HEAT_ MAP	不支持	-
6	不支持	DBE_HEAT_MA P.ROW_HEAT_ MAP	GaussDB该接口仅支持在DN调用,详见GaussDB《 开发指南 》中"存储过程> 高级包 > 二次封装接口(推荐)> DBE_HEAT_MAP"章节。

表 1-115 DBMS_DESCRIBE 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_DESCRI BE.DESCRIBE_P ROCEDURE	DBE_DESCRIBE.D ESCRIBE_PROCED URE	 datatype参数与Oracle存在差 异,GaussDB返回数据类型的 oid,Oracle数据库返回Oracle数 据库内部的数据类型的编号。
			 datalength、dataprecision和 scale因GaussDB创建存储过程或 函数时无法保留类型的约束(如 number(7,2)、varchar2(20) 等),该三个参数置0处理; Oracle可使用%type方法获得带 约束的数据类型。
			 GaussDB的具体使用请参见《开 发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_DESCRIBE"章节。

表 1-116 DBMS_STATS 兼容性说明

序号	Oracle数据库	GaussDB数据 库	差异
1	DBMS_STATS.ALTER_STATS_ HISTORY_RETENTION	不支持	-
2	DBMS_STATS.CANCEL_ADVI SOR_TASK	不支持	-
3	DBMS_STATS.CONFIGURE_ ADVISOR_FILTER	不支持	-
4	DBMS_STATS.CONFIGURE_ ADVISOR_OBJ_FILTER	不支持	-
5	DBMS_STATS.CONFIGURE_ ADVISOR_OPR_FILTER	不支持	-
6	DBMS_STATS.CONFIGURE_ ADVISOR_RULE_FILTER	不支持	-
7	DBMS_STATS.CREATE_ADVI SOR_TASK	不支持	-
8	DBMS_STATS.CONVERT_RA W_VALUE	不支持	-
9	DBMS_STATS.CONVERT_RA W_VALUE_NVARCHAR	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
10	DBMS_STATS.CONVERT_RA W_VALUE_ROWID	不支持	-
11	DBMS_STATS.COPY_TABLE_ STATS	不支持	-
12	DBMS_STATS.CREATE_EXTE NDED_STATS	不支持	-
13	DBMS_STATS.CREATE_STAT _TABLE	DBE_STATS.C REATE_STAT_ TABLE	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
14	DBMS_STATS.DELETE_COLU MN_STATS	DBE_STATS.D ELETE_COLU MN_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 GaussDB中,使用该接口可以设置表达式统计信息,但tabname应传该表达式对应的索引名。
15	DBMS_STATS.DELETE_DATA BASE_PREFS	不支持	-
16	DBMS_STATS.DELETE_DATA BASE_STATS	不支持	-
17	DEDBMS_STATS.DELETE_DI CTIONARY_STATS	不支持	-
18	DBMS_STATS.DELETE_FIXED _OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
19	DBMS_STATS.DELETE_INDE X_STATS	DBE_STATS.D ELETE_INDEX _STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
20	DBMS_STATS.DELETE_PEND ING_STATS	不支持	-
21	DBMS_STATS.DELETE_PROC ESSING_RATE	不支持	-
22	DBMS_STATS.DELETE_SCHE MA_PREFS	不支持	-
23	DBMS_STATS.DELETE_SCHE MA_STATS	DBE_STATS.D ELETE_SCHE MA_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
24	DBMS_STATS.DELETE_SYST EM_STATS	不支持	-
25	DBMS_STATS.DELETE_TABL E_PREFS	不支持	-
26	DBMS_STATS.DELETE_TABL E_STATS	DBE_STATS.D ELETE_TABLE_ STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
27	DBMS_STATS.DIFF_TABLE_S TATS_IN_HISTORY	不支持	-
28	DBMS_STATS.DIFF_TABLE_S TATS_IN_PENDING	不支持	-
29	DBMS_STATS.DIFF_TABLE_S TATS_IN_STATTAB	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
30	DBMS_STATS.DROP_ADVIS OR_TASK	不支持	-
31	DBMS_STATS.DROP_EXTEN DED_STATS	不支持	-
32	DBMS_STATS.DROP_STAT_T ABLE	DBE_STATS.D ROP_STAT_TA BLE	-
33	DBMS_STATS.EXECUTE_ADV ISOR_TASK	不支持	-
34	DBMS_STATS.EXPORT_COL UMN_STATS	DBE_STATS.E XPORT_COLU MN_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发》中"存储过程等口(推荐工)》,可以推荐工义。 DBE_STATS"章节。 导出的列级统表中的是索引表的结合是索引表的是索引的是索引的是索引的是索引的是索引的是索引的是索引的是索引的是索引的是索引
35	DBMS_STATS.EXPORT_DAT ABASE_PREFS	不支持	-
36	DBMS_STATS.EXPORT_DAT ABASE_STATS	不支持	-
37	DBMS_STATS.EXPORT_DICTI ONARY_STATS	不支持	-
38	DBMS_STATS.EXPORT_FIXE D_OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
39	DBMS_STATS.EXPORT_INDE X_STATS	DBE_STATS.E XPORT_INDE X_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程>高级包>二次封装接口(推荐)>DBE_STATS"章节。 stattab表中,导出的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
40	DBMS_STATS.EXPORT_PEN DING_STATS	不支持	-
41	DBMS_STATS.EXPORT_SCHE MA_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
42	DBMS_STATS.EXPORT_SCHE MA_STATS	DBE_STATS.E XPORT_SCHE MA_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程>同级包>二次封装接口(推荐)>DBE_STATS"章节。 stattab表中,导出的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。导出表列级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 权限:需要具有stattab表的siud权限。
43	DBMS_STATS.EXPORT_SYST EM_STATS	不支持	-
44	DBMS_STATS.EXPORT_TABL E_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
45	DBMS_STATS.EXPORT_TABL E_STATS	DBE_STATS.E XPORT_TABLE _STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程口(推荐)>DBE_STATS"章节。 stattab表中,导出的表、分区级东计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。级联导的列级统计信息为g_statistic表和pg_statistic表和pg_statistic_ext表保持一致。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
46	DBMS_STATS.FLUSH_DATA BASE_MONITORING_INFO	不支持	-
47	DBMS_STATS.GATHER_DAT ABASE_STATS	不支持	-
48	DBMS_STATS.GATHER_DICT IONARY_STATS	不支持	-
49	DBMS_STATS.GATHER_FIXE D_OBJECTS_STATS	不支持	-
50	DBMS_STATS.GATHER_IND EX_STATS	不支持	-
51	DBMS_STATS.GATHER_PRO CESSING_RATE	不支持	-
52	DBMS_STATS.GATHER_SCH EMA_STATS	不支持	-
53	DBMS_STATS.GATHER_SYST EM_STATS	不支持	-
54	DBMS_STATS.GATHER_TABL E_STATS	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
55	DBMS_STATS.GENERATE_ST ATS	不支持	-
56	DBMS_STATS.GET_ADVISOR _OPR_FILTER	不支持	-
57	DBMS_STATS.GET_ADVISOR _RECS	不支持	-
58	DBMS_STATS.GET_COLUMN _STATS	不支持	-
59	DBMS_STATS.GET_INDEX_S TATS	不支持	-
60	DBMS_STATS.GET_PARAM	不支持	-
61	DBMS_STATS.GET_PREFS	不支持	-
62	DBMS_STATS.GET_STATS_HI STORY_AVAILABILITY	DBE_STATS.G ET_STATS_HIS TORY_AVAILA BILITY	GaussDB查询到的是全库 存在的最早历史统计信息 的收集时间。
63	DBMS_STATS.GET_STATS_HI STORY_RETENTION	DBE_STATS.G ET_STATS_HIS TORY_RETEN TION	-
64	DBMS_STATS.GET_SYSTEM_ STATS	不支持	-
65	DBMS_STATS.GET_TABLE_S TATS	不支持	-
66	DBMS_STATS.IMPLEMENT_ ADVISOR_TASK	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
67	DBMS_STATS.IMPORT_COL UMN_STATS	DBE_STATS.I MPORT_COL UMN_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程子口(推荐)> DBE_STATS"章节。 导出单列col导出的统保持一致。多列ext-col导出的统计信息与pg_statistic_ext表保持一致。 支持息。要求tabname传的是索引名称,colname传的是索引名称,colname传的是索引表达式名称。 权限:需要具有查询表的ANALYZE权限以及stattab表的siud权限。
68	DBMS_STATS.IMPORT_DAT ABASE_PREFS	不支持	-
69	DBMS_STATS.IMPORT_DAT ABASE_STATS	不支持	-
70	DBMS_STATS.IMPORT_DICT IONARY_STATS	不支持	-
71	DBMS_STATS.IMPORT_FIXE D_OBJECTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
72	DBMS_STATS.IMPORT_INDE X_STATS	DBE_STATS.I MPORT_INDE X_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 stattab表中,导入的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
73	DBMS_STATS.IMPORT_SCH EMA_PREFS	不支持	-
74	DBMS_STATS.IMPORT_SCH EMA_STATS	DBE_STATS.I MPORT_SCHE MA_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程>二(推荐)>DBE_STATS"章节。 stattab表中,导入的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。导入表外级统计信息与pg_statistic表和pg_statistic_ext表保持一致。 权限:需要具有stattab表的siud权限。

序号	Oracle数据库	GaussDB数据 库	差异	
75	DBMS_STATS.IMPORT_SYST EM_STATS	不支持	-	
76	DBMS_STATS.IMPORT_TABL E_PREFS	不支持	-	
77	DBMS_STATS.IMPORT_TABL E_STATS	DBE_STATS.I MPORT_TABL E_STATS	_TABL 应传schema名。	
78	DBMS_STATS.INTERRUPT_A DVISOR_TASK	不支持	-	
79	DBMS_STATS.LOCK_PARTITI		• GaussDB中ownname 应传schema名。	
80	DBMS_STATS.LOCK_SCHEM A_STATS	DBE_STATS.L OCK_SCHEM A_STATS	• GaussDB中ownname 应传schema名。	
81	DBMS_STATS.LOCK_TABLE_ STATS	DBE_STATS.L OCK_TABLE_S TATS	• GaussDB中ownname 应传schema名。	
82	DBMS_STATS.MERGE_COL_ USAGE	不支持	-	
83	DBMS_STATS.PREPARE_COL UMN_VALUES	不支持	-	

序号	Oracle数据库	GaussDB数据 库	差异	
84	DBMS_STATS.PREPARE_COL UMN_VALUES_ROWID	不支持	-	
85	DBMS_STATS.PUBLISH_PEN DING_STATS	不支持	-	
86	DBMS_STATS.PURGE_STATS	DBE_STATS.P URGE_STATS	-	
87	DBMS_STATS.REMAP_STAT_ TABLE	不支持	-	
88	DBMS_STATS.REPORT_ADVI SOR_TASK	不支持	-	
89	DBMS_STATS.REPORT_COL_ USAGE	不支持	-	
90	DBMS_STATS.REPORT_GAT 不支持 - HER_AUTO_STATS		-	
91	DBMS_STATS.REPORT_GAT 不支持 - HER_DATABASE_STATS		-	
92	DBMS_STATS.REPORT_GAT HER_DICTIONARY_STATS	不支持	-	
93	DBMS_STATS.REPORT_GAT HER_FIXED_OBJ_STATS	不支持	-	
94	DBMS_STATS.REPORT_GAT HER_SCHEMA_STATS	不支持	-	
95	DBMS_STATS.REPORT_STAT S_OPERATIONS	不支持	-	
96	DBMS_STATS.RESET_ADVIS OR_TASK	不支持	-	
97	DBMS_STATS.RESET_COL_U 不支持 - SAGE		-	
98	DBMS_STATS.RESET_GLOBA 不支持 - L_PREF_DEFAULTS -		-	
99	DBMS_STATS.RESET_PARA M_DEFAULTS	不支持	-	
100	DBMS_STATS.RESTORE_DIC TIONARY_STATS	不支持	-	
101	DBMS_STATS.RESTORE_FIXE D_OBJECTS_STATS	不支持	-	

序号	Oracle数据库	GaussDB数据 库	差异	
102	DBMS_STATS.RESTORE_SCH EMA_STATS	DBE_STATS.R ESTORE_SCH EMA_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 	
103	DBMS_STATS.RESTORE_SYS TEM_STATS	不支持	-	
104	DBMS_STATS.RESTORE_TAB LE_STATS	DBE_STATS.R ESTORE_TABL E_STATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 	
105	DBMS_STATS.RESUME_ADV ISOR_TASK	不支持	-	
106	DBMS_STATS.SCRIPT_ADVIS OR_TASK	不支持	-	
107	DBMS_STATS.SEED_COL_US AGE	不支持	-	
108	DBMS_STATS.SET_ADVISOR _TASK_PARAMETER	不支持	-	
109	DBMS_STATS.SET_COLUMN _STATS	DBE_STATS.SE T_COLUMN_S TATS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 	
110	DBMS_STATS.SET_DATABAS E_PREFS	不支持	-	
111	DBMS_STATS.SET_GLOBAL_ PREFS	不支持	-	

序号	Oracle数据库	GaussDB数据 库	差异	
112	ATS T_INDEX_STA		应传schema名。 应传schema名。 GaussDB仅支持部分入 参功能,详见《开发指 南》中"存储过程 > 高 级包 > 二次封装接口	
113	DBMS_STATS.SET_PARAM	不支持	-	
114	DBMS_STATS.SET_PROCESSI NG_RATE	不支持	-	
115	DBMS_STATS.SET_SCHEMA_ PREFS	不支持	-	
116	DBMS_STATS.SET_SYSTEM_ STATS	不支持	-	
117	DBMS_STATS.SET_TABLE_PR EFS	不支持	-	
118	DBMS_STATS.SET_TABLE_ST ATS	DBE_STATS.SE T_TABLE_STA TS	 GaussDB中ownname 应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 GaussDB中新增了relallvisible入参。 	
119	DBMS_STATS.SHOW_EXTEN DED_STATS_NAME	不支持	-	
120	DBMS_STATS.TRANSFER_ST ATS	不支持	-	
121	DBMS_STATS.UNLOCK_PAR TITION_STATS	DBE_STATS.U NLOCK_PARTI TION_STATS	GaussDB中ownname应传 schema名。	
122	DBMS_STATS.UNLOCK_SCH EMA_STATS	DBE_STATS.U NLOCK_SCHE MA_STATS	GaussDB中ownname应传 schema名。	

序号	Oracle数据库	GaussDB数据 库	差异
123	DBMS_STATS.UNLOCK_TAB LE_STATS	DBE_STATS.U NLOCK_TABL E_STATS	GaussDB中ownname应传 schema名。
124	DBMS_STATS.UPGRADE_ST AT_TABLE	不支持	-

表 1-117 DBMS_XMLGEN 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLGE N.CONVERT	DBE_XMLGEN.CON VERT	-
2	DBMS_XMLGE N.NEWCONTEX T	DBE_XMLGEN.NEW CONTEXT	-
3	DBMS_XMLGE N.NEWCONTEX TFROMHIERAR CHY	DBE_XMLGEN.NEW CONTEXTFROMHIE RARCHY	 GaussDB生成的递归XML最大深度不能超过5000万层。 Oracle的newcontextfromhierarchy方法对于connect by语句生成的xml是带xml头的,但是对于直接构造的数据不带xml头,GaussDB均带xml头。
4	DBMS_XMLGE N.SETCONVERT SPECIALCHARS	DBE_XMLGEN.SETC ONVERTSPECIALCH ARS	-
5	DBMS_XMLGE N.SETNULLHA NDLING	DBE_XMLGEN.SETN ULLHANDLING	-
6	DBMS_XMLGE N.SETROWSET TAG	DBE_XMLGEN.SETR OWSETTAG	-
7	DBMS_XMLGE N.SETROWTAG	DBE_XMLGEN.SETR OWTAG	-
8	DBMS_XMLGE N.USENULLATT RIBUTEINDICA TOR	DBE_XMLGEN.USEN ULLATTRIBUTEINDI CATOR	-
9	DBMS_XMLGE N.USEITEMTAG SFORCOLL	DBE_XMLGEN.USEIT EMTAGSFORCOLL	-

序号	Oracle数据库	GaussDB数据库	差异
10	DBMS_XMLGE N.GETNUMRO WSPROCESSED	DBE_XMLGEN.GETN UMROWSPROCESSE D	-
11	DBMS_XMLGE N.SETMAXROW S	DBE_XMLGEN.SETM AXROWS	-
12	DBMS_XMLGE N.SETSKIPROW S	DBE_XMLGEN.SETS KIPROWS	-
13	DBMS_XMLGE N.RESTARTQUE RY	DBE_XMLGEN.REST ARTQUERY	GaussDB分布式中不允许cursor 反向移动,因此restartquery函数 不可使用。
14	DBMS_XMLGE N.GETXMLTYPE	DBE_XMLGEN.GETX MLTYPE	-
15	DBMS_XMLGE N.GETXML	DBE_XMLGEN.GETX ML	-
16	DBMS_XMLGE N.CLOSECONT EXT	DBE_XMLGEN.CLOS ECONTEXT	-

2 分布式版 MySQL 兼容性说明

2.1 概述

本手册为GaussDB数据库(MySQL兼容性MySQL模式和MySQL兼容性M-Compatibility模式)与MySQL 5.7数据库信息对比。其中,**MySQL兼容性MySQL模式**为MySQL模式下的MySQL数据库兼容性;**MySQL兼容性M-Compatibility模式**为M-Compatibility模式下的MySQL数据库兼容性说明。

2.2 MySQL 兼容性 MySQL 模式

2.2.1 MySQL 兼容性 MySQL 模式概述

本章节主要介绍GaussDB数据库的MySQL兼容性MySQL模式(即sql_compatibility = 'MYSQL'、且设置参数b_format_version='5.7'、b_format_dev_version='s1'时)与MySQL5.7数据库的兼容性对比信息。仅介绍503.0.0版本后新增的兼容性特性,特性的相关规格和约束建议在开发指南中查看。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

由于GaussDB数据库与MySQL数据库底层框架实现存在差异,GaussDB数据库与MySQL数据库仍存在部分差异。

2.2.2 数据类型

GaussDB数据库的数据类型大部分功能场景与MySQL一致,但存在部分差异。

除特别说明,部分数据类型精度、标度、位数大小等不支持用浮点型数值定义, 建议使用合法的整型数值定义。

2.2.2.1 数值数据类型

表 2-1 整数类型

序号	MySQL数据	GaussDB数	差异
	库	据库	
1	BOOL	不完全兼容	MySQL: BOOL/BOOLEAN类型实际映射为TINYINT类型。
2	BOOLEAN	不完全兼容	GaussDB: 支持BOOL, 其中:
			 "真"值的有效文本值是: TRUE、't'、 'true'、'y'、'yes'、'1'、'TRUE'、true、 'on'以及所有非0数值。 "假"值的有效文本值是: FALSE、'f'、
			False'、'n'、'no'、'0'、0、'FALSE'、 false、'off'。
			使用TRUE和FALSE是比较规范的用法(也是 SQL兼容的用法)。
3	TINYINT[(M)] [UNSIGNED]	支持	详细请参见说明。
4	SMALLINT[(M)] [UNSIGNED]	支持	详细请参见说明。
5	MEDIUMIN	支持	MySQL存储MEDIUMINT数据需要3字节。
	T[(M)] [UNSIGNED		● 带符号的范围是-8,388,608 ~ +8,388,607。
]		● 无符号的范围是0~+16,777,215。
			GaussDB映射为INT类型,存储需要4字节。
			● 带符号的范围是-2,147,483,648 ~ +2,147,483,647。
			● 无符号的范围是0~+4,294,967,295。
			其他差异请参见说明。
6	INT[(M)] [UNSIGNED]	支持	详细请参见说明。
7	INTEGER[(M)] [UNSIGNED]	支持	详细请参见说明。

序号	MySQL数据 库	GaussDB数 据库	差异
8	BIGINT[(M)] [UNSIGNED]	支持	详细请参见说明。

□ 说明

• 输入格式:

MySQL

对于类似"asbd"、"12dd"、"12 12"等字符场景的输入,会采取截断或返回0值 并上报WARNING处理,在严格模式插表时失败。

GaussDB

- 整数类型(TINYINT、SMALLINT、MEDIUMINT、INT、INTEGER、BIGINT)的 输入,当非法字符串部分被截断时,如"12@3",会直接截断并无提示信息,插 表成功。
- 当整数类型全部被截断(如"@123")或字符串为空时,返回0,且插表成功。

操作符:

● +、-、*操作符

GaussDB: INT/INTEGER/SMALLINT/BIGINT在进行运算时,返回值为类型本身,不会向上提升类型,当返回值超范围时报错。

MySQL: 支持提升类型到BIGINT后计算。

● |、&、^、~运算符

GaussDB: 在类型所占用BIT位中计算; GaussDB中^表示指数运算,如需使用异或运算符,使用#替换。

MySQL: 提升类型计算。

• 负数显示类型转换:

GaussDB: 宽松模式结果为0, 严格模式报错。

MySQL: 依据其对应的二进制将最高位替换成数值位计算结果,例如(-1)::uint4 = 4294967295。

• 其他差异:

INT[(M)]精度,MySQL控制格式化输出,GaussDB仅语法支持,不支持功能。

- 聚集函数:
 - variance: GaussDB表示样本方差, MySQL表示总体方差。
 - stddev: GaussDB表示样本标准差,MySQL表示总体标准差。

● 显示宽度:

- 在为整型数字列指明宽度信息时,如果不同时指定ZEROFILL,则宽度信息在表结构描述中不显示。
- INSERT语句插入字符类型字段,GaussDB统一补齐0后插入。
- JOIN USING语句,涉及类型推导,MySQL默认第一张表列,GaussDB若结果为有符号 类型则宽度信息失效,否则为第一张表字段宽度。
- greatest/least、ifnull/if、case when/decode,MySQL不补齐0,GaussDB在类型及宽度信息一致时补齐0。
- 作为函数/存储过程出入参、返回值时,MySQL支持功能、GaussDB语法不报错功能不支持。

表 2-2 任意精度类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DECIMAL[(M[,D])]	支持	 操作符: GaussDB中 "^"表示指数运算, 如需使用异或运算符,使用 "#"替换;
2	NUMERIC[(M[,D])]	支持	MySQL中"^"表示异或。 ● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
3	DEC[(M[,D])	支持	 输入格式: 当字符串入参全部被截断时不会报错,如'@123'; 只有被部分截断时才会报错,如'12@3'。
4	FIXED[(M[,D])]	不支持	-

表 2-3 浮点类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	FLOAT[(M,D)]	支持	 分区表支持: FLOAT数据类型不支持KEY键值分区策略分区表。 操作符: GaussDB中 "^"表示指数运算,如需使用异或运算符,使用 "#"替换;MySQL中 "^"表示异或。 取值范围: 精度M,标度D不支持浮点型数值输入,只支持整型数值输入。 输出格式: 对于非法入参一律报错ERROR,不会在sql_mode="的宽松模式下
2	FLOAT(p)	支持	报WARNING。 • 分区表支持: FLOAT数据类型不支持KEY键值分区策略分区表。 • 操作符: 数值类型使用^操作符,与MySQL不一致,GaussDB中^操作符为取指数运算。 • 取值范围: 定义精度p时,仅支持使用合法的整型数据类型。 • 输出格式: 对于非法入参一律报错ERROR,不会在sql_mode=''的宽松模式下报WARNING。

序号	MySQL数据 库	GaussDB数 据库	差异
3	DOUBLE[(M ,D)]	支持	分区表支持: DOUBLE数据类型不支持KEY 键值分区策略分区表。
			● 操作符: GaussDB中 "^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或。
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode=''的宽松模式下 报WARNING。
4	DOUBLE PRECISION[(M,D)]	支持	• 操作符: GaussDB中 "^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode=''的宽松模式下 报WARNING。
5	REAL[(M,D)	支持	分区表支持: REAL数据类型不支持KEY值 分区策略分区表。
			● 操作符:GaussDB中"^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode=''的宽松模式下 报WARNING。

表 2-4 序列整数

-	序列金数 Marson *****	C	보 묘
序号	MySQL数据 库	GaussDB数 据库	差异
1	SERIAL	不完全兼容	GaussDB中SERIAL具体介绍请见《 开发指南 》中的"数值类型"章节。
			规格上与MySQL的差异如下: CREATE TABLE test(f1 serial, f2 CHAR(20));
			 类型定义的差异,MySQL的serial是映射到 BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE,GaussDB 的serial是映射到INTEGER NOT NULL DEFAULT
			nextval('test_f1_seq'::regclass)。如: MySQL serial的定义: mysql> SHOW CREATE TABLE test\G ************************************
			Table: test Create Table: CREATE TABLE `test` (`f1` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `f2` char(20) DEFAULT NULL, UNIQUE KEY `f1` (`f1`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 1 row in set (0.00 sec)
			GaussDB serial的定义 gaussdb=# \d+ test Table "public.test" Column Type Modifiers Storage Stats target Description
			+
			Has OIDs: no Options: orientation=row, compression=no, storage_type=USTORE
			● INSERT场景下serial类型DEFAULT值的差异。如:
			MySQL插入serial的DEFAULT值 mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa'); Query OK, 1 row affected (0.00 sec)
			mysql> INSERT INTO test VALUES(10, 'aaaa'); Query OK, 1 row affected (0.00 sec)
			mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa'); Query OK, 1 row affected (0.00 sec)
			mysql> SELECT * FROM test; ++ f1 f2
			++ 1 aaaa 10 aaaa 11 aaaa

序号	MySQL数据 库	GaussDB数 据库	差异
			++ 3 rows in set (0.00 sec) GaussDB插入serial的DEFAULT值 gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa'); INSERT 0 1 gaussdb=# INSERT INTO test VALUES(10, 'aaaa'); INSERT 0 1 gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa'); INSERT 0 1 gaussdb=# SELECT * FROM test; f1

2.2.2.2 日期与时间数据类型

表 2-5 日期与时间数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DATE	支持	GaussDB支持date数据类型,与MySQL相比 规格上存在如下差异:
			● 输入格式
			- GaussDB只支持字符类型,不支持数值 类型。如支持'2020-01-01'或 '20200101'字符串格式,不支持 20200101数值输入。MySQL支持数值 输入转换为date类型。
			- 分隔符: GaussDB不支持加号"+"、 冒号":"作为年、月、日之间的分隔符,其他的符号都支持。MySQL所有符号均可作为分隔符。 分隔符混合使用的某些场景也不支持,与MySQL也有差异,如'2020-01>01','2020/01+01'等,不建议混合使用分隔符,建议使用最常用的"-"、"/"作为分隔符。
			- 无分隔符:推荐使用完整格式,如 'YYYYMMDD' 或者 'YYMMDD'。其他 不完整的格式(包括超长格式)解析的 规则与MySQL存在差异,可能报错或者 解析的结果与MySQL不一致,不推荐使 用。
			● 输出格式 GaussDB在sql_mode参数不包含 'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、日的值是0,但是输出时会按照年、月、日的顺序依次转换为合法的值,如date '0000-00-10' 转换为: 0002-12-10 BC。非法输入或者超过范围时,会报warning信息,并返回0000-00-00值。MySQL对于包含0值年、月、日的date值会原样输出。
			● 取值范围 GaussDB的范围是4713-01-01 BC ~ 5874897-12-31 AD,支持公元前的日期,宽松模式下超过范围时,返回的是0值:0000-00-00,严格模式下会报错。MySQL的范围是 0000-00-00 ~ 9999-12-31,宽松模式下超过范围后,各个场景下的表现并不一致,可能报错(如select查询语句中),也可能返回0000-00-00值(如insert时)。此差异会导致date类型作为函数入参时,函数返回的结果存在差异。

序号	MySQL数据 库	GaussDB数 据库	差异
			 ● 操作符 GaussDB仅支持date类型之间的比较操作符 "="、"!="、"<"、"<="、"<"、"<="、">"、">=",返回true或者false;date与interval类型的加法运算,返回结果为date类型;date类型;date类型;date类型;date类型;date类型之间的减法运算,返回结果为interval类型。 MySQL date类型和其他数值类型运算时,会先将date转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如: MySQL: date + 数值,先将date类型转换为数值20200101,再与1相加,结果为数值类型20200102 mysql> select date'2020-01-01' + 1; +

序号	MySQL数据 库	GaussDB数 据库	差异
2	DATETIME[(f sp)]	支持	GaussDB支持datetime数据类型,与MySQL 相比规格上存在如下差异:
			● 输入格式
			- GaussDB只支持字符类型,不支持数值 类型。如支持'2020-01-01 10:20:30.123456'或 '20200101102030.123456'字符串格 式,不支持如 20200101102030.123456的数值类型 输入。MySQL支持数值输入转换为 datetime类型。
			- 分隔符: GaussDB不支持加号"+"、 冒号":"作为年、月、日之间的分隔符,其他的符号都支持。仅支持冒号 ":"作为时、分、秒之间的分隔符,其他的符号都不支持。分隔符混合使用的某些场景也不支持,与MySQL也有差异,不推荐使用。MySQL支持所有符号作为分隔符。
			- 无分隔符: GaussDB推荐使用完整格式 'YYYYMMDDhhmiss.ffffff'。其他不完整的格式(包括超长格式)解析的规则可能与MySQL存在差异,可能报错或者解析的结果与MySQL不一致,不推荐使用。
			● 输出格式:
			- 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的 格式,格式与MySQL无差异,且不受 DateStyle参数的影响。但是对于精度 部分,如果最后几位为0,GaussDB不 显示,MySQL会显示。
			- GaussDB在sql_mode参数不包含 'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、日值是0,但是输出时会按照年、月、日的顺序依次转换为合法的值,如datetime '0000-00-10 00:00:00' 转换为: 0002-12-10 00:00:00 BC。 非法输入或者超过范围时,会报warning信息,并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的datetime值会原样输出。
			● 取值范围 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。 294277-01-09 04:00:54.775807 AD 返回 的是infinity。对于超过范围的值,严格模

序号	MySQL数据 库	GaussDB数 据库	差异
			式下GaussDB会报错,MySQL是否报错取决于使用场景。一般查询场景不报错,而执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00值,MySQL根据使用场景可能报错,也可能返回0000-00-00 00:00:00值或者null值。这个差异会导致以datetime类型为入参的函数执行结果与MySQL也存在差异。
			 精度 范围0~6,作为表列的类型时缺省为0,与 MySQL一致。对于 datetime[(p)] 'str' 表达式场景,GaussDB将(p)作为精度解析,缺省为6,将'str'按照p指定的精度格式化成datetime类型。MySQL不支持datetime[(p)] 'str'表达式。
			● 操作符 - GaussDB仅支持datetime类型之间的比较操作符"="、"!="、"<"、"<"、"<="、">"、">=",返回true或者false; datetime与interval类型的加法运算,返回结果为datetime类型; datetime与interval类型的减法运算,返回结果为datetime类型; datetime类型之间的减法运算,返回结果为interval类型。
			 MySQL datetime类型和其他数值类型运算时,会先将datetime转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如: MySQL: datetime + 数值,先将datetime类型转换为数值20201010123456,再与1相加,结果为数值类型20201010123457 mysql> select cast('2020-10-10 12:34:56.123456' as datetime) + 1;
			+
			2020-10-11 12:34:56 (1 row)

序号	MySQL数据 库	GaussDB数 据库	差异
			将datetime类型与数值的运算结果作为函数的入参,可能导致函数的结果与MySQL也存在差异。 • 类型转换相比较MySQL,GaussDB仅支持datetime类型与char(n)、varchar(n)、timestamp类型之间的相互转换、datetime到date、time类型的转换(仅赋值和显式转换)。不支持与binary、decimal、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致,参考数据类型转换章节的描述。 • 时区 GaussDB支持datetime值中携带时区信息(时区偏移或者时区名),如'2020-01-01 12:34:56.123456 +01:00'或者'2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。MySQL不支持(5.7版本不支持,8.0及之后的版本支持)。 • GaussDB的datetime数据类型的表字段实际上会被转换为timestamp(p) without time zone类型,查询表信息或者使用工具导出的表结构,其字段的数据类型显示的是timestamp(p) without time zone类型,查询表信息或者使用工具导出的表结构,其字段的数据类型显示的是timestamp(p) without time zone,而不是datetime。MySQL显示的是datetime(p)。

序号	MySQL数据 库	GaussDB数 据库	差异
3	TIMESTAMP[(fsp)]	支持	GaussDB支持timestamp数据类型,与 MySQL相比规格上存在如下差异:
			● 输入格式:
			- 只支持字符类型,不支持数值类型。如 支持'2020-01-01 10:20:30.123456'或 '20200101102030.123456'字符串格 式,不支持如 20200101102030.123456的数值类型 输入。MySQL支持数值输入转换为 timestamp类型。
			- 分隔符:不支持加号"+"、冒号":" 作为年、月、日之间的分隔符,其他的 符号都支持。仅支持冒号":"作为时、 分、秒之间的分隔符,其他的符号都不 支持。分隔符混合使用的某些场景也不 支持,与MySQL也有差异,不推荐使 用。MySQL支持所有符号作为分隔符。
			- 无分隔符:推荐使用完整格式 'YYYYMMDDhhmiss.ffffff'。其他不完整的格式(包括超长格式)解析的规则可能与MySQL存在差异,可能报错或者解析的结果与MySQL不一致,不推荐使用。
			● 输出格式:
			- 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的 格式,格式与MySQL无差异,且不受 DateStyle参数的影响。但是对于精度 部分,如果最后几位为0,GaussDB不 显示,MySQL会显示。
			- GaussDB在sql_mode参数不包含 'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、日值是0,但是输出时会按照年、月、日的顺序依次转换为合法的值,如timestamp '0000-00-10 00:00:00' 转换为: 0002-12-10 00:00:00 BC。非法输入或者超过范围时,会报warning信息,并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的timestamp值会原样输出。
			● 取值范围: 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。 294277-01-09 04:00:54.775807 AD 返回 的是infinity。对于超过范围的值,严格模 式下GaussDB会报错,MySQL是否报错取 决于使用场景。一般查询场景不报错,而

序号	MySQL数据 库	GaussDB数 据库	差异
			执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00值,MySQL根据使用场景可能报错,也可能返回0000-00-00 00:00:00值或者null值。这个差异会导致以timestamp类型为入参的函数执行结果与MySQL也存在差异。
			● 精度: 范围0~6,作为表列的类型时缺省为0,与 MySQL一致。对于 timestamp[(p)] 'str' 表达式场景:
			– GaussDB将(p)作为精度解析,缺省为 6,将'str'按照p指定的精度格式化成 timestamp类型。
			- MySQL将timestamp 'str'的含义与 GaussDB一致,缺省精度也为6。但是 将timestamp(p) 'str'解析为函数调用, p作为timestamp函数的入参,结果返 回一个timestamp类型的值,'str'作为 投影列的别名。
			● 操作符:
			- GaussDB仅支持timestamp类型之间的比较操作符"="、"!="、"<"、 "<="、">"、">=",返回true或者false; timestamp与interval类型的加法运算,返回结果为timestamp类型; timestamp与interval类型的减法运算,返回结果为timestamp类型之间的减法运算,返回结果为interval类型。
			- MySQL timestamp类型和其他数值类型运算时,会先将timestamp转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如:
			MySQL: timestamp + 数值,先将timestamp类型 转换为数值20201010123456.123456,再与1相加,结果 为数值类型20201010123457.123456 mysql> select timestamp '2020-10-10 12:34:56.123456' + 1; +
			timestamp '2020-10-10 12:34:56.123456' + 1 +
			20201010123457.123456 ++
			1 row in set (0.00 sec)
			GaussDB: timestamp + 数值,数值类型会转换为 interval类型 1 day,然后相加得到新的timestamp。 gaussdb=# select timestamp '2020-10-10 12:34:56.123456' + 1;

序号	MySQL数据 库	GaussDB数 据库	差异
			?column?

序号	MySQL数据 库	GaussDB数 据库	差异
4	TIME[(fsp)]	支持	GaussDB支持time数据类型,与MySQL相比 规格上存在如下差异:
			● 输入格式:
			- 只支持字符类型,不支持数值类型。如 支持'1 10:20:30'或'102030'字符串格 式,不支持102030数值输入。MySQL 支持数值输入转换为time类型。
			- 分隔符: GaussDB仅支持冒号 ":"作为时、分、秒之间的分隔符,其他的符号都不支持。MySQL支持所有的符号作为分隔符。
			- 无分隔符:推荐使用完整格式,如 'hhmiss.ffffff'。其他不完整的格式(包 括超长格式)解析的规则可能与MySQL 存在差异,可能报错或者解析的结果与 MySQL不一致,不推荐使用。
			- 分、秒、精度输入负数时,GaussDB数据库可能会忽略第一个负数开始的部分,涉及的部分解析为0,如: '00:00:-10' 解析结果为 '00:00:00'。 也可能报错,如: '00:00:-10000' 会解析报错。取决于输入值的范围。而MySQL数据库统一报错。
			 输出格式: 统一为hh:mi:ss.ffffff的格式,格式与 MySQL无差异。但是对于精度部分,如果 最后几位为0,GaussDB不显示,MySQL 会显示。
			• 取值范围: -838:59:59.000000 ~ 838:59:59.000000, 与MySQL一致。对于超过范围的值,宽松模式下GaussDB无论是查询还是insert/ update等DML操作,返回的值都是就近的边界值: -838:59:59或838:59:59。MySQL是查询时报错,DML操作返回的值才是就近边界值,场景上存在差异。此差异会导致time类型作为函数入参时,函数返回的结果也存在差异。
			• 精度: 范围0~6,作为表列的类型时缺省为0,与 MySQL一致。对于 time(p) 'str' 表达式场景,GaussDB将(p)作为精度解析,缺省为6,将'str'按照p指定的精度格式化成time类型。MySQL是解析为time函数,p是入参,'str'是投影列的别名。
			● 操作符:

- GaussDB仅支持time类型 作符"="、"!="、"·	
# > " 、 " >= " ,返回trime与interval类型的加结果为time类型,time与的减法运算,返回结果为time类型之间的减法运算interval类型。 - MySQL time类型和其他时,会先将time转换为数度类型运算,结果型。与GaussDB存在差异	wind with a second with a sec

序号	MySQL数据 库	GaussDB数 据库	差异
5	YEAR[(4)]	支持	GaussDB支持year数据类型,与MySQL相比 规格上存在如下差异:
			操作符:
			- GaussDB仅支持year类型之间的比较操 作符:=、!=、<、<=、>、>=。返回 true或者false。
			- GaussDB仅支持year类型与int4类型之 间的算术操作符:+、-。返回整型值, MySQL是返回无符号整型值。
			● 类型转换: 相比较MySQL,GaussDB仅支持year类型与int4类型的转换,仅支持int4、 varchar、numeric、date、time、 timestamp、timestamptz类型到year类型的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致,参考数据类型转换章节的描述。
6	INTERVAL	支持	GaussDB支持INTERVAL数据类型,但 INTERVAL在MySQL中为表达式,同时存在以 下差异:
			不支持字符串类型的日期输入作为运算,如: select '2023-01-01' + interval 1 day。
			● 不支持interval expr unit语法中,expr为 负整数或浮点数的输入,如:select date'2023-01-01' + interval -1 day。
			● 不支持interval expr unit语法中,expr为 运算表达式的输入,如:select date'2023-01-01' + interval 4/2 day。
			 interval表达式参与运算时,返回值固定为 datetime类型,MySQL为datetime或date 类型。运算的逻辑与原有GaussDB保持一 致,与MySQL有差异。
			interval expr unit语法中,expr数值支持的范围会根据unit单位的不同有所差异,最大可支持的范围为[-2147483648, 2147483647]。超过范围时,严格模式报error,宽松模式报warning并返回0值。
			• interval expr unit语法中,expr指定的字段数量大于unit预期的字段数量时,在严格模式,报error;在宽松模式,报warning并返回0值。如unit取值为DAY_HOUR,预期的字段数量为2,expr取值为'1-2-3',字段数量为3。

2.2.2.3 字符串数据类型

表 2-6 字符串数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	CHAR[(M)]	支持	 输入格式 GaussDB自定义函数参数和返回值不支持长度校验,存储过程参数不支持长度校验,同时也不支持在PAD_CHAR_TO_FULL_LENGTH打开时补齐正确的空格,MySQL支持。 GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 语法 GaussDB的 Cast (expr as char)语法无法根据输入的字符串长度转成对应的类型,只支持转成varchar类型。不支持cast(''as char())将空串转成char(0)类型。MySQL支持按长度转成对应的类型。 操作符 GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 GaussDB除以0会报错,MySQL返回null。 "~":GaussDB返回负数,MySQL返回8字节无符号整数。 "^":GaussDB表示次方幂,MySQL表示按位异或。

序号	MySQL数据 库	GaussDB数 据库	差异
2	VARCHAR(M)	支持	 输入格式 GaussDB的自定义函数参数和返回值不支持长度校验,存储过程参数不支持长度校验,所以SQL支持。 GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警,MySQL不支持。 GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 操作符 GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 GaussDB除以0会报错,MySQL返回null。 "~":GaussDB返回负数,MySQL返回8字节无符号整数。 "^":GaussDB表示次方幂,MySQL表示按位异或。
3	TINYTEXT	支持	● 输入格式 - GaussDB不支持长度限制255字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。 - GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 - 操作符 - GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 - GaussDB除以0会报错,MySQL返回null。 - "~":GaussDB返回负数,MySQL返回8字节无符号整数。 - "^":GaussDB表示次方幂,MySQL表示按位异或。

序号	MySQL数据 库	GaussDB数 据库	差异
4	TEXT	支持	● 输入格式 - GaussDB不支持长度限制65535字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。 - GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 - GaussDB除以0会报错,MySQL返回null。 - "~":GaussDB返回负数,MySQL返回8字节无符号整数。 - "^":GaussDB表示次方幂,MySQL表示按位异或。
5	MEDIUMTEX T	支持	● 输入格式 - GaussDB不支持长度限制16777215字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。 - GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 - 操作符 - GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 - GaussDB除以0会报错,MySQL返回null。 - "~":GaussDB返回负数,MySQL返回8字节无符号整数。 - "^":GaussDB表示次方幂,MySQL表示按位异或。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGTEXT	支持	 输入格式 GaussDB只支持不超过1G,MySQL支持4G-1字节长度。 GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 操作符 GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 GaussDB除以0会报错,MySQL返回null。 "~":GaussDB返回负数,MySQL返回8字节无符号整数。 "^":GaussDB表示次方幂,MySQL表示按位异或。
7	ENUM('value 1','value2',)	不支持	-
8	SET('value1',' value2',)	不支持	-

2.2.2.4 二进制数据类型

表 2-7 二进制数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BINARY[(M)]	不支持	-
2	VARBINARY(M)	不支持	-

序号	MySQL数据 库	GaussDB数 据库	差异
3	TINYBLOB	支持	 取值范围:表现规格为BYTEA类型。不支持长度限制255字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。
			● 输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常
			用逻辑运算符或、与、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^")。
4	BLOB	支持	 取值范围:表现规格为BYTEA类型。不支持长度限制65535字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。
			输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常
			用逻辑运算符或、与、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^"。
5	MEDIUMBLO B	支持	• 取值范围:表现规格为BYTEA类型。不支持长度限制16777215字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。
			● 输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常
			用逻辑运算符或、与、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^"。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGBLOB	支持	取值范围:表现规格为BYTEA类型。只支持不超过1G,具体范围参照bytea数据类型集中式和分布式规格。
			输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常 用逻辑运算符或、与、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^"。
7	BIT[(M)]	不支持	-

2.2.2.5 JSON 数据类型

表 2-8 JSON 数据类型

序号	MySQL数据 库	GaussDB数据库
1	JSON	不完全兼容

2.2.2.6 数据类型支持的属性

表 2-9 数据类型支持的属性

序号	MySQL数据库	GaussDB数据库
1	NULL	支持
2	NOT NULL	支持
3	DEFAULT	支持
4	ON UPDATE	支持
5	PRIMARY KEY	支持
6	CHARACTER SET name	支持
7	COLLATE name	支持

2.2.2.7 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换:

- 操作符(比较操作符、运算操作符等)的操作数的数据类型不一致。常见于查询 条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新(包括insert、update、merge、replace等)的目标列,数据的 类型和列的定义类型不一致。
- 显式的类型转换: cast(expr as datatype),将expr表达式类型转换为datatype类型。
- 集合运算(UNION、MINUS、EXCEPT、INTERSECT)确定最终投影列的目标数据类型后,各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景,根据不同表达式的数据类型,来决定用于比较或者最终结果的目标数据类型。
 - DECODE
 - CASE WHEN
 - lexpr [NOT] IN (expr_list)
 - BETWEEN AND
 - JOIN USING(a,b)
 - GREATEST和LEAST
 - NVL 和 COALESCE

GaussDB和MySQL数据库对于数据类型转换、转换的目标数据类型有着完全不同的规则。如下示例体现了两者处理的差异:

1. 数据类型转换规则的差异:

- GaussDB数据库对于不同数据类型之间的转换规则有明确的定义:
 - 是否支持转换:pg_cast系统表中是否定义两种类型的转换路径,没有定义则 不支持。
 - 支持转换的场景:支持任意场景转换、仅支持显式(cast表达式)转换、仅 支持赋值时转换。不支持的场景下即使定义了转换路径,也不能做数据类型 转换。

MySQL数据库支持任意两种数据类型之间做转换。

由于存在以上差异,基于MySQL数据库的应用程序向GaussDB数据库迁移时,SQL语句可能由于不支持不同数据类型之间的转换而报错。或者支持转换的场景下,转换的规则有差异导致SQL语句执行的结果不同。

推荐的做法是: SQL语句中尽量使用相同的数据类型做比较或者赋值等操作,避免因为数据类型转换导致非预期结果或者性能损耗。

2. 选择目标数据类型的规则差异:

对于有些场景,比较的数据类型或者返回的数据类型需要综合考虑多个表达式的类型才能确定。比如UNION运算中,不同SELECT语句中相同位置的投影列具有不同的数据类型,查询结果的最终数据类型,需要由各个SELECT语句投影列的数据类型共同确定。

确定目标数据类型的规则,GaussDB数据库和MySQL数据库存在体系上的差异。

- GaussDB数据库规则:
 - 操作符的操作数类型不一致时,并不是将操作数的类型统一转换为目标类型 再计算。而是直接注册两个数据类型的操作符,操作符处理中定义两个不同 类型的处理规则。此方式不存在类型隐式转换,但自定义的处理规则隐含了 转换的操作。
 - 集合运算和表达式场景,确定目标数据类型的规则:
 - 如果所有类型都相同,则此类型即为目标类型。
 - 两个数据类型如果不同,检查数据类型是否属于同一种类的数据类型, 如数值类型、字符类型、日期时间类型等。不属于同一种类的数据类型,无法确定目标类型,此时SQL语句执行会报错。
 - 对于category属性(在pg_type系统表中定义)相同的数据类型,具有 preferred属性(在pg_type系统表中定义)的数据类型会被选为目标类型。或者操作数1能转换为操作数2(没有转换路径),而操作数2无法转换为操作数1或数值类型优先级小于操作数2,则选择操作数2作为目标类型。
 - 如果涉及到3个及以上的数据类型,确定目标类型的规则为: common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3),依次迭代处理,得 到最终的结果。
 - 对于IN和NOT IN表达式,如果根据以上规则无法确认目标类型,会将lexpr与expr_list中每一个表达式单独按照等值操作符(=)逐个比较。
 - 精度的确定:以最终选定的表达式的精度作为最终结果。
- MySQL数据库规则:
 - 操作符的操作数类型不一致时,先按照如下规则确定目标类型。确定后将类型不一致的操作数转换成目标类型后再做处理。
 - 两个参数都是string类型,则都按照string类型比较。
 - 两个参数都是integer类型,则都按照integer类型比较。
 - 十六进制数值如果不与数值比较,则当做二进制字符串比较。

- 一个参数是datetime/timestamp类型,另一个参数是常量,将常量转换为时间戳类型然后比较。
- 如果其中一个参数是decimal类型,比较时使用的数据类型取决于另外一个参数。另外一个是decimal或者integer类型时,按照decimal类型;另外一个是其他类型,按照real类型比较。
- 其他场景都转换为 real 类型后比较。
- 集合运算和表达式场景,确定目标数据类型的规则如下:
 - 建立任意两个类型之间的目标类型矩阵。给定两个类型,通过矩阵即可以确定目标类型。
 - 如果涉及到3个及以上的数据类型,确定目标类型的规则为: common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3),依次迭代处理,得 到最终的结果。
 - 如果目标类型是integer类型,且各个表达式类型包含有符号和无符号的混合场景,则会将类型提升到更高精度的integer类型。符号的确定:所有表达式都是无符号时,结果才为无符号,否则结果为有符号。
 - 精度确定:以表达式中的最大精度作为最终结果。

从以上规则可知: GaussDB和MySQL数据库在数据类型的转换规则上有很大差异,不能直接对比。在上述场景下,SQL语句的执行结果可能和MySQL数据库不一致。当前版本推荐各个表达式使用相同的类型,或提前使用cast转换成需要的类型来规避差异。

2.2.3 系统函数

GaussDB数据库兼容绝大多数MySQL的系统函数,但存在部分差异。如未列出,函数 行为默认为GaussDB原生行为。

2.2.3.1 流量控制函数

表 2-10 流量控制函数列表

序号	MySQL 数据库	GaussD B数据库	差异	
1	IF()	支持	expr1入参仅支持bool类型。非bool类型入参若不能转 换为bool类型则报错。	
			● 若expr2、expr3两入参类型不同且两类型间不存在隐式转换函数则报错。	
			● 两入参类型相同时,返回该入参类型。	
			● 若expr2、expr3两入参类型分别为NUMERIC、 STRING或TIME其中一个时,输出为TEXT类型, MySQL输出为VARCHAR类型。	

序号	MySQL 数据库	GaussD B数据库	差异
2	IFNUL L()	支持	 若expr1、expr2两入参类型不同且两类型间不存在隐式转换函数则报错。 两入参类型相同时,返回该入参类型。 若expr1、expr2两入参类型范畴分别为NUMEIRC、STRING或TIME其中一个时,输出为TEXT类型,MySQL输出为VARCHAR类型。 两入参类型其中一个为FLOAT4类型另一个为numeric范畴中任一类型,返回值为DOUBLE类型。MySQL其中一入参为FLOAT4,另一入参为TINYINT、UNSIGNED TINYINT、SMALLINT、UNSIGNED SMALLINT、MEDIUMINT、UNSIGNED MEDIUMINT、BOOL任一类型时,返回FLOAT4类型,第一个入参为FLOAT4,第二个入参为BIGINT或UNSIGNED BIGINT时,返回FLOAT类型。

序号	MySQL 数据库	GaussD B数据库	差异
3	NULLI	支持	● GaussDB中NULLIF()类型推导遵从以下逻辑:
	F()		 如果两个参数的数据类型不同,且两入参类型存在等值比较操作符,则返回对应等值操作符对应的左值类型,否则会对两入参类型进行强制类型兼容。
			 若强制类型兼容后,存在等值比较操作符,则返回 强制类型兼容后对应等值操作符的左值类型。
			3. 若强制类型兼容后,仍找不到对应等值操作符,则 报错。 两入参类型存在等值比较操作符 gaussdb=# select pg_typeof(nullif(1::int2, 2::int8)); pg_typeof
			smallint (1 row) 两入参类型不存在等值比较操作符,但在强制类型兼容后可以找 到等值比较操作符 gaussdb=# select pg_typeof(nullif(1::int1, 2::int2)); pg_typeof
			bigint (1 row)
			两入参类型不存在等值比较操作符,且强制类型兼容后也不存在 等值比较操作符 gaussdb=# SELECT nullif(1::bit, '1'::MONEY); ERROR: operator does not exist: bit = money LINE 1: SELECT nullif(1::bit, '1'::MONEY);
			HINT: No operator matches the given name and argument type(s). You might need to add explicit type casts. CONTEXT: referenced column: nullif
			● MySQL输出类型仅与第一个入参类型有关:
			1. 第一个入参为tinyint、smallint、mediumint、 int、bool时,输出为int类型。
			2. 第一个入参为bigint时,输出为bigint类型。
			3. 第一个入参为unsigned tinyint、unsigned smallint、unsigned mediumint、unsigned int、 bit时,输出为unsigned int类型。
			4. 第一个入参为unsigned bigint时,输出为unsigned bigint。
			5. 第一个入参为浮点型即float、double、real时,输 出为double类型。
			6. 第一个入参类型为decimal或numeric类型时,输出 为decimal类型 。
			7. 第一个入参类型为时间类型或字符串类型即date、 time、date、datetime、timestamp、char、 varchar以及tinytext、enum、set时,输出为 varchar类型。
			8. 第一个入参类型为text、mediumtext、longtext 时,输出为longtext类型。

序号	MySQL 数据库	GaussD B数据库	差异
			9. 第一个入参类型为tinyblob时,输出为varbinary类型。
			10.第一个入参类型为mediumblob或longblob时,输 出为longblob类型。
			11.第一个入参为blob时,输出为blob类型。
4	ISNUL L()	支持	GaussDB中返回值为BOOLEAN类型的t或f,MySQL中返回值为INT类型的1或0。

2.2.3.2 日期和时间函数

□ 说明

以下为GaussDB MySQL兼容性日期时间函数公共说明。

• 开发指南中函数入参为时间类型表达式的情况:

时间类型表达式主要包括text、datetime、date或time,但所有可以隐式转换为时间表达式的类型都可以作为入参,比如数字类型可以通过先隐式转化为text,再作为时间类型表达式生效。

但是生效的情况根据函数有所不同,比如:datediff由于只计算日期的差值,所以时间表达式会 当做日期来解析。timestampdiff由于计算时间差值时是根据unit来决定的,所以会根据unit选择 将时间表达式当做date或者time或者datetime来解析。

• 函数入参为无效日期的情况:

一般而言,日期时间函数支持date、datetime的范围和MySQL保持一致。date支持的范围为 '0000-01-01'到'9999-12-31',datetime支持的范围为'0000-01-01 00:00:00'到'9999-12-31 23:59:59'。虽然GaussDB支持的date、datetime范围大于MySQL,但是越界仍然算无效日期。

大部分时间函数会告警并返回NULL,只有能通过cast正常转换的日期,才是正常合理的日期。

• 函数入参的分隔符场景:

对于时间函数,处理入参时会将所有非数字字符视作分隔符,然后根据数字所处的位置进行 计算,推荐使用标准写法,年月日之间使用-分隔符,时分秒之间使用:分隔符,毫秒之前通 过.来进行分隔。

易错场景: "SELECT timestampdiff(hour, '2020-03-01 00:00:00', '2020-02-28 00:00:00+08');" B兼容模式数据库中时间函数不会自动计算时区,所以此处+08并未识别为时区,而是+作为分隔符当作秒来进行计算。

GaussDB的日期时间函数的大部分功能场景与MySQL一致,但仍有差异,一些差异如下:

函数入参为NULL时,函数返回NULL,无warning或error告警。这些函数包括:
from_days、date_format、str_to_date、datediff、timestampdiff、date_add、subtime、month、time_to_sec、to_days、to_seconds、dayname、monthname、convert_tz、sec_to_time、addtime、adddate、date_sub、timediff、last_day、weekday、from_unixtime、unix_timestamp、subdate、day、year、weekofyear、dayofmonth、dayofyear、week、yearweek、dayofweek、time_format、hour、minute、second、microsecond、quarter、get_format、extract、makedate、period_add、timestampadd、period_diff、utc_time、utc_timestamp、maketime、curtime

示例:

gaussdb=# select day(null);
day

---- (1 row)

纯数字入参个别函数与MySQL有差异,不带引号的数字入参统一转成text入参来 处理。

示例:

```
gaussdb=# select day(19231221.123141);
WARNING: Incorrect datetime value: "19231221.123141"
CONTEXT: referenced column: day
day
-----
(1 row)
```

时间日期运算函数: adddate、subdate、date_add、date_sub。当运算后的结果为日期时,支持的范围为[0000-01-01, 9999-12-31], 当运算后的结果为日期时间时,支持的范围为[0000-01-01 00:00:00:0000000, 9999-12-31 23:59:59.999999], 当运算后的结果超过支持的范围时,在严格模式下报ERROR,在宽松模式下报WARNING。另外,当运算后的日期结果在范围[0000-01-01, 0001-01-01]中时, GaussDB正常返回结果, MySQL返回'0000-00-00'。

示例:

```
gaussdb=# SELECT subdate('0000-01-01', interval 1 hour);
ERROR: Datetime function: datetime field overflow
CONTEXT: referenced column: subdate
gaussdb=# SELECT subdate('0001-01-01', interval 1 day);
subdate
------
0000-12-31
(1 row)
```

 对于日期和时间函数的date或datetime类型入参,含有0月或0日则为非法值,在 严格模式下报error;在宽松模式,当输入为字符串或数字时,报warning,输入 为date或datetime类型时视为上一年12月或上一月最后一日处理。

对于cast函数,转换为date、datetime时,严格模式下会报error;宽松模式下不会报warning,而是视为上一年12月或上一月最后一日处理,需要注意此区别。 MySQL对于包含0年、0月或0日的情况会原样输出。

示例:

若函数入参为numeric数据类型,在非法输入的情况下不会产生报错,会把入参当 做0值处理。

示例:

• 最多保留6位小数,不保留后置都为0的小数。

示例:

● 时间函数参数为字符串时,只保证年月日之间使用"-"分隔,时分秒之间使用 ":"分隔时结果正确。

示例:

```
gaussdb=# select adddate('20-12-12',interval 1 day);
    adddate
------
2020-12-13
(1 row)
```

● 在MySQL中,当函数的返回值为varchar时,在GaussDB中,函数对应的返回值为text。

```
-- GaussDB中函数的返回值。
gaussdb=# SELECT pg_typeof(adddate('2023-01-01', 1));
pg_typeof
-------
text
(1 row)
-- MySQL中函数的返回值。
mysql> CREATE VIEW v1 AS SELECT adddate('2023-01-01', 1);
Query OK, 0 rows affected (0.00 sec)

mysql> DESC v1;
+-------+
Field | Type | Null | Key | Default | Extra |
+------+
| adddate('2023-01-01', 1) | varchar(29) | YES | NULL | |
+------+
| row in set (0.00 sec)
```

表 2-11 日期与和时间函数列表

序号	MySQL数据库	GaussDB数 据库	差异
1	ADDDATE()	支持	此函数的表现会因为interval表达式 的差异与MySQL有差异,具体可见 INTERVAL差异说明。

序号	MySQL数据库	GaussDB数 据库	差异
2	ADDTIME()	支持	 MySQL对第二入参为DATETIME 样式字符串返回NULL, GaussDB 可以计算。
			入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。
			MySQL中ADDTIME函数如果第一个参数是动态参数(例如在预准备语句中),则返回类型为TIME。否则,函数的解析类型派生自第一个参数的解析类型。GaussDB中ADDTIME函数的返回值规则如下:
			- 第一个入参为date,第二个入 参为date,返回值为time。 - 第一个入参为date,第二个入
			参为text,返回值为text。
			- 第一个入参为date,第二个入 参为datetime,返回值为 time。
			- 第一个入参为date,第二个入 参为time,返回值为time。
			- 第一个入参为text,第二个入 参为date,返回值为text。
			- 第一个入参为text,第二个入 参为text,返回值为text。
			- 第一个入参为text,第二个入 参为datetime,返回值为
			text。 - 第一个入参为text,第二个入 参为time,返回值为text。
			- 第一个入参为datetime,第二 个入参为date,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为text,返回值为 text。
			- 第一个入参为datetime,第二 个入参为datetime,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为time,返回值为 datetime。
			- 第一个入参为time,第二个入 参为date,返回值为time。

序号	MySQL数据库	GaussDB数 据库	差异
			- 第一个入参为time,第二个入参为text,返回值为text。
			- 第一个入参为time,第二个入参为datetime,返回值为time。
			- 第一个入参为time,第二个入 参为time,返回值为time。
3	CONVERT_TZ()	支持	-
4	CURDATE()	支持	-
5	CURRENT_DATE(), CURRENT_DATE	支持	-
6	CURRENT_TIME(), CURRENT_TIME	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值) 末尾0都不显示;MySQL会显示。 GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,其他均报错;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
7	CURRENT_TIMESTAM P(), CURRENT_TIMESTAMP	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按 精度输出的时间值(小数点后的值) 末尾0都不显示;MySQL会显示。 GaussDB只支持输入[0,6]范围内的 整型值,作为返回时间的精度,超过 6的整型值,会告警并按照精度6输 出时间值;MySQL的精度值有效值 是[0,6],但是输入的整型值内部会 对256求余(例257,会返回精度1的 时间值)。

序号	MySQL数据库	GaussDB数 据库	差异
8	CURTIME()	支持	GaussDB此函数输入字符串或者非整型值,会被隐式转成整型,然后再校验精度,[0,6]范围之外的会报错,范围之内会正常输出时间值;MySQL直接报错。GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的;MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示;MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,其他均报错;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
9	YEARWEEK()	支持	-
10	DATE_ADD()	支持	此函数的表现会因为interval表达式 的差异与MySQL有差异,具体可见 INTERVAL差异说明。
11	DATE_FORMAT()	支持	-
12	DATE_SUB()	支持	此函数的表现会因为interval表达式 的差异与MySQL有差异,具体可见 INTERVAL差异说明。
13	DATEDIFF()	支持	-
14	DAY()	支持	-
15	DAYNAME()	支持	-
16	DAYOFMONTH()	支持	-
17	DAYOFWEEK()	支持	-
18	DAYOFYEAR()	支持	-
19	EXTRACT()	支持	-
20	FROM_DAYS()	支持	-
21	FROM_UNIXTIME()	支持	-
22	GET_FORMAT()	支持	-
23	HOUR()	支持	-
24	LAST_DAY	支持	-

序号	MySQL数据库	GaussDB数 据库	差异
25	LOCALTIME(), LOCALTIME	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值) 末尾0都不显示;MySQL会显示。 GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,其他整型值直接报错;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
26	LOCALTIMESTAMP, LOCALTIMESTAMP()	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值) 末尾0都不显示;MySQL会显示。 GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,超过6的整型值,会告警并按照精度6输出时间值;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
27	MAKEDATE()	支持	-
28	MAKETIME()	支持	与MySQL相比, 入参为NULL时, GaussDB不支持maketime函数自嵌 套,MySQL支持。
29	MICROSECOND()	支持	-
30	MINUTE()	支持	-
31	MONTH()	支持	-
32	MONTHNAME()	支持	-
33	NOW()	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值) 末尾0都不显示;MySQL会显示。 GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,超过6的整型值,会告警并按照精度6输出时间值;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。

序号	MySQL数据库	GaussDB数 据库	差异
34	PERIOD_ADD()	支持	当入参period或结果小于0时, GaussDB参考MySQL8.0.x版本的表 现,报错处理。MySQL5.7会发生整 数回绕,导致计算结果异常。
35	PERIOD_DIFF()	支持	当入参或结果小于0时,GaussDB参 考MySQL8.0.x版本的表现,报错处 理。MySQL5.7会发生整数回绕,导 致计算结果异常。
36	QUARTER()	支持	-
37	SEC_TO_TIME()	支持	-
38	SECOND()	支持	-
39	STR_TO_DATE()	支持	返回值与MySQL有差异,GaussDB 返回的是text,MySQL返回的是 datetime、date。
40	SUBDATE()	支持	此函数的表现会因为interval表达式 的差异与MySQL有差异,具体可见 INTERVAL差异说明。

序号	MySQL数据库	GaussDB数 据库	差异
41	SUBTIME()	支持	 MySQL对第二入参为DATETIME 样式字符串返回NULL, GaussDB 可以计算。
			入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。
			MySQL中SUBTIME函数如果第一个参数是动态参数(例如在预准备语句中),则返回类型为TIME。否则,函数的解析类型派生自第一个参数的解析类型。GaussDB中SUBTIME函数的返回值规则如下:
			第一个入参为date,第二个入参为date,返回值为time。
			- 第一个入参为date,第二个入 参为text,返回值为text。
			- 第一个入参为date,第二个入 参为datetime,返回值为 time。
			- 第一个入参为date,第二个入 参为time,返回值为time。
			- 第一个入参为text,第二个入 参为date,返回值为text。
			- 第一个入参为text,第二个入 参为text,返回值为text。
			- 第一个入参为text,第二个入 参为datetime,返回值为 text。
			- 第一个入参为text,第二个入 参为time,返回值为text。
			- 第一个入参为datetime,第二 个入参为date,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为text,返回值为 text。
			- 第一个入参为datetime,第二 个入参为datetime,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为time,返回值为 datetime。
			- 第一个入参为time,第二个入 参为date,返回值为time。

序号	MySQL数据库	GaussDB数 据库	差异
			 第一个入参为time,第二个入参为text,返回值为text。 第一个入参为time,第二个入参为datetime,返回值为time。 第一个入参为time,第二个入参为time,返回值为time。
42	SYSDATE()	支持	MySQL入参整型值会按照一字节最 大值255整数回绕,Gauss不回绕。
43	YEAR()	支持	-
44	TIME_FORMAT()	支持	-
45	TIME_TO_SEC()	支持	-
46	TIMEDIFF()	支持	-
47	WEEKOFYEAR()	支持	-
48	TIMESTAMPADD()	支持	-
49	TIMESTAMPDIFF()	支持	-
50	TO_DAYS()	支持	-
51	TO_SECONDS()	支持	-
52	UNIX_TIMESTAMP()	支持	返回值与MySQL有差异,GaussDB 返回的是numeric,MySQL返回的是 int。
53	UTC_DATE()	支持	MySQL支持无括号调用,GaussDB
54	UTC_TIME()	支持	不支持。MySQL入参整型值会按照 一字节最大值255整数回绕。
55	UTC_TIMESTAMP()	支持	MySQL入参只支持0-6整数, GaussDB支持可以隐式转换为0-6的 输入。
56	WEEK()	支持	-
57	WEEKDAY()	支持	-

2.2.3.3 字符串函数

表 2-12 字符串函数列表

序号	MySQL数据库	GaussDB数 据库	差异
1	BIN()	支持	函数入参1支持类型存在差异,GaussDB入 参1支持类型如下:
			● 整数类型: tinyint、smallint、 mediumint、int、bigint
			● 无符号整数类型: tinyint unsigned、 smallint unsigned、int unsigned、 bigint unsigned
			● 字符和文本类型: char, varchar、 tinytext,text,mediumtext, longtext,仅支持纯数字整数字符串, 且整数范围在bigint范围内。
			● 浮点类型: float、real、double
			● 定点类型: numeric、decimal、dec
			● 布尔类型: bool
2	CONCAT()	支持	无论参数的数据类型如何,concat返回值的数据类型始终为text;MySQL的concat在含有二进制类型参数时,返回值为二进制类型。
3	CONCAT_WS()	支持	无论参数的数据类型如何,concat_ws返回 值的数据类型始终为text;MySQL的 concat_ws在含有二进制类型参数时,返回 值为二进制类型,其他情况返回值为字符串 类型。

序号	MySQL数据库	GaussDB数 据库	差异
4	ELT()	支持	 函数入参1支持类型如下:整数类型: tinyint、smallint、mediumint、int、bigint 无符号整数类型: tinyint unsigned、smallint unsigned、int unsigned 字符和文本类型: char, varchar、tinytext, text, mediumtext, longtext, 仅支持纯数字整数字符串, 且整数范围在bigint范围内。浮点类型: float、real、double定点类型: numeric、decimal、dec布尔类型: bool 函数入参2支持类型存在差异, GaussDB入参2支持类型如下:整数类型: tinyint、smallint、mediumint、int、bigint无符号整数类型: tinyint unsigned、smallint unsigned、int unsigned、smallint unsigned、int unsigned、bigint unsigned 字符和文本类型: char, varchar、tinytext, text, mediumtext, longtext 浮点类型: float、real、double定点类型: numeric、decimal、dec布尔类型: bool大对象类型: tinyblob, blob, mediumblob, longblob日期类型: datetime, timestamp, date, time
5	FIELD	支持	函数入参在bigint最大值 ~ bigint unsigned 最大值范围内的数字,存在不兼容。 函数入参为浮点型float(m, d)、double(m, d)、real(m, d)时精度更高,存在不兼容。
6	FIND_IN_SET()	支持	当数据库encoding = 'SQL_ASCII'时,不支持默认的大小写判断规则,即在用户不指定字符集规则的情况下,大写与小写区分判断。

序号	MySQL数据库	GaussDB数 据库	差异
7	INSERT()	支持	 Int64类型传参有范围限制,一旦超出-9223372036854775808~9223372036854775807范围会直接报错,MySQL对数值类型传参范围无限制,异常会告警按照上限或下限数值处理。 字符串传参有限制,入参text类型字符串长度最大为2^30-5字节,入参bytea类型字符串长度最大为2^30-512字节。 s1和s2任意参数为bytea类型时,涉及到结果出现非法字符的情况可能展示结果与MySQL有差异但是字符编码与MySQL是一致的。
8	LOCATE()	支持	入参1为bytea类型,入参2为text类型时, GaussDB与MySQL行为存在差异。
9	MAKE_SET()	支持	 bits参数为整型时,最大范围支持到int128,低于MySQL范围。 bits参数为日期类型datetime、timestamp、date、time,由于时间类型转整型与MySQL存在差异,目前均未做支持。 bit类型或bool类型由于此类数据类型Gauss与MySQL存在差异,返回结果导致的差异为Gauss与MySQL固有差异。bits入参为bool类型,str入参为bit类型与bool类型均不做支持。 bits入参为字符串或文本类型时,仅支持纯整型数字形式,其他形式存在差异。且纯整型数字形式,其他形式存在差异。且纯整型数字范围限制在bigint范围。 str入参整型数值超过正负81个9,返回值与MySQL有差异。 str入参当以科学计数法表示时,GaussDB末尾0值会显示,MySQL不显示,以科学计数法打印,此为固有差异。

序号	MySQL数据库	GaussDB数 据库	差异
10	QUOTE()	支持	 已知str字符串中含有"\Z","\r","\%","_",GaussDB未进行转义,与MySQL存在差异。斜线后跟部分数字也会引起差异,如"\563"。由转义字符引起的本函数与MySQL的差异,此为Gauss与MySQL的转义字符差异。 str字符串中的"\b",输出结果表现形式与MySQL有差异。此为Gauss与MySQL的固有差异。 str字符串中含有"\0"时,GaussDB由于UTF-8字符集不识别该字符,输入不成功。此为GaussDB与MySQL的固有差异 str为bit或bool类型时,由于Gauss与MySQL此类型目前有差异,暂不支持此类型。 Gauss最大支持1GB数据传输,str入参长度最大支持536870908,函数返回结果字符串最大支持1GB。 str入参整型数值超过正负81个9,返回值与MySQL有差异。 str入参当以科学计数法表示时,GaussDB末尾0值会显示,MySQL不显示,以科学计数法打印,此为固有差异。
11	SPACE()	支持	 GaussDB入参1最大支持1073741818, 超出返回空字符串。MySQL的入参1默 认最大支持4194304,超出告警。 函数入参1支持类型存在差异,GaussDB 入参1支持类型如下: 整数类型: tinyint、smallint、mediumint、int、bigint 无符号整数类型: tinyint unsigned、smallint unsigned、int unsigned 字符和文本类型: char, varchar、tinytext,text,mediumtext,longtext,仅支持纯数字整数字符串,且整数范围在bigint范围内。 浮点类型: float、real、double 定点类型: numeric、decimal、dec 布尔类型: bool
12	SUBSTR()	支持	-

序号	MySQL数据库	GaussDB数 据库	差异
13	SUBSTRING()	支持	-
14	SUBSTRING_IN DEX()	支持	-
15	STRCMP()	支持	1. 支持入参类型与M存在差异,GaussDB 支持类型如下:
			● 字符类型: CHAR、VARCHAR、 NVARCHAR2、TEXT
			● 二进制类型: BYTEA
			 数值类型: TINYING [UNSIGNED]、 SMALLINT [UNSIGNED]、INTEGER [UNSIGNED]、BIGINT [UNSIGNED]、FLOAT4、FLOAT8、 NUMERIC
			● 日期时间类型: DATE、TIME WITHOUT TIME ZONE、 DATETIME、TIMESTAMPTZ
			2. 对于数值类型中的浮点类型,由于连接参数设置不同,精度可能与M有差异,不建议使用该场景,或使用NUMERIC类型代替。
16	SHA()/SHA1()	支持	-
17	SHA2()	支持	-

2.2.3.4 强制转换函数

表 2-13 强制转换函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	CAST()	支持	数据类型转换规则和支持的转换类型均以GaussDB支持的转换范围和规则为准。
2	CONVERT()	支持	数据类型转换规则和支持的转换类型均以 GaussDB支持的转换范围和规则为准。

2.2.3.5 加密函数

表 2-14 加密函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	AES_DECRYPT()	支持	-
2	AES_ENCRYPT()	支持	-

2.2.3.6 JSON 函数

□说明

json函数差异说明:

- 1. 对于json函数和其他字符入参函数来说,如果含有转义字符的输入,默认情况下会与MySQL有一定差异,需要设置GUC参数set standard_conforming_strings= off;的情况下,对于转义字符的场景才会与MySQL兼容,但是会产生非标准字符输入的warning告警,在转义字符中\t、\u与转义数字与MySQL有差异。JSON_UNQUOTE()函数该场景下已做兼容,不设置GUC参数时,仍与MySQL兼容,且不会报警。
- 2. 在处理超长数字(数字的字符长度超过64)时,GaussDB的json函数会将数字解析为一个double处理,并使用科学计数法计数。和MySQL的非json类型入参相同。但是在json类型入参时,由于json类型未完全与MySQL兼容,此场景下会产生差异。MySQL会完整显示数字(并且当数字长度超过82时,MySQL会给出错误的结果。),GaussDB依然将超长数字解析为一个double精度的值。考虑到超长数字内部都是使用浮点数进行储存,进行运算时无论GaussDB还是MySQL都会有精度丢失,建议您使用字符串来储存超长数字。gaussdb=# select json_insert('[1, 4,

json_insert ------[1, 4, 1e+74, [1, 4, 1e+74]] (1 row)

表 2-15 JSON 函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	JSON_APPEND()	支持	-
2	JSON_ARRAY()	支持	-
3	JSON_ARRAY_APPEND()	支持	-
4	JSON_ARRAY_INSERT()	支持	-
5	JSON_CONTAINS()	支持	-
6	JSON_CONTAINS_PATH()	支持	-

序号	MySQL数据库	GaussDB数据库	差异
7	JSON_DEPTH()	支持	返回值与 MySQL有差 异,GaussDB 返回的是int, MySQL返回的 是bigint。
8	JSON_EXTRACT()	支持	-
9	JSON_INSERT()	支持	-
10	JSON_KEYS()	支持	-
11	JSON_LENGTH()	支持	返回值与 MySQL有差 异,GaussDB 返回的是int, MySQL返回的 是bigint。
12	JSON_MERGE()	支持	-
13	JSON_OBJECT()	支持	-
14	JSON_QUOTE()	支持	返回值与 MySQL有差 异,GaussDB 返回的是 json,MySQL 返回的是 varchar或者 text。
15	JSON_REMOVE()	支持	-
16	JSON_REPLACE()	支持	-
17	JSON_SEARCH()	支持	返回值与 MySQL有差 异,GaussDB 返回的是 text,MySQL 返回的是 json。
18	JSON_SET()	支持	-
19	JSON_TYPE()	支持	数值类型的 json值统一识 别为 number,与 MySQL有差 异。

序号	MySQL数据库	GaussDB数据库	差异
20	JSON_UNQUOTE()	支持	-
21	JSON_VALID()	支持	-

2.2.3.7 聚合函数

表 2-16 聚合函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	GROUP_CONCAT()	支持	1. 当group_concat参数中同时有DISTINCT和ORDER BY语法时,所有ORDER BY后的表达式必须也在DISTINCT的表达式之中。
			2. group_concat(order by 数字)不代表按照第几个参数的顺序,数字只是一个常量表达式,相当于不排序。
			3. 无论参数的数据类型如何, group_concat返回值的数据类型始终为text;MySQL的 group_concat在含有二进制类型参数时,返回值为二进制类型,其他情况返回值为字符串类型,并且返回值长度大于512时,其数据类型为字符串大对象或二进制大对象。
			4. GUC参数 group_concat_max_len有效范 围是0-1073741823,最大值比 MySQL小。

序号	MySQL数据库	GaussDB数据库	差异
2	DEFAULT()	支持	1. 字段默认值为数组形式, GaussDB返回数组形式, MySQL不支持数组类型。
			2. GaussDB字段是隐藏列(比如 xmin、cmin),函数返回空 值。
			3. GaussDB支持分区表、临时 表、多表连接查询默认值。
			4. GaussDB支持查询列名包含字符串值节点(表示名称)和A_Star节点(表示出现"*"),如default(tt.t4.id)和default(tt.t4.*)。不合法的查询列名和A_Star节点,GaussDB和MySQL报错信息有差异。
			5. GaussDB创建字段默认值,没 有检验字段类型的范围,使用 default函数可能报错。
			6. 字段的默认值是函数表达式 时,GaussDB的default函数返 回建表时字段的default表达式 的计算值。MySQL的default函 数返回NULL。

2.2.3.8 数字操作函数

表 2-17 数字操作函数列表

序号	MySQ L数据 库	Gauss DB数据 库	差异
1	log2()	支持	 小数位显示与MySQL存在差异,受GaussDB浮点数据类型限制,可通过参数extra_float_digits控制小数位个数显示。 由于输入精度内部处理差异,GaussDB与MySQL会存在结果计算差异。 支持数据类型有: bigint、int16、int、smallint、tinyint整数类型。 bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned无符号整数类型。 numeric、real浮点数类型。 character、character varying、clob、text字符串类型,仅支持纯数字整数字符串。
			● set类型。
			● NULL空类型。
2	log10()	支持	1. 小数位显示与MySQL存在差异,受GaussDB浮点数据 类型限制,可通过参数extra_float_digits控制小数位 个数显示;
			2. 由于输入精度内部处理差异,GaussDB与MySQL会存在结果计算差异;
			3. 支持数据类型有:
			● bigint、int16、int、smallint、tinyint整数类型。
			 bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned无符号整数类型。
			● numeric、real浮点数类型。
			● character、character varying、clob、text字符串 类型,仅支持纯数字整数字符串。
			● set类型。
			● NULL空类型。

2.2.3.9 其他函数

表 2-18 其他函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	UUID()	支持	-
2	UUID_SHORT()	支持	-

2.2.4 操作符

GaussDB数据库兼容绝大多数MySQL的操作符,但存在部分差异。如未列出,操作符行为默认为GaussDB原生行为。

序号	MySQL数据库	GaussDB数据库	差异
1	安全等于(<=>)	支持	-

2.2.5 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集,支持的范围如下。

表 2-19 字符集列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4	支持
2	gbk	支持
3	gb18030	支持

山 说明

目前GaussDB对不属于当前字符集的非法字符未执行严格的编码逻辑校验,可能导致此类非法字符成功输入。而MySQL会校验报错。

2.2.6 排序规则

GaussDB数据库支持指定模式、表或列的排序规则,支持的范围如下。

山 说明

排序规则差异说明:

- 当前仅有字符串类型、部分二进制类型支持指定排序规则,其他类型不支持指定排序规则,可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。
 MySQL中所有类型可以指定字符序,但除字符串、二进制类型其他排序规则无实际意义。
- 当前排序规则仅支持在其对应字符集与库级字符集一致时可以指定。
- utf8mb4字符集下默认字符序为utf8mb4_general_ci,与MySQL5.7保持一致, utf8mb4_0900_ai_ci为utf8mb4的非默认字符序。
- GaussDB中utf8和utf8mb4为同一个字符集。

表 2-20 排序规则列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4_general_ci	支持
2	utf8mb4_unicode_ci	支持
3	utf8mb4_bin	支持
4	gbk_chinese_ci	支持
5	gbk_bin	支持
6	gb18030_chinese_ci	支持
7	gb18030_bin	支持
8	binary	支持

序号	MySQL数据库	GaussDB数据库
9	utf8mb4_0900_ai_ci	支持
10	utf8_general_ci	支持
11	utf8_bin	支持

2.2.7 SQL

GaussDB数据库兼容绝大多数MySQL语法,但存在部分差异。本章节介绍GaussDB数据库当前支持的MySQL语法。

2.2.7.1 DDL

表 2-21 DDL 语法兼容介绍

序号	概述	详细语法说明	差异
1	建表和修改表时支持 创建主键、UNIQUE 索引	ALTER TABLE CREATE TABLE	 GaussDB当前不支持UNIQUE INDEX KEY index_name语法,使用UNIQUE INDEX KEY index_name语法时会报错。MySQL支持。 当约束被建立为全局二级索引,SQL语句中指定using btree时,底层会建立为ubtree。 当约束关联的表为ustore,且SQL语句中指定为using btree时,底层会建立为ubtree。

序号	概述	详细语法说明	差异
2	支持前缀索引	CREATE INDEX	前缀长度不得超过2676,键值 的实际长度受内部页面限制, 若字段中含有多字节字符或者 一个索引上有多个键,索引行 长度可能会超限报错。
			 CREATE INDEX语法中,不支持以下关键字作为前缀键的字段名称: COALESCE、EXTRACT、GREATEST、LEAST、LNNVL、NULLIF、NVL、NVL2、OVERLAY、POSITION、REGEXP_LIKE、SUBSTRING、TIMESTAMPDIFF、TREAT、TRIM、XMLCONCAT、XMLELEMENT、XMLEXISTS、XMLFOREST、XMLPARSE、XMLPI、XMLROOT、XMLSERIALIZE。 主键和唯一键索引中不支持前缀键。
3	支持指定字符集与排 序规则	ALTER SCHEMA、 ALTER TABLE、 CREATE SCHEMA、 CREATE TABLE	-
4	创建分区表语法兼容	CREATE TABLE PARTITION	-
5	建表和修改表时支持 指定表级和列级 comment	CREATE TABLE、 ALTER TABLE	-
6	创建索引时支持指定 索引级comment	CREATE INDEX	-

序号	概述	详细语法说明	差异
7	交换普通表和分区表 分区的数据	ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点:
			 MySQL表或分区使用 tablespace时,则无法进行分 区和普通表数据的交换; GaussDB表或分区使用不同的 tablespace时,仍可进行分区 和普通表数据的交换。
			● 对于列默认值,MySQL不会校验默认值,因此默认值不同时也可进行分区和普通表数据的交换;GaussDB会校验默认值,如果默认值不同,则无法进行分区和普通表数据的交换。
			 MySQL在分区表或普通表上进行DROP列操作后,表结构仍然一致,则可进行分区和普通表数据的交换; GaussDB需要保证普通表和分区表的被删除列严格对齐才能进行分区和普通表数据的交换。
			 MySQL和GaussDB的哈希算法不同,所以两者在相同的hash分区存储的数据可能不一致,导致最后交换的数据也可能不一致。
			● MySQL的分区表不支持外键, 普通表包含外键或其他表引用 普通表的外键,则无法进行分 区和普通表数据的交换; GaussDB的分区表支持外键, 在两个表的外键约束一致时, 则可进行分区和普通表数据的 交换;GaussDB的分区表不带 外键,普通表有其他表引用, 如果分区表和普通表表一致, 则可进行分区和普通表数据的 交换。

序号	概述	详细语法说明	差异
8	支持自增列	ALTER TABLE、 CREATE TABLE	● 目前仅支持各个DN独立自增的 本地自动增长列。
			● 自动增长列建议为索引(非全局二级索引)的第一个字段,否则建表时产生警告,含有自动增长列的表进行某些操作时会产生错误,例如:ALTERT TABLE EXCHANGE PARTITION。MySQL自动增长列必须为索引第一个字段。
			● AUTO_INCREMENT = value语 法,value必须为小于2^127的 正数。MySQL不校验value。
			 当自增值已经达到字段数据类型的最大值时,继续自增将产生错误。MySQL有些场景产生错误或警告,有些场景仍自增为最大值。
			 不支持 innodb_autoinc_lock_mode系 统变量, GaussDB的GUC参数 auto_increment_cache=0时, 批量插入自动增长列的行为与 MySQL系统变量 innodb_autoinc_lock_mode=1 相似。
			● 自动增长列在导入数据或者进行Batch Insert执行计划的插入操作时,对于混合0、NULL和确定值的场景,如果产生错误,后续插入自增值不一定与MySQL完全一致。
			- 提供auto_increment_cache 参数,可以控制预留自增值 的数量。
			• 批量插入在不同执行计划下,自增的顺序、自增值预留数量可能与MySQL不完全相同。例如:"INSERT INTO table VALUES(),(),"由于要分布到不同DN,在某些执行计划中,DN获取不到即将插入的行数。
			- 提供auto_increment_cache 参数,可以控制预留自增值 的数量。

序号	概述	详细语法说明	差异
			 并行导入或插入自动增长列触发自增时,每个并行线程预留的缓存值也只在其线程中使用,未完全使用完毕的话,也会出现表中自动增长列的值不连续的情况。并行插入产生的自增值结果无法保证与MySQL完全一致。
			● SERIAL数据类型为原有的自增列,与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。
			 不允许auto_increment_offset 的值大于 auto_increment_increment的 值,会产生错误。MySQL允 许,并说明 auto_increment_offset会被忽 略。
			● 在表有主键或索引的情况下, ALTER TABLE命令重写表数据 的顺序与MySQL不一定相同, GaussDB按表数据存储顺序重 写,MySQL会按主键或索引顺 序重写,导致自增值的顺序可 能不同。
			 ALTER TABLE命令添加或修改 自增列时,第一次预留自增值 的数量是表统计信息中的行 数,统计信息的行数不一定与 MySQL一致。
			● 在触发器或用户自定义函数中 自增时,刷新last_insert_id返 回值。MySQL不刷新。
			 对GUC参数 auto_increment_offset和 auto_increment_increment设 置超出范围的值会产生错误。 MySQL会自动改为边界值。
			暂不支持last_insert_id函数。本地临时表暂不支持自动增长列。
			sql_mode设置no_auto_value_on_zero参数, 表定义的自动增长列为非NOT

序号	概述	详细语法说明	差异
			NULL约束,向表中插入数据不 指定自动增长列的值时, GaussDB中自动增长列插入 NULL值,且不触发自增; MySQL中自动增长列插入 NULL值,触发自增。
9	支持删除表的主键约 束	ALTER TABLE	-

序号	概述	详细语法说明	差异
1 0	支持CREATE TABLE LIKE语法兼 容	CREATE TABLE LIKE	• 在MySQL 8.0.16 之前的版本中,CHECK约束会被语法解析但功能会被忽略,表现为不复制CHECK约束,GaussDB支持复制CHECK约束。
			对于set数据类型,在建表时, MySQL支持复制,GaussDB不 支持复制。
			 对于主键约束名称,在建表 时,MySQL所有主键约束名称 固定为PRIMARY KEY, GaussDB不支持复制。
			对于唯一键约束名称,在建表 时,MySQL支持复制, GaussDB不支持复制。
			 对于CHECK约束名称,在建表 时,MySQL 8.0.16 之前的版本 无CHECK约束信息,GaussDB 支持复制。
			对于索引名称,在建表时, MySQL支持复制,GaussDB不 支持复制。
			● 在跨sql_mode模式建表时, MySQL受宽松模式和严格模式 控制,GaussDB可能存在严格 模式失效的情况。 例如:源表存在默认值 "0000-00-00",在 "no_zero_date"严格模式 下,GaussDB建表成功,且包 含默认值"0000-00-00",严 格模式失效;而MySQL建表失 败,受严格模式控制。
			针对跨数据库创建表,MySQL 支持,GaussDB不支持。
			● 针对源表为临时表,创建非临时表,MySQL支持,GaussDB不支持。

序号	概述	详细语法说明	差异
1	支持更改表名兼容语法	ALTER TABLE[IF EXISTS] tbl_name RENAME [TO AS =] new_tbl_name; RENAME {TABLE TABLES} tbl_name TO new_tbl_name [, tbl_name2 TO new_tbl_name2,];	 GaussDB的alter rename语法 仅支持修改表名称功能操作, 不能耦合其它功能操作; GaussDB的仅旧表名字段支持 如schema.table_name用法; 且新表名与旧表名将属于同一 Schema下; GaussDB的不支持新旧表跨 Schema重命名操作;但如有权限,则可在当前Schema下修改其它Schema下表名称; GaussDB的rename多组表的语法支持全为本地临时表的重命名,不支持本地临时表和非本地临时表组合的场景。

序号	概述	详细语法说明	差异
1 2	支持増加子分区语法兼容	ALTER TABLE [IF EXISTS] { table_name [*] ONLY table_name ONLY (table_name)} action [,]; action: move_clause exchange_clause row_clause modify_clause modify_clause add_clause drop_clause ilm_clause add_clause: ADD {{partition_less_th an_item partition_start_en d_item partition_list_item } PARTITION({partition_less_than_ite m partition_start_en d_item partition_start_en d_item partition_start_en d_item partition_start_en d_item partition_list_item } })}	 不支持ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition1,);语法添加多分区。 仅支持原有添加多分区语法: ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2[y1]),;。

2.2.7.2 DML

表 2-22 DML 语法兼容介绍

序号	概述	详细 语法 说明	差异
1	DELETE支持 ORDER BY和 LIMIT	DELE TE	-
2	UPDATE支持 ORDER BY和 LIMIT	UPDA TE	-

序号	概述	详细 语法 说明	差异
3	REPLACE INTO 语法兼容	REPL	● 时间类型初始值的差异。例如: - MySQL不受严格模式和宽松模式的影响,可向表中插入时间0值,即: mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, f2, f3); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test;

序号	概述	详细 语法 说明	差异
			- GaussDB BIT类型的初始值为NULL,则报错。 gaussdb=# CREATE TABLE test(f1 int, f2 BIT(3) NOT NULL) DISTRIBUTE BY HASH(f1); CREATE TABLE gaussdb=# REPLACE INTO test VALUES(1, f2); ERROR: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (1, null).
4	SELECT支持指 定多分区查询	SELEC T	-
5	UPDATE支持指 定多分区更新	UPDA TE	-

序号	概述	详细 语法 说明	差异
6	LOAD DATA导 入数据功能	LOAD DATA	LOAD DATA语法执行结果与MySQL严格模式 一致,宽松模式暂未适配。
			IGNORE与LOCAL参数功能仅为当导入数据与表中数据存在冲突时,忽略当前冲突行数据功能和当文件中字段数小于指定表中列数时自动为其余列填充默认值功能,其余功能暂未适配。
			指定LOCAL关键字,且文件路径为相对路径 时,文件从二进制目录下搜索;不指定LOCAL 关键字,且文件路径为相对路径时,文件从数 据目录下搜索。
			语法中指定分隔符,转义字符,分行符等符号 时,若指定为单引号,将导致词法解析错误。
			● [(col_name_or_user_var [, col_name_or_user_var])]指定列参数不支持重复指定列。
			● [FIELDS TERMINATED BY 'string']指定换行符不能与[LINES TERMINATED BY 'string']分隔符相同。
			执行LOAD DATA语法写入表中的数据若无法 转换为表中数据类型格式时报错。
			指定列时,仅支持指定为列名,不支持指定为用户变量。
			LOAD DATA SET表达式中不支持指定列名计算。
			若set表达式返回值类型与对应列类型之间不存在隐式转换函数则报错。
			● LOAD DATA不支持INSERT、DELETE触发器。
			● LOAD DATA只能用于表,不能用于视图。
			 Windows下的文件与Linux环境下文件默认换 行符存在差异,LOAD DATA无法识别此场景 会报错,建议用户导入时检查导入文件行结尾 的换行符。

序号	概述	详细 语法 说明	差异
7	INSERT IGNORE兼容	INSER T IGNO RE	● GaussDB会返回降级后的错误信息,MySQL则会将降级后的错误信息记录到错误堆栈中,然后调用show warnings;命令查看。例如: ● 时间类型的差异。例如: - GaussDB中date、datetime、timestamp默认零值。gaussdb=# CREATE TABLE test(f1 DATE NOT NULL, f2 DATETIME NOT NULL, f3 TIMESTAMP NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL, NULL, NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). INSERT 0 1 gaussdb=# SELECT* FROM test; f1

序号	概述	详细 语法 说明	差异
			长度与定义不同的场景下不支持INSERT IGNORE错误降级。 - GaussDB中bit类型 gaussdb=# CREATE TABLE test(f1 BIT(10) NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL); ERROR: Un-support feature DETAIL: ignore null for insert statement is not supported in column f1. gaussdb=# INSERT IGNORE INTO test VALUES(1010'); ERROR: bit string length 4 does not match type bit(10) CONTEXT: referenced column: f1 - MySQL中bit类型 mysql> CREATE TABLE test(f1 BIT(10) NOT NULL); Query OK, 0 rows affected (0.00 sec) mysql> INSERT IGNORE INTO test VALUES(NULL); Query OK, 1 row affected, 1 warning (0.00 sec) mysql> INSERT IGNORE INTO test VALUES(1010'); Query OK, 1 row affected, 1 warning (0.01 sec) • MySQL数据库时间类型指定精度时,插入时间零值会显示精度,GaussDB则不显示,例如: - GaussDB指定时间精度 gaussdb=# CREATE TABLE test(f1 TIME(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL, NULL, NULL, NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2 f3

序号	概述	详细 语法 说明	差异
			VALUES(NULL,NULL,NULL); Query OK, 1 row affected, 3 warnings (0.00 sec) mysql> SELECT * FROM test; +
			5.12.11.2.11

序号	概述	详细 语法 说明	差异
			gaussdb=# CREATE TABLE test2(f1 INT); CREATE TABLE gaussdb=# CREATE OR REPLACE FUNCTION trig_test() RETURNS TRIGGER AS \$\$ gaussdb\$# BEGIN gaussdb\$# BEGIN yaussdb\$# RETURN NEW; gaussdb\$# END; gaussdb\$# \$\$ LANGUAGE plpgsql; CREATE FUNCTION gaussdb=# CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW EXECUTE PROCEDURE trig_test(); CREATE TRIGGER gaussdb=# INSERT INTO test2 VALUES(NULL); WARNING: null value in column "f1" violates not- null constraint DETAIL: Failing row contains (null). CONTEXT: SQL statement "INSERT IGNORE INTO test1 VALUES(NULL)" PL/pgSQL function trig_test() line 3 at SQL statement INSERT 0 1 gaussdb=# SELECT * FROM test1; f1 0 (1 rows) MySQL触发器中使用INSERT IGNORE
			mysql> CREATE TABLE test1(f1 INT NOT NULL); Query OK, 0 rows affected (0.01 sec) mysql> CREATE TABLE test2(f1 INT); Query OK, 0 rows affected (0.00 sec) mysql> DELIMITER mysql> CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW -> BEGIN -> INSERT IGNORE into test1 values(NULL); -> END Query OK, 0 rows affected (0.01 sec) mysql> DELIMITER; mysql> INSERT INTO test2 VALUES(NULL); ERROR 1048 (23000): Column 'f1' cannot be null mysql> INSERT IGNORE INTO test2 VALUES(NULL); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test1; ++ f1 ++ 0 ++ 1 row in set (0.00 sec)

序号	概述	详细 语法 说明	差异
			mysql> SELECT * FROM test2; ++ f1 ++ NULL ++ row in set (0.00 sec) GaussDB的bool、serial的实现机制与MySQL 不同,因此其默认零值与MySQL不同,例 如: - GaussDB的行为 gaussdb=# CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL); NOTICE: CREATE TABLE will create implicit sequence "test_f1_seq" for serial column "test.f1" CREATE TABLE gaussdb=# INSERT IGNORE INTO test values(NULL,NULL); WARNING: null value in column "f1" violates not- null constraint DETAIL: Failing row contains (null, null). WARNING: null value in column "f2" violates not- null constraint DETAIL: Failing row contains (null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2+ 0 f (1 row) - MySQL的行为 mysql> CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL); Query OK, 0 rows affected (0.00 sec) mysql> INSERT IGNORE INTO test values(NULL,NULL); Query OK, 1 row affected, 1 warning (0.00 sec) mysql> SELECT * FROM test; +++ f1 f2 +++ 1 0 +++ 1 row in set (0.00 sec)

2.2.7.3 DCL

表 2-23 DCL 语法兼容介绍

序号	概述	详细语法说明	差异
1	SET NAMES指定 COLLATE字句	SET [SESSION LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] DEFAULT};	GaussDB中暂不支持指定 charset_name与数据库字 符集不同。具体请参考 《开发指南》中"SQL参 考 > SQL语法 > S > SET"章节。

2.2.8 驱动

2.2.8.1 JDBC

2.2.8.1.1 JDBC 接口参考

获取结果集中的数据

ResultSet对象提供了丰富的方法,以获取结果集中的数据。获取数据常用的方法如表 2-24所示,其他方法请参考JDK官方文档。

表 2-24 ResultSet 对象的常用方法

方法	描述	差异
int getInt(int columnIndex)	按列标获取 int型数据。	-
int getInt(String columnLabel)	按列名获取 int型数据。	-
String getString(int columnIndex)	按列标获取 String型数 据。	字段类型为整型且带有ZEROFILL属性时, GaussDB按照ZEROFILL属性要求的宽度信息用0 进行补位后输出结果,MySQL直接输出结果。
String getString(String columnLabel)	按列名获取 String型数 据。	字段类型为整型且带有ZEROFILL属性时, GaussDB按照ZEROFILL属性要求的宽度信息用0 进行补位后输出结果,MySQL直接输出结果。
Date getDate(int columnIndex)	按列标获取 Date型数据	-
Date getDate(String columnLabel)	按列名获取 Date型数 据。	-

2.3 MySQL 兼容性 M-Compatibility 模式

2.3.1 MySQL 兼容性 M-Compatibility 模式概述

本章节主要介绍GaussDB数据库的MySQL兼容性M-Compatibility模式(即 sql_compatibility='M')与MySQL 5.7数据库的兼容性对比信息。仅介绍505.2版本后新增的兼容性特性,特性的相关规格和约束建议在开发指南中查看。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

GaussDB的执行计划和优化、explain显示结果与MySQL不同。

由于GaussDB数据库与MySQL数据库底层框架实现存在差异,GaussDB数据库与MySQL数据库仍存在部分差异。

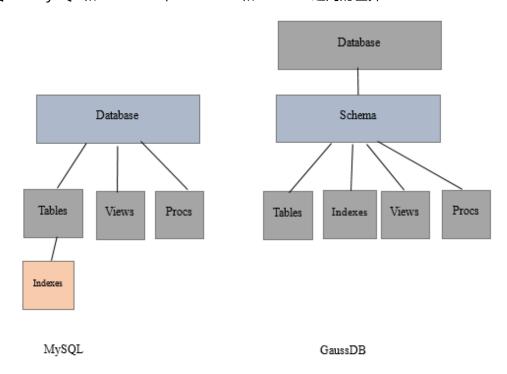
山 说明

由于GaussDB的底层架构与MySQL存在差异,对于information_schema和m_schema下与 MySQL名称相同的Schema(具体参见M-Compatibility开发指南中的"Schema"章节),其查 询性能可能存在差异。例如count函数无法做执行上的优化,表现为SELECT *和SELECT count(*) 语句耗时近似。

Database 和 Schema 设计

MySQL的数据对象包括DATABASE、TABLE、INDEX、VIEW、TRIGGER、PROC等,MySQL的对象层次跟GaussDB的对应关系是从上至下且一对多包含关系。如下图所示:

图 2-1 MySQL 和 GaussDB 中 Database 和 Schema 之间的差异



- 在MySQL中Database和Schema是同义词;而在GaussDB中,一个Database 下可以有多个Schema。在该特性中,每个MySQL中的Database都被映射到GaussDB的一个Schema。
- 在MySQL中,INDEX从属于一个TABLE,但在GaussDB中,INDEX从属于一个Schema。这个差异导致INDEX名在GaussDB中要求在Schema内唯一,但在MySQL中仅要在在一个表内唯一。这个差异将作为当前约束予以保留。

2.3.2 数据类型

GaussDB的数据类型的大部分功能场景与MySQL一致,但存在部分差异。

● 除特别说明,部分数据类型精度、标度、位数大小等不支持用浮点型数值定义, 建议使用合法的整型数值定义。

2.3.2.1 数值数据类型

表 2-25 整数类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BOOL	支持,存在 差异	输出格式:GaussDB中SELECT TRUE/FALSE 输出结果为t/f,MySQL为1/0。
2	BOOLEAN	支持,存在 差异	MySQL: BOOL/BOOLEAN类型实际映射为TINYINT类型。
3	TINYINT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
4	SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
5	MEDIUMIN T[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	MySQL存储MEDIUMINT数据需要3字节。

序号	MySQL数据 库	GaussDB数 据库	差异
6	INT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
7	INTEGER[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
8	BIGINT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。

□ 说明

输入格式:

MySQL:

整型类型对于类似"1.2.3.4.5"有多个小数点的字符串形式输入,在宽松模式下MySQL会发生错误解析,抛出WARNING并插表成功,例如将"1.2.3.4.5"插入表后值为12。

GaussDB:

整型类型对于类似"1.2.3.4.5"有多个小数点的的字符串形式输入,在宽松模式下,会将第二个小数点后的字符当作非法字符全部截断,抛出WARNING并插表成功,例如将"1.2.3.4.5"插入表后值为1,"1.6.3.4.5"插入表后值为2。

表 2-26 任意精度类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DECIMAL[(M[,D])] [ZEROFILL]	支持,存在 差异	MySQL decimal用一个9*9的数组存储数值,整数部分和小数部分分开存储,超过该长度时优先截小数部分。GaussDB只会在整数位数超
2	NUMERIC[(M[,D])] [ZEROFILL]	支持,存在 差异	过81位时截断。
3	DEC[(M[,D])] [ZEROFILL]	支持,存在 差异	
4	FIXED[(M[,D])] [ZEROFILL]	支持,存在 差异	

表 2-27 浮点类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	FLOAT[(M,D)] [ZEROFILL]	支持,存在 差异	FLOAT数据类型不支持KEY键值分区策略分区 表。
2	FLOAT(p) [ZEROFILL]	支持,存在 差异	FLOAT数据类型不支持KEY键值分区策略分区 表。
3	DOUBLE[(M ,D)] [ZEROFILL]	支持,存在 差异	DOUBLE数据类型不支持KEY键值分区策略分区表。
4	DOUBLE PRECISION[(M,D)] [ZEROFILL]	支持,存在 差异	DOUBLE PRECISION数据类型不支持KEY键值分区策略分区表。
5	REAL[(M,D)] [ZEROFILL]	支持,存在 差异	REAL数据类型不支持KEY值分区策略分区表。

2.3.2.2 日期与时间数据类型

表 2-28 日期与时间数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DATE	支持,存在 差异。	GaussDB支持date数据类型,与MySQL相比 规格上存在如下差异:
			● 输入格式:
			- 分隔符:反斜杠\在MySQL和GaussDB 中都视为转义,但MySQL支持\0, GaussDB暂不支持,因此反斜杠作为分 隔符且分隔符后为字符0时GaussDB会 报错。
2	DATETIME[(f sp)]	支持,存在 差异。	GaussDB支持datetime数据类型,与MySQL 相比规格上存在如下差异:
			輸入格式:
			- 分隔符:反斜杠\在MySQL和GaussDB中都视为转义,但MySQL支持\0,GaussDB暂不支持,因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。

序号	MySQL数据 库	GaussDB数 据库	差异
3	TIMESTAMP[(fsp)]	支持,存在 差异。	GaussDB支持timestamp数据类型,与 MySQL相比规格上存在如下差异: • 输入格式:
			- 分隔符:反斜杠\在MySQL和GaussDB 中都视为转义,但MySQL支持\0, GaussDB暂不支持,因此反斜杠作为分 隔符且分隔符后为字符0时GaussDB会 报错。
			 默认值: MySQL 5.7中timestamp列默认有 default value,为数据插入时的实时时间。GaussDB与MySQL 8.0一致,均不设置默认值,即插入null时,值为null。
4	TIME[(fsp)]	支持,存在 差异。	GaussDB支持time数据类型,与MySQL相比规格上存在如下差异: • 输入格式: - 分隔符:反斜杠\在MySQL和GaussDB中都视为转义,但MySQL支持\0,GaussDB暂不支持,因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 • 当时间类型的时、分、秒、纳秒为0时,GaussDB和MySQL可能存在符号位不同的情况。
5	YEAR[(4)]	支持。	-

山 说明

• GaussDB不支持ODBC语法的字面量:

{ d 'str' }

{ t 'str' }

{ ts 'str' }

● GaussDB支持标准SQL字面量,且类型关键字后面可选择添加精度,MySQL不支持:

DATE[(n)] 'str'

TIME[(n)] 'str'

TIMESTAMP[(n)] 'str'

 当给DATETIME、TIME、TIMESTAMP数据类型指定的精度超过其支持的最大精度时, GaussDB会将精度截断成支持的最大精度,MySQL则会报错。

2.3.2.3 字符串数据类型

表 2-29 字符串数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	CHAR(M)	支持,存在 差异	● 输入格式: - 输入二进制或十六进制字符串, GaussDB输出为十六进制,MySQL中 根据ASCII码表转义,无法转义的输出 为空。
2	VARCHAR(M)	支持,存在差异	 输入格式: GaussDB的自定义函数参数和返回值不支持长度校验,存储过程参数不支持长度校验,MySQL支持。 GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警,MySQL不支持。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。
3	TINYTEXT	支持,存在差异	 输入格式: 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中TINYTEXT类型不支持主键,GaussDB支持。 索引: MySQL中TINYTEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。

序号	MySQL数据 库	GaussDB数 据库	差异
4	TEXT	支持,存在 差异	 输入格式: 二 默认值: 创建表列时语法上允许设置默认值。 二 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中TEXT类型不支持主键,GaussDB支持。 索引: MySQL中TEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。
			 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列, GaussDB支持。
5	MEDIUMTEX T	支持,存在差异	 输入格式: 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中MEDIUMTEXT类型不支持主键,GaussDB支持。 索引: MySQL中MEDIUMTEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGTEXT	支持,存在 差异	 输入格式: GaussDB只支持不超过1G字节长度,MySQL支持4G-1字节长度。 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中LONGTEXT类型不支持主键,GaussDB支持。 索引: MySQL中LONGTEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。

2.3.2.4 二进制数据类型

表 2-30 二进制数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BINARY[(M)]	支持,存在差异	● 输入格式: - 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 - 插入字符串长度小于目标长度时,GaussDB填充符是0x20,MySQL是0x00。 - 字符集:默认字符集为数据库初始化字符集,MySQL默认类型字符集为BINARY字符集。 - 输出格式: - JDBC协议输出时BINARY类型的末尾空格显示为空格,MySQL末尾空格显示为公0。 - 宽松模式下,BINARY类型面对输入超过n的字节数的字符输入(例如中文字符),会将超限的整个字符截断。MySQL中会将超限的整个字符的前几位满足n范围内的字节信息保留,但输出时字符信息显示乱码。 - 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。 说明 GaussDB中,由于BINARY类型填充符和\0截断与MySQL的差异,在操作符比较计算,字符串相关系统函数计算,索引匹配,数据导入导出等场景下与MySQL的表现会存在差异。差异场景请见本节示例。

序号	MySQL数据 库	GaussDB数 据库	差异
2	VARBINARY(M)	支持,存在差异	 輸入格式: 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 字符集:默认字符集为数据库初始化字符集,MySQL默认类型字符集为BINARY字符集。 输出格式: JDBC协议输出时BINARY类型的末尾空格显示为空格,MySQL末尾空格显示为\x00。 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。回。
3	TINYBLOB	支持,存在差异	 输入格式: 二 默认值: 创建表列时语法上允许设置默认值。 一 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 ● 主键: MySQL中TINYBLOB类型不支持主键,GaussDB支持。 ● 索引: MySQL中TINYBLOB类型不支持除前缀索引外其他索引方法,GaussDB支持。 ● 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。 ● 输出格式: 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。

序号	MySQL数据 库	GaussDB数 据库	差异
4	BLOB	支持,存在差异	 輸入格式: 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中BLOB类型不支持主键,GaussDB支持。 索引: MySQL中BLOB类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。 输出格式:在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。
5	MEDIUMBLO B	支持,存在差异	 输入格式: 器认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键:MySQL中MEDIUMBLOB类型不支持主键,GaussDB支持。 索引:MySQL中MEDIUMBLOB类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键:MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。 输出格式:在MySQL 8.0以上版本,默认以公开头形式返回,GaussDB以多个\x形式返回。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGBLOB	支持,存在差异	● 取值范围:只支持不超过1G字节长度, MySQL支持4G-1字节长度。
			輸入格式:
			- 默认值:创建表列时语法上允许设置默 认值,MySQL不允许设置默认值。
			- 输入二进制或十六进制字符串, GaussDB输出为十六进制,MySQL中 根据ASCII码表转义,无法转义的输出 为空。
			● 主键:MySQL中LONGBLOB类型不支持 主键,GaussDB支持。
			索引: MySQL中LONGBLOB类型不支持 除前缀索引外其他索引方法, GaussDB支 持。
			外键: MySQL中TINYTEXT类型不支持作 为外键的参考列/被参考列,GaussDB支 持。
			• 输出格式:在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。
7	BIT[(M)] 支持,存在 差异	输出格式:	
		差异	所有输出按照二进制字符串形式输出。 MySQL中根据ASCII码表转义,无法转义 的输出为空。
			● 在MySQL 8.0以上版本,默认会开头补 0,GaussDB不会补0。

示例:

```
-- GaussDB场景
m_db=# CREATE TABLE test(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test VALUES(0x8000);
INSERT 0 1
m_db=# SELECT hex(a) FROM test;
    hex
80202020202020202020
a
(0 rows)
m_db=# CREATE TABLE test2(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test2 VALUES(0x80008000);
INSERT 0 1
m_db=# SELECT hex(a) FROM test2;
```

```
hex
802020202020202020
(1 row)
m_db=# DROP TABLE test;
DROP TABLE
m_db=# DROP TABLE test2;
DROP TABLE
-- Mysql场景
mysql> CREATE TABLE test(a BINARY(10));
Query OK, 0 rows affected (0.01 sec)
mysql> INSERT INTO test VALUES(0x8000);
Query OK, 1 row affected (0.00 sec)
mysql> SELECT hex(a) FROM test;
| hex(a)
              1 row in set (0.00 sec)
l a
      +----+
1 row in set (0.00 sec)
mysql> CREATE TABLE test2(a binary(10));
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO test2 VALUES(0x80008000);
Query OK, 1 row affected (0.00 sec)
mysql> SELECT hex(a) FROM test2;
| hex(a)
            | 80008000000000000000000
1 row in set (0.00 sec)
mysql> DROP TABLE test;
Query OK, 0 rows affected (0.00 sec)
mysql> DROP TABLE test2;
Query OK, 0 rows affected (0.00 sec)
```

2.3.2.5 数据类型支持的属性

表 2-31 数据类型支持的属性

序号	MySQL数据库	GaussDB数据库
1	NULL	支持
2	NOT NULL	支持
3	DEFAULT	支持
4	ON UPDATE	支持
5	PRIMARY KEY	支持

序号	MySQL数据库	GaussDB数据库
6	AUTO_INCREMENT	支持
7	CHARACTER SET name	支持
8	COLLATE name	支持
9	ZEROFILL	支持

使用CREATE TABLE AS方式建表,对VARBINARY类型的字段设置默认值,在使用SHOW CREATE TABLE、DESC或\d 查询的时候回显与MySQL存在差异,GaussDB显示为转换成十六进制后的值,而MySQL显示为原值。

```
示例:
```

```
m_db=# CREATE TABLE test_int(
    int_col INT
m_db=# CREATE TABLE test_varbinary(
    varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
m_db=# SHOW CREATE TABLE test_varbinary;
  Table |
                             Create Table
test_varbinary | SET search_path = public;
          | CREATE TABLE test_varbinary (
             varbinary_col varbinary(20) DEFAULT X'6761757373',
             int_col integer
          CHARACTER SET = "UTF8" COLLATE = "utf8mb4_general_ci"
          | WITH (orientation=row, compression=no, storage_type=USTORE, segment=off);
m_db=# DROP TABLE test_int, test_varbinary;
mysql> CREATE TABLE test_int(
    int_col INT
mysql> CREATE TABLE test_varbinary(
    varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
mysql> SHOW CREATE TABLE test_varbinary;
| Table | Create
Table
| test_varbinary | CREATE TABLE `test_varbinary` (
 `varbinary_col` varbinary(20) DEFAULT 'gauss',
 `int_vol` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |
1 row in set (0.00 sec)
mysql> DROP TABLE test_int, test_varbinary;
```

2.3.2.6 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换:

操作符(比较操作符、运算操作符等)的操作数的数据类型不一致。常见于查询 条件或者关联条件中的比较运算。

- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新(包括INSERT、UPDATE、MERGE、REPLACE等)的目标列,数据的类型和列的定义类型不一致。
- 显式的类型转换: CAST(expr AS datatype),将expr表达式类型转换为datatype 类型。
- 集合运算(UNION、MINUS、EXCEPT、INTERSECT)确定最终投影列的目标数据类型后,各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景,根据不同表达式的数据类型,来决定用于比较或者最终结果的目标数据类型。

数据类型转换差异点主要分为三类: 隐式转换,显式转换和UNION/CASE。

双冒号转换差异点:

● GaussDB中使用双冒号将函数入参转换为期望类型可能导致结果超出预期,且 MySQL中无双冒号功能。

示例:

隐式类型转换差异点:

- GaussDB中统一平铺成小类型到小类型的转换规则,MySQL中使用小类型转大类型,大类型转小类型的转换规则。
- GaussDB中隐式转换因数据类型本身差异点,输出格式存在部分行为不一致。示例:

GaussDB中的隐式转换,BIT数据类型到字符数据类型和二进制数据类型转换,输出存在部分行为不一致。GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。

示例:

```
m_db=# CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS_COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS_COL6 BIT(4),
  VS_COL7 BIT(4),
  VS COL8 BIT(4)
) DISTRIBUTE BY REPLICATION;
m_db=# CREATE TABLE string_storage (
  VS_COL1 BLOB,
  VS_COL2 TINYBLOB,
  VS_COL3 MEDIUMBLOB,
  VS_COL4 LONGBLOB,
  VS COL5 TEXT,
  VS_COL6 TINYTEXT,
  VS_COL7 MEDIUMTEXT,
  VS_COL8 LONGTEXT
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
m_db=# INSERT INTO string_storage SELECT * FROM bit_storage;
m_db=# SELECT * FROM string_storage;
VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8
\x05 |\x05 |\x05 |\x05 |\x05 |\x05 |\x05
(1 row)
m_db=# DROP TABLE bit_storage, string_storage;
mysql> CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS_COL2 BIT(4),
  VS COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS_COL6 BIT(4),
  VS_COL7 BIT(4),
  VS_COL8 BIT(4)
mysql> CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS_COL6 BIT(4),
  VS COL7 BIT(4),
  VS COL8 BIT(4)
mysql> INSERT INTO bit storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
mysql> INSERT INTO string_storage SELECT * FROM bit_storage;
mysql> SELECT * FROM string_storage;
| VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8 |
                   1 row in set (0.00 sec)
mysgl> DROP TABLE bit_storage, string_storage;
```

● 中间带有0x00的二进制或十六进制字符串插入二进制数据类型时,GaussDB部分插入,0x00后的字符会截断,MySQL可以完整插入。

```
示例:
```

```
m_db=# CREATE TABLE blob_storage (
A BLOB
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO blob_storage VALUES (0xBB00BB);
```

中间带有0x00的二进制或十六进制字符串插入字符串数据类型时,GaussDB部分插入,0x00后的字符会截断,MySQL严格模式不允许插入,宽松模式插入空字符串。

示例:

```
m_db=# CREATE TABLE text_storage (
    A TEXT
m_db=# INSERT INTO text_storage VALUES (b'101110110000000010111011');
m_db=# SELECT hex(A) FROM text_storage;
hex
BB
(1 row)
m_db=# DROP TABLE text_storage;
mysql> CREATE TABLE text_storage (
    A TEXT
mysql> INSERT INTO text_storage VALUES (b'101110110000000010111011');
ERROR 1366 (HY000): Incorrect string value: '\xBB\x00\xBB' for column 'A' at row 1
mysql> SELECT hex(A) FROM text_storage;
Empty set (0.00 sec)
mysql> SET SQL_MODE=";
mysql> INSERT INTO text_storage VALUES (b'101110110000000010111011');
mysql> SELECT hex(A) FROM text_storage;
| hex(A) |
    1 row in set (0.01 sec)
mysql> DROP TABLE text_storage;
```

 WHERE子句中只带有普通字符串, GaussDB中't'、'true'、'yes'、 'y'、'on'返回 TRUE, 'no'、'f'、'off'、'false'、'n'返回FALSE, 其余字符串报错。MySQL通过字 符串转换为INT1判断返回TRUE/FALSE。

示例:

● 对于YEAR类型的输入,在将字符串转换为整型的过程中,MySQL考虑科学计数法,GaussDB暂不支持,统一做截断处理。

示例:

```
m_db=# CREATE TABLE test_year (
    A YEAR
m_db=# SET sql_mode = ";
m_db=# INSERT INTO test_year VALUES ('2E3x');
WARNING: Data truncated for column.
LINE 1: insert into t1 values ('2E3x');
CONTEXT: referenced column: a
m_db=# SELECT * FROM test_year ORDER BY A;
а
2002
(1 row)
m_db=# DROP TABLE test_year;
mysql> CREATE TABLE test_year (
     A YEAR
mysql> INSERT INTO test_year VALUES ('2E3x');
mysql> SELECT * FROM test_year ORDER BY A;
|a |
2000 |
1 row in set (0.01 sec)
mysql> DROP TABLE test_year;
```

显式类型转换差异点:

● GaussDB中平铺成对各目标类型的转换规则,MySQL中使用C++多态重载函数, 在嵌套场景中存在不一致行为。

示例:

UNION, CASE和相关构造差异点:

- POLYGON + NULL、POINT + NULL、POLYGON + POINT组合在MySQL中均返回GEOMETRY类型,GaussDB中未涉及,暂时当做报错处理。
- SET和ENUM两种类型暂未支持,暂时当做报错处理。
- 常量类型和其他类型做类型聚合的时候,输出类型的精度为其他类型的精度。如 "select "helloworld" union select p from t;"的结果的精度为属性p的精度。
- 定点常量和不带精度约束的类型(非字符串类型如int、bool、year等,聚合结果 类型为定点类型)聚合时,精度约束会按照定点数默认精度31输出。
- merge rule差异:

MySQL 5.7中YEAR和TINYINT、INT、MEDIUMINT、BIGINT、BOOL聚合的结果类型为带UNSIGNED的类型,GaussDB的结果类型为不带UNSIGNED的类型;MySQL中BIT和INT、NUMERIC、FLOAT、DOUBLE等数值类型的聚合类型为VARBINARY类型,GaussDB中BIT和INT|NUMERIC聚合为NUMERIC类型,和FLOAT或者DOUBLE类型聚合时,结果为DOUBLE类型,和无符号整型聚合时为UINT8类型。

 MySQL中BINARY和CHAR填充字符不相同,BINARY填充'\0',CHAR填充空格, GaussDB中BINARY和CHAR都是填充空格。

2.3.3 系统函数

GaussDB数据库兼容绝大多数MySQL的系统函数,但存在部分差异。只能使用M-Compatibility兼容模式下的系统函数,不能使用原GaussDB的系统函数,避免产生非预期的结果。当前存在原GaussDB的系统函数和MySQL系统函数同名,但是M-Compatibility兼容模式下尚未支持这些函数的情况;一部分未支持的同名函数会提示用户在M-Compatibility兼容模式下不支持,另外一部分同名函数仍然保持原GaussDB系统函数的行为。用户禁止使用这些同名函数,避免产生非预期的结果。具体同名函数列表如下:

表 2-32 M-Compatibility 兼容模式下提示不支持的同名函数

cot	isEmpty	last_insert_id	mod	octet_length
overlaps	point	radians	regexp_instr	regexp_like
regexp_replac e	regexp_substr	stddev_pop	stddev_samp	var_pop
var_samp	variance	-	-	-

表 2-33 M-Compatibility 兼容模式下保持原 GaussDB 系统函数行为的同名函数

ceil	decode	encode	format	instr
position	round	stddev	row_num	-

□ 说明

- MySQL数据库支持通过可加载函数接口,向MySQL中添加自定义函数,在调用此类函数时,函数的入参支持指定别名。GaussDB不支持可加载函数接口,在调用函数时,函数入参不支持指定别名。
- M-Compatibility模式下,系统函数存在以下公共差异:
 - 系统函数的返回值类型仅考虑入参node类型为Var(表中数据)和Const(常量输入) 类型时的情况与MySQL保持一致,其他情况(如入参为运算表达式、函数表达式等) 可能返回值的类型与MySQL有差异。
 - 当聚合函数以其他函数、操作符或SELECT子句等表达式作为入参时(如SELECT sum(abs(n)) FROM t;),聚合函数将获取不到入参表达式传递的精度信息,导致函数的结果精度与MySQL有差异。
 - 系统函数的调用不推荐使用pg_catalog.func_name()形式的调用,当被调用函数存在语法形式的入参时(如SELECT substr('demo' from 1 for 2)),函数的调用可能存在错误。

2.3.3.1 流程控制函数

表 2-34 流程控制函数列表

序号	MySQL数据库	GaussDB 数据库	差异	
1	IF()	支持。	当第一个参数为TRUE且第三个参数表达式中存在隐式类型转换错误,或者第一个参数为FALSE且第二个参数表达式中存在隐式类型转换错误时,MySQL会忽略该错误,GaussDB会提示类型转换错误。	
2	IFNULL()	支持。	第一个参数不为NULL且第二个参数表 达式中存在隐式类型转换错误时, MySQL会忽略该错误,GaussDB会提 示类型转换错误。	
3	NULLIF()	支持,有 差异。	函数返回值类型在MySQL5.7和 MySQL8.0中存在差异,考虑到 MySQL8.0更合理,因此函数返回值类 型兼容MySQL8.0。	

2.3.3.2 日期和时间函数

山 说明

以下为GaussDB M-Compatibility兼容性日期时间函数公共说明。

 《M-Compatibility开发指南》中"函数和操作符"章节函数入参为时间类型表达式的情况: 时间类型表达式主要包括TEXT、DATETIME、DATE或TIME,但所有可以隐式转换为时间表 达式的类型都可以作为入参,比如数字类型可以通过先隐式转化为TEXT,再作为时间类型表 达式生效。

生效的情况根据函数有所不同,比如: DATEDIFF函数由于只计算日期的差值,所以时间表达式会当做日期来解析。TIMESTAMPDIFF函数由于计算时间差值时是根据UNIT来决定的,所以会根据UNIT选择将时间表达式当做DATE或者TIME或者DATETIME来解析。

• 函数入参为无效日期的情况:

一般而言,日期时间函数支持DATE、DATETIME的范围和MySQL保持一致。DATE支持的范围为'0000-01-01'到'9999-12-31',DATETIME支持的范围为'0000-01-01 00:00:00'到 '9999-12-31 23:59:59'。虽然GaussDB支持的DATE、DATETIME范围大于MySQL,但是越界仍然算无效日期。

大部分时间函数会告警并返回NULL,只有能通过cast正常转换的日期,才是正常合理的日期。

新框架下GaussDB的大部分日期时间函数与MySQL一致,一些函数的差异如下表所示:

表 2-35 日期与和时间函数列表

序号	MySQL数据库	GaussD B数据库	差异
1	ADDDATE()	支持	-
2	ADDTIME()	支持	-
3	CONVERT_TZ()	支持	-
4	CURDATE()	支持	-
5	CURRENT_DATE()/ CURRENT_DATE	支持	-
6	CURRENT_TIME()/ CURRENT_TIME	支持,有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例: SELECT CURRENT_TIME(257) == SELECT CURRENT_TIME(1))。 GaussDB只支持[0,6]合法值,其他值报
			错。
7	CURRENT_TIMESTAMP()/ CURRENT_TIMESTAMP	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例: SELECT CURRENT_TIMESTAMP(257) == SELECT CURRENT_TIMESTAMP(1))。 GaussDB只支持[0,6]合法值,其他值报 错。

序号	MySQL数据库	GaussD B数据库	差异
8	CURTIME()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例: SELECT CURTIME(257) == SELECT CURTIME(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
9	DATE()	支持	-
10	DATE_ADD()	支持	-
11	DATE_FORMAT()	支持	-
12	DATE_SUB()	支持	-
13	DATEDIFF()	支持	-
14	DAY()	支持	-
15	DAYNAME()	支持	-
16	DAYOFMONTH()	支持	-
17	DAYOFWEEK()	支持	-
18	DAYOFYEAR()	支持	-
19	EXTRACT()	支持	-
20	FROM_DAYS()	支持	-
21	FROM_UNIXTIME()	支持	-
22	GET_FORMAT()	支持	-
23	HOUR()	支持	-
24	LAST_DAY()	支持	-
25	LOCALTIME()/ LOCALTIME	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例SELECT LOCALTIME(257) == SELECT LOCALTIME(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
26	LOCALTIMESTAMP/ LOCALTIMESTAMP()	支持,有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例SELECT LOCALTIMESTAMP(257) == SELECT LOCALTIMESTAMP(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
27	MAKEDATE()	支持	-

序号	MySQL数据库	GaussD B数据库	差异
28	MAKETIME()	支持	分布式下推场景下当TIME类型秒位无精度时,MYSQL默认补齐6个0,GaussDB不做补齐。
29	MICROSECOND()	支持	-
30	MINUTE()	支持	-
31	MONTH()	支持	-
32	MONTHNAME()	支持	-
33	NOW()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例SELECT NOW(257)==SELECT NOW(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
34	PERIOD_ADD()	支持, 有差异	1. 整数溢出处理的行为。 MySQL在5.7版本,此函数入参结果最大值为2^32=4294967296,在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题;在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。 2. 负数period的表现。 MySQL在5.7版本,会将负数年份解析为异常值而不是报错。GaussDB入参或结果(如100年1月减去10000月)出现负数时报错。在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。 3. period月份越界的表现。 MySQL在5.7版本中,若月份大于12或等于0,例如200013、199900,会将其顺延到之后的年份,或者将0月作为上一年12月处理。GaussDB会对越界月份进行报错。在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。

序号	MySQL数据库	GaussD B数据库	差异
35	PERIOD_DIFF()	支持,有差异	1. 整数溢出处理的行为。 MySQL在5.7版本,此函数入参结果最大值为2^32=4294967296,在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题;在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。 2. 负数period的表现。 MySQL在5.7版本,会将负数年份解析为异常值而不是报错。GaussDB入参或结果(如100年1月减去10000月)出现负数时报错。在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。 3. period月份越界的表现。 MySQL在5.7版本中,若月份大于12或等于0,例如200013、199900,会将其顺延到之后的年份,或者将0月作为上一年12月处理。GaussDB会对越界月份进行报错。在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。
36	QUARTER()	支持	-
37	SEC_TO_TIME()	支持	-
38	SECOND()	支持	-
39	STR_TO_DATE()	支持	返回值与MySQL有差异,GaussDB返回 的是text,MySQL返回的是datetime、 date。
40	SUBDATE()	支持	-
41	SUBTIME()	支持	-
42	SYSDATE()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕。 GaussDB不回绕。
43	TIME()	支持	-
44	TIME_FORMAT()	支持	-
45	TIME_TO_SEC()	支持	-
46	TIMEDIFF()	支持	-
47	TIMESTAMP()	支持	-

序号	MySQL数据库	GaussD B数据库	差异
48	TIMESTAMPADD()	支持	-
49	TIMESTAMPDIFF()	支持	-
50	TO_DAYS()	支持	-
51	TO_SECONDS()	支持	-
52	UNIX_TIMESTAMP()	支持	MySQL会根据入参是否存在小数位,决定返回定点型还是整型。当前GaussDB在内层嵌套操作符或函数时,返回的类型与MySQL可能存在不同。当内层节点返回定点、浮点、字符型、时间类型(不包括DATE类型)时,MySQL可能返回整型,GaussDB会返回定点型。
53	UTC_DATE()	支持	-
54	UTC_TIME()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕,GaussDB只支持[0,6]合 法值,其他值报错。
55	UTC_TIMESTAMP()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕,GaussDB只支持[0,6]合 法值,其他值报错。
56	WEEK()	支持	-
57	WEEKDAY()	支持	-
58	WEEKOFYEAR()	支持	-
59	YEAR()	支持	-
60	YEARWEEK()	支持	-

2.3.3.3 字符串函数

表 2-36 字符串函数列表

序号	MySQL数据库	GaussDB数 据库	差异
1	ASCII()	支持。	-
2	BIT_LENGTH()	支持。	-
3	CHAR_LENGTH()	支持,有差 异。	GaussDB此函数如果数据库字符集是 SQL_ASCII,CHAR_LENGTH()会返回字节 数而非字符数。

序号	MySQL数据库	GaussDB数 据库	差异
4	CHARACTER_LE NGTH()	支持,有差 异。	GaussDB此函数如果数据库字符集是 SQL_ASCII,CHARACTER_LENGTH()会返 回字节数而非字符数。
5	CONCAT()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为LONGBLOB; 当MySQL返回值类型为非二进制字符串类 型(CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。
6	CONCAT_WS()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为LONGBLOB; 当MySQL返回值类型为非二进制字符串类 型(CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。
7	HEX()	支持。	-
8	LENGTH()	支持。	-
9	LPAD()	支持,有差 异。	 MySQL默认最大填充长度为1398101,GaussDB默认最大长度为1048576。在不同字符集下,最大填充长度会有差异,例如字符集为'GBK'时,GaussDB默认最大长度为2097152。 如果数据库字符集是SQL_ASCII,可能产生未预期的结果。 当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)
			时,GaussDB对应的返回值类型为 LONGBLOB;当MySQL返回值类型为非 二进制字符串类型(CHAR、 VARCHAR、TEXT等)时,GaussDB对应 的返回值类型为TEXT。
10	REPEAT()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为LONGBLOB; 当MySQL返回值类型为非二进制字符串类 型(CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。
11	REPLACE()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为LONGBLOB; 当MySQL返回值类型为非二进制字符串类 型(CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。

序号	MySQL数据库	GaussDB数 据库	差异
12	RPAD()	支持,有差 异。	MySQL默认最大填充长度为1398101,GaussDB默认最大长度为1048576。在不同字符集下,最大填充长度会有差异,例如字符集为'GBK'时,GaussDB默认最大长度为2097152。 如果数据库字符集是SQL_ASCII,可能产
			生未预期的结果。 • 当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)时,GaussDB对应的返回值类型为LONGBLOB;当MySQL返回值类型为非二进制字符串类型(CHAR、VARCHAR、TEXT等)时,GaussDB对应的返回值类型为TEXT。
13	SPACE()	支持。	-
14	STRCMP()	支持,有差 异。	如果数据库字符集是SQL_ASCII,可能产生 未预期的结果。
15	FIND_IN_SET()	支持,有差	当指定数据库使用的字符编码是SQL_ASCII
16	LCASE()	异。	│时,服务器把字节值0~127根据ASCII标准解 │释,而字节值128~255则当作无法解析的字
17	LEFT()	1	符;如果该函数的输入输出包含了任何非 ASCII数据,数据库将无法帮助用户转换或
18	LOWER()		者校验非ASCII字符。
19	LTRIM()		当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等) 时,
20	REVERSE()		GaussDB对应的返回值类型为LONGBLOB; 当MySQL返回值类型为非二进制字符串类
21	RIGHT()		型(CHAR、VARCHAR、TEXT等)时,
22	RTRIM()		GaussDB对应的返回值类型为TEXT。
23	SUBSTR()		
24	SUBSTRING()		
25	SUBSTRING_IN DEX()		
26	TRIM()		
27	UCASE()		
28	UPPER()		
29	UNHEX()	支持。	MySQL的返回值类型为BINARY、 VARBINARY、BLOB、MEDIUMBLOB或 LONGBLOB;GaussDB返回值类型固定为 LONGBLOB。

序号	MySQL数据库	GaussDB数 据库	差异
30	FIELD()	支持。	-
31	FORMAT()	支持。	-

2.3.3.4 强制转换函数

表 2-37 强制转换函数列表

序号	MySQL数据 库	GaussDB数据 库	差异
1	CAST()	支持	 GaussDB不支持使用CAST(expr AS CHAR[(N)] charset_info或者 CAST(expr AS NCHAR[(N)])转换字符 集。
			 GaussDB支持使用CAST(expr AS FLOAT[(p)])或CAST(expr AS DOUBLE) 将表达式转换为浮点类型, MySQL 5.7 版本不支持此转换。
			● GaussDB不支持使用CAST(expr AS JSON)将表达式转换为JSON。
			● 对于CAST嵌套子查询场景,如果子查询语句返回的是非确的数值,MySQL 5.7版本返回失真数值,BINARY函数使用CAST实现,同理。GaussDB m_db=# CREATE TABLE sub_query_table (myfloat float) DISTRIBUTE BY REPLICATION; CREATE TABLE m_db=# INSERT INTO sub_query_table (myfloat) VALUES (1.23); INSERT 0 1 m_db=# SELECT BINARY (select MyFloat from sub_query_table) from sub_query_table; binary 1.23 (1 row) m_db=# SELECT CAST((select MyFloat from sub_query_table) AS char); cast 1.23 (1 row)Mysql 5.7 mysql> CREATE TABLE sub_query_table (myfloat float); Query OK, 0 rows affected (0.02 sec) mysql> INSERT INTO sub_query_table (myfloat float); Query OK, 1 row affected (0.00 sec) mysql> SELECT BINARY (select MyFloat from sub_query_table) from sub_query_table; +
			AS char)

序号	MySQL数据 库	GaussDB数据 库	差异	
			++ 1.2300000190734863	
2	CONVERT()	支持	 GaussDB不支持使用CONVERT(expr, CHAR[(N)] charset_info或者 CAST(expr, NCHAR[(N)])转换字符集。 	
			 GaussDB支持使用CONVERT(expr, FLOAT[(p)])或CONVERT(expr, DOUBLE)将表达式转换为浮点类型, MySQL 5.7版本不支持此转换。 	
			● GaussDB不支持使用CONVERT(expr, JSON)将表达式转换为JSON。	

2.3.3.5 加密函数

表 2-38 加密函数列表

序号	MySQL数 据库	GaussDB 数据库	差异
1	AES_DECRY PT()	支持。	1. ecb为不安全加密模式,GaussDB不支持,默 认为cbc模式。
2	AES_ENCRY PT()	支持。	 GaussDB中,当指定数据库使用的字符编码是SQL_ASCII时,服务器把字节值0-127根据ASCII标准解释,而字节值128-255则当作无法解析的字符;如果该函数的输入输出包含了任何非ASCII数据,数据库将无法帮助用户转换或者校验非ASCII字符。 MySQL的返回值类型为BINARY、VARBINARY、BLOB、MEDIUMBLOB、LONGBLOB,GaussDB返回值类型固定为LONGBLOB。
3	SHA()/ SHA1()	支持。	-
4	SHA2()	支持。	-

2.3.3.6 比较函数

表 2-39 比较函数列表

序号	MySQL 数据库	GaussDB 数据库	差异
1	COALES CE()	支持,有 差异。	union distinct场景下,返回值精度与MySQL不完全一致。 当第一个不为NULL的参数的后续参数表达式中存在隐式类型转换错误时,MySQL会忽略该错误,GaussDB会提示类型转换错误。当参数为MIN函数、MAX函数时,返回值类型与MySQL不一致。
2	INTERV AL()	支持。	-
3	GREATE ST()	支持,有 差异。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时,GaussDB对 应的返回值类型为LONGBLOB;当MySQL返回值类 型为非二进制字符串类型(CHAR、VARCHAR、 TEXT等)时,GaussDB对应的返回值类型为TEXT。 当该函数入参含有NULL且在WHERE关键字之后调 用,返回结果与MySQL5.7不一致,此处为MySQL5.7 存在的问题,MySQL8.0修复了该问题,目前 GaussDB和MySQL8.0保持一致。
4	LEAST()	支持,有 差异。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时,GaussDB对 应的返回值类型为LONGBLOB;当MySQL返回值类 型为非二进制字符串类型(CHAR、VARCHAR、 TEXT等)时,GaussDB对应的返回值类型为TEXT。 当该函数入参含有NULL且在WHERE关键字之后调 用,返回结果与MySQL5.7不一致,此处为MySQL5.7 存在的问题,MySQL8.0修复了该问题,目前 GaussDB和MySQL8.0保持一致。
5	ISNULL()	支持。	-

2.3.3.7 聚合函数

表 2-40 聚合函数列表

序号	MySQL数据 库	GaussDB数 据库	差异
1	AVG()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。
			GaussDB中当expr中的列为BIT、BOOL、整数类型,且所有行的和超过BIGINT的范围时,会发生溢出导致整数翻转。
2	BIT_AND()	支持。	-
3	BIT_OR()	支持。	-
4	BIT_XOR()	支持。	-
5	COUNT()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。
6	GROUP_CO NCAT()	支持,有差 异。	 GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。
			 GaussDB中当GROUP_CONCAT参数中同时 有DISTINCT和ORDER BY语法时,所有 ORDER BY后的表达式必须也在DISTINCT的 表达式之中。
			 GaussDB中GROUP_CONCAT(ORDER BY 数字)不代表按照第几个参数的顺序,数字 只是一个常量表达式,相当于不排序。
			 GaussDB中使用参数group_concat_max_len 限制GROUP_CONCAT最大返回长度,超长 截断,目前能返回的最大长度是 1073741823,小于MySQL。
7	MAX()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。当参 数为非表字段时,MAX函数返回值类型和 MySQL 5.7不一致。
8	MIN()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。当参 数为非表字段时,MIN函数返回值类型和 MySQL 5.7不一致。
9	SUM()	支持,有差 异。	 GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。
			 GaussDB中当expr中的列为BIT、BOOL、整数类型,且所有行的和超过BIGINT的范围时,会发生溢出导致整数翻转。

2.3.3.8 数字操作函数

表 2-41 数字操作函数列表

序号	MySQL数据库	GaussDB数据	差异
		库	
1	ABS()	支持。	-
2	ACOS()	支持。	-
3	ASIN()	支持。	-
4	ATAN()	支持。	-
5	ATAN2()	支持。	-
6	CEILING()	支持。	部分操作结果类型与MySQL不一致。对于推导结果类型为 NUMERIC或者整型的,如果能够 被整型类型存储,MySQL中结果 类型为整型;GaussDB中类型仍然 为NUMERIC类型。
7	COS()	支持。	-
8	DEGREES()	支持。	-
9	EXP()	支持。	-
10	FLOOR()	支持,有差 异。	FLOOR函数的返回值类型与 MySQL的有差异:入参类型为 INT,GaussDB返回值类型为 BIGINT,MySQL返回值类型为 INT。 部分操作结果类型与MySQL不一致。对于推导结果类型为 NUMERIC或者整型的,如果能够被整型类型存储,MySQL中结果 类型为整型;GaussDB中类型仍然 为NUMERIC类型。
11	LN()	支持。	-
12	LOG()	支持。	-
13	LOG10()	支持。	-
14	LOG2()	支持。	-
15	PI()	支持。	PI函数的返回值精度与MySQL的有差异: MySQL中PI函数的结果仅保留四舍五入之后的小数后6位,而GaussDB的结果会保留四舍五入之后的小数后15位。
16	POW()	支持。	-

序号	MySQL数据库	GaussDB数据 库	差异
17	POWER()	支持。	-
18	RAND()	支持。	-
19	SIGN()	支持。	-
20	SIN()	支持。	-
21	SQRT()	支持。	-
22	TAN()	支持。	-
23	TRUNCATE()	支持。	-
24	CEIL()	支持。	-

2.3.3.9 其他函数

表 2-42 其他函数列表

序号	MySQL数 据库	GaussDB 数据库	差异
1	DATABAS E()	支持。	-
2	UUID()	支持。	-
3	UUID_SH ORT()	支持。	-

2.3.4 操作符

GaussDB数据库兼容绝大多数MySQL的操作符,但存在部分差异。如未列出,操作符行为默认为GaussDB原生行为,目前存在MySQL不支持但是GaussDB支持的语句,不建议使用这类语句。

操作符差异

- ORDER BY排序对NULL值处理的差异。MySQL在排序时会将NULL值排序在前面; GaussDB默认将NULL值默认排在最后面。GaussDB可以通过nulls first和nulls last设置NULL值排序顺序。
- 有ORDER BY时,GaussDB输出顺序与MySQL一致。没有ORDER BY时,GaussDB不保证结果有序。
- MySQL操作符要使用括号严格表达式的结合性,否则执行报错。例如: SELECT 1 regexp ('12345' regexp '123');。

GaussDB M-Compatibility操作符不用括号严格表达的结合性也能成功执行。

● NULL值显示不同。MySQL会将NULL显示为"NULL";GaussDB将NULL值显示为空值。

MySQL输出结果:

```
mysql> Select NULL;
+-----+
| NULL |
+-----+
| NULL |
+-----+
1 row in set (0.00 sec)
```

GaussDB输出结果:

```
m_db=# select NULL;
?column?
------
(1 row)
```

- 操作符执行后,列名显示不一致。MySQL会将NULL显示为"NULL";GaussDB将NULL值显示为空值。
- 字符串转double遇到非法字符串时,告警信息不一致。MySQL在常量非法字符串报错,字段非法字符串不报错;GaussDB在常量非法字符串和字段非法字符串都报错。
- 比较操作符返回结果显示不同。MySQL返回1/0; GaussDB返回t/f。

表 2-43 操作符

序号	MySQL数据库	GaussDB数据库	差异
1	<>	支持,但有差异。	MySQL支持索引, GaussDB不支持索引。
2	<=>	支持,但有差异。	MySQL支持索引, GaussDB不支持索引、 hash连接和合并连接。

行比较、GaussDB不支 持<=>操作符行比较。 ● MySQL不支持行表达式 与NULL比较。GaussDB	序号	MySQL数据库	GaussDB数据库	差异
>、<>操作符对行表达式与NULL值比较。 MySQL不支持IS NULL、ISNULL对行表达式的不支持的操作。GaussDB错误信息与MySQL不一致。 GaussDB错误信息与MySQL不一致。 GaussDB:m_db=# SELECT (1,2) <> row(2,3); FEROR: could not determine interpretation of row comparison operator <=> LINE 1: select (1,2) <> row(2,3); All INT: unsupported operator, m_db=# SELECT (1,2) <> row(2,3); All INT: unsupported operator, m_db=# SELECT (1,2) <> NULL; Redumn? (1 row) m_db=# SELECT (1,2) <> NULL; Redumn? (1 row) m_db=# SELECT (1,2) IS NULL; Redumn? (1 row) m_db=# SELECT (1,2) IS NULL; Redumn? f (1 row) m_db=# SELECT (1,2) IS N	序号 3	-	7.7.50	 MySQL支持<=>操作符行比较。 GaussDB不支持<=>操作符行比较。 GaussDB 支持<=>等人<=>等人<=>等人<=>等人<=>等人<=>等人<=>等人<=>等人
1 row in set (0.00 sec)				mysql> SELECT (1,2) < NULL;

序号	MySQL数据库	GaussDB数据库	差异
			ERROR 1241 (21000): Operand should contain 2 column(s) mysql> SELECT (1,2) <> NULL; ERROR 1241 (21000): Operand should contain 2 column(s) mysql> SELECT (1, 2) IS NULL; ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT ISNULL((1, 2)); ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT NULL BETWEEN NULL AND ROW(2,2); ERROR 1241 (21000): Operand should contain 1 column(s)
4		支持。	MySQL表示对一个操作数 进行两次取反,结果等于 原操作数;GaussDB表示 注释。
5	!!	支持,但有差异。	MySQL: !!含义同!,表示取非。 GaussDB: !表示取非操作,当!与!中间存在空格时,表示连续两次取非(!!);当!与!中间没有空格时,表示阶乘(!!)。 说明 GaussDB中,当同时使用阶乘(!!)和取非(!)时,阶乘(!!)和取非(!)时,阶乘(!!)和取非(!)中间需要添加空格,否则会报错。 GaussDB中,当需要多次取非操作时,!与!之间需使用空格隔开。

序号	MySQL数据库	GaussDB数据库	差异
6	[NOT] REGEXP	支持,但有差异。	• GaussDB和MySQL在正则表达式中支持的元字符有所不同。例如,GaussDB支持"\d"表示数字,"\w"表示字母、数字和下划线,"\s"表示空格,而MySQL不支持这些元字符,MySQL会把这些字符当成正常字符串。
			\b'匹配,MySQL匹配失 败。 ■ GaussDB新框架下中使 用"\"表示转义字符, 而MySQL中使用"\
			│ \"。 ● MySQL不支持2个操作 符连在一起使用。
			● 模式字符串pat非法入 参,只存在右单括 号')'时,GaussDB数 据库报错。MySQL存在 bug,后续版本已经修 复此问题。
			在de abc匹配序列de或abc的匹配规则,当 左右存在空值时,MySQL存在bug,会报错,后续版本已经修复此问题。
			 空白字符[\t]正则匹配字符类[:blank:], GaussDB可匹配, MySQL\t不能匹配 [:blank:], MySQL存在bug,后续版本已经修复此问题。
			• GaussDB支持非贪婪模式匹配,即尽可能少的匹配字符,在部分特殊字符后加'?'问号字符,例如:"??,*?,+?,{n}?,{n,}?,{n,m}?"。MySQL 5.7版本不支持非贪婪模式匹配,并报错:Got error 'repetition-operator

序号	MySQL数据库	GaussDB数据库	差异
			operand invalid' from regexp。MySQL 8.0版 本已经支持。
			在binary字符集下,text 类型、blob类型均会转 换成bytea类型,由于 REGEXP操作符不支持 bytea类型,因此无法匹配。
7	LIKE	支持,但有差异。	MySQL: LIKE的左操作数 只能是位运算或者算术运 算或者由括号组成的表达 式,LIKE的右操作数只能是 单目运算符(不含NOT)或者 括号组成的表达式。
			GaussDB: LIKE的左右操作数可以是任意表达式。
8	[NOT] BETWEEN AND	支持,但有差异。	MySQL: [NOT] BETWEEN AND嵌套使用 时从右到左结合。[NOT] BETWEEN AND的第1个操 作数和第2个操作数只能是 位运算或者算术运算或者 由括号组成的表达式。 GaussDB: [NOT] BETWEEN AND嵌套使用 时从左到右结合。[NOT] BETWEEN AND的第1个操 作数和第2个操作数可以是 任意表达式。
9	IN	支持,但有差异。	MySQL: IN的左操作数只 能是位运算或者算术运算 或者由括号组成的表达 式。 GaussDB: IN的左操作数 可以是任意表达式。
10	!	支持,但有差异。	MySQL:!的操作数只能是 单目运算符(不含not)或者 括号组成的表达式。 GaussDB:!的操作数可以 是任意表达式。
11	#	不支持。	MySQL支持#注释, GaussDB不支持#注释。

序号	MySQL数据库	GaussDB数据库	差异
12	BINARY	支持,但有差异。	GaussDB中支持的表达式和MySQL并不完全一致(包括一些函数、操作符等)。GaussDB独有的表达式'~'、'IS DISTINCTFROM'等,由于BINARY关键字优先级更高,使用BINARY expr会优先将BINARY与'~'、'ISDISTINCTFROM'的左参数合并,导致报错。
13	取反(-)	支持,但有差异。	连续取反次数超过1次时, 结果类型与MySQL存在差 异。

序号	MySQL数据库	GaussDB数据库	差异
序号 14	MySQL数据库 XOR、 、&、 <、>、<=、 >=、=、!=	支持,但执行机制存在差异。	差异 MySQL执行机制为执行左操作数后,对结果进行判断是否为空,进而决定是否需要执行右操作数。 GaussDB执行机制是执行左右操作数后,对结果再进行判断是否为空。 当左操作数结果为空,右操作数执行报错时,MySQL不会报错直接返回,GaussDB会执行报错。 MySQL行为: mysql> SELECT version(); +
			version() ++ 5.7.44-debug-log ++ 1 row in set (0.00 sec) mysql> dROP TABLE IF EXISTS data_type_table; Query OK, 0 rows affected (0.02 sec)
			mysql> CREATE TABLE data_type_table (-> MyBool BOOL, -> MyBinary BINARY(10), -> MyYear YEAR ->); Query OK, 0 rows affected (0.02 sec)
			mysql> INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021'); Query OK, 1 row affected (0.00 sec) mysql> SELECT (MyBool %
			MyBinary) (MyBool - MyYear) FROM data_type_table; (MyBool % MyBinary) (MyBool - MyYear) +
			+ NULL + + 1 row in set, 2 warnings (0.00 sec)
			GaussDB行为: m_db=# DROP TABLE IF EXISTS data_type_table; DROP TABLE m_db=# CREATE TABLE data_type_table (m_db(# MyBool BOOL,

序号	MySQL数据库	GaussDB数据库	差异
			m_db(# MyBinary BINARY(10), m_db(# MyYear YEAR m_db(#); CREATE TABLE m_db=# INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021'); INSERT 0 1 m_db=# SELECT (MyBool % MyBinary) (MyBool - MyYear) FROM data_type_table; WARNING: Truncated incorrect double value: '4Vx ' CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear) WARNING: division by zero CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear) ERROR: Bigint is out of range. CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear) ERROR: Bigint is out of range. CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear)

表 2-44 操作符组合存在差异

操作符组合示例	MySQL 数据库	GaussD B数据库	说明
SELECT 1 LIKE 3 & 1;	不支持	支持	LIKE的右操作数不能是位运算符组 成的表达式。
SELECT 1 LIKE 1 +1;	不支持	支持	LIKE的右操作数不能是算术运算符 组成的表达式。
SELECT 1 LIKE NOT 0;	不支持	支持	LIKE的右操作数只能是+、-、!等 单目操作符或者括号组成的表达 式,NOT除外。
SELECT 1 BETWEEN 1 AND 2 BETWEEN 2 AND 3;	从右到 左结合	从左到 右结合	建议加上括号明确优先级。
SELECT 2 BETWEEN 1=1 AND 3;	不支持	支持	BETWEEN的第2个操作数不能是 比较操作符组成的表达式。
SELECT 0 LIKE 0 BETWEEN 1 AND 2;	不支持	支持	BETWEEN的第1个操作数不能是 模式匹配操作符组成的表达式。
SELECT 1 IN (1) BETWEEN 0 AND 3;	不支持	支持	BETWEEN的第1个操作数不能是 IN操作符组成的表达式。
SELECT 1 IN (1) IN (1);	不支持	支持	第2个IN表达式左操作数不能是IN 组成的表达式。

操作符组合示例	MySQL 数据库	GaussD B数据库	说明
SELECT! NOT 1;	不支持	支持	!的操作数只能是+、-、!等单目操作符或者括号组成的表达式, NOT除外。

山 说明

在GaussDB中支持,但在MySQL中不支持的操作符组合不建议使用。建议按照MySQL中的使用规则对操作符进行组合使用。

索引差异

- GaussDB当前仅支持UBTree和B-tree索引。
- 针对模糊匹配(LIKE操作符), MySQL创建默认索引可以走索引; GaussDB默认的索引不走索引,需要用户使用以下语法指定opclass,比如指定为text_pattern_ops,LIKE操作符才可以走索引。
 CREATE INDEX indexname ON tablename(col [opclass]);
- B-tree/UBTree索引场景保持原生GaussDB原有逻辑,即同一操作符族内的类型比较,支持索引扫描,其余索引类型暂未支持。
- 通过GaussDB JDBC连接数据库,GaussDB YEAR类型在含有绑定参数的PBE场景下索引无法使用。
- where子句中,索引字段类型和常量类型操作场景下,GaussDB中索引与MySQL 索引支持存在差异,如下表所示。例如以下语句GaussDB不支持索引:

create table t(_int int);

create index idx on t(_int) using BTREE;

select * from t where _int > 2.0;

□ 说明

where子句里索引字段类型和常量类型操作场景中,可以使用cast函数将常数类型显示转为字段类型,以便实现索引。

select * from t where _int > cast(2.0 as signed);

表 2-45 索引支持存在差异

索引字段类型	常量类型	GaussDB	MySQL
整型	整型	是	是
浮点型	浮点型	是	是
定点型	定点型	是	是
字符串类型	字符串类型	是	是
二进制类型	二进制类型	是	是
带日期的时间类 型	带日期的时间类 型	是	是
TIME类型	TIME类型	是	是

索引字段类型	常量类型	GaussDB	MySQL
带日期的时间类 型	可转为带日期的 时间类型(如 20231130等整 型)	是	是
带日期的时间类 型	TIME类型	是	是
TIME类型	可转为TIME类型 的常量(如 203008等整型)	是	是
浮点型	整型	是	是
浮点型	定点型	是	是
浮点型	字符串类型	是	是
浮点型	二进制类型	是	是
浮点型	带日期的时间类 型	是	是
浮点型	TIME类型	是	是
定点型	整型	是	是
字符串类型	带日期的时间类 型	是	否
字符串类型	TIME类型	是	否
二进制类型	字符串类型	是	是
二进制类型	带日期的时间类 型	是	否
二进制类型	TIME类型	是	否
整型	浮点型	否	是
整型	定点型	否	是
整型	字符串类型	否	是
整型	二进制类型	否	是
整型	带日期的时间类 型	否	是
整型	TIME类型	否	是
定点型	浮点型	否	是
定点型	字符串类型	否	是
定点型	二进制类型	否	是

索引字段类型	常量类型	GaussDB	MySQL
定点型	带日期的时间类 型	否	是
定点型	TIME类型	否	是
字符串类型	二进制类型	否	是
带日期的时间类 型	整型(不可转为 带日期的时间类 型)	否	是
带日期的时间类 型	浮点型(不可转 为带日期的时间 类型)	否	是
带日期的时间类 型	定点型(不可转 为带日期的时间 类型)	否	是
TIME类型	整型(不可转为 TIME类型)	否	是
TIME类型	字符串类型(不可转为TIME类型)	否	是
TIME类型	二进制类型(不可转为TIME类型)	否	是
TIME类型	带日期的时间类 型	否	是
YEAR类型	YEAR类型	是	是
YEAR类型	可转为YEAR类型 的常量(如2034 等整型)	是	是
BIT类型	BIT类型	否	是

表 2-46 是否支持走索引

索引字段类型	常量类型	是否走索引	MySQL
字符串类型	整型	否	否
字符串类型	浮点型	否	否
字符串类型	定点型	否	否
二进制类型	整型	否	否
二进制类型	浮点型	否	否

索引字段类型	常量类型	是否走索引	MySQL
二进制类型	定点型	否	否
带日期的时间类 型	字符串类型(不可转为带日期的时间类型)	否	否
带日期的时间类 型	二进制类型(不可转为带日期的时间类型)	否	否
TIME类型	浮点型(不可转 为TIME类型)	否	否
TIME类型	定点型(不可转 为TIME类型)	否	否
YEAR类型	不可转为YEAR类 型的常量	否	否
BIT类型	字符串类型	否	否

2.3.5 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集,支持的范围如下。

表 2-47 字符集列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4	支持
2	utf8	支持
3	gbk	支持
4	gb18030	支持
5	binary	支持

□ 说明

- utf8和utf8mb4在GaussDB中为同一个字符集,编码最大长度为4字节。当前字符串字符集为 utf8,指定其字符序为utf8mb4_bin/utf8mb4_general_ci/utf8mb4_unicode_ci/ utf8mb4_0900_ai_ci时(例如select _utf8'a' collate utf8mb4_bin),MySQL会发生报错,GaussDB不报错。当字符串字符集为utf8mb4,指定其字符序为utf8_bin/utf8_general_ci/ utf8_unicode_ci时,存在同样差异。
- 词法语法解析按照字节流解析,当多字节字符中包含与'\', '\'', '\\'等符号一致的编码时,会导致与MySQL行为不一致,建议暂时关闭转义符开关进行规避。

2.3.6 排序规则

GaussDB数据库支持指定模式、表或列的排序规则,支持的范围如下。

山 说明

排序规则差异说明:

- 当前仅有字符串类型、部分二进制类型支持指定排序规则,其他类型不支持指定排序规则,可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。
 MySQL中所有类型可以指定字符序,但除字符串、二进制类型其他排序规则无实际意义。
- 当前排序规则(除binary外)仅支持在其对应字符集与库级字符集一致时可以指定, GaussDB数据库中,字符集必须与数据库的字符集一致,且不支持表内多种字符集混合使用。
- utf8mb4字符集下默认字符序为utf8mb4_general_ci,与MySQL5.7保持一致。

表 2-48 排序规则列表

序号	MySQL数据库	GaussDB数据库	
1	utf8mb4_general_ci	支持	
2	utf8mb4_unicode_ci	支持	
3	utf8mb4_bin	支持	
4	gbk_chinese_ci	支持	
5	gbk_bin	支持	
6	gb18030_chinese_ci	支持	
7	gb18030_bin	支持	
8	binary	支持	
9	utf8mb4_0900_ai_ci	支持	
10	utf8_general_ci	支持	
11	utf8_bin	支持	
12	utf8_unicode_ci	支持	

2.3.7 事务

GaussDB数据库兼容MySQL的事务,但存在部分差异。本章节介绍GaussDB的M-Compatibility数据库中事务相关的差异。

事务默认隔离级别

M-Compatibility默认隔离级别为READ COMMITTED,MySQL默认隔离级别为REPEATABLE-READ。

-- 查看当前事务隔离级别。

m_db=# SHOW transaction_isolation;

子事务

M-Compatibility中,通过SAVEPOINT用于在当前事务里建立一个新的保存点(子事务),使用ROLLBACK TO SAVEPOINT回滚到一个保存点(子事务),子事务回滚后父事务可以继续运行,子事务的回滚不影响父事务的事务状态。

MySQL不存在创建保存点(子事务)。

嵌套事务

嵌套事务指在事务块中开启新事务。

M-Compatibility中,正常事务块中开启新事务会警告存在一个进行中的事务,忽略开启命令;异常事务块中开启新事物将报错,必须在执行ROLLBACK/COMMIT之后才可以执行,执行ROLLBACK/COMMIT会回滚之前语句。

MySQL中,正常事务块中开启新事务会先把之前事务提交,然后开启新事务; 异常事务块中开启新事务会忽略错误,提交之前无错误的语句并开启新事务。

```
-- M-Compatibility正常事务块中,开启新事务会警告并忽略。
m_db=# DROP TABLE IF EXISTS test_t;
m_db=# CREATE TABLE test_t(a int, b int);
m_db=# BEGIN;
m_db=# BEGIN; -- 会警告there is already a transaction in progress。
m_db=# SELECT * FROM test_t ORDER BY 1;
m_db=# COMMIT;
-- M-Compatibility异常事务块中,开启新事务会报错,必须ROLLBACK/COMMIT之后才可以执行。
m_db=# BEGIN;
m_db=# BEGIN;
m_db=# BEGIN; -- 错误语句。
m_db=# BEGIN; -- 报错。
m_db=# COMMIT; -- ROLLBACK/COMMIT之后才可以执行。
```

隐式提交的语句

M-Compatibility使用GaussDB存储,继承GaussDB事务机制,事务中执行DDL、DCL不会自动提交。

MySQL在DDL、DCL、管理类语句,锁相关语句会自动提交。

```
-- M-Compatibility创建表和设置GUC参数可以回滚掉。
m_db=# DROP TABLE IF EXISTS test_table_rollback;
m_db=# BEGIN;
m_db=# CREATE TABLE test_table_rollback(a int, b int);
m_db=# \d test_table_rollback;
m_db=# ROLLBACK;
m_db=# \d test_table_rollback; -- 不存在该表。
```

SET TRANSACTION 差异

M-Compatibility中,SET TRANSACTION同时设置多次隔离级别/事务访问模式时,只有最后一个会生效;多个事务特性支持使用空格和逗号分隔。

MySQL中SET TRANSACTION不允许设置多次隔离级别/事务访问模式;多个事务特性只支持使用逗号分隔。

表 2-49 SET TRANSACTION 差异

序号	语法	功能	差异
1	SET TRANSACTIO N	设置事务特性。	M-Compatibility中,SET TRANSACTION在会话级别生效; MySQL中SET TRANSACTION在下一个 事务生效。
2	SET SESSION TRANSACTIO N	设置会话级事务特 性。	-
3	SET GLOBAL TRANSACTIO N	设置全局会话级事 务特性,该特性适 用于后续会话,对 当前会话无影响。	M-Compatibility中,GLOBAL是全局会 话级别生效,只针对当前数据库实例, 其它数据库不影响。 MySQL中,会使所有数据库生效。

-- SET TRANSACTION会话级生效。

m_db=# SET TRANSACTION ISOLATION LEVEL READ COMMITTED READ WRITE;

m db=# SHOW transaction isolation;

m_db=# SHOW transaction_read_only;

-- M-Compatibility同时设置多次隔离级别/事务访问模式,最后一个生效。

m_db=# SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED, ISOLATION LEVEL

REPEATABLE READ, READ WRITE, READ ONLY;

m_db=# SHOW transaction_isolation; -- repeatable read

m_db=# SHOW transaction_read_only; -- on

START TRANSACTION 差异

M-Compatibility中,START TRANSACTION开启事务时,同时支持设置隔离级别;同时设置多次隔离级别/事务访问模式时,只有最后一个会生效;当前版本不支持立即开启一致性快照;多个事务特性支持空格和逗号分隔。

MySQL的start transaction 开启事务时,不支持设置隔离级别,不支持设置多次事务访问模式;多个事务特性只支持逗号分隔。

-- 开启事务设置隔离级别。

m_db=# START TRANSACTION ISOLATION LEVEL READ COMMITTED;

m db=# COMMIT;

-- 多次设置访问模式。

m_db=# START TRANSACTION READ ONLY, READ WRITE;

m_db=# COMMIT;

事务相关的 GUC 参数

表 2-50 事务相关的 GUC 参数差异

序号	GUC参数	功能	差异
1	autocomm it	设置事务自 动提交模 式。	-

序号	GUC参数	功能	差异
2	transactio n_isolation	在M-Compatibility当高 在是级离图 在MySQL中前离 MySQL等别。在MySQL等别。	 GaussDB中,通过使用SET命令,只能改变当前事务的隔离级别。如果想要改变会话级的隔离级别,可以使用default_transaction_isolation。在MySQL中,通过使用SET命令,可以改变会话级的事务隔离级别。 支持范围差异。 MySQL中,支持以下隔离级别设置,对大小写不敏感,对空格敏感: - READ-COMMITTED - READ-UNCOMMITTED - REPEATABLE-READ - SERIALIZABLE GaussDB中,支持以下隔离级别设置,对大小写和空格敏感: - read committed - read uncommitted - repeatable read - serializable - default (设置和会话中默认隔离级别一样) - 设置m_format_dev_version = 's2'时,支持MySQL的隔离级别设置。 在GaussDB中,新事务的transaction_isolation值有被初始化为default_transaction_isolation的值。
3	tx_isolatio n	设置事务的 隔离级别; tx_isolation 和 transaction _isolation是 同义词。	M-Compatibility中只支持查询,不支持修改。
4	default_tr ansaction_ isolation	设置事务的 隔离级别。	M-Compatibility中通过SET设置会改变会话级事 务隔离级别。 MySQL中不支持该系统参数。

序号	GUC参数	功能	差异
5	transactio n_read_on ly	设置事务的访问模式。	 在M-Compatibility中,通过使用SET命令,只能改变当前事务的访问模式。如果想要改变会话级的访问模式,可以使用default_transaction_read_only。在MySQL中,通过使用SET命令,可以改变会话级的事务隔离级别。 在GaussDB中,新事务的transaction_read_only值将被初始化为default_transaction_read_only的值。
6	tx_read_o nly	设置事务的 访问模式。 tx_read_onl y和 transaction _read_only 是同义词。	M-Compatibility中只支持查询,不支持修改。
7	default_tr ansaction_ read_only	设置事务的 访问模式。	M-Compatibility中通过SET设置会改变会话级事务访问模式;MySQL中不支持该系统参数。

2.3.8 SQL

GaussDB数据库兼容绝大多数MySQL语法,但存在部分差异。本章节介绍GaussDB数据库当前支持的MySQL语法。

● 部分关键字在MySQL可以做标识符但M-Compatibility不可以或存在限制,以下为表2-51。

表 2-51 限制做标识符列表

关键字类型	关键字	约束
保留(可以是类型或函数)	COLLATION COMPACT	除函数和变量,不可以 作为其他数据库标识 符。
非保留(不能是类型或 函数)	BIT、BOOLEAN、 COALESCE、DATE、 NATIONAL、NCHAR、 NONE、NUMBER、 TEXT、TIME、 TIMESTAMP、 TIMESTAMPDIFF	不可以作为函数或变量 的标识符。

关键字类型	关键字	约束
保留	ANY、ARRAY、 BUCKETS、DO、END、 LESS、MODIFY、 OFFSET、ONLY、 RETURNING、SOME、 USER	不可以作为任意数据库 标识符。

GaussDB优化器与MySQL的优化器存在差异,由于优化器生成的执行计划的差异,可能导致GaussDB行为与MySQL行为的不一致,不影响业务数据结果。
 例如以下场景,GaussDB在计算col1以及使用col1进行where比较时,均会调用

cast函数,产生两条WARNING记录。

MySQL在计算col1时会调用cast函数,在进行where比较时,使用计算好的值直接 进行比较,因而产生一条WARNING记录。

```
- GaussDB行为:
m_db=# select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
WARNING: Truncated incorrect DECIMAL value: 'abc'
WARNING: Truncated incorrect DECIMAL value: 'abc'
CONTEXT: referenced column: col1
col1
 0
(1 row)
m_db=# explain verbose select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
WARNING: Truncated incorrect DECIMAL value: 'abc'
WARNING: Truncated incorrect DECIMAL value: 'abc'
CONTEXT: referenced column: col1
        QUERY PLAN
Result (cost=0.00..0.01 rows=1 width=0)
 Output: 0::decimal
(2 rows)
-- MySQL行为:
mysql> select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
| col1 |
0 |
1 row in set, 1 warning (0.00 sec)
mysql> show warnings;
+-----
| Level | Code | Message |
+-----+
| Warning | 1292 | Truncated incorrect DECIMAL value: 'abc' |
1 row in set (0.00 sec)
mysql> explain select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
+----+
id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered |
Extra |
      | 1 | PRIMARY | <derived2> | NULL | system | NULL | NULL | NULL | NULL | 1 |
NULL | No tables used |
```

2.3.8.1 关键字

约束差异:

- 当关键字在M-Compatibility模式下为保留关键字,在MySQL中为非保留关键字, 其差异为:在M-Compatibility模式下不可作为表名、列名、列别名、AS列别名、 AS表别名、表别名、函数名和变量名,在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字,在MySQL中为保留关键字, 其差异为:在M-Compatibility模式下可作为表名、列名、列别名、AS列别名、AS 表别名、表别名、函数名和变量名,在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字(可以是函数或类型),在 MySQL中为保留关键字,其差异为:在M-Compatibility模式下可作为列别名、 AS列别名、函数名和变量名,在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字(可以是函数或类型),在
 MySQL中为非保留关键字,其差异为:在M-Compatibility模式下不可作为表名、
 列名、AS表别名和表别名,在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字(不能是函数或类型),在 MySQL中为保留关键字,其差异为:在M-Compatibility模式下可作为表名、列 名、列别名、AS列别名、AS表别名、表别名和变量名,在MySQL中不支持。
- 当关键字在M-Compatibility模式下为非保留关键字(不能是函数或类型),在 MySQL中为非保留关键字,其差异为:在M-Compatibility模式下不可作为函数 名,在MySQL中支持。

□ 说明

在M-Compatibility模式下的非保留关键字、保留关键字(可以是函数或类型)以及非保留 关键字(不能是函数或类型)之中,以下关键字不能作为列别名进行使用:

BETWEEN, BIGINT, BLOB, CHAR, CHARACTER, CROSS, DEC, DECIMAL, DIV, DOUBLE, EXISTS, FLOAT, FLOAT4, FLOAT8, GROUPING, INNER, INOUT, INT, INT1, INT2, INT3, INT4, INT8, INTEGER, JOIN, LEFT, LIKE, LONGBLOB, LONGTEXT, MEDIUMBLOB, MEDIUMINT, MEDIUMTEXT, MOD, NATURAL, NUMERIC, OUT, OUTER, PRECISION, REAL, RIGHT, ROW, ROW_NUMBER, SIGNED, SMALLINT, SOUNDS, TINYBLOB, TINYINT, TINYTEXT, VALUES, VARCHAR, VARYING, WITHOUT

其中,SIGNED和WITHOUT在MySQL中可以作为列别名进行使用。

2.3.8.2 标识符

M-Compatibility模式下标识符存在以下差异:

GaussDB无引号标识符中不支持以美元符号(\$)开头, MySQL无引号标识符中支持。

- GaussDB无引号标识符中的支持大小写敏感的数据库对象。
- GaussDB标识符支持U+0080~U+00FF扩展字符,MySQL标识符支持U+0080~U+FFFF的扩展字符。
- 无引号标识符中,GaussDB不支持创建以数字开头包含一个e或E结尾作为标识符的表,例如:

```
-- GaussDB报错不支持,MySQL支持
m_db=# CREATE TABLE 23e(c1 int);
ERROR: syntax error at or near "23"
LINE 1: CREATE TABLE 23e(c1 int);

m_db=# CREATE TABLE t1(23E int);
ERROR: syntax error at or near "23"
LINE 1: CREATE TABLE t1(23E int);
```

 有引号标识符中,GaussDB对于创建了列名为纯数字或科学计算法的表,不支持 直接使用,需要在引号中使用;对于点操作符(.)场景,列名为纯数字或科学计 算法的表也需要在引号中使用。例如:

```
-- 创建列名为纯数字或科学计算法的表
m_db=# CREATE TABLE t1(`123` int, `1e3` int, `1e` int);
CREATE TABLE
-- 向表中插入数据
m_db=# INSERT INTO t1 VALUES(7, 8, 9);
-- 结果非预期,但与MySQL结果一致
m_db=# SELECT 123 FROM t1;
?column?
  123
(1 row)
-- 结果非预期,但与MySQL结果一致
m_db=# SELECT 1e3 FROM t1;
?column?
  1000
(1 row)
-- 结果非预期,并且与MySQL结果不一致
m db=# SELECT 1e FROM t1;
e
1
(1 row)
-- 正确用法
m_db=# SELECT `123` FROM t1;
123
 7
(1 row)
m_db=# SELECT `1e3` FROM t1;
1e3
(1 row)
m_db=# SELECT `1e` FROM t1;
1e
9
(1 row)
-- 点操作符的场景, GaussDB不支持, MySQL支持
```

```
m_db=# SELECT t1.123 FROM t1;
ERROR: syntax error at or near ".123"
LINE 1: SELECT t1.123 FROM t1;
m_db=# SELECT t1.1e3 FROM t1;
ERROR: syntax error at or near "1e3"
LINE 1: SELECT t1.1e3 FROM t1;
m_db=# SELECT t1.1e FROM t1;
ERROR: syntax error at or near "1"
LINE 1: SELECT t1.1e FROM t1;
-- 点操作符的场景,正确用法:
m_db=# SELECT t1.`123` FROM t1;
123
7
(1 row)
m_db=# SELECT t1.`1e3` FROM t1;
1e3
 8
(1 row)
m_db=# SELECT t1.`1e` FROM t1;
1e
----
9
(1 row)
m_db=# DROP TABLE t1;
DROP TABLE
```

● GaussDB分区名使用双引号(需要设置SQL_MODE为ANSI_QUOTES)或反引号 是区分大小写的,MySQL不区分。

2.3.8.3 DDL

表 2-52 DDL 语法兼容介绍

概述	详细语法说明	差异
建表和修改表时支持创建主键、UNIQUE索引、外键约束	ALTER TABLE、CREATE TABLE	 在GaussDB中,当约束关联的表为ustore,且SQL语句中指定为using btree时,底层会建立为ubtree。 在GaussDB中,允许将外键作为分区键。 索引名、约束名、key名GaussDB是SCHEMA下唯一,MySQL是表下唯一。

概述	详细语法说明	差异
支持自增列	ALTER TABLE、CREATE TABLE	● GaussDB的自动增长列建议 为索引的第一个字段,否则 建表时产生警告,MySQL 自动增长列必须为索引第一 个字段,否则建表时会报 错。GaussDB含有自动增长 列的表进行某些操作时会产 生错误,例如:ALTER TABLE EXCHANGE PARTITION。
		● GaussDB的 AUTO_INCREMENT = value语法,value必须为小 于2^127的正数。MySQL可 以为0,GaussDB不可以。
		● GaussDB中当自增值已经达到字段数据类型的最大值时,继续自增将产生错误。 MySQL有些场景产生错误或警告,有些场景仍自增为最大值。
		● 不支持 innodb_autoinc_lock_mod e系统变量,GaussDB的 GUC参数 auto_increment_cache=0 时,批量插入自动增长列的 行为与MySQL系统变量 innodb_autoing_lock_mod
		innodb_autoinc_lock_mod e=1相似。 GaussDB的自动增长列在导入数据或者进行Batch Insert执行计划的插入操作时,对于混合0、NULL和确定值的场景,如果产生错误,后续插入自增值不一定与MySQL完全一致。
		- 提供 auto_increment_cache 参数,可以控制预留自 增值的数量。
		GaussDB的并行导入或插入自动增长列触发自增时,每个并行线程预留的缓存值也只在其线程中使用,未完全使用完毕的话,也会出现表中自动增长列的值不连续的情况。并行插入产生的自增

概述	详细语法说明	差	异
			值结果无法保证与MySQL 完全一致。
		•	GaussDB的本地临时表中的自动增长列批量插入时不会预留自增值,正常场景不会产生不连续的自增值。 MySQL临时表与普通表中的自动增长列自增结果一致。
		•	GaussDB的SERIAL数据类型为原有的自增列,与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。
		•	GaussDB的不允许 auto_increment_offset的值 大于 auto_increment_increment 的值,会产生错误。 MySQL允许,并说明 auto_increment_offset会被 忽略。
		•	在表有主键或索引的情况下,ALTER TABLE命令重写表数据的顺序与MySQL不一定相同,GaussDB按表数据存储顺序重写,MySQL会按主键或索引顺序重写,导致自增值的顺序可能不同。
		•	GaussDB的ALTER TABLE命 令添加或修改自增列时,第 一次预留自增值的数量是表 统计信息中的行数,统计信 息的行数不一定与MySQL 一致。
		•	GaussDB的last_insert_id函 数返回值为128位的整型。
		•	GaussDB在触发器或用户自 定义函数中自增时,刷新 last_insert_id返回值。 MySQL不刷新。
		•	GaussDB的对GUC参数 auto_increment_offset和 auto_increment_increment

概述	详细语法说明	差异
		设置超出范围的值会产生错 误。MySQL会自动改为边 界值。
支持前缀索引	CREATE INDEX、ALTER TABLE、CREATE TABLE	• GaussDB中前缀长度不得超过2676,键值的实际长度受内部页面限制,若字段中含有多字节字符或者一个索引上有多个键,索引行长度可能会超限报错。
		GaussDB中主键索引中不支 持前缀键,创建或添加主键 时不支持指定前缀长度。
支持指定字符集与排序 规则	ALTER SCHEMA、ALTER TABLE、CREATE SCHEMA、CREATE TABLE	• 指定库级字符集时,除 BINARY字符集外,暂不支 持创建新库/模式的字符集 与数据库的 server_encoding不同。
		● 指定表级、列级字符集和字符序时,MySQL支持指定与库级字符集、字符序不同的字符集和字符序。在GaussDB中,表级、列级字符集和字符序仅支持BINARY字符集、字符序或者与库级字符集、字符序相同的字符集、字符序。
修改表时支持在表第一 列前面或者在指定列后 面添加列	ALTER TABLE	-
修改列名称/定义语法	ALTER TABLE	暂不支持DROP INDEX DROP KEY ORDER BY子项。

概述	详细语法说明	差异
创建分区表语法	CREATE TABLE PARTITION	MySQL在以下场景支持表 达式,不支持多个分区键:
		- 使用LIST分区/RANGE分 区策略,不指定 COLUMNS关键字。
		- 使用HASH分区策略。
		MySQL在以下场景不支持 表达式,支持多个分区键:
		- 使用LIST分区/RANGE分 区策略,指定COLUMNS 关键字。
		- 使用KEY分区策略。
		GaussDB不支持使用表达式 作为分区键,不支持指定二 级分区。
		GaussDB仅在以下场景支持 使用多个分区键:使用LIST 分区/RANGE分区策略。
		● GaussDB分区表不支持用生成列作为分区键。
建表和修改表时支持指 定表级和列级 comment	CREATE TABLE、ALTER TABLE	-
创建索引时支持指定索 引级comment	CREATE INDEX	-

概述	详细语法说明	差异
交换普通表和分区表分区的数据	详细语法说明 ALTER TABLE PARTITION	差异 ALTER TABLE EXCHANGE PARTITION的差异点: • 对于自增列,MySQL执行 alter exchange partition 后,自增列则控重证置。 GaussDB则不会被自增。 • MySQL表或分区使用 tablespace时,数区区则据区使无的度和进换。 • MySQL表或分区例明据区域的,数区区别超过的区域的,数区区,对方区域的,对方面域的,对方区域的,对方区域的,对方区域的,对方面域的,对方面域的,可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以
		分区表的被删除列严格对齐 才能进行分区和普通表数据

概述	详细语法说明	差异
修改分区表的分区键信 息	ALTER TABLE	MySQL支持修改分区表的分区 键信息,GaussDB中不支持。
支持CREATE TABLE LIKE语法	CREATE TABLE LIKE	 在MySQL 8.0.16 之前的版本中,CHECK约束会被高法解析但功能会被忽束,GaussDB支持复制CHECK约束。 对于主键约束名称,在建表时,MySQL所有主键约束名称的固定为PRIMARY KEY,GaussDB不支持复制。 对于唯一键约束名称,在建表的大区,GaussDB不支持复制。 对于叶上区K约束名称,在建表的大区中区K约束名称,在建表的大区中区K约束名称,在建表的版本无CHECK约束名称,之前的版本无CHECK约束后息,GaussDB支持复制。 对于CHECK约束名称,之前的版本无CHECK约束后息,GaussDB支持复制,在建表的版本无CHECK约束后息,GaussDB交持复制。 在跨sql_mode模式建表的,MySQL支持复制。 在跨sql_mode模式建表的,MySQL受宽松模式控制,GaussDB可能存在严格模式失效的情况。 源表存在默认值
		"0000-00-00",在 "no_zero_date"严格模式 下,GaussDB建表成功,且 包含默认值 "0000-00-00",严格模 式失效;而MySQL建表失 败,受严格模式控制。
支持增加子分区语法	ALTER TABLE [IF EXISTS] { table_name [*] ONLY table_name ONLY (table_name)} add_clause; add_clause: ADD {{partition_less_than_item partition_start_end_item partition_list_item} PARTITION({partition_less_than _item partition_start_end_item partition_list_item})}	保留原分区表语法。 不支持下述语法添加多分区: ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition1,…); 仅支持原有添加多分区语法: ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2[y1]),…;

概述	详细语法说明	差异
TRUNCATE子分区语法	ALTER TABLE [IF EXISTS] table_name truncate_clause;	支持子项有差异,对于 truncate_clause: M-Compatibility模式: TRUNCATE PARTITION { { ALL partition_name [,] } FOR (partition_value [,]) } [UPDATE GLOBAL INDEX] MySQL支持: TRUNCATE PARTITION {partition_names ALL}
主键索引名	CREATE TABLE table_name (col_definitine ,PRIMARY KEY [index_name] [USING method] ({ column_name (expression) }[ASC DESC] } [,]) index_parameters [USING method COMMENT 'string'])	GaussDB中的主键指定索引名 后创建的索引名为用户所指定 的索引名,MySQL索引名为 PRIMARY。
删除有依赖的对象	DROP drop_type name CASCADE;	在GaussDB中,删除有依赖的 对象需要加CASCADE, MySQL不需要。
NOT NULL约束不允许 插入NULL值	CREATE TABLE t1 (id int NOT NULL DEFAULT 8); INSERT INTO t1 VALUES(NULL); INSERT INTO t1 VALUES(1), (NULL),(2);	在MySQL宽松模式下,会将 NULL进行类型转换,并成功插 入数据;在MySQL严格模式下 不允许插入NULL值。在 GaussDB不支持此特性,在宽 松模式和严格模式下均不允许 插入NULL值。
CHECK约束生效	CREATE TABLE	CREATE TABLE带CHECK约束的时候,MySQL8.0会生效,MySQL5.7只解析语法但不生效。GaussDB在此功能上同步MySQL8.0版本,且GaussDBCHECK约束可以引用其他列,而MySQL不能。 GaussDB一个表中最多只能加32767个CHECK约束。
索引的algorithm和 lock选项不起作用	CREATE INDEX DROP INDEX	M-Compatibility模式的 CREATE/DROP INDEX语句中 INDEX选项algorithm_option 和lock_option目前只在语法上 支持,创建时不报错,但实际 不起作用。

概述	详细语法说明	差异
CREATE TABLE hash 分区和二级分区的存储 与MySQL不同	CREATE TABLE	GaussDB的CREATE TABLE语句中hash分区表和二级分区表所使用的hash函数与MySQL不一致,因此hash分区表和二级分区表的存储与MySQL有区别。
分区表索引	CREATE INDEX	GaussDB的分区表索引分为LOCAL和GLOBAL两种。LOCAL索引与某个具体分区绑定,而GLOBAL索引则对应整个分区表。LOCAL和GLOBAL索引的创建方法和默认规则具体说明参见《开发指南》中"SQL语法》与C CREATE INDEX"章节,例如:在非分区键上创建地一索引,会默认创建为GLOBAL索引的概念。在GaussDB中,当分区进行DROP、TRUNCATE、EXCHANGE等有别,以更新GLOBAL索引,以更新GLOBAL索引,以更新GLOBAL索引,以更新GLOBAL索引,以现用户在使用分区操作不会默认更新GLOBAL INDEX字句,或配置全局GUC参数enable_gpi_auto_update为true(推荐),使得在进行分区操作时自定更新GLOBAL索引。
CREATE/ALTER TABLE 语句中分区表为KEY分 区,不支持指定 algorithm。部分分区 定义入参不支持表达 式。	CREATE TABLE、ALTER TABLE	GaussDB的CREATE/ALTER TABLE语句中分区表为KEY分区,不支持指定algorithm。不支持表达式入参的语法: PARTITION BY HASH() PARTITION BY KEY() VALUES LESS THAN()
分区表不支持 LINEAR/KEY hash	CREATE TABLE PARTITION	GaussDB分区表不支持 LINEAR/KEY hash。

概述	详细语法说明	差异
check和 auto_increment语法不 能作用在同一字段	CREATE TABLE	由于MySQL5.7的check字段不 生效,check和 auto_increment同时作用于同 一字段只有auto_increment生 效,但GaussDB报错。
删除存在依赖关系的表	DROP TABLE	GaussDB删除存在依赖的表必 须加上CASCADE才能成功, MySQL不需要。
表定义相关选项	CREATE TABLE \ ALTER TABLE	● GaussDB不支持以下选项: AVG_ROW_LENGTH、 CHECKSUM、 COMPRESSION、 CONNECTION、DATA DIRECTORY、INDEX DIRECTORY、 DELAY_KEY_WRITE、 ENCRYPTION、 INSERT_METHOD、 KEY_BLOCK_SIZE、 MAX_ROWS、 MIN_ROWS、 PACK_KEYS、 PASSWORD、 STATS_AUTO_RECALC、 STATS_PERSISTENT、 STATS_SAMPLE_PAGES。 ● 以下选项在GaussDB中不报 错,但实际上也不生效: ENGINE、 ROW_FORMAT。
CMK密钥轮转,轮换 加密COLUMN ENCRYPTION KEY的 CLIENT MASTER KEY,对COLUMN ENCRYPTION KEY明文 进行重加密。	ALTER COLUMN ENCRYPTION KEY	M-Compatibility模式不支持全 密态。故不支持该语法。
密态等值查询特性使用 多级加密模型,主密钥 加密列密钥,列密钥加 密数据。本语法用于创 建主密钥对象。	CREATE CLIENT MASTER KEY	M-Compatibility模式不支持全 密态。故不支持该语法。
创建一个列加密密钥, 该密钥可用于加密表中 的指定列。	CREATE COLUMN ENCRYPTION KEY	M-Compatibility模式不支持全 密态。故不支持该语法。

概述	详细语法说明	差异
全密态功能,传输密钥 到服务端缓存,只在开 启内存解密逃生通道的 情况下使用。	\send_token	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,传输密钥 到服务端缓存,只在开 启内存解密逃生通道的 情况下使用。	\st	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,销毁服务 端缓存的密钥,只在开 启内存解密逃生通道的 情况下使用。	\clear_token	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,销毁服务 端缓存的密钥,只在开 启内存解密逃生通道的 情况下使用。	\ct	M-Compatibility模式不支持全 密态。故不支持该语法。
在全密态数据库特性 中,用于设置访问外部 密钥管理者的参数。	\key_info KEY_INFO	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,用于开启 三方动态库功能与加载 三方动态库时的参数设 置。	\crypto_module_info MODULE_INFO	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,用于开启 三方动态库功能与加载 三方动态库时的参数设 置。	\cmi MODULE_INFO	M-Compatibility模式不支持全 密态。故不支持该语法。
generated always as 语句不能再引用由 generated always as 生成的列。	Generated Always AS	GaussDB generated always as语句不能再引用由generated always as生成的列,MySQL 可以。
支持更改表名语法	ALTER TABLE tbl_name RENAME [TO AS =] new_tbl_name;	GaussDB的ALTER RENAME语 法仅支持修改表名称功能操 作,不能耦合其它功能操作。
		GaussDB仅旧表名字段支持如 schema.table_name用法;且 新表名与旧表名将属于同一 Schema下。
		GaussDB不支持新旧表跨 Schema重命名操作;但如有权 限,则可在当前Schema下修改 其他Schema下表名称。

概述 详细语法说明 差异		差异
禁用GUC参数 enable_expr_fusion	SET enable_expr_fusion= ON	M-Compatibility模式暂不支持 GUC参数enable_expr_fusion 打开。

概述	详细语法说明	差异
支持CREATE VIEW AS SELECT语法	CREATE VIEW table_name AS query;	● 针对以下类型,不支持CREATE VIEW view_name AS query语法中query包含计算操作(如函数调用、使用操作符计算)。 - BINARY[(n)] - BOOLEAN/BOOL - VARBINARY(n) - CHAR[(n)] - VARCHAR(n) - TIME[(p)] - DATETIME[(p)] - DIMERIC[(p[,s])] - DEC[(p[,s])] - DEC[(p[,s])] - FLOAT4[(p, s)] - FLOAT8[(p,s)] - FLOAT[(p)] - REAL[(p, s)] - PLOAT[(p, s)] - DOUBLE[(p,s)] - DOUBLE[(p,s)] - DOUBLE PRECISION[(p,s)] - TINYTEXT - MEDIUMTEXT - LONGTEXT - BLOB - TINYBLOB - MEDIUMBLOB - LONGBLOB - MED

概述	详细语法说明	差异
		VALUES(8000); INSERT 0 1 m_db=# CREATE VIEW view1 AS SELECT salary/10 as te FROM TEST; ERROR: Unsupported type numeric used with expression in CREATE VIEW statement. m_db=# CREATE TABLE TEST (salary int(10)); CREATE TABLE m_db=# INSERT INTO TEST VALUES(8000); INSERT 0 1 m_db=# CREATE VIEW view2 AS SELECT sec_to_time(salary) as te FROM TEST; ERROR: Unsupported type time used with expression in CREATE VIEW statement. • 在query为复合查询,子查 询等非简单查询场景下, M-Compatibility模式针对 上述类型的计算操作与 MySQL存在差异,M- Compatibility模式下新创建 表的数据类型列精度属性不 保留。
索引名可重名范围	CREATE TABLE CREATE INDEX	MySQL中索引名在一个表下唯一,在不同的表下可以有相同的索引名。M-Compatibility模式中的索引名在同一个SCHEMA下唯一,在同一的SCHEMA下不可用相同的索引名。在M-Compatibility模式下,针对会自动创建索引的约束和key,也会有相同的规则。

概述	详细语法说明	差异
视图依赖差异	CREATE VIEW、ALTER TABLE	MySQL中视图存储,只记录名、知图存储,只记录名、列名、数据中视图存储,数据中视图存储,数据中心表的表记录目标表的表记录目标表的表记录目标。例如是一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个
		改,且后续可以查询该视 图。
外键差异	CREATE TABLE	GaussDB外键约束对类型不敏感,如果主表和从表对应的字段数据类型存在隐式类型转换就可以建成。MySQL外键类型敏感。如果两个表对应的列类型不同外键无法建成。
		COLUMN或CHANGE COLUMN方式修改表列外键所 在列的数据类型或列名等, GaussDB可以。
索引升降序差异	CREATE INDEX	在MySQL 5.7中,ASC DESC 被解析但是被忽略,默认行为 为ASC;在MySQL 8.0及 GaussDB中,ASC DESC被解 析且生效。

概述	详细语法说明	差异
指定字段的默认值	CREATE TABLE、ALTER TABLE	● MySQL5.7指定默认值时, 仅支持不带括号形式的默认 值。MySQL8.0与GaussDB 支持带括号形式的默认值。 GaussDB m_db=# DROP TABLE IF EXISTS t1, t2; DROP TABLE m_db=# CREATE TABLE t1(a DATETIME DEFAULT NOW()); CREATE TABLE m_db=# CREATE TABLE t2(a DATETIME DEFAULT (NOW())); CREATE TABLE MySQL5.7 mysql> DROP TABLE IF EXISTS t1, t2;
		Query OK, 0 rows affected (0.04 sec)
		mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW()); Query OK, 0 rows affected (0.04 sec)
		mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW())); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(NOW()))' at line 1
		MySQL8.0 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.17 sec)
		mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW()); Query OK, 0 rows affected (0.19 sec)
		mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW())); Query OK, 0 rows affected (0.20 sec)
		 MySQL给BLOB、TEXT、 JSON数据类型指定默认值 时必须给默认值添加括号, GaussDB给上述数据类型指 定默认值时可以不添加括 号。
		GaussDB指定默认值时不会校验默认值是否溢出; MySQL指定不带括号形式的默认值时会校验是否溢

概述	详细语法说明	差异
		出,指定带括号形式的默认 值时不会校验是否溢出。
		GaussDB支持使用DATE/TIME/TIMESTAMP开头的时间常量给字段指定默认值,MySQL使用DATE/TIME/TIMESTAMP开头的时间常量给字段指定默认值时必须在默认值外添加括号。
		GaussDB m_db=# DROP TABLE IF EXISTS t1, t2; DROP TABLE m_db=# CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); CREATE TABLE m_db=# CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); CREATE TABLE
		MySQL5.7 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.02 sec)
		mysql> CREATE TABLE t1 (a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); ERROR 1067 (42000): Invalid default value for 'a' mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(TIMESTAMP '2000-01-01 00:00:00'))' at line 1
		MySQL8.0 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.14 sec)
		mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); ERROR 1067 (42000): Invalid default value for 'a' mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); Query OK, 0 rows affected (0.19 sec)

2.3.8.4 DML

表 2-53 DML 语法兼容介绍

序号	概述	详细语 法说明	差异
1	DELETE支持从指定分区 (或子分区)删除数据	DELETE	-
2	UPDATE支持ORDER BY 和LIMIT	UPDATE	-
3	SELECT INTO语法	SELECT	GaussDB可以使用SELECT INTO根据查询结果创建一个新表,MySQL不支持。
			GaussDB的SELECT INTO语法不支 持将多个查询进行集合运算后的结 果作为查询结果。

序号	概述	详细语 法说明	差异
4	REPLACE INTO语法	REPLAC	● MySQL不受严格模式和宽松模式的影响,可向表中插入时间0值,即: mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, f2, f3); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; ++ f1
5	SELECT支持指定多分区 查询	SELECT	-
6	UPDATE支持指定多分区 更新	UPDATE	-

序号	概述	详细语 法说明	差异
7	LOAD DATA导入数据功能	LOAD DATA	在使用LOAD DATA导入数据功能 时,GaussDB与MySQL相比有如下差 异:
			LOAD DATA语法执行结果与M*严格模式一致,宽松模式暂未适配。
			 IGNORE与LOCAL参数功能仅为当 导入数据与表中数据存在冲突时, 忽略当前冲突行数据功能和当文件 中字段数小于指定表中列数时自动 为其余列填充默认值功能,其余功 能暂未适配。
			 指定LOCAL关键字,且文件路径 为相对路径时,文件从二进制目录 下搜索;不指定LOCAL关键字, 且文件路径为相对路径时,文件从 数据目录下搜索。
			● LOAD DATA仅支持从服务端导入 文件。
			● [(col_name_or_user_var [, col_name_or_user_var])]指定 列参数不支持重复指定列。
			 [FIELDS TERMINATED BY 'string']指定换行符不能与[LINES TERMINATED BY 'string']分隔符相同。
			执行LOAD DATA语法写入表中的 数据若无法转换为表中数据类型格 式时报错。
			LOAD DATA SET表达式中不支持 指定列名计算。
			● LOAD DATA只能用于表,不能用于视图。
			 windows下的文件与linux环境下 文件默认换行符存在差异,LOAD DATA无法识别此场景会报错,建 议用户导入时检查导入文件行尾换 行符。

序号	概述	详细语 法说明	差异
8	INSERT支持VALUES引用 列语法	INSERT INTO tabnam e VALUE S(1,2,3) ON DUPLIC ATE KEY UPDATE b = VALUE S(colum n_name)	GaussDB的ON DUPLICATE KEY UPDATE子句中的VALUES()不支持表 名.列名格式,MySQL支持。
9	LIMIT限制差异	DELETE SELECT UPDATE	各个语句的limit子项与MySQL的limit 项当前存在差异。 GaussDB中limit参数最大值为BIG INT类型限制(超过 9223372036854775807报错)。在 MySQL中,limit最大值为unsigned LONGLONG类型限制(超过 18446744073709551615 报错)。 limit可以设置小数值,实际执行时四 舍五入。MySQL不能取小数。 GaussDB的delete语句中,不允许 limit 0。MySQL在delete语句中允许 limit 0。

序号	概述	详细语 法说明	差异
10	反斜杠(\)用法差异	INSERT	反斜杠(\)的用法在GaussDB和 MySQL中都可以由参数控制但当前默 认用法不同:
			MySQL中使用参数 NO_BACKSLASH_ESCAPES控制字符 串和标识符中的反斜杠(\)被解析为普通字符还是转义字符,默认反斜杠字符(\)作为字符串和标识符中的转义字符。设置set sql_mode='NO_BACKSLASH_ESCAPES';可以禁用反斜杠字符(\)作为字符串和标识符中的转义字符。
			GaussDB中使用参数 standard_conforming_strings控制字符串和标识符中的反斜杠(\)被解析为普通字符还是转义字符。默认值为on,在普通字符串文本中按照SQL标准把反斜杠(\)当普通文本。使用setstandard_conforming_strings=off;将反斜杠字符(\)作为字符串和标识符中的转义字符。
11	插入值少于字段数目 时,MySQL报错, GaussDB补充空值。	INSERT	GaussDB不指定列的列表时,如果插入值少于字段数目,默认按建表时的字段顺序赋值。字段上有非空约束时报错,没有非空约束时,如果指定了默认值则缺省部分补充默认值,若未指定默认值则补充空。
12	ORDER BY中排序的列必须包括在结果集的列中。	SELECT	在GaussDB中,在与GROUP BY子句一起使用的情况下,ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。在与DISTINCT关键字一起使用的情况下,ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。
13	不允许对约束字段用 ON DUPLICATE KEY UPDATE 进行修改。	INSERT	-
14	SELECT结果允许存在重 复列名。	SELECT	-
15	NATURAL JOIN与 MySQL有差异。	SELECT	在GaussDB中,NATURAL [[LEFT RIGHT] OUTER] JOIN允许不指定 LEFT RIGHT,不指定时NATURAL OUTER JOIN为NATURAL JOIN。允 许连续使用多次JOIN。

序号	概述	详细语 法说明	差异
16	外键数据类型是 timestamp/datetime 时,update/delete外表 报错。	UPDATE /DELETE	外键数据类型是timestamp/datetime 时,update/delete外表报错, MySQL成功。
17	nature join和using兼容。	SELECT	 GaussDB join的顺序严格按照从左往右,MySQL可能会调整顺序。 GaussDB和MySQL在natural join与using时均不允许左表或右表参与join的字段出现歧义(一般由左或右临时表中重名字段造成)。因为两者join的顺序有差别,故行为上可能有差别。 GaussDB的行为:
			NATURAL JOIN t3; ERROR 1052 (23000): Column 'a' in from clause is ambiguous
18	with clause兼容 MySQL8.0版本	SELECT	-
	, 5 2 20.0111.7.7.	INSERT	
		UPDATE	
		DELETE	
19	join兼容	SELECT	GaussDB join不支持使用逗号","的 连接方式,MySQL支持。 GaussDB不支持use index for join。

序号	概述	详细语 法说明	差异
20	SELECT语句中列表达式 为函数表达式、算数表 达式等情形时,查询结 果显示的列名为? column?	SELECT	GaussDB SELECT语句中列表达式为 函数表达式、算数表达式等情形时, 查询结果显示的列名为 ?column?, MySQL为对应表达式。
21	SELECT导出文件(into outfile)	SELECT . INTO OUFILE 	SELECT INTO OUTFILE语法,导出文件中FLOAT、DOUBLE、REAL类型的值显示精度和MySQL存在差异,不影响COPY导入和导入后的值。
22	UPDATE/INSERT/ REPLACE SET指定模 式名、表名	UPDATE / INSERT/ REPLAC E SET	 SELECT语句指定投影列时,MySQL支持模式名.表别名.列名的三段式用法,GaussDB不支持。m_db=# CREATE SCHEMA test; CREATE SCHEMA m_db=# CREATE TABLE test.t1(a int); CREATE TABLE m_db=# SELECT test.alias1.a FROM t1 alias1; ERROR: invalid reference to FROM-clause entry for table "alias1" LINE 1: select test.alias1.a from t1 alias1;
23	UPDATE SET执行顺序与 MySQL存在差异	UPDATE SET	MySQL中,UPDATE SET的顺序是从前往后依次UPDATE,前面UPDATE 的结果会影响后面的结果,且允许多次设置同一列;GaussDB中为先取出原来的所有相关的数据,再一次性UPDATE,且不允许多次设置同一列,二者存在差异。

序号	概述	详细语 法说明	差异
24	IGNORE特性	UPDATE / DELETE/ INSERT	MySQL数据库和GaussDB执行过程的 差异,因此产生的WARNING条数和 WARNING信息可能存在不同。
25	SHOW COLUMNS语法	SHOW	 ● 用户权限验证与MySQL存在差异。 - GaussDB中需要拥有指定表所在Schema的USAGE权限,同时还需要拥有指定表的任意表级权限或列级权限,仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 - MySQL中需要拥有指定表的任意表级权限或列级权限,仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 ● LIKE和WHERE子句中涉及到字符串比较操作时,排序规则与MySQL存在差异。 - MySQL中使用uf8_general_ci。 - GaussDB中排序规则使用当前客户端的collation_connection。GaussDB中建议用户在WHERE子句中,不要对返回字能会出现非预期的报错。 - 预期报错m_db=# SHOW FULL COLUMNS FROM to2 WHERE 'b'='pri'; ERROR: Column'b' does not exist. LINE 1: SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; ERROR: input of anonymous composite types is not implemented LINE 1: SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; - 非预期报错m_db=# SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; - 非预期报错m_db=# SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; - 非预期报错m_db=# SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; - 非预期报错m_db=# SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; - 非预期报错m_db=# SHOW FULL COLUMNS FROM to2 WHERE 'c'='pri'; - ** ** ** ** ** ** ** ** ** ** ** ** **

序号	概述	详细语 法说明	差异
26	SHOW CREATE DATABASE语法	SHOW	用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定Schema的USAGE权限。 MySQL中需要拥有任意库级权限(除GRANT OPTION和USAGE)、任意表级权限(除GRANT OPTION)或任意列级权限。
27	SHOW CREATE TABLE 语法	SHOW	 ● 用户权限验证与MySQL存在差异。 - GaussDB中需要拥有指定表所在Schema的USAGE权限和指定表的任意表级权限。 - MySQL中需要拥有指定表的任意表级权限(除GRANTOPTION)。 ● 返回的建表语句与MySQL存在差异。 - GaussDB中索引以CREATEINDEX语句的形式返回。MySQL中表的索引在CREATETABLE语句中返回。主要因为GaussDB中CREATEINDEX语法支持的可选参数范围与CREATETABLE语法中创建索引不同,因此某些索引不固定。 - GaussDB中CREATETABLE语法中创建。 - GaussDB中CREATETABLE语法中创建。 - GaussDB中CREATETABLE语法中创建。 - GaussDB中CREATETABLE语法项个的型建表语句才是这种个不多显示。 ● 设置兼容性参数m_format_dev_version为's2'后,返回的建表语句才兼容MySQL。兼容的内容包括:列注释位置的中不多包括:列注释位置的中的中的中的以同时是对表的内容包括:列注释位置变更、未过程的中的中的以同时是对表的内容更多更多的内容更多更多的内容更多更多的内容更多更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容更多更多的内容的可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以

序号	概述	详细语 法说明	差异
28	SHOW CREATE VIEW语法	SHOW	 用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定视图所在Schema的USAGE权限和指定视图的任意表级权限。 MySQL中需要拥有指定视图的表级SELECT和表级SHOWVIEW权限。 返回的视图创建语句与MySQL存在差异。以SELECT*FROMtbl_name形式创建的视图,GaussDB中*不会被展开,而MySQL中会展开。 返回结果中的character_set_client和collation_connection字段与MySQL存在差异。 MySQL中显示视图创建时系统变量character_set_client和collation_connection的会话值 GaussDB中未记录相关元数据,显示为NULL。
29	SHOW PROCESSLIST语 法	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.processlist视图内字段内容与大小写保持一致,MySQL中可能存在差异。 GaussDB中用户只能访问自己的线程信息,拥有SYSADMIN权限的用户可以访问所有用户的线程信息。 MySQL中用户只能访问自己的线程信息,拥有PROCESS权限的用户可以访问所有用户的线程信息。
30	SHOW [STORAGE] ENGINES	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.engines视图内字段内容与大小写保持一致,MySQL中可能存在差异。因为MySQL与GaussDB的存储引擎不同,所以该指令查询的结果不同。

序号	概述	详细语 法说明	差异
31	SHOW [SESSION] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.session_status视图内字段内容与大小写保持一致,MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。
32	SHOW [GLOBAL] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.global_status视图内字段内容与大小写保持一致,MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。
33	SHOW INDEX	SHOW	 用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定SCHEMA的USAGE权限和指定表的任意表级权限或者任意列级权限。 MySQL中需要拥有指定表的任意表级权限(除GRANTOPTION)或者任意列级权限。 GaussDB中临时表存储于独立的临时Schema中,在使用FROM或INdb_name条件来展示指定临时表索引信息时,须指明db_name为临时表所在的Schema才能展示临时表索引信息,否则会提示不存在该临时表,这一点和Mysql在部分情况下存在差异。
34	SHOW SESSION VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与 information_schema.session_variables视图内字段内容及大小写保持一致,与MySQL可能存在差异。
35	SHOW GLOBAL VARIABLES	SHOW	GaussDB中查询结果中字段内容及大 小写与 information_schema.global_variabl es视图内字段内容及大小写保持一 致,与MySQL可能存在差异。

序号	概述	详细语 法说明	差异
36	SHOW CHARACTER SET	SHOW	GaussDB中查询结果中字段内容及大小写与 information_schema.character_sets 视图内字段内容及大小写保持一致, 与MySQL可能存在差异。
37	SHOW COLLATION	SHOW	GaussDB中查询结果中字段内容及大 小写与 information_schema.collations视图 内字段内容及大小写保持一致,与 MySQL可能存在差异。
38	EXCEPT语法	SELECT	-
39	SELECT支持 STRAIGHT_JOIN语法	SELECT	GaussDB中多表关联JOIN场景下生成的执行计划,与MySQL可能存在差异。
40	SHOW TABLES	SHOW	 LIKE行为存在差异,具体请参见操作符章节的"LIKE"。 WHERE表达式行为存在差异,具体行为请参见GaussDB数据库的"WHERE表达式"。 GaussDB中:表和数据库的权限需要分开赋予用户,查询的数据库必须是用户在SHOW SCHEMAS上可以查询到,不能仅仅有表的权限,必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。 GaussDB中:校验逻辑中优先校验Schema是否存在,后校验当前用户是否对Schema具有权限,与MySQL存在差异。

序号	概述	详细语 法说明	差异
41	SHOW TABLE STATUS	SHOW	 GaussDB中:该语法展示数据依赖information_schema下的tables视图。MySQL中tables指定的是表。
			GaussDB中: 表和数据库的权限需要分开赋予用户,查询的数据库必须是用户在SHOW SCHEMAS上可以查询到,不能仅仅有表的权限,必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。
			 GaussDB中:校验逻辑中优先校 验Schema是否存在,后校验当前 用户是否对Schema具有权限,与 MySQL存在差异。
42	HAVING语法	SELECT	GaussDB的HAVING必须且只能引用GROUP BY子句中的列或聚合函数中使用的列。MySQL支持对此行为的扩展,并允许HAVING引用列表中的SELECT列和外部子查询中的列。
43	SELECT后跟行表达式	SELECT	MySQL不支持SELECT后跟行表达式,GaussDB支持SELECT后跟行表达式。 MySQL的行为: mysql> SELECT row(1,2); ERROR 1241 (21000): Operand should contain 1 column(s)
			GaussDB的行为: m_db=# SELECT row(1,2); row(1,2) (1,2) (1 row)
44	SELECT FOR SHRAE/FOR UPDATE/ LOCK IN SHRAE MODE	SELECT	 GaussDB不支持FOR SHARE/FOR UPDATE/LOCK IN SHARE MODE 子句和UNION/EXCEPT/DISTINCT/GROUP BY/HAVING子句一起使用, MySQL 5.7部分支持(FOR SHARE/EXCEPT语法不支持), MySQL8.0均支持。
			当将锁子句与LEFT/RIGHT [OUTER] JOIN子句连用时,LEFT JOIN不支持给右表上锁,RIGHT JOIN不支持给左表上锁;MySQL 可以给JOIN两侧的表同时上锁。

序号	概述	详细语 法说明	差异
序号 45	概述 SELECT语法支持范围		● GaussDB指定from子句中的表别名时,支持带字段名称。 MySQL5.7不支持指定表别名时带字段名称。 — GaussDB — db=# DROP TABLE IF EXISTS t1; DROP TABLE M_db=# CREATE TABLE t1(a INT, b INT); CREATE TABLE M_db=# INSERT INTO t1 VALUES(1,2); INSERT 0 1 M_db=# SELECT * FROM t1 t2(a, b); a b+ 1 2 (1 row) M_db=# SELECT * FROM (SELECT * FROM t1) t2(a, b); a b+ 1 2 (1 row)
			Query OK, 0 rows affected (0.18 sec) mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.03 sec) mysql> SELECT * FROM t1 t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1

序号	概述	详细语 法说明	差异
			mysql> SELECT * FROM (SELECT * FROM t1) t2(a, b); ++ a b ++ 1 row in set (0.00 sec) • 当查询语句不带from子句时, GaussDB支持带where子句,与 MySQL8.0保持一致,MySQL5.7 不支持。 GaussDB m_db=# SELECT 1 WHERE true; 1 1 (1 row) MySQL5.7 mysql> SELECT 1 WHERE true; ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'where true' at line 1 MySQL8.0 mysql> SELECT 1 WHERE true; ++ 1 ++ 1 ++ 1 ++ 1 row in set (0.00 sec)

2.3.8.5 DCL

表 2-54 DCL 语法兼容介绍

序号	概述	详细语法说明	差异
1	SET NAMES指定 COLLATE子句	SET [SESSION LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] DEFAULT};	GaussDB中暂不支持指定 charset_name与数据库字 符集不同。具体请参考 《M-Compatibility开发指 南》中"SQL参考 > SQL 语法 > SQL语句 > S > SET"章节。

2.3.8.6 其他语句

表 2-55 其他语法兼容介绍

序号	概述	详细语法说明	差异
1	事务相关语法	数据库默认隔离级别	M-Compatibility默认隔离级别为READ COMMITTED,MySQL默认隔离级别为 REPEATABLE READ。 M-Compatibility隔离级别只有READ COMMITTED REPEATABLE READ生效。
2	事务相关语法	事务嵌套	M-Compatibility中嵌套事务不会自动提交,MySQL会自动提交。
3	事务相关语法	自动提交	M-Compatibility使用GaussDB存储,继承GaussDB事务机制,事务中执行DDL,DCL不会自动提交。MySQL在DDL、DCL、管理类语句,锁相关语句会自动提交。
4	事务相关语法	报错后需 rollback	M-Compatibility事务中报错,需要执行 rollback,MySQL无限制。
5	事务相关语法	锁机制	M-Compatibility锁机制只能在事务块中 使用,MySQL无限制。
6	锁机制	锁机制	 MySQL获取read锁后,当前会话无法进行写操作,M-Compatibility获取read锁后,当前会话可以进行写操作。 MySQL给表上锁后,读取其他表报错,M-Compatibility无限制。 MySQL同一会话中获取同一个表的锁,会自动释放上一个锁,并提交事务,M-Compatibility无该机制。 M-Compatibility中LOCK TABLE只能在一个事务块的内部有用,且无UNLOCK TABLE命令,锁总是在事务结束时释放。

序号	概述	详细语法说明	差异
7	PBE	PBE	• 重复创建同名的PREPARE语句,M-Compatibility会报已经存在的错误,需要先删除已有statement,MySQL会覆盖旧的statement。
			 M-Compatibility和MySQL在SQL语句 执行过程中对异常场景的报错阶段不 同,例如解析层、执行层等;而 PREPARE语句对预备语句只处理到解 析层。因此PBE下对于异常场景,报错 位置在PREPARE阶段还是EXECUTE阶 段,M-Compatibility和MySQL存在可 能差异。

2.3.8.7 用户与权限

概述

在M-Compatibility中,用户与权限管控相关的行为、语法整体沿用GaussDB的机制,暂不同步MySQL。

用户与权限的行为与GaussDB保持一致,具体行为说明请参见《开发指南》中的"数据库安全 > 用户及权限"章节。

用户与权限的语法在原有GaussDB的基础上,裁剪了部分语法,具体语法说明请参见《M-Compatibility开发指南》中的"SQL参考 > SQL语法 > SQL语句"章节。M-Compatibility与GaussDB的语法差异请参见表2-56。

M-Compatibility创建USER时会自动创建与USER同名的Schema,MySQL不创建。

表 2-56 M-Compatibility 与 GaussDB 的语法差异

序号	语法说明	概述	M-Compatibility与 GaussDB的差异
1	CREATE ROLE	创建一个角色。	在M-Compatibility中,
2	CREATE USER	创建一个用户。	不支持指定涉及以下关键 字的选项: ENCRYPTED、
3	CREATE GROUP	创建一个新用户组。 CREATE GROUP是 CREATE ROLE的别名,不 推荐使用。	UNENCRYPTED、 RESOURCE POOL、PERM SPACE、TEMP SPACE、 SPILL SPACE。
4	ALTER ROLE	修改角色属性。	
5	ALTER UER	修改用户属性。	
6	ALTER GROUP	修改一个用户组的属性。	-
7	DROP ROLE	删除角色。	-

序号	语法说明	概述	M-Compatibility与 GaussDB的差异
8	DROP USER	删除用户。	-
9	DROP GROUP	删除用户组。	-
10	DROP OWNED	删除一个数据库角色所拥 有的数据库对象。	-
11	REASSIGN OWNED	修改数据库对象的属主。	M-Compatibility中不支持 该语法。
12	GRANT	对角色和用户进行授权操 作。	M-Compatibility中不支持 授予或回收函数、存储过
13	REVOKE	用于撤销一个或多个角色 的权限。	程、表空间、DATABASE LINK等对象的权限。
14	ALTER DEFAULT PRIVILEGES	设置应用于将来创建的对象的权限(这不会影响分配到已有对象中的权限)。	M-Compatibility中不支持 该语法。

差异说明

● 语法格式差异

M-Compatibility的授权语法请参见《M-Compatibility开发指南》中的"SQL参考 > SQL语法 > SQL语句 > G > GRANT"章节,MySQL中的授权语法如下:

```
-- 全局级、数据库级、表级、存储过程级赋权语法
GRANT
  priv_type [(column_list)]
    [, priv_type [(column_list)]] ...
   ON [object_type] priv_level
   TO user [auth_option] [, user [auth_option]] ...
   [REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
   [WITH {GRANT OPTION | resource_option} ...]
-- 用户代理赋权语法
GRANT PROXY ON user
   TO user [, user] ...
   [WITH GRANT OPTION]
object_type: {
  TABLE
  FUNCTION
 PROCEDURE
priv_level: {
 db_name.*
 db_name.tbl_name
 | tbl_name
 | db_name.routine_name
user:
'user_name'@'host_name'
```

```
auth_option: {
    IDENTIFIED BY 'auth_string'
    | IDENTIFIED WITH auth_plugin
    | IDENTIFIED WITH auth_plugin BY 'auth_string'
    | IDENTIFIED WITH auth_plugin AS 'auth_string'
    | IDENTIFIED BY PASSWORD 'auth_string'
}

tls_option: {
    SSL
    | X509
    | CIPHER 'cipher'
    | ISSUER 'issuer'
    | SUBJECT 'subject'
}

resource_option: {
    | MAX_QUERIES_PER_HOUR count
    | MAX_UPDATES_PER_HOUR count
    | MAX_CONNECTIONS_PER_HOUR count
    | MAX_USER_CONNECTIONS count
}
```

● 赋权类型差异

MySQL支持的赋权类型如下:

表 2-57 MySQL 支持的赋权类型

权限类型	释义及权限级别
ALL [PRIVILEGES]	授予指定访问级别的所有权限,除了 GRANT OPTION和 PROXY。
ALTER	启用ALTER TABLE。级别:全局、数 据库、表。
ALTER ROUTINE	允许更改或删除存储过程。级别:全 局、数据库、例程。
CREATE	启用数据库和表创建。级别:全局、 数据库、表。
CREATE ROUTINE	启用存储过程创建。级别:全局、数 据库。
CREATE TABLESPACE	允许创建、更改或删除表空间和日志 文件组。级别:全局。
CREATE TEMPORARY TABLES	启用CREATE TEMPORARY TABLE。 级别:全局、数据库。
CREATE USER	启用CREATE USER、 DROP USER、 RENAME USER和REVOKE ALL PRIVILEGES。级别:全局。
CREATE VIEW	允许创建或更改视图。级别:全局、 数据库、表。
DELETE	启用DELETE. 级别:全局、数据库、 表。

权限类型	释义及权限级别
DROP	允许删除数据库、表和视图。级别: 全局、数据库、表。
EVENT	启用定时任务。级别:全局、数据 库。
EXECUTE	使用户能够执行存储过程。级别:全局、数据库、存储过程。
FILE	使用户能够使服务器读取或写入文 件。级别:全局。
GRANT OPTION	允许向其他账户授予权限或从其他账 户删除权限。级别:全局、数据库、 表、存储过程、代理。
INDEX	允许创建或删除索引。级别:全局、 数据库、表。
INSERT	启用INSERT。级别:全局、数据库、 表、列。
LOCK TABLES	在具有SELECT权限的表上启用LOCK TABLES 。级别:全局、数据库。
PROCESS	使用户能够通过SHOW PROCESSLIST 查看所有正在运行的线程. 级别:全 局。
PROXY	启用用户代理。级别: 从用户到用 户。
REFERENCES	启用外键创建。级别:全局、数据 库、表、列。
RELOAD	启用FLUSH操作的使用。级别:全 局。
REPLICATION CLIENT	使用户能够查询源服务器或副本服务 器的位置。级别:全局。
REPLICATION SLAVE	允许副本从源读取二进制日志。级 别:全局。
SELECT	启用使用SELECT。级别:全局、数据库、表、列。
SHOW DATABASES	启用SHOW DATABASES以显示所有数据库。级别:全局。
SHOW VIEW	启用SHOW CREATE VIEW。级别: 全局、数据库、表。
SHUTDOWN	启用mysqladmin shutdown 的使用。级别:全局。

权限类型	释义及权限级别
SUPER	启用其他管理操作,例如 CHANGE MASTER TO、 KILL、 PURGE BINARY LOGS、 SET GLOBAL和 mysqladmin debug命令。级别:全 局。
TRIGGER	启用触发器操作。级别:全局、数据 库、表。
UPDATE	启用UPDATE。 级别:全局、数据库、表、列。
USAGE	等价于"没有特权"。

M-Compatibility以级别划分支持以下权限:

表 2-58 M-Compatibility 支持的赋权类型

授权对象	支持授予的权限
模式	CREATE、USAGE、ALTER、DROP、 COMMENT
表、视图	SELECT、INSERT、UPDATE、 DELETE、TRUNCATE、 REFERENCES、TRIGGER、ALTER、 DROP、COMMENT、INDEX、 VACUUM
列	SELECT、INSERT、UPDATE、 REFERENCES、COMMENT
序列	SELECT、USAGE、UPDATE、 ALTER、DROP、COMMENT

- MySQL中通过'dbname.*'表示模式层级的授权对象;在M-Compatibility中,使用'{DATABASE | SCHEMA} dbname'表示模式层级的授权对象。
- MySQL中用户名为两部分:用户名@主机名;M-Compatibility当前仅支持用户名。
- MySQL支持在GRANT赋权语法中修改用户验证,安全连接,资源参数属性,即 auth_option、tls_option和resource option; M-Compatibility赋权语法中不支持 以上特性,需使用CREATE USER、ALTER USER设置用户相关属性。
- MySQL支持用户代理赋权,GRANT PROXY ON主要用于对多个用户进行统一的权限管理。MySQL5.7未提供角色机制,而在MySQL8.0和M-Compatibility中都提供了角色机制。角色能满足用户对于多个用户权限统一管控的目标,可以替代GRANT PROXY ON。
- M-Compatibility拥有public的概念,所用用户都拥有public的权限,部分系统表、系统视图可供所有用户查询。用户可以对public所拥有的权限进行grant和

- revoke; MySQL中,新创建的用户只拥有全局的usage权限,这个权限很小,几乎为0,只有连接数据库和查询information schema 数据库的权限。
- M-Compatibility中,对象的所有者缺省具有该对象上的所有权限,出于安全考虑 所有者可以舍弃部分权限,但ALTER、DROP、COMMENT、INDEX、VACUUM以 及对象的可再授予权限属于所有者固有的权限,隐式拥有;MySQL中,没有 owner的概念,即使用户创建了表,如果没赋予用户对应权限,那么用户也不能 对其创建的表进行IUD等操作。
- 在MySQL中,USAGE实际上表示无权限,所用用户都拥有该权限,当执行revoke 或grant usage时,实际上不会进行任何修改;在M-Compatibility中,USAGE权 限如下:
 - 对于模式,USAGE允许访问包含在指定模式中的对象,若没有该权限,则只能看到这些对象的名称。
 - 对于序列,USAGE允许使用nextval函数。
- 在M-Compatibility中,支持给用户设置管理员角色,包括系统管理员(SYSADMIN)、安全管理员(CREATEROLE)、审计管理员(AUDITADMIN)、监控管理员(MONADMIN)、运维管理员(OPRADMIN)、安全策略管理员(POLADMIN)。默认情况下拥有SYSADMIN属性的系统管理员,具备系统最高权限。三权分立后,系统管理员将不再具有CREATEROLE属性(安全管理员)和AUDITADMIN属性(审计管理员)能力,即不再拥有创建角色和用户的权限,也不再拥有查看和维护数据库审计日志的权限;在MySQL中,不支持该用户设置管理员角色,也没有三权分立相关设计。
- 在M-Compatibility中,可以给用户赋予ANY权限,表示用户能够在非系统模式下期有对应的权限,包括CREATE ANY TABLE、SELECT ANY TABLE、CREATE ANY INDEX等;在MySQL中,不支持ANY权限的赋予。
- MySQL中提供SHOW GRANTS查询用户权限; M-Compatibility中,可以通过gsql客户端元命令'\l+'、'\dn+'、'\dp'查询权限信息,也可以通过查询pg_namespace、pg_class、pg_attribute等系统表的权限相关字段查询权限信息。
- MySQL中数据库、表、列被删除时,相关的授权信息在系统表中依然保留,如果 重新创建同名对象用户依然拥有权限; M-Compatibility中当数据库、表、列被删 除时,相关的授权信息会被删除,在重新创建同名对象后需要重新授权。
- MySQL在授予数据库层级的权限时,支持'_'和'%'对数据库名进行模糊匹配; M-Compatibility不支持对象名模糊匹配, '_'或'%'等特殊字符被识别为普通字符。
- MySQL中,GRANT语句中指定用户不存在时默认会创建该账户(此特性已在 MySQL8.0中移除);M-Compatibility不支持给未创建用户赋权。

2.3.8.8 系统表和系统视图

表 2-59 M-Compatibility 与 GaussDB 的系统表或系统视图的差异

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
1	information_schema. columns	generation_expression	该字段输出结果因涉及M-Compatibility与MySQL的表达式的字符串拼接逻辑的不同而存在差异。

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
2	information_schema. columns	data_type	该字段输出结果因涉及M- Compatibility的数据类型 format_type输出,目前未 修改,与MySQL存在差 异。
3	information_schema. columns	column_type	该字段输出结果因涉及M- Compatibility的数据类型 format_type输出,目前未 修改,与MySQL存在差 异。
4	information_schema. tables	engine	M-Compatibility中: ENGINE对齐 information_schema.e ngines数据。 部分系统表ENGINE为 空。 在默认为ASTORE表且 未指定STORAGE_TYPE 时ENGINE为空。
5	information_schema. tables	version	M-Compatibility中不支持 该字段。
6	information_schema. tables	row_format	M-Compatibility中不支持 该字段。
7	information_schema. tables	avg_row_length	M-Compatibility下表示使用数据文件除以所有元组数(包括活元组和死元组)的结果。表中没有元组,值为null。
8	information_schema. tables	max_data_length	M-Compatibility中不支持 该字段。
9	information_schema. tables	data_free	M-Compatibility中表示死 元组在总元组中的比例乘 以数据文件大小。如果表 中没有元组,则为null。
10	information_schema. tables	check_time	M-Compatibility中不支持 该字段。

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
11	information_schema. tables	create_time	在M-Compatibility下,此字段与MySQL行为表现有差异,对于创建视图的情形MySQL中该字段置null,M-Compatibility则显示实际的创建表时间。数据库自带的表,视图设置null。
12	information_schema. tables	update_time	M-Compatibility数据库自 带的表,视图设置null。
13	information_schema. statistics	collation	M-Compatibility只有值 A、D,不会是NULL。
14	information_schema. statistics	packed	M-Compatibility中不支持 该字段。
15	information_schema. statistics	sub_part	M-Compatibility中不支持 该字段。
16	information_schema. statistics	comment	M-Compatibility中不支持 该字段。
17	information_schema. partitions	subpartition_name	M-Compatibility中,如果 分区不是子分区,则为 null。
18	information_schema. partitions	subpartition_ordinal_p osition	M-Compatibility中,如果 分区不是子分区,则为 null。
19	information_schema. partitions	partition_method	M-Compatibility中, 分区策略。如果分区不是 一级分区,则为null。 • 'r': 范围分区。 • 'i': 间隔分区。 • 'l': list分区。 • 'h': hash分区。
20	information_schema. partitions	subpartition_method	M-Compatibility中, 子分区策略。如果分区不是二级分区,则为null。 • 'r': 范围分区。 • 'i': 间隔分区。 • 'l': list分区。 • 'h': hash分区。

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
21	information_schema. partitions	partition_description	M-Compatibility中,是区 分一级分区和二级分区 的。
22	information_schema. partitions	partition_expression	M-Compatibility中不支持 该字段。
23	information_schema. partitions	subpartition_expressio n	M-Compatibility中不支持 该字段。
24	information_schema. partitions	data_length	M-Compatibility中不支持 该字段。
25	information_schema. partitions	max_data_length	M-Compatibility中不支持 该字段。
26	information_schema. partitions	index_length	M-Compatibility中不支持 该字段。
27	information_schema. partitions	data_free	M-Compatibility中不支持 该字段。
28	information_schema. partitions	create_time	M-Compatibility中不支持 该字段。
29	information_schema. partitions	update_time	M-Compatibility中不支持 该字段。
30	information_schema. partitions	check_time	M-Compatibility中不支持 该字段。
31	information_schema. partitions	checksum	M-Compatibility中不支持 该字段。
32	information_schema. partitions	partition_comment	M-Compatibility中不支持 该字段。
33	information_schema. partitions	nodegroup	M-Compatibility中不支持 该字段。

□ 说明

- 视图中对于整型的类型回显,不支持指定精度范围。如MySQL的bigint(1), M-Compatibility下对应的是bigint类型, MySQL中bigint(21) unsigned,在M-Compatibility下对应的是bigint unsigned类型。
- MySQL中int类型,在M-Compatibility中是integer类型。
- MySQL中的set、enum类型的字段,M-Compatibility中都不支持。
 m_schema.columns_priv视图的Column_priv字段、m_schema.tables_priv视图的
 Table_priv,Column_priv字段、m_schema.procs_priv视图的Routine_type,Proc_priv字段、m_schema.proc视图的
 type,language,sql_data_access,is_deterministic,security_type,sql_mode字段、
 m_schema.func视图的type字段均不支持,在此版本中不予显示。
- 由于information_schema.tables中table_rows、avg_row_length、data_length、data_free、index_length及information_schema.statistics中cardinality基于统计信息获取,查看前请先执行ANALYZE,更新统计信息后再查看(如果数据库中更新数据,建议延迟执行ANALYZE)。
- information_schema.statistics包含的索引列需要是创建索引中索引列是完整的表列,如果索引列是表达式,不在这个视图中。
- information_schema.partitions 中table_row基于统计信息获取,查看前请先执行 ANALYZE,更新统计信息后再查看(如果数据库中更新数据,建议延迟执行ANALYZE)。
- 对于支持的grantee字段,MySQL的格式是'*user_name*'@'*host_name*',在M-Compatibility数据库,是被授予权限的用户或角色的名称。
- 对于支持的host字段,在M-Compatibility数据库,返回当前节点的hostname。
- m_schema.tables_priv、information_schema.user_privileges、information_schema.schema_privileges,information_schema.table_privileges、information_schema.column_privileges、m_schema.columns_priv、m_schema.func、m_schema.procs_priv 在MySQL下需要授权后才能查看视图内容,M-Compatibility数据库可以根据默认权限查看到对应的内容。如对于表t1,在MySQL下需要先对t1给对应的用户授权,才能在权限视图中看到对应的权限信息,M-Compatibility数据库下则可以直接在视图中看到t1表相关的权限信息。
- m schema中的系统视图,但在MySQL是系统表。
- information_schema.views中的VIEW_DEFINITION以及information_schema.routines中的ROUTINE_DEFINITION不做字符序控制。
- 《 M-Compatibility开发指南》中"Schema"章节所列举的字符类型的视图字段,字符集使用utf8mb4,字符序使用utf8mb4_bin或utf8mb4_general_ci,字符序优先级为《 M-Compatibility开发指南》中"SQL参考>字符集与字符序>字符集和字符序合并规则"所描述的"支持字符序的数据类型的列"的优先级,和MySQL存在差异。

2.3.9 驱动

2.3.9.1 ODBC

2.3.9.1.1 ODBC 接口参考

获取参数描述信息

SQLDescribeParam接口是ODBC API中的一个函数,用于获取与预处理SQL语句(如调用SQLPrepare)相关参数的描述信息。它可以返回参数的类型、大小、是否允许NULL值等元数据,这对于动态构建SQL语句和绑定参数非常有用。

原型

SQLRETURN SQLDescribeParam(
SQLHSTMT StatementHandle,
SQLUSMALLINT ParameterNumber,
SQLSMALLINT *DataTypePtr,
SQLULEN *ParameterSizePtr,
SQLSMALLINT *DecimalDigitsPtr,
SQLSMALLINT *NullablePtr);

表 2-60 SQLDescribeParam 参数说明

参数名	参数说明	差异
StatementHand le	语句句柄。	-
ParameterNum ber	参数序号,起始为1,依次递增。	-
DataTypePtr	指向返回参数数据类型的指针。	MySQL ODBC对于任意类型均返回SQL_VARCHAR。
		GaussDB ODBC的会根据内 核返回的不同类型判断返回 给应用相应的DataType类 型。
ParameterSizeP tr	指向返回参数大小的指针。	MySQL ODBC若允许ODBC 驱动程序使用更大的数据包 进行数据传输,则返回 24M,否则返回255。
		GaussDB ODBC根据实际类型返回参数大小。
DecimalDigitsPt r	指向返回参数十进制位数的指针。	-
NullablePtr	指向返回参数是否允许NULL值的 指针。	MySQL ODBC直接返回 SQL_NULLABLE_UNKNOW N。 GaussDB ODBC直接返回
		SQL_NULLABLE。

3 集中式版 Oracle 兼容性说明

3.1 Oracle 数据库兼容性概述

GaussDB数据库在基本功能(数据类型、SQL、数据库对象等)和PL/SQL方面与 Oracle数据库基本兼容。但是由于架构设计方面的差异,还是存在一些不兼容的项, 本章节主要介绍GaussDB数据库的Oracle兼容模式与Oracle数据库19c版本的兼容性对 比信息。

3.2 SQL 的基本元素

3.2.1 数据类型

表 3-1 数值类型

序号	Oracle数据 库	GaussDB数 据库	差异
1	NUMBER [(p[, s])]	支持,有差 异	精度和用法存在差异。 NUMBER带参数时,GaussDB的精度p与标度s的最大边界值比Oracle更大。 NUMBER不带参数时,GaussDB的精度p的默认值远大于带参数时的最大边界值;而在Oracle中,精度p的默认值等于带参数时的最大边界值。 GaussDB不支持标度s为负值;在Oracle中,标度s为负值时会精确到相应的整数位。
2	FLOAT [(p)]	支持	-
3	BINARY_FLO AT	不支持	-

序号	Oracle数据 库	GaussDB数 据库	差异
4	BINARY_DO UBLE	支持	-

表 3-2 日期时间类型

序号	Oracle数据库	GaussDB 数据库	差异
1	DATE	支持,有差 异	精度有差异,GaussDB支持的公 元时间较Oracle范围更大。
2	TIMESTAMP [(fractional_seconds_pre cision)]	支持,有差 异	-
3	TIMESTAMP [(fractional_seconds_pre cision)] WITH TIME ZONE	支持,有差异	GaussDB的timestamptz等价于Oracle的timestampwithlocaltimezone,缺少Oracle对应的timestamptz类型。 时区更新:部分国家或地区经常会更新时区信息,数据库系统因此需要同步修改时区文件以确保时间内容的正确性。 GaussDB时区类型目前只涉及timestampwithtimezone,当新的时区文件生效时,不会对已有的数据进行变更,新数据会随时区文件信息进行同步调整。与Oracle的同类型数据能力存在差异。
4	TIMESTAMP [(fractional_seconds_pre cision)] WITH LOCAL TIME ZONE	不支持	-
5	INTERVAL YEAR [(year_precision)] TO MONTH	支持	-
6	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_pre cision)]	支持	-

□ 说明

- A兼容模式下,DATE类型被替换为TIMESTAMP(0) WITHOUT TIME ZONE,差异同 TIMESTAMP(0) WITHOUT TIME ZONE。
- 对于TIMESTAMP [(fractional_seconds_precision)]/TIMESTAMP [(fractional_seconds_precision)]WITH TIME ZONE与Oracle存在以下差异。
 - fractional_seconds_precision GaussDB支持的精度范围为0~6,Oracle支持的精度范围为0~9。
 - GaussDB通过DateStyle设置日期和时间值的显示格式,以及有歧义的输入值的解析规则。具体请参见《开发指南》中"SQL参考>数据类型>日期/时间类型"章节中日期输入的说明介绍。Oracle在一般情况下的输入格式校验及输出显示由NLS_TIMESTAMP_FORMAT/NLS_TIMESTAMP_TZ_FORMAT参数进行控制。
 - 对于秒的小数部分的显示,GaussDB默认去除末尾的零,Oracle依据格式化参数中的设置(FF/FF1-FF9)进行显示控制。如 '2017-09-01 10:32:19.212000',GaussDB显示为 '2017-09-01 10:32:19.212',Oracle在format参数中含FF时显示为'2017-09-01 10:32:19.212000',在format参数中含FF9时显示为'2017-09-01 10:32:19.212000000'。
 - 支持的时间范围存在差异,GaussDB支持的公元时间较Oracle范围更大。

表 3-3 字符类型

序号	Oracle数据库	GaussDB数 据库	差异
1	VARCHAR2 (size [BYTE CHAR])	支持,有差 异	在GaussDB中,size单位为字节,即仅支持BYTE,不支持在BYTE和CHAR之间选择,最大10MB;而在Oracle中,size的单位可以在BYTE和CHAR之间选择,MAX_STRING_SIZE=EXTENDED时,最大长度为32767字节,MAX_STRING_SIZE=STANDARD时,最大长度为4000字节,实际能容纳的字符数与使用的字符集有关。
2	NVARCHAR2 (size)	支持,有差 异	在GaussDB中,NVARCHAR2(n)类型是VARCHAR2(n)类型的别名;而在Oracle中NVARCHAR2(n)与VARCHAR2(n)有一定的区别,MAX_STRING_SIZE=EXTENDED时,最大长度为32767字节,MAX_STRING_SIZE=STANDARD时,最大长度为4000字节,实际能容纳的字符数与使用的字符集有关。
3	CHAR [(size [BYTE CHAR])]	支持,有差 异	在GaussDB中,size单位为字节,即仅支持BYTE,不支持在BYTE和CHAR之间选择,最大10MB;而在Oracle中,size的单位可以在BYTE和CHAR之间选择,最大容量为2000个字节,实际能容纳的字符数与使用的字符集有关。

序号	Oracle数据库	GaussDB数 据库	差异
4	NCHAR [(size)]	支持,有差 异	在GaussDB中,size单位为字节,最大 10MB;而在Oracle中,size单位为字 符,最大容量为2000个字节,实际能容 纳的字符数与使用的字符集有关。
5	CLOB	支持,有差 异	不支持定位器。
6	NCLOB	不支持	-
7	LONG	不支持	-

表 3-4 二进制类型

序号	Oracle数 据库	GaussDB数 据库	差异
1	RAW (size)	支持,有差 异	在GaussDB中,size是指字节长度建议值,不 会用于校验输入raw类型的字节长度。
2	LONG RAW	不支持	-
3	BLOB	支持	-
4	BFILE	不支持	-

表 3-5 ROWID 类型

序号	Oracle数据库	GaussDB数据库
1	ROWID	不支持
2	UROWID	不支持

表 3-6 用户自定义类型

序号	Oracle数据库	GaussDB数据库
1	对象类型	不支持
2	REF数据类型	不支持
3	可变数组	支持
4	嵌套表	支持

表 3-7 伪类型

序号	Oracle数据库	GaussDB数据库
1	anytype	不支持
2	anydata	不支持
3	anydataset	不支持

表 3-8 xml 类型

序号	Oracle数 据库	GaussDB数 据库	差异
1	XMLTYPE	支持,有差 异	在GaussDB中,不支持部分操作,如不能通过 使用XMLELEMENT函数将字符串转变为 XMLTYPE类型,而是转变成XML类型。具体请 参见《开发指南》中"SQL参考 > 数据类型 > XMLTYPE类型"章节。
2	URIType	不支持	-

表 3-9 空间类型

序号	Oracle数据库	GaussDB数据库
1	SDO_GEOMETRY	不支持
2	SDO_TOPO_GEOMETRY	不支持
3	SDO_GEORASTER	不支持

表 3-10 锁模式

序号	Oracle数据库	GaussDB数据库
1	none	-
2	null	AccessShare
3	RS	RowShare
4	RX	RowExclusive
5	S	ShareUpdateExclusive
6	SRX	Share
7	-	ShareRowExclusive
8	Х	Exclusive

序号	Oracle数据库	GaussDB数据库
9	-	AccessExclusive
10	-	INVALID 说明 GaussDB的INVALID表示被赋予了非法锁。仅在运 行过程中出现了GaussDB无法识别的锁时会被赋予 INVALID锁。

3.2.2 数据类型比较规则

数据类型比较(排序)规则是指相同数据类型的值之间发生比较(排序)时遵循的比较(排序)规则。

表 3-11 比较规则

序号	Oracle数 据库	GaussDB 数据库	差异
1	Numeric 值	支持	-
2	日期时间 值	支持	-
3	二进制值	支持	-
4	字符值	支持,有	 在GaussDB和Oracle中,支持的比较规则不完全相同,相同比较规则的名称也可能不同。 GaussDB和Oracle的比较规则在可指定性上有差异,例如GaussDB不支持指定表级别的比较规则,而Oracle支持。 GaussDB和Oracle在指定比较规则的语法上有差异,例如在GaussDB中,使用ENCODING、LC_CTYPE和LC_COLLATE三个参数决定创建数据库时使用的字符集、字符分类和比较规则,具体请参见《开发指南》中"SQL参考>SQL语法>C>CREATE DATABASE"章节。而在Oracle中,各级别的比较规则通常由一系列带有NLS前缀的参数确定。
5	对象值	不支持	-
6	Varrays和 嵌套表	支持,有 差异	Oracle和GaussDB均支持Varrays的比较,与 Oracle不同的是,GaussDB不仅支持比较两个 Varrays中的元素个数,还支持同类型的Varrays之 间的比较。
7	数据类型 优先级	支持	-

序号	Oracle数 据库	GaussDB 数据库	差异
8	数据转换 (显示/隐 式类型转 换)	支持	-

3.2.3 字面量

表 3-12 字面量

序号	Oracle数据库	GaussDB数据库
1	文本字面量	支持
2	数值字面量	支持
3	日期时间字面量	支持
4	区间字面量	支持

3.2.4 格式模型

表 3-13 格式

序号	Oracle数据 库	GaussDB数 据库	差异
1	数字格式	支持,有差异	GaussDB仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下,支持\$、C、TM、TM9、TME、U格式。同时在该参数下,不支持TH、PL、SG格式。GaussDB具体支持情况请参见《开发指南》中"SQL参考>函数和操作符>类型转换函数"章节的"number类型fmt参数表"。
2	日期时间格式	支持,有差 异	GaussDB中用于时间截断和时间四舍五入的参数,仅在参数a_format_version值为10c和a_format_dev_version值为s1的情况下有效。GaussDB具体支持情况请参见《开发指南》中"SQL参考>函数和操作符>时间和日期处理函数和操作符"章节的"用于日期/时间格式化的模式"。
3	格式模型修 饰符	支持	-

序号	Oracle数据 库	GaussDB数 据库	差异
4	字符串到日期转换规则	支持,有差 异	GaussDB中to_timestamp_tz函数在参数 a_format_version值为10c和 a_format_dev_version值为s1的情况下有效。 GaussDB具体支持情况请参见《开发指南》中 "SQL参考>函数和操作符>类型转换函数"章 节的"to_date/to_timestamp/ to_timestamp_tz"。
5	xml格式模 式	不支持	-

3.2.5 空值

表 3-14 空值

序号	Oracle数据库	GaussDB数据库
1	IS NULL和IS NOT NULL	支持
2	NULLS in conditions	支持

3.2.6 注释

表 3-15 comment

序号	Oracle数据 库	GaussDB 数据库	差异
1	斜杠和星号 (/*)	支持	-
2	两个连字符 ()	支持	-
3	COMMENT 命令	支持	-
4	HINT	支持,有差 异	GaussDB不支持'+'hint形式。 具体信息请参见《 开发指南 》中"SQL调优指 南 > 使用Plan Hint进行调优"章节。

3.2.7 数据库对象

表 3-16 schema 对象

序号	Oracle数 据库	GaussDB数 据库	差异
1	分析视图	不支持	-
2	属性维度	不支持	-
3	集群	支持	-
4	约束	支持	-
5	数据库链 接	支持	-
6	数据库触 发器	支持	-
7	尺寸	支持	-
8	外部过程 库	不支持	-
9	分层结构	不支持	-
10	索引组织 表	不支持	-
11	索引	支持	-
12	索引类型	不支持	-
13	java类	不支持	-
14	java资源	不支持	-
15	java源码	不支持	-
16	join groups	不支持	-
17	物化视图	支持	-
18	物化视图 日志	不支持	-
19	挖掘模型	不支持	-
20	对象表	不支持	-
21	对象类型	不支持	-
22	对象视图	不支持	-
23	operators	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
24	包	支持	-
25	序列	支持	-
26	存储函数	支持	-
27	存储过程	支持	-
28	同义词	支持,有差 异	Oracle数据库的数据库对象在同一namespace内不能重名;GaussDB内同一namespace内同义词可与表、视图、函数、package重名,重名时,优先访问该名称对象,未寻找到该名称对象时才会寻找该名称的同义词指向的对象;同义词指向的对象的Schema名是用户名时才会搜索PUBLIC同义词。详细的搜索顺序参考《开发指南》中的"SQL参考 > SQL语法 > C > CREATE SYNONYM"章节中的注意事项,同义词的搜索过程。
29	表	支持	-
30	视图	支持	-
31	zone map	不支持	-

表 3-17 nonschema 对象

序号	Oracle数据库	GaussDB数据库
1	上下文	不支持
2	目录	支持
3	版本	不支持
4	闪回存档	不支持
5	锁定配置文件	不支持
6	配置文件	不支持
7	还原点	支持
8	角色	支持
9	回滚段	ustore支持回滚段astore不支持回滚段
10	表空间	支持
11	表空间集	不支持
12	统一审计策略	支持

序号	Oracle数据库	GaussDB数据库
13	用户	支持

3.2.8 数据库对象名称和限定符

表 3-18 命名规则

序号	Oracle数据库	GaussDB数据库	差异
1	数据库对象命名 规则	支持,有差异。	GaussDB默认小写。
2	模式对象命名规 则	支持	-

3.2.9 SQL 语句中的引用架构对象和部件的语法

表 3-19 对象引用

序号	Oracle数据库	GaussDB数据库
1	引用对象的一般语法	支持
2	解析对象的引用	支持
3	引用外部模式对象	支持
4	引用外部数据库对象	支持
5	引用表和索引分区和子分区	支持

3.3 伪列

GaussDB数据库兼容分层查询伪列、序列伪列、rownum伪列,其余暂不支持。

分层查询伪列

表 3-20 分层查询伪列

序号	Oracle数据库	GaussDB数据库
1	connect_by_iscycle	支持
2	connect_by_isleaf	支持
3	level伪列	支持

序列伪列

表 3-21 序列

序号	Oracle数 据库	GaussDB数据库	差异
1	currval	支持,有差异	GaussDB以函数形式实现。兼容Oracle调用方式。
2	nextval	支持,有差异	GaussDB以函数形式实现。兼容Oracle调用方式。

rownum 伪列

表 3-22 rownum

序号	Oracle数据 库	GaussDB数据 库	差异
1	rownum	支持,有差异	Oracle在left、right、full join的条件中使用rownum进行 过滤时,不同的条件下表现不尽相同,可能存在忽略或部 分忽略rownum条件的现象,而GaussDB在此情况下则表 现为对left、right、full join后的结果进行过滤。

3.4 操作符

GaussDB数据库基本兼容Oracle数据库的运算符。

SQL 运算符

表 3-23 SQL 运算符

序号	Oracle数据库	GaussDB数据库
1	一元运算符和二元运算符	支持
2	运算符优先级	支持

算术运算符

表 3-24 算术运算符

序号	Oracle数据库	GaussDB数据库
1	正负(+-)一元运算符	支持

序号	Oracle数据库	GaussDB数据库
2	加减(+-)二元运算符	支持
3	乘除(*/)二元运算符	支持

COLLATE 运算符

表 3-25 COLLATE 运算符

序号	Oracle数据库	GaussDB数据库
1	COLLATE collation_name	支持

连接运算符

表 3-26 连接运算符

序号	Oracle数据库	GaussDB数据库
1		支持

分层查询运算符

表 3-27 分层查询运算符

序号	Oracle数据库	GaussDB 数据库	差异
1	prior	支持,有 差异	GaussDB中仅支持对普通列调用,不支 持对函数等调用。
2	connect_by_root	支持,有 差异	GaussDB中,connect_by_root调用时, 用括号修饰操作值时,行为与Oracle一 致;不使用括号时,仅支持对普通列调 用此运算符。
3	connect_by_iscycl e	支持,有 差异	GaussDB中可以单独使用,Oracle中必 须搭配NOCYCLE一起使用。
4	start with, connect by, level	支持,有 差异	GaussDB中start with,connect by, level可以作为列别名,Oracle不支持。
5	start with	支持,有 差异	GaussDB中start with后面支持空字符查询,查询结果为空,支持对伪列的操作;Oracle会抛出异常。

序号	Oracle数据库	GaussDB 数据库	差异
6	connect by	支持,有 差异	GaussDB中connect by支持对NULL值操作,Oracle会抛出异常。

集合运算符

表 3-28 集合运算符

序号	Oracle数据库	GaussDB数据库
1	union	支持
2	union all	支持
3	intersect	支持
4	minus	支持

多集合运算符

表 3-29 多集合运算符

序号	Oracle数据库	GaussDB数据库
1	multiset except	支持
2	multiset intersect	支持
3	multiset union	支持

用户自定义运算符

表 3-30 用户自定义运算符

序号	Oracle数据 库	GaussDB 数据库	差异
1	CREATE OPERATOR	支持,有 差异	 Oracle中提供的CONTEXT_CLAUSE支持自 定义功能评估函数,和GaussDB约束选择 性评估函数不同。GaussDB不支持自定义 功能评估函数。
			 Oracle和GaussDB可选参数差异较大。 GaussDB具体请参考《开发指南》中 "SQL参考 > SQL语法 > C > CREATE OPERATOR"章节的参数说明部分。

比较操作符

序号	Oracle数据库	GaussDB数据库
1	<=	支持
2	<>	支持
3	>=	支持
4	∧ =	支持
5	! =	不支持,!=中间存在空格时,!会被识别为阶 乘。

当比较操作符(<=、<>、>=、^=)中间存在空格时,也可以识别成没有空格进行正常操作。!=中间存在空格时,!会被识别为阶乘,可能会导致结果与预期不一致。

3.5 表达式

GaussDB数据库兼容大部分Oracle数据库表达式。

表 3-31 表达式

序号	Oracle数据 库	GaussDB数 据库	差异
1	简单表达式	支持	-
2	分析视图表达 式	不支持	-
3	复合表达式	支持	-
4	case表达式	支持	-
5	列表示式	支持	-
6	cursor表达式	不支持	-
7	日期时间表达式	支持,有差 异	GaussDB的输出结果中不会带时区信息,而 Oracle会带有类似"PM AMERICA/ LOS_ANGELES"的时区信息。
8	函数表达式	支持	-
9	区间表达式	部分支持	GaussDB支持形如SELECT INTERVAL '999999999 23:59:59:999' day(9) to second FROM DUAL;的语句,而不支持形 如SELECT(SYSDATE- SYSDATE) DAY TO SECOND FROM DUAL;的语句。Oracle均支 持。

序号	Oracle数据 库	GaussDB数 据库	差异	
10	json对象访问 表达式	部分支持, 有差异	1. GaussDB支持通过 "->'key' "的方式从 JSON对象中提取value,而Oracle支持通过 ".key "方式来提取value。	
			2. 对于JSONARRY对象而言,Oracle支持通 过".key"方式一次性提取所有key对应的 value,但GaussDB目前不支持。	
11	模型表达式	不支持	-	
12	对象表达式	不支持	-	
13	占位符表达式	部分支持	对于形如":var"的一般占位符表达式, GaussDB支持,但不支持通过INDICATOR关 键字将两个一般占位符表达式结合起来。	
14	标量子查询表 达式	支持	-	
15	类型构造器表 达式	部分支持	GaussDB不支持在类型构造器前指定New关键字,而Oracle支持。	
16	表达式list	支持	-	

3.6 条件

本章节描述常见的条件兼容项,包含比较条件、浮点条件、逻辑条件、模型条件、多集合条件、模式匹配条件、NULL值条件、XML条件、SQL/JSON条件、复合条件、BETWEEN条件、EXISTS条件、IN条件、IS OF TYPE条件。详情请参见表3-32。

表 3-32 条件

序号	Oracle数据 库	GaussDB数据 库	差异
1	比较条件	支持,有差异	语句中存在ANY、SOME、ALL操作符时存在差异,Oracle支持对list对象进行操作,而GaussDB中需要将list对象转换成数组表达式的形式后再进行操作。
2	浮点条件	不支持	-
3	逻辑条件	支持	-
4	模型条件	不支持	-
5	多集合条件	不支持	-
6	模式匹配条 件	支持	-
7	NULL值条件	支持	-

序号	Oracle数据 库	GaussDB数据 库	差异
8	XML条件	不支持	-
9	SQL/JSON条 件	部分支持,有 差异	● GaussDB不支持IS JSON条件和 JSON_TEXTCONTAINS条件。
			 GaussDB中JSONB_EQ条件等同于Oracle 中JSON_EQUAL条件,但GaussDB不支 持ERROR子句。
			 GaussDB中JSONB_EXISTS条件等同于 Oracle中JSON_EXISTS条件,但GaussDB 不支持ERROR子句、EMPTY子句和 PASSING子句。
10	复合条件	支持	-
11	BETWEEN条 件	支持	-
12	EXISTS条件	支持	-
13	IN条件	支持	-
14	IS OF TYPE 条件	不支持	-

3.7 常见的 SQL DDL 子句

本章节描述常见的SQL DDL子句兼容项,包含分配扩展子句、约束、取消分配未使用子句、文件规范、日志记录子句、并行子句、物理属性子句、大小子句、存储子句、聚集函数嵌套。详情请参见<mark>表3-33</mark>。

表 3-33 常用 SQL DDL 子句

序号	Oracle数据库	GaussD B数据 库	差异
1	分配扩展子句 语法: ALLOCATE EXTENT [({SIZE size_clause DATAFILE 'filename' INSTANCE integer })] 例如: 创建employees表后,改变表的分配扩展size为10M。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. SQL> ALTER TABLE employees ALLOCATE EXTENT (SIZE 10M); Table altered.	不支持	
2	约束 语法: { inline_constraint out_of_line_constraint inline_ref_constraint out_of_line_ref_constraint } 例如: 创建表staff,约束子句中指定ID列、NAME列不为空。 SQL> CREATE TABLE staff(ID INT NOT NULL, NAME char(8) NOT NULL, AGE INT, ADDRESS CHAR(50), SALARY REAL); Table created.	支持	-

序号	Oracle数据库	GaussD B数据 库	差异
3	取消分配未使用子句语法: DEALLOCATE UNUSED [KEEP size_clause] 例如: 创建employees表,进行了一些插入、删除操作后,希望使用取消分配未使用子句释放employees表末使用的空间。 SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), JOB_ID NUMBER(38), LAST_NAME VARCHAR2(16)); Table created 进行一些插入、删除操作 SQL> ALTER TABLE employees DEALLOCATE UNUSED;	不支持	
	Table altered.		
4	文件规范 语法: {['filename' 'ASM_filename'] [SIZE size_clause] [REUSE] [autoextend_clause]} {['filename ASM_filename' ('filename ASM_filename'])] [SIZE size_clause] [BLOCKSIZE size_clause [REUSE]} 例如: 创建一个临时表空间 tbs_temp_01, SQL语句的文件规范子句中指定在表空间中创建一个临时数据库文件 templ01.dbf,可以自动扩展,并将表空间分配给表空间组tbs_grp_01。 SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	不支持	

序号	Oracle数据库	GaussD B数据 库	差异
5	日志记录子句 语法: { LOGGING NOLOGGING FILESYSTEM_LIKE_LOGGING }	(有)	● GaussDB不支持LOGGING约束子句和FILESYSTEM_LIKE_LOGGING约束子句。例如: GaussDB创建表,带LOGGING约束子句,语法报错。gaussdb=# CREATE LOGGING TABLE my_tab(id int, name char(16)); ERROR: syntax error at or near "LOGGING" LINE 1: CREATE LOGGING TABLE my_tab(id int, name char(16)); ^ CaussDB创建表,带FILESYSTEM_LIKE_LOGGING约束子句,语法报错。gaussdb=# CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name char(16)); ERROR: syntax error at or near "FILESYSTEM_LIKE_LOGGING" LINE 1: CREATE FILESYSTEM_LIKE_LOGGING" LINE 1: CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name cha ^
			 GaussDB仅支持表级的UNLOGGED约束,不支持列级的UNLOGGED约束。例如:GaussDB创建表,带列级的UNLOGGED约束子句,语法报错。gaussdb=# CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16)); ERROR: syntax error at or near "UNLOGGED" LINE 1: CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16)) GaussDB仅支持在CREATE
			FABLE SUBLES OF TABLE AS SELECT INTO语句中使用日志记录子句。例如: GaussDB创建TABLESPACE,带UNLOGGED约束子句,语法报错。gaussdb=# CREATE UNLOGGEDTABLESPACE tbs1 RELATIVE LOCATION 'tablespace1/tablespace_1'; ERROR: syntax error at or near"TABLESPACE"LINE 1: CREATE UNLOGGED

序号	Oracle数据库	GaussD B数据 库	差异
			TABLESPACE tbs1 RELATIVE LOCATION 'tablespac ^
6	并行子句 语法: { NOPARALLEL PARALLEL [integer] } 例如: 创建表t1,并在并行 子句中指定PARALLEL 4,意 为查询和更新表t1时最多使 用4个并行进程操作。 SQL> CREATE TABLE t1 (id NUMBER, name VARCHAR2(50)) PARALLEL 4; Table created.	不支持	
7	物理属性子句 语法: [{PCTFREE integer PCTUSED integer INITRANS integer storage_clause }]	部持, 是	● GaussDB不支持PCTUSED。例如: 执行在表tbl1中创建一个tbl1_ind的索引的SQL语句,并在该语句的物理属性子句中指定索引的空间利用率PCTUSED为20%,GaussDB执行该SQL语句语法报错。gaussdb=# CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20; ERROR: syntax error at or near "PCTUSED" LINE 1: CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20;

序号	Oracle数据库	GaussD B数据 库	差异
8	大小子句语法: integer [K M G T P E] 例如:创建一个临时表空间 tbs_temp_01,并在表空间 中创建一个临时数据库文件 templ01.dbf,SQL语句的大 小子句中指定初始大小是 5M,可以自动扩展,并将表 空间分配给表空间组 tbs_grp_01。 SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' SIZE 5M AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	不支持	

序号	Oracle数据库	GaussD B数据 库	差异
9	存储子句 语法: STORAGE ({ INITIAL size_clause NEXT size_clause MINEXTENTS integer MAXEXTENTS { integer UNLIMITED } maxsize_clause PCTINCREASE integer FREELISTS integer FREELIST GROUPS integer OPTIMAL [size_clause NULL] BUFFER_POOL { KEEP RECYCLE DEFAULT } FLASH_CACHE { KEEP NONE DEFAULT } CELL_FLASH_CACHE (KEEP NONE DEFAULT)) ENCRYPT })	部持差异	 Oracle中由STORAGE子句指定存储参数,而GaussDB中由WITH子句指定存储参数。例如: Oracle中创建表my_tab1,在存储子句中指定表初始大小为10M,需要更多空间时每次增加5M的SQL语句如下: SQL> CREATE TABLE my_tab1 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) STORAGE (INITIAL 10M NEXT 5M); Table created. GaussDB中创建表my_tab2,在存储子句中指定存储引擎类型为USTORE的SQL语句如下:gaussdb=# CREATE TABLE my_tab2 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) with (storage_type=ustore); NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "my_tab2_pkey" for table "my_tab2" CREATE TABLE GaussDB中可选的存储参数和 Oracle存在很大差异。GaussDB具体可参见《开发指南》中"SQL参考>SQL语法>C> CREATE TABLE"的参数说明部分,"WITH ({storage_parameter = value}[,])"中描述了CREATE TABLE语句支持的存储参数。

序号	Oracle数据库	GaussD B数据 库	差异
10	聚集函数嵌套 例如:创建由sales表的 sales_amount列嵌套聚集函 数MIN()、SUM()生成的表 revenue。 SQL> CREATE TABLE sales(ID INT, SALES_AMOUNT INT); Table created. SQL> INSERT INTO sales VALUES(1, 100); 1 row created. SQL> INSERT INTO sales VALUES (3, 200); 1 row created. SQL> CREATE TABLE revenue as SELECT SUM(MIN(sales_amount)) as total from sales group by sales_amount; Table created.	支持	

3.8 SQL 查询和子查询

GaussDB数据库兼容除分层查询以外的SQL查询和子查询。

表 3-34 SQL 查询和子查询

序号	Oracle数据库	GaussDB数 据库	差异
1	创建简单查询	支持	-
2	分层查询	支持,有差 异	GaussDB仅支持Oracle中的 CONNECT_BY_FILTERING 模式,不支持 CONNECT_BY_NOFILTERING。
3	UNION [ALL], INTERSECT, 减运算符	支持	-
4	查询结果排序	支持,有差 异	GaussDB查询不包含分组,且目标列同时包含聚集函数和集合返回函数时,不忽略对集合返回函数时,不忽略对集合返回函数列的排序。

序号	Oracle数据库	GaussDB数 据库	差异
5	Joins	支持,有差 异	GaussDB只支持和Oracle的Join Types,如 left/right、self 、natural、full outer join 等。不支持 In-Memory Join Group 等 Join Optimizations方法。
6	使用子查询	支持	-
7	嵌套子查询的 解嵌套	支持,有差 异	GaussDB不支持显式指定HASH_AJ或 MERGE_AJ。
8	分布式查询	支持,有差 异	GaussDB需要显式指定DBLINK查询。
9	聚集函数嵌套	支持	-

3.9 PL/SQL 语言

GaussDB数据库基本兼容的PL/SQL操作符、表达式,控制语句、集合和record等等,不支持预定义的PL/SQL常量和类型、子类型等。

3.9.1 PL/SQL 基本语法

表 3-35 PL/SQL 操作符

序号	Oracle数据库	GaussDB数据库
1	+	支持
2	:=	支持
3	=>	支持
4	%	支持
5	1	支持
6		支持
7		支持
8	/	支持
9	**	不支持
10	(支持
11)	支持
12	:	支持
13	1	支持

序号	Oracle数据库	GaussDB数据库
14	<<	支持
15	>>	支持
16	/*	支持
17	*/	支持
18	*	支持
19	II	支持
20		支持
21	=	支持
22	<>	支持
23	!=	支持
24	~=	支持
25	Λ=	支持
26	<	支持
27	>	支持
28	<=	支持
29	>=	支持
30	@	支持
31		支持
32	;	支持
33	-	支持

表 3-36 逻辑运算符

序号	Oracle数据库	GaussDB数据库
1	NOT	支持
2	AND	支持
3	OR	支持

表 3-37 比较表达式

序号	Oracle数据库	GaussDB数据库
1	IS [NOT] NULL	支持
2	LIKE	支持
3	BETWEEN	支持
4	IN	支持

表 3-38 条件表达式

序号	Oracle数据库	GaussDB数据库
1	simple CASE	支持
2	searched CASE	支持

表 3-39 变量声明相关参数

序号	Oracle数 据库	GaussDB数 据库	差异
1	%TYPE	支持,有差异	 GaussDB不支持record变量%type。 GaussDB不支持pkg.record变量%type 、
			%type、pkg.record变量.column.column %type嵌套1层及以上的record的某列类型, 作为变量类型和或者出入参类型。
2	%ROWTYP E	支持,有差 异	 GaussDB在多个CN的环境下,存储过程中无法声明临时表的%ROWTYPE及%TYPE属性。因为临时表仅在当前session有效,在编译阶段其他CN无法看到当前CN的临时表。故多个CN的环境下,会提示该临时表不存在。 GaussDB不支持view%rowtype、
			schema.view%rowtype作为出入参类型。 • GaussDB不支持package.cursor变
			量%rowtype作为出入参类型。

3.9.2 数据类型兼容性

表 3-40 其他 PL/SQL 数据类型

序号	Oracle数据 库	GaussDB数 据库	差异
1	CHARACTER	支持,有差异	GaussDB中字节长度限制为: 1~10485760。Oracle中字节长度限制为: 1~32767。
2	VARCHAR	支持,有差异	GaussDB中字节长度限制为: 1~10485760。Oracle中字节长度限制为: 1~32767。
3	STRING	不支持	-
4	PLS_INTEGE R	不支持	GaussDB可使用int类型替代。
5	BINARY_INT EGER	支持	-

表 3-41 用户自定义 PL/SQL 子类型

序号	Oracle数据库	GaussDB 数据库	差异
1	SUBTYPE subtype_name IS base_type	支持	-
2	SUBTYPE subtype_name IS base_type { precision [, scale] RANGE low_value high_value } [NOT NULL]	支持,有差 异	 GaussDB仅INT类型支持range约束。Oracle仅PLS_INTEGER、BINARY_INTEGER及其预定义子类型支持RANGE约束。 GaussDB中在定义变量时,不支持指定range约束。 GaussDB中,subtype基类型为字符数据类型时,不支持指定字符集信息。
3	SUBTYPE subtype_name IS base_type [NOT NULL]	支持	-

3.9.3 控制语句

表 3-42 条件语句

序号	Oracle数据库	GaussDB数据库
1	IF THEN	支持
2	IF THEN ELSE	支持
3	IF THEN ELSIF	支持
4	simple CASE:	支持
	CASE selector	
	WHEN selector_value_1 THEN statements_1	
	WHEN selector_value_2 THEN statements_2	
	WHEN selector_value_n THEN statements_n	
	[ELSE	
	else_statements	
	END CASE;]	
5	searched CASE:	支持
	CASE	
	WHEN condition_1 THEN statements_1	
	WHEN condition_2 THEN statements_2	
	WHEN condition_n THEN statements_n	
	[ELSE	
	else_statements	
	END CASE;]	

表 3-43 LOOP 循环语句

序号	Oracle数据库	GaussDB数据库
1	[label] LOOP	支持
	statements	
	END LOOP [label];	

序号	Oracle数据库	GaussDB数据库
2	EXIT;	支持
3	EXIT WHEN;	支持
4	CONTINUE;	支持
5	CONTINUE WHEN;	支持

表 3-44 FOR 循环语句

序号	Oracle数据库	GaussDB 数据库	差异
1	[label] FOR index IN [REVERSE] lower_boundupper _bound LOOP statements END LOOP [label];	支持,有差 异	GaussDB使用REVERSE关键字时, lower_bound必须大于等于 upper_bound,否则循环体不会被执 行。
2	EXIT WHEN;	支持	-
3	CONTINUE WHEN;	支持	-

表 3-45 WHILE LOOP 循环语句

序号	Oracle数据库	GaussDB数据库
1	[label] WHILE condition LOOP	支持
	statements	
	END LOOP [label];	

表 3-46 GOTO 语句

序号	Oracle数据库	GaussDB数据库
1	GOTO	支持

表 3-47 NULL 语句

序号	Oracle数据库	GaussDB数据库
1	NULL	支持

3.9.4 集合和 Record

表 3-48 类型

序号	Oracle数据库	GaussDB数据库
1	Associative array (or index-by table)	支持
2	VARRAY (variable-size array)	支持
3	Nested table	支持
4	record	支持

表 3-49 语法

序号	Oracle数据库	GaussDB 数据库	差异
1	Associative array (or index-by table)语法: TABLE OF datatype [NOT NULL] INDEX BY { PLS_INTEGER BINARY_INTEGER VARCHAR2 (v_size) data_type }	支持,有 差异	 GaussDB不支持 PLS_INTEGER类型,GaussDB内data_type可以为基础数据类型、或存储过程内定义的record类型、集合类型、数组类型,不支持ref cursor类型。 GaussDB内NOT NULL只支持语法不支持功能,即不会校验元素是否为NULL。 详情可参考《开发指南》中"存储过程>数组、集合和record>集合"章节。
2	VARRAY (variable- size array)语法: { VARRAY [VARYING] ARRAY } (size_limit) OF datatype [NOT NULL]	支持,有 差异	 GaussDB内不支持NOT NULL语法。 GaussDB内不支持datatype为varray类型(varray不能嵌套)。 size_limit功能生效需要在behavior_compat_optionsGUC参数中开启varray_compat参数。 详情可参考《开发指南》中"存储过程>数组、集合和record>数组"章节。
3	Nested table语法: TABLE OF datatype [NOT NULL]	支持,有 差异	 GaussDB内NOT NULL只支持语法 不支持功能,即不会校验元素是否 为NULL。 详情可参考《开发指南》中"存储 过程 > 数组、集合和record > 集 合"章节。

序号	Oracle数据库	GaussDB 数据库	差异
4	record语法: TYPE record_type IS RECORD (field_definition [, field_definition]);	支持	 record的列可以定义为NOT NULL 属性也可以指定默认值。其他类型 嵌套record类型,record类型的默 认值和NOT NULL不生效;通过 package.record_type访问类型的形 式来创建record变量,该record变 量的默认值和NOT NULL不生效。 详情可参考《开发指南》中"存储 过程 > 数组、集合和record > record"章节。

表 3-50 构造器

序号	Oracle数据库	GaussDB数据库
1	collection_type ([value [, value]])	支持

表 3-51 变量赋值

序号	Oracle数据库	GaussDB 数据库	差异
1	Associative array (or index-by table)	支持	-
2	VARRAY (variable- size array)	支持,有 差异	 GaussDB不同VARRAY类型的数据可以相互赋值,取决于其元素之间是否能相互隐式转换。 详情可参考《开发指南》中"存储过程>数组、集合和record>数组"章节。
3	Nested table	支持	-
4	record	支持,有 差异	 GaussDB不同record类型的数据可以相互赋值,取决于列与列之间是否能相互隐式转换。 详情可参考《开发指南》中"存储过程>数组、集合和record>
			record "章节。

表 3-52 集合操作符

序号	Oracle数据库	GaussDB数 据库	差异
1	=	支持,有差异	Oracle:比较时忽略集合成员先后顺序。GaussDB:比较时严格按照集合成员先后顺序。
2	<>	支持,有差异	Oracle:比较时忽略集合成员先后顺序。GaussDB:比较时严格按照集合成员先后顺序。
3	IS[NOT] NULL	支持。	-
4	^=	支持,有差 异	Oracle:比较时忽略集合成员先后顺序。GaussDB:比较时严格按照集合成员先后顺序。
5	~=	不支持	-
6	IS[NOT] A SET	不支持	-
7	IS [NOT] EMPTY	不支持	-
8	expr [NOT] MEMBER [OF] nested_table	不支持	-
9	nested_table1 [NOT] SUBMULTISET [OF] nested_table2	不支持	-
10	[NOT] IN	支持,有差 异	Oracle:比较时忽略集合成员先后顺序。GaussDB:比较时严格按照集合成员先后顺序。

表 3-53 集合 MULTISET 函数

序号	Oracle数据库	GaussDB数据库
1	MULTISET UNION [ALL DISTINCT]	支持
2	MULTISET EXCEPT [ALL DISTINCT]	支持

序号	Oracle数据库	GaussDB数据库
3	MULTISET INTERSECT [ALL DISTINCT]	支持

表 3-54 集合类型函数

序号	Oracle数据库	GaussDB数 据库	差异
1	exists(idx)	支持	-
2	extend[(count[, idx])]	支持,有差 异	GaussDB仅支持nesttable类型。
3	delete[(idx1[, idx2])]	支持	-
4	trim[(n)]	支持,有差 异	GaussDB仅支持nesttable类型。
5	count	支持	-
6	first	支持	-
7	last	支持	-
8	prior(idx)	支持	-
9	next(idx)	支持	-
10	limit	支持,有差 异	GaussDB仅支持nesttable类型。

表 3-55 record 变量操作

序号	Oracle数据库	GaussDB数据库
1	构造器	支持
2	%ROWTYPE声明变量	支持
3	定义常量constant	不支持

表 3-56 集合相关函数

序号	Oracle数据库	GaussDB数据库	差异
1	unnest_table(anynesttable)	支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	unnest_table(anyindexbyta ble)	支持	-
3	table(anyarray)	不支持	GaussDB使用 unnest(anyarray)函数进行 等价改写。

3.9.5 静态 SQL

表 3-57 静态查询 SQL 语句

序号	Oracle数 据库	GaussDB数 据库	差异
1	SELECT	支持,有差异	GaussDB和Oracle在某些场景下有不同。 GaussDB中不同事务中的共享锁在如下场景中不会互相阻塞: SELECT FOR SHARE - SELECT FOR SHARE; SELECT FOR SHARE - SELECT FOR KEY SHARE; SELECT FOR KEY SHARE - SELECT FOR KEY SHARE; SELECT FOR KEY SHARE - SELECT FOR NO
			KEY UPDATE; 上述场景中,由于锁与锁之间未阻塞,对在其 他事务中存在非阻塞锁的数据指定SKIP LOCKED时,锁不会被跳过。

表 3-58 静态 DML SQL 语句

序号	Oracle数 据库	GaussDB数 据库	差异
1	INSERT	支持	-
2	UPDATE	支持	-
3	DELETE	支持	-
4	MERGE	支持	-
5	LOCK TABLE	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
6	INSERT ALL	支持,有差 异	● Oracle不支持对into_clause的表设置别名, GaussDB支持。
			● into_clause指定sequence:
			- Oracle:首次引用nextval会生成下一个数字,但所有非首次引用的nextval都将返回相同数字。
			– GaussDB:引用nextval生成的数字可以正 常自增。
			● Oracle设置plan_hint语句可以正常生效, GaussDB不生效。

表 3-59 静态 TCL SQL 语句

序号	Oracle数据 库	GaussDB数 据库	差异
1	COMMIT	支持	-
2	ROLLBACK	支持	-
3	SAVEPOINT	支持	-
4	SET TRANSACTI ON	支持,有差 异	GaussDB不支持NAME string语法、USE ROLLBACK SEGMENT rollback_segment语 法。

表 3-60 伪列

序号	Oracle数据库	GaussDB 数据库	差异
1	CURRVAL and NEXTVAL	支持	-
2	LEVEL	支持	-
3	OBJECT_VALU E	不支持	-
4	ROWID	不支持	-
5	ROWNUM	支持,有 差异	不推荐ROWNUM条件用于JOIN ON子句。 GaussDB中ROWNUM条件用于JOIN ON子句 时在LEFT JOIN、RIGHT JOIN、FULL JOIN场 景下和MERGE INTO场景下与其他数据库行 为不一致,直接进行业务迁移存在风险。

表 3-61 隐式游标属性

序号	Oracle数据库	GaussDB 数据库	差异
1	SQL%FOUND	支持,有 差异	GaussDB在commit\rollback之后不刷新隐式 游标结果,Oracle会在commit\rollback之后
2	SQL %NOTFOUND	支持,有 差异	刷新隐式游标结果。
3	SQL %ROWCOUN T	支持,有 差异	
4	SQL%ISOPEN	支持,有 差异	
5	SQL %BULK_ROW COUNT	不支持	
6	SQL %BULK_EXCEP TIONS	支持,有 差异	

表 3-62 显式游标语法及关键字

序号	Oracle数据库	GaussDB数 据库	差异
1	CURSOR cursor_name [parameter_list] RETURN return_type;	支持	-
2	CURSOR cursor_name [parameter_list] [RETURN return_type] IS select_statement;	支持	-
3	OPEN	支持	-
4	CLOSE	支持,有差 异	GaussDB在exception内部会自动关闭,Oracle在exception内部不会自动关闭。
5	FETCH	支持	-
6	CURRENT OF CURSOR	支持	-

表 3-63 显式游标属性

序号	Oracle数据库	GaussDB数据库
1	SQL%FOUND	支持
2	SQL%NOTFOUND	支持
3	SQL%ROWCOUNT	支持
4	SQL%ISOPEN	支持

表 3-64 游标循环

序号	Oracle数据库	GaussDB数据库
1	FOR LOOP	支持

表 3-65 自治事务支持场景

序号	Oracle数据库	GaussDB数据库
1	存储过程	支持
2	匿名块	支持
3	函数	支持
4	Package	支持

3.9.6 动态 SQL

表 3-66 动态 SQL 语句执行方式

序号	Oracle数据库	GaussDB数 据库	差异
1	EXECUTE IMMEDIATE	支持,有差异	 GaussDB通过dynamic_sql_compat参数 控制同名变量是否读取同一参数,并且 检查调用存储过程时绑定参数出入参类 型是否与语句参数类型一致。 GaussDB不支持调用匿名块中部分绑定
			参数场景,例如匿名块中嵌套动态语句,使用表达式绑定参数,具体请参见《 开发指南》中"存储过程 > 动态语句 > 动态调用匿名块"章节。
			● GaussDB不支持RETURNING/RETURN INTO。

序号	Oracle数据库	GaussDB数 据库	差异
2	OPEN FOR \ FETCH \ CLOSE	支持	-

3.9.7 Trigger

表 3-67 trigger 类型

序号	Oracle数据库	GaussDB数 据库	差异
1	DML TRIGGER	支持,有差异	GaussDB不支持Compound DML Triggers。
2	SYSTEM TRIGGER	不支持	-

表 3-68 create trigger

序号	Oracle数据库	GaussDB 数据库	差异
1	create语法: CREATE [OR REPLACE] [EDITIONABLE NONEDITIONABLE] TRIGGER plsql_trigger_source	支持,有 差异	GaussDB不支持EDITIONABLE NONEDITIONABLE,支持 plsql_trigger_source部分行为。
2	plsql_trigger_source ::=语 法: [schema.] trigger_name [sharing_clause] [default_collation_claus e] { simple_dml_trigger instead_of_dml_trigger compound_dml_trigger system_trigger }	支持,有差异	GaussDB不支持schema、 sharing_clause、 default_collation_clause。

序号	Oracle数据库	GaussDB 数据库	差异
3	simple_dml_trigger ::=语法: { BEFORE AFTER } dml_event_clause [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [ENABLE DISABLE] [WHEN (condition)] trigger_body	支持,有差异	GaussDB不支持 referencing_clause、 referencing_clause(用from referencing_table代替)、 trigger_edition_clause、 trigger_ordering_clause、 ENABLE DISABLE; 支持 trigger_body部分行为。 GaussDB在没有INSTEAD OF TRIGGER的视图上创建语句级 BEFORE/AFTER TRIGGER时不会报错,执行DML时报错。
4	dml_event_clause ::=语 法: { DELETE INSERT UPDATE [OF column [, column]] } [OR { DELETE INSERT UPDATE [OF column [, column]] } ON [schema.] { table view }	不支持	-
5	trigger_body ::=语法: { plsql_block CALL routine_clause }	支持,有 差异	GaussDB的plsql_block 不允许声明为PRAGMA AUTONOMOUS_TRANSACTION。对于第二个分支,支持类似语法,具体为EXECUTE PROCEDURE function_name(arguments);方式执行function,并且function需要用户定义,必须声明为不带参数并返回类型为触发器,在触发器触发时执行。

序号	Oracle数据库	GaussDB 数据库	差异
6	instead_of_dml_trigger :: =语法: INSTEAD OF { DELETE INSERT UPDATE } [OR { DELETE INSERT UPDATE }] ON [NESTED TABLE nested_table_column OF] [schema.] noneditioning_view [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [trigger_ordering_clause] [ENABLE DISABLE] trigger_body	支持,有差异	GaussDB不支持NESTED TABLE nested_table_column OF、 referencing_clause、 trigger_edition_clause、 trigger_ordering_clause、 ENABLE DISABLE。
7	compound_dml_trigger :: =语法: CREATE trigger FOR dml_event_clause ON view COMPOUND TRIGGER INSTEAD OF EACH ROW IS BEGIN statement; END INSTEAD OF EACH ROW;	不支持	
8	system_trigger ::=语法: { BEFORE AFTER INSTEAD OF } { ddl_event [OR ddl_event] database_event [OR database_event] } ON { [schema.] SCHEMA [PLUGGABLE] DATABASE } [trigger_ordering_clause] [ENABLE DISABLE] trigger_body	不支持	-

表 3-69 alter trigger

序号	Oracle数据库	GaussDB 数据库	差异
1	ALTER TRIGGER [schema.] trigger_name { trigger_compile_clause { ENABLE DISABLE } RENAME TO new_name { EDITIONABLE NONEDITIONABLE } };	支持,有	GaussDB不支持schema、 trigger_compile_clause、 { ENABLE DISABLE }、 { EDITIONABLE NONEDITIONABLE }。

表 3-70 drop trigger

序号	Oracle数据库	GaussDB数 据库	差异
1	DROP TRIGGER [schema.] trigger ;	支持,有差 异	GaussDB不支持schema,需要在 trigger_name后面加上ON table_name。

Oracle数据库名为*_TRIGGERS的视图统计了trigger的相关信息,GaussDB相关视图与Oracle存在差异,GaussDB视图具体请参见《开发指南》中"系统表和系统视图 > 系统视图 > 其他视图"中的DB_TRIGGERS、ADM_TRIGGERS、MY_TRIGGERS章节。

表 3-71 Nested, Package, and Standalone Subprograms 兼容性

序号	Oracle数据库	GaussDB 数据库	差异
1	nested subprogram (子 块)	支持,有 差异	不支持重载,不支持定义为自治 事务,不支持SETOF的使用,仅 限一个限定符引用嵌套子程序或 嵌套子程序的变量。
2	package subprogram	支持	-
3	standalone subprogram (包含Function & Procedure)	支持	-
4	匿名块	支持	-

表 3-72 RETURN 语句支持情况

序号	Oracle数据库	GaussDB数据库
1	Function	支持
2	Procedure	支持
3	匿名块	支持

表 3-73 Function 相关参数

序号	Oracle数据库	GaussDB数据 库	差异
1	DETERMINISTI C	支持,有差异	GaussDB中为IMMUTABLE。
2	PARALLEL_EN ABLE	不支持	-
3	PIPELINED	不支持	-
4	RESULT_CACH E	不支持	-

表 3-74 参数形式支持

序号	Oracle数据库	GaussDB数据库
1	IN	支持
2	OUT	支持
3	IN OUT	支持

表 3-75 CREATE 语句

序号	Oracle数据库	GaussDB 数据库	差异
1	CREATE FUNCTION	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、 不支持sharing_clause、仅支持部分指定 function属性的子句(属性的子句仅支持 invoker_rights_clause子句)、不支持关键 字[EDITIONABLE NONEDITIONABLE]。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > C >CREATE FUNCTION"章 节。

序号	Oracle数据库	GaussDB 数据库	差异
2	CREATE LIBRARY	不支持	-
3	CREATE PACKAGE	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、 不支持sharing_clause、仅支持部分指定 package属性的子句(属性的子句仅支持 invoker_rights_clause子句)、不支持关键 字[EDITIONABLE NONEDITIONABLE]。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > C >CREATE PACKAGE "章 节。
4	CREATE PACKAGE BODY	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、 不支持sharing_clause、不支持关键字 [EDITIONABLE NONEDITIONABLE]。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > C >CREATE PACKAGE "章 节。
5	CREATE PROCEDURE	支持,有 差异	GaussDB不支持IF NOT EXISTS语法、 不支持sharing_clause以及后面的子句、不 支持关键字[EDITIONABLE NONEDITIONABLE]。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > C > CREATE PROCEDURE "章 节。
6	CREATE TRIGGER	支持,有 差异	GaussDB的具体语法请参见《开发指南》 中"SQL参考 > SQL语法 > C >CREATE TRIGGER "章节。
7	CREATE TYPE	支持,有 差异	GaussDB不支持varray、object类型、 UNDER语法。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > C >CREATE TYPE "章节。
8	CREATE TYPE BODY	不支持	-

表 3-76 ALTER 语句

序号	Oracle数据库	GaussDB数 据库	差异
1	ALTER FUNCTION	支持,有差 异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、 SETTINGS、DEBUG。
			具体语法请参见《开发指南》中"SQL参考 > SQL语法 > A > ALTER FUNCTION "章节。
2	ALTER LIBRARY	不支持	-
3	ALTER PACKAGE	支持,有差 异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、 SETTINGS、DEBUG。
			具体语法请参见《开发指南》中"SQL参考 > SQL语法 > A > ALTER PACKAGE "章节。
4	ALTER PROCEDURE	支持,有差 异	GaussDB不支持关键字[EDITIONABLE NONEDITIONABLE]、REUSE 、 SETTINGS、DEBUG。
			具体语法请参见《开发指南》中"SQL参考 > SQL语法 > A > ALTER PROCEDURE "章 节。
5	ALTER TRIGGER	支持,有差 异	GaussDB仅支持修改trigger名字。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > A > ALTER TRIGGER "章节。
6	ALTER TYPE	支持,有差 异	GaussDB仅支持部分语句。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > A > ALTER TYPE "章节。

表 3-77 DROP 语句

序号	Oracle数据 库	GaussDB数 据库	差异
1	DROP FUNCTION	支持	-
2	DROP LIBRARY	不支持	-
3	DROP PACKAGE	支持	-
4	DROP PROCEDURE	支持	-

序号	Oracle数据 库	GaussDB数 据库	差异
5	DROP TRIGGER	支持,有差 异	GaussDB的语法不同。 具体语法可参考:请参见《开发指南》中" SQL参考 > SQL语法 > D > DROP TRIGGER "章节。
6	DROP TYPE	支持,有差异	GaussDB不支持关键字FORCE、VALIDATE。 具体语法请参见《开发指南》中"SQL参考 > SQL语法 > D > DROP TYPE"章节。
7	DROP TYPE BODY	不支持	-

表 3-78 Function、Procedure、匿名块相关关键字

序号	Oracle数据 库	GaussDB数 据库	差异
1	ACCESSIBLE BY	不支持	-
2	AGGREGATE	支持,有差 异	 GaussDB不支持Oracle的aggregate using [schema.] implementation_type用法。 GaussDB的具体用法请参见《开发指南》中"SQL参考 > SQL语法 > C > CREATE AGGREGATE"章节。 语法不同,但实现功能相同。
3	DETERMINIS TIC	支持,有差 异	GaussDB仅在语法上支持关键字 DETERMINISTIC,未实现功能。
4	PIPE ROW	不支持	-
5	PIPELINED	不支持	-
6	SQL_MACRO	不支持	-
7	RESTRICT_R EFERENCES	不支持	-
8	INLINE	不支持	-

表 3-79 异常处理相关关键字

序号	Oracle数据 库	GaussDB数 据库	差异
1	EXCEPTION_ INIT	支持,有差 异	GaussDB不支持与系统错误码进行绑定
2	Exception	支持	-
3	Exception Handler	支持	-
4	SQLCODE	支持	-
5	SQLERRM	支持	-

3.10 系统函数

兼容函数分为:单行函数、用户自定义函数、AGG函数、分析函数、对象引用函数、模型函数、OLAP函数。

3.10.1 单行函数

序号	Oracle数据库	GaussDB数据库
1	数值函数	支持,有差异
2	返回字符值的字符函数	支持,有差异
3	返回数值的字符函数	支持,有差异
4	字符集函数	不支持
5	Collation函数	不支持
6	日期时间函数	支持,有差异
7	通用比较函数	支持,有差异
8	转换函数	支持,有差异
9	大对象函数	支持,有差异
10	集合函数	不支持
11	层次函数	支持
12	数据挖掘功能函数	不支持
13	XML类型函数	支持,有差异
14	JSON函数	不支持
15	编码解码函数	支持,有差异

序号	Oracle数据库	GaussDB数据库
16	空值相关的函数	支持
17	环境和标识符函数	支持,有差异

表 3-80 数值函数

序号	Oracle数 据库	GaussDB数 据库	差异
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,有差 异	 返回类型不一致,Oracle数据库类型包括BINARY_DOUBLE,BINARY_FLOAT,NUMBER;GaussDB返回类型包括int2,int4,int8,numeric。 当第一个入参为数值类型时,第二个参数必须为int、numeric类型或能够转换为numeric的类型。在a_format_version为10c,a_format_dev_version为s6时,当第一个参数为能够转为numeric的text类型时,第二个参数必须为不超过int4的数值类型。
15	NANVL	支持,有差 异	GaussDB不支持直接声明或浮点数除0得到NaN
16	POWER	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
17	REMAIND ER	支持,有差 异	返回值数据类型不同 GaussDB: ● 当一个输入是float4时,另一个是numeric 时,返回float4类型。 ● 当两个输入都是float4时,返回float4类型。 ● 当两个输入都是float8时,返回float8类型。 ● 其他数据类型,返回numeric。 Oracle: 返回值类型是number。
18	ROUND	支持,有差异	 第一个参数n的float类型GaussDB存在精度损失,比Oracle数据库精度低。 返回类型不一致。round(n, integer)形式,Oracle数据库NUMBER类型,GaussDB返回numeric类型;round(n)形式,Oracle数据库n的数据类型,GaussDB只能返回float8和numeric类型,缺少float4返回类型。 GaussDB判断入参有null,执行框架返回null的逻辑与Oracle数据库不一致SELECT round(NULL,'q');Oracle数据库null,GaussDB报错invalid input syntax for integer: "q"。
19	SIGN	支持	-
20	SIN	支持	-
21	SINH	支持	-
22	SQRT	支持	-
23	TAN	支持	-
24	TANH	支持,有差 异	返回值数据类型不同gaussdb 当输入是float8时,返回float8类型; 当输入是numeric时,返回numeric类型; oracle 返回值类型是number。
25	TRUNC	支持	-
26	WIDTH_B UCKET	支持	-

表 3-81 返回字符值的字符函数

序号	Oracle数 据库	GaussDB数 据库	差异
1	CHR	支持,有差 异	 输入的数字不符合现有字符集时,在JDBC下 GaussDB会报错,Oracle数据库会返回乱 码。 输入0、256等时Oracle数据库会返回Ascii码 为0的字符,GaussDB会在'\0;处截断。
2	CONCAT	支持	-
3	INITCAP	支持,有差 异	返回值受限于数据库字符集,导致返回结果与 Oracle数据库不一致。
4	LOWER	支持,有差 异	返回值类型不一致,Oracle数据库和输入类型一致的数据类型。 对时间格式上隐式转换问题,输入时间类型
			时,隐式转换为字符串再进行lower操作。 SELECT LOWER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle返回10-DEC-12,GaussDB返回 2012-12-10 00:00:00。
			返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
5	LPAD	支持	-
6	LTRIM	支持,有差 异	返回值类型不一致,输入是字符数据类型时 Oracle返回VARCHAR2类型,输入是数据库创建 时指定的国家字符集时Oracle返回NVARCHAR2 类型,输入是LOB类型时Oracle返回LOB类型, GaussDB返回TEXT类型。
7	NCHR	支持,有差	● 返回值字节长度与Oracle数据库不一致。
		异 	返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
			● 返回入参对应的字节数组时,单个字节在 [0x80-0xFF]范围,会返回"?",Oracle数 据库返回"?"、或者不输出、或者会报错。
8	NLS_LO WER	支持,有差异	 返回值类型不同,输入是字符数据类型时 Oracle返回VARCHAR2类型,输入是LOB类 型时Oracle返回LOB类型,GaussDB返回 TEXT类型。
			 nlsparam参数Oracle数据库还可以传入除 nls_sort外的其他参数种类而不报错, GaussDB只支持nls_sort。
			返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。

序号	Oracle数 据库	GaussDB数 据库	差异
9	NLS_UPP ER	支持,有差 异。	 返回值类型不同,输入是字符数据类型时 Oracle返回VARCHAR2类型,输入是LOB类型时Oracle返回LOB类型,GaussDB返回TEXT类型。 nlsparam参数Oracle数据库还可以传入除nls_sort外的其他参数种类而不报错,GaussDB只支持nls_sort。 返回值受限于数据库字符集,导致返回结果
10	NII CCORT	±+±	与Oracle数据库不一致。
10	NLSSORT	支持	-
11	REGEXP_ REPLACE	支持,有差 异 	GaussDB入参source_char不支持NCLOB类型。入参match_param选项 'n' 含义有差异:
			GaussDB中 'n' 选项与 'm' 选项含义相同,表示采用多行模式匹配;而 Oracle 表示 (.) 能匹配 '\n' 字符,没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项,不需要指定选项。
			● 不同正则表达式匹配结果可能不一致。 SELECT REGEXP_REPLACE('abc01234xyz', '(.*?)(\d+)(.*)', '#', 'g') FROM DUAL; Oracle报错,GaussDB返回#####xyz。
			在UTF-8编码字符集下中文输入时匹配结果可能不一致。Oracle需要在GBK字符集实现中文字符串的正则表达式匹配。
			● 包含部分转义字符的正则表达式匹配结果可能不一致。 SELECT REGEXP_REPLACE('abcabc', '\abc', '#', 'g') FROM DUAL;
			Oracle报错,GaussDB返回abcabc。 • 匹配规则受aformat_regexp_match参数影响,具体影响规格请参见《开发指南》中 "SQL参考 > 函数和操作符 > 字符处理函数和操作符"章节REGEXP_REPLACE函数部分。
12	REGEXP_ SUBSTR	支持,有差 异	匹配规则受aformat_regexp_match参数影响, 具体影响规格请参见《开发指南》中"SQL参考 > 函数和操作符 > 字符处理函数和操作符"章节 REGEXP_SUBSTR函数部分。
13	REPLACE	支持	-
14	RPAD	支持	-
15	RTRIM	支持	-
16	SUBSTR	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
17	TRANSLA TE	支持	-
18	TRIM	支持	-
19	UPPER	支持,有差 异	返回值类型不一致,Oracle数据库和输入类型一致的数据类型,GaussDB返回TEXT类型。
			 对时间格式上隐式转换问题,输入时间类型时,隐式转换为字符串再进行upper操作。 SELECT UPPER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle返回10-DEC-12, GaussDB返回 2012-12-10 00:00:00。
			返回值受限于数据库字符集,导致返回结果与Oracle数据库不一致。
20	INSTRB	支持	-

表 3-82 返回数值的字符函数

序号	Oracle数 据库	GaussDB数 据库	差异
1	ASCII	支持,有差 异	返回值类型不同。Oracle数据库返回类型为 uint4,GaussDB为int4。
2	INSTR	支持	-
3	LENGTH	支持	-

序号	Oracle数 据库	GaussDB数 据库	差异
4	REGEXP_ COUNT	支持,有差 异	● GaussDB入参source_char不支持NCLOB类型。
			 入参match_param选项 'n' 含义有差异: GaussDB中 'n' 选项与 'm' 选项含义相同,表示采用多行模式匹配;而 Oracle 表示 (.) 能匹配 '\n' 字符,没有指定该选项时默认不能匹配 '\n' 字符。GaussDB中 (.) 默认匹配 '\n' 选项,不需要指定选项。
			● 不同正则表达式匹配结果可能不一致。
			在UTF-8编码字符集下中文输入时匹配结果 可能不一致。Oracle需要在GBK字符集实现 中文字符串的正则表达式匹配。
			包含部分转义字符的正则表达式匹配结果可能不一致
			 匹配规则受aformat_regexp_match参数影响,具体影响规格请参见《开发指南》中 "SQL参考 > 函数和操作符 > 字符处理函数和操作符"章节REGEXP_COUNT函数部分。
5	REGEXP_I NSTR	支持,有差 异	匹配规则受aformat_regexp_match参数影响, 具体影响规格请参见《开发指南》中"SQL参考 > 函数和操作符 > 字符处理函数和操作符"章 节REGEXP_INSTR函数部分。
6	LENGTHC	支持	-

表 3-83 日期时间函数

序号	Oracle数据库	GaussDB 数据库	差异
1	ADD_MONTH S	支持,有差 异	 公元后到公元前,GaussDB会和Oracle数据库相差1年。 GaussDB的计算结果范围可以到-4714年,Oracle数据库只到-4713年。
2	CURRENT_DA TE	支持,有差 异	GaussDB不支持nls_date_format参数设置时间显示格式。
3	CURRENT_TI MESTAMP	支持,有差 异	Oracle数据库参数支持范围(0~9)。 GaussDB支持范围(0~6),微秒末位零不显示。
4	DBTIMEZONE	支持,有差 异	GaussDB不支持自带tz的timestamp类型接口的调用。
5	EXTRACT	支持	-

序号	Oracle数据库	GaussDB 数据库	差异
6	LAST_DAY	支持,有差 异	返回值类型不一致。GaussDB返回值类型为 timestamp without time zone,Oracle返回 值类型为date。
7	LOCALTIMEST AMP	支持,有差 异	Oracle数据库参数支持范围(0~9)。 GaussDB支持范围(0~6),微秒末位零不显示。
8	MONTHS_BE TWEEN	支持,有差 异	入参类型不一致。GaussDB入参均为 timestamp without time zone类型,Oracle 入参均为date类型。
9	NEW_TIME	支持,有差 异	new_time函数的第一个入参为字面量时,字 面量的格式以及函数的返回值类型均与 Oracle数据库不一致。
10	NEXT_DAY	支持	-
11	NUMTODSIN TERVAL	支持,有差 异	GaussDB不支持dsinterval类型,暂时用 interval兼容dsinterval类型。
12	NUMTOYMIN TERVAL	支持,有差 异	GaussDB不支持yminterval类型,暂时用 interval兼容yminterval类型。
13	SESSIONTIME ZONE	支持,有差 异	 赋值语法差异: GaussDB为set session time zone 8。Oracle为alter session set time_zone= '+08:00'。 默认值差异: GaussDB为时区名称形式如:PRC。Oracle为偏移量形式,如:+08:00。
14	SYS_EXTRACT _UTC	支持	-
15	SYSDATE	支持,有差 异	返回值类型不一致。GaussDB返回值类型为 timestamp without time zone,Oracle返回 值类型为date。
16	SYSTIMESTA MP	支持,有差 异	GaussDB毫秒计算只支持6位,Oracle数据库 支持9位。
17	TO_CHAR	支持,有差 异	fmt'5'未在Oracle数据库文档中,未适配。
18	TO_DSINTER VAL	支持,有差 异	GaussDB不支持dsinterval类型,暂时用 interval兼容dsinterval类型。
19	TO_TIMESTA MP	支持,有差 异	GaussDB毫秒计算只支持6位,Oracle数据库 支持9位。

序号	Oracle数据库	GaussDB 数据库	差异
20	TO_TIMESTA MP_TZ	支持,有差 异	GaussDB的timestamptz等价于Oracle的 timestampwithloacltimezone,缺少Oracle 对应的timestamptz类型。 nls_date_language只支持ENGLISH和 AMERICAN两种语言。
21	TO_YMINTER VAL	支持,有差 异	GaussDB不支持yminterval类型,暂时用 interval兼容yminterval类型。
22	TRUNC	支持,有差 异	GaussDB返回的类型与第一个入参的类型保持一致,Oracle始终返回date类型,另外支持指定的format也有区别,具体支持的列表详见《开发指南》的"SQL参考 > 函数和操作符 > 时间和日期处理函数和操作符"章节。
23	TZ_OFFSET	支持,有差 异	接收一个时区名称为入参的时候,时区名称 的类型比Oracle数据库要少。

表 3-84 通用比较函数

序号	Oracle数 据库	GaussDB数据 库	差异
1	GREATEST	支持,有差异	GaussDB不支持NLS_SORT参数指定的比较方式,只支持二进制比较。GaussDB不支持多语种的表达式。
2	LEAST	支持,有差异	GaussDB不支持NLS_SORT参数指定的比较方式,只支持二进制比较。GaussDB不支持多语种的表达式。

表 3-85 转换函数

序号	Oracle数据 库	GaussDB数 据库	差异
1	ASCIISTR	支持	-
2	CAST	支持,有差 异	GaussDB不支持multiset子句。GaussDB不支持nlsparam参数。
3	HEXTORAW	支持	-
4	RAWTOHEX	支持	-

序号	Oracle数据 库	GaussDB数 据库	差异
5	TO_BINARY _DOUBLE	支持,有差 异	GaussDB不支持nlsparam参数。
6	TO_BINARY _FLOAT	支持,有差 异	GaussDB不支持nlsparam参数。
7	TO_BLOB	支持,有差 异	GaussDB不支持long raw类型。GaussDB不支持bfile、mime_type类型。
8	TO_CLOB	支持	-
9	TO_DATE	支持,有差 异	 不支持多语种参数。 返回类型不一致。 缺少控制参数NLS_DATE_FORMAT。 部分format格式不支持。 fmt = 'j'。 1582年10月15日之前Oracle数据库与GaussDB输出不一致。 无分割符时,不保证与Oracle数据库完全一致。如to_date('220725', 'yymmdd'),yy/rr按照固定长度4解析,会解析为2207年25月,25非法月份则会报错。
10	TO_MULTI_ BYTE	支持	-
11	TO_NCHAR	支持,有差 异	GaussDB:将入参的类型转换为text。 Oracle:将入参的类型转换为国家字符集 (national character set)。

序号	Oracle数据 库	GaussDB数 据库	差异
12	_		差异 GaussDB不支持NLS_PARAM参数 GaussDB与Oracle的fmt选项差异点描述: 1、\$ GaussDB不支持该fmt。 2、, (comma) GaussDB: , 可以出现在fmt的任意位置。 Oracle: • 在format中,逗号只能出现在整数部分,且不能出现在数字开头,在原数据中可以在数字的开头位置。 • 支持format中的逗号与原数据的逗号个数和位置不一致,但最后一个逗号的位置需一致。 • 原数据和Format中的连续逗号,等同于没有逗号。 • 当原数据中没有逗号时,format的最后一个逗号后面的数字位数需与原数据相等。 3、B GaussDB未实现该功能。 4、C GaussDB不支持NLS参数。 5、G GaussDB不支持NLS参数。 7、U GaussDB不支持NLS参数。 8、D
			GaussDB不支持NLS参数。

序号	Oracle数据 库	GaussDB数 据库	差异
			GaussDB未实现该功能。
			11、V
			GaussDB未实现该功能。
			12、FM
			Gauss当有fm时,允许format中的逗号与原 数据中多,不然需严格保持一致。
			Oracle 返回值保留前后的空格。
			13、EEEE
			GaussDB未实现该功能。
13	TO_SINGLE _BYTE	支持	-
14	TREAT	支持,有差 异	GaussDB不支持使用"."操作符取值,不支持转化为object类型。
15	UNISTR	支持,有差 异	GaussDB只支持UTF-8编码,Oracle数据库支持UTF-8和UTF-16编码。

表 3-86 大对象函数

序号	Oracle数据 库	GaussDB数 据库	差异
1	EMPTY_BL OB	支持	-
2	EMPTY_CL OB	支持,有差 异	GaussDB的clob类型不支持Oracle数据库中的 定位器概念。

表 3-87 层次函数

序号	Oracle数据库	GaussDB 数据库	差异
1	SYS_CONNEC T_BY_PATH	支持,有差 异	GaussDB中第一个入参指定的列的类型仅支持CHAR/VARCHAR/NVARCHAR2/TEXT/INT1/INT2/INT4/INT8/FLOAT4/FLOAT8/NUMERIC类型。当前该函数col输入不支持表达式输入,列内容与separator分隔符重复不支持报错。

表 3-88 XML 类型函数

序号	Oracle数据 库	GaussDB数 据库	差异
1	EXISTSNOD E	支持,有差 异	在入参有命名空间时,xpath和命名空间都需要定义别名。
2	EXTRACTVA LUE	支持,有差 异	目前仅支持xpath1.0版本。
3	SYS_XMLA GG	支持,有差 异	xmlagg的别名,可使用xmlagg代替。
4	XMLAGG	支持	-
5	XMLCOMM ENT	支持	-
6	XMLCONCA T	支持	-
7	XMLELEME NT	支持,有差 异	xmlelement和xmlattributes的name字段赋值为NULL时,行为与Oracle不一致。 xmlelement的name字段赋值为NULL时,结果显示name信息为空,且不显示属性信息。 xmlattributes的name字段赋值为NULL时,不显示属性信息。
8	XMLEXISTS	支持,有差 异	GaussDB入参为xml类型。
9	XMLFORES T	支持,有差 异	GaussDB返回值为xml类型。GaussDB不支持 EVALNAME语法。
10	XMLPARSE	支持,有差 异	GaussDB返回值为xml类型。GaussDB不支持 WELLFORMED语法。
11	XMLROOT	支持,有差 异	GaussDB返回值为xml类型。
12	JSON_OBJE CT	支持	-
13	XMLTABLE	支持,有差 异	GaussDB从xml中选取数据使用的为XPath 1.0 表达式,不支持声明默认命名空间,不支持多组输入及取别名,不支持省略传入数据的 passing_clause子句,不支持RETURNING SEQUENCE BY REF子句和(SEQUENCE) BY REF子句。
14	GETSTRING VAL	支持	-
15	GETCLOBV AL	支持	-

序号	Oracle数据 库	GaussDB数 据库	差异
16	XMLSEQUE NCE	支持	-

表 3-89 编码解码函数

序号	Oracle数 据库	GaussDB数据 库	差异
1	DECODE	支持	-
2	DUMP	支持,有差异	因存储格式不同,GaussDB数值和时间类型 返回结果和Oracle数据库不一致。如:
			● GaussDB中SELECT dump(123); 返回 Typ=23 Len=4: 123,0,0,0。
			● Oracle中SELECT dump(123) FROM dual; 返回Typ=2 Len=3: 194,2,24。
3	ORA_HAS	支持,有差异	GaussDB中有以下行为:
	H		● 时间类型的入参转换成字符串类型再进行 hash。
			● 不支持maxbucket参数。
4	VSIZE	支持,有差异	因存储格式不同,GaussDB数值和时间类型 返回结果和Oracle数据库不一致。如:
			● GaussDB中SELECT vsize(999);返回4。
			● Oracle中SELECT vsize(999) FROM dual; 返回3。

表 3-90 空值相关的函数

序号	Oracle数据库	GaussDB数据库
1	COALESCE	支持
2	LNNVL	支持
3	NULLIF	支持
4	NVL	支持
5	NVL2	支持

表 3-91 环境和标识符函数

序号	Oracle数据 库	GaussDB数据库	差异
1	SYS_CONT	支持,有差异	GaussDB对不支持的参数返回NULL。
	EXT		以下为不支持的参数列表:
			'action'
			'is_application_root'
			'is_application_pdb'
			'audited_cursorid'
			'authenticated_identity'
			'authentication_data'
			'authentication_method'
			'cdb_domain'
			'cdb_name'
			'client_identifier'
			'con_id'
			'con_name'
			'current_sql_length'
			'db_domain'
			'db_supplemental_log_level'
			'dblink_info'
			'drain_status'
			'entryid'
			'enterprise_identity'
			'fg_job_id'
			'global_uid'
			'identification_type'
			'instance'
			'is_dg_rolling_upgrade'
			'ldap_server_type'
			'module'
			'network_protocol'
			'nls_calendar'
			'nls_sort'
			'nls_territory'
			'oracle_home'
			'os_user'
			'platform_slash'
			'policy_invoker'
			'proxy_enterprise_identity'

序号	Oracle数据 库	GaussDB数据库	差异
			'proxy_user'
			'proxy_userid'
			'scheduler_job'
			'session_edition_id'
			'session_edition_name'
			'sessionid'
			'statementid'
			'terminal'
			'unified_audit_sessionid'
			'session_default_collation'
			'client_info'
			'bg_job_id'
			'client_program_name'
			'current_bind'
			'global_context_memory'
			'host'
			'current_sqln'
2	SYS_GUID	支持	-
3	USER	支持,有差异	返回值类型不一致,GaussDB返回值类型 为name,Oracle返回值类型为varchar2。

序号	Oracle数据 库	GaussDB数据库	差异
4	USERENV	支持,有差异	GaussDB对不支持的参数返回NULL。以下为不支持的参数列表: 'action' 'is_application_root' 'is_application_pdb' 'audited_cursorid' 'authenticated_identity' 'authentication_data' 'authentication_method' 'cdb_domain' 'cdb_name' 'client_identifier' 'con_id' 'con_name' 'current_sql_length' 'db_domain' 'db_supplemental_log_level' 'dblink_info' 'drain_status' 'entryid' 'enterprise_identity' 'fg_job_id' 'global_uid' 'identification_type' 'is_dg_rolling_upgrade' 'ldap_server_type' 'module' 'network_protocol' 'nls_calendar' 'nls_sort' 'nls_territory' 'oracle_home' 'os_user' 'plolicy_invoker' 'proxy_enterprise_identity'

序号	Oracle数据 库	GaussDB数据库	差异
			• 'proxy_user'
			• 'proxy_userid'
			• 'scheduler_job'
			• 'session_edition_id'
			'session_edition_name'
			• 'sessionid'
			• 'statementid'
			• 'terminal'
			'unified_audit_sessionid'
			'session_default_collation'
			'client_info'
			• 'bg_job_id'
			'client_program_name'
			'current_bind'
			'global_context_memory'
			• 'host'
			• 'current_sqln'

3.10.2 其它函数

序号	Oracle数据库	GaussDB数据库
1	聚合函数	支持
2	分析函数	支持
3	对象引用函数	不支持
4	模型函数	不支持
5	OLAP函数	不支持
6	数据盒功能函数	不支持
7	关于用户定义的函数	支持

表 3-92 聚合函数

序号	Oracle数据库	GaussDB数据 库	差异
1	AVG	支持	-
2	CORR	支持	-
3	COUNT	支持	-
4	COVAR_POP	支持	-
5	COVAR_SAM P	支持	-
6	CUME_DIST	支持	-
7	DENSE_RANK	支持	-
8	FIRST	支持	GaussDB使用KEEP的语法,兼容Oracle此功能。
9	GROUPING	支持	-
10	LAST	支持	GaussDB使用KEEP的语法,兼容Oracle此功能。
11	LISTAGG	支持	-
12	MAX	支持	-
13	MEDIAN	支持	-
14	MIN	支持	-
15	PERCENT_RA NK	支持	-
16	PERCENTILE_ CONT	支持	-
17	RANK	支持	-
18	REGR_ (Linear Regression)	支持	-
19	STDDEV	支持	-
20	STDDEV_POP	支持	-
21	STDDEV_SA MP	支持	-
22	SUM	支持	-
23	VAR_POP	支持	-
24	VAR_SAMP	支持	-

序号	Oracle数据库	GaussDB数据 库	差异
25	VARIANCE	支持	-
26	WM_CONCA T	支持	-

表 3-93 分析函数

序号	Oracle数据 库	GaussDB数据 库	差异
1	FIRST_VALUE	支持	-
2	LAG	支持	-
3	LAST_VALUE	支持	-
4	LEAD	支持	-
5	NTH_VALUE	支持,有差异	 Oracle: 支持FROM FIRST LAST语法格式。 GaussDB: 不支持FROM FIRST LAST语法格式。
6	NTILE	支持	-
7	ROW_NUMB ER	支持	-
8	RATIO_TO_R EPORT	支持	-

3.11 系统视图

GaussDB数据库兼容了部分Oracle数据库的系统视图,兼容的详细列表如下。 更多系统视图的字段说明信息请参考《开发指南》中"系统视图"章节。

表 3-94 支持视图列表

序号	Oracle数据库	GaussDB数据库
1	ALL_ALL_TABLES	DB_ALL_TABLES
2	ALL_COL_PRIVS	DB_COL_PRIVS
3	ALL_COLL_TYPES	DB_COLL_TYPES
4	ALL_ERRORS	DB_ERRORS

序号	Oracle数据库	GaussDB数据库	
5	ALL_IND_COLUMNS	DB_IND_COLUMNS	
6	ALL_COL_COMMENTS	DB_COL_COMMENTS	
7	ALL_CONS_COLUMNS	DB_CONS_COLUMNS	
8	ALL_CONSTRAINTS	DB_CONSTRAINTS	
9	ALL_DEPENDENCIES	DB_DEPENDENCIES	
10	ALL_DIRECTORIES	DB_DIRECTORIES	
11	ALL_IND_EXPRESSIONS	DB_IND_EXPRESSIONS	
12	ALL_IND_PARTITIONS	DB_IND_PARTITIONS	
13	ALL_INDEXES	DB_INDEXES	
14	ALL_IND_SUBPARTITIONS	DB_IND_SUBPARTITIONS	
15	ALL_OBJECTS	DB_OBJECTS	
16	ALL_PART_COL_STATISTICS	DB_PART_COL_STATISTICS	
17	ALL_PART_KEY_COLUMNS	DB_PART_KEY_COLUMNS	
18	ALL_PART_TABLES	DB_PART_TABLES	
19	ALL_SCHEDULER_JOB_ARGS	DB_SCHEDULER_JOB_ARGS	
20	ALL_SCHEDULER_PROGRAM_ ARGS	DB_SCHEDULER_PROGRAM_ARGS	
21	ALL_SEQUENCES	DB_SEQUENCES	
22	ALL_SUBPART_COL_STATISTI CS	DB_SUBPART_COL_STATISTICS	
23	ALL_SUBPART_KEY_COLUMN S	DB_SUBPART_KEY_COLUMNS	
24	ALL_SYNONYMS	DB_SYNONYMS	
25	ALL_TAB_COL_STATISTICS	DB_TAB_COL_STATISTICS	
26	ALL_TAB_COMMENTS	DB_TAB_COMMENTS	
27	ALL_TAB_HISTOGRAMS	DB_TAB_HISTOGRAMS	
28	ALL_TAB_STATS_HISTORY	DB_TAB_STATS_HISTORY	
29	ALL_TYPES	DB_TYPES	
30	ALL_PART_INDEXES	DB_PART_INDEXES	
31	ALL_PROCEDURES	DB_PROCEDURES	
32	ALL_SOURCE	DB_SOURCE	

序号	Oracle数据库	GaussDB数据库	
33	ALL_TAB_COLUMNS	DB_TAB_COLUMNS	
34	ALL_TAB_PARTITIONS	DB_TAB_PARTITIONS	
35	ALL_TAB_SUBPARTITIONS	DB_TAB_SUBPARTITIONS	
36	ALL_TABLES	DB_TABLES	
37	ALL_TRIGGERS	DB_TRIGGERS	
38	ALL_USERS	DB_USERS	
39	ALL_VIEWS	DB_VIEWS	
40	DBA_AUDIT_OBJECT	ADM_AUDIT_OBJECT	
41	DBA_AUDIT_SESSION	ADM_AUDIT_SESSION	
42	DBA_AUDIT_STATEMENT	ADM_AUDIT_STATEMENT	
43	DBA_AUDIT_TRAIL	ADM_AUDIT_TRAIL	
44	DBA_COL_COMMENTS	ADM_COL_COMMENTS	
45	DBA_COL_PRIVS	ADM_COL_PRIVS	
46	DBA_COLL_TYPES	ADM_COLL_TYPES	
47	DBA_ARGUMENTS	ADM_ARGUMENTS	
48	DBA_CONSTRAINTS	ADM_CONSTRAINTS	
49	DBA_DATA_FILES	ADM_DATA_FILES	
50	DBA_CONS_COLUMNS	ADM_CONS_COLUMNS	
51	DBA_DEPENDENCIES	ADM_DEPENDENCIES	
52	DBA_DIRECTORIES	ADM_DIRECTORIES	
53	DBA_PART_COL_STATISTICS	ADM_PART_COL_STATISTICS	
54	DBA_PART_TABLES	ADM_PART_TABLES	
55	DBA_RECYCLEBIN	ADM_RECYCLEBIN	
56	DBA_ROLE_PRIVS	ADM_ROLE_PRIVS	
57	DBA_ROLES	ADM_ROLES	
58	DBA_SCHEDULER_JOB_ARGS	ADM_SCHEDULER_JOB_ARGS	
59	DBA_SCHEDULER_PROGRAM S	ADM_SCHEDULER_PROGRAMS	
60	DBA_SCHEDULER_PROGRAM _ARGS	ADM_SCHEDULER_PROGRAM_ARGS	
61	DBA_HIST_SNAPSHOT	ADM_HIST_SNAPSHOT	

序号	Oracle数据库	GaussDB数据库	
62	DBA_HIST_SQL_PLAN	ADM_HIST_SQL_PLAN	
63	DBA_HIST_SQLSTAT	ADM_HIST_SQLSTAT	
64	DBA_HIST_SQLTEXT	ADM_HIST_SQLTEXT	
65	DBA_ILMDATAMOVEMENTP OLICIES	GS_ADM_ILMDATAMOVEMENTPOLICIES	
66	DBA_ILMEVALUATIONDETAIL S	GS_ADM_ILMEVALUATIONDETAILS	
67	DBA_ILMOBJECTS	GS_ADM_ILMOBJECTS	
68	DBA_ILMPARAMETERS	GS_ADM_ILMPARAMETERS	
69	DBA_ILMPOLICIES	GS_ADM_ILMPOLICIES	
70	DBA_ILMRESULTS	GS_ADM_ILMRESULTS	
71	DBA_ILMTASKS	GS_ADM_ILMTASKS	
72	DBA_IND_COLUMNS	ADM_IND_COLUMNS	
73	DBA_IND_EXPRESSIONS	ADM_IND_EXPRESSIONS	
74	DBA_IND_PARTITIONS	ADM_IND_PARTITIONS	
75	DBA_INDEXES	ADM_INDEXES	
76	DBA_OBJECTS	ADM_OBJECTS	
77	DBA_PART_INDEXES	ADM_PART_INDEXES	
78	DBA_PROCEDURES	ADM_PROCEDURES	
79	DBA_SCHEDULER_JOBS	ADM_SCHEDULER_JOBS	
80	DBA_SCHEDULER_RUNNING_ JOBS	ADM_SCHEDULER_RUNNING_JOBS	
81	DBA_SEGMENTS	ADM_SEGMENTS	
82	DBA_SEQUENCES	ADM_SEQUENCES	
83	DBA_SOURCE	ADM_SOURCE	
84	DBA_IND_SUBPARTITIONS	ADM_IND_SUBPARTITIONS	
85	DBA_SUBPART_COL_STATISTI	ADM_SUBPART_COL_STATISTICS	
86	DBA_SUBPART_KEY_COLUMN S	ADM_SUBPART_KEY_COLUMNS	
87	DBA_SYS_PRIVS	ADM_SYS_PRIVS	
88	DBA_TAB_COL_STATISTICS	ADM_TAB_COL_STATISTICS	

序号	Oracle数据库	GaussDB数据库	
89	DBA_TAB_HISTOGRAMS	ADM_TAB_HISTOGRAMS	
90	DBA_TAB_STATISTICS	ADM_TAB_STATISTICS	
91	DBA_TAB_STATS_HISTORY	ADM_TAB_STATS_HISTORY	
92	DBA_TABLESPACES	ADM_TABLESPACES	
93	DBA_TYPES	ADM_TYPES	
94	DBA_USERS	ADM_USERS	
95	DBA_SYNONYMS	ADM_SYNONYMS	
96	DBA_TAB_COLS	ADM_TAB_COLS	
97	DBA_TAB_COLUMNS	ADM_TAB_COLUMNS	
98	DBA_TAB_COMMENTS	ADM_TAB_COMMENTS	
99	DBA_TABLES	ADM_TABLES	
100	DBA_TAB_PARTITIONS	ADM_TAB_PARTITIONS	
101	DBA_TAB_SUBPARTITIONS	ADM_TAB_SUBPARTITIONS	
102	DBA_TRIGGERS	ADM_TRIGGERS	
103	DBA_TYPE_ATTRS	ADM_TYPE_ATTRS	
104	DBA_VIEWS	ADM_VIEWS	
105	ROLE_ROLE_PRIVS	ROLE_ROLE_PRIVS	
106	ROLE_SYS_PRIVS	ROLE_SYS_PRIVS	
107	ROLE_TAB_PRIVS	ROLE_TAB_PRIVS	
108	USER_COL_COMMENTS	MY_COL_COMMENTS	
109	USER_COL_PRIVS	MY_COL_PRIVS	
110	USER_COLL_TYPES	MY_COLL_TYPES	
111	USER_CONSTRAINTS	MY_CONSTRAINTS	
112	USER_DEPENDENCIES	MY_DEPENDENCIES	
113	DICT	DICT	
114	DICTIONARY	DICTIONARY	
115	DUAL	DUAL	
116	NLS_DATABASE_PARAMETER S	NLS_DATABASE_PARAMETERS	
117	NLS_INSTANCE_PARAMETERS	NLS_INSTANCE_PARAMETERS	

序号	Oracle数据库	GaussDB数据库	
118	PLAN_TABLE	PLAN_TABLE	
119	USER_ERRORS	MY_ERRORS	
120	USER_ILMDATAMOVEMENTP OLICIES	GS_MY_ILMDATAMOVEMENTPOLICIES	
121	USER_ILMEVALUATIONDETAI LS	GS_MY_ILMEVALUATIONDETAILS	
122	USER_ILMOBJECTS	GS_MY_ILMOBJECTS	
123	USER_ILMPOLICIES	GS_MY_ILMPOLICIES	
124	USER_ILMRESULTS	GS_MY_ILMRESULTS	
125	USER_ILMTASKS	GS_MY_ILMTASKS	
126	USER_IND_COLUMNS	MY_IND_COLUMNS	
127	USER_IND_EXPRESSIONS	MY_IND_EXPRESSIONS	
128	USER_IND_PARTITIONS	MY_IND_PARTITIONS	
129	USER_IND_SUBPARTITIONS	MY_IND_SUBPARTITIONS	
130	USER_INDEXES	MY_INDEXES	
131	USER_JOBS	MY_JOBS	
132	USER_OBJECTS	MY_OBJECTS	
133	USER_PART_COL_STATISTICS	MY_PART_COL_STATISTICS	
134	USER_PART_INDEXES	MY_PART_INDEXES	
135	USER_PART_TABLES	MY_PART_TABLES	
136	USER_PROCEDURES	MY_PROCEDURES	
137	USER_RECYCLEBIN	MY_RECYCLEBIN	
138	USER_SCHEDULER_JOB_ARGS	MY_SCHEDULER_JOB_ARGS	
139	USER_SCHEDULER_PROGRA M_ARGS	MY_SCHEDULER_PROGRAM_ARGS	
140	USER_SEQUENCES	MY_SEQUENCES	
141	USER_SOURCE	MY_SOURCE	
142	USER_SUBPART_KEY_COLUM NS	MY_SUBPART_KEY_COLUMNS	
143	USER_SYNONYMS	MY_SYNONYMS	
144	USER_SYS_PRIVS	MY_SYS_PRIVS	
145	USER_TAB_COL_STATISTICS	MY_TAB_COL_STATISTICS	

序号	Oracle数据库	GaussDB数据库	
146	USER_TAB_COLUMNS	MY_TAB_COLUMNS	
147	USER_TAB_COMMENTS	MY_TAB_COMMENTS	
148	USER_TAB_HISTOGRAMS	MY_TAB_HISTOGRAMS	
149	USER_TAB_PARTITIONS	MY_TAB_PARTITIONS	
150	USER_TAB_STATISTICS	MY_TAB_STATISTICS	
151	USER_TAB_STATS_HISTORY	MY_TAB_STATS_HISTORY	
152	USER_TAB_SUBPARTITIONS	MY_TAB_SUBPARTITIONS	
153	USER_TABLES	MY_TABLES	
154	USER_TABLESPACES	MY_TABLESPACES	
155	USER_TRIGGERS	MY_TRIGGERS	
156	USER_TYPE_ATTRS	MY_TYPE_ATTRS	
157	USER_TYPES	MY_TYPES	
158	USER_VIEWS	MY_VIEWS	
159	V\$GLOBAL_TRANSACTION	V\$GLOBAL_TRANSACTION	
160	V\$NLS_PARAMETERS	V\$NLS_PARAMETERS	
161	V\$SESSION_WAIT	V\$SESSION_WAIT	
162	V\$SYSSTAT	V\$SYSSTAT	
163	V\$SYSTEM_EVENT	V\$SYSTEM_EVENT	
164	V\$VERSION	V\$VERSION	
165	V\$INSTANCE	V_INSTANCE	
166	GV\$INSTANCE	GV_INSTANCE	
167	V\$MYSTAT	V_MYSTAT	
168	V\$SESSION	V_SESSION	
169	GV\$SESSION	GV_SESSION	
170	V\$SESSION_LONGOPS	DV_SESSION_LONGOPS	
171	V\$SESSION	DV_SESSIONS	
172	ALL_ARGUMENTS	DB_ARGUMENTS	
173	USER_CONS_COLUMNS	MY_CONS_COLUMNS	
174	USER_PART_KEY_COLUMNS	MY_PART_KEY_COLUMNS	

序号	Oracle数据库	GaussDB数据库
175	USER_SUBPART_COL_STATIST ICS	MY_SUBPART_COL_STATISTICS
176	USER_ROLE_PRIVS	MY_ROLE_PRIVS
177	DBA_TAB_PRIVS	ADM_TAB_PRIVS
178	USER_SCHEDULER_JOBS	MY_SCHEDULER_JOBS
179	V\$LOCK	V\$LOCK
180	V\$DBLINK	V\$DBLINK
181	V\$OPEN_CURSOR	V\$OPEN_CURSOR
182	ALL_TAB_PRIVS	DB_TAB_PRIVS
183	ALL_TAB_MODIFICATIONS	DB_TAB_MODIFICATIONS
184	USER_TAB_MODIFICATIONS	MY_TAB_MODIFICATIONS
185	USER_AUDIT_TRAIL	MY_AUDIT_TRAIL

3.12 高级包

GaussDB数据库兼容的高级包如表3-95所示。

表 3-95 支持高级包列表

序号	Oracle数据库	GaussDB数据 库	差异
1	DBMS_LOB	DBE_LOB	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_LOB"章节。
2	DBMS_RAND OM	DBE_RANDOM	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_RANDOM"章节。
3	DBMS_OUTP UT	DBE_OUTPUT	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_OUTPUT"章节。
4	UTL_RAW	DBE_RAW	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_RAW"章节。
5	DBMS_SCHED ULER	DBE_SCHEDUL ER	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_SCHEDULER"章节。

序号	Oracle数据库	GaussDB数据 库	差异
6	DBMS_UTILIT	DBE_UTILITY	GaussDB具体用法请参见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_UTILITY"章节。
7	DBMS_SQL	DBE_SQL	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_SQL"章节。
8	UTL_FILE	DBE_FILE	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_FILE"章节。
9	DBMS_SESSIO N	DBE_SESSION	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_SESSION"章节。
10	UTL_MATCH	DBE_MATCH	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_MATCH"章节。
11	DBMS_APPLIC ATION_INFO	DBE_APPLICAT ION_INFO	GaussDB具体用法请参见《开发指南》 中"存储过程 > 高级包 > 二次封装接口 (推荐) > DBE_APPLICATION_INFO"章节。
12	DBMS_XMLD OM	DBE_XMLDOM	GaussDB中具体信息请参见《开发指 南》中"存储过程 > 高级包 > 二次封装 接口(推荐) > DBE_XMLDOM"章 节。
13	DBMS_XMLPA RSER	DBE_XMLPARS ER	GaussDB中具体信息请参见《开发指 南》中"存储过程 > 高级包 > 二次封装 接口(推荐) > DBE_XMLPARSER"章 节。
14	DBMS_ILM	DBE_ILM	GaussDB中具体信息请参见《开发指 南》中"存储过程 > 高级包 > 二次封装 接口(推荐) > DBE_ILM"章节。
15	DBMS_ILM_A DMIN	DBE_ILM_ADM IN	GaussDB中具体信息请参见《开发指 南》中"存储过程 > 高级包 > 二次封装 接口(推荐) > DBE_ILM_ADMIN"章 节。
16	DBMS_COMP RESSION	DBE_COMPRE SSION	GaussDB中具体信息请参见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_COMPRESSION"章节。
17	DBMS_HEAT_ MAP	DBE_HEAT_MA P	GaussDB中具体信息请参见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_HEAT_MAP"章节。

序号	Oracle数据库	GaussDB数据 库	差异
18	DBMS_DESCRI BE	DBE_DESCRIBE	GaussDB中具体信息请参见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_DESCRIBE"章节。
19	DBMS_XMLGE N	DBE_XMLGEN	GaussDB中具体信息请参见《开发指 南》中"存储过程 > 高级包 > 二次封装 接口(推荐) > DBE_XMLGEN"章 节。
20	DBMS_STATS	DBE_STATS	GaussDB中具体信息请参见《开发指 南》中"存储过程 > 高级包 > 二次封装 接口(推荐) > DBE_STATS"章节。

表 3-96 DBMS_LOB 兼容性说明

序号	Oracle数据库	GaussDB数据 库	差异
1	APPEND Procedures	APPEND Procedures	-
2	CLOB2FILE Procedure	不支持	-
3	CLOSE Procedure	BFILECLOSE Procedure	GaussDB: 参数类型为BFILE,不存在函数重载。 Oracle: 该过程存在3个重载,3个重载的参数lob_loc、lob_loc和file_loc的类型分别为BLOB、CLOB CHARACTER SET ANY_CS和BFILE。
4	COMPARE Functions	COMPARE Functions	GaussDB:存在3个重载函数,对于第三个参数(len)均为BIGINT。 Oracle:存在3个重载函数,对于第三个参数(amount)均为INTEGER。
5	CONVERTTOB LOB Procedure	LOB_CONVERT TOBLOB Procedure	GaussDB: 该过程共有5个参数,且第3、4、5个参数类型为BIGINT。 Oracle: 该过程共有8个参数,在GaussDB所有参数的基础上增加了blob_csid、lang_context和warning3个参数,类型分别为NUMBER、INTEGER和INTEGER,且第3、4、5个参数类型为INTEGER。

序号	Oracle数据库	GaussDB数据 库	差异
6	CONVERTTOC LOB Procedure	LOB_CONVERT TOCLOB Procedure	GaussDB:该过程共有5个参数。第3、4、5个参数类型为BIGINT。 Oracle:该过程共有8个参数。第3、4、5个参数类型为INTEGER。Oracle的该过程在GaussDB所有参数的基础上增加了3个参数:blob_csid、lang_context和warning,参数类型分别为NUMBER、INTEGER和INTEGER。
7	COPY Procedures	LOB_COPY Functions	-
8	COPY_DBFS_L INK Procedures	不支持	-
9	COPY_FROM_ DBFS_LINK	不支持	-
10	CREATETEMP ORARY Procedures	CREATE_TEMP ORARY Procedures	GaussDB:该过程存在2个重载。第一个重载过程的第一个参数(lob_loc)为BLOB,第二个重载过程的第一个参数(lob_loc)为CLOB;两个重载过程的第三个参数(dur)为INTEGER,默认值为10。 Oracle:该过程存在2个重载。第一个重载过程的第一个参数(lob_loc)为BLOB,第二个重载过程的第一个参数(lob_loc)为CLOB;两个重载过程的第三个参数(dur)的参数类型为PLS_INTEGER,第一个重载过程的dur默认值为DBMS_LOB.SESSION,第二个重载过程的dur默认值为DBMS_LOB.SESSION,第二个重载过程的dur默认值为DBMS_LOB.SESSION,第二个重载过程的dur默认值为10。
11	DBFS_LINK_G ENERATE_PAT H Functions	不支持	-
12	ERASE Procedures	LOB_ERASE Procedures	-
13	FILECLOSE Procedure	不支持	-
14	FILECLOSEALL Procedure	不支持	-
15	FILEEXISTS Function	不支持	-
16	FILEGETNAME Procedure	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
17	FILEISOPEN Function	不支持	-
18	FILEOPEN Procedure	不支持	-
19	FRAGMENT_D ELETE Procedure	不支持	-
20	FRAGMENT_I NSERT Procedures	不支持	-
21	FRAGMENT_ MOVE Procedure	不支持	-
22	FRAGMENT_R EPLACE Procedures	不支持	-
23	FREETEMPOR ARY Procedures	不支持	-
24	GET_DBFS_LI NK Functions	不支持	-
25	GET_DBFS_LI NK_STATE Procedures	不支持	-
26	GETCHUNKSI ZE Functions	GETCHUNKSIZ E Functions	-
27	GETCONTENT TYPE Functions	不支持	-
28	GETLENGTH Functions	不支持	-
29	GETOPTIONS Functions	不支持	-
30	GET_STORAG E_LIMIT Function	不支持	-

序号	Oracle数据库	GaussDB数据 库	差异
31	INSTR Functions	MATCH Functions	GaussDB:存在3个重载函数。3个重载函数的第三、四个参数均为BIGINT。
			Oracle:存在3个重载函数。3个重载函数的第三、四个参数均为INTEGER。
32	ISOPEN Functions	不支持	-
33	ISREMOTE Function	不支持	-
34	ISSECUREFILE Function	不支持	-
35	ISTEMPORAR Y Functions	不支持	-
36	LOADBLOBFR OMFILE Procedure	LOADBLOBFRO MBFILE Procedure	-
37	LOADCLOBFR OMFILE Procedure	LOADCLOBFRO MBFILE Procedure	-
38	LOADFROMFI LE Procedure	LOADFROMBFI LE Procedure	-
39	MOVE_TO_DB FS_LINK Procedures	不支持	-
40	OPEN Procedures	BFILEOPEN Procedure	GaussDB: 该过程不存在重载。第一个参数(bfile)类型为DBE_LOB.BFILE,第二个参数(open_mode)类型为TEXT,且只支持read模式。 Oracle: 该过程存在3个重载。第一个重载过程的第一个参数(lob_loc)类型为NOCOPY BLOB,第二个参数(openmode)类型为BINARY_INTEGER;第二个重载过程的第一个参数(lob_loc)类型为NOCOPY CLOB CHARACTER SET ANY_CS,第二个参数(openmode)类型为BINARY_INTEGER;第三个重载过程的第一个参数(file_loc)类型为NOCOPY BFILE,第二个参数(openmode)类型为BINARY_INTEGER,且只能为file_readonly。

序号	Oracle数据库	GaussDB数据 库	差异
41	READ Procedures	READ Procedures	GaussDB:该过程存在2个重载。 Oracle:该过程存在3个重载。其中前两个重载与GaussDB无差异,第三个过程 重载包括4个参数:file_loc、amount、 offset和buffer,其类型分别为BFILE、 NOCOPY INTEGER、INTEGER和 RAW。
42	SET_DBFS_LIN K Procedures	不支持	-
43	SETCONTENT TYPE Procedure	不支持	-
44	SETOPTIONS Procedures	不支持	-
45	SUBSTR Functions	LOB_SUBSTR Functions	-
46	TRIM Procedures	STRIP Functions	GaussDB:该过程存在2个重载。两个重载过程的第二个参数(newlen)均为BIGINT。 Oracle:该过程存在2个重载。两个重载过程的第二个参数(newlen)均为INTEGER。
47	WRITE Procedures	WRITE Functions	-
48	WRITEAPPEN D Procedures	WRITEAPPEND Functions	-

表 3-97 DBMS_RANDOM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	INITIALIZE Procedure	不支持	-
2	NORMAL Function	不支持	-
3	RANDOM Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
4	SEED Procedures	DBE_RANDOM.SET_S EED Function	GaussDB:该函数无重载,参数 类型为INTEGER。
			Oracle:该过程存在2个重载,2 个重载过程的参数类型分别为 VARCHAR2和 BINARY_INTEGER。
5	STRING Function	不支持	-
6	TERMINATE Procedure	不支持	-
7	VALUE Functions	DBE_RANDOM.GET_V ALUE Function	GaussDB:该函数无重载。 Oracle:存在无参数的VALUE函数重载,返回NUMBER类型。

表 3-98 DBMS_OUTPUT 兼容性说明

序号	Oracle数据库	GaussDB数据 库	差异
1	DISABLE Procedure	DISABLE Function	-
2	ENABLE Procedure	ENABLE Function	-
3	GET_LINE Procedure	GET_LINE Function	-
4	GET_LINES Procedure	GET_LINES Function	GaussDB: 该函数无重载,首个参数 (lines)数据类型为VARCHAR[]。 Oracle: 该过程存在2个重载,2个重载 过程的首个参数(lines)分别为 CHARARR和 DBMSOUTPUT_LINESARRAY。
5	NEW_LINE Procedure	NEW_LINE Function	-

序号	Oracle数据库	GaussDB数据 库	差异
6	PUT Procedure	PUT Function	GaussDB: 当数据库服务端字符集 server_encoding不是UTF8编码格式且入 参的字符编码是合法的UTF8编码时,该 函数不会区分入参的数据类型,都会先 把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换 后再输出。
			Oracle: 当数据库服务端字符集 server_encoding不是UTF8编码格式且入 参的字符编码是合法的UTF8编码时,若 入参类型是NVARCHAR2,则该过程会先 把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换 后再输出;若入参为其他字符类型,则 会将该字符编码视作非法字符,以占位 符的形式输出。
7	PUT_LINE Procedure	PUT_LINE Function	GaussDB: 当数据库服务端字符集 server_encoding不是UTF8编码格式且入 参的字符编码是合法的UTF8编码时,该 函数不会区分入参的数据类型,都会先 把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换 后再输出。
			Oracle: 当数据库服务端字符集 server_encoding不是UTF8编码格式且入 参的字符编码是合法的UTF8编码时,若 入参类型是NVARCHAR2,则该过程会先 把该字符编码按照"UTF8 > server_encoding"的转换关系进行转换 后再输出;若入参为其他字符类型,则 会将该字符编码视作非法字符,以占位 符的形式输出。

表 3-99 UTL_RAW 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	BIT_AND Function	BIT_AND Function	-
2	BIT_COMPLEMENT Function	BIT_COMPLEMENT Function	-
3	BIT_OR Function	BIT_OR Function	GaussDB:两个参数类型被定义为TEXT类型并且返回TEXT类型。
			Oracle:两个参数为RAW类型 并且返回RAW类型。

序号	Oracle数据库	GaussDB数据库	差异
4	BIT_XOR Function	BIT_XOR Function	-
5	CAST_FROM_BINA RY_DOUBLE Function	CAST_FROM_BINA RY_DOUBLE_TO_R AW Function	-
6	CAST_FROM_BINA RY_FLOAT Function	CAST_FROM_BINA RY_FLOAT_TO_RA W Function	GaussDB:参数n为FLOAT4类型。 Oracle:参数n为FLOAT类型。
7	CAST_FROM_BINA RY_INTEGER Function	CAST_FROM_BINA RY_INTEGER_TO_R AW Function	GaussDB:参数value为 BIGINT类型。 Oracle:参数value为INTEGER 类型。
8	CAST_FROM_NUM BER Function	CAST_FROM_NUM BER_TO_RAW Function	GaussDB:参数n为NUMERIC 类型 Oracle:参数n为NUMBER类 型。
9	CAST_TO_BINARY_ DOUBLE Function	CAST_FROM_RAW _TO_BINARY_DOU BLE Function	-
10	CAST_TO_BINARY_ FLOAT Function	CAST_FROM_RAW _TO_BINARY_FLOA T Function	GaussDB: 函数返回类型为 FLOAT4。 Oracle: 函数返回类型为 FLOAT。
11	CAST_TO_BINARY_I NTEGER Function	CAST_FROM_RAW _TO_BINARY_INTE GER Function	GaussDB: 参数endianess为 INTEGER类型,函数返回类型 为INTEGER。 Oracle: 参数endianess为 PLS_INTEGER类型,函数返回 类型为BINARY_INTEGER。
12	CAST_TO_NUMBER Function	CAST_FROM_RAW _TO_NUMBER Function	GaussDB:函数返回类型为 NUMERIC。 Oracle:函数返回类型为 NUMBER。
13	CAST_TO_NVARCH AR2 Function	CAST_FROM_RAW _TO_NVARCHAR2 Function	-
14	CAST_TO_RAW Function	CAST_FROM_VARC HAR2_TO_RAW Function	-
15	CAST_TO_VARCHA R2 Function	CAST_TO_VARCHA R2 Function	-

序号	Oracle数据库	GaussDB数据库	差异
16	COMPARE Function	COMPARE Function	GaussDB: 函数返回类型为 INTEGER。 Oracle: 函数返回类型为 NUMBER。
17	CONCAT Function	CONCAT Function	-
18	CONVERT Function	CONVERT Function	-
19	COPIES Function	COPIES Function	GaussDB:参数n为NUMERIC 类型。 Oracle:参数n为NUMBER类型。
20	LENGTH Function	GET_LENGTH Function	GaussDB:函数返回类型为INTEGER。 Oracle:函数返回类型为NUMBER。
21	OVERLAY Function	OVERLAY Function	-
22	REVERSE Function	REVERSE Function	-
23	SUBSTR Function	SUBSTR Function	GaussDB:参数lob_loc为 BLOB类型;参数off_set为 INTEGER类型,默认值为1;参 数amount为INTEGER类型,默 认值为32767。
			Oracle:参数r为RAW类型,参数pos为BINARY_INTEGER类型且无默认值,参数len为BINARY_INTEGER类型,默认值为NULL。
24	TRANSLATE Function	TRANSLATE Function	-
25	TRANSLITERATE Function	TRANSLITERATE Function	-
26	XRANGE Function	XRANGE Function	GaussDB:参数start_byte和 end_byte无默认值。 Oracle:参数start_byte和 end_byte默认为NULL。

表 3-100 DBMS_SCHEDULER 兼容性说明

序号	Oracle数据库	GaussDB数据库
1	ADD_EVENT_QUEUE_SUBSCRI BER Procedure	不支持
2	ADD_GROUP_MEMBER Procedure	不支持
3	ADD_JOB_EMAIL_NOTIFICATI ON Procedure	不支持
4	ADD_TO_INCOMPATIBILITY Procedure	不支持
5	ALTER_CHAIN Procedure	不支持
6	ALTER_RUNNING_CHAIN Procedure	不支持
7	CLOSE_WINDOW Procedure	不支持
8	COPY_JOB Procedure	不支持
9	CREATE_CHAIN Procedure	不支持
10	CREATE_CREDENTIAL Procedure	CREATE_CREDENTIAL Procedure
11	CREATE_DATABASE_DESTINAT ION Procedure	不支持
12	CREATE_EVENT_SCHEDULE Procedure	不支持
13	CREATE_FILE_WATCHER Procedure	不支持
14	CREATE_GROUP Procedure	不支持
15	CREATE_INCOMPATIBILITY Procedure	不支持
16	CREATE_JOB Procedure	CREATE_JOB Procedure
17	CREATE_JOB_CLASS Procedure	CREATE_JOB_CLASS Procedure
18	CREATE_JOBS Procedure	不支持
19	CREATE_PROGRAM Procedure	CREATE_PROGRAM Procedure
20	CREATE_RESOURCE Procedure	不支持
21	CREATE_SCHEDULE Procedure	CREATE_SCHEDULE Procedure
22	CREATE_WINDOW Procedure	不支持
23	DEFINE_ANYDATA_ARGUMEN T Procedure	不支持

序号	Oracle数据库	GaussDB数据库
24	DEFINE_CHAIN_EVENT_STEP Procedure	不支持
25	DEFINE_CHAIN_RULE Procedure	不支持
26	DEFINE_CHAIN_STEP Procedure	不支持
27	DEFINE_METADATA_ARGUME NT Procedure	不支持
28	DEFINE_PROGRAM_ARGUME NT Procedure	DEFINE_PROGRAM_ARGUMENT Procedure
29	DISABLE Procedure	DISABLE Procedure
30	DROP_AGENT_DESTINATION Procedure	不支持
31	DROP_CHAIN Procedure	不支持
32	DROP_CHAIN_RULE Procedure	不支持
33	DROP_CHAIN_STEP Procedure	不支持
34	DROP_CREDENTIAL Procedure	DROP_CREDENTIAL Procedure
35	DROP_DATABASE_DESTINATI ON Procedure	不支持
36	DROP_FILE_WATCHER Procedure	不支持
37	DROP_GROUP Procedure	不支持
38	DROP_INCOMPATIBILITY Procedure	不支持
39	DROP_JOB Procedure	DROP_JOB Procedure
40	DROP_JOB_CLASS Procedure	DROP_JOB_CLASS Procedure
41	DROP_PROGRAM Procedure	DROP_PROGRAM Procedure
42	DROP_PROGRAM_ARGUMENT Procedure	不支持
43	DROP_SCHEDULE Procedure	DROP_SCHEDULE Procedure
44	DROP_WINDOW Procedure	不支持
45	ENABLE Procedure	ENABLE Procedure
46	END_DETACHED_JOB_RUN Procedure	不支持

序号	Oracle数据库	GaussDB数据库
47	EVALUATE_CALENDAR_STRIN G Procedure	EVALUATE_CALENDAR_STRING Procedure
48	EVALUATE_RUNNING_CHAIN Procedure	不支持
49	GENERATE_JOB_NAME Function	GENERATE_JOB_NAME Function
50	GET_AGENT_INFO Function	不支持
51	GET_AGENT_VERSION Function	不支持
52	GET_ATTRIBUTE Procedure	不支持
53	GET_FILE Procedure	不支持
54	GET_SCHEDULER_ATTRIBUTE Procedure	不支持
55	OPEN_WINDOW Procedure	不支持
56	PURGE_LOG Procedure	不支持
57	PUT_FILE Procedure	不支持
58	REMOVE_EVENT_QUEUE_SUB SCRIBER Procedure	不支持
59	REMOVE_FROM_INCOMPATIB ILITY Procedure	不支持
60	REMOVE_GROUP_MEMBER Procedure	不支持
61	REMOVE_JOB_EMAIL_NOTIFIC ATION Procedure	不支持
62	RESET_JOB_ARGUMENT_VALU E Procedure	不支持
63	RUN_CHAIN Procedure	不支持
64	RUN_JOB Procedure	RUN_JOB Procedure
65	SET_AGENT_REGISTRATION_P ASS Procedure	不支持
66	SET_ATTRIBUTE Procedure	SET_ATTRIBUTE Procedure
67	SET_ATTRIBUTE_NULL Procedure	不支持
68	SET_JOB_ANYDATA_VALUE Procedure	不支持

序号	Oracle数据库	GaussDB数据库
69	SET_JOB_ARGUMENT_VALUE Procedure	SET_JOB_ARGUMENT_VALUE Procedure
70	SET_JOB_ATTRIBUTES Procedure	不支持
71	SET_RESOURCE_CONSTRAINT Procedure	不支持
72	SET_SCHEDULER_ATTRIBUTE Procedure	不支持
73	STOP_JOB Procedure	STOP_JOB Procedure

表 3-101 DBMS_UTILITY 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	ACTIVE_INSTANCES Procedure	不支持	-
2	ANALYZE_DATABAS E Procedure	不支持	-
3	ANALYZE_PART_OBJ ECT Procedure	不支持	-
4	ANALYZE_SCHEMA Procedure	不支持	-
5	CANONICALIZE Procedure	CANONICALIZE Procedure	GaussDB:参数canon_len默认 为1024字节。 Oracle:参数canon_len无默认 值。
6	COMMA_TO_TABLE Procedures	COMMA_TO_TAB LE Procedure	GaussDB:参数tab为 VARCHAR2数组。 Oracle:该过程存在2个重载。 参数tab可以为两种类型之一: 一种为uncl_array,另一种为 lname_array。
7	COMPILE_SCHEMA Procedure	不支持	-
8	CREATE_ALTER_TYP E_ERROR_TABLE Procedure	不支持	-
9	CURRENT_INSTANC E Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
10	DATA_BLOCK_ADDR ESS_BLOCK Function	不支持	-
11	DATA_BLOCK_ADDR ESS_FILE Function	不支持	-
12	DB_VERSION Procedure	DB_VERSION Procedure	GaussDB: 只有参数version, 类型为VARCHAR2。Oracle: 有参数version和 compatibility,类型均为 VARCHAR2。
13	EXEC_DDL_STATEM ENT Procedure	EXEC_DDL_STATE MENT Function	GaussDB:参数parse_string为 TEXT类型。
			Oracle:参数parse_string为 VARCHAR2类型。
14	EXPAND_SQL_TEXT Procedure	EXPAND_SQL_TE XT Function	GaussDB: 参数 output_sql_text为CLOB。
			Oracle:参数 output_sql_text 为NOCOPY CLOB,通过传引 用方式传递OUT参数。
15	FORMAT_CALL_STA CK Function	FORMAT_CALL_S TACK Function	GaussDB: 函数返回类型为 TEXT。 Oracle: 函数返回类型为 VARCHAR2。
16	FORMAT_ERROR_B	FORMAT_ERROR_	GaussDB: 函数返回类型为
	ACKTRACE Function	BACKTRACE Function	TEXT。 Oracle: 函数返回类型为 VARCHAR2。
17	FORMAT_ERROR_ST ACK Function	FORMAT_ERROR_ STACK Function	GaussDB: 函数返回类型为 TEXT。
			Oracle:函数返回类型为 VARCHAR2。
18	GET_CPU_TIME Function	GET_CPU_TIME Function	GaussDB: 函数返回类型为 BIGINT。
			Oracle:函数返回类型为 NUMBER。
19	GET_DEPENDENCY Procedure	不支持	-
20	GET_ENDIANNESS Function	GET_ENDIANNES S Function	GaussDB: 函数返回类型为 INTEGER。
			Oracle:函数返回类型为 NUMBER。

序号	Oracle数据库	GaussDB数据库	差异
21	GET_HASH_VALUE Function	GET_HASH_VALU E Function	GaussDB:参数base、 hash_size和返回类型均为 INTEGER。
			Oracle:参数base、hash_size 和返回类型均为NUMBER。
22	GET_PARAMETER_V ALUE Function	不支持	-
23	GET_SQL_HASH Function	GET_SQL_HASH Function	GaussDB:参数last4bytes为BIGINT类型,代表MD5哈希值的最后四字节,以无符号整数形式展现,函数返回类型为BIGINT。
			Oracle:对应参数pre10ihash 为NUMBER类型,用于存储 MD5计算得到的16字节中的4 字节哈希值。
24	GET_TIME Function	GET_TIME Function	GaussDB: 函数返回类型为 BIGINT。 Oracle: 函数返回类型为 NUMBER。
25	GET_TZ_TRANSITIO NS Procedure	不支持	-
26	INVALIDATE Procedure	不支持	-
27	IS_BIT_SET Function	IS_BIT_SET Function	GaussDB:参数n和返回值类型为INTEGER。 Oracle:参数n和返回值类型为
			NUMBER。
28	IS_CLUSTER_DATAB ASE Function	IS_CLUSTER_DAT ABASE Function	-
29	MAKE_DATA_BLOCK _ADDRESS Function	不支持	-
30	NAME_RESOLVE Procedure	NAME_RESOLVE Procedure	GaussDB:参数context和 part1_type为INTEGER,参数 object_number为OID; GaussDB不支持NUMBER到 OID的隐式转换。 Oracle:参数context、part1_type和object_number均为NUMBER。

序号	Oracle数据库	GaussDB数据库	差异
31	NAME_TOKENIZE Procedure	NAME_TOKENIZE Procedure	GaussDB:参数nextpos为 INTEGER类型。 Oracle:参数nextpos为
			BINARY_INTEGER类型。
32	OLD_CURRENT_SCH EMA Function	OLD_CURRENT_S CHEMA Function	GaussDB:函数返回类型为 VARCHAR。
			Oracle: 函数返回类型为 VARCHAR2。
33	OLD_CURRENT_USE R Function	OLD_CURRENT_U SER Function	GaussDB: 函数返回类型为 TEXT。
			Oracle: 函数返回类型为 VARCHAR2。
34	PORT_STRING Function	不支持	-
35	SQLID_TO_SQLHAS H Function	不支持	-
36	TABLE_TO_COMMA Procedures	TABLE_TO_COM MA Procedure	GaussDB:参数tab为 VARCHAR2数组。
			Oracle:该存储过程存在2个重载。参数tab可以为两种类型之一:一种为uncl_array,另一种为lname_array。
37	VALIDATE Procedure	不支持	-
38	WAIT_ON_PENDING _DML Function	不支持	-

表 3-102 DBMS_SQL 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	BIND_ARRAY Procedures	SQL_BIND_ARRAY Function	-
2	BIND_VARIABLE Procedures	SQL_BIND_VARIABLEF unction	-
3	BIND_VARIABLE_PKG Procedure	不支持	-
4	CLOSE_CURSOR Procedure	SQL_UNREGISTER_CO NTEXT Function	-

序号	Oracle数据库	GaussDB数据库	差异
5	COLUMN_VALUE Procedure	GET_RESULT Procedure	-
6	COLUMN_VALUE_LON G Procedure	不支持	-
7	DEFINE_ARRAY Procedure	SET_RESULTS_TYPE Procedure	-
8	DEFINE_COLUMN Procedures	SET_RESULT_TYPE Procedure	-
9	DEFINE_COLUMN_CHA R Procedure	不支持	-
10	DEFINE_COLUMN_LON G Procedure	不支持	-
11	DEFINE_COLUMN_RAW Procedure	不支持	-
12	DEFINE_COLUMN_RO WID Procedure	不支持	-
13	DESCRIBE_COLUMNS Procedure	DESCRIBE_COLUMNS Procedure	-
14	DESCRIBE_COLUMNS2 Procedure	不支持	-
15	DESCRIBE_COLUMNS3 Procedure	不支持	-
16	EXECUTE Function	SQL_RUN Function	GaussDB:返回值为常量1。 Oracle:返回值对于INSERT,UPDATE,DELETE语句是影响的行数,对于其他语句则无意义。
17	EXECUTE_AND_FETCH Function	RUN_AND_NEXT Function	-
18	FETCH_ROWS Function	NEXT_ROW Function	-
19	GET_NEXT_RESULT Procedures	不支持	-
20	IS_OPEN Function	IS_ACTIVE Function	-
21	LAST_ERROR_POSITIO N Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
22	LAST_ROW_COUNT Function	LAST_ROW_COUNT Function	-
23	LAST_ROW_ID Function	不支持	-
24	LAST_SQL_FUNCTION_ CODE Function	不支持	-
25	OPEN_CURSOR Functions	REGISTER_CONTEXT Function	-
26	PARSE Procedures	支持,有差异	GaussDB中为 SQL_SET_SQL Function,不支持重 载。
27	RETURN_RESULT Procedures	不支持	-
28	TO_CURSOR_NUMBER Function	不支持	-
29	TO_REFCURSOR Function	不支持	-
30	VARIABLE_VALUE Procedures	GET_VARIABLE_RESUL T Procedures	-
31	VARIABLE_VALUE_PKG Procedure	不支持	-

表 3-103 DBMS_SQL 数据类型兼容性说明

序号	Oracle数据库	GaussDB数据库
1	DBMS_SQL DESC_REC	DBE_SQL.DESC_REC
2	DBMS_SQL DATE_TABLE	DBE_SQL.DATE_TABLE
3	DBMS_SQL NUMBER_TABLE	DBE_SQL.NUMBER_TABLE
4	DBMS_SQL VARCHAR2_TABLE	DBE_SQL.VARCHAR2_TABLE
5	DBMS_SQL BLOB_TABLE	DBE_SQL.BLOB_TABLE

表 3-104 UTL_FILE 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	FCLOSE Procedure	CLOSE Procedure	-

序号	Oracle数据库	GaussDB数据库	差异
2	FCLOSE_ALL Procedure	CLOSE_ALL Procedure	-
3	FCOPY Procedure	COPY Procedure	-
4	FFLUSH Procedure	FLUSH Procedure	-
5	FGETATTR Procedure	GET_ATTR Procedure	-
6	FGETPOS Function	GET_POS Function	-
7	FOPEN Function	FOPEN Function	-
8	FOPEN_NCHAR Function	FOPEN_NCHAR Function	-
9	FREMOVE Procedure	REMOVE Procedure	-
10	FRENAME Procedure	RENAME Procedure	-
11	FSEEK Procedure	SEEK Procedure	-
12	GET_LINE Procedure	READ_LINE Procedure	-
13	GET_LINE_NCHA R Procedure	READ_LINE_NCHAR Procedure	-
14	GET_RAW Procedure	GET_RAW Procedure	-
15	IS_OPEN Function	IS_OPEN Function	-
16	NEW_LINE Procedure	支持,有差异, NEW_LINE Function	GaussDB将接口定义为 Function。
17	PUT Procedure	支持,有差异, WRITE Function	GaussDB将接口定义为 Function。
18	PUT_LINE Procedure	支持,有差异, WRITE_LINE Function	GaussDB将接口定义为 Function。
19	PUT_LINE_NCHA R Procedure	支持,有差异, WRITE_LINE_NCHAR Function	GaussDB将接口定义为 Function。
20	PUT_NCHAR Procedure	支持,有差异, WRITE_NCHAR Function	GaussDB将接口定义为 Function。

序号	Oracle数据库	GaussDB数据库	差异
21	PUTF Procedure	支持,有差异, FORMAT_WRITE Function	GaussDB将接口定义为 Function。
22	PUTF_NCHAR Procedure	支持,有差异, FORMAT_WRITE_NC HAR Function	GaussDB将接口定义为 Function。
23	PUT_RAW Procedure	支持,有差异, PUT_RAW Function	GaussDB将接口定义为 Function。

表 3-105 DBMS_SESSION 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	CLEAR_ALL_CONTEXT Procedure	不支持	-
2	CLEAR_CONTEXT Procedure	CLEAR_CONTEX T Function	-
3	CLEAR_IDENTIFIER Procedure	不支持	-
4	CLOSE_DATABASE_LI NK Procedure	不支持	-
5	CURRENT_IS_ROLE_E NABLED Function	不支持	-
6	FREE_UNUSED_USER_ MEMORY Procedure	不支持	-
7	GET_PACKAGE_MEM ORY_UTILIZATION Procedure	不支持	-
8	IS_ROLE_ENABLED Function	不支持	-
9	IS_SESSION_ALIVE Function	不支持	-
10	LIST_CONTEXT Procedures	不支持	-
11	MODIFY_PACKAGE_ST ATE Procedure	MODIFY_PACKA GE_STATE Procedure	GaussDB: 仅支持入参flags = 1的场景使用。 Oracle: 支持flags=1或flags= 2的场景使用。

序号	Oracle数据库	GaussDB数据库	差异
12	RESET_PACKAGE Procedure	不支持	-
13	SESSION_IS_ROLE_EN ABLED Function	不支持	-
14	SESSION_TRACE_DISA BLE Procedure	不支持	-
15	SESSION_TRACE_ENA BLE Procedure	不支持	-
16	SET_CONTEXT Procedure	SET_CONTEXT Function	GaussDB: 仅包括参数 namespace, attribute和 value, 类型均为text。 Oracle: 包括参数 namespace, attribute, value, username和 client_id, 类型均为 VARCHAR2。
17	SET_EDITION_DEFERR ED Procedure	不支持	-
18	SET_IDENTIFIER Procedure	不支持	-
19	SET_NLS Procedure	不支持	-
20	SET_ROLE Procedure	不支持	-
21	SET_SQL_TRACE Procedure	不支持	-
22	SLEEP Procedure	不支持	-
23	SWITCH_CURRENT_C ONSUMER_GROUP Procedure	不支持	
24	UNIQUE_SESSION_ID Function	不支持	-

表 3-106 UTL_MATCH 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	EDIT_DISTANCE Function	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
2	EDIT_DISTANCE_SI MILARITY Function	EDIT_DISTANCE_SI MILARITY Function	GaussDB:参数str1和str2均为 TEXT类型,函数返回类型为 INTEGER。
			Oracle: 参数s1和s2为 VARCHAR2类型,函数返回类 型为PLS_INTEGER。
3	JARO_WINKLER Function	不支持	-
4	JARO_WINKLER_SI MILARITY Function	不支持	-

表 3-107 DBMS_APPLICATION_INFO 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	READ_CLIENT_INF O Procedure	READ_CLIENT_INF O Function	GaussDB:参数client_info为 TEXT类型。
			Oracle:参数client_info为 VARCHAR2类型。
2	READ_MODULE Procedure	READ_MODULE Procedure	GaussDB:参数 module_name、action_name 为TEXT类型。
			Oracle:参数module_name、 action_name为VARCHAR2类 型。
3	SET_ACTION Procedure	SET_ACTION Procedure	GaussDB:参数action_name 为TEXT类型。
			Oracle:参数action_name为 VARCHAR2类型。
4	SET_CLIENT_INFO Procedure	SET_CLIENT_INFO Function	GaussDB:参数str为TEXT类型,且返回类型为void。
			Oracle:参数client_info为 VARCHAR2类型,无返回值。 二者均为写入客户端信息,最 大输入64字节,超过64字节将 被截断。
5	SET_MODULE Procedure	SET_MODULE Procedure	GaussDB:参数 module_name、action_name 为TEXT类型。
			Oracle:参数module_name、 action_name为VARCHAR2类 型。

序号	Oracle数据库	GaussDB数据库	差异
6	SET_SESSION_LON GOPS Procedure	不支持	-

表 3-108 DBMS_XMLDOM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLDOM.AP PENDCHILD	DBE_XMLDOM.AP PENDCHILD	 GaussDB: DOCUMENT类型节点下APPEND ATTR类型节点会报"operation not support"错误。Oracle: 在此场景下不报错,但实际并没有挂载成功。 GaussDB: ATTR类型节点下APPEND ATTR类型节点会报"operation not support"错误。Oracle: 在此场景下不报错,但实际并没有挂载成功。 GaussDB: 父节点在添加多个ATTR类型子节点时,不允许KEY值相同的子节点下。Oracle: 允许KEY值相同的子节点下。
2	DBMS_XMLDOM.CR EATEELEMENT	DBE_XMLDOM.CR EATEELEMENT	-
3	DBMS_XMLDOM.CR EATETEXTNODE	DBE_XMLDOM.CR EATETEXTNODE	-
4	DBMS_XMLDOM.FRE EDOCUMENT	DBE_XMLDOM.FR EEDOCUMENT	GaussDB:释放时不会立刻释放对象,累积一定数量后释放。document下全部节点失效。 Oracle:立即释放对象。
5	DBMS_XMLDOM.FRE EELEMENT	DBE_XMLDOM.FR EEELEMENT	-
6	DBMS_XMLDOM.FRE ENODE	DBE_XMLDOM.FR EENODE	-

序号	Oracle数据库	GaussDB数据库	差异
7	DBMS_XMLDOM.FRE ENODELIST	DBE_XMLDOM.FR EENODELIST	GaussDB:nodelist会被释 放。
			Oracle:释放nodelist后,在 原始的doc中还能被查询到。
8	DBMS_XMLDOM.GE TATTRIBUTE	DBE_XMLDOM.G ETATTRIBUTE	-
9	DBMS_XMLDOM.GE TATTRIBUTES	DBE_XMLDOM.G ETATTRIBUTES	-
10	DBMS_XMLDOM.GE TCHILDNODES	DBE_XMLDOM.G ETCHILDNODES	GaussDB:对document的 node使用时会包含dtd。
			Oracle:不包含dtd。
11	DBMS_XMLDOM.GE TCHILDRENBYTAGN AME	DBE_XMLDOM.G ETCHILDRENBYTA GNAME	GaussDB: DBE_XMLDOM.GETCHILDRE NBYTAGNAME接口的参数ns 不支持传入参数" * ",如需获 取节点下全部属性,可使用 DBE_XMLDOM.GETCHILDN ODES接口。
			Oracle: 支持传入参数" * "。
12	DBMS_XMLDOM.GE TDOCUMENTELEME NT	DBE.XMLDOM.GE TDOCUMENTELE MENT	-
13	DBMS_XMLDOM.GE TFIRSTCHILD	DBE_XMLDOM.G ETFIRSTCHILD	-
14	DBMS_XMLDOM.GE TLASTCHILD	DBE_XMLDOM.G ETLASTCHILD	-
15	DBMS_XMLDOM.GE TLENGTH	DBE_XMLDOM.G ETLENGTH	-
16	DBMS_XMLDOM.GE TLOCALNAME	DBE_XMLDOM.G ETLOCALNAME	-
17	DBMS_XMLDOM.GE TNAMEDITEM	DBE_XMLDOM.G ETNAMEDITEM	-
18	DBMS_XMLDOM.GE TNEXTSIBLING	DBE_XMLDOM.G ETNEXTSIBLING	-
19	DBMS_XMLDOM.GE TNODENAME	DBE_XMLDOM.G ETNODENAME	-
20	DBMS_XMLDOM.GE TNODETYPE	DBE_XMLDOM.G ETNODETYPE	-
21	DBMS_XMLDOM.GE TTAGNAME	DBE_XMLDOM.G ETTAGNAME	-

序号	Oracle数据库	GaussDB数据库	差异
22	DBMS_XMLDOM.IM PORTNODE	DBE_XMLDOM.IM PORTNODE	-
23	DBMS_XMLDOM.ISN ULL	DBE_XMLDOM.IS NULL	GaussDB: 入参为 DOMNODELIST类型时,若对 象在哈希表中不存在会发生报 错。 Oracle: 不会报错。
24	DBMS_XMLDOM.ITE M	DBE_XMLDOM.IT EM	-
25	DBMS_XMLDOM.MA KENODE	DBE_XMLDOM.M AKENODE	GaussDB:该函数不支持直接 作为函数返回值返回。 Oracle:支持直接作为函数返 回值返回。

序号	Oracle数据库	GaussDB数据库	差异
26	DBMS_XMLDOM.NE WDOMDOCUMENT	DBE_XMLDOM.N EWDOMDOCUM ENT	 GaussDB入参大小需限制 在2GB以内。 Oracle:与CLOB类型大小 一致。
			 GaussDB目前暂不支持外部 部DTD解析。 Oracle: 支持解析外部 DTD。
			● GaussDB newdomdocument创建的 doc,默认UTF-8字符集。 Oracle:根据服务端字符 集生成。
			● GaussDB从同一个xmltype 实例中解析出的每一个doc 都是独立的,对doc的修改 也不会影响到xmltype。Oracle:从同一个xmltype 实例中解析出的每一个doc 不独立,有关联关系。
			● GaussDB version字段只支 持1.0,1.0-1.9解析警告但 正常执行,1.9以上报错。 Oracle: 不报错。
			● GaussDB与Oracle数据库DTD校验差异:!ATTLIST to type (CHECK check Check "Check" "不属于括号中枚举值,而Oracle数据库不报错。 ENTITY baidu"www.baidu.com" &Baidu&writer将报错,因区分字母大小写,Baidu无法与baidu对应。Oracle:不报错。
			● GaussDB 与Oracle数据库 命名空间校验差异:解析 未声明的命名空间标签正 常执行。 Oracle:报错。
27	DBMS_XMLDOM.SET ATTRIBUTE	DBE_XMLDOM.SE TATTRIBUTE	GaussDB:属性key不支持为null或空字符串。 Oracle:属性key允许为null或空字符串。

序号	Oracle数据库	GaussDB数据库	差异
28	DBMS_XMLDOM.SET CHARSET	DBE_XMLDOM.SE TCHARSET	GaussDB目前支持的字符集有: UTF-8、UTF-16、UCS-4、UCS-2、ISO-8859-1、ISO-8859-2、ISO-8859-3、ISO-8859-4、ISO-8859-5、ISO-8859-6、ISO-8859-7、ISO-8859-8、ISO-8859-9、ISO-2022-JP、Shift_JIS、EUC-JP、ASCII。输入其他字符集会报错或者可能导致输出乱码。
29	DBMS_XMLDOM.SET DOCTYPE	DBE_XMLDOM.SE TDOCTYPE	GaussDB name、sysid、 pubid的总长度限制在32500 个字节以内。 Oracle:限制在32767字节 内。
30	DBMS_XMLDOM.WR ITETOBUFFER	DBE_XMLDOM.W RITETOBUFFER	 GaussDB writetobuffer输出buffer限制在1GB以内。Oracle:限制在32767字节内。 GaussDB输出doc将包含XML声明version和encoding。Oracle:用户不主动指定将不包含。 GaussDB入参为domnode类型时,如果节点是doc转换的,输出节点将包含XML声明version和encoding。Oracle:用户不主动指定将不包含。 GaussDB默认以UTF-8字符集输出xml。Oracle:根据数据库字符集生成。

序号	Oracle数据库	GaussDB数据库	差异
31	DBMS_XMLDOM.WR ITETOCLOB	DBE_XMLDOM.W RITETOCLOB	 GaussDB writetoclob大小 支持1GB以内。 Oracle:按CLOB大小支 持。
			 GaussDB输出doc将包含 XML声明version和 encoding。 Oracle: 用户不主动指定 将不包含。
			● GaussDB入参为domnode 类型时,如果节点是doc转 换的,输出节点将包含 XML声明version和 encoding。 Oracle: 用户不主动指定 将不包含。
			● GaussDB 默认以UTF-8字 符集输出xml。 Oracle:根据数据库字符 集生成。

序号	Oracle数据库	GaussDB数据库	差异
32	DBMS_XMLDOM.WR ITETOFILE	DBE_XMLDOM.W RITETOFILE	● GaussDB document入 参,filename长度限制在 255个字节以内,charset 请参考 dbe_xmldom.setcharset接 口。 Oracle: filename长度限 制受操作系统影响,大于 255个字节。
			 GaussDB domnode入参, filename长度限制在255个 字节以内,charset请参考 dbe_xmldom.setcharset接 口。 Oracle: filename长度限 制受操作系统影响,大于 255个字节。
			● GaussDB该函数会添加缩 进等内容,将输出格式 化。输出doc将包含XML声 明version和encoding。入 参为domnode类型时,如 果节点是doc转换的,输出 节点将包含XML声明 version和encoding。 Oracle: 用户不主动指定 将不包含。
			● GaussDB传入 newdomdocument()无参 创建的doc,在不指定 charset时不会报错,默认 UTF-8字符集。 Oracle:会进行报错。
			● GaussDB filename需要在 pg_directory中创建的路径 下,filename中的\会被转 换成/,只允许存在一个/。 文件名格式应为 pg_directory_name/ file_name。 Oracle:按用户输入不进 行转义。
33	DBMS_XMLDOM.GE TNODEVALUE	DBE_XMLDOM.G ETNODEVALUE	-
34	DBMS_XMLDOM.GE TPARENTNODE	DBE_XMLDOM.G ETPARENTNODE	-

序号	Oracle数据库	GaussDB数据库	差异
35	DBMS_XMLDOM.HA SCHILDNODES	DBE_XMLDOM.H ASCHILDNODES	-
36	DBMS_XMLDOM.MA KEELEMENT	DBE_XMLDOM.M AKEELEMENT	-
37	DBMS_XMLDOM.SET NODEVALUE	DBE_XMLDOM.SE TNODEVALUE	 GaussDB nodeValue入参,可以输入空字符串和NULL值,但不会对节点值进行修改。Oracle:空字符串和NULL会将节点值修改为空字符串。 GaussDB nodeValue入参,暂不支持转义字符'&',如字符串中包含该转义字符,会清空节点值。Oracle:支持转义字符。
38	DBMS_XMLDOM.GE TELEMENTSBYTAGN AME	DBE_XMLDOM.G ETELEMENTSBYT AGNAME	-

表 3-109 DBMS_XMLPARSER 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLPARSER. FREEPARSER	DBE_XMLPARSER .FREEPARSER	-
2	DBMS_XMLPARSER. GETDOCUMENT	DBE_XMLPARSER .GETDOCUMENT	-
3	DBMS_XMLPARSER. GETVALIDATIONM ODE	DBE_XMLPARSER .GETVALIDATION MODE	-
4	DBMS_XMLPARSER. NEWPARSER	支持,有差异, DBE_XMLPARSER .NEWPARSER	GaussDB中parser对象的数量上 限为16777215,Oracle数据库 中约为1亿。

序号 Oracle数据库 GaussDB数据库	差异
5 DBMS_XMLPARSER. 支持,有差异,DBE_XMLPARSER .PARSEBUFFER	2. 与Oracle数据库解析字段差异:字符串encoding只支持UTF-8;version字段只支持1.0, 1.0-1.9解析警告但正常执行, 1.9以上报错。 2. 与Oracle数据库命名空间校验差异:解析未声明的命名空间校验差异:解析未声明的命名空间校验差异:解析差异: '"会被解析转译为字符'",而Oracle数据库中预定义实体统一都没有转译为字符。 4. 与Oracle数据库DTD校验差异: • !ATTLIST to type (CHECK check Check) "Ch"将报错,因默认值"Ch"不属于括号中枚举值,而Oracle数据库不报错。 • ENTITY baidu"www.baidu.com"

序号	Oracle数据库	GaussDB数据库	差异
6	DBMS_XMLPARSER. PARSECLOB	支持,有差异, DBE_XMLPARSER	1. PARSECLOB不支持解析大于 等于2GB的clob。
		.PARSECLOB	2. 与Oracle数据库解析字段差异:
			字符串encoding只支持UTF-8; version字段只支持1.0,1.0-1.9 解析警告但正常执行,1.9以上 报错。
			3. 与Oracle数据库命名空间校验 差异:解析未声明的命名空间标 签正常执行,而Oracle数据库会 报错。
			4. 与Oracle数据库xml预定义实体解析差异:'"会被解析转译为字符'",而Oracle数据库预定义实体统一都没有转译为字符。
			5. 与Oracle数据库DTD校验差 异:
			 !ATTLIST to type (CHECK check Check) "Ch"将报错,因默认值"Ch"不属于括号中枚举值,而Oracle数据库不报错。
			 <!--ENTITY baidu "www.baidu.com"--> &Baidu&writer将报错,因 区分字母大小写,Baidu无法 与baidu对应,而Oracle数据 库不报错。
7	DBMS_XMLPARSER. SETVALIDATIONM ODE	DBE_XMLPARSER .SETVALIDATION MODE	-

表 3-110 DBMS_ILM 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_ILM.ADD_TO_ ILM	不支持	-
2	DBMS_ILM.ARCHIVE STATENAME	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
3	DBMS_ILM.EXECUTE _ILM	支持,有差异, DBE_ILM.EXECU TE_ILM	● GaussDB数据库的入参 schema在Oracle数据库中对 应为owner。
			GaussDB数据库不支持指定 ilm_scope(一次指定多个对象)的操作。
4	DBMS_ILM.EXECUTE _ILM_TASK	不支持	-
5	DBMS_ILM.PREVIEW _ILM	不支持	-
6	DBMS_ILM.REMOVE _FROM_ILM	不支持	-
7	DBMS_ILM.STOP_IL M	DBE_ILM.STOP_ ILM	-

表 3-111 DBMS_ILM_ADMIN 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_ILM_ADMIN. CLEAR_HEAT_MAP_ ALL	不支持	-
2	DBMS_ILM_ADMIN. CLEAR_HEAT_MAP_ TABLE	不支持	-
3	DBMS_ILM_ADMIN. CUSTOMIZE_ILM	支持,有差异, DBE_ILM_ADMI N.CUSTOMIZE_I LM	入参parameter取值对应的特性 参数存在差异。GaussDB数据库 param取值支持1、2、7、11、 12、13、14和15。GaussDB数 据库param取值为14时,对应的 特性参数为 WIND_DURATION,用于控制 自动调度中执行窗口的持续时 长,而ORACLE数据库对应的特性参数则为 AUTO_OPTIMIZE_INACTIVITY_ THRESHOLD,其表示ado的不 活动时间长度。
4	DBMS_ILM_ADMIN. DISABLE_ILM	DBE_ILM_ADMI N.DISABLE_ILM	-
5	DBMS_ILM_ADMIN. ENABLE_AUTO_OPT IMIZE	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
6	DBMS_ILM_ADMIN. ENABLE_ILM	DBE_ILM_ADMI N.ENABLE_ILM	-
7	DBMS_ILM_ADMIN. IGNORE_AUTO_OPT IMIZE_ CRITERIA	不支持	-
8	DBMS_ILM_ADMIN. SET_HEAT_MAP_ALL	不支持	-
9	DBMS_ILM_ADMIN. SET_HEAT_MAP_STA RT	不支持	-
10	DBMS_ILM_ADMIN. SET_HEAT_MAP_TA BLE	不支持	-

表 3-112 DBMS_COMPRESSION 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_COMPRESSI ON.GET_COMPRES SION_RATIO	支持,有差异, DBE_COMPRESSI ON.GET_COMPR ESSION_RATIO	 GaussDB不支持LOBs的压缩率获取。 对于单个对象的压缩率获取: GaussDB入参comptype取值仅支持1(未压缩)和2(高级压缩),Oracle还支持1024、2048等取值。 GaussDB入参objtype取值仅支持1(表对象),而Oracle还支持2(索引对象)。 Oracle数据库使用subset_numrows参数直接来决定采样的行数(即为参数的取值),而GaussDB则使用sample_ratio(采样率)来间接确定采样的行数。

序号	Oracle数据库	GaussDB数据库	差异
2	DBMS_COMPRESSI ON.GET_COMPRES SION_TYPE	支持,有差异, DBE_COMPRESSI ON.GET_COMPR ESSION_TYPE	 Oracle使用rowid来指定待获取压缩类型的行,而GaussDB则是使用行的ctid来指定。 返回值为comptype,其取值差异同GET_COMPRESSION_RATIO。

表 3-113 DBMS_HEAT_MAP 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_HEAT_MAP.B LOCK_HEAT_MAP	不支持	-
2	DBMS_HEAT_MAP.E XTENT_HEAT_MAP	不支持	-
3	DBMS_HEAT_MAP.O BJECT_HEAT_MAP	不支持	-
4	DBMS_HEAT_MAP.S EGMENT_HEAT_MA P	不支持	-
5	DBMS_HEAT_MAP.T ABLESPACE_HEAT_ MAP	不支持	-
6	不支持	DBE_HEAT_MAP .ROW_HEAT_M AP	详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推 荐)> DBE_HEAT_MAP"章 节。

表 3-114 DBMS_DESCRIBE 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_DESCRIBE. DESCRIBE_PROCE DURE	支持,有差异, DBE_DESCRIBE.DE SCRIBE_PROCEDU RE	 datatype参数与O存在差 异,GaussDB返回数据类型 的oid,O数据库返回O数据 库内部的数据类型的编号。
			 datalength、dataprecision 和scale因GaussDB创建存储 过程或函数时无法保留类型 的约束(如number(7,2)、 varchar2(20)等),该三个 参数置0处理;Oracle可使 用%type方法获得带约束的 数据类型。
			 具体的行为差异详见《开发 指南》> "存储过程 > 高级 包 > 二次封装接口(推荐) > DBE_DESCRIBE"章节。

表 3-115 DBMS_STATS 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_STATS.ALTE R_STATS_HISTORY _RETENTION	不支持	-
2	DBMS_STATS.CAN CEL_ADVISOR_TAS K	不支持	-
3	DBMS_STATS.CON FIGURE_ADVISOR _FILTER	不支持	-
4	DBMS_STATS.CON FIGURE_ADVISOR _OBJ_FILTER	不支持	-
5	DBMS_STATS.CON FIGURE_ADVISOR _OPR_FILTER	不支持	-
6	DBMS_STATS.CON FIGURE_ADVISOR _RULE_FILTER	不支持	-
7	DBMS_STATS.CRE ATE_ADVISOR_TA SK	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
8	DBMS_STATS.CON VERT_RAW_VALU E	不支持	-
9	DBMS_STATS.CON VERT_RAW_VALU E_NVARCHAR	不支持	-
10	DBMS_STATS.CON VERT_RAW_VALU E_ROWID	不支持	-
11	DBMS_STATS.COP Y_TABLE_STATS	不支持	-
12	DBMS_STATS.CRE ATE_EXTENDED_S TATS	不支持	-
13	DBMS_STATS.CRE ATE_STAT_TABLE	DBE_STATS.CREAT E_STAT_TABLE	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
14	DBMS_STATS.DEL ETE_COLUMN_ST ATS	DBE_STATS.DELET E_COLUMN_STAT S	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 GaussDB中,使用该接口可以设置表达式统计信息,但tabname应传该表达式对应的索引名。
15	DBMS_STATS.DEL ETE_DATABASE_P REFS	不支持	-
16	DBMS_STATS.DEL ETE_DATABASE_S TATS	不支持	-
17	DEDBMS_STATS.D ELETE_DICTIONAR Y_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
18	DBMS_STATS.DEL ETE_FIXED_OBJEC TS_STATS	不支持	-
19	DBMS_STATS.DEL ETE_INDEX_STATS	DBE_STATS.DELET E_INDEX_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
20	DBMS_STATS.DEL ETE_PENDING_ST ATS	不支持	-
21	DBMS_STATS.DEL ETE_PROCESSING _RATE	不支持	-
22	DBMS_STATS.DEL ETE_SCHEMA_PRE FS	不支持	-
23	DBMS_STATS.DEL ETE_SCHEMA_STA TS	DBE_STATS.DELET E_SCHEMA_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
24	DBMS_STATS.DEL ETE_SYSTEM_STAT S	不支持	-
25	DBMS_STATS.DEL ETE_TABLE_PREFS	不支持	-
26	DBMS_STATS.DEL ETE_TABLE_STATS	DBE_STATS.DELET E_TABLE_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
27	DBMS_STATS.DIFF _TABLE_STATS_IN_ HISTORY	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
28	DBMS_STATS.DIFF _TABLE_STATS_IN_ PENDING	不支持	-
29	DBMS_STATS.DIFF _TABLE_STATS_IN_ STATTAB	不支持	-
30	DBMS_STATS.DRO P_ADVISOR_TASK	不支持	-
31	DBMS_STATS.DRO P_EXTENDED_STA TS	不支持	-
32	DBMS_STATS.DRO P_STAT_TABLE	DBE_STATS.DROP _STAT_TABLE	-
33	DBMS_STATS.EXE CUTE_ADVISOR_T ASK	不支持	-
34	DBMS_STATS.EXP ORT_COLUMN_ST ATS	DBE_STATS.EXPO RT_COLUMN_STA TS	GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 导出的列级统计信息与pg_statistic表保持一致,多列和pg_statistic_ext表保持一致。 支持导出索引表达式统计信息。要求tabname传的是索引名称,colname传的是索引表达式名称。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
35	DBMS_STATS.EXP ORT_DATABASE_P REFS	不支持	-
36	DBMS_STATS.EXP ORT_DATABASE_S TATS	不支持	-
37	DBMS_STATS.EXP ORT_DICTIONARY _STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
38	DBMS_STATS.EXP ORT_FIXED_OBJEC TS_STATS	不支持	-
39	DBMS_STATS.EXP ORT_INDEX_STAT S	DBE_STATS.EXPO RT_INDEX_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 stattab表中,导出的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples、relpages、relallvisible。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
40	DBMS_STATS.EXP ORT_PENDING_ST ATS	不支持	-
41	DBMS_STATS.EXP ORT_SCHEMA_PR EFS	不支持	-
42	DBMS_STATS.EXP ORT_SCHEMA_ST ATS	DBE_STATS.EXPO RT_SCHEMA_STAT S	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 stattab表中,导出的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。导出表相关列级统计信息与pg_statistic表和pg_statistic表和pg_statistic_ext表保持一致。 权限:需要具有stattab表的siud权限。

序号	Oracle数据库	GaussDB数据库	差异
43	DBMS_STATS.EXP ORT_SYSTEM_STA TS	不支持	-
44	DBMS_STATS.EXP ORT_TABLE_PREFS	不支持	-
45	DBMS_STATS.EXP ORT_TABLE_STATS	DBE_STATS.EXPO RT_TABLE_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 stattab表中,导出的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。级联导出的列级统计信息与pg_statistic表和pg_statistic表和pg_statistic是xt表保持一致。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
46	DBMS_STATS.FLU SH_DATABASE_M ONITORING_INFO	不支持	-
47	DBMS_STATS.GAT HER_DATABASE_S TATS	不支持	-
48	DBMS_STATS.GAT HER_DICTIONARY _STATS	不支持	-
49	DBMS_STATS.GAT HER_FIXED_OBJEC TS_STATS	不支持	-
50	DBMS_STATS.GAT HER_INDEX_STAT S	不支持	-
51	DBMS_STATS.GAT HER_PROCESSING _RATE	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
52	DBMS_STATS.GAT HER_SCHEMA_ST ATS	不支持	-
53	DBMS_STATS.GAT HER_SYSTEM_STA TS	不支持	-
54	DBMS_STATS.GAT HER_TABLE_STATS	不支持	-
55	DBMS_STATS.GEN ERATE_STATS	不支持	-
56	DBMS_STATS.GET _ADVISOR_OPR_FI LTER	不支持	-
57	DBMS_STATS.GET _ADVISOR_RECS	不支持	-
58	DBMS_STATS.GET _COLUMN_STATS	不支持	-
59	DBMS_STATS.GET _INDEX_STATS	不支持	-
60	DBMS_STATS.GET _PARAM	不支持	-
61	DBMS_STATS.GET _PREFS	不支持	-
62	DBMS_STATS.GET _STATS_HISTORY_ AVAILABILITY	DBE_STATS.GET_S TATS_HISTORY_A VAILABILITY	GaussDB查询到的是全库存在的 最早历史统计信息的收集时间。
63	DBMS_STATS.GET _STATS_HISTORY_ RETENTION	DBE_STATS.GET_S TATS_HISTORY_RE TENTION	-
64	DBMS_STATS.GET _SYSTEM_STATS	不支持	-
65	DBMS_STATS.GET _TABLE_STATS	不支持	-
66	DBMS_STATS.IMP LEMENT_ADVISO R_TASK	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
67	DBMS_STATS.IMP ORT_COLUMN_ST	DBE_STATS.IMPO RT_COLUMN_STA	● GaussDB中ownname应传 schema名。
	ATS	TS	● GaussDB仅支持部分入参功 能,详见《开发指南》中 "存储过程 > 高级包 > 二次 封装接口(推荐) > DBE_STATS"章节。
			 导出单列col导出的统计信息 与pg_statistic表保持一致。 多列ext-col导出的统计信息 与pg_statistic_ext表保持一 致。
			支持导入索引表达式统计信息。要求tabname传的是索引名称,colname传的是索引表达式名称。
			● 权限:需要具有查询表的 analyze权限以及stattab表的 siud权限。
68	DBMS_STATS.IMP ORT_DATABASE_P REFS	不支持	-
69	DBMS_STATS.IMP ORT_DATABASE_S TATS	不支持	-
70	DBMS_STATS.IMP ORT_DICTIONARY _STATS	不支持	-
71	DBMS_STATS.IMP ORT_FIXED_OBJEC TS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
72	DBMS_STATS.IMP ORT_INDEX_STAT S	DBE_STATS.IMPO RT_INDEX_STATS	● GaussDB中ownname应传 schema名。
	3		● GaussDB仅支持部分入参功 能,详见《开发指南》中 "存储过程 > 高级包 > 二次 封装接口(推荐) > DBE_STATS"章节。
			 stattab表中,导入的表、分 区级统计信息为numrows、 numblocks、relallvisible, 分别对应系统表pg_class、 pg_partition的reltuples, relpages, relallvisible。
			● 权限:需要具有查询表的 analyze权限以及stattab表的 siud权限。
73	DBMS_STATS.IMP ORT_SCHEMA_PR EFS	不支持	-
74		A_ST RT_SCHEMA_STAT	• GaussDB中ownname应传 schema名。
			● GaussDB仅支持部分入参功 能,详见《开发指南》中 "存储过程 > 高级包 > 二次 封装接口(推荐) > DBE_STATS"章节。
			 stattab表中,导入的表、分 区级统计信息为numrows、 numblocks、relallvisible, 分别对应系统表pg_class、 pg_partition的reltuples, relpages,relallvisible。导 入表相关列级统计信息与 pg_statistic表和 pg_statistic表和 pg_statistic_ext表保持一 致。
			● 权限:需要具有stattab表的 siud权限。
75	DBMS_STATS.IMP ORT_SYSTEM_STA TS	不支持	-
76	DBMS_STATS.IMP ORT_TABLE_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
77	DBMS_STATS.IMP ORT_TABLE_STATS	DBE_STATS.IMPO RT_TABLE_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 stattab表中,导入的表、分区级统计信息为numrows、numblocks、relallvisible,分别对应系统表pg_class、pg_partition的reltuples,relpages,relallvisible。级联导入的列级统计信息与pg_statistic表和pg_statistic表和pg_statistic_ext表保持一致。 权限:需要具有查询表的analyze权限以及stattab表的siud权限。
78	DBMS_STATS.INTE RRUPT_ADVISOR_ TASK	不支持	-
79	DBMS_STATS.LOC K_PARTITION_STA TS	DBE_STATS.LOCK_ PARTITION_STATS	GaussDB中ownname应传 schema名。
80	DBMS_STATS.LOC K_SCHEMA_STATS	DBE_STATS.LOCK_ SCHEMA_STATS	GaussDB中ownname应传 schema名。
81	DBMS_STATS.LOC K_TABLE_STATS	DBE_STATS.LOCK_ TABLE_STATS	GaussDB中ownname应传 schema名。
82	DBMS_STATS.MER GE_COL_USAGE	不支持	-
83	DBMS_STATS.PRE PARE_COLUMN_V ALUES	不支持	-
84	DBMS_STATS.PRE PARE_COLUMN_V ALUES_ROWID	不支持	-
85	DBMS_STATS.PUB LISH_PENDING_ST ATS	不支持	-
86	DBMS_STATS.PUR GE_STATS	DBE_STATS.PURG E_STATS	-

序号	Oracle数据库	GaussDB数据库	差异
87	DBMS_STATS.REM AP_STAT_TABLE	不支持	-
88	DBMS_STATS.REP ORT_ADVISOR_TA SK	不支持	-
89	DBMS_STATS.REP ORT_COL_USAGE	不支持	-
90	DBMS_STATS.REP ORT_GATHER_AU TO_STATS	不支持	
91	DBMS_STATS.REP ORT_GATHER_DA TABASE_STATS	不支持	-
92	DBMS_STATS.REP ORT_GATHER_DIC TIONARY_STATS	不支持	-
93	DBMS_STATS.REP ORT_GATHER_FIX ED_OBJ_STATS	不支持	-
94	DBMS_STATS.REP ORT_GATHER_SC HEMA_STATS	不支持	-
95	DBMS_STATS.REP ORT_STATS_OPER ATIONS	不支持	-
96	DBMS_STATS.RESE T_ADVISOR_TASK	不支持	-
97	DBMS_STATS.RESE T_COL_USAGE	不支持	-
98	DBMS_STATS.RESE T_GLOBAL_PREF_ DEFAULTS	不支持	-
99	DBMS_STATS.RESE T_PARAM_DEFAUL TS	不支持	-
100	DBMS_STATS.RES TORE_DICTIONAR Y_STATS	不支持	-
101	DBMS_STATS.RES TORE_FIXED_OBJE CTS_STATS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
102	DBMS_STATS.RES TORE_SCHEMA_S TATS	DBE_STATS.RESTO RE_SCHEMA_STAT S	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
103	DBMS_STATS.RES TORE_SYSTEM_ST ATS	不支持	-
104	DBMS_STATS.RES TORE_TABLE_STAT S	DBE_STATS.RESTO RE_TABLE_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
105	DBMS_STATS.RES UME_ADVISOR_TA SK	不支持	-
106	DBMS_STATS.SCRI PT_ADVISOR_TAS K	不支持	-
107	DBMS_STATS.SEE D_COL_USAGE	不支持	-
108	DBMS_STATS.SET_ ADVISOR_TASK_P ARAMETER	不支持	-
109	DBMS_STATS.SET_ COLUMN_STATS	DBE_STATS.SET_C OLUMN_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。
110	DBMS_STATS.SET_ DATABASE_PREFS	不支持	-
111	DBMS_STATS.SET_ GLOBAL_PREFS	不支持	-

序号	Oracle数据库	GaussDB数据库	差异
112	DBMS_STATS.SET_ INDEX_STATS	DBE_STATS.SET_I NDEX_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 GaussDB中新增了relallvisible入参。
113	DBMS_STATS.SET_ PARAM	不支持	-
114	DBMS_STATS.SET_ PROCESSING_RAT E	不支持	-
115	DBMS_STATS.SET_ SCHEMA_PREFS	不支持	-
116	DBMS_STATS.SET_ SYSTEM_STATS	不支持	-
117	DBMS_STATS.SET_ TABLE_PREFS	不支持	-
118	DBMS_STATS.SET_ TABLE_STATS	DBE_STATS.SET_T ABLE_STATS	 GaussDB中ownname应传schema名。 GaussDB仅支持部分入参功能,详见《开发指南》中"存储过程 > 高级包 > 二次封装接口(推荐) > DBE_STATS"章节。 GaussDB中新增了relallvisible入参。
119	DBMS_STATS.SHO W_EXTENDED_ST ATS_NAME	不支持	-
120	DBMS_STATS.TRA NSFER_STATS	不支持	-
121	DBMS_STATS.UNL OCK_PARTITION_S TATS	DBE_STATS.UNLO CK_PARTITION_ST ATS	GaussDB中ownname应传 schema名。
122	DBMS_STATS.UNL OCK_SCHEMA_ST ATS	DBE_STATS.UNLO CK_SCHEMA_STAT S	GaussDB中ownname应传 schema名。

序号	Oracle数据库	GaussDB数据库	差异
123	DBMS_STATS.UNL OCK_TABLE_STAT S	DBE_STATS.UNLO CK_TABLE_STATS	GaussDB中ownname应传 schema名。
124	DBMS_STATS.UPG RADE_STAT_TABL E	不支持	-

表 3-116 DBMS_XMLGEN 兼容性说明

序号	Oracle数据库	GaussDB数据库	差异
1	DBMS_XMLGEN.C ONVERT	DBE_XMLGEN.CO NVERT	-
2	DBMS_XMLGEN.N EWCONTEXT	DBE_XMLGEN.NE WCONTEXT	-
3	DBMS_XMLGEN.N EWCONTEXTFRO MHIERARCHY	DBE_XMLGEN.NE WCONTEXTFROM HIERARCHY	 GaussDB生成的递归XML最大深度不能超过5000万层。 Oracle的newcontextfromhierarchy方法对于connect by语句生成的xml是带xml头的,但是对于直接构造的数据不带xml头,GaussDB均带xml头。
4	DBMS_XMLGEN.S ETCONVERTSPECI ALCHARS	DBE_XMLGEN.SET CONVERTSPECIAL CHARS	-
5	DBMS_XMLGEN.S ETNULLHANDLIN G	DBE_XMLGEN.SET NULLHANDLING	-
6	DBMS_XMLGEN.S ETROWSETTAG	DBE_XMLGEN.SET ROWSETTAG	-
7	DBMS_XMLGEN.S ETROWTAG	DBE_XMLGEN.SET ROWTAG	-
8	DBMS_XMLGEN.U SENULLATTRIBUT EINDICATOR	DBE_XMLGEN.USE NULLATTRIBUTEI NDICATOR	-
9	DBMS_XMLGEN.U SEITEMTAGSFORC OLL	DBE_XMLGEN.USE ITEMTAGSFORCO LL	-

序号	Oracle数据库	GaussDB数据库	差异
10	DBMS_XMLGEN.G ETNUMROWSPRO CESSED	DBE_XMLGEN.GET NUMROWSPROC ESSED	-
11	DBMS_XMLGEN.S ETMAXROWS	DBE_XMLGEN.SET MAXROWS	-
12	DBMS_XMLGEN.S ETSKIPROWS	DBE_XMLGEN.SET SKIPROWS	-
13	DBMS_XMLGEN.R ESTARTQUERY	DBE_XMLGEN.RES TARTQUERY	● GaussDB:调用 RESTARTQUERY方法后对更 新的数据不可见。
			● Oracle:调用 RESTARTQUERY方法后对更 新的数据可见。
14	DBMS_XMLGEN.G ETXMLTYPE	DBE_XMLGEN.GET XMLTYPE	-
15	DBMS_XMLGEN.G ETXML	DBE_XMLGEN.GET XML	-
16	DBMS_XMLGEN.C LOSECONTEXT	DBE_XMLGEN.CL OSECONTEXT	-

4 集中式版 MySQL 兼容性说明

4.1 概述

本手册为GaussDB数据库(MySQL兼容性B模式和MySQL兼容性M-Compatibility模式)与MySQL 5.7数据库信息对比。其中,**MySQL兼容性B模式**为B模式下的MySQL兼容性说明;**MySQL兼容性M-Compatibility模式**为M-Compatibility模式下的MySQL兼容性说明。M-Compatibility模式在语法、数据类型、元数据、协议等功能上对MySQL数据库有更好的兼容度,推荐使用。B模式由于架构限制无法很好的与MySQL兼容,后续不再演进,不推荐使用。

4.2 MySQL 兼容性 B 模式

4.2.1 MySQL 兼容性 B 模式概述

本章节主要介绍GaussDB数据库的MySQL兼容性B模式(即sql_compatibility='B'、且设置参数b_format_version='5.7'、b_format_dev_version='s1'时)与MySQL 5.7数据库的兼容性对比信息。仅介绍503.0.0版本后新增的兼容性特性,特性的相关规格和约束建议在开发指南中查看。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

由于GaussDB数据库与MySQL数据库底层框架实现存在差异,GaussDB数据库与MySQL数据库仍存在部分差异。

4.2.2 数据类型

GaussDB的数据类型的大部分功能场景与MySQL一致,但存在部分差异。

除特别说明,部分数据类型精度、标度、位数大小等不支持用浮点型数值定义, 建议使用合法的整型数值定义。

4.2.2.1 数值数据类型

表 4-1 整数类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BOOL	不完全兼容	MySQL: BOOL/BOOLEAN类型实际映射为TINYINT类型。
2	BOOLEAN	不完全兼容	GaussDB: 支持BOOL,其中:
			"真"值的有效文本值是: TRUE、't'、 'true'、'y'、'yes'、'1'、'TRUE'、true、 'on'以及所有非0数值。
			● "假"值的有效文本值是: FALSE、'f'、 'false'、'n'、'no'、'0'、0、'FALSE'、 false、'off'。
			使用TRUE和FALSE是比较规范的用法(也是 SQL兼容的用法)。
3	TINYINT[(M)] [UNSIGNED]	支持	详细请参见说明。
4	SMALLINT[(M)] [UNSIGNED]	支持	详细请参见说明。
5	MEDIUMIN	支持	MySQL存储MEDIUMINT数据需要3字节。
	T[(M)] [UNSIGNED		● 带符号的范围是-8,388,608 ~ +8,388,607。
]		● 无符号的范围是0~+16,777,215。
			GaussDB映射为INT类型,存储需要4字节。
			● 带符号的范围是-2,147,483,648 ~ +2,147,483,647。
			● 无符号的范围是0~+4,294,967,295。
			其他差异请参见说明。
6	INT[(M)] [UNSIGNED]	支持	详细请参见说明。
7	INTEGER[(M)] [UNSIGNED]	支持	详细请参见说明。

序号	MySQL数据 库	GaussDB数 据库	差异
8	BIGINT[(M)] [UNSIGNED]	支持	详细请参见说明。

□ 说明

• 输入格式:

MySQL

对于类似"asbd"、"12dd"、"12 12"等字符场景的输入,会采取截断或返回0值 并上报WARNING处理,在严格模式插表时失败。

GaussDB

- 整数类型(TINYINT、SMALLINT、MEDIUMINT、INT、INTEGER、BIGINT)的 输入,当非法字符串部分被截断时,如"12@3",会直接截断并无提示信息,插 表成功。
- 当整数类型全部被截断(如"@123")或字符串为空时,返回0, 且插表成功。

操作符:

● +、-、*操作符

GaussDB: INT/INTEGER/SMALLINT/BIGINT在进行运算时,返回值为类型本身,不会向上提升类型,当返回值超范围时报错。

MySQL: 支持提升类型到BIGINT后计算。

● |、&、^、~运算符

GaussDB: 在类型所占用BIT位中计算; GaussDB中^表示指数运算,如需使用异或运算符,使用#替换。

MySQL: 提升类型计算。

• 负数显式类型转换:

GaussDB: 宽松模式结果为0,严格模式报错。

MySQL: 依据其对应的二进制将最高位替换成数值位计算结果,例如(-1)::uint4 = 4294967295。

• 其他差异:

INT[(M)]精度,MySQL控制格式化输出,GaussDB仅语法支持,不支持功能。

- 聚集函数:
 - variance: GaussDB表示样本方差, MySQL表示总体方差。
 - stddev: GaussDB表示样本标准差,MySQL表示总体标准差。

● 显示宽度:

- 在为整型数字列指明宽度信息时,如果不同时指定ZEROFILL,则宽度信息在表结构描述中不显示。
- INSERT语句插入字符类型字段,GaussDB统一补齐0后插入。
- JOIN USING语句,涉及类型推导,MySQL默认第一张表列,GaussDB若结果为有符号 类型则宽度信息失效,否则为第一张表字段宽度。
- greatest/least、ifnull/if、case when / decode,MySQL不补齐0,GaussDB在类型及 宽度信息一致时补齐0。
- 作为函数/存储过程出入参、返回值时、MySQL支持功能、GaussDB语法不报错功能不 支持。

表 4-2 任意精度类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DECIMAL[(M[,D])]	支持	• 操作符: GaussDB中 "^" 表示指数运算,如需使用异或运算符,使用 "#" 替换;
2	NUMERIC[(M[,D])]	支持	MySQL中"^"表示异或。 ● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
3	DEC[(M[,D])	支持	 输入格式: 当字符串入参全部被截断时不会报错,如'@123'; 只有被部分截断时才会报错,如'12@3'。
4	FIXED[(M[,D])]	不支持	-

表 4-3 浮点类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	FLOAT[(M,D	支持	分区表支持: FLOAT数据类型不支持KEY键值分区策略分区表。
			操作符: GaussDB中 "^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或。
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode=''的宽松模式下 报WARNING。
2	FLOAT(p)	支持	分区表支持: FLOAT数据类型不支持KEY键值分区策略分区表。
			● 操作符:数值类型使用^操作符,与 MySQL不一致,GaussDB中^操作符为取 指数运算。
			• 取值范围:定义精度p时,仅支持使用合法的整型数据类型。
			● 输出格式: 对于非法入参一律报错ERROR,不会在 sql_mode=''的宽松模式下报WARNING。

序号	MySQL数据 库	GaussDB数 据库	差异
3	DOUBLE[(M ,D)]	支持	分区表支持:DOUBLE数据类型不支持KEY 键值分区策略分区表。
			● 操作符: GaussDB中 "^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或。
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode=''的宽松模式下 报WARNING。
4	DOUBLE PRECISION[(M,D)]	支持	● 操作符:GaussDB中"^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode=''的宽松模式下 报WARNING。
5	REAL[(M,D)	支持	分区表支持:REAL数据类型不支持KEY值 分区策略分区表。
			● 操作符:GaussDB中"^"表示指数运算, 如需使用异或运算符,使用"#"替换; MySQL中"^"表示异或
			● 取值范围:精度M,标度D不支持浮点型数值输入,只支持整型数值输入。
			● 输出格式:对于非法入参一律报错 ERROR,不会在sql_mode="的宽松模式下 报WARNING。

表 4-4 序列整数

序号	MySQL数据 库	GaussDB数 据库	差异
1	SERIAL	不完全兼容	SERIAL介绍详见GaussDB 开发指南 数值类型 章节。
			规格上与MySQL的差异如下: CREATE TABLE test(f1 serial, f2 CHAR(20));
			 类型定义的差异, MySQL的serial是映射到 BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE, GaussDB 的serial是映射到INTEGER NOT NULL DEFAULT
			nextval('test_f1_seq'::regclass)。如: MySQL serial的定义: mysql> SHOW CREATE TABLE test\G ************************************
			`f1` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `f2` char(20) DEFAULT NULL, UNIQUE KEY `f1` (`f1`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 1 row in set (0.00 sec)
			GaussDB serial的定义 gaussdb=# \d+ test Table "public.test" Column Type Modifiers Storage Stats target Description
			+
			Has OIDs: no Options: orientation=row, compression=no, storage_type=USTORE
			● INSERT场景下serial类型DEFAULT值的差 异。如: MySQL插入serial的DEFAULT值 mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa'); Query OK, 1 row affected (0.00 sec)
			mysql> INSERT INTO test VALUES(10, 'aaaa'); Query OK, 1 row affected (0.00 sec)
			mysql> INSERT INTO test VALUES(DEFAULT, 'aaaa'); Query OK, 1 row affected (0.00 sec)
			mysql> SELECT * FROM test; ++ f1 f2 ++
			1 aaaa 10 aaaa 10 aaaa 11 aaaa

序号	MySQL数据 库	GaussDB数 据库	差异
			++ 3 rows in set (0.00 sec) GaussDB插入serial的DEFAULT值 gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa'); INSERT 0 1 gaussdb=# INSERT INTO test VALUES(10, 'aaaa'); INSERT 0 1 gaussdb=# INSERT INTO test VALUES(DEFAULT, 'aaaa'); INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2+

4.2.2.2 日期与时间数据类型

表 4-5 日期与时间数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
			GaussDB支持date数据类型,与MySQL相比规格上存在如下差异: ● 输入格式 - GaussDB只支持字符类型,不支持数值类型。如支持'2020-01-01'或'20200101'字符串格式,不支持20200101数值输入。MySQL支持数值输入转换为date类型。 - 分隔符: GaussDB不支持加号"+"、冒号":"作为年、月、日之间的分隔符,其他的符号都支持。MySQL所有符号均可作为分隔符。分隔符混合使用的某些场景也不支持,与MySQL也有差异,如'2020-01>01','2020/01+01'等,不建议混合使用分隔符,建议使用最常用的"-"、"/"作为分隔符。 - 无分隔符: 推荐使用完整格式,如'YYYYMMDD'或者'YYMMDD'。其他不完整的格式(包括超长格式)解析的规则与MySQL存在差异,可能报错或者解析的结果与MySQL不一致,不推荐使用。 ● 输出格式 GaussDB在sql_mode参数不包含'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、
			规则与MySQL存在差异,可能报错或者解析的结果与MySQL不一致,不推荐使用。 • 输出格式 GaussDB在sql_mode参数不包含 'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、日的值是0,但是输出时会按照年、月、日的顺序依次转换为合法的值,如date '0000-00-10' 转换为: 0002-12-10 BC。非法输入或者超过范围时,会报warning
			非法制入或者超过范围的,会报warning 信息,并返回0000-00-00值。MySQL对于包含0值年、月、日的date值会原样输出。 ■ 取值范围 GaussDB的范围是4713-01-01 BC~5874897-12-31 AD,支持公元前的日期,宽松模式下超过范围时,返回的是0值:0000-00-00,严格模式下会报错。MySQL的范围是0000-00-00~9999-12-31,宽松模式下超过范围后,各个场景下的表现并不一致,可能报错(如select查询语句中),也可能返回0000-00-00值(如insert时)。此差异会导致date类型作为函

序号	MySQL数据 库	GaussDB数 据库	差异
			 ● 操作符 - GaussDB仅支持date类型之间的比较操作符 "="、"!="、"<"、"<="、">"、">=",返回true或者false;date与interval类型的加法运算,返回结果为date类型;date与interval类型的减法运算,返回结果为date类型;date类型之间的减法运算,返回结果为interval类型。 - MySQL date类型和其他数值类型运算时,会先将date转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如: - MySQL: date + 数值,先将date类型转换为数值20200101,再与1相加,结果为数值类型20200102mysql> select date'2020-01-01' + 1; +
			 类型转换相比较MySQL, GaussDB仅支持date类型与char(n)、nchar(n)、datetime、timestamp类型之间的相互转换,不支持与binary、decimal、json、integer、unsigned integer、time类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致,参考数据类型转换章节的描述。

序号	MySQL数据 库	GaussDB数 据库	差异
2	DATETIME[(f sp)]	支持	GaussDB支持datetime数据类型,与MySQL 相比规格上存在如下差异:
			● 输入格式:
			- GaussDB只支持字符类型,不支持数值 类型。如支持'2020-01-01 10:20:30.123456'或 '20200101102030.123456'字符串格 式,不支持如 20200101102030.123456的数值类型 输入。MySQL支持数值输入转换为 datetime类型。
			- 分隔符: GaussDB不支持加号"+"、 冒号":"作为年、月、日之间的分隔符,其他的符号都支持。仅支持冒号 ":"作为时、分、秒之间的分隔符,其他的符号都不支持。分隔符混合使用的某些场景也不支持,与MySQL也有差异,不推荐使用。MySQL支持所有符号作为分隔符。
			- 无分隔符: GaussDB推荐使用完整格式 'YYYYMMDDhhmiss.ffffff'。其他不完整的格式(包括超长格式)解析的规则可能与MySQL存在差异,可能报错或者解析的结果与MySQL不一致,不推荐使用。
			● 输出格式:
			- 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的 格式,格式与MySQL无差异,且不受 DateStyle参数的影响。但是对于精度 部分,如果最后几位为0,GaussDB不 显示,MySQL会显示。
			- GaussDB在sql_mode参数不包含 'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、日值是0,但是输出时会按照年、月、日的顺序依次转换为合法的值,如datetime '0000-00-10 00:00:00' 转换为: 0002-12-10 00:00:00 BC。 非法输入或者超过范围时,会报warning信息,并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的datetime值会原样输出。
			● 取值范围 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。 294277-01-09 04:00:54.775807 AD 返回 的是infinity。对于超过范围的值,严格模

序号	MySQL数据 库	GaussDB数 据库	差异
			式下GaussDB会报错,MySQL是否报错取决于使用场景。一般查询场景不报错,而执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00值,MySQL根据使用场景可能报错,也可能返回0000-00-00 00:00:00值或者null值。这个差异会导致以datetime类型为入参的函数执行结果与MySQL也存在差异。
			● 精度 范围0~6,作为表列的类型时缺省为0,与 MySQL一致。对于 datetime[(p)] 'str' 表 达式场景,GaussDB将(p)作为精度解析, 缺省为6,将'str'按照p指定的精度格式化 成datetime类型。MySQL不支持 datetime[(p)] 'str'表达式。
			● 操作符 - GaussDB仅支持datetime类型之间的比较操作符"="、"!="、"<"、"<="、">=",返回true或者false; datetime与interval类型的加法运算,返回结果为datetime类型; datetime与interval类型的减法运算,返回结果为datetime类型; datetime类型; datetime类型; datetime类型; datetime类型; datetime类型; datetime类型之间的减法运算,返回结果为interval类型。
			 MySQL datetime类型和其他数值类型运算时,会先将datetime转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如: MySQL: datetime + 数值,先将datetime类型转换为数值20201010123456,再与1相加,结果为数值类型20201010123457 mysql> select cast('2020-10-10 12:34:56.123456' as datetime) + 1;
			+
			2020-10-11 12:34:56 (1 row)

序号	MySQL数据 库	GaussDB数 据库	差异
			将datetime类型与数值的运算结果作为函数的入参,可能导致函数的结果与MySQL也存在差异。 • 类型转换相比较MySQL,GaussDB仅支持datetime类型与char(n)、nchar(n)、timestamp类型之间的相互转换、datetime到date、time类型的转换(仅赋值和显式转换)。不支持与binary、decimal、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致,参考数据类型转换章节的描述。 • 时区 GaussDB支持datetime值中携带时区信息(时区偏移或者时区名),如'2020-01-01 12:34:56.123456 +01:00'或者'2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。MySQL不支持(5.7版本不支持,8.0及之后的版本支持)。 • GaussDB的datetime数据类型的表字段实际上会被转换为timestamp(p) without time zone类型,查询表信息或者使用工具导出的表结构,其字段的数据类型显示的是timestamp(p) without time zone类型,查询表信息或者使用工具导出的表结构,其字段的数据类型显示的是timestamp(p) without time zone,而不是datetime。MySQL显示的是datetime(p)。

序号	MySQL数据 库	GaussDB数 据库	差异
3	TIMESTAMP[(fsp)]	支持	GaussDB支持timestamp数据类型,与 MySQL相比规格上存在如下差异:
			● 输入格式
			- 只支持字符类型,不支持数值类型。如 支持'2020-01-01 10:20:30.123456'或 '20200101102030.123456'字符串格 式,不支持如 20200101102030.123456的数值类型 输入。MySQL支持数值输入转换为 timestamp类型。
			- 分隔符:不支持加号"+"、冒号":" 作为年、月、日之间的分隔符,其他的 符号都支持。仅支持冒号":"作为时、 分、秒之间的分隔符,其他的符号都不 支持。分隔符混合使用的某些场景也不 支持,与MySQL也有差异,不推荐使 用。MySQL支持所有符号作为分隔符。
			- 无分隔符:推荐使用完整格式 'YYYYMMDDhhmiss.ffffff'。其他不完整的格式(包括超长格式)解析的规则可能与MySQL存在差异,可能报错或者解析的结果与MySQL不一致,不推荐使用。
			● 输出格式
			- 统一为'YYYY-MM-DD hh:mi:ss.ffffff'的 格式,格式与MySQL无差异,且不受 DateStyle参数的影响。但是对于精度 部分,如果最后几位为0,GaussDB不 显示,MySQL会显示。
			- GaussDB在sql_mode参数不包含 'strict_trans_tables'选项(定义为宽松模式,否则为严格模式)时,允许年、月、日值是0,但是输出时会按照年、月、日的顺序依次转换为合法的值,如timestamp '0000-00-10 00:00:00' 转换为: 0002-12-10 00:00:00 BC。非法输入或者超过范围时,会报warning信息,并返回0000-00-00 00:00:00值。MySQL对于包含0值年、月、日的timestamp值会原样输出。
			● 取值范围 4713-11-24 00:00:00.000000 BC ~ 294277-01-09 04:00:54.775806 AD。 294277-01-09 04:00:54.775807 AD 返回 的是infinity。对于超过范围的值,严格模 式下GaussDB会报错,MySQL是否报错取 决于使用场景。一般查询场景不报错,而

序号	MySQL数据 库	GaussDB数 据库	差异
			执行DML SQL语句更改表属性的值时报错。宽松模式下GaussDB返回0000-00-00 00:00:00:00值,MySQL根据使用场景可能报错,也可能返回0000-00-00 00:00:00值或者null值。这个差异会导致以timestamp类型为入参的函数执行结果与MySQL也存在差异。
			 精度 范围0~6,作为表列的类型时缺省为0,与 MySQL一致。对于 timestamp[(p)] 'str' 表达式场景:
			– GaussDB将(p)作为精度解析,缺省为 6,将'str'按照p指定的精度格式化成 timestamp类型。
			- MySQL将timestamp 'str'的含义与GaussDB一致,缺省精度也为6。但是将timestamp(p) 'str'解析为函数调用,p作为timestamp函数的入参,结果返回一个timestamp类型的值,'str'作为投影列的别名。
			● 操作符
			- GaussDB仅支持timestamp类型之间的 比较操作符"="、"!="、"<"、 "<="、">"、">=",返回true或 者false; timestamp与interval类型的 加法运算,返回结果为timestamp类 型; timestamp与interval类型的减法运 算,返回结果为timestamp类型; timestamp类型之间的减法运算,返回 结果为interval类型。
			- MySQL timestamp类型和其他数值类型运算时,会先将timestamp转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如:
			MySQL: timestamp + 数值,先将timestamp类型 转换为数值20201010123456.123456,再与1相加,结果 为数值类型20201010123457.123456 mysql> select timestamp '2020-10-10 12:34:56.123456' + 1;
			timestamp '2020-10-10 12:34:56.123456' + 1
			20201010123457.123456
			1 row in set (0.00 sec)
			GaussDB: timestamp + 数值,数值类型会转换为 interval类型 1 day,然后相加得到新的timestamp。 gaussdb=# select timestamp '2020-10-10 12:34:56.123456' + 1;

序号	MySQL数据 库	GaussDB数 据库	差异
			?column? 2020-10-11 12:34:56.123456 (1 row) 将timestamp类型与数值的运算结果作为函数的入参,可能导致函数的结果与MySQL也存在差异。 ● 类型转换相比较MySQL,GaussDB仅支持timestamp类型与char(n)、varchar(n)、datetime类型之间的相互转换、timestamp到date、time类型的转换(仅赋值和显式转换)。不支持与binary、decimal、json、integer、unsigned integer类型之间的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致,参考数据类型转换章节
			 的描述。 ● 时区 GaussDB支持timestamp值中携带时区信息(时区偏移或者时区名),如 '2020-01-01 12:34:56.123456 +01:00'或者'2020-01-01 2:34:56.123456 CST'。GaussDB会将其转换为当前服务器时区的时间。如果更改服务器时区,timestamp类型的值输出时会转换为更改后时区的时间戳。MySQL不支持(5.7版本不支持,8.0及之后的版本支持)。
			GaussDB的timestamp数据类型的表字段实际上会被转换为timestamp(p) with time zone类型,查询表信息或者使用工具导出的表结构,其字段的数据类型显示的是timestamp(p) with time zone,而不是timestamp。MySQL显示的是timestamp(p)。

序号	MySQL数据 库	GaussDB数 据库	差异
4	TIME[(fsp)]	支持	GaussDB支持time数据类型,与MySQL相比 规格上存在如下差异:
			● 输入格式
			- 只支持字符类型,不支持数值类型。如 支持'1 10:20:30'或'102030'字符串格 式,不支持102030数值输入。MySQL 支持数值输入转换为time类型。
			- 分隔符: GaussDB仅支持冒号":"作为时、分、秒之间的分隔符,其他的符号都不支持。MySQL支持所有的符号作为分隔符。
			- 无分隔符:推荐使用完整格式,如 'hhmiss.ffffff'。其他不完整的格式(包 括超长格式)解析的规则可能与MySQL 存在差异,可能报错或者解析的结果与 MySQL不一致,不推荐使用。
			- 分、秒、精度输入负数时,GaussDB数据库可能会忽略第一个负数开始的部分,涉及的部分解析为0,如: '00:00:-10' 解析结果为 '00:00:00'。 也可能报错,如: '00:00:-10000' 会解析报错。取决于输入值的范围。而MySQL数据库统一报错。
			 输出格式 统一为hh:mi:ss.ffffff的格式,格式与 MySQL无差异。但是对于精度部分,如果 最后几位为0,GaussDB不显示,MySQL 会显示。
			• 取值范围 -838:59:59.000000 ~ 838:59:59.000000, 与MySQL一致。对于超过范围的值,宽松模式下GaussDB无论是查询还是insert/ update等DML操作,返回的值都是就近的边界值: -838:59:59或838:59:59。MySQL是查询时报错,DML操作返回的值才是就近边界值,场景上存在差异。此差异会导致time类型作为函数入参时,函数返回的结果也存在差异。
			• 精度 范围0~6,作为表列的类型时缺省为0,与 MySQL一致。对于 time(p) 'str' 表达式场 景,GaussDB将(p)作为精度解析,缺省为 6,将'str'按照p指定的精度格式化成time 类型。MySQL是解析为time函数,p是入 参,'str'是投影列的别名。
			● 操作符

序号	MySQL数据 库	GaussDB数 据库	差异
与)	店	- GaussDB仅支持time类型之间的比较操作符"="、"!="、"<"、"<="、"<"、"<="、"*>"、"=",返回true或者false;time与interval类型的加法运算,返回结果为time类型;time与interval类型的加法运算,返回结果为interval类型。 - MySQL time类型和其他数值类型运算时,会先将time转换为数值类型,然后按照数值类型运算,结果也为数值类型。与GaussDB存在差异。如: MySQL: time + 数值,先将time类型转换为数值123456,再与1相加,结果为数值类型123457mysql> select time '12:34:56' + 1;
			致,参考 数据类型转换 章节的描述。

序号	MySQL数据 库	GaussDB数 据库	差异
5	YEAR[(4)]	支持	GaussDB支持year数据类型,与MySQL相比 规格上存在如下差异:
			● 操作符
			 GaussDB仅支持year类型之间的比较操作符"="、"!="、"<"、"<="、"<="、">"、">=",返回true或者false。 GaussDB仅支持year类型与int4类型之间的算术操作符"+"、"-",返回整型值,MySQL是返回无符号整型值。
			 类型转换相比较MySQL, GaussDB仅支持year类型与int4类型的转换,仅支持int4、varchar、numeric、date、time、timestamp、timestamptz类型到year类型的转换。集合等场景和复杂表达式场景下公共类型的确定原则与MySQL也不一致,参考数据类型转换章节的描述。
6	INTERVAL	支持	GaussDB支持INTERVAL数据类型,但 INTERVAL在MySQL中为表达式,同时存在以 下差异:
			不支持字符串类型的日期输入作为运算,如: select '2023-01-01' + interval 1 day。
			● 不支持interval expr unit语法中,expr为 负整数或浮点数的输入,如:select date'2023-01-01' + interval -1 day。
			 不支持interval expr unit语法中, expr为 运算表达式的输入,如: select date'2023-01-01' + interval 4/2 day。
			 interval表达式参与运算时,返回值固定为 datetime类型,MySQL为datetime或date 类型。运算的逻辑与原有GaussDB保持一 致,与MySQL有差异。
			• interval expr unit语法中,expr数值支持的范围会根据unit单位的不同有所差异,最大可支持的范围为[-2147483648, 2147483647]。超过范围时,严格模式报error,宽松模式报warning并返回0值。
			• interval expr unit语法中,expr指定的字段数量大于unit预期的字段数量时,在严格模式,报error;在宽松模式,报warning并返回0值。如unit取值为DAY_HOUR,预期的字段数量为2,expr取值为'1-2-3',字段数量为3。

4.2.2.3 字符串数据类型

表 4-6 字符串数据类型

序号	MySQ数据库	GaussDB数 据库	差异
1	CHAR[(M)]	支持	 ● 输入格式 - GaussDB自定义函数参数和返回值不支持长度校验,存储过程参数不支持长度校验,同时也不支持在PAD_CHAR_TO_FULL_LENGTH打开时补齐正确的空格,MySQL支持。 - GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 ● 语法 GaussDB的 Cast (expr as char)语法无法根据输入的字符串长度转成对应的类型,只支持转成varchar类型。不支持cast(''as char())将空串转成char(0)类型。MySQL支持按长度转成对应的类型。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 - GaussDB除以0会报错,MySQL返回null。 - "~":GaussDB返回负数,MySQL返回客字节无符号整数。 - "^":GaussDB表示次方幂,MySQL表示按位异或。

序号	MySQ数据库	GaussDB数 据库	差异
2	VARCHAR(M	支持	● 输入格式:
	,		- GaussDB的自定义函数参数和返回值不 支持长度校验,存储过程参数不支持长 度校验,MySQL支持。
			- GaussDB的自定义函数和存储过程中的 临时变量支持长度校验以及严格宽松模 式下的报错和截断告警,MySQL不支 持。
			- GaussDB不支持转义字符输入,不支持 ""双引号输入,MySQL支持。
			● 操作符
			- GaussDB能正常转成浮点型的字符串与 整型值加减乘除求余,返回值是整型 值,MySQL是返回浮点型。
			– GaussDB除以0会报错,MySQL返回 null。
			– "~": GaussDB返回负数,MySQL返 回8字节无符号整数。
			- "^": GaussDB表示次方幂,MySQL 表示按位异或。
3	TINYTEXT	支持	● 输入格式
			- GaussDB不支持长度限制255字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。 MySQL支持。
			- GaussDB不支持转义字符输入,不支持 ""双引号输入,MySQL支持。
			● 操作符
			- GaussDB能正常转成浮点型的字符串与 整型值加减乘除求余,返回值是整型 值,MySQL是返回浮点型。
			– GaussDB除以0会报错,MySQL返回 null。
			– "~": GaussDB返回负数,MySQL返 回8字节无符号整数。
			- "^": GaussDB表示次方幂,MySQL 表示按位异或。

序号	MySQ数据库	GaussDB数 据库	差异
4	TEXT	支持	● 输入格式 - GaussDB不支持长度限制65535字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。 - GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 ● 操作符 - GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 - GaussDB除以0会报错,MySQL返回null。 - "~":GaussDB返回负数,MySQL返回8字节无符号整数。 - "^":GaussDB表示次方幂,MySQL表示按位异或。
5	MEDIUMTEX T	支持	● 输入格式 - GaussDB不支持长度限制16777215字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。 - GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 - 操作符 - GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 - GaussDB除以0会报错,MySQL返回null。 - "~":GaussDB返回负数,MySQL返回8字节无符号整数。 - "^":GaussDB表示次方幂,MySQL表示按位异或。

序号	MySQ数据库	GaussDB数 据库	差异
6	LONGTEXT	支持	 输入格式 GaussDB只支持不超过1G,MySQL支持4G-1字节长度。 GaussDB不支持转义字符输入,不支持""双引号输入,MySQL支持。 操作符 GaussDB能正常转成浮点型的字符串与整型值加减乘除求余,返回值是整型值,MySQL是返回浮点型。 GaussDB除以0会报错,MySQL返回null。 "~":GaussDB返回负数,MySQL返回8字节无符号整数。 "^":GaussDB表示次方幂,MySQL表示按位异或。
7	ENUM('value 1','value2',)	不支持	-
8	SET('value1',' value2',)	支持	-

4.2.2.4 二进制数据类型

表 4-7 二进制数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BINARY[(M)]	不支持	-
2	VARBINARY(M)	不支持	-

序号	MySQL数据 库	GaussDB数 据库	差异
3	TINYBLOB	支持	 取值范围:表现规格为BYTEA类型。不支持长度限制255字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。
			● 输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常
			用逻辑运算符或、与、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^")。
4	BLOB	支持	 取值范围:表现规格为BYTEA类型。不支持长度限制65535字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。
			输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常
			用逻辑运算符与、或、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^"。
5	MEDIUMBLO B	支持	• 取值范围:表现规格为BYTEA类型。不支持长度限制16777215字节(最大不超过1G),不支持超限后,根据严格宽松模式报错和截断告警。MySQL支持。
			● 输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			● 操作符:不支持算数运算符"+"、 "-"、"*"、"/"、"%";不支持常
			用逻辑运算符与、或、非(" "、 "&&"、"!");不支持常用位运算符 "~"、"&"、" "、"^"。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGBLOB	支持	取值范围:表现规格为BYTEA类型。只支持不超过1G,具体范围参照bytea数据类型集中式和分布式规格。
			输入格式:不支持转义字符输入,不支持 ""双引号输入。
			● 输出格式:对于'\0'字符,查询结果表现为 "\000",使用jdbc驱动的getBytes接口 获取表现为'\0'字符。
			 操作符:不支持算数运算符"+"、"-"、"*"、"/"、"%";不支持常用逻辑运算符或、与、非(" "、"&&"、"!");不支持常用位运算符"~"、"&"、" "、"^")。
7	BIT[(M)]	不支持	-

4.2.2.5 JSON 数据类型

表 4-8 JSON 数据类型

序号	MySQ数据库	GaussDB数据库
1	JSON	不完全兼容

4.2.2.6 数据类型支持的属性

表 4-9 数据类型支持的属性

序号	MySQ数据库	GaussDB数据库
1	NULL	支持
2	NOT NULL	支持
3	DEFAULT	支持
4	ON UPDATE	支持
4	PRIMARY KEY	支持
5	AUTO_INCREMENT	支持
6	CHARACTER SET name	支持
7	COLLATE name	支持

4.2.2.7 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换:

- 操作符(比较操作符、运算操作符等)的操作数的数据类型不一致。常见于查询 条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新(包括insert、update、merge、replace等)的目标列,数据的 类型和列的定义类型不一致。
- 显式的类型转换: cast(expr as datatype),将expr表达式类型转换为datatype类型。
- 集合运算(UNION、MINUS、EXCEPT、INTERSECT)确定最终投影列的目标数据类型后,各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景,根据不同表达式的数据类型,来决定用于比较或者最终结果的目标数据类型。
 - DECODE
 - CASE WHEN
 - lexpr [NOT] IN (expr_list)
 - BETWEEN AND
 - JOIN USING(a,b)
 - GREATEST和LEAST
 - NVL 和 COALESCE

GaussDB和MySQL数据库对于数据类型转换、转换的目标数据类型有着完全不同的规则。如下示例体现了两者处理的差异:

1. 数据类型转换规则的差异:

- GaussDB数据库对于不同数据类型之间的转换规则有明确的定义:
 - 是否支持转换:pg_cast系统表中是否定义两种类型的转换路径,没有定义则 不支持。
 - 支持转换的场景:支持任意场景转换、仅支持显式(cast表达式)转换、仅 支持赋值时转换。不支持的场景下即使定义了转换路径,也不能进行数据类 型转换。

MySQL数据库支持任意两种数据类型之间做转换。

由于存在以上差异,基于MySQL数据库的应用程序向GaussDB数据库迁移时,SQL语句可能由于不支持不同数据类型之间的转换而报错。或者支持转换的场景下,转换的规则有差异导致SQL语句执行的结果不同。

推荐的做法是: SQL语句中尽量使用相同的数据类型做比较或者赋值等操作,避免因为数据类型转换导致非预期结果或者性能损耗。

2. 选择目标数据类型的规则差异:

对于有些场景,比较的数据类型或者返回的数据类型需要综合考虑多个表达式的类型才能确定。比如UNION运算中,不同SELECT语句中相同位置的投影列具有不同的数据类型,查询结果的最终数据类型,需要由各个SELECT语句投影列的数据类型共同确定。

确定目标数据类型的规则,GaussDB数据库和MySQL数据库存在体系上的差异。

- GaussDB数据库规则:
 - 操作符的操作数类型不一致时,并不是将操作数的类型统一转换为目标类型 再计算。而是直接注册两个数据类型的操作符,操作符处理中定义两个不同 类型的处理规则。此方式不存在类型隐式转换,但自定义的处理规则隐含了 转换的操作。
 - 集合运算和表达式场景,确定目标数据类型的规则:
 - 如果所有类型都相同,则此类型即为目标类型。
 - 两个数据类型如果不同,检查数据类型是否属于同一种类的数据类型, 如数值类型、字符类型、日期时间类型等。不属于同一种类的数据类型,无法确定目标类型,此时SQL语句执行会报错。
 - 对于category属性(在pg_type系统表中定义)相同的数据类型,具有 preferred属性(在pg_type系统表中定义)的数据类型会被选为目标类型。或者操作数1能转换为操作数2(没有转换路径),而操作数2无法转换为操作数1或数值类型优先级小于操作数2,则选择操作数2作为目标类型。
 - 如果涉及到3个及以上的数据类型,确定目标类型的规则为: common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3),依次迭代处理,得 到最终的结果。
 - 对于IN和NOT IN表达式,如果根据以上规则无法确认目标类型,会将lexpr与expr_list中每一个表达式单独按照等值操作符(=)逐个比较。
 - 精度的确定:以最终选定的表达式的精度作为最终结果。
- MySQL数据库规则:
 - 操作符的操作数类型不一致时,先按照如下规则确定目标类型。确定后将类型不一致的操作数转换成目标类型后再做处理。
 - 两个参数都是string类型,则都按照string类型比较。
 - 两个参数都是integer类型,则都按照integer类型比较。
 - 十六进制数值如果不与数值比较,则当做二进制字符串比较。

- 一个参数是datetime/timestamp类型,另一个参数是常量,将常量转换为时间戳类型然后比较。
- 如果其中一个参数是decimal类型,比较时使用的数据类型取决于另外一个参数。另外一个是decimal或者integer类型时,按照decimal类型;另外一个是其他类型,按照real类型比较。
- 其他场景都转换为 real 类型后比较。
- 集合运算和表达式场景,确定目标数据类型的规则如下:
 - 建立任意两个类型之间的目标类型矩阵。给定两个类型,通过矩阵即可以确定目标类型。
 - 如果涉及到3个及以上的数据类型,确定目标类型的规则为: common_type(type1,type2,type3) = common_type(common_type(type1,type2),type3),依次迭代处理,得 到最终的结果。
 - 如果目标类型是integer类型,且各个表达式类型包含有符号和无符号的混合场景,则会将类型提升到更高精度的integer类型。符号的确定:所有表达式都是无符号时,结果才为无符号,否则结果为有符号。
 - 精度确定:以表达式中的最大精度作为最终结果。

从以上规则可知: GaussDB和MySQL数据库在数据类型的转换规则上有很大差异,不能直接对比。在上述场景下,SQL语句的执行结果可能和MySQL数据库不一致。当前版本推荐各个表达式使用相同的类型,或提前使用cast转换成需要的类型来规避差异。

4.2.3 系统函数

GaussDB数据库兼容绝大多数MySQL的系统函数,但存在部分差异。如未列出,函数 行为默认为GaussDB原生行为。

□说明

GaussDB MySQL兼容性的绝大部分系统函数目前均存在返回值与MySQL精度不一致(结果后面 0的位数)的问题,这是由于部分数据类型在某些场景下仍存在精度丢失问题,无法正确获取精度,导致目前部分函数未完全做到适配。

4.2.3.1 流量控制函数

表 4-10 流量控制函数列表

序号	MySQL 数据库	GaussD B数据库	差异
1	IF()	支持	expr1入参仅支持bool类型。非bool类型入参若不能 转换为bool类型则报错。
			● 若expr2、expr3两入参类型不同且两类型间不存在隐式转换函数则报错。
			● 两入参类型相同时,返回该入参类型。
			 若expr2、expr3两入参类型分别为NUMERIC、 STRING或TIME其中一个时,输出为text类型, MySQL输出为varchar类型。
2	IFNULL()	支持	● 若expr1、expr2两入参类型不同且两类型间不存在隐式转换函数则报错。
			● 两入参类型相同时,返回该入参类型。
			● 若expr1、expr2两入参类型范畴分别为NUMEIRC、 STRING或TIME其中一个时,输出为text类型, MySQL输出为varchar类型。
			● 两入参类型第一个入参为float4,另一个为bigint或 unsigned bigint时返回double类型,MySQL返回 float类型。

序号	MySQL 数据库	GaussD B数据库	差异
3	NULLIF()	支持	 GaussDB中NULLIF()类型推导遵从以下逻辑: 1. 如果两个参数的数据类型不同,且两入参类型存在等值比较操作符,则返回对应等值操作符对应的左值类型,否则会对两入参类型进行强制类型
			兼容。 2. 若强制类型兼容后,存在等值比较操作符,则返回强制类型兼容后对应等值操作符的左值类型。 3. 若强制类型兼容后,仍找不到对应等值操作符,则报错。两入参类型存在等值比较操作符gaussdb=# select pg_typeof(nullif(1::int2, 2::int8)); pg_typeof
			两入参类型不存在等值比较操作符,且强制类型兼容后也不存在等值比较操作符 gaussdb=# SELECT nullif(1::bit, '1'::MONEY); ERROR: operator does not exist: bit = money LINE 1: SELECT nullif(1::bit, '1'::MONEY); HINT: No operator matches the given name and argument type(s). You might need to add explicit type casts. CONTEXT: referenced column: nullif
			MySQL输出类型仅与第一个入参类型有关:1. 第一个入参为tinyint、smallint、mediumint、int、bool时,输出为int类型。
			 第一个入参为bigint时,输出为bigint类型。 第一个入参为unsigned tinyint、unsigned smallint、unsigned mediumint、unsigned int、bit时,输出为unsigned int类型。 第一个入参为unsigned bigint时,输出为unsigned bigint时,输出为unsigned bigint。
			5. 第一个入参为浮点型即float、double、real时,输出为double类型。 6. 第一个入参类型为decimal或numeric类型时,输出为decimal类型。
			7. 第一个入参类型为时间类型或字符串类型即date、time、date、datetime、timestamp、char、varchar以及tinytext、enum、set时,输出为varchar类型。
			8. 第一个入参类型为text、mediumtext、longtext 时,输出为longtext类型。

序号	MySQL 数据库	GaussD B数据库	差异	
			9. 第一个入参类型为tinyblob时,输出为varbinary 类型。	
			10.第一个入参类型为mediumblob或longblob时,输 出为longblob类型。	
			11.第一个入参为blob时,输出为blob类型。	
4	ISNULL()	支持	GaussDB中返回值为boolean类型的t或f,MySQL中返回值为int类型的1或0。	

4.2.3.2 日期和时间函数

山 说明

以下为GaussDB MySQL兼容性日期时间函数公共说明。

开发指南中函数入参为时间类型表达式的情况:

时间类型表达式主要包括TEXT、DATETIME、DATE或TIME,但所有可以隐式转换为时间表达式的类型都可以作为入参,比如数字类型可以通过先隐式转化为TEXT,再作为时间类型表达式生效。

但是生效的情况根据函数有所不同,比如:DATEDIFF由于只计算日期的差值,所以时间表 达式会当做日期来解析。TIMESTAMPDIFF由于计算时间差值时是根据UNIT来决定的,所以 会根据UNIT选择将时间表达式当做DATE或者TIME或者DATETIME来解析。

• 函数入参为无效日期的情况:

一般而言,日期时间函数支持DATE、DATETIME的范围和MySQL保持一致。DATE支持的范围为'0000-01-01'到'9999-12-31',DATETIME支持的范围为'0000-01-01 00:00:00'到 '9999-12-31 23:59:59'。虽然GaussDB支持的DATE、DATETIME范围大于MySQL,但是越界仍然算无效日期。

大部分时间函数会告警并返回NULL,只有能通过cast正常转换的日期,才是正常合理的日期。

• 函数入参的分隔符场景:

对于时间函数,处理入参时会将所有非数字字符视作分隔符,然后根据数字所处的位置进行计算,推荐使用标准写法,年月日之间使用-分隔符,时分秒之间使用:分隔符,毫秒之前通过.来进行分隔。

易错场景: "SELECT timestampdiff(hour, '2020-03-01 00:00:00', '2020-02-28 00:00:00+08');" B兼容模式数据库中时间函数不会自动计算时区,所以此处+08并未识别为时区,而是+作为分隔符当作秒来进行计算。

GaussDB的日期时间函数的大部分功能场景与MySQL一致,但仍有差异,一些差异如下:

函数入参为NULL时,函数返回NULL,无warning或error告警。这些函数包括:
from_days、date_format、str_to_date、datediff、timestampdiff、date_add、
subtime、month、time_to_sec、to_days、to_seconds、dayname、
monthname、convert_tz、sec_to_time、addtime、adddate、date_sub、
timediff、last_day、weekday、from_unixtime、unix_timestamp、subdate、
day、year、weekofyear、dayofmonth、dayofyear、week、yearweek、
dayofweek、time_format、hour、minute、second、microsecond、quarter、
get_format、extract、makedate、period_add、timestampadd、period_diff、
utc_time、utc_timestamp、maketime、curtime
 示例:

```
gaussdb=# select day(null);
day
----
(1 row)
```

纯数字入参个别函数与MySQL有差异,不带引号的数字入参统一转成text入参来处理。

示例:

```
gaussdb=# select day(19231221.123141);
WARNING: Incorrect datetime value: "19231221.123141"
CONTEXT: referenced column: day
day
-----
(1 row)
```

时间日期运算函数: adddate、subdate、date_add、date_sub。当运算后的结果为日期时,支持的范围为[0000-01-01, 9999-12-31], 当运算后的结果为日期时间时,支持的范围为[0000-01-01 00:00:00:0000000, 9999-12-31 23:59:59.99999], 当运算后的结果超过支持的范围时,在严格模式下报error,在宽松模式下报warning。另外,当运算后的日期结果在范围[0000-01-01, 0001-01-01]中时,GaussDB正常返回结果,MySQL返回'0000-00-00'。

示例:

```
gaussdb=# select subdate('0000-01-01', interval 1 hour);
ERROR: Datetime function: datetime field overflow
CONTEXT: referenced column: subdate

gaussdb=# select subdate('0001-01-01', interval 1 day);
    subdate
------
0000-12-31

(1 row)
```

 对于日期和时间函数的date或datetime类型入参,含有0月或0日时为非法值,在 严格模式下报error;在宽松模式,当输入为字符串或数字时,报warning,输入 为date或datetime类型时视为上一年12月或上一月最后一日处理。

对于cast函数,转换为date、datetime时,严格模式下会报error。宽松模式下不会报warning,而是视为上一年12月或上一月最后一日处理,需要注意此区别。 MySQL对于包含0年、0月或0日的情况会原样输出。

示例:

● 若函数入参为numeric数据类型,在非法输入的情况下不会产生报错,会把入参当做0值处理。

示例:

最多保留6位小数,不保留后置都为0的小数。

示例:

时间函数参数为字符串时,只保证年月日之间使用"-"分隔,时分秒之间使用 ":"分隔时结果正确。

示例:

```
gaussdb=# select adddate('20-12-12',interval 1 day);
adddate
------
2020-12-13
(1 row)
```

在MySQL中,当函数的返回值为varchar时,在GaussDB中,函数对应的返回值为 text。

```
-- GaussDB中函数的返回值。
gaussdb=# SELECT pg_typeof(adddate('2023-01-01', 1));
pg_typeof
---------
text
(1 row)
-- MySQL中函数的返回值。
mysql> CREATE VIEW v1 AS SELECT adddate('2023-01-01', 1);
Query OK, 0 rows affected (0.00 sec)

mysql> DESC v1;
+------+
Field | Type | Null | Key | Default | Extra |
+-----+
| adddate('2023-01-01', 1) | varchar(29) | YES | NULL | |
+------+
| 1 row in set (0.00 sec)
```

表 4-11 日期与和时间函数列表

序号	MySQL数据库	GaussDB数 据库	差异
1	ADDDATE()	支持	此函数的表现会因为interval表达式的 差异与MySQL有差异,具体可见 INTERVAL差异说明。

序号	MySQL数据库	GaussDB数 据库	差异
2	ADDTIME()	支持	 MySQL对第二入参为DATETIME样 式字符串返回NULL, GaussDB可 以计算。
			● 入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。
			MySQL中ADDTIME函数如果第一个参数是动态参数(例如在预准备语句中),则返回类型为TIME。否则,函数的解析类型派生自第一个参数的解析类型。GaussDB中ADDTIME函数的返回值规则如下:
			- 第一个入参为date,第二个入 参为date,返回值为time。
			- 第一个入参为date,第二个入 参为text,返回值为text。
			– 第一个入参为date,第二个入 参为datetime,返回值为 time。
			- 第一个入参为date,第二个入 参为time,返回值为time。
			- 第一个入参为text,第二个入参 为date,返回值为text。
			- 第一个入参为text,第二个入参为text,返回值为text。
			- 第一个入参为text,第二个入参为datetime,返回值为text。
			- 第一个入参为text,第二个入参为time,返回值为text。
			- 第一个入参为datetime,第二 个入参为date,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为text,返回值为text。
			- 第一个入参为datetime,第二 个入参为datetime,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为time,返回值为 datetime。
			- 第一个入参为time,第二个入 参为date,返回值为time。

序号	MySQL数据库	GaussDB数 据库	差异
			 第一个入参为time,第二个入参为text,返回值为text。 第一个入参为time,第二个入参为datetime,返回值为time。 第一个入参为time,第二个入参为time,返回值为time。
3	CONVERT_TZ()	支持	-
4	CURDATE()	支持	-
5	CURRENT_DATE(), CURRENT_DATE	支持	-
6	CURRENT_TIME(), CURRENT_TIME	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的;MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示;MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,其他均报错;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
7	CURRENT_TIMESTAM P(), CURRENT_TIMESTAMP	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的;MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示;MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,超过6的整型值,作为返回时间的精度的输出时间值;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。

序号	MySQL数据库	GaussDB数 据库	差异
8	CURTIME()	支持	GaussDB此函数输入字符串或者非整型值,会被隐式转成整型,然后再校验精度,[0,6]范围之外的会报错,范围之内会正常输出时间值;MySQL直接报错。GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的;MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示;MySQL会显示,GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,其他均报错;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
9	YEARWEEK()	支持	-
10	DATE_ADD()	支持	此函数的表现会因为interval表达式的 差异与MySQL有差异,具体可见 INTERVAL差异说明。
11	DATE_FORMAT()	支持	-
12	DATE_SUB()	支持	此函数的表现会因为interval表达式的 差异与MySQL有差异,具体可见 INTERVAL差异说明。
13	DATEDIFF()	支持	-
14	DAY()	支持	-
15	DAYNAME()	支持	-
16	DAYOFMONTH()	支持	-
17	DAYOFWEEK()	支持	-
18	DAYOFYEAR()	支持	-
19	EXTRACT()	支持	-
20	FROM_DAYS()	支持	-
21	FROM_UNIXTIME()	支持	-
22	GET_FORMAT()	支持	-
23	HOUR()	支持	-
24	LAST_DAY	支持	-

序号	MySQL数据库	GaussDB数 据库	差异
25	LOCALTIME(), LOCALTIME	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的;MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示;MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,其他整型值直接报错;MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
26	LOCALTIMESTAMP, LOCALTIMESTAMP()	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示; MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,超过6的整型值,会告警并按照精度6输出时间值; MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。
27	MAKEDATE()	支持	-
28	MAKETIME()	支持	与MySQL相比, 入参为NULL时, GaussDB不支持maketime函数自嵌 套,MySQL支持。
29	MICROSECOND()	支持	-
30	MINUTE()	支持	-
31	MONTH()	支持	-
32	MONTHNAME()	支持	-
33	NOW()	支持	GaussDB的按精度输出的时间值(小数点后的值)是四舍五入的; MySQL是直接截断的。GaussDB按精度输出的时间值(小数点后的值)末尾0都不显示; MySQL会显示。GaussDB只支持输入[0,6]范围内的整型值,作为返回时间的精度,超过6的整型值,会告警并按照精度6输出时间值; MySQL的精度值有效值是[0,6],但是输入的整型值内部会对256求余(例257,会返回精度1的时间值)。

序号	MySQL数据库	GaussDB数 据库	差异
34	PERIOD_ADD()	支持	当入参period或结果小于0时, GaussDB参考MySQL8.0.x版本的表 现,报错处理。MySQL5.7会发生整 数回绕,导致计算结果异常。
35	PERIOD_DIFF()	支持	当入参或结果小于0时,GaussDB参考MySQL8.0.x版本的表现,报错处理。MySQL5.7会发生整数回绕,导致计算结果异常。
36	QUARTER()	支持	-
37	SEC_TO_TIME()	支持	-
38	SECOND()	支持	-
39	STR_TO_DATE()	支持	返回值与MySQL有差异,GaussDB返 回的是text,MySQL返回的是 datetime、date。
40	SUBDATE()	支持	此函数的表现会因为interval表达式的 差异与MySQL有差异,具体可见 INTERVAL 差异说明 。

序号	MySQL数据库	GaussDB数 据库	差异
41	SUBTIME()	支持	 MySQL对第二入参为DATETIME样 式字符串返回NULL, GaussDB可 以计算。
			● 入参取值范围为['0001-01-01 00:00:00', 9999-12-31 23:59:59.999999]。
			MySQL中SUBTIME函数如果第一个参数是动态参数(例如在预准备语句中),则返回类型为 TIME。否则,函数的解析类型派生自第一个参数的解析类型。GaussDB中SUBTIME函数的返回值规则如下:
			- 第一个入参为date,第二个入 参为date,返回值为time。
			- 第一个入参为date,第二个入 参为text,返回值为text。
			- 第一个入参为date,第二个入 参为datetime,返回值为 time。
			- 第一个入参为date,第二个入 参为time,返回值为time。
			- 第一个入参为text,第二个入参 为date,返回值为text。
			- 第一个入参为text,第二个入参为text,返回值为text。
			- 第一个入参为text,第二个入参为datetime,返回值为text。
			- 第一个入参为text,第二个入参为time,返回值为text。
			- 第一个入参为datetime,第二 个入参为date,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为text,返回值为text。
			- 第一个入参为datetime,第二 个入参为datetime,返回值为 datetime。
			- 第一个入参为datetime,第二 个入参为time,返回值为 datetime。
			- 第一个入参为time,第二个入 参为date,返回值为time。

序号	MySQL数据库	GaussDB数 据库	差异
			 第一个入参为time,第二个入参为text,返回值为text。 第一个入参为time,第二个入参为datetime,返回值为time。 第一个入参为time,第二个入参为time,返回值为time。
42	SYSDATE()	支持	MySQL入参整型值会按照一字节最大值255整数回绕,GaussDB不回绕。
43	YEAR()	支持	-
44	TIME_FORMAT()	支持	-
45	TIME_TO_SEC()	支持	-
46	TIMEDIFF()	支持	-
47	WEEKOFYEAR()	支持	-
48	TIMESTAMPADD()	支持	-
49	TIMESTAMPDIFF()	支持	-
50	TO_DAYS()	支持	-
51	TO_SECONDS()	支持	-
52	UNIX_TIMESTAMP()	支持	返回值与MySQL有差异,GaussDB返 回的是numeric,MySQL返回的是 int。
53	UTC_DATE()	支持	MySQL支持无括号调用,GaussDB不
54	UTC_TIME()	支持	支持。MySQL入参整型值会按照一字 节最大值255整数回绕。
55	UTC_TIMESTAMP()	支持	MySQL入参只支持0-6整数, GaussDB支持可以隐式转为0-6的输 入。
56	WEEK()	支持	-
57	WEEKDAY()	支持	-

4.2.3.3 字符串函数

表 4-12 字符串函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	BIN()	支持	函数入参1支持类型存在差异, GaussDB入参1支持类型如下:
			● 整数类型: tinyint、smallint、 mediumint、int、bigint
			● 无符号整数类型: tinyint unsigned、smallint unsigned、int unsigned、 bigint unsigned
			 字符和文本类型: char, varchar、tinytext, text, mediumtext, longtext, 仅支 持纯数字整数字符串, 且整数 范围在bigint范围内。
			● 浮点类型: float、real、 double
			● 定点类型: numeric、 decimal、dec
			布尔类型: bool
2	CONCAT()	支持	无论参数的数据类型如何,concat 返回值的数据类型始终为text; MySQL的concat在含有二进制类 型参数时,返回值为二进制类型。
3	CONCAT_WS()	支持	无论参数的数据类型如何,concat_ws返回值的数据类型始终为text; MySQL的concat_ws在含有二进制类型参数时,返回值为二进制类型,其他情况返回值为字符串类型。

序号	MySQL数据库	GaussDB数据库	差异
4	ELT()	支持	 差异 1. 函数入参1支持类型存在差异,GaussDB入参1支持类型如下: 整数类型: tinyint、smallint、mediumint、int、bigint 无符号整数类型: tinyint unsigned、smallint unsigned、smallint unsigned、int unsigned 字符和文本类型: char, varchar、tinytext, text, mediumtext, longtext, 仅支持纯数字符串,且整数范围在bigint范围内。 浮点类型: float、real、double 定点类型: numeric、decimal、dec 布尔类型: bool 2. 函数入参2支持类型中存差异,GaussDB入参2支持类型如下: 整数类型: tinyint、smallint、mediumint、int、bigint 无符号整数类型: tinyint unsigned、smallint unsigned、smallint unsigned、int unsigned、bigint unsigned 字符和文本类型: char, varchar、tinytext, text, mediumtext, longtext 浮点类型: float、real、double 定点类型: numeric、decimal、dec 布尔类型: bool 大对象类型: tinyblob, blob, mediumblob, longblob 日期类型: datetime,
			timestamp,date,time

序号	MySQL数据库	GaussDB数据库	差异
5	FIELD()	支持	函数入参为在bigint最大值~ bigint unsigned最大值范围内的数字,存在不兼容。 函数入参为浮点型float(m, d)、double(m, d)、real(m, d)时精度更高,存在不兼容。
6	FIND_IN_SET()	支持	当数据库encoding = 'SQL_ASCII' 时,不支持默认的大小写判断规 则,即在用户不指定字符集规则的 情况下,大写与小写区分判断。
7	INSERT()	支持	● Int64类型传参有范围限制,一旦超出-9223372036854775808~9223372036854775807范围会直接报错,MySQL对数值类型传参范围无限制,异常会告警按照上限或下限数值处理。 ● 字符串传参有限制,入参text类型字符串长度最大为2^30-5字节,入参bytea类型字符串长度最大为2^30-512字节。 ● s1和s2任意参数为bytea类型时,涉及到结果出现非法字符的情况可能展示结果与MySQL有差异但是字符编码与MySQL是一致的。
8	LOCATE()	支持	入参1为bytea类型,入参2为text 类型时,GaussDB与MySQL行为 存在差异。

序号	MySQL数据库	GaussDB数据库	差异
9	MAKE_SET()	支持	bits参数为整型时,最大范围支持到int128,低于MySQL范围。
			 bits参数为日期类型datetime、 timestamp、date、time,由 于时间类型转整型与MySQL存 在差异,目前均未做支持。
			 bit类型或bool类型由于此类数据类型Gauss与MySQL存在差异,返回结果导致的差异为GaussDB与MySQL固有差异。bits入参为bool类型,str入参为bit类型与bool类型均不做支持。
			bits入参为字符串或文本类型 时,仅支持纯整型数字形式, 其他形式存在差异。且纯整型 数字范围限制在bigint范围。
			● str入参整型数值超过正负81个 9,返回值与MySQL有差异。
			• str入参当以科学计数法表示 时,GaussDB末尾0值会显示, MySQL不显示,以科学计数法 打印,此为固有差异。

序号 MySQL数据库 GaussDB数据库 差异	
DUOTE() Description Descri	字符、"、", B未在会引生的, "、", B未在会引生的, "、", B未在会引生的, "、", 是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是是

序号	MySQL数据库	GaussDB数据库	差异
11	SPACE()	支持	GaussDB入参1最大支持 1073741818,超出返回空字符 串。MySQL的入参1默认最大支持 4194304,超出告警。 函数入参1支持类型存在差异, GaussDB入参1支持类型如下: 整数类型: tinyint、smallint、mediumint、int、bigint 无符号整数类型: tinyint unsigned、smallint unsigned、smallint unsigned、int unsigned 字符和文本类型: char, varchar、tinytext,text, mediumtext, longtext,仅支持纯数字整数字符串,且整数范围在bigint范围内。 浮点类型: float、real、double 定点类型: numeric、decimal、dec 布尔类型: bool
12	SUBSTR()	支持	-
13	SUBSTRING()	支持	-
14	SUBSTRING_IN DEX()	支持	-

序号	MySQL数据库	GaussDB数据库	差异
15	STRCMP()	支持	1. 支持入参类型与M存在差异, GaussDB支持类型如下:
			● 字符类型: CHAR、 VARCHAR、NVARCHAR2、 TEXT
			● 二进制类型: BYTEA
			 数值类型: TINYING [UNSIGNED]、SMALLINT [UNSIGNED]、INTEGER [UNSIGNED]、BIGINT [UNSIGNED]、FLOAT4、 FLOAT8、NUMERIC
			● 日期时间类型: DATE、 TIME WITHOUT TIME ZONE、DATETIME、 TIMESTAMPTZ
			2. 对于数值类型中的浮点类型, 由于连接参数设置不同,精度 可能与M有差异,不建议使用 该场景,或使用NUMERIC类型 代替。
16	SHA() / SHA1()	支持	-
17	SHA2()	支持	-

4.2.3.4 强制转换函数

表 4-13 强制转换函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	CAST()	支持	数据类型转换规则和支持的转换类型均以 GaussDB支持的转换范围和规则为准。
2	CONVERT()	支持	数据类型转换规则和支持的转换类型均以 GaussDB支持的转换范围和规则为准。

4.2.3.5 加密函数

表 4-14 加密函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	AES_DECRYPT()	支持	-
2	AES_ENCRYPT()	支持	-

4.2.3.6 信息函数

表 4-15 信息函数列表

序号	MySQ数据库	GaussDB数据库	差异
1	LAST_INSERT_ID()	支持	-

4.2.3.7 JSON 函数

□ 说明

json函数差异说明:

- 1. 对于json函数和其他字符入参函数来说,如果含有转义字符的输入,默认情况下会与MySQL有一定差异,需要设置GUC参数set standard_conforming_strings= off;的情况下,对于转义字符的场景才会与MySQL兼容,但是会产生非标准字符输入的warning告警,在转义字符中\t、\u与转义数字与MySQL有差异。JSON_UNQUOTE()函数该场景下已做兼容,不设置GUC参数时,仍与MySQL兼容,且不会报警。
- 2. 在处理超长数字(数字的字符长度超过64)时,GaussDB的json函数会将数字解析为一个double处理,并使用科学计数法计数。和MySQL的非json类型入参相同。但是在json类型入参时,由于json类型未完全与MySQL兼容,此场景下会产生差异。MySQL会完整显示数字(并且当数字长度超过82时,MySQL会给出错误的结果。),GaussDB依然将超长数字解析为一个double精度的值。考虑到超长数字内部都是使用浮点数进行储存,进行运算时无论GaussDB还是MySQL都会有精度丢失,建议您使用字符串来储存超长数字。

json_insert

[1, 4, 1e+74, [1, 4, 1e+74]] (1 row)

(110W)

表 4-16 JSON 函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	JSON_APPEND()	支持	-
2	JSON_ARRAY()	支持	-
3	JSON_ARRAY_APPEND()	支持	-

序号	MySQL数据库	GaussDB数据库	差异
4	JSON_ARRAY_INSERT()	支持	-
5	JSON_CONTAINS()	支持	-
6	JSON_CONTAINS_PATH()	支持	-
7	JSON_DEPTH()	支持	-
8	JSON_EXTRACT()	支持	-
9	JSON_INSERT()	支持	-
10	JSON_KEYS()	支持	-
11	JSON_LENGTH()	支持	-
12	JSON_MERGE()	支持	-
13	JSON_OBJECT()	支持	-
14	JSON_QUOTE()	支持	-
15	JSON_REMOVE()	支持	-
16	JSON_REPLACE()	支持	-
17	JSON_SEARCH()	支持	返回值与 MySQL有差 异,GaussDB 返回的是 text,MySQL 返回的是 json。
18	JSON_SET()	支持	-
19	JSON_TYPE()	支持	-
20	JSON_UNQUOTE()	支持	-
21	JSON_VALID()	支持	-

4.2.3.8 聚合函数

表 4-17 聚合函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	GROUP_CONCAT()	支持	当group_concat参数中同时有 DISTINCT和ORDER BY语法时,所 有ORDER BY后的表达式必须也在 DISTINCT的表达式之中。
			group_concat(order by 数字) 不代表按照第几个参数的顺序,数 字只是一个常量表达式,相当于不 排序。
			无论参数的数据类型如何, group_concat返回值的数据类型始 终为text; MySQL的group_concat 在含有二进制类型参数时,返回值 为二进制类型,其他情况返回值为 字符串类型,并且返回值长度大于 512时,其数据类型为字符串大对 象或二进制大对象。
			GUC参数group_concat_max_len 有效范围是0-1073741823,最大 值比MySQL小。
2	DEFAULT()	支持	字段默认值为数组形式, GaussDB返回数组形式, MySQL不支持数组类型。
			GaussDB字段是隐藏列(比如 xmin、cmin),default函数返 回空值。
			● GaussDB支持分区表、临时 表、多表连接查询默认值。
			 GaussDB支持查询列名包含字符串值节点(表示名称)和A_Star节点(表示出现"*"),如default(tt.t4.id)和default(tt.t4.*)。不合法的查询列名和A_Star节点,GaussDB和MySQL报错信息有差异。
			● GaussDB创建字段默认值,没有检验字段类型的范围,使用default函数可能报错。
			字段的默认值是函数表达式 时,GaussDB的default函数返 回建表时字段的default表达式 的计算值。MySQL的default函 数返回NULL。

4.2.3.9 数字操作函数

表 4-18 数字操作函数列表

序号	MySQ L数据 库	Gauss DB数据 库	差异	
1	log2()	支持	 小数位显示与MySQL存在差异,受GaussDB浮点数据 类型限制,可通过参数extra_float_digits控制小数位 个数显示; 	
			● 由于输入精度内部处理差异,GaussDB与MySQL会存在结果计算差异;	
			 支持数据类型有: bigint、int16、int、smallint、tinyint整数类型; bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned无符号整数类型: numeric、real浮点数类型; character、character varying、clob、text字符串类型,仅支持纯数字整数字符串; set类型; NULL空类型。 	
2	log10()	支持	 小数位显示与MySQL存在差异,受GaussDB浮点数据 类型限制,可通过参数extra_float_digits控制小数位 个数显示; 	
			由于输入精度内部处理差异,GaussDB与MySQL会存在结果计算差异;	
			 支持数据类型有: bigint、int16、int、smallint、tinyint整数类型; bigint unsigned、integer unsigned、smallint unsigned、tinyint unsigned无符号整数类型: numeric、real浮点数类型; character、character varying、clob、text字符串类型,仅支持纯数字整数字符串; set类型; NULL空类型。 	

4.2.3.10 其他函数

表 4-19 其他函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	UUID()	支持	-
2	UUID_SHORT()	支持	-

4.2.4 操作符

GaussDB数据库兼容绝大多数MySQL的操作符,但存在部分差异。如未列出,操作符行为默认为GaussDB原生行为。

表 4-20 操作符

序号	MySQL数据库	GaussDB数据库	差异
1	安全等于(<=>)	支持	-

\a', \\\d', \\\e', \\\n', \\\Z', \\\\e'\\sigma \\\equiv \\\e' \\\\n', \\\\\\e'\\\e'\\\\e'\\\\\\\\\\\\\\\\\\	序号	MySQL数据库	GaussDB数据库	差异
时,MySQL存在bug,会报错,后续版本已经修复此问题。 • 空白字符[\t]正则匹配字符类 [:blank:],GaussDB可匹配,MySQL\t不能匹配[:blank:],MySQL存在bug,后续版本已经修复此问题。 • GaussDB支持非贪婪模式匹配,即尽可能少的匹配字符,在部分特殊字符后加'?'问号字符,例如:"?',*?,+?,{n}?,{n,P?,{n,m}?"。MySQ5.7版本不支持非贪婪模式匹配,并报错:Got error 'repetition-operator operan invalid' from regexp。MySQ8.0版本已经支持。 • 在binary字符集下,text类型、blob类型均会转换成bytea类型,由于REGEXP操作符不支持bytea类型,因此无	,,,			 当开启b_format_dev_version='s2' 时,模式字符串pattern中有'\\a', '\\d', '\\e', '\\n', '\\z', '\\u'' 等转义字符串pattern中有'\\a', '\\d', '\\e', '\\n', '\\z', '\\u'' 等转义字符的,匹强深字符串'\a', '\d', '\e', '\n', '\z', '\\u'' 的,还是这个方面,例ySQL5.7不一致。MySQL5.7不一致。MySQL后续版本已经修复与GaussDB一致。 当开启b_format_dev_version='s2' 时,GaussDB '\b' 可以败 看在经修复与GaussDB '\b' 可以败 条件。 模在有单括号')' 匹配 朱人,GaussDB \\b' 可以败 条件。 有在的题。 在delabc匹配序列de或abc的匹配,所有会还是有的证别则,后续下午已经修复此的。 在delabc匹配常子符配,从外SQL存在的明显。 空宫情况可见应配常的,所有的人类。 GaussDB可应是有关键。 GaussDB可应是有关键。 GaussDB可应是有关键。 GaussDB可应是有关键。 GaussDB可应是有关键。 GaussDB可以有关。 有别是有关的。 有别是有关的。 有别是有关的。 在的证别对字符集下,text类型,并接错:Got error 'repetition-operator operand invalid' from regexp。MySQL 8.0版本已经支持。 在的证别对字符集下,text类型、自己的类型均会是类型,由于REGEXP操作符不支持bytea类型,因此无
法匹配。 3 [NOT] RLIKE 支持 同[NOT] REGEXP。	3	[NOT] RLIKE	支持	

4.2.5 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集,支持的范围如下。

表 4-21 字符集列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4	支持
2	gbk	支持
3	gb18030	支持
4	utf8	支持
5	binary	支持

4.2.6 排序规则

GaussDB数据库支持指定库、模式、表或列的排序规则,支持的范围如下。

山 说明

排序规则差异说明:

- 当前仅有字符串类型、部分二进制类型支持指定排序规则,其他类型不支持指定排序规则,可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。
 MySQL中所有类型可以指定字符序,但除字符串、二进制类型其他排序规则无实际意义。
- 当前排序规则(除binary外)仅支持在其对应字符集与库级字符集一致时可以指定, GaussDB数据库中,字符集必须与数据库的字符集一致,且不支持表内多种字符集混合使用。
- utf8mb4字符集下默认字符序为utf8mb4_general_ci,与MySQL5.7保持一致。
- GaussDB中utf8和utf8mb4为同一个字符集。

表 4-22 排序规则列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4_general_ci	支持
2	utf8mb4_unicode_ci	支持
3	utf8mb4_bin	支持
4	gbk_chinese_ci	支持
5	gbk_bin	支持
6	gb18030_chinese_ci	支持
7	gb18030_bin	支持
8	binary	支持

序号	MySQL数据库	GaussDB数据库
9	utf8mb4_0900_ai_ci	支持
10	utf8_general_ci	支持
11	utf8_bin	支持

4.2.7 表达式

GaussDB数据库兼容绝大多数MySQL的表达式,但存在部分差异。如未列出,表达式行为默认为GaussDB原生行为。

表 4-23 表达式

序号	MySQL数据库	GaussDB数据库
1	用户自定义变量@var_name	部分支持
2	全局变量@@var_name	部分支持

4.2.8 SQL

GaussDB数据库兼容绝大多数MySQL语法,但存在部分差异。本章节介绍GaussDB数据库当前支持的MySQL语法。

4.2.8.1 DDL

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
1	建表和修改表时支持创建主键、 UNIQUE索引、外键约束	ALTER TABLE CREATE TABLE	 GaussDB当前不支持 UNIQUE INDEX KEY index_name语法,使 用UNIQUE INDEX KEY index_name语法时会 报错。 当约束被建立为全局二级索引,SQL语句中指定using btree时,底层会建立为ubtree。 当约束关联的表为ustore,且SQL语句中指定为using btree时,底层会建立为ubtree。

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
2	支持自增列	ALTER TABLE CREATE TABLE	• 自动增长列建议为索引 (非全局二级索引)的 第一个字段,否则建表 时产生警告,含有自动 增长列的表进行某些操 作时会产生错误,例 如: ALTERT TABLE EXCHANGE PARTITION。MySQL自 动增长列必须为索引第 一个字段。
			● AUTO_INCREMENT = value语法,value必须 为小于2^127的正数。 MySQL不校验value。
			 当自增值已经达到字段数据类型的最大值时,继续自增将产生错误。 MySQL有些场景产生错误或警告,有些场景仍自增为最大值。
			● 不支持 innodb_autoinc_lock_ mode系统变量, GaussDB的GUC参数 auto_increment_cache =0时,批量插入自动增长列的行为与MySQL系统变量 innodb_autoinc_lock_ mode=1相似。
			● 自动增长列在导入数据 或者进行Batch Insert 执行计划的插入操作 时,对于混合0、NULL 和确定值的场景,如果 产生错误,后续插入自 增值不一定与MySQL完 全一致。
			- 提供 auto_increment_ca che参数,可以控制 预留自增值的数 量。
			并行导入或插入自动增长列触发自增时,每个并行线程预留的缓存值

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
			也只在其线程中使用, 未完全使用完毕的话, 也会出现表中自动增长 列的值不连续的情况。 并行插入产生的自增值 结果无法保证与MySQL 完全一致。
			长列批量插入时不会预留自增值,正常场景不会产生不连续的自增值。MySQL临时表与普通表中的自动增长列自增结果一致。
			SERIAL数据类型为原有的自增列,与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。
			 不允许 auto_increment_offset 的值大于 auto_increment_incre ment的值,会产生错 误。MySQL允许,并说 明 auto_increment_offset
			会被忽略。 • 在表有主键或索引的情况下,ALTER TABLE命令重写表数据的顺序与MySQL不一定相同,GaussDB按照表数据存储顺序重写,MySQL会按主键或索引顺序重写,导致自增值的顺序可能不同。
			ALTER TABLE命令添加或修改自增列时,第一次预留自增值的数量是表统计信息中的行数,统计信息的行数不一定与MySQL一致。 last_insert_id函数返回
			值为128位的整型。

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
			在触发器或用户自定义 函数中自增时,刷新 last_insert_id返回值。 MySQL不刷新。
			 对GUC参数 auto_increment_offset 和 auto_increment_incre ment设置超出范围的 值会产生错误。MySQL 会自动改为边界值。
			• sql_mode设置 no_auto_value_on_zer o参数,表定义的自动 增长列为非NOT NULL 约束,向表中插入数据 不指定自动增长列的值 时,GaussDB中自动增长列插入NULL值,且 不触发自增; MySQL中自动增长列插入NULL值,触发自增。

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
3	支持前缀索引	CREATE INDEX, ALTER TABLE, CREATE TABLE	 前缀长度不得超过 2676,健值的实力有多的。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个有引力长度。 使用的一个点面。 使用的一个点面。
4	支持指定字符集与 排序规则	ALTER SCHEMA、 ALTER TABLE、 CREATE SCHEMA、 CREATE TABLE	-
5	修改表时支持在表 第一列前面或者在 指定列后面添加列	ALTER TABLE	-
6	修改列名称/定义语 法兼容	ALTER TABLE	-
7	定时任务EVENT语 法兼容	ALTER EVENT、 CREATE EVENT、 DROP EVENT、 SHOW EVENTS	-

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
8	创建分区表语法兼 容	CREATE TABLE PARTITION、CREATE TABLE SUBPARTITION	-
9	建表和修改表时支 持指定表级和列级 comment	CREATE TABLE \ ALTER TABLE	-
10	创建索引时支持指 定索引级comment	CREATE INDEX	-

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
11	交换普通表和分区 表分区的数据	ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点:
			对于自增列,MySQL执 行alter exchange partition后,自增列会 被重置;GaussDB 则 不会被重置,自增列则 按照旧的自增值递增。
			MySQL表或分区使用 tablespace时,则无法 进行分区和普通表数据 的交换;GaussDB表或 分区使用不同的 tablespace时,仍可进 行分区和普通表数据的 交换。
			• 对于列默认值,MySQL 不会校验默认值,因此 默认值不同时也可进行 分区和普通表数据的交 换;GaussDB会校验默 认值,如果默认值不 同,则无法进行分区和 普通表数据的交换。
			MySQL在分区表或普通 表上进行DROP列操作 后,表结构仍然一致, 则可进行分区和普通表 数据的交换;GaussDB 需要保证普通表和分区 表的被删除列严格对齐 才能进行分区和普通表 数据的交换。
			MySQL和GaussDB的哈 希算法不同,所以两者 在相同的hash分区存储 的数据可能不一致,导 致最后交换的数据也可 能不一致。
			MySQL的分区表不支持外键,普通表包含外键或其他表引用普通表的外键,则无法进行分区和普通表数据的交换;GaussDB的分区表支持外键,在两个表的外键约束一致时,则可进行

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
			分区和普通表数据的交换,GaussDB的分区表积,GaussDB的分区表不带外键,普通表有其他表引用,如果分区表和普通表表一致,则可进行分区和普通表数据的交换。
12	支持删除表的主键 外键约束	ALTER TABLE DROP [PRIMARY FOREIGN]KEY	-

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
13	支持CREATE TABLE LIKE语法 兼容	CREATE TABLE LIKE	Round Barrier Round B

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
14	支持更改表名兼容语法	ALTER TABLE tbl_name RENAME [TO AS =] new_tbl_name; RENAME {TABLE TABLES} tbl_name TO new_tbl_name [, tbl_name2 TO new_tbl_name2,];	 GaussDB的alter rename语法仅支持修改表名称功能操作,不能耦合其它功能操作。 GaussDB仅旧表名字段支持如schema.table_name用法;且新是名与旧类名与旧类。 GaussDB不支持新旧,所以其前Schema重和权限,所以为的表面的形式的形式的形式的形式的形式的形式的形式的形式的形式的形式的形式的形式的形式的

序号	MySQL数据库功能 概述	详细语法说明	GaussDB数据库实现差异
15	支持増加子分区语法兼容	ALTER TABLE [IF EXISTS] { table_name [*] ONLY table_name ONLY (table_name)} action [,]; action: move_clause exchange_clause row_clause modify_clause split_clause add_clause drop_clause ilm_clause add_clause: ADD {{partition_less_than_i} tem partition_start_end_ite m partition_less_than_i less_than_item partition_start_end_ite m partition_start_end_ite m partition_start_end_ite m partition_start_end_ite m partition_list_item})}	● 不支持如下语法添加多分区: ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition1, partition_definition1, partition_definition1); ● 仅支持原有添加多分区语法: ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2[y1]), ···;

4.2.8.2 DML

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
1	DELETE支持从多个表中 删除数据	DELETE	-
2	DELETE支持ORDER BY 和LIMIT	DELETE	-
3	DELETE支持从指定分区 (或子分区)删除数据	DELETE	-

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
4	UPDATE支持从多个表中 更新数据	UPDATE	-
5	UPDATE支持ORDER BY 和LIMIT	UPDATE	-
6	SELECT INTO语法兼容	SELECT	GaussDB可以使用SELECT INTO根据查询结果创建一个新表,MySQL不支持。
			● GaussDB的SELECT INTO语法 不支持将多个查询进行集合运 算后的结果作为查询结果。

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
7	REPLACE INTO语法兼容	REPLACE	● 时间类型初始值的差异。例如: - MySQL不受严格模式和宽松模式的影响,可向表中插入时间0值,即: mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, f2, f3); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; +

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
			- MySQLBIT类型的初始值为空串",即: mysql> CREATE TABLE test(f1 BIT(3) NOT NULL); Query OK, 0 rows affected (0.01 sec) mysql> REPLACE INTO test VALUES(f1); Query OK, 1 row affected (0.00 sec) mysql> SELECT f1, f1 IS NULL FROM test; +++ f1 f1 is null +++ 2 rows in set (0.00 sec) - GaussDB位串类型BIT的初始值为NULL,则报错。gaussdb=# CREATE TABLE test(f1 BIT(3) NOT NULL); CREATE TABLE gaussdb=# REPLACE INTO test VALUES(f1); ERROR: null value in column "f1" violates not-null constraint
8	SELECT支持指定多分区 查询	SELECT	DETAIL: Failing row contains (null).
9	UPDATE支持指定多分区 更新	UPDATE	-

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
10	LOAD DATA导入数据功能	LOAD DATA	 LOAD DATA语法执行结果与 MySQL严格模式一致,宽松模 式暂未适配。
			IGNORE与LOCAL参数功能仅为 当导入数据与表中数据存在冲 突时,忽略当前冲突行数据功 能和当文件中字段数小于指定 表中列数时自动为其余列填充 默认值功能,其余功能暂未适 配。
			• 指定LOCAL关键字,且文件路 径为相对路径时,文件从二进 制目录下搜索;不指定LOCAL 关键字,且文件路径为相对路 径时,文件从数据目录下搜 索。
			语法中指定分隔符,转义字符,分行符等符号时,若指定为单引号,将导致词法解析错误。
			● [(col_name_or_user_var [, col_name_or_user_var])]指 定列参数不支持重复指定列。
			● [FIELDS TERMINATED BY 'string']指定换行符不能与 [LINES TERMINATED BY 'string']分隔符相同。
			执行LOAD DATA语法写入表中 的数据若无法转换为表中数据 类型格式时报错。
			● LOAD DATA SET表达式中不支 持指定列名计算。
			若set表达式返回值类型与对应 列类型之间不存在隐式转换函 数则报错。
			● LOAD DATA只能用于表,不能 用于视图。
			 windows下的文件与linux环境 下文件默认换行符存在差异, LOAD DATA无法识别此场景会 报错,建议用户导入时检查导 入文件行尾换行符。

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
11	INSERT IGNORE兼容	INSERT	● GaussDB会返回降级后的错误信息,MySQL则会将降级后的错误信息记录到错误堆栈中,然后调用show warnings;命令查看。 ● 时间类型的差异。例如: - GaussDB中date、datetime、timestamp默认零值。gaussdb=# CREATE TABLE test(f1 DATE NOT NULL, f2 DATETIME NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL, NULL, NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1

序号	MySQL数据库功能概述	详细语法说明	GaussDB数据库实现差异
			Warning 1048 Column 'f1' cannot be null Warning 1048 Column 'f2' cannot be null Warning 1048 Column 'f3' cannot be null ++

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
序号	MySQL数据库功能概述		度,GaussDB则不显示,例如: GaussDB指定时间精度 gaussdb=# CREATE TABLE test(f1 TIME(3) NOT NULL, f2 DATETIME(3) NOT NULL, f3 TIMESTAMP(3) NOT NULL); CREATE TABLE gaussdb=# INSERT IGNORE INTO test VALUES(NULL,NULL), WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null, null). WARNING: null value in column "f3" violates not-null constraint DETAIL: Failing row contains (null, null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2 f3
			● 由于MySQL数据库和GaussDB 执行过程的差异,因此,产生

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
			的warnings条数可能不同,例 如:
			– GaussDB产生的warnings条 数
			gaussdb=# CREATE TABLE test(f1 INT, f2 INT not null); CREATE TABLE gaussdb=# INSERT INTO test VALUES(1,0),(3,0),(5,0); INSERT 0 3 gaussdb=# INSERT IGNORE INTO test SELECT f1+1, f1/f2 FROM test; WARNING: division by zero CONTEXT: referenced column: f2 WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (2, null). WARNING: division by zero CONTEXT: referenced column: f2 WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (4, null). WARNING: division by zero CONTEXT: referenced column: f2 WARNING: division by zero CONTEXT: referenced column: f2 WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (6, null). INSERT 0 3
			- MySQL产生的warnings条数 mysql> CREATE TABLE test(f1 INT, f2 INT not null); Query OK, 0 rows affected (0.01 sec) mysql> INSERT INTO test
			VÁLÚES(1,0),(3,0),(5,0); Query OK, 3 rows affected (0.00 sec) Records: 3 Duplicates: 0 Warnings: 0
			mysql> INSERT IGNORE INTO test SELECT f1+1, f1/f2 FROM test; Query OK, 3 rows affected, 4 warnings (0.00 sec) Records: 3 Duplicates: 0 Warnings: 4
			MySQL数据库和GaussDB INSERT IGNORE在触发器中的 差异,例如:
			– GaussDB触发器中使用 INSERT IGNORE gaussdb=# CREATE TABLE test1(f1 INT NOT NULL); CREATE TABLE gaussdb=# CREATE TABLE test2(f1

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
			INT); CREATE TABLE gaussdb=# CREATE OR REPLACE FUNCTION trig_test() RETURNS TRIGGER AS \$\$ gaussdb\$# BEGIN gaussdb\$# INSERT IGNORE INTO test1 VALUES(NULL); gaussdb\$# ETURN NEW; gaussdb\$# END; gaussdb\$# END; gaussdb\$# \$\$ LANGUAGE plpgsql; CREATE FUNCTION gaussdb=# CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW EXECUTE PROCEDURE trig_test(); CREATE TRIGGER gaussdb=# INSERT INTO test2 VALUES(NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null). CONTEXT: SQL statement "INSERT IGNORE INTO test1 VALUES(NULL)" PL/pgSQL function trig_test() line 3 at SQL statement INSERT 0 1 gaussdb=# SELECT * FROM test1; f1 0 (1 rows) MySQL触发器中使用INSERT IGNORE mysql> CREATE TABLE test1(f1 INT NOT NULL); Query OK, 0 rows affected (0.01 sec) mysql> CREATE TABLE test2(f1 INT); Query OK, 0 rows affected (0.00 sec) mysql> CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW -> BEGIN -> INSERT IGNORE into test1 values(NULL); -> END Query OK, 0 rows affected (0.01 sec) mysql> DELIMITER mysql> CREATE TRIGGER trig2 BEFORE INSERT ON test2 FOR EACH ROW -> BEGIN -> INSERT IGNORE into test1 values(NULL); -> END Query OK, 0 rows affected (0.01 sec) mysql> DELIMITER;

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
序号	MySQL数据库功能概述		mysql> INSERT INTO test2 VALUES(NULL); ERROR 1048 (23000): Column 'f1' cannot be null mysql> INSERT IGNORE INTO test2 VALUES(NULL); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test1; ++ f1 ++ 1 row in set (0.00 sec) squssDB的bool、serial的实现机制与MySQL不同,因此其默认零值与MySQL不同,因此其默认零值与MySQL不同,例如: - GaussDB的行为 gaussdb=# CREATE TABLE test(f1 SERIAL, f2 BOOL NOT NULL); NOTICE: CREATE TABLE will create implicit sequence "test_f1_seq" for serial column "test.f1" CREATE TABLE gaussdb=# INSERT IGNORE INTO test values(NULL,NULL); WARNING: null value in column "f1" violates not-null constraint DETAIL: Failing row contains (null, null). WARNING: null value in column "f2" violates not-null constraint DETAIL: Failing row contains (null, null). INSERT 0 1 gaussdb=# SELECT * FROM test; f1 f2
			values(NULL,NULL); Query OK, 1 row affected, 1 warning (0.00 sec) mysql> SELECT * FROM test;

序号	MySQL数据库功能概述	详细语法说 明	GaussDB数据库实现差异
			++ f1 f2 ++ 1 0 ++ 1 row in set (0.00 sec)

4.2.8.3 DCL

序号	概述	详细语法说明	差异
1	支持SET用户自定义 变量	SET	● 自定义变量长度的差异。例如: - MySQL自定义变量名长度没有约束。 - GaussDB自定义变量名长度不超过64字节,超过部分的变量名会截断并提示告警。
2	SET TRANSACTION 语法兼容	SET TRANSACTION	MySQL可以设置当前会话(session)和全局(global)的事务隔离级别和读写模式,GaussDB设置当前会话需要设置参数 b_format_behavior_compat_options包含set_session_tranasction,设置全局只对当前数据库生效。
3	SET NAMES指定 COLLATE字句	SET [SESSION LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] DEFAULT};	GaussDB暂不支持指定 charset_name与数据库字 符集不同。具体请参考 《开发指南》中"SQL参 考 > SQL语法 > S > SET"章节。

4.2.9 驱动

4.2.9.1 JDBC

4.2.9.1.1 JDBC 接口参考

获取结果集中的数据

ResultSet对象提供了丰富的方法,以获取结果集中的数据。获取数据常用的方法如表1 所示,其他方法请参考JDK官方文档。

表 4-24 ResultSet 对象的常用方法

方法	描述	差异
int getInt(int columnIndex)	按列标获取 int型数据。	-
int getInt(String columnLabel)	按列名获取 int型数据。	-
String getString(int columnIndex)	按列标获取 String型数 据。	字段类型为整型且带有ZEROFILL属性时, GaussDB按照ZEROFILL属性要求的宽度信息用0 进行补位后输出结果,MySQL直接输出结果。
String getString(String columnLabel)	按列名获取 String型数 据。	字段类型为整型且带有ZEROFILL属性时, GaussDB按照ZEROFILL属性要求的宽度信息用0 进行补位后输出结果,MySQL直接输出结果。
Date getDate(int columnIndex)	按列标获取 Date型数 据。	-
Date getDate(String columnLabel)	按列名获取 Date型数 据。	-

4.3 MySQL 兼容性 M-Compatibility 模式

4.3.1 MySQL 兼容性 M-Compatibility 模式概述

本章节主要介绍GaussDB数据库的MySQL兼容性M-Compatibility模式(即 sql_compatibility='M')与MySQL5.7数据库的兼容性对比信息。仅介绍505.1版本后新增的兼容性特性,特性的相关规格和约束建议在开发指南中查看。

GaussDB数据库在数据类型、SQL功能和数据库对象等基本功能上与MySQL数据库兼容。

GaussDB的执行计划和优化、explain显示结果与MySQL不同。

由于GaussDB数据库与MySQL数据库底层框架实现存在差异,GaussDB数据库与MySQL数据库仍存在部分差异。

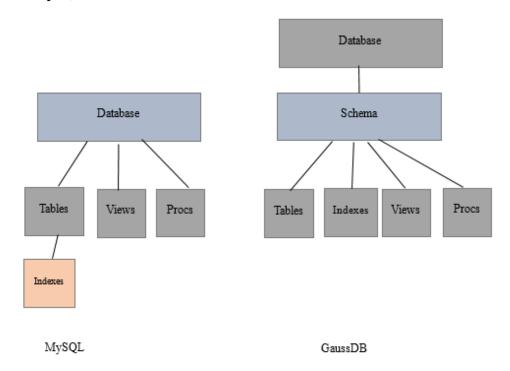
□ 说明

由于M-Compatibility的底层架构与MySQL存在差异,对于information_schema和m_schema下与MySQL名称相同的Schema(具体参见M-Compatibility开发指南第六章),其查询性能可能存在差异。例如count函数无法做执行上的优化,表现为select *和select count(*)语句耗时近似。

Database 和 Schema 设计

MySQL的数据对象包括DATABASE、TABLE、INDEX、VIEW、TRIGGER、PROC等,MySQL的对象层次跟GaussDB的对应关系是从上至下且一对多包含关系。如下图所示:

图 4-1 MySQL 和 GaussDB 中 Database 和 Schema 之间的差异



- 在MySQL中Database和Schema是同义词;而在GaussDB中,一个Database下可以有多个Schema。在该特性中,每个MySQL中的Database都被映射到GaussDB的一个Schema。
- 在MySQL中,INDEX从属于一个TABLE,但在GaussDB中,INDEX从属于一个Schema。这个差异导致INDEX名在GaussDB中要求在Schema内唯一,但在MySQL中仅要在在一个表内唯一。这个差异将作为当前约束予以保留。

4.3.2 数据类型

GaussDB的数据类型的大部分功能场景与MySQL一致,但存在部分差异。

- 除特别说明,部分数据类型精度、标度、位数大小等不支持用浮点型数值定义, 建议使用合法的整型数值定义。
- GaussDB回显以\0结束, MySQL会把整个字符串显示, 因此GaussDB会将被解析 为\0后的字节截断, MySQL不会。
 示例:

```
-- GaussDB场景
m_db=# SELECT FORMAT(1000, 4, 'bg_BG');
format
m_db=# SELECT CONCAT('123', b'00000000', 'aa');
concat
123
(1 row)
-- Mysql场景
mysql> SELECT FORMAT(1000, 4, 'bg_BG');
| FORMAT(1000, 4, 'bg_BG') |
+-----+
| 1 000,0000 |
1 row in set (0.01 sec)
mysql> SELECT CONCAT('123', b'00000000', 'aa');
| CONCAT('123', b'00000000', 'aa') |
| 123 aa |
1 row in set (0.00 sec)
```

4.3.2.1 数值数据类型

表 4-25 整数类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BOOL	支持,存在 差异	输出格式:GaussDB中SELECT TRUE/FALSE 输出结果为t/f,MySQL为1/0。
2	BOOLEAN	支持,存在 差异	MySQL: BOOL/BOOLEAN类型实际映射为TINYINT类型。
3	TINYINT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
4	SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。

序号	MySQL数据 库	GaussDB数 据库	差异
5	MEDIUMIN T[(M)] [UNSIGNED] [ZEROFILL]	支持,存在	MySQL存储MEDIUMINT数据需要3字节。
6	INT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
7	INTEGER[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。
8	BIGINT[(M)] [UNSIGNED] [ZEROFILL]	支持,存在 差异	详细请参见说明。

山 说明

输入格式:

MySQL:

整型类型对于类似"1.2.3.4.5"有多个小数点的字符串形式输入,在宽松模式下MySQL会发生错误解析,抛出WARNING并插表成功,例如将"1.2.3.4.5"插入表后值为12。

GaussDB:

整型类型对于类似"1.2.3.4.5"有多个小数点的的字符串形式输入,在宽松模式下,会将第二个小数点后的字符当作非法字符全部截断,抛出WARNING并插表成功,例如将"1.2.3.4.5"插入表后值为1,"1.6.3.4.5"插入表后值为2。

示例:在UNION的CREATE TABLE AS场景中,GaussDB对于整形的CONST节点,取该整形的默认max_length,MySQL根据实际值计算max_length。

表 4-26 任意精度类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DECIMAL[(M[,D])] [ZEROFILL]	支持,存在 差异	MySQL decimal用一个9*9的数组存储数值,整数部分和小数部分分开存储,超过该长度时优先截小数部分。GaussDB只会在整数位数超
2	NUMERIC[(M[,D])] [ZEROFILL]	支持,存在 差异	过81位时截断。
3	DEC[(M[,D])] [ZEROFILL]	支持,存在 差异	
4	FIXED[(M[,D])] [ZEROFILL]	支持,存在 差异	

表 4-27 浮点类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	FLOAT[(M,D)] [ZEROFILL]	支持,存在 差异	FLOAT数据类型不支持KEY键值分区策略分区 表。
2	FLOAT(p) [ZEROFILL]	支持,存在 差异	FLOAT数据类型不支持KEY键值分区策略分区 表。
3	DOUBLE[(M ,D)] [ZEROFILL]	支持,存在 差异	DOUBLE数据类型不支持KEY键值分区策略分区表。
4	DOUBLE PRECISION[(M,D)] [ZEROFILL]	支持,存在 差异	DOUBLE PRECISION数据类型不支持KEY键值分区策略分区表。

序号	MySQL数据 库	GaussDB数 据库	差异
5	REAL[(M,D)] [ZEROFILL]	支持,存在 差异	REAL数据类型不支持KEY值分区策略分区表。

🗀 说明

驱动中FLOAT类型和DOUBLE类型带精度标度场景,数据输入超范围不支持报错。

4.3.2.2 日期与时间数据类型

表 4-28 日期与时间数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	DATE	支持,存在 差异。	GaussDB支持date数据类型,与MySQL相比 规格上存在如下差异:
			● 输入格式:
			- 分隔符:反斜杠\在MySQL和GaussDB 中都视为转义,但MySQL支持\0, GaussDB暂不支持,因此反斜杠作为分 隔符且分隔符后为字符0时GaussDB会 报错。
2	DATETIME[(f sp)]	支持,存在 差异。	GaussDB支持datetime数据类型,与MySQL 相比规格上存在如下差异:
			輸入格式:
			- 分隔符:反斜杠\在MySQL和GaussDB 中都视为转义,但MySQL支持\0, GaussDB暂不支持,因此反斜杠作为分 隔符且分隔符后为字符0时GaussDB会 报错。

序号	MySQL数据 库	GaussDB数 据库	差异
3	TIMESTAMP[(fsp)]	支持,存在差异。	GaussDB支持timestamp数据类型,与MySQL相比规格上存在如下差异: • 输入格式: - 分隔符: 反斜杠\在MySQL和GaussDB中都视为转义,但MySQL支持\0,GaussDB暂不支持,因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 • 默认值: - MySQL 5.7中timestamp列默认有default value,为数据插入时的实时时间。GaussDB与MySQL 8.0一致,均不设置默认值,即插入null时,值为null。
4	TIME[(fsp)]	支持,存在 差异。	GaussDB支持time数据类型,与MySQL相比规格上存在如下差异: • 输入格式: - 分隔符:反斜杠\在MySQL和GaussDB中都视为转义,但MySQL支持\0,GaussDB暂不支持,因此反斜杠作为分隔符且分隔符后为字符0时GaussDB会报错。 • 当时间类型的时、分、秒、纳秒为0时,GaussDB和MySQL可能存在符号位不同的情况。

序号	MySQL数据 库	GaussDB数 据库	差异
5	YEAR[(4)]	支持。	GaussDB支持year数据类型,与MySQL相比 规格上存在如下差异:
			● 创建year类型的字段,和mysql显示有所不同,GaussDB显示year,Mysql显示 year(4):
			GaussDB场景 m_db=# create table t_year (c_year year); CREATE TABLE m_db=# desc t_year; Field Type Null Key Default Extra
			Tield Type Null Rey Default Extra ++++
			1 row in set (0.00 sec) mysql> create table t1 as(select * from t_year); Query OK, 0 rows affected (0.02 sec) Records: 0 Duplicates: 0 Warnings: 0 mysql> desc t1; ++++ Field Type Null Key Default Extra ++++ c_year year(4) YES NULL 1 row in set (0.00 sec)

山 说明

• GaussDB不支持ODBC语法的字面量:

{ d 'str' } { t 'str' } { ts 'str' }

• GaussDB支持标准SQL字面量,且类型关键字后面可选择添加精度,MySQL不支持:

DATE[(n)] 'str'

TIME[(n)] 'str'

TIMESTAMP[(n)] 'str'

 当指定DATETIME、TIME、TIMESTAMP数据类型的精度超过其支持的最大精度时, GaussDB会将精度截断成支持的最大精度,MySQL则会报错。

4.3.2.3 字符串数据类型

表 4-29 字符串数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	CHAR(M)	支持,存在 差异	● 输入格式: - 输入二进制或十六进制字符串, GaussDB输出为十六进制,MySQL中 根据ASCII码表转义,无法转义的输出 为空。
2	VARCHAR(M)	支持,存在差异	 输入格式: GaussDB的自定义函数参数和返回值不支持长度校验,存储过程参数不支持长度校验,MySQL支持。 GaussDB的自定义函数和存储过程中的临时变量支持长度校验以及严格宽松模式下的报错和截断告警,MySQL不支持。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。
3	TINYTEXT	支持,存在差异	 输入格式: 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中TINYTEXT类型不支持主键,GaussDB支持。 索引: MySQL中TINYTEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。

序号	MySQL数据 库	GaussDB数 据库	差异
4	TEXT	支持,存在 差异	 输入格式: 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中TEXT类型不支持主键,GaussDB支持。 索引: MySQL中TEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。
5	MEDIUMTEX T	支持,存在 差异	 输入格式: 默认值: 创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中MEDIUMTEXT类型不支持主键,GaussDB支持。 索引: MySQL中MEDIUMTEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGTEXT	支持,存在 差异	 输入格式: GaussDB只支持不超过1G字节长度,MySQL支持4G-1字节长度。 默认值:创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中LONGTEXT类型不支持主键,GaussDB支持。 索引: MySQL中LONGTEXT类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。

4.3.2.4 二进制数据类型

表 4-30 二进制数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	BINARY[(M)]	支持,存在差异	● 输入格式: - 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 - 插入字符串长度小于目标长度时,GaussDB填充符是0x20,MySQL是0x00。 - 字符集: 默认字符集为数据库初始化字符集,MySQL默认类型字符集为BINARY字符集。 - 输出格式: - JDBC协议输出时BINARY类型的末尾空格显示为空格,MySQL末尾空格显示为\x00。 - 宽松模式下,BINARY类型面对输入超过n的字节数的字符输入(例如中文字符),会将超限的整个字符截断。MySQL中会将超限的整个字符的前几位满足n范围内的字节信息保留,但输出时字符信息显示乱码。 - 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。 说明 GaussDB中,由于BINARY类型填充符和\0截断与MySQL的差异,在操作符比较计算,字符串相关系统函数计算,索引匹配,数据导入导出等场景下与MySQL的表现会存在差异。差异场景请查看本节示例。

序号	MySQL数据 库	GaussDB数 据库	差异
2	VARBINARY(M)	支持,存在差异	 輸入格式: 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 字符集:默认字符集为数据库初始化字符集,MySQL默认类型字符集为BINARY字符集。 输出格式: JDBC协议输出时BINARY类型的末尾空格显示为空格,MySQL末尾空格显示为\x00。 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。
3	TINYBLOB	支持,存在差异	 输入格式: 器认值: 创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中TINYBLOB类型不支持主键,GaussDB支持。 索引: MySQL中TINYBLOB类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。 输出格式: 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。

序号	MySQL数据 库	GaussDB数 据库	差异
4	BLOB	支持,存在 差异	 输入格式: 默认值: 创建表列时语法上允许设置默认值。 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中BLOB类型不支持主键,GaussDB支持。 索引: MySQL中BLOB类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支
			持。 ● 输出格式:在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。
5	MEDIUMBLO B	支持,存在差异	 输入格式: 二 默认值: 创建表列时语法上允许设置默认值。 一 输入二进制或十六进制字符串,GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。 主键: MySQL中MEDIUMBLOB类型不支持主键,GaussDB支持。 索引: MySQL中MEDIUMBLOB类型不支持除前缀索引外其他索引方法,GaussDB支持。 外键: MySQL中TINYTEXT类型不支持作为外键的参考列/被参考列,GaussDB支持。 输出格式: 在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。

序号	MySQL数据 库	GaussDB数 据库	差异
6	LONGBLOB	支持,存在 差异	● 取值范围:只支持不超过1G字节长度, MySQL支持4G-1字节长度。
			● 输入格式:
			– 默认值:创建表列时语法上允许设置默 认值,MySQL不允许设置默认值。
			- 输入二进制或十六进制字符串, GaussDB输出为十六进制,MySQL中 根据ASCII码表转义,无法转义的输出 为空。
			● 主键:MySQL中LONGBLOB类型不支持 主键,GaussDB支持。
			 索引: MySQL中LONGBLOB类型不支持 除前缀索引外其他索引方法, GaussDB支 持。
			外键: MySQL中TINYTEXT类型不支持作 为外键的参考列/被参考列,GaussDB支 持。
			• 输出格式:在MySQL 8.0以上版本,默认以0x开头形式返回,GaussDB以多个\x形式返回。
7	BIT[(M)]	支持,存在	输出格式:
		差异	所有输出按照二进制字符串形式输出。 MySQL中根据ASCII码表转义,无法转义 的输出为空。
			在MySQL 8.0以上版本,默认会开头补0,GaussDB不会补0。

```
-- GaussDB场景
m_db=# CREATE TABLE test(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test VALUES(0x8000);
INSERT 0 1
m_db=# SELECT hex(a) FROM test;
   hex
80202020202020202020
a
(0 rows)
m_db=# DROP TABLE test;
DROP TABLE
m_db=# CREATE TABLE test2(a BINARY(10)) DISTRIBUTE BY REPLICATION;
CREATE TABLE
m_db=# INSERT INTO test2 VALUES(0x80008000);
```

```
INSERT 0 1
m_db=# SELECT hex(a) FROM test2;
80202020202020202020
(1 row)
m_db=# DROP TABLE test2;
DROP TABLE
-- Mysql场景
mysql> CREATE TABLE test(a BINARY(10));
Query OK, 0 rows affected (0.01 sec)
mysql> INSERT INTO test VALUES(0x8000);
Query OK, 1 row affected (0.00 sec)
mysql> SELECT hex(a) FROM test;
| hex(a)
             1 row in set (0.00 sec)
l a
     1 🚟
      +----+
1 row in set (0.00 sec)
mysql> DROP TABLE test;
Query OK, 0 rows affected (0.00 sec)
mysql> CREATE TABLE test2(a BINARY(10));
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO test2 VALUES(0x80008000);
Query OK, 1 row affected (0.00 sec)
mysql> SELECT hex(a) FROM test2;
| hex(a)
1 row in set (0.00 sec)
mysql> DROP TABLE test2;
Query OK, 0 rows affected (0.00 sec)
```

4.3.2.5 JSON 类型

表 4-31 JSON 数据类型

序号	MySQL数据 库	GaussDB数 据库	差异
1	JSON	支持,存在 差异。	GaussDB支持JSON数据类型与MySQL相比,规格存在如下差异: • 取值范围: 在MySQL中,JSON数据类型的最大长度为4GB,但在GaussDB中,JSON数据类型的最大长度小于1GB,对象的键值对个数最大值和数组的元素个数最大值也小于MySQL。 • 字符序差异: 在MySQL中,使用collation函数单独查询JSON类型的列,返回的字符序是BINARY,但GaussDB中返回utf8mb4_bin。其他使用的场景都使用utf8mb4_bin,与MySQL相同。

4.3.2.6 数据类型支持的属性

表 4-32 数据类型支持的属性

序号	MySQL数据库	GaussDB数据库		
1	NULL	支持		
2	NOT NULL	支持		
3	DEFAULT	支持		
4	ON UPDATE	支持		
4	PRIMARY KEY	支持		
5	AUTO_INCREMENT	支持		
6	CHARACTER SET name	支持		
7	COLLATE name	支持		
8	ZEROFILL	支持		

使用CREATE TABLE AS方式建表,对VARBINARY类型的字段设置默认值,在使用SHOW CREATE TABLE、DESC或\d 查询的时候回显与MySQL存在差异,GaussDB显示为转换成十六进制后的值,而MySQL显示为原值。

```
m_db=# CREATE TABLE test_int(
    int_col INT
m db=# CREATE TABLE test varbinary(
     varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
m_db=# SHOW CREATE TABLE test_varbinary;
   Table
                                Create Table
test_varbinary | SET search_path = public;
          | CREATE TABLE test varbinary (
             varbinary_col varbinary(20) DEFAULT X'6761757373',
             int col integer
          CHARACTER SET = "UTF8" COLLATE = "utf8mb4_general_ci"
          | WITH (orientation=row, compression=no, storage_type=USTORE, segment=off);
m_db=# DROP TABLE test_int, test_varbinary;
mysql> CREATE TABLE test_int(
     int_col INT
mysql> CREATE TABLE test_varbinary(
     varbinary_col VARBINARY(20) default 'gauss'
) AS SELECT * FROM test_int;
mysql> SHOW CREATE TABLE test_varbinary;
| Table
          | Create
Table
| test_varbinary | CREATE TABLE `test_varbinary` (
 `varbinary_col` varbinary(20) DEFAULT 'gauss',
 int vol int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |
1 row in set (0.00 sec)
mysql> DROP TABLE test_int, test_varbinary;
```

4.3.2.7 数据类型转换

不同的数据类型之间支持转换。有如下场景涉及到数据类型转换:

- 操作符(比较操作符、运算操作符等)的操作数的数据类型不一致。常见于查询 条件或者关联条件中的比较运算。
- 函数调用时实参和形参的数据类型不一致。
- DML语句要更新(包括INSERT、UPDATE、MERGE、REPLACE等)的目标列,数据的类型和列的定义类型不一致。
- 显式的类型转换: CAST(expr AS datatype),将expr表达式类型转换为datatype 类型。
- 集合运算(UNION、MINUS、EXCEPT、INTERSECT)确定最终投影列的目标数据类型后,各个SELECT查询的投影列的类型和目标数据类型不一致。
- 其他表达式计算场景,根据不同表达式的数据类型,来决定用于比较或者最终结果的目标数据类型。
- 普通的字符串类型当字符序为BINARY时,将转换成对应的二进制类型(TEXT转换成BLOB,VARCHAR转换成VARBINARY等)。

数据类型转换差异点主要分为: 隐式转换, 显式转换、UNION/CASE、decimal类型。

隐式类型转换差异点

- GaussDB中统一平铺成小类型到小类型的转换规则,MySQL中使用小类型转大类型,大类型转小类型的转换规则。
- GaussDB中隐式转换因数据类型本身差异点,输出格式存在部分行为不一致。 示例:

● GaussDB中的隐式转换,BIT数据类型到字符数据类型和二进制数据类型转换,输出存在部分行为不一致。GaussDB输出为十六进制,MySQL中根据ASCII码表转义,无法转义的输出为空。

```
m_db=# CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS_COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS_COL6 BIT(4),
  VS_COL7 BIT(4),
  VS_COL8 BIT(4)
) DISTRIBUTE BY REPLICATION;
m_db=# CREATE TABLE string_storage (
  VS_COL1 BLOB,
  VS_COL2 TINYBLOB,
  VS_COL3 MEDIUMBLOB,
  VS_COL4 LONGBLOB,
  VS_COL5 TEXT,
  VS_COL6 TINYTEXT,
  VS_COL7 MEDIUMTEXT,
  VS COL8 LONGTEXT
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
m_db=# INSERT INTO string_storage SELECT * FROM bit_storage;
m_db=# SELECT * FROM string_storage;
VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8
\x05 |\x05 |\x05 |\x05 |\x05 |\x05 |\x05
(1 row)
m_db=# DROP TABLE bit_storage, string_storage;
mysql> CREATE TABLE bit_storage (
  VS COL1 BIT(4),
  VS_COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
```

```
VS_COL6 BIT(4),
  VS_COL7 BIT(4),
  VS_COL8 BIT(4)
):
mysql> CREATE TABLE bit_storage (
  VS_COL1 BIT(4),
  VS_COL2 BIT(4),
  VS_COL3 BIT(4),
  VS_COL4 BIT(4),
  VS_COL5 BIT(4),
  VS COL6 BIT(4),
  VS_COL7 BIT(4),
  VS_COL8 BIT(4)
);
mysql> INSERT INTO bit_storage VALUES(B'101', B'101', B'101', B'101', B'101', B'101', B'101', B'101');
mysql> INSERT INTO string_storage SELECT * FROM bit_storage;
mysql> SELECT * FROM string_storage;
| VS_COL1 | VS_COL2 | VS_COL3 | VS_COL4 | VS_COL5 | VS_COL6 | VS_COL7 | VS_COL8 |
                        1 row in set (0.00 sec)
mysql> DROP TABLE bit_storage, string_storage;
```

中间带有0x00的二进制或十六进制字符串插入二进制数据类型时,GaussDB部分插入,0x00后的字符会截断,MySQL可以完整插入。

示例:

```
m_db=# CREATE TABLE blob_storage (
     A BLOB
) DISTRIBUTE BY REPLICATION;
m_db=# INSERT INTO blob_storage VALUES (0xBB00BB);
m_db=# SELECT hex(A) FROM blob_storage;
hex
ВВ
(1 row)
m_db=# DROP TABLE blob_storage;
mysql> CREATE TABLE blob_storage (
     A BLOB
mysql> INSERT INTO blob_storage VALUES (0xBB00BB);
mysql> SELECT hex(A) FROM blob_storage;
| hex(a) |
| BB00BB |
1 row in set (0.01 sec)
mysql> DROP TABLE blob_storage;
```

中间带有0x00的二进制或十六进制字符串插入字符串数据类型时,GaussDB部分插入,0x00后的字符会截断,MySQL严格模式不允许插入,宽松模式插入空字符串。

```
A TEXT
);
mysql> INSERT INTO text_storage VALUES (b'10111011000000010111011');
ERROR 1366 (HY000): Incorrect string value: '\xBB\x00\xBB' for column 'A' at row 1
mysql> SELECT hex(A) FROM text_storage;
Empty set (0.00 sec)
mysql> SET SQL_MODE=";
mysql> INSERT INTO text_storage VALUES (b'101110110000000010111011');
mysql> SELECT hex(A) FROM text_storage;
+------+
| hex(A) |
+------+
| row in set (0.01 sec)
mysql> DROP TABLE text_storage;
```

 WHERE子句中只带有普通字符串, GaussDB中't'、'true'、'yes'、 'y'、'on'返回 TRUE, 'no'、'f'、'off'、'false'、'n'返回FALSE, 其余字符串报错。MySQL通过字符串转换为INT1判断返回TRUE/FALSE。

示例:

● 对于YEAR类型的输入,在将字符串转换为整型的过程中,MySQL考虑科学计数法,GaussDB暂不支持,统一做截断处理。

```
m_db=# CREATE TABLE test_year (
    A YEAR
m_db=# SET sql_mode = ";
m db=# INSERT INTO test year VALUES ('2E3x');
WARNING: Data truncated for column.
LINE 1: insert into t1 values ('2E3x');
CONTEXT: referenced column: a
m_db=# SELECT * FROM test_year ORDER BY A;
а
2002
(1 row)
m_db=# DROP TABLE test_year;
mysql> CREATE TABLE test_year (
    A YEAR
mysql> INSERT INTO test_year VALUES ('2E3x');
mysql> SELECT * FROM test_year ORDER BY A;
```

```
+----+
|a |
+-----+
| 2000 |
+-----+
1 row in set (0.01 sec)
mysql> DROP TABLE test_year;
```

 对于UNION的CREATE TABLE AS场景, GaussDB不区分左右子节点的顺序, MySQL区分左右子节点的顺序,左右子节点互换会导致结果不同。

```
m_db=# CREATE TABLE test2(
  F1 FLOAT,
  11 TINYINT,
  12 SMALLINT,
  DTT1 DATETIME(6),
  DEC3 DECIMAL(32, 15),
  JS1 JSON,
  D2 DOUBLE,
  CH1 CHAR(255),
  D3 DOUBLE,
  TX1 TINYTEXT
m_db=# CREATE TABLE test1 SELECT DISTINCT concat((F1 + I1 - DTT1) * DEC3 % D2 / CH1) a from
test2 UNION ALL SELECT sqrt((DEC3 + DTT1 - JS1) * D3 / - TX1 % I2) FROM test2;
m_db=# DESC test1;
Field | Type | Null | Key | Default | Extra
-----+-----+-----+-----+------
a | text | YES | | |
(1 row)
m_db=# CREATE TABLE test3 SELECT DISTINCT sqrt((DEC3 + DTT1 - JS1) * D3 / - TX1 % I2) a from
test2 UNION ALL SELECT concat((F1 + I1 - DTT1) * DEC3 % D2 / CH1) FROM test2;
m db=# DESC test3;
Field | Type | Null | Key | Default | Extra
a | text | YES | | |
(1 row)
m_db=# DROP TABLE test1, test2, test3;
mysql> CREATE TABLE test2(
 F1 FLOAT,
  I1 TINYINT.
  12 SMALLINT,
  DTT1 DATETIME(6),
  DEC3 DECIMAL(32, 15),
  JS1 JSON,
  D2 DOUBLE
  CH1 CHAR(255),
  D3 DOUBLE.
  TX1 TINYTEXT
mysql> CREATE TABLE test1 SELECT DISTINCT concat((F1 + I1 - DTT1) * DEC3 % D2 / CH1) a from
test2 UNION ALL SELECT sqrt((DEC3 + DTT1 - JS1) * D3 / - TX1 % I2) FROM test2;
mysql> DESC test1;
     --+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----
| a | varchar(53) | YES | | NULL | |
     --+-----+
1 row in set (0.00 sec)
mysql> CREATE TABLE test3 SELECT DISTINCT sqrt((DEC3 + DTT1 - JS1) * D3 / - TX1 % I2) a from
test2 UNION ALL SELECT concat((F1 + I1 - DTT1) * DEC3 % D2 / CH1) FROM test2;
mysql> DESC test3;
| a | varchar(23) | YES | | NULL | |
```

```
+-----+
1 row in set (0.00 sec)
mysql> DROP TABLE test1, test2, test3;
```

GaussDB中函数嵌套场景下,涉及到聚合函数(如max、min、sum和avg)中存在字符串类型包含非数值字符,隐式转换到数值类型发生截断或置零,且包含操作符比较、having比较的场景时,GaussDB统一进行类型转换并产生告警,MySQL在相同场景下不会全部产生告警。

```
m_db=# SET m_format_behavior_compat_options= 'enable_precision_decimal';
m_db=# SELECT max(c4) <> 0 FROM ((SELECT 2.22 id, '2006-04-27 20:19:02.132' c4)) tb_1;
WARNING: Truncated incorrect double value: '2006-04-27 20:19:02.132'
?column?
t
(1 row)
m_db=# SELECT sum(c4) <> 0 FROM ((SELECT 2.22 id, '2006-04-27 20:19:02.132' c4)) tb_1;
WARNING: Truncated incorrect double value: '2006-04-27 20:19:02.132'
?column?
(1 row)
m db=# SELECT (SELECT max(c4) f5 FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4) UNION
all (SELECT 2.22 id, '1985-09-01 07:59:59' c4)) tb_1
m db(# WHERE exists (SELECT max(c4) FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4)
UNION all (SELECT 2.22 id, '1985-09-01 07:59:59' c4)) tb_2)
m_db(# GROUP BY id WITH rollup HAVING f5<>0 LIMIT 0,1) + INTERVAL '33.22'
SECOND MICROSECOND col5;
WARNING: Truncated incorrect double value: '2006-04-27 20:19:08.132'
CONTEXT: referenced column: col5
       col5
2006-04-27 20:19:41.352000
(1 row)
m_db=# SELECT (SELECT sum(c4) f5 FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4) UNION
all (SELECT 2.22 id, '1985-09-01 07:59:59' c4)) tb_1
m_db(# WHERE exists (SELECT sum(c4) FROM ((select 2.22 id, '2006-04-27 20:19:08.132' c4) UNION
all (select 2.22 id, '1985-09-01 07:59:59' c4)) tb_2)
m_db(# GROUP BY id WITH rollup HAVING f5<>0 LIMIT 0,1) + INTERVAL '33.22'
SECOND_MICROSECOND col5;
WARNING: Truncated incorrect double value: '2006-04-27 20:19:08.132'
CONTEXT: referenced column: col5
WARNING: Truncated incorrect double value: '1985-09-01 07:59:59'
CONTEXT: referenced column: col5
WARNING: Truncated incorrect double value: '2006-04-27 20:19:08.132'
CONTEXT: referenced column: col5
WARNING: Truncated incorrect double value: '2006-04-27 20:19:08.132'
CONTEXT: referenced column: col5
WARNING: Truncated incorrect double value: '1985-09-01 07:59:59'
CONTEXT: referenced column: col5
WARNING: Truncated incorrect double value: '1985-09-01 07:59:59'
CONTEXT: referenced column: col5
WARNING: Incorrect datetime value: '3991'
CONTEXT: referenced column: col5
col5
-----
(1 row)
mysql> SELECT max(c4) <> 0 FROM ((SELECT 2.22 id, '2006-04-27 20:19:02.132' c4)) tb_1;
| \max(c4) <> 0 |
      1 |
```

```
1 row in set (0.00 sec)
mysql> SELECT sum(c4) <> 0 FROM ((SELECT 2.22 id, '2006-04-27 20:19:02.132' c4)) tb 1;
| sum(c4) <> 0 |
+----+
  1 |
1 row in set, 1 warning (0.00 sec)
mysql> SHOW warnings;
| Level | Code | Message |
| Warning | 1292 | Truncated incorrect DOUBLE value: '2006-04-27 20:19:02.132' |
1 row in set (0.00 sec)
mysql> SELECT (SELECT max(c4) f5 FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4) UNION all
(select 2.22 id, '1985-09-01 07:59:59' c4)) tb_1
  -> WHERE exists (SELECT max(c4) FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4) UNION
all (SELECT 2.22 id, '1985-09-01 07:59:59' c4)) tb_2)
 -> GROUP BY id WITH rollup HAVING f5<>0 limit 0,1) + INTERVAL '33.22'
SECOND_MICROSECOND col5;
col5
| 2006-04-27 20:19:41.352000 |
1 row in set (0.00 sec)
mysql> SELECT (SELECT sum(c4) f5 FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4) UNION all
(SELECT 2.22 id, '1985-09-01 07:59:59' c4)) tb_1
  -> WHERE exists (SELECT sum(c4) FROM ((SELECT 2.22 id, '2006-04-27 20:19:08.132' c4) UNION
all (select 2.22 id, '1985-09-01 07:59:59' c4)) tb_2)
  -> GROUP BY id WITH rollup HAVING f5<>0 LIMIT 0,1) + INTERVAL '33.22'
SECOND_MICROSECOND col5;
| col5 |
| NULL |
1 row in set, 7 warnings (0.01 sec)
mysql> SHOW warnings;
+-----+
| Level | Code | Message |
| Warning | 1292 | Truncated incorrect DOUBLE value: '2006-04-27 20:19:08.132' |
| Warning | 1292 | Truncated incorrect DOUBLE value: '1985-09-01 07:59:59'
Warning | 1292 | Truncated incorrect DOUBLE value: '2006-04-27 20:19:08.132' |
Warning | 1292 | Truncated incorrect DOUBLE value: '2006-04-27 20:19:08.132' |
| Warning | 1292 | Truncated incorrect DOUBLE value: '1985-09-01 07:59:59'
Warning | 1292 | Truncated incorrect DOUBLE value: '1985-09-01 07:59:59'
| Warning | 1292 | Incorrect datetime value: '3991'
<del>|-----</del>
7 rows in set (0.00 sec)
```

显式类型转换差异点

 GaussDB中平铺成对各目标类型的转换规则, MySQL中使用C++多态重载函数, 在嵌套场景中存在不一致行为。

```
示例:
```

```
m_db=# SELECT CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED);
WARNING: Truncated incorrect INTEGER value: '2023-01-01'
CONTEXT: referenced column: cast
cast
```

```
2023
(1 row)

mysql> SELECT CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED);
+-----+
| CAST(GREATEST(date'2023-01-01','2023-01-01') AS SIGNED) |
+-----+
| 20230101 |
+-------+
```

● 在GaussDB中,BLOB、TINYBLOB、MEDIUMBLOB、LONGBLOB、BINARY、VARBINARY、BIT、及YEAR类型显式转换为JSON类型,结果与MySQL不同。

m_db=# CREATE TABLE test_blob (c1 BLOB, c2 TINYBLOB, c3 MEDIUMBLOB, c4 LONGBLOB, c5 BINARY(32), c6 VARBINARY(100), c7 BIT(64), c8 YEAR); m_db=# INSERT INTO test_blob VALUES('[1, "json"]', 'true', 'abc', '{"jsnid": 1, "tag": "ab"}', '[1, "json"]', '{"jsnid": 1, "tag": "ab"}', '20', '2020'); INSERT 0 1 m_db=# SELECT CAST(c1 AS JSON), CAST(c2 AS JSON), CAST(c3 AS JSON), CAST(c4 AS JSON), CAST(c5 AS JSON), CAST(c6 AS JSON), CAST(c7 AS JSON), CAST(c8 AS JSON) FROM test_blob; CAST | CAST | CAST | CAST | CAST CAST | CAST | CAST '[1, \"json\"]" | "true" | "abc" | "{\"jsnid\": 1, \"tag\": \"ab\"}" | "[1, \"json\"] " | "{\"jsnid \": 1, \"tag\": \"ab\"}" | "20" | "2020" (1 row) mysql> CREATE TABLE test_blob (c1 BLOB, c2 TINYBLOB, c3 MEDIUMBLOB, c4 LONGBLOB, c5 BINARY(32), c6 VARBINARY(100), c7 BIT(64), c8 YEAR); Query OK, 0 rows affected (0.02 sec) mysql> INSERT INTO test_blob VALUES('[1, "json"]', 'true', 'abc', '{"jsnid": 1, "tag": "ab"}', '[1, "json"]', '{"jsnid": 1, "tag": "ab"}', '20', '2020'); Query OK, 1 row affected (0.00 sec) mysql> SELECT CAST(c1 AS JSON), CAST(c2 AS JSON), CAST(c3 AS JSON), CAST(c4 AS JSON), CAST(c5 AS JSON), CAST(c6 AS JSON), CAST(c7 AS JSON), CAST(c8 AS JSON) FROM test_blob; | CAST(c6 AS "base64:type252:WzEsICJqc29uIl0=" | "base64:type249:dHJ1ZQ==" | "base64:type250:YWJj" | 'base64:type251:eyJqc25pZCl6IDEslCJ0YWciOiAiYWlifQ==" | "base64:type254:WzEsICJqc29uIl0AAAAAAAAAAAAAAAAAAAAAAAAAAAA" | "base64:type15:eyJqc25pZCI6IDEsICJ0YWciOiAiYWIifQ==" | "base64:type16:AAAAAAAAMjA=" | "base64:type13:MjAyMA==" |

● GaussDB在JSON数据类型显式转换后运用于精度计算时,与MySQL5.7不一致,与MySQL8.0一致,计算时精度与使用JSON类型表中数据精度保持一致。

示例:

1 row in set (0.00 sec)

```
test=# drop table tt01;
DROP TABLE
test=# create table tt01 as select -cast('98.7654321' as json) as c1;
INSERT 0 1
```

```
test=# desc tt01;
Field | Type | Null | Key | Default | Extra
c1 | double | YES | | |
(1 row)
test=# select * from tt01;
c1
-98.7654321
(1 row)
mysql> select version();
version()
| 5.7.44-debug-log |
1 row in set (0.00 sec)
mysql> drop table tt01;
Query OK, 0 rows affected (0.02 sec)
mysql> create table tt01 as select -cast('98.7654321' as json) as c1;
Query OK, 1 row affected (0.03 sec)
Records: 1 Duplicates: 0 Warnings: 0
mysql> desc tt01;
| Field | Type | Null | Key | Default | Extra |
+----+
| c1 | double(17,0) | YES | | NULL | |
+----+
1 row in set (0.00 sec)
mysql> select * from tt01;
| c1 |
| -99 |
1 row in set (0.00 sec)
mysql> select version();
version()
| 8.0.36-debug |
1 row in set (0.00 sec)
mysql> drop table tt01;
Query OK, 0 rows affected (0.05 sec)
mysql> create table tt01 as select -cast('98.7654321' as json) as c1;
Query OK, 1 row affected (0.12 sec)
Records: 1 Duplicates: 0 Warnings: 0
mysql> desc tt01;
+----+
| Field | Type | Null | Key | Default | Extra |
+----+
c1 | double | YES | NULL | |
+----+
1 row in set (0.01 sec)
mysql> select * from tt01;
| c1 |
```

UNION, CASE 和相关构造差异点

- POLYGON + NULL、POINT + NULL、POLYGON + POINT组合在MySQL中均返回GEOMETRY类型,GaussDB中未涉及,暂时当做报错处理。
- SET和ENUM两种类型暂未支持,暂时当做报错处理。
- JSON和二进制类型(BINARY、VARBINARY、TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB)的UNION和UNION ALL组合,MySQL中返回LONGBLOB类型,GaussDB中返回JSON类型,同时支持二进制类型(BINARY、VARBINARY、TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB)到JSON的隐式类型转换。
- 未设置m_format_behavior_compat_options为enable_precision_decimal时,常量类型和其他类型做类型聚合的时候,输出类型的精度为其他类型的精度。如 "select "helloworld" union select p from t;"的结果的精度为属性p的精度。
- 未设置m_format_behavior_compat_options为enable_precision_decimal时,定点常量和不带精度约束的类型(非字符串类型如int、bool、year等,聚合结果类型为定点类型)聚合时,精度约束会按照定点数默认精度31输出。
- merge rule差异:
 - MySQL 5.7中存在部分不合理的类型推导,如BIT类型和整型/YEAR类型推导会得出VARBINARY类型,UNSIGNED类型和非UNSIGNED类型推导会得到带UNSIGNED的类型等,同时CASE WHEN和UNION的聚合结果也存在差异,类型推导结果太小时存在数据溢出风险。在MySQL 8.0版本修复了上述相关的问题,因此merge rule聚合规则以8.0为准。
- MySQL中BINARY和CHAR填充字符不相同,BINARY填充'\0',CHAR填充空格,GaussDB中BINARY和CHAR都是填充空格。
- 在精度传递场景下,使用CASE WHEN语句时,会进行类型转换和精度重新计算, 导致最终的输出结果与CASE子句对比会出现末尾多零场景或末尾少零场景:
 - 末尾多零场景: CASE节点会根据CASE子句的精度计算CASE节点精度,当 THEN子句的精度比CASE节点的精度小时,会在CASE节点末尾补零。
 - 末尾少零场景:多层CASE WHEN嵌套时,内层CASE执行类型转换之后,只保留内层CASE的精度,外层CASE无法得到THEN子句的精度信息,因此外层CASE会根据内层CASE的精度计算的精度进行类型转换。当外层CASE类型转换时内层CASE精度比THEN子句少,会出现末尾少零场景。

示例:

- 末尾多零少零场景。

```
-- 末尾多零场景。
m_db=# SELECT 15.6 AS result;
result
------
15.6
(1 row)

m_db=# SELECT CASE WHEN 1 < 2 THEN 15.6 ELSE 23.578 END AS result;
result
------
15.600
(1 row)

m_db=# SELECT greatest(12, 3.4, 15.6) AS result;
result
```

```
15.6
(1 row)
m_db=# SELECT CASE WHEN 1 < 2 THEN greatest(12, 3.4, 15.6) ELSE greatest(123.4, 23.578,
36) END AS result;
result
15.600
(1 row)
-- 末尾少零场景。
m_db=# create table t1 as select (false/-timestamp '2008-12-31 23:59:59.678') as result;
INSERT 0 1
m_db=# desc t1;
Field | Type | Null | Key | Default | Extra
result | double(8,7) | YES | |
(1 row)
m_db=# select (false/-timestamp '2008-12-31 23:59:59.678') as result;
 result
-0.0000000
(1 row)
m_db=# create table t1 as select (case when 1<2 then false/-timestamp '2008-12-31
23:59:59.678' else 0016.11e3/'22.2' end) as result;
INSERT 0.1
m db=# desc t1;
Field | Type | Null | Key | Default | Extra
result | double | YES | | |
(1 row)
m_db=# select (case when 1<2 then false/-timestamp '2008-12-31 23:59:59.678' else
0016.11e3/'22.2' end) as result;
result
  -0
(1 row)
m_db=# drop table t1;
DROP TABLE
m_db=# create table t1 as select (case when 1+1=2 then case when 1<2 then false/-timestamp
'2008-12-31 23:59:59.678' else 0016.11e3/'22.2' end else 'test' end) as result;
INSERT 0 1
m_db=# desc t1;
Field | Type | Null | Key | Default | Extra
result | varchar(23) | YES | |
(1 row)
m db=# select (case when 1+1=2 then case when 1<2 then false/-timestamp '2008-12-31
23:59:59.678' else 0016.11e3/'22.2' end else 'test' end) as result;
result
-0
```

- 在开启精度传递的场景下,使用集合运算(UNION、MINUS、EXCEPT、INTERSECT),如果参与集合运算的查询语句,其查询的字段为函数、表达式而不是直接使用表中的字段,且查询的结果数据类型为INT/INTUNSIGNED,则最后返回的数据类型存在差异。在MySQL中,返回的数据类型为BIGINT/BIGINT UNSIGNED;在GaussDB中,返回的数据类型为INT/INT UNSIGNED。

```
-- GaussDB执行结果。
m_db=# SET
```

```
m_format_behavior_compat_options='select_column_name,enable_precision_decimal';
SFT
m_db=# DROP TABLE IF EXISTS t1,t2,ctas1,ctas2;
DROP TABLE
m_db=# CREATE TABLE t1(a INT, b INT);
CREATE TABLE
m_db=# CREATE TABLE t2(c INT UNSIGNED, d INT UNSIGNED);
CREATE TABLE
m_db=# CREATE TABLE ctas1 AS (SELECT a, ABS(a) FROM t1) UNION (SELECT b, ABS(b) FROM
t1);
INSERT 0 0
m_db=# DESC ctas1;
Field | Type | Null | Key | Default | Extra
a | integer(11) | YES | |
ABS(a) | integer(11) | YES | |
(2 rows)
m_db=# CREATE TABLE ctas2 AS (SELECT c, ABS(c) FROM t2) UNION (SELECT d, ABS(d) FROM
t2);
INSERT 0 0
m_db=# DESC ctas2;
                    | Null | Key | Default | Extra
Field | Type
c | integer(11) unsigned | YES | |
ABS(c) | integer(11) unsigned | YES | |
(2 rows)
m_db=# DROP TABLE IF EXISTS t1,t2,ctas1,ctas2;
DROP TABLE
-- MySQL执行结果。
mysql> DROP TABLE IF EXISTS t1,t2,ctas1,ctas2;
Query OK, 0 rows affected, 4 warnings (0.00 sec)
mysql> CREATE TABLE t1(a INT, b INT);
Query OK, 0 rows affected (0.05 sec)
mysql> CREATE TABLE t2(c INT UNSIGNED, d INT UNSIGNED);
Query OK, 0 rows affected (0.03 sec)
mysql> CREATE TABLE ctas1 AS (SELECT a, ABS(a) FROM t1) UNION (SELECT b, ABS(b) FROM
t1);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC ctas1;
               .--+----+----+
| Field | Type | Null | Key | Default | Extra |
+----+
  |int(11) | YES | | NULL | |
| ABS(a) | bigint(20) | YES | | NULL |
2 rows in set (0.01 sec)
mysql> CREATE TABLE ctas2 AS (SELECT c, ABS(c) FROM t2) UNION (SELECT d, ABS(d) FROM
t2);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC ctas2;
+----+
| Field | Type
                | Null | Key | Default | Extra |
           -----+
c | int(11) unsigned | YES | NULL |
                                          | ABS(c) | bigint(20) unsigned | YES | | NULL | |
2 rows in set (0.00 sec)
```

mysql> DROP TABLE IF EXISTS t1,t2,ctas1,ctas2; Query OK, 0 rows affected (0.07 sec)

在开启精度传递的场景下, case when被嵌套场景的结果与MySQL保持差异。MySQL的类型可以透过多层直接转换, 而GaussDB结果精度是逐层确定并且逐层转换的, 因此可能导致结果小数位或进位和MySQL不一致。

- 对于需要int类型的运算符(如 ~, &, |, <<, >>) 嵌套case when语句,若case when语句返回的是varchar类型,则实际情况可以会发生截断(根据原表数据分析是否会发生截断),GaussDB会报出相应错误(select查询warning告警,create建表error报错),Mysql不会报错。(若GaussDB想要完成createtable建表操作,可以通过设置sql_mode关闭严格模式)

```
-- GaussDB:
m_db=# create table t_base (num_var numeric(20, 10), time_var time(6));
CREATE TABLE
m_db=# insert into t_base values ('-2514.1441000000','12:10:10.125000'),('-417.2147000000','
11:30:25.258000');
INSERT 0 2
m_db=# select (~(case when false then time var else num var end)) as res2 from t_base;
WARNING: Truncated incorrect INTEGER value: '-2514.1441000000'
CONTEXT: referenced column: res2
WARNING: Truncated incorrect INTEGER value: '-417.2147000000'
CONTEXT: referenced column: res2
res2
2513
416
m db=# create table t1 as select (~(case when false then time var else num var end)) as res2
from t_base;
ERROR: Truncated incorrect INTEGER value: '-2514.1441000000'
CONTEXT: referenced column: res2
m_db=# set sql_mode="";
SFT
m_db=# create table t1 as select (~(case when false then time var else num var end)) as res2
from t base:
WARNING: Truncated incorrect INTEGER value: '-2514.1441000000'
CONTEXT: referenced column: res2
WARNING: Truncated incorrect INTEGER value: '-417.2147000000'
CONTEXT: referenced column: res2
INSERT 0 2
m_db=# desc t1;
Field |
                     | Null | Key | Default | Extra
res2 | bigint(21) unsigned | YES | |
(1 row)
mysql> create table t_base (num_var numeric(20, 10), time_var time(6));
```

```
Query OK, 0 rows affected (0.01 sec)
mysql> insert into t_base values ('-2514.1441000000','12:10:10.125000'),('-417.2147000000','
11:30:25.258000');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> select (~(case when false then time_var else num_var end)) as res2 from t_base;
res2
| 2513 |
| 416 |
2 rows in set (0.00 sec)
mysql> create table t1 as select (~(case when false then time_var else num_var end)) as res2
from t_base;
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> desc t1;
| Field | Type | Null | Key | Default | Extra |
          .----+---
| res2 | bigint(21) unsigned | YES | NULL |
+-----+
1 row in set (0.00 sec)
```

在开启精度传递的场景下,对于CREATE VIEW AS SELECT CASE WHEN语句和SELECT CASE WHEN语句嵌套常量(包括常量计算、函数嵌套常量等)的情况,GaussDB在该情况下值保持一致,MySQL在SELECT CASE WHEN语句中可能会丢失部分精度。

```
-- GaussDB:
m db=# CREATE OR REPLACE VIEW test view AS
m_db-# SELECT (CASE WHEN 1<2 THEN 3.33/4.46 ELSE 003.3630/002.2600 END) c1,(CASE
WHEN 1>2 THEN IFNULL(null,3.363/2.2) ELSE NULLIF(3.33/4.46,3.363/2.2) END) c2;
CREATE VIEW
m_db=# SELECT * FROM test_view;
 c1 | c2
0.74663677 | 0.7466368
(1 row)
m_db=# SELECT (CASE WHEN 1<2 THEN 3.33/4.46 ELSE 003.3630/002.2600 END) c1,(CASE
WHEN 1>2 THEN IFNULL(null,3.363/2.2) ELSE NULLIF(3.33/4.46,3.363/2.2) END) c2;
c1 | c2
0.74663677 | 0.7466368
(1 row)
-- MySQL:
mysql> CREATE OR REPLACE VIEW test_view AS
  -> SELECT (CASE WHEN 1<2 THEN 3.33/4.46 ELSE 003.3630/002.2600 END) c1,(CASE WHEN
1>2 THEN IFNULL(null,3.363/2.2) ELSE NULLIF(3.33/4.46,3.363/2.2) END) c2;
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT * FROM test_view;
+----+
| c1 | c2 |
+----+
| 0.74663677 | 0.7466368 |
1 row in set (0.00 sec)
mysql> SELECT (CASE WHEN 1<2 THEN 3.33/4.46 ELSE 003.3630/002.2600 END) c1,(CASE
WHEN 1>2 THEN IFNULL(null,3.363/2.2) ELSE NULLIF(3.33/4.46,3.363/2.2) END) c2;
| c1 | c2 |
     ----+-----+
| 0.746637 | 0.746637 |
+----+
1 row in set (0.00 sec)
```

- 在开启精度传递的场景下,M-Compatibility模式数据库支持UNION/CASE WHEN语句建表,但是由于架构不同,M-Compatibility模式数据库无法保证 创建的表的所有类型与MySQL 8.0完全相同。MySQL返回字符串、二进制相 关类型的场景,以及部分函数嵌套场景,与GaussDB存在不一致。

```
m_db=# CREATE TABLE IF NOT EXISTS testcase (id int, col_text1 tinytext, col_text2 text,
col_blob1 tinyblob, col_blob2 blob, col_blob3 mediumblob, col_blob4 longblob);
CREATE TABLE
m_db=# CREATE TABLE t1 AS SELECT id,(CASE WHEN id=2 THEN col_text1 ELSE 'test' END)
f35, (CASE WHEN id=2 THEN col_text2 ELSE 'test' END) f36,(CASE WHEN id=2 THEN col_blob1
ELSE 'test' END) f41, (CASE WHEN id=2 THEN col_blob2 ELSE 'test' END) f42, (CASE WHEN
id=2 THEN col blob3 ELSE 'test' END) f43, (CASE WHEN id=2 THEN col blob4 ELSE 'test' END)
f44 FROM testcase:
INSERT 0 0
m_db=# DESC t1;
Field | Type | Null | Key | Default | Extra
id | integer(11) | YES | |
f35 | varchar(255) | YES | |
f36 | mediumtext | YES | |
f41 | varbinary(255) | YES | |
f42 | blob | YES |
                                f43 | mediumblob | YES | |
f44 | longblob
                | YES | |
(7 rows)
m_db=# CREATE TABLE IF NOT EXISTS testtext1 (col10 text);
CREATE TABLE
m_db=# CREATE TABLE IF NOT EXISTS testtext2 (col10 text);
CREATE TABLE
m_db=# CREATE TABLE testtext AS (select * from testtext1) UNION (select * from testtext2);
CREATE TABLE
m_db=# desc testtext;
m_db=#
Field | Type | Null | Key | Default | Extra
col10 | text | YES | |
(1 row)
m_db=# create table testchar as select (select lcase(-6873.4354)) a, (select
sec_to_time(-485769.567)) b union all select (select bin(-58768923.21321)), (select
asin(-0.7237465));
INSERT 02
m db=# desc testchar:
Field | Type | Null | Key | Default | Extra
     text | YES | |
b | varchar(23) | YES | |
(2 rows)
m_db=# CREATE TABLE test_func (col_text char(29));
CREATE TABLE
m_db=# CREATE TABLE test1 AS SELECT * FROM ( SELECT
     GREATEST(2.22, col_text) f1, LEAST(2.22, col_text) f2,
     ADDDATE(col_text, INTERVAL '1.28.16.31' HOUR_MICROSECOND) f3,
     SUBDATE(col_text, INTERVAL '39.49.15' MINUTE_MICROSECOND) f4,
     DATE_SUB(col_text, INTERVAL '45' MICROSECOND) f5,
     DATE ADD(col text, INTERVAL '12.00.00.00.001' DAY MICROSECOND) f6,
     ADDTIME(col_text, '8:20:20.3554') f7,
     SUBTIME(col_text, '8:20:20.3554') f8 from test_func) t1
UNION ALL
     SELECT * FROM ( SELECT
     GREATEST(2.22, col text) f1, LEAST(2.22, col text) f2.
     ADDDATE(col_text, INTERVAL '1.28.16.31' HOUR_MICROSECOND) f3,
     SUBDATE(col_text, INTERVAL '39.49.15' MINUTE_MICROSECOND) f4,
     DATE_SUB(col_text, INTERVAL '45' MICROSECOND) f5,
     DATE_ADD(col_text, INTERVAL '12.00.00.00.001' DAY_MICROSECOND) f6,
     ADDTIME(col_text, '8:20:20.3554') f7,
```

```
SUBTIME(col_text, '8:20:20.3554') f8 from test_func) t2;
INSERT 0 0
m_db=# DESC test1;
Field | Type | Null | Key | Default | Extra
f1 | double | YES | |
f2 | double | YES | |
f3 | varchar(29) | YES |
    | varchar(29) | YES | |
f4 | varchar(29) | YES |
f5 | varchar(29) | YES | |
f6
f7
    | varchar(29) | YES |
| varchar(29) | YES |
f8 | varchar(29) | YES |
(8 rows)
-- MySQL:
mysql> CREATE TABLE IF NOT EXISTS testcase (id int, col_text1 tinytext, col_text2 text,
col_blob1 tinyblob, col_blob2 blob, col_blob3 mediumblob, col_blob4 longblob);
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE TABLE t1 AS SELECT id, (CASE WHEN id=2 THEN col_text1 ELSE 'test' END) f35,
(CASE WHEN id=2 THEN col_text2 ELSE 'test' END) f36,(CASE WHEN id=2 THEN col_blob1 ELSE
'test' END) f41, (CASE WHEN id=2 THEN col_blob2 ELSE 'test' END) f42, (CASE WHEN id=2
THEN col_blob3 else 'test' END) f43, (CASE WHEN id=2 THEN col_blob4 ELSE 'test' END) f44
FROM testcase;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC t1;
| Field | Type | Null | Key | Default | Extra |
+----+
|id |int |YES| |NULL |
| f35 | longtext | YES | NULL |
| f36 | longtext | YES | NULL |
| f44 | longblob | YES | | NULL |
7 rows in set (0.00 sec)
mysql> CREATE TABLE IF NOT EXISTS testtext1 (col10 text);
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE TABLE IF NOT EXISTS testtext2 (col10 text);
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE testtext AS (select * from testtext1) UNION (select * from testtext2);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc testtext;
| Field | Type | Null | Key | Default | Extra |
+----+
| col10 | mediumtext | YES | | NULL | |
+----+
1 row in set (0.00 sec)
mysql> set sql_mode=";
Query OK, 0 rows affected, 1 warning (0.01 sec)
mysgl> create table testchar as select (select lcase(-6873.4354)) a, (select
sec_to_time(-485769.567)) b union all select (select bin(-58768923.21321)), (select
asin(-0.7237465));
Query OK, 2 rows affected, 1 warning (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 1
mysql> desc testchar;
```

```
| Field | Type | Null | Key | Default | Extra |
          -----+----+----+----+----
    2 rows in set (0.00 sec)
mysql> CREATE TABLE test_func (col_text char(29));
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE test1 AS SELECT * FROM ( SELECT
  -> GREATEST(2.22, col_text) f1, LEAST(2.22, col_text) f2,
  -> ADDDATE(col_text, INTERVAL '1.28.16.31' HOUR_MICROSECOND) f3,
  -> SUBDATE(col_text, INTERVAL '39.49.15' MINUTE_MICROSECOND) f4,
  -> DATE_SUB(col_text, INTERVAL '45' MICROSECOND) f5,
  -> DATE_ADD(col_text, INTERVAL '12.00.00.00.001' DAY_MICROSECOND) f6,
  -> ADDTIME(col_text, '8:20:20.3554') f7,
  -> SUBTIME(col_text, '8:20:20.3554') f8 from test_func) t1
  -> UNION ALL
  -> SELECT * FROM ( SELECT
  -> GREATEST(2.22, col_text) f1, LEAST(2.22, col_text) f2,
  -> ADDDATE(col_text, INTERVAL '1.28.16.31' HOUR_MICROSECOND) f3,
  -> SUBDATE(col_text, INTERVAL '39.49.15' MINUTE_MICROSECOND) f4,
  -> DATE_SUB(col_text, INTERVAL '45' MICROSECOND) f5,
  -> DATE_ADD(col_text, INTERVAL '12.00.00.00.001' DAY_MICROSECOND) f6,
  -> ADDTIME(col_text, '8:20:20.3554') f7,
  -> SUBTIME(col_text, '8:20:20.3554') f8 from test_func) t2;
  -> SUBTIME(col_text, '8:20:20.3554') f8 from test_func) t1
  -> UNION ALL
  -> SELECT * FROM ( SELECT
  -> GREATEST(2.22, col_text) f1, LEAST(2.22, col_text) f2,
  -> ADDDATE(col_text, INTERVAL '1.28.16.31' HOUR_MICROSECOND) f3,
  -> SUBDATE(col_text, INTERVAL '39.49.15' MINUTE_MICROSECOND) f4,
  -> DATE_SUB(col_text, INTERVAL '45' MICROSECOND) f5,
  -> DATE_ADD(col_text, INTERVAL '12.00.00.00.001' DAY_MICROSECOND) f6,
  -> ADDTIME(col_text, '8:20:20.3554') f7,
  -> SUBTIME(col_text, '8:20:20.3554') f8 from test_func) t2;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql>
mysql> DESC test1;
| Field | Type | Null | Key | Default | Extra |
+----+---+----+----+
| f1 | binary(23) | YES | | NULL |
| f2 | binary(23) | YES | | NULL |
    | char(29) | YES | NULL |
| char(29) | YES | NULL |
| f3
     |char(29) | YES | | NULL |
    | char(29) | YES | NULL
| char(29) | YES | NULL
l f6
l f7
| f8 | char(29) | YES | | NULL |
8 rows in set (0.01 sec)
```

- 在开启精度传递的场景下,对于CREATE TABLE AS SELECT A % (CASE WHEN)语句,如果A是DECIMAL类型,case when结果为日期类型(DATE、TIME、DATETIME),两者进行取模运算(%)得到的精度保持差异。GaussDB得到的精度跟decimal类型与日期类型直接取模运算得到的精度保持一致。

```
-- GaussDB: (decimal % date类型case)与(numeric%date), 精度一致,都是decimal(24,10)。m_db=# SET m_format_behavior_compat_options = 'enable_precision_decimal'; SET
```

m_db=# DROP TABLE IF EXISTS t1, t2;

DROP TABLE

m_db=# CREATE TABLE t1 (num_var numeric(20, 10), date_var date, time_var time(6), dt_var datetime(6));

```
CREATE TABLE
m_db=# CREATE TABLE t2 AS SELECT num_var % (CASE WHEN true THEN dt_var ELSE dt_var
END) AS res1 FROM t1;
INSERT 0 0
m_db=# DESC t2;
Field | Type | Null | Key | Default | Extra
res1 | decimal(24,10) | YES | |
(1 row)
m db=# DROP TABLE IF EXISTS t1, t2;
DROP TABLE
m_db=# CREATE TABLE t1 (num_var numeric(20, 10), date var date, time var time(6), dt_var
datetime(6));
CREATE TABLE
m_db=# CREATE TABLE t2 AS SELECT num_var % dt_var AS RES1 from t1;
INSERT 0 0
m_db=# DESC t2;
               | Null | Key | Default | Extra
Field | Type
res1 | decimal(24,10) | YES | | |
(1 row)
-- MySQL5.7,精度存在差异。(decimal % date类型case)精度为decimal(65,10),(numeric%date)
精度为decimal(24,10)。
mysql> DROP TABLE IF EXISTS t1, t2;
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE t1 (num_var numeric(20, 10), date_var date, time_var time(6), dt_var
datetime(6));
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE t2 AS SELECT num_var % (CASE WHEN true THEN dt_var ELSE dt_var
END) AS res1 FROM t1;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC t2;
+----+
| Field | Type | Null | Key | Default | Extra |
             -----+----+----+----+----
| res1 | decimal(65,10) | YES | | NULL | |
+-----+
1 row in set (0.00 sec)
mysql> DROP TABLE IF EXISTS t1, t2;
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE t1 (num_var numeric(20, 10), date_var date, time_var time(6), dt_var
datetime(6));
Query OK, 0 rows affected (0.02 sec)
mysgl> CREATE TABLE t2 AS SELECT num_var % dt_var AS res1 FROM t1;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC t2;
+----+
| Field | Type | Null | Key | Default | Extra |
               ----+----+----+----+--
| res1 | decimal(24,10) | YES | | NULL | |
1 row in set (0.00 sec)
-- MySQL8.0,(decimal % date类型case)和(numeric%date)精度都为decimal(20,10)。
mysql> DROP TABLE IF EXISTS t1, t2;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE t1 (num_var numeric(20, 10), date_var date, time_var time(6), dt_var
datetime(6));
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE t2 AS SELECT num_var % (CASE WHEN true THEN dt_var ELSE dt_var
END) AS res1 FROM t1;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC t2;
| Field | Type | Null | Key | Default | Extra |
+-----+-----
| res1 | decimal(20,10) | YES | | NULL | |
+----+
1 row in set (0.00 sec)
mysql> DROP TABLE IF EXISTS t1, t2;
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE t1 (num var numeric(20, 10), date var date, time var time(6), dt var
datetime(6));
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE t2 AS SELECT num_var % dt_var AS res1 FROM t1;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> DESC t2;
| Field | Type | Null | Key | Default | Extra |
+----+----+-----+-----+
| res1 | decimal(20,10) | YES | | NULL | |
+-----+
1 row in set (0.00 sec)
```

- 在开启精度传递的场景下,使用UNION,如果参与集合运算的查询语句,其查询的字段为常量,且查询的结果数据类型为INT/DECIMAL,则最后返回的精度存在差异。在MySQL 5.7中,返回的精度与UNION左右两侧的顺序有关;在MySQL 8.0中修复了这个问题,返回的精度与UNION左右两侧的顺序无关;在GaussDB中,返回的精度与UNION左右两侧的顺序无关,与MySQL 8.0一致,与MySQL 5.7不一致。

```
-- GaussDB:
m_db=# CREATE TABLE t1 AS (SELECT -23.45 c2) UNION ALL (SELECT -45.678 c2);
INSERT 0 2
m_db=# DESC t1;
Field | Type | Null | Key | Default | Extra
c2 | decimal(5,3) | YES | | |
(1 row)
m db=# CREATE TABLE t2 AS (SELECT -45.678 c2) UNION ALL (SELECT -23.45 c2);
INSERT 0 2
m_db=# DESC t2;
Field | Type | Null | Key | Default | Extra
c2 | decimal(5,3) | YES | | |
(1 row)
-- Mysql5.7:
mysql> CREATE TABLE t1 AS (SELECT -23.45 c2) UNION ALL (SELECT -45.678 c2);
Query OK, 2 rows affected (2.28 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> DESC t1;
    ---+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+
```

```
| c2 | decimal(6,3) | NO | | 0.000 | |
+----+
1 row in set (0.00 sec)
mysgl> CREATE TABLE t2 AS (SELECT -45.678 c2) UNION ALL (SELECT -23.45 c2);
Query OK, 2 rows affected (2.22 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> DESC t2;
| Field | Type | Null | Key | Default | Extra |
+-----+
| c2 | decimal(5,3) | NO | | 0.000 | |
+----+
1 row in set (0.00 sec)
-- Mysql8.0:
mysql> CREATE TABLE t1 AS (SELECT -23.45 c2) UNION ALL (SELECT -45.678 c2);
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> DESC t1;
    --+---+
. Field | Type | Null | Key | Default | Extra |
+-----+
| c2 | decimal(5,3) | NO | | 0.000 | |
+----+
1 row in set (0.03 sec)
mysql> CREATE TABLE t2 AS (SELECT -45.678 c2) UNION ALL (SELECT -23.45 c2);
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> DESC t2;
+----+
| Field | Type | Null | Key | Default | Extra |
+----+
| c2 | decimal(5,3) | NO | | 0.000 | |
+----+----+----+----+
1 row in set (0.02 sec)
```

双冒号转换差异点

GaussDB中使用双冒号将函数入参转换为期望类型可能导致结果超出预期;MySQL中无双冒号功能。

示例:

decimal 类型差异点

在Create table ... AS (select ...)语句中,使用decimal数据类型,若含有前缀0,M-Compatibility兼容模式下忽略前缀0,长度计算不包括0,在MySQL 5.7下,长度计算加上前缀0的数量,在MySQL 8.0下,无论存在多少个前缀0,长度计算只加上1。

```
加工削級U的致重,在MySQL 8.0下,无比存在多少下削級U,长度计算只加工T。
m_db=# create table test as select 004.01 col1;
INSERT 0 1
m_db=# desc test;
```

```
Field | Type | Null | Key | Default | Extra
col1 | decimal(3,2) | YES | | |
(1 row)
mysql 5.7
mysql> create table test as select 004.01 col1;
Query OK, 1 row affected (0.02 sec)
Records: 1 Duplicates: 0 Warnings: 0
mysgl> desc test;
| Field | Type | Null | Key | Default | Extra |
+----+
| col1 | decimal(5,2) | NO | | 0.00 | | +-----+
1 row in set (0.00 sec)
mysql 8.0
mysql> create table test as select 004.01 col1;
Query OK, 1 row affected (0.23 sec)
Records: 1 Duplicates: 0 Warnings: 0
mysql> desc test;
+----+
| Field | Type | Null | Key | Default | Extra |
+----+
| col1 | decimal(4,2) | NO | | 0.00 | |
+----+
1 row in set (0.01 sec)
```

4.3.3 系统函数

GaussDB数据库兼容绝大多数MySQL的系统函数,但存在部分差异。只能使用M-Compatibility兼容模式下的系统函数,不能使用原GaussDB的系统函数,避免产生非预期的结果。当前存在原GaussDB的系统函数和MySQL系统函数同名,但是M-Compatibility兼容模式下尚未支持这些函数的情况;一部分未支持的同名函数会提示用户在M-Compatibility兼容模式下不支持,另外一部分同名函数仍然保持原GaussDB系统函数的行为。用户禁止使用这些同名函数,避免产生非预期的结果。具体同名函数列表如下:

cot	isEmpty	last_insert_id	mod	octet_length
overlaps	point	radians	regexp_instr	regexp_like
regexp_replac e	regexp_substr	stddev_pop	stddev_samp	var_pop
var_samp	variance	-	-	-

表 4-34 M-Compatibility 兼容模式下保持原 GaussDB 系统函数行为的同名函数

ceil	decode	encode	format	instr
position	round	stddev	row_num	-

□ 说明

- 函数regexp_instr、regexp_like、regexp_replace、regexp_substr在参数
 m_format_dev_version值为's2'或以上版本并且参数m_format_behavior_compat_options值
 包含'enable_conflict_funcs'的情况下使用会报错,并提示M-Compatibility兼容模式数据库
 不支持;其他行为和《开发指南》中的"SQL参考 > 函数和操作符 > 字符处理函数和操作符"章节中的同名函数保持一致。
- MySQL数据库支持通过可加载函数接口,向MySQL中添加自定义函数,在调用此类函数时,函数的入参支持指定别名。GaussDB不支持可加载函数接口,在调用函数时,函数入参不支持指定别名。
- M-Compatibility模式下,系统函数存在以下公共差异:
 - 系统函数的返回值类型仅考虑入参node类型为Var(表中数据)和Const(常量输入) 类型时的情况与MySQL保持一致,其他情况(如入参为运算表达式、函数表达式等) 可能返回值的类型与MySQL有差异。
 - 系统函数在涉及LIMIT与OFFSET同时使用的查表场景下,由于GaussDB和MySQL的执行层机制不同,GaussDB会逐行调用函数,此行为会导致在存在报错的情况下会直接报错且中断执行,但MySQL不会逐行执行故不会报错中断,从而造成返回结果存在不一致。
 - 系统函数的调用不推荐使用pg_catalog.func_name()形式的调用,当被调用函数存在语法形式的入参时(如SELECT substr('demo' from 1 for 2)),函数的调用可能存在错误。

4.3.3.1 流程控制函数

表 4-35 流程控制函数列表

序号	MySQL数据库	GaussDB 数据库	差异
1	IF()	支持,有 差异。	当第一个参数为TRUE且第三个参数表达式中存在隐式类型转换错误,或者第一个参数为FALSE且第二个参数表达式中存在隐式类型转换错误时,MySQL会忽略该错误,GaussDB会提示类型转换错误。
2	IFNULL()	支持,有 差异。	第一个参数不为NULL且第二个参数表 达式中存在隐式类型转换错误时, MySQL会忽略该错误,GaussDB会提 示类型转换错误。
3	NULLIF()	支持,有 差异。	函数返回值类型在MySQL5.7和 MySQL8.0中存在差异,考虑到 MySQL8.0更合理,因此函数返回值类 型兼容MySQL8.0。

4.3.3.2 日期和时间函数

山 说明

以下为GaussDB M-Compatibility兼容性日期时间函数公共说明。

 《M-Compatibility开发指南》中"函数和操作符"章节函数入参为时间类型表达式的情况: 时间类型表达式主要包括TEXT、DATETIME、DATE或TIME,但所有可以隐式转换为时间表 达式的类型都可以作为入参,比如数字类型可以通过先隐式转化为TEXT,再作为时间类型表 达式生效。

生效的情况根据函数有所不同,比如: DATEDIFF函数由于只计算日期的差值,所以时间表达式会当做日期来解析。TIMESTAMPDIFF函数由于计算时间差值时是根据UNIT来决定的,所以会根据UNIT选择将时间表达式当做DATE或者TIME或者DATETIME来解析。

● 当SELECT子查询中包含且仅包含时间函数,且函数入参包含表中的列时,使用算数运算符(如+、-、*、/、取反等)对结果进行运算时,会截断日期与时间函数返回值后再进行算数运算。

```
m_db=# CREATE TABLE t1(int_var int);
CREATE TABLE
m_db=# INSERT INTO t1 VALUES(100);
INSERT 0 1
m_db=# SELECT (SELECT (1 * DATE_ADD('2020-10-20', interval int_var microsecond))) AS a FROM
t1; -- 不进行截断处理。
20201020000000
(1 row)
m_db=# SELECT (1 * (SELECT DATE_ADD('2020-10-20', interval int_var microsecond))) AS a FROM
t1; -- 进行截断处理。
a
2020
(1 row)
m_db=# SELECT 1 * a FROM (SELECT (SELECT 1 * DATE_ADD('2020-10-20', interval int_var
microsecond)) AS a FROM t1) AS t2; -- 不进行截断处理。
20201020000000
(1 row)
m_db=# SELECT 1 * a FROM (SELECT (SELECT DATE_ADD('2020-10-20', interval int_var
microsecond)) AS a FROM t1) AS t2; -- 进行截断处理。
1 * a
2020
(1 row)
```

● 函数入参为无效日期的情况:

一般而言,日期时间函数支持DATE、DATETIME的范围和MySQL保持一致。DATE支持的范围为'0000-01-01'到'9999-12-31',DATETIME支持的范围为'0000-01-01 00:00:00'到 '9999-12-31 23:59'。虽然GaussDB支持的DATE、DATETIME范围大于MySQL,但是越界仍然算无效日期。

大部分时间函数会告警并返回NULL,只有能通过cast正常转换的日期,才是正常合理的日期。

新框架下GaussDB的大部分日期时间函数与MySQL一致,一些函数的差异如下表所 示:

表 4-36 日期与和时间函数列表

序号	MySQL数据库	GaussD B数据库	差异
1	ADDDATE()	支持	-
2	ADDTIME()	支持	-
3	CONVERT_TZ()	支持	-
4	CURDATE()	支持	-
5	CURRENT_DATE()/ CURRENT_DATE	支持	-
6	CURRENT_TIME()/ CURRENT_TIME	支持,有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例: SELECT CURRENT_TIME(257) == SELECT CURRENT_TIME(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
7	CURRENT_TIMESTAMP()/ CURRENT_TIMESTAMP	支持,有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例: SELECT CURRENT_TIMESTAMP(257) == SELECT CURRENT_TIMESTAMP(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
8	CURTIME()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例: SELECT CURTIME(257) == SELECT CURTIME(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
9	DATE()	支持	-
10	DATE_ADD()	支持	-
11	DATE_FORMAT()	支持	-
12	DATE_SUB()	支持	-
13	DATEDIFF()	支持	-
14	DAY()	支持	-
15	DAYNAME()	支持	-
16	DAYOFMONTH()	支持	-
17	DAYOFWEEK()	支持	-
18	DAYOFYEAR()	支持	-
19	EXTRACT()	支持	-

序号	MySQL数据库	GaussD B数据库	差异
20	FROM_DAYS()	支持	-
21	FROM_UNIXTIME()	支持	-
22	GET_FORMAT()	支持	-
23	HOUR()	支持	-
24	LAST_DAY()	支持	-
25	LOCALTIME()/ LOCALTIME	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例SELECT LOCALTIME(257) == SELECT LOCALTIME(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
26	LOCALTIMESTAMP/ LOCALTIMESTAMP()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例SELECT LOCALTIMESTAMP(257) == SELECT LOCALTIMESTAMP(1))。 GaussDB只支持[0,6]合法值,其他值报 错。
27	MAKEDATE()	支持	-
28	MAKETIME()	支持	-
29	MICROSECOND()	支持	-
30	MINUTE()	支持	-
31	MONTH()	支持	-
32	MONTHNAME()	支持	-
33	NOW()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕(例SELECT NOW(257)==SELECT NOW(1))。 GaussDB只支持[0,6]合法值,其他值报 错。

序号	MySQL数据库	GaussD B数据库	差异
34	PERIOD_ADD()	支持, 有差异	1. 整数溢出处理的行为。 MySQL在5.7版本,此函数入参结果最大值为2^32=4294967296,在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题;在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。
			2. 负数period的表现。 MySQL在5.7版本,会将负数年份解析为异常值而不是报错。GaussDB入参或结果(如100年1月减去10000月)出现负数时报错。在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。
			3. period月份越界的表现。 MySQL在5.7版本中,若月份大于12 或等于0,例如200013、199900,会 将其顺延到之后的年份,或者将0月 作为上一年12月处理。在MySQL8.0 中已修复此问题,对越界月份报错。 GaussDB下此函数的表现与 MySQL8.0版本保持一致。
35	PERIOD_DIFF()	支持,有差异	1. 整数溢出处理的行为。 MySQL在5.7版本,此函数入参结果最大值为2^32=4294967296,在入参或结果的period对应的月份累加值以及month_number超过uint32范围时存在整数回绕问题;在MySQL8.0中已修复此问题。GaussDB下此函数的表现与MySQL8.0版本保持一致。
			2. 负数period的表现。 MySQL在5.7版本,会将负数年份解析为异常值而不是报错。GaussDB入参或结果(如100年1月减去10000月)出现负数时报错。在MySQL8.0中已修复此问题,对越界月份报错。GaussDB下此函数的表现与MySQL8.0版本保持一致。
			3. period月份越界的表现。 MySQL在5.7版本中,若月份大于12 或等于0,例如200013、199900,会 将其顺延到之后的年份,或者将0月 作为上一年12月处理。在MySQL8.0 中已修复此问题,对越界月份报错。 GaussDB下此函数的表现与 MySQL8.0版本保持一致。

序号	MySQL数据库	GaussD B数据库	差异
36	QUARTER()	支持	-
37	SEC_TO_TIME()	支持	-
38	SECOND()	支持	-
39	STR_TO_DATE()	支持	返回值与MySQL有差异,GaussDB返回 的是text,MySQL返回的是datetime、 date。
40	SUBDATE()	支持	-
41	SUBTIME()	支持	-
42	SYSDATE()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕。
			GaussDB不回绕。
43	TIME()	支持	-
44	TIME_FORMAT()	支持	-
45	TIME_TO_SEC()	支持	-
46	TIMEDIFF()	支持	-
47	TIMESTAMP()	支持	-
48	TIMESTAMPADD()	支持	-
49	TIMESTAMPDIFF()	支持	-
50	TO_DAYS()	支持	-
51	TO_SECONDS()	支持	在MySQL 5.7版本中,此函数的精度信息有误。
			开启精度传递参数下,GaussDB精度信息正常,和MySQL 8.0版本保持一致。
52	UNIX_TIMESTAMP()	支持	MySQL会根据入参是否存在小数位,决定返回定点型还是整型。当前GaussDB在内层嵌套操作符或函数时,返回的类型与MySQL可能存在不同。当内层节点返回定点、浮点、字符型、时间类型(不包括DATE类型)时,MySQL可能返回整型,GaussDB会返回定点型。
53	UTC_DATE()	支持	-
54	UTC_TIME()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕,GaussDB只支持[0,6]合 法值,其他值报错。

序号	MySQL数据库	GaussD B数据库	差异
55	UTC_TIMESTAMP()	支持, 有差异	MySQL入参整型值会按照一字节最大值 255整数回绕,GaussDB只支持[0,6]合 法值,其他值报错。
56	WEEK()	支持	-
57	WEEKDAY()	支持	-
58	WEEKOFYEAR()	支持	-
59	YEAR()	支持	-
60	YEARWEEK()	支持	-

4.3.3.3 字符串函数

表 4-37 字符串函数列表

序号	MySQL数据库	GaussDB数 据库	差异
1	ASCII()	支持。	-
2	BIT_LENGTH()	支持。	-
3	CHAR_LENGTH()	支持,有差 异。	GaussDB此函数如果数据库字符集是 SQL_ASCII,CHAR_LENGTH()会返回字节 数而非字符数。
4	CHARACTER_LE NGTH()	支持,有差 异。	GaussDB此函数如果数据库字符集是 SQL_ASCII,CHARACTER_LENGTH()会返 回字节数而非字符数。
5	CONCAT()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为BLOB;当 MySQL返回值类型为非二进制字符串类型 (CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。
6	CONCAT_WS()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为BLOB;当 MySQL返回值类型为非二进制字符串类型 (CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。
7	HEX()	支持。	-
8	LENGTH()	支持。	-

序号	MySQL数据库	GaussDB数 据库	差异
9	LPAD()	支持,有差异。	 MySQL默认最大填充长度为1398101,GaussDB默认最大长度为1048576。在不同字符集下,最大填充长度会有差异,例如字符集为'GBK'时,GaussDB默认最大长度为2097152。 如果数据库字符集是SQL_ASCII,可能产生未预期的结果。 当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)时,GaussDB对应的返回值类型为BLOB;当MySQL返回值类型为非二进制字符串类型(CHAR、VARCHAR、TEXT等)时,GaussDB对应的返回值类型为TEXT。
10	MD5()	支持,有差 异。	当BINARY类型插入字符串长度小于目标长度时,GaussDB填充符和MySQL不同;因此导致入参为BINARY类型时,函数结果和MySQL不一致。
11	RANDOM_BYTE S()	支持。	GaussDB与MySQL都使用OPENSSL生成随机字符串。GaussDB使用OPENSSL3.x.x生成随机字符串,与使用OPENSSL1.x.x版本的MySQL相比性能可能存在劣化。
12	REPEAT()	支持。	当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等)时, GaussDB对应的返回值类型为BLOB;当 MySQL返回值类型为非二进制字符串类型 (CHAR、VARCHAR、TEXT等)时, GaussDB对应的返回值类型为TEXT。
13	REPLACE()	支持。	 当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)时,GaussDB对应的返回值类型为 BLOB;当MySQL返回值类型为非二进制字符串类型(CHAR、VARCHAR、TEXT等)时,GaussDB对应的返回值类型为TEXT。 当第三个入参为null,且第二个入参的字符串长度不为0时,GaussDB返回NULL,MySQL可能返回第一个参数的字符。

序号	MySQL数据库	GaussDB数 据库	差异
14	RPAD()	支持,有差异。	 MySQL默认最大填充长度为1398101,GaussDB默认最大长度为1048576。在不同字符集下,最大填充长度会有差异,例如字符集为'GBK'时,GaussDB默认最大长度为2097152。 如果数据库字符集是SQL_ASCII,可能产生未预期的结果。 当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)时,GaussDB对应的返回值类型为BLOB;当MySQL返回值类型为非二进制字符串类型(CHAR、VARCHAR、TEXT等)时,GaussDB对应的返回值类型为TEXT。
15	SHA()/SHA1()	支持。	-
16	SHA2()	支持。	-
17	SPACE()	支持。	-
18	STRCMP()	支持,有差 异。	如果数据库字符集是SQL_ASCII,可能产生 未预期的结果。

序号	MySQL数据库	GaussDB数 据库	差异
19	FIND_IN_SET()	支持,有差	当指定数据库使用的字符编码是SQL_ASCII
20	LCASE()	异。 	时,服务器把字节值0~127根据ASCII标准解
21	LEFT()		符;如果该函数的输入输出包含了任何非 ASCII数据,数据库将无法帮助用户转换或
22	LOWER()		者校验非ASCII字符。
23	LTRIM()		当MySQL返回值类型为二进制字符串类型 (BINARY、VARBINARY、BLOB等) 时,
24	REVERSE()		GaussDB对应的返回值类型为BLOB; 当
25	RIGHT()		MySQL返回值类型为非二进制字符串类型 (CHAR、VARCHAR、TEXT等)时,
26	RTRIM()		GaussDB对应的返回值类型为TEXT。 SUBSTRING 函数在第一个入参为嵌套场景
27	SUBSTR()		下与MySQL存在差异:
28	SUBSTRING()		● 第一个入参节点返回的字符序为BINARY 时,MySQL可能依旧以不同的字符序逻
29	SUBSTRING_IN DEX()		辑处理(取决于内层嵌套的函数),而 GaussDB以BINARY字符序进行函数处 理,导致截取的字节长度不同。
30	TRIM()		SUBSTRING_INDEX函数存在差异:
31	UCASE()		● 在第三个入参为负数时,MySQL与
32	UPPER()		GaussDB的比较逻辑不同,会导致结果 可能存在差异。
			• 当第三个入参为正数时,由于MySQL5.7 以int32位存储,会存在回绕问题导致结 果不正确。MySQL 8.0修复了该问题以 int64为存储,GaussDB以8.0为准。当入 参值超过2^63 -1时,也会发生回绕,可 能导致第三个参数获取为负数,结果存 在差异。
33	UNHEX()	支持。	MySQL的返回值类型为BINARY、 VARBINARY、BLOB、MEDIUMBLOB或 LONGBLOB;GaussDB返回值类型固定为 LONGBLOB。
34	FIELD()	支持。	-
35	COMPRESS()	支持,有差 异。	MySQL返回类型为VARBINARY、BLOB或 LONGBLOB;GaussDB返回二进制字符串类 型LONGBLOB。
36	UNCOMPRESS()	支持。	-
37	UNCOMPRESS_ LENGTH()	支持。	-
38	EXPORT_SET()	支持。	-
39	POSITION()	支持。	-

序号	MySQL数据库	GaussDB数 据库	差异
40	LOCATE()	支持。	-
41	CHAR()	支持,有差 异。	● CHAR函数指定字符集时,若转码失败, GaussDB产生报错,MySQL提示 WARNING并返回NULL。
			 MySQL在参数为ASCII表中第0~31个和 127个码值时,返回结果不可见, GaussDB会以\x01、\x02等16进制返 回。
			MySQL中CHAR函数入参个数无限制, GaussDB函数的入参不超过8192个。
42	ELT()	支持,有差 异。	MySQL中ELT函数入参个数无限制, GaussDB函数的入参不超过8192个。
43	FORMAT()	支持。	-
44	BIN()	支持。	-
45	MAKE_SET()	支持。	MySQL 5.7版本,当MAKE_SET函数选中的第一个参数为整型、浮点型或定点型且返回结果中存在非ASCII字符,显示结果可能为乱码;GaussDB显示结果正常,和MySQL8.0版本保持一致。
46	TO_BASE64()	支持。	-
47	FROM_BASE64()	支持。	-
48	ORD()	支持。	-
49	MID()	支持。	-

序号	MySQL数据库	GaussDB数 据库	差异
50	QUOTE()	支持,有差异。	 在M-Compatibility兼容模式下,开启MySQL转义:SETm_format_behavior_compat_options=enable_escape_string; 当入参字符串中含"\0"时, GaussDB中由于字符集不支持导致无法输入。由转义字符导致的本函数与MySQL的差异,与本函数无关。 GaussDB最大支持1GB数据传输,str入参长度最大支持536870908,函数返回结果字符串最大支持1GB。 入参为固定长度BINARY类型时,对于未填充字符:MySQL默认补充空字符"\0",GaussDB默认补充空格。 在MySQL中,QUOTE()会处理空字符,GaussDB中,QUOTE()无法处理空字符。
51	INSERT()	支持。	-
52	INSTR()	支持。	-

4.3.3.4 强制转换函数

表 4-38 强制转换函数列表

序号	MySQL数据 库	GaussDB数据 库	差异
1	CAST()	支持	 由于函数执行机制不同,flags无法传递给内层函数,在cast函数嵌套其他函数(如greatest、least等)时,内层函数返回小于1的值,结果与MySQL不一致。 GaussDB: m_db=# SELECT cast(least(1.23, 1.23, 0.23400) AS date); WARNING: Incorrect datetime value: '0.23400' CONTEXT: referenced column: cast cast cast cast cast cast cast cast

序号	MySQL数据 库	GaussDB数据 库	差异
			(1 row) Mysql 5.7 mysql> CREATE TABLE sub_query_table (myfloat float); Query OK, 0 rows affected (0.02 sec) mysql> INSERT INTO sub_query_table (myfloat) VALUES (1.23); Query OK, 1 row affected (0.00 sec) mysql> SELECT BINARY (select MyFloat from sub_query_table) from sub_query_table; ++ BINARY (select MyFloat from sub_query_table) ++ 1.2300000190734863
2	CONVERT()	支持	 GaussDB不支持使用CONVERT(expr, CHAR[(N)] charset_info或者 CAST(expr, NCHAR[(N)])转换字符集。 GaussDB支持使用CONVERT(expr, FLOAT[(p)])或CONVERT(expr, DOUBLE)将表达式转换为浮点类型, MySQL 5.7版本不支持此转换。

4.3.3.5 加密函数

表 4-39 加密函数列表

序号	MySQL数 据库	GaussDB 数据库	差异
1	AES_DECRY PT()	支持。	1. ecb为不安全加密模式,GaussDB不支持,默 认为cbc模式。
2	AES_ENCRY PT()	支持。	 GaussDB中,当指定数据库使用的字符编码是SQL_ASCII时,服务器把字节值0-127根据ASCII标准解释,而字节值128-255则当作无法解析的字符;如果该函数的输入输出包含了任何非ASCII数据,数据库将无法帮助用户转换或者校验非ASCII字符。 MySQL的返回值类型为BINARY、VARBINARY、BLOB、MEDIUMBLOB、LONGBLOB,GaussDB返回值类型固定为LONGBLOB。 GUC参数: block_encryption_mode不支持设置数字。
3	PASSWOR D()	支持,有差异。	 MySQL中可以通过GUC参数old_passwords控制生成密码的哈希方式: old_passwords的默认值为0。 old_passwords为0:表示使用MySQL 4.1 native hashing加密。 old_passwords为2:表示使用SHA-256 hashing加密。 GaussDB中没有实现GUC参数 old_passwords,password函数的行为只与默认(即old_passwords为0)的行为保持一致。 当BINARY类型插入字符串长度小于目标长度时,GaussDB填充符和MySQL不同;因此入参为BINARY类型时,函数结果和MySQL不一致。

4.3.3.6 比较函数

表 4-40 比较函数列表

序号	MySQL数据 库	GaussDB 数据库	差异
1	COALESCE(支持,有差 异。	union distinct场景下,返回值精度与 MySQL不完全一致。 当第一个不为NULL的参数的后续参数表达 式中存在隐式类型转换错误时,MySQL会忽 略该错误,GaussDB会提示类型转换错误。 当参数为MIN函数、MAX函数时,返回值类 型与MySQL不一致。
2	INTERVAL()	支持。	-
3	GREATEST(支持,有差 异。	当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)时,GaussDB对应的返回值类型为LONGBLOB;当MySQL返回值类型为非二进制字符串类型(CHAR、VARCHAR、TEXT等)时,GaussDB对应的返回值类型为TEXT。当该函数入参含有NULL且在WHERE关键字之后调用,返回结果与MySQL5.7不一致,此处为MySQL5.7存在的问题,MySQL8.0修复了该问题,目前GaussDB和MySQL8.0保持一致。
4	LEAST()	支持,有差 异。	当MySQL返回值类型为二进制字符串类型(BINARY、VARBINARY、BLOB等)时,GaussDB对应的返回值类型为LONGBLOB;当MySQL返回值类型为非二进制字符串类型(CHAR、VARCHAR、TEXT等)时,GaussDB对应的返回值类型为TEXT。 当该函数入参含有NULL且在WHERE关键字之后调用,返回结果与MySQL5.7不一致,此处为MySQL5.7存在的问题,MySQL8.0修复了该问题,目前GaussDB和MySQL8.0保持一致。

序号	MySQL数据 库	GaussDB 数据库	差异
5	ISNULL()	支持,有差异。	 函数返回值类型在MySQL5.7和MySQL8.0的行为更为合理,因此函数返回值类型兼容MySQL8.0。 内层嵌套部分聚合函数时,部分场景返回结果MySQL5.7和MySQL8.0中存在差异,结合MySQL8.0的行为更为合理,因此函数返回值兼容MySQL8.0。m_db=# SELECT isnull(avg(1.23)); ?column?

4.3.3.7 聚合函数

表 4-41 聚合函数列表

序号	MySQL数据 库	GaussDB数 据库	差异
1	AVG()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。
			GaussDB中当expr中的列为BIT、BOOL、整数类型,且所有行的和超过BIGINT的范围时,会发生溢出导致整数翻转。
			● GaussDB在AVG函数入参为TEXT/BLOB类型 时行为存在差异:
			- MySQL5.7中,AVG(tEXT/BLOB)返回 值类型为MEDIUMTEXT类型;MySQL8.0 中,AVG(TEXT/BLOB)返回值类型为 DOUBLE类型。
			- 在GaussDB中,AVG(TEXT/BLOB)返 回值类型与MySQL 8.0版本保持一致。
2	BIT_AND()	支持。	BIT_AND函数入参为NULL且被其他函数嵌套时行为有差异:在MySQL 5.7中,结果为-1;在MySQL 8.0中,结果为NULL;在GaussDB中,此函数嵌套的表现与MySQL 8.0版本保持一致。 GaussDB: m_db=# SELECT acos(bit_and(null)); acos
3	BIT_OR()		-
4	BIT_XOR()	支持。	-
5	COUNT()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。 GaussDB支持count(tablename.*)语法, MySQL不支持。

序号	MySQL数据 库	GaussDB数 据库	差异
6	GROUP_CO NCAT()	支持,有差 异。	 GaussDB中指定DISTINCT且SQL语句包含 GROUP BY子句时,不保证结果的顺序。
			 GaussDB中当GROUP_CONCAT参数中同时 有DISTINCT和ORDER BY语法时,所有 ORDER BY后的表达式必须也在DISTINCT的 表达式之中。
			 GaussDB中GROUP_CONCAT(ORDER BY 数字)不代表按照第几个参数的顺序,数字 只是一个常量表达式,相当于不排序。
			 GaussDB中使用参数group_concat_max_len 限制GROUP_CONCAT最大返回长度,超长 截断,目前能返回的最大长度是 1073741823,小于MySQL。
			默认UTF8字符集下,由于GaussDB的UTF8字符集的最大字节数与MySQL的UTF8字符集最大字节数不同。会导致创出来表结构与MySQL存在差异。
			GaussDB: m_db=# SET m_format_behavior_compat_options='enable_precision_ decimal'; SET
			m_db=# CREATE TABLE t1 AS SELECT * FROM (SELECT case WHEN 1 < 2 THEN group_concat(1.23, 3.24) ELSE 12.34 END v1) c1; INSERT 0 1 m_db=# DESC t1; Field Type Null Key Default Extra
			v1 varchar(256) YES (1 row) MySQL 5.7: mysql> CREATE TABLE t1 AS SELECT * FROM (SELECT case WHEN 1 < 2 THEN group_concat(1.23, 3.24) ELSE 12.34 END v1) c1; Query OK, 1 row affected (0.01 sec) Records: 1 Duplicates: 0 Warnings: 0
			mysql> DESC t1;
			++ Field Type
			v1
			1 row in set (0.00 sec) GROUP CONCAT函数作为NULLIF函数的入
			参,嵌套场景行为有差异:在MySQL 5.7中,NULLIF入参嵌套GROUP_CONCAT与非嵌套GROUP_CONCAT的数值会判断为相等,返回NULL;在MySQL 8.0中,因精度差异会判断为不等;在GaussDB中,此函数嵌套的表现与MySQL 8.0版本保持一致。
			GaussDB: m_db=# SELECT nullif(group_concat(1/7), 1/7); nullif

序号	MySQL数据 库	GaussDB数 据库	差异
			0.1429 (1 row) MySQL 5.7: mysql> SELECT nullif(group_concat(1/7), 1/7); ++ nullif(group_concat(1/7), 1/7) ++ NULL
7	MAX()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时,不保证结果的顺序。当参数为非表字段时,MAX函数返回值类型和MySQL 5.7不一致。 开启精度传递下,MAX函数嵌套time、date、datetime、timestamp类型的时间间隔运算,返回值及返回类型与MySQL 8.0保持一致。 开启精度传递下,MAX函数和INTERVAL的时间间隔运算,返回值及返回类型与MySQL 8.0保持一致。
8	MIN()	支持,有差 异。	GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时,不保证结果的顺序。当参数为非表字段时,MIN函数返回值类型和MySQL 5.7不一致。 开启精度传递下,MIN函数嵌套time、date、datetime、timestamp类型的时间间隔运算,返回值及返回类型与MySQL 8.0保持一致。开启精度传递下,MIN函数和INTERVAL的时间间隔运算,返回值及返回类型与MySQL 8.0保持一致。
9	SUM()	支持,有差 异。	 GaussDB中指定DISTINCT且SQL语句包含GROUP BY子句时,不保证结果的顺序。 GaussDB中当expr中的列为BIT、BOOL、整数类型,且所有行的和超过BIGINT的范围时,会发生溢出导致整数翻转。
10	STD()	支持。	-

序号	MySQL数据 库	GaussDB数 据库	差异
11	聚合函数	支持,有差异。	● ORDER BY语句中包含聚合函数GaussDB不报错,MySQL会报错。 ● 在未开启精度传递(没有设置m_format_behavior_compat_options = 'enable_precision_decimal')的情况下,当聚合函数以其他函数、操作符或SELECT子句等表达式作为入参时(如SELECT sum(abs(n)) FROM t),聚合函数将获取不到入参表达式传递的精度信息,导致函数的结果精度与MySQL有差异。 ● 聚合函数的结果与数据输入顺序相关,不同的数据输入顺序会导致结果存在差异。 - 例如与ORDER BY同时使用时,改变了聚合函数的执行顺序,会导致结果与MySQL不一致。 - 准备基表: CREATE TABLE test_n(col_unumeric1 decimal(4,3) unsigned, col_znumeric2 decimal(3,2) unsigned zerofill, col_znumeric3 decimal(5,3) unsigned zerofill, col_sy 3,338), (1.190, 2.29, 3.339), (1.180, 2.28, 3.338); Query OK, 3 rows affected (0.00 sec) Records: 3 Duplicates: 0 Warnings: 0 CREATE TABLE test_n_2(col_unumeric1 decimal(4,3) unsigned, col_znumeric3 decimal(5,3) unsigned zerofill, col_znumeric3 decimal(5,3) unsigned zerofill); Query OK, 0 rows affected (0.00 sec) Records: 3 Duplicates: 0 Warnings: 0 CREATE TABLE test_n_2 VALUES(1.180, 2.28, 3.338), (1.190, 2.29, 3.339), (1.010, 2.02, 3.303); Query OK, 3 rows affected (0.00 sec) Records: 3 Duplicates: 0 Warnings: 0 CREATE TABLE test_n_2 VALUES(1.180, 2.28, 3.338), (1.190, 2.29, 3.339), (1.010, 2.02, 3.303); Query OK, 3 rows affected (0.00 sec) Records: 3 Duplicates: 0 Warnings: 0 CREATE TABLE IF NOT EXISTS fun_op_case_tb_1 (id int, name varchar(20), col_unumeric1 NUMERIC(4,3) unsigned, col_znumeric2 DECIMAL(3,2) zerofill,col_znumeric2 DECIMAL(3,2) zerofill,col_znumeric2 col_znumeric3 numeric3 nu

序号	MySQL数据 库	GaussDB数 据库	差异
			(1,软件', 1.11, 2.12, 3.133), (2,软件', 3.11, 2.32, 3.233), (4,软件', 1.41, 2.42, 3.343), (5,软件', 1.41, 2.42, 3.343), (6,软件', 1.61, 2.26, 3.363), (7,软件', 1.17, 2.27, 3.337), (8,软件', 1.17, 2.27, 3.337), (8,软件', 1.19, 2.29, 3.339), (10,软件', 1.01, 2.02, 3.303), (1,数据库', 1.11, 2.12, 3.133), (2,数据库', 2.11, 2.12, 3.133), (3,数据库', 3.11, 2.32, 3.333), (4,数据库', 1.11, 2.12, 3.133), (5,数据库', 1.17, 2.27, 3.337), (8,数据库', 1.16, 2.26, 3.363), (7,数据库', 1.17, 2.27, 3.337), (8,数据库', 1.10, 2.02, 3.303), (10,数据库', 1.10, 2.02, 3.303), (10,数据库', 1.10, 2.02, 3.303), (10,数据库', 1.10, 2.02, 3.303); INSERT 0.30 GaussDB: m_db=# SELECT* FROM test_n; col_unumeric1 [col_znumeric2] col_znumeric3

序号	MySQL数据 库	GaussDB数 据库	差异
			1.180 2.28 03.338 +++++ 3 rows in set (0.00 sec) mysql> SELECT *FROM test_n_2; +++
			col_znumeric3)) FROM test_n_2; +

序号	MySQL数据 库	GaussDB数 据库	差异
			(数据库, 666,-55.155), (数据库, 928,-53.963), (数据库, 666,-54.555), (数据库, 666,-54.555), (数据库, 666,-54.555), (数据库, 666,-55.555); GaussDB: m_db=# select name, std(c1/c2) c5 from t1 group by name with rollup; name c5

序号	MySQL数据 库	GaussDB数 据库	差异
			m_db=# SELECT sum(col_bit1/col_time2) FROM fun_op_case_tb_1 WHERE id<=10 GROUP BY name; sum

4.3.3.8 JSON 函数

表 4-42 JSON 函数列表

序号	MySQL数据库	GaussDB数据库	差异
1	JSON_APPEND()	支持。	-
2	JSON_ARRAY()	支持。	-
3	JSON_ARRAY_APP END()	支持。	-
4	JSON_ARRAY_INSE RT()	支持。	-
5	JSON_CONTAINS()	支持。	-
6	JSON_CONTAINS_ PATH()	支持。	-
7	JSON_DEPTH()	支持。	-
8	JSON_EXTRACT()	支持。	-
9	JSON_INSERT()	支持。	-

序号	MySQL数据库	GaussDB数据库	差异
10	JSON_KEYS()	支持。	-
11	JSON_LENGTH()	支持。	-
12	JSON_MERGE()	支持。	-
13	JSON_MERGE_PAT CH()	支持。	-
14	JSON_MERGE_PRE SERVE()	支持。	-
15	JSON_OBJECT()	支持。	-
16	JSON_QUOTE()	支持。	-
17	JSON_REMOVE()	支持。	-
18	JSON_REPLACE()	支持。	-
19	JSON_SEARCH()	支持。	-
20	JSON_SET()	支持。	-
21	JSON_TYPE()	支持。	-
22	JSON_UNQUOTE()	支持,有差异。	在转义字符中\0和\uxxxx的场景与 MySQL有差异: SELECT JSON_UNQUOTE("\0"); mysql> SELECT JSON_UNQUOTE("\0"); ERROR 3141 (22032): Invalid JSON text in argument 1 to function json_unquote: "Missing a closing quotation mark in string." at position 1. m_db=# select JSON_UNQUOTE("\0"); ERROR: invalid byte sequence for encoding "UTF8": 0x00 SELECT JSON_UNQUOTE("\\u4656"); mysql> SELECT JSON_UNQUOTE("\\u4656"); ++ JSON_UNQUOTE("\\u4656") ++ row in set (0.00 sec) m_db=# select JSON_UNQUOTE("\\u4656"); json_unquote
23	JSON_VALID()	支持。	-

□ 说明

JSON函数差异说明:对于JSON函数和其他字符入参函数,如果含有转义字符的输入,默认情况下与MySQL存在一定差异,需要设置GUC参数(SET m_format_behavior_compat_options = 'enable_escape_string';);对于转义字符的场景才会与MySQL兼容,在转义字符中,\f、\Z、\0和\uxxxx的场景与MySQL有差异。

4.3.3.9 窗口函数

表 4-43 窗口函数列表

序号	MySQL数 据库	GaussDB 数据库	差异
1	LAG()	支差异。	● 偏移量N的取值范围不同:

序号	MySQL数 据库	GaussDB 数据库	差异
			- 对于表中的列类型(Type)为BIGINT类型或 INTEGER类型时会显示宽度。
			– 对于表中的列类型(Type)的宽度为0时,不 显示宽度,如binary(0)只显示成binary。
			– 表结构中标识Null和Default值的列,功能暂 未实现。
			该函数作为子查询结合CREATE TABLE AS使用 时,单独执行该函数的子查询语句没有报错或告 警时:
			– GaussDB在严格模式和宽松模式下,CREATE TABLE AS语句执行成功,建表成功。
			- MySQL在严格模式下,CREATE TABLE AS语 句执行可能报错,建表失败。
2	LEAD()	支持,有 差异。	与LAG()函数差异点相同。
3	ROW_NU MBER()	支持,有 差异。	ORDER BY子句排序时,NULL值的排序不同: MySQL中,NULL值默认升序排在前面。 GaussDB中,NULL值默认升序排在后面。

山 说明

窗口函数差异说明:MySQL数据库管理系统在调用窗口函数时,OVER子句中的ORDER BY子句与PARTITION BY子句不支持使用列别名,GaussDB数据库支持使用列别名。

4.3.3.10 数字操作函数

表 4-44 数字操作函数列表

序号	MySQL数据库	GaussDB数据 库	差异
1	ABS()	支持。	-
2	ACOS()	支持。	-
3	ASIN()	支持。	-
4	ATAN()	支持。	-
5	ATAN2()	支持。	-

序号	MySQL数据库	GaussDB数据 库	差异
6	CEILING()	支持,有差异。	部分操作结果类型以及CREATE TABLE AS的精度与MySQL不一致。 1. 入参为INT类型时,GaussDB返回值类型为BIGINT,MySQL返回值类型为INT。 2. 入参为BIGINT类型或BIGINT UNSIGNED类型,入参值大于等于20位时(包括符号位),GaussDB返回整型,MySQL5.7返回DECIMAL。该情况下会导致嵌套场景时,CEILING函数作为内层函数存在差异。SET m_format_behavior_compat_options='e nable_precision_decimal'; CREATE TABLE tt AS SELECT ceiling(-9223372036854775808); DESC tt; MySQL表字段返回类型为:DECIMAL(16,0)。GaussDB表字段返回类型为:BIGINT(17)。 3. 入参为NUMERIC类型时,返回类型可能与MySQL不一致。对于其他类型可能与MySQL不一致。对于其他类型的入参如嵌套情况,结果会有差异,GaussDB返回NUMERIC类型,MySQL可能返回整型。SET m_format_behavior_compat_options='e nable_precision_decimal'; CREATE TABLE t AS SELECT ceiling(-5.5); DESC t; MySQL表字段返回类型为:INT(5)。GaussDB表字段返回类型为:INT(5)。 GaussDB表字段返回类型为:INT(5)。SET m_format_behavior_compat_options='e nable_precision_decimal'; CREATE TABLE t AS SELECT ceiling(-5.5); DESC t; MySQL表字段返回类型为:INT(5)。SET m_format_behavior_compat_options='e nable_precision_decimal'; CREATE TABLE t AS SELECT ceiling(abs(5.5)); DESC t; MySQL表字段返回类型为:INT(4)。GaussDB表字段返回类型为:INT(4)。GaussDB表字段返回类型为:INT(4)。GaussDB表字段返回类型为:DECIMAL(3,0)。

序号	MySQL数据库	GaussDB数据 库	差异
7	COS()	支持。	-
8	DEGREES()	支持。	-
9	EXP()	支持。	-

序号	MySQL数据库	GaussDB数据 库	差异
10	FLOOR()	支持,有差异。	部分操作结果类型以及CREATE TABLE AS的精度与MySQL不一致。 1. 入参为INT类型时,GaussDB返回值类型为BIGINT,MySQL返回值类型为INT。 2. 入参为BIGINT类型或BIGINT UNSIGNED类型,入参值大于等于20位时(包括符号位),GaussDB返回整型,MySQL5.7返回DECIMAL。该情况下会导致嵌套场景时,floor函数作为内层函数存在差异。 宽松模式下:SET m_format_behavior_compat_options='e nable_precision_decimal';CREATE TABLE tt AS SELECT floor(-9223372036854775808);DESC tt;MySQL表字段返回类型为:DECIMAL(16,0)。GaussDB表字段返回类型为:BIGINT(17)。 3. 入参为NUMERIC类型时,返回类型对常与性类型的入参如嵌套情况,结果会有差异,GaussDB返回NUMERIC类型,MySQL可能返回整型。SET m_format_behavior_compat_options='e nable_precision_decimal';CREATE TABLE t AS SELECT floor(-5.5);DESC t;MySQL表字段返回类型为:INT(5)。GaussDB表字段返回类型为:INT(5)。GaussDB表字段返回类型为:INT(5)。 GaussDB表字段返回类型为:INT(5)。 GaussDB表字段返回类型为:INT(5)。 GaussDB表字段返回类型为:INT(5)。 GaussDB表字段返回类型为:INT(4)。GaussDB表字段返回类型为:INT(4)。GaussDB表字段返回类型为:INT(4)。GaussDB表字段返回类型为:DECIMAL(3,0)。

序号	MySQL数据库	GaussDB数据 库	差异
11	LN()	支持。	-
12	LOG()	支持。	-
13	LOG10()	支持。	-
14	LOG2()	支持。	-
15	PI()	支持。	精度传递开关关闭的情况下,也即m_format_behavior_compat_options中的enable_precision_decimal未设置时,PI函数的返回值精度与MySQL的有差异:MySQL中PI函数的结果仅保留四舍五入之后的小数后6位,而GaussDB的结果会保留四舍五入之后的小数后15位。
16	POW()	支持。	-
17	POWER()	支持。	-
18	RAND()	支持。	-
19	SIGN()	支持。	-
20	SIN()	支持。	-
21	SQRT()	支持。	-
22	TAN()	支持。	-
23	TRUNCATE()	支持。	-
24	CEIL()	支持。	-
25	CRC32()	支持,有差 异。	当BINARY类型插入字符串长度小于目标长度时,GaussDB填充符和MySQL不同;因此入参为BINARY类型时,函数结果和MySQL不一致。
26	CONV()	支持。	-

4.3.3.11 网络地址函数

表 4-45 网络地址函数列表

序号	MySQL数 据库	GaussDB 数据库	差异
1	INET_ATO N()	支持。	-

序号	MySQL数 据库	GaussDB 数据库	差异
2	INET_NTO A()	支持。	-
3	INET6_AT ON()	支持。	-
4	INET6_NT OA()	支持,有 差异。	GaussDB中,有效入参类型支持VARBINARY或 BINARY类型; MySQL中,有效入参类型还支持TINYBLOB、 MEDIUMBLOB、LONGBLOB或BLOB类型。
5	IS_IPV6()	支持。	-
6	IS_IPV4()	支持。	-

4.3.3.12 其他函数

表 4-46 其他函数列表

序号	MySQL数 据库	GaussDB 数据库	差异
1	DATABAS E()	支持。	-
2	UUID()	支持。	-
3	UUID_SH ORT()	支持。	-

序号	MySQL数 据库	GaussDB 数据库	差异	
4	ANY_VAL UE()	支差异。	● 作为分组的第一条数据是不确定的,与底层算子相关。例如同一条sql语句,GaussDB返回5和 4,MySQL返回5和2。 CREATE TABLE t1(a IINT, b INT); INSERT INTO t1 VALUES(2, 4); INSERT INTO t1 VALUES(2, 2); INSERT INTO t1 VALUES(3, 9); INGERT INTO t2 VALUES(1, 9); INGERT INTO t2 VALUES(1, 9); INGERT INTO t1 VALUES(1, 10); INGERT INTO t1 VALUES(1, 10); INSERT INTO t1 VALUES(1, 10); INGERT INTO t1	

序号	MySQL数 据库	GaussDB 数据库	差异	
5	SLEEP()	支持,有 差异。	 调用SLEEP函数的过程中,若使用Ctrl+C提前结束调用,GaussDB只显示Cancel request sent信息,与MySQL显示不一致。 	
			除上述情况外,当SLEEP函数在其他SQL语句中被调用时,使用Ctrl+C提前结束语句,若操作恰好被SLEEP函数内部获取,则不会报错,若被系统中其他函数获取则报错,与MySQL表现不一致。	
			● 若在SLEEP函数执行的过程中,其所在的进程被相关命令结束(如:SELECT PG_TERMINATE_BACKEN(xxx);),GaussDB目前行为为报错处理,与MySQL不一致。	
6	COLLATI ON()	支持,有 差异。	GaussDB仅支持utf8、utf8mb4、gbk、gb18030 和latin1字符集下的字符序。	
7	FOUND_R OWS()	支持。	-	
8	ROW_CO UNT()	支持,有 差异。	● GaussDB没有SIGNAL语句,MySQL支持 SIGNAL语句。	
			● GaussDB中,因为不存在连接参数 CLIENT_FOUND_ROWS(设置之后返回匹配行 数而不是影响行数),不受此参数的影响,统一 返回影响行数,MySQL受此参数影响。	
			● GaussDB中INSERT ON DUPLICATE KEY UPDATE中触发冲突的场景,如果冲突一条返回 1,MySQL返回2。	
9	SYSTEM_ USER()	支持,有 差异。	MySQL的配置文件中配置skip-name-resolve时, 不会将127.0.0.1或::1解析为localhost,GaussDB没 有相关参数,始终将127.0.0.1和::1解析为 localhost。	
10	DEFAULT()	支持,有 差异。	GaussDB支持使用列别名,MySQL不支持。	
11	BENCHM ARK()	支持,有 差异。	 因为MySQL和GaussDB执行层框架存在差异,所以MySQL和GaussDB使该函数评估同一个表达式的执行时间不具有可比性。该函数仅用于评估GaussDB不同表达式的执行效率对比。 执行时间较长时,当在客户端输入Ctrl+C时,MySQL返回0后结束任务,GaussDB统一显示Cancel request sent后结束任务。 	

4.3.4 操作符

GaussDB数据库兼容绝大多数MySQL的操作符,但存在部分差异。如未列出,操作符行为默认为GaussDB原生行为,目前存在MySQL不支持但是GaussDB支持的语句,不建议使用这类语句。

操作符差异

- ORDER BY排序对NULL值处理的差异。MySQL在排序时会将NULL值排序在前面;GaussDB默认将NULL值默认排在最后面。GaussDB可以通过nulls first和nulls last设置NULL值排序顺序。
- 有ORDER BY时, GaussDB输出顺序与MySQL一致。没有ORDER BY时, GaussDB不保证结果有序。
- MySQL操作符要使用括号严格表达式的结合性,否则执行报错。例如: SELECT 1 regexp ('12345' regexp '123');。
 - GaussDB M-Compatibility操作符不用括号严格表达的结合性也能成功执行。
- NULL值显示不同。MySQL会将NULL显示为"NULL";GaussDB将NULL值显示为空值。

MySQL输出结果:

```
mysql> Select NULL;
+-----+
| NULL |
+-----+
| NULL |
+-----+
1 row in set (0.00 sec)
```

GaussDB输出结果:

m_db=# select NULL;
?column?
-----(1 row)

- 操作符执行后,列名显示不一致。MySQL会将NULL显示为"NULL"; GaussDB将NULL值显示为空值。
- 字符串转double遇到非法字符串时,告警信息不一致。MySQL在常量非法字符串报错,字段非法字符串不报错;GaussDB在常量非法字符串和字段非法字符串都报错。
- 比较操作符返回结果显示不同。MySQL返回1/0; GaussDB返回t/f。

表 4-47 操作符

序号	MySQL数据库	GaussDB数 据库	差异
1	<>	支持,但有 差异。	MySQL支持索引,GaussDB不支持索 引。
2	<=>	支持,但有 差异。	MySQL支持索引,GaussDB不支持索 引、hash连接和合并连接。

序号 MySQL数据库	GaussDB数 据库	差异
3 行表达式	支持,但有差异。	● MySQL支持<=>操作符行比较。 GaussDB不支持<=>操作符行比较。 MySQL不支持行表达式与NULL比较。GaussDB支持<、<=、=、>=、>=、>、<>操作符对行表达式与NULL追比较。 MySQL不支持IS NULL、ISNULL对行表达式的操作。GaussDB支持。 ● 操作符对于行表达式的不支持的操作,GaussDB错误信息与MySQL不一致。 ● MySQL不支持ROW(values)其中values只有一列数据,GaussDB支持。 GaussDB: m_db=# SELECT (1,2) <=> row(2,3); ERROR: could not determine interpretation of row comparison operator <=> LINE 1: select (1,2) <=> row(2,3);

序号	MySQL数据库	GaussDB数 据库	差异
			ERROR 1241 (21000): Operand should contain 2 column(s) mysql> SELECT (1,2) <> NULL; ERROR 1241 (21000): Operand should contain 2 column(s) mysql> SELECT (1, 2) IS NULL; ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT ISNULL((1, 2)); ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT ISNULL((1, 2)); ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT NULL BETWEEN NULL AND ROW(2,2); ERROR 1241 (21000): Operand should contain 1 column(s) mysql> SELECT ROW(NULL) AS x; ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ') as x' at line 1
4		支持。	MySQL表示对一个操作数进行两次取 反,结果等于原操作数;GaussDB表示 注释。
5	!!	支持,但有 差异。	MySQL: !!含义同!,表示取非。 GaussDB: !表示取非操作,当!与!中间存在空格时,表示连续两次取非(!!);当!与!中间没有空格时,表示阶乘(!!)。 说明 • GaussDB中,当同时使用阶乘(!!)和取非(!)时,阶乘(!!)和取非(!)中间需要添加空格,否则会报错。 • GaussDB中,当需要多次取非操作时,!与!之间需使用空格隔开。

序号	MySQL数据库	GaussDB数 据库	差异
6	[NOT] REGEXP	支持,但有 差异。	• GaussDB和MySQL在正则表达式中支持的元字符有所不同。例如,GaussDB支持"\d"表示数字,"\w"表示字母、数字和下划线,"\s"表示空格,而MySQL不支持这些元字符,MySQL会把这些字符当成正常字符串。
			● GaussDB '\b' 可以与'\\b'匹配, MySQL匹配失败。
			● GaussDB新框架下中使用"\"表示转 义字符,而MySQL中使用"\\"。
			● MySQL不支持2个操作符连在一起使 用。
			 模式字符串pat非法入参,只存在右单括号')'时,GaussDB数据库报错。 MySQL存在bug,后续版本已经修复此问题。
			 在de abc匹配序列de或abc的匹配规则,当 左右存在空值时,MySQL存在bug,会报错,后续版本已经修复此问题。
			空白字符[\t]正则匹配字符类 [:blank:], GaussDB可匹配, MySQL \t不能匹配[:blank:], MySQL存在 bug,后续版本已经修复此问题。
			● GaussDB支持非贪婪模式匹配,即尽可能少的匹配字符,在部分特殊字符后加'?'问号字符,例如:"??,*?,+?,{n}?,{n,}?,{n,m}?"。MySQL5.7版本不支持非贪婪模式匹配,并报错:Got error 'repetition-operator operand invalid' from regexp。MySQL 8.0版本已经支持。
			 在binary字符集下,text类型、blob 类型均会转换成bytea类型,由于 REGEXP操作符不支持bytea类型,因 此无法匹配。
7	LIKE	支持,但有 差异。	MySQL: LIKE的左操作数只能是位运算或者算术运算或者由括号组成的表达式,LIKE的右操作数只能是单目运算符(不含NOT)或者括号组成的表达式。 GaussDB: LIKE的左右操作数可以是任意表达式。

序号	MySQL数据库	GaussDB数 据库	差异
8	[NOT] BETWEEN AND	支持,但有 差异。	MySQL: [NOT] BETWEEN AND嵌套使用时从右到左结合。[NOT] BETWEEN AND的第1个操作数和第2个操作数只能是位运算或者算术运算或者由括号组成的表达式。
			GaussDB: [NOT] BETWEEN AND嵌套使用时从左到右结合。[NOT] BETWEEN AND的第1个操作数和第2个操作数可以是任意表达式。
9	IN	支持,但有	MySQL: IN的左操作数只能是位运算或者算术运算或者由括号组成的表达式。GaussDB: IN的左操作数可以是任意表达式。不支持ROW IN (ROW,ROW)形式的查询。在开启精度传递的场景下,对表中的数据使用in操作符时,若表中数据为FLOAT类型、DOUBLE类型且包括相应精度和标度(如float(4,2),double(4,2))),GaussDB会根据精度和标度对值进行比较,MySQL会读取内存值(失真数值,导致比较不相等)。GaussDB: m_db=# CREATE TABLE test1(t_float float(4,2));CREATE TABLE m_db=# INSERT INTO test1 VALUES(1.42),(2.42); INSERT 0.2
			m_db=# SELECT t_float, t_float in (1.42,2.42) FROM test1; t_float ?column?
10	!	支持,但有 差异。	MySQL:!的操作数只能是单目运算符 (不含not)或者括号组成的表达式。 GaussDB:!的操作数可以是任意表达 式。

序号	MySQL数据库	GaussDB数 据库	差异
11	#	不支持。	MySQL支持#注释,GaussDB不支持#注 释。
12	BINARY	支持,但有 差异。	GaussDB中支持的表达式和MySQL并不完全一致(包括一些函数、操作符等)。GaussDB独有的表达式'~'、'IS DISTINCT FROM'等,由于BINARY关键字优先级更高,使用BINARY expr会优先将BINARY与'~'、'IS DISTINCT FROM'的左参数合并,导致报错。

序号 MySQL数据	库 GaussDB数 据库	差异
序号 MySQL数据 13 取负(-)		取反结果类型和精度与MySQL不一致: CREATE TABLE t as select1; MySQL表字段返回类型为: decimal(2,0)。 GaussDB表字段返回类型为: integer(1)。 在开启精度传递 (
		MySQL:

序号	MySQL数据库	GaussDB数 据库	差异
序	MySQL数据库		mysql> DROP TABLE IF EXISTS test; Query OK, 0 rows affected (0.01 sec) mysql> CREATE TABLE test as
14	/**/	不支持。	2 rows in set (0.00 sec) GaussDB中语句中不支持/**/注释。
		<u> </u>	, ,

序号	MySQL数据库	GaussDB数 据库	差异
15	xor	支持,存在 差异。	GaussDB的xor和MySQL的行为不同。 GaussDB 优化器会有常数优化,会出现 先计算表示结果是常数的情况。
			GaussDB: m_db=# SELECT 1 xor null xor pow(200, 2000000) FROM dual; ERROR: value out of range: overflow m_db=# CREATE TABLE t1 (a int, b int); CREATE TABLE m_db=# INSERT INTO t1 VALUES(2,2), (200, 2000000000); INSERT 0 2 m_db=# m_db=# m_db=# m_db=# SELECT 1 xor null xor pow(a, b) FROM t1; ?column?
			(2 rows)
			MySQL: mysql> SELECT 1 xor null xor pow(200, 2000000) FROM dual; +
16	IS NULL和IS	支持,但有	MySQL的优先级小于逻辑运算符,
	NOT NULL	差异。	GaussDB的优先级大于逻辑运算符。

序号	MySQL数据库	GaussDB数 据库	差异
17	XOR、 、&、 <、>、<=、 >=、=、!=	支持,但执 行机制存在 差异。	MySQL执行机制为执行左操作数后,对 结果进行判断是否为空,进而决定是否 需要执行右操作数。
			GaussDB执行机制是执行左右操作数 后,对结果再进行判断是否为空。
			当左操作数结果为空,右操作数执行报 错时,MySQL不会报错直接返回, GaussDB会执行报错。
			MySQL行为: mysql> SELECT version();
			++ version()
			++ 5.7.44-debug-log ++
			1 row in set (0.00 sec)
			mysql> DROP TABLE IF EXISTS data_type_table; Query OK, 0 rows affected (0.02 sec)
			mysql> CREATE TABLE data_type_table (-> MyBool BOOL, -> MyBinary BINARY(10), -> MyYear YEAR ->);
			Query OK, 0 rows affected (0.02 sec) mysql> INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021'); Query OK, 1 row affected (0.00 sec)
			mysql> SELECT (MyBool % MyBinary) (MyBool - MyYear) FROM data_type_table;
			(MyBool % MyBinary) (MyBool - MyYear)
			NULL
			1 row in set, 2 warnings (0.00 sec) GaussDB行为:
			m_db=# DROP TABLE IF EXISTS data_type_table; DROP TABLE m_db=# CREATE TABLE data_type_table (m_db(# MyBool BOOL, m_db(# MyBinary BINARY(10), m_db(# MyYear YEAR m_db(#); CREATE TABLE m_db=# INSERT INTO data_type_table VALUES (TRUE, 0x1234567890, '2021'); INSERT 0 1 m_db=# SELECT (MyBool % MyBinary) (MyBool - MyYear) FROM data_type_table;
			WARNING: Truncated incorrect double value: '4Vx ' CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear) WARNING: division by zero CONTEXT: referenced column: (MyBool %
			MyBinary) (MyBool - MyYear) ERROR: Bigint is out of range.

序号	MySQL数据库	GaussDB数 据库	差异
			CONTEXT: referenced column: (MyBool % MyBinary) (MyBool - MyYear)
18	-	支持,但有差异。	CREATE VIEW AS SELECT算数操作符('+', '-', '*', '/', '%', 'mod', 'div')内嵌b''常量时,MySQL 5.7返回类型可能存在unsigned标识,GaussDB不会。 MySQL输出结果: mysql> CREATE VIEW v22 as SELECT b'101' / b'101' c22; Query OK, 0 rows affected (0.00 sec) mysql> DESC v22; ++ Field Type

表 4-48 操作符组合存在差异

操作符组合示例	MySQL 数据库	GaussD B数据库	说明
SELECT 1 LIKE 3 & 1;	不支持	支持	LIKE的右操作数不能是位运算符组 成的表达式。
SELECT 1 LIKE 1 +1;	不支持	支持	LIKE的右操作数不能是算术运算符 组成的表达式。
SELECT 1 LIKE NOT 0;	不支持	支持	LIKE的右操作数只能是+、-、!等 单目操作符或者括号组成的表达 式,NOT除外。
SELECT 1 BETWEEN 1 AND 2 BETWEEN 2 AND 3;	从右到 左结合	从左到 右结合	建议加上括号明确优先级。
SELECT 2 BETWEEN 1=1 AND 3;	不支持	支持	BETWEEN的第2个操作数不能是 比较操作符组成的表达式。

操作符组合示例	MySQL 数据库	GaussD B数据库	说明
SELECT 0 LIKE 0 BETWEEN 1 AND 2;	不支持	支持	BETWEEN的第1个操作数不能是 模式匹配操作符组成的表达式。
SELECT 1 IN (1) BETWEEN 0 AND 3;	不支持	支持	BETWEEN的第1个操作数不能是 IN操作符组成的表达式。
SELECT 1 IN (1) IN (1);	不支持	支持	第2个IN表达式左操作数不能是IN 组成的表达式。
SELECT! NOT 1;	不支持	支持	!的操作数只能是+、-、!等单目操 作符或者括号组成的表达式, NOT除外。

□说明

在GaussDB中支持,但在MySQL中不支持的操作符组合不建议使用。建议按照MySQL中的使用规则对操作符进行组合使用。

索引差异

- GaussDB当前仅支持UBTree和B-tree索引。
- 针对模糊匹配(LIKE操作符), MySQL创建默认索引可以走索引; GaussDB默认的索引不走索引,需要用户使用以下语法指定opclass,比如指定为text_pattern_ops,LIKE操作符才可以走索引。
 CREATE INDEX indexname ON tablename(col [opclass]);
- B-tree/UBTree索引场景保持原生GaussDB原有逻辑,即同一操作符族内的类型比较,支持索引扫描,其余索引类型暂未支持。
- 通过GaussDB JDBC连接数据库,GaussDB YEAR类型在含有绑定参数的PBE场景下索引无法使用。
- where子句中,索引字段类型和常量类型操作场景下,GaussDB中索引与MySQL索引支持存在差异,如下表所示。例如以下语句GaussDB不支持索引:

create table t(_int int);

create index idx on t(_int) using BTREE;

select * from t where _int > 2.0;

□说明

where子句里索引字段类型和常量类型操作场景中,可以使用cast函数将常数类型显示转为字段类型,以便实现索引。

select * from t where _int > cast(2.0 as signed);

表 4-49 索引支持存在差异

索引字段类型	常量类型	GaussDB	MySQL
整型	整型	是	是
浮点型	浮点型	是	是
定点型	定点型	是	是

索引字段类型	常量类型	GaussDB	MySQL
字符串类型	字符串类型	是	是
二进制类型	二进制类型	是	是
带日期的时间类 型	带日期的时间类 型	是	是
TIME类型	TIME类型	是	是
带日期的时间类 型	可转为带日期的 时间类型(如 20231130等整 型)	是	是
带日期的时间类 型	TIME类型	是	是
TIME类型	可转为TIME类型 的常量(如 203008等整型)	是	是
浮点型	整型	是	是
浮点型	定点型	是	是
浮点型	字符串类型	是	是
浮点型	二进制类型	是	是
浮点型	带日期的时间类 型	是	是
浮点型	TIME类型	是	是
定点型	整型	是	是
字符串类型	带日期的时间类 型	是	否
字符串类型	TIME类型	是	否
二进制类型	字符串类型	是	是
二进制类型	带日期的时间类 型	是	否
二进制类型	TIME类型	是	否
整型	浮点型	否	是
整型	定点型	否	是
整型	字符串类型	否	是
整型	二进制类型	否	是

索引字段类型	常量类型	GaussDB	MySQL
整型	带日期的时间类 型	否	是
整型	TIME类型	否	是
定点型	浮点型	否	是
定点型	字符串类型	否	是
定点型	二进制类型	否	是
定点型	带日期的时间类 型	否	是
定点型	TIME类型	否	是
字符串类型	二进制类型	否	是
带日期的时间类 型	整型(不可转为 带日期的时间类 型)	否	是
带日期的时间类 型	浮点型(不可转 为带日期的时间 类型)	否	是
带日期的时间类 型	定点型(不可转 为带日期的时间 类型)	否	是
TIME类型	整型(不可转为 TIME类型)	否	是
TIME类型	字符串类型(不可转为TIME类型)	否	是
TIME类型	二进制类型(不可转为TIME类型)	否	是
TIME类型	带日期的时间类 型	否	是
YEAR类型	YEAR类型	是	是
YEAR类型	可转为YEAR类型 的常量(如2034 等整型)	是	是
BIT类型	BIT类型	否	是
SET/ENUM类型	字符串类型	否	是
SET/ENUM类型	整型	否	是
SET/ENUM类型	浮点型	否	是

索引字段类型	常量类型	GaussDB	MySQL
SET/ENUM类型	时间类型	否	是

表 4-50 是否支持走索引

索引字段类型	常量类型	是否走索引	MySQL
字符串类型	整型	否	否
字符串类型	浮点型	否	否
字符串类型	定点型	否	否
二进制类型	整型	否	否
二进制类型	浮点型	否	否
二进制类型	定点型	否	否
带日期的时间类 型	字符串类型(不 可转为带日期的 时间类型)	否	否
带日期的时间类 型	二进制类型(不 可转为带日期的 时间类型)	否	否
TIME类型	浮点型(不可转 为TIME类型)	否	否
TIME类型	定点型(不可转 为TIME类型)	否	否
YEAR类型	不可转为YEAR类 型的常量	否	否
BIT类型	字符串类型	否	否

4.3.5 字符集

GaussDB数据库支持指定数据库、模式、表或列的字符集,支持的范围如下。

表 4-51 字符集列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4	支持
2	utf8	支持
3	gbk	支持
4	gb18030	支持

序号	MySQL数据库	GaussDB数据库
5	binary	支持

山 说明

- utf8和utf8mb4在GaussDB中为同一个字符集,编码最大长度为4字节。当前字符串字符集为utf8,指定其字符序为utf8mb4_bin/utf8mb4_general_ci/utf8mb4_unicode_ci/utf8mb4_0900_ai_ci时(例如select_utf8'a' collate utf8mb4_bin), MySQL中会发生报错,GaussDB不报错。当字符串字符集为utf8mb4,指定其字符序为utf8_bin/utf8_general_ci/utf8_unicode_ci时,存在同样差异。
- 词法语法解析按照字节流解析,当多字节字符中包含与'\', '\'', '\\'等符号一致的编码时,会导致与mysql行为不一致,建议暂时关闭转义符开关进行规避。
- 目前GaussDB对不属于当前字符集的非法字符未执行严格的编码逻辑校验,可能导致此类非 法字符成功输入。而MySQL会校验报错。

4.3.6 排序规则

GaussDB数据库支持指定模式、表或列的排序规则,支持的范围如下。

□ 说明

排序规则差异说明:

- 当前仅有字符串类型、部分二进制类型支持指定排序规则,其他类型不支持指定排序规则,可以通过查询pg_type系统表中类型的typcollation属性不为0来判断该类型支持字符序。
 MySQL中所有类型可以指定字符序,但除字符串、二进制类型其他排序规则无实际意义。
- 当前排序规则(除binary外)仅支持在其对应字符集与库级字符集一致时可以指定, GaussDB数据库中,字符集必须与数据库的字符集一致,且不支持表内多种字符集混合使用。
- utf8mb4字符集下默认字符序为utf8mb4_general_ci,与MySQL5.7保持一致。
- 使用latin1字符序需要设置兼容性参数m_format_dev_version = 's2'。

表 4-52 排序规则列表

序号	MySQL数据库	GaussDB数据库
1	utf8mb4_general_ci	支持
2	utf8mb4_unicode_ci	支持
3	utf8mb4_bin	支持
4	gbk_chinese_ci	支持
5	gbk_bin	支持
6	gb18030_chinese_ci	支持
7	gb18030_bin	支持
8	binary	支持
9	utf8mb4_0900_ai_ci	支持

序号	MySQL数据库	GaussDB数据库
10	utf8_general_ci	支持
11	utf8_bin	支持
12	utf8_unicode_ci	支持
13	latin1_swedish_ci	支持
14	latin1_bin	支持

4.3.7 事务

GaussDB数据库兼容MySQL的事务,但存在部分差异。本章节介绍GaussDB的M-Compatibility数据库中事务相关的差异。

事务默认隔离级别

M-Compatibility默认隔离级别为READ COMMITTED,MySQL默认隔离级别为REPEATABLE-READ。

-- 查看当前事务隔离级别。 m_db=# SHOW transaction_isolation;

子事务

M-Compatibility中,通过SAVEPOINT用于在当前事务里建立一个新的保存点(子事务),使用ROLLBACK TO SAVEPOINT回滚到一个保存点(子事务),子事务回滚后父事务可以继续运行,子事务的回滚不影响父事务的事务状态。

MySQL不存在创建保存点(子事务)。

嵌套事务

嵌套事务指在事务块中开启新事务。

M-Compatibility中,正常事务块中开启新事务会警告存在一个进行中的事务,忽略开启命令;异常事务块中开启新事物将报错,必须在执行ROLLBACK/COMMIT之后才可以执行,执行ROLLBACK/COMMIT会回滚之前语句。

MySQL中,正常事务块中开启新事务会先把之前事务提交,然后开启新事务; 异常事务块中开启新事务会忽略错误,提交之前无错误的语句并开启新事务。

```
-- M-Compatibility正常事务块中,开启新事务会警告并忽略。
m_db=# DROP TABLE IF EXISTS test_t;
m_db=# CREATE TABLE test_t(a int, b int);
m_db=# BEGIN;
m_db=# BEGIN;
m_db=# BEGIN; -- 会警告there is already a transaction in progress。
m_db=# SELECT * FROM test_t ORDER BY 1;
m_db=# COMMIT;
-- M-Compatibility异常事务块中,开启新事务会报错,必须ROLLBACK/COMMIT之后才可以执行。
m_db=# BEGIN;
m_db=# ERROR sql; -- 错误语句。
m_db=# BEGIN; -- 报错。
m_db=# COMMIT; -- ROLLBACK/COMMIT之后才可以执行。
```

隐式提交的语句

M-Compatibility使用GaussDB存储,继承GaussDB事务机制,事务中执行DDL、DCL不会自动提交。

MySQL在DDL、DCL、管理类语句,锁相关语句会自动提交。

-- M-Compatibility创建表和设置GUC参数可以回滚掉。

m_db=# DROP TABLE IF EXISTS test_table_rollback;

m_db=# BEGIN;

m_db=# CREATE TABLE test_table_rollback(a int, b int);

m_db=# \d test_table_rollback;

m_db=# ROLLBACK;

m_db=# \d test_table_rollback; -- 不存在该表。

SET TRANSACTION 差异

M-Compatibility中,SET TRANSACTION同时设置多次隔离级别/事务访问模式时,只有最后一个会生效;多个事务特性支持使用空格和逗号分隔。

MySQL中SET TRANSACTION不允许设置多次隔离级别/事务访问模式;多个事务特性只支持使用逗号分隔。

表 4-53 SET TRANSACTION 差异

序号	语法	功能	差异
1	SET TRANSACTIO N	设置事务特性。	M-Compatibility中,SET TRANSACTION在会话级别生效; MySQL中SET TRANSACTION在下一个 事务生效。
2	SET SESSION TRANSACTIO N	设置会话级事务特 性。	-
3	SET GLOBAL TRANSACTIO N	设置全局会话级事 务特性,该特性适 用于后续会话,对 当前会话无影响。	M-Compatibility中,GLOBAL是全局会 话级别生效,只针对当前数据库实例, 其它数据库不影响。 MySQL中,会使所有数据库生效。

⁻⁻ SET TRANSACTION会话级生效。

m_db=# SET TRANSACTION ISOLATION LEVEL READ COMMITTED READ WRITE;

m_db=# SHOW transaction_isolation;

m_db=# SHOW transaction_read_only;

-- M-Compatibility同时设置多次隔离级别/事务访问模式,最后一个生效。

m_db=# SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED, ISOLATION LEVEL

REPEATABLE READ, READ WRITE, READ ONLY;

m_db=# SHOW transaction_isolation; -- repeatable read

m_db=# SHOW transaction_read_only; -- on

START TRANSACTION 差异

M-Compatibility中,START TRANSACTION开启事务时,同时支持设置隔离级别;同时设置多次隔离级别/事务访问模式时,只有最后一个会生效;当前版本不支持立即开启一致性快照;多个事务特性支持空格和逗号分隔。

MySQL的start transaction 开启事务时,不支持设置隔离级别,不支持设置多次事务访问模式;多个事务特性只支持逗号分隔。

-- 开启事务设置隔离级别。

m_db=# START TRANSACTION ISOLATION LEVEL READ COMMITTED;

m_db=# COMMIT;

-- 多次设置访问模式。

m_db=# START TRANSACTION READ ONLY, READ WRITE;

m_db=# COMMIT;

事务相关的 GUC 参数

表 4-54 事务相关的 GUC 参数差异

序号	GUC参数	功能	差异
1	autocomm it	设置事务自 动提交模 式。	-

序号	GUC参数	功能	差异
2	transactio n_isolation	在GaussDB 中事级 MySQL事级 在是没多别。 在是没多别。 在是没多别。	 GaussDB中,通过使用SET命令,只能改变当前事务的隔离级别。如果想要改变会话级的隔离级别,可以使用default_transaction_isolation。在MySQL中,通过使用SET命令,可以改变会话级的事务隔离级别。 支持范围差异。 MySQL中,支持以下隔离级别设置,对大小写不敏感,对空格敏感: READ-COMMITTED READ-UNCOMMITTED REPEATABLE-READ SERIALIZABLE GaussDB中,支持以下隔离级别设置,对大小写和空格敏感: read committed read uncommitted read uncommitted read uncommitted refault (设置和会话中默认隔离级别一样) 设置m_format_dev_version = 's2'时,支持MySQL的隔离级别设置。 在GaussDB中,新事务的transaction_isolation值将被初始化为default_transaction_isolation的值。 设置m_format_dev_version = 's2'时:
3	tx_isolatio n	设置事务的 隔离级别; tx_isolation 和 transaction _isolation是 同义词。	GaussDB中只支持查询,不支持修改。

序号	GUC参数	功能	差异
4	default_tr ansaction_ isolation	设置事务的 隔离级别。	GaussDB中通过SET设置会改变会话级事务隔离级别。 MySQL中不支持该系统参数。
5	transactio n_read_on ly	在GaussDB 中事等式。 在MySQL中是级为问模式。 在MySQL中是多数,问模式。	 在GaussDB中,通过使用SET命令,只能改变当前事务的访问模式。如果想要改变会话级的访问模式,可以使用default_transaction_read_only。在MySQL中,通过使用SET命令,可以改变会话级的事务隔离级别。 在GaussDB中,新事务的transaction_read_only值将被初始化为default_transaction_read_only的值。 设置m_format_dev_version = 's2'时: 以下语句是设置下一个事务特性: set@@transaction_read_only = value; set transaction {read write read only}. 以下语句修改会话级事务特性: set [local session @@session.] transaction_read_only = value. 下一个事务特性不允许在事务内使用,如果隐式事务报错,即单个SQL语句报错,继续保持下一个事务特性。
6	tx_read_o nly	设置事务的 访问模式。 tx_read_onl y和 transaction _read_only 是同义词。	GaussDB中只支持查询,不支持修改。
7	default_tr ansaction_ read_only	设置事务的 访问模式。	GaussDB中通过SET设置会改变会话级事务访问 模式;MySQL中不支持该系统参数。

4.3.8 SQL

GaussDB数据库兼容绝大多数MySQL语法,但存在部分差异。本章节介绍GaussDB数据库当前支持的MySQL语法。

● 部分关键字在MySQL可以做标识符但M-Compatibility不可以或存在限制,以下为表4-55。

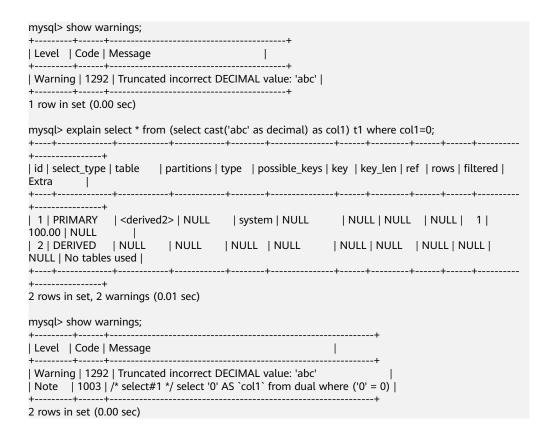
表 4-55 限制做标识符列表

关键字类型	关键字	约束
保留(可以是类型或函数)	COLLATION COMPACT	除函数和变量,不可以 作为其他数据库标识 符。
非保留(不能是类型或 函数)	BIT、BOOLEAN、 COALESCE、DATE、 NATIONAL、NCHAR、 NONE、NUMBER、 TEXT、TIME、 TIMESTAMP、 TIMESTAMPDIFF	不可以作为函数或变量 的标识符。
保留	ANY、ARRAY、 BUCKETS、DO、END、 LESS、MODIFY、 OFFSET、ONLY、 RETURNING、SOME、 USER	不可以作为任意数据库 标识符。

GaussDB优化器与MySQL的优化器存在差异,由于优化器生成的执行计划的差异,可能导致GaussDB行为与MySQL行为的不一致,不影响业务数据结果。
 例如优化器对SQL语句做常数优化,会提前计算常量表达式,如xor操作符。
 例如以下场景,GaussDB在计算col1以及使用col1进行where比较时,均会调用cast函数,产生两条WARNING记录。

MySQL在计算col1时会调用cast函数,在进行where比较时,使用计算好的值直接进行比较,因而产生一条WARNING记录。

```
-- GaussDB行为:
m_db=# select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
WARNING: Truncated incorrect DECIMAL value: 'abc'
WARNING: Truncated incorrect DECIMAL value: 'abc'
CONTEXT: referenced column: col1
col1
  0
(1 row)
m db=# explain verbose select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
WARNING: Truncated incorrect DECIMAL value: 'abc'
WARNING: Truncated incorrect DECIMAL value: 'abc'
CONTEXT: referenced column: col1
          QUERY PLAN
Result (cost=0.00..0.01 rows=1 width=0)
 Output: 0::decimal
(2 rows)
-- MySQL行为:
mysql> select * from (select cast('abc' as decimal) as col1) t1 where col1=0;
| col1 |
0 |
1 row in set, 1 warning (0.00 sec)
```



4.3.8.1 关键字

约束差异:

- 当关键字在M-Compatibility模式下为保留关键字,在MySQL中为非保留关键字, 其差异为:在M-Compatibility模式下不可作为表名、列名、列别名、AS列别名、 AS表别名、表别名、函数名和变量名,在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字,在MySQL中为保留关键字, 其差异为:在M-Compatibility模式下可作为表名、列名、列别名、AS列别名、AS 表别名、表别名、函数名和变量名,在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字(可以是函数或类型),在 MySQL中为保留关键字,其差异为:在M-Compatibility模式下可作为列别名、 AS列别名、函数名和变量名,在MySQL中不支持。
- 当关键字在M-Compatibility模式下为保留关键字(可以是函数或类型),在
 MySQL中为非保留关键字,其差异为:在M-Compatibility模式下不可作为表名、列名、AS表别名和表别名,在MySQL中支持。
- 当关键字在M-Compatibility模式下为非保留关键字(不能是函数或类型),在 MySQL中为保留关键字,其差异为:在M-Compatibility模式下可作为表名、列 名、列别名、AS列别名、AS表别名、表别名和变量名,在MySQL中不支持。
- 当关键字在M-Compatibility模式下为非保留关键字(不能是函数或类型),在 MySQL中为非保留关键字,其差异为:在M-Compatibility模式下不可作为函数 名,在MySQL中支持。

□ 说明

在M-Compatibility模式下的非保留关键字、保留关键字(可以是函数或类型)以及非保留 关键字(不能是函数或类型)之中,以下关键字不能作为列别名进行使用:

BETWEEN, BIGINT, BLOB, CHAR, CHARACTER, CROSS, DEC, DECIMAL, DIV, DOUBLE, EXISTS, FLOAT, FLOAT4, FLOAT8, GROUPING, INNER, INOUT, INT, INT1, INT2, INT3, INT4, INT8, INTEGER, JOIN, LEFT, LIKE, LONGBLOB, LONGTEXT, MEDIUMBLOB, MEDIUMINT, MEDIUMTEXT, MOD, NATURAL, NUMERIC, OUT, OUTER, PRECISION, REAL, RIGHT, ROW, ROW_NUMBER, SIGNED, SMALLINT, SOUNDS, TINYBLOB, TINYINT, TINYTEXT, VALUES, VARCHAR, VARYING, WITHOUT

其中,SIGNED和WITHOUT在MySQL中可以作为列别名进行使用。

4.3.8.2 标识符

M-Compatibility模式下标识符存在以下差异:

- GaussDB无引号标识符中不支持以美元符号(\$)开头, MySQL无引号标识符中 支持。
- GaussDB无引号标识符中的支持大小写敏感的数据库对象。
- GaussDB标识符支持U+0080~U+00FF扩展字符,MySQL标识符支持U+0080~U+FFFF的扩展字符。
- 无引号标识符中,GaussDB不支持创建以数字开头包含一个e或E结尾作为标识符的表,例如:

有引号标识符中,GaussDB对于创建了列名为纯数字或科学计算法的表,不支持直接使用,需要在引号中使用;对于点操作符(.)场景,列名为纯数字或科学计算法的表也需要在引号中使用。例如:

```
-- 创建列名为纯数字或科学计算法的表
m_db=# CREATE TABLE t1('123' int, '1e3' int, '1e' int);
CREATE TABLE
-- 向表中插入数据
m_db=# INSERT INTO t1 VALUES(7, 8, 9);
INSERT 0 1
-- 结果非预期,但与MySQL结果一致
m_db=# SELECT 123 FROM t1;
?column?
   123
(1 row)
-- 结果非预期,但与MySQL结果一致
m_db=# SELECT 1e3 FROM t1;
?column?
  1000
(1 row)
-- 结果非预期,并且与MySQL结果不一致
m_db=# SELECT 1e FROM t1;
```

```
1
(1 row)
-- 正确用法
m_db=# SELECT `123` FROM t1;
123
 7
(1 row)
m_db=# SELECT `1e3` FROM t1;
1e3
 8
(1 row)
m_db=# SELECT `1e` FROM t1;
1e
9
(1 row)
-- 点操作符的场景,GaussDB不支持,MySQL支持
m_db=# SELECT t1.123 FROM t1;
ERROR: syntax error at or near ".123"
LINE 1: SELECT t1.123 FROM t1;
m_db=# SELECT t1.1e3 FROM t1;
ERROR: syntax error at or near "1e3"
LINE 1: SELECT t1.1e3 FROM t1;
m_db=# SELECT t1.1e FROM t1;
ERROR: syntax error at or near "1"
LINE 1: SELECT t1.1e FROM t1;
-- 点操作符的场景,正确用法:
m_db=# SELECT t1.`123` FROM t1;
123
7
(1 row)
m_db=# SELECT t1.`1e3` FROM t1;
1e3
 8
(1 row)
m_db=# SELECT t1.`1e` FROM t1;
1e
9
(1 row)
m_db=# DROP TABLE t1;
DROP TABLE
```

- GaussDB分区名使用双引号(需要设置SQL_MODE为ANSI_QUOTES)或反引号 是区分大小写的,MySQL不区分。
- MySQL标识符长度限制为64,而GaussDB标识符长度限制为63。
- GaussDB不支持可执行注释。

4.3.8.3 DDL

表 4-56 DDL 语法兼容介绍

概述	详细语法说明	差异
建表和修改表时支持创 建主键、UNIQUE索 引、外键约束	ALTER TABLE CREATE TABLE	• 在GaussDB中,当约束关联的表为ustore,且SQL语句中指定为using btree时,底层会建立为ubtree。
		● 在GaussDB中,允许将外键 作为分区键。
		● 索引名、约束名、key名 GaussDB是SCHEMA下唯 一,MySQL是表下唯一。
		MySQL和GaussDB主键上 支持列的个数上限不同。

概述	详细语法说明	差异
支持自增列	ALTER TABLE、CREATE TABLE	● GaussDB的自动增长列建议 为索引的第一个字段,否则 建表时产生警告,MySQL 自动增长列必须为索引第一 个字段,否则建表时会报 错。GaussDB含有自动增长 列的表进行某些操作时会产 生错误,例如:ALTER TABLE EXCHANGE PARTITION。
		● GaussDB的 AUTO_INCREMENT = value语法,value必须为小 于2^127的正数。MySQL可 以为0,GaussDB不可以。
		● GaussDB中当自增值已经达到字段数据类型的最大值时,继续自增将产生错误。 MySQL有些场景产生错误或警告,有些场景仍自增为最大值。
		 不支持 innodb_autoinc_lock_mod e系统变量, GaussDB的 GUC参数 auto_increment_cache=0
		时,批量插入自动增长列的 行为与MySQL系统变量 innodb_autoinc_lock_mod e=1相似。
		● GaussDB的自动增长列在导入数据或者进行Batch Insert执行计划的插入操作时,对于混合0、NULL和确定值的场景,如果产生错误,后续插入自增值不一定与MySQL完全一致。
		- 提供 auto_increment_cache 参数,可以控制预留自 增值的数量。
		GaussDB的并行导入或插入自动增长列触发自增时,每个并行线程预留的缓存值也只在其线程中使用,未完全使用完毕的话,也会出现表中自动增长列的值不连续的情况。并行插入产生的自增

概述	详细语法说明	差	异
			值结果无法保证与MySQL 完全一致。
		•	GaussDB的本地临时表中的自动增长列批量插入时不会预留自增值,正常场景不会产生不连续的自增值。 MySQL临时表与普通表中的自动增长列自增结果一致。
		•	GaussDB的SERIAL数据类型为原有的自增列,与AUTO_INCREMENT自增列有差异。MySQL的SERIAL数据类型就是AUTO_INCREMENT自增列。
		•	GaussDB的不允许 auto_increment_offset的值 大于 auto_increment_increment 的值,会产生错误。 MySQL允许,并说明 auto_increment_offset会被 忽略。
		•	在表有主键或索引的情况下,ALTER TABLE命令重写表数据的顺序与MySQL不一定相同,GaussDB按表数据存储顺序重写,MySQL会按主键或索引顺序重写,导致自增值的顺序可能不同。
		•	GaussDB的ALTER TABLE命 令添加或修改自增列时,第 一次预留自增值的数量是表 统计信息中的行数,统计信 息的行数不一定与MySQL 一致。
		•	GaussDB的last_insert_id函 数返回值为128位的整型。
		•	GaussDB在触发器或用户自 定义函数中自增时,刷新 last_insert_id返回值。 MySQL不刷新。
		•	GaussDB的对GUC参数 auto_increment_offset和 auto_increment_increment

概述	详细语法说明	差异
		设置超出范围的值会产生错 误。MySQL会自动改为边 界值。
		● sql_mode设置 no_auto_value_on_zero参数,表定义的自动增长列为非NOT NULL约束,向表中插入数据不指定自动增长列的值时,GaussDB中自动增长列插入NULL值,且不触发自增;MySQL中自动增长列插入NULL值,触发自增。
支持前缀索引	CREATE INDEX、ALTER TABLE、CREATE TABLE	• GaussDB中前缀长度不得超过2676,键值的实际长度受内部页面限制,若字段中含有多字节字符或者一个索引上有多个键,索引行长度可能会超限报错。
		GaussDB中主键索引中不支 持前缀键,创建或添加主键 时不支持指定前缀长度。
支持指定字符集与排序 规则	ALTER SCHEMA、ALTER TABLE、CREATE SCHEMA、CREATE TABLE	• 指定库级字符集时,除 BINARY字符集外,暂不支 持创建新库/模式的字符集 与数据库的 server_encoding不同。
		● 指定表级、列级字符集和字符序时,MySQL支持指定与库级字符集、字符序不同的字符集和字符序。在GaussDB中,表级、列级字符集和字符序仅支持BINARY字符集、字符序或者与库级字符集、字符序相同的字符集、字符序。
		若重复指定字符集或字符 序,仅最后一个生效。
修改表时支持在表第一 列前面或者在指定列后 面添加列	ALTER TABLE	-

概述	详细语法说明	差异
修改列名称/定义语法	ALTER TABLE	 暂不支持DROP INDEX DROP KEY ORDER BY子 项。 ALTER TABLE新增列时, MySQL指定字段为NOT NULL时,会将NULL值转为 对应类型的默认值插入该 列,GaussDB会检查NULL
(A)(本)(D) 本语计	CDEATE TABLE	值。
创建分区表语法	CREATE TABLE PARTITION CREATE TABLE SUBPARTITION	● MySQL在以下场景支持表达式,不支持多个分区键: - 使用LIST分区/RANGE分区策略,不指定COLUMNS关键字。 - 使用HASH分区策略。 ● MySQL在以下场景不支持表达式,支持多个分区键: - 使用LIST分区/RANGE分区策略,指定COLUMNS关键字。 - 使用KEY分区策略。 ● GaussDB不支持使用表达式作为分区键。 ● GaussDB仅在以下场景支持使用S个分区键:使用LIST分区/RANGE分区策略,且不指定二级分区。 ● GaussDB分区表不支持用虚拟生成列作为分区键。
建表和修改表时支持指 定表级和列级 comment	CREATE TABLE、ALTER TABLE	-
创建索引时支持指定索 引级comment	CREATE INDEX	-

概述	详细语法说明	差异
交换普通表和分区表分区的数据	详细语法说明 ALTER TABLE PARTITION	ALTER TABLE EXCHANGE PARTITION的差异点: • 对于自增列,MySQL执行 alter exchange partition 后,自增列会被重重重增。 • MySQL表或会的则照明的。 • MySQL表或合的,是是有一个人,是是是一个人,是一个人,

概述	详细语法说明	差异
修改分区表的分区键信 息	ALTER TABLE	MySQL支持修改分区表的分区 键信息,GaussDB中不支持。
支持CREATE TABLE LIKE语法	CREATE TABLE LIKE	• 在MySQL 8.0.16 之前的版本中,CHECK约束会被语法解析但功能会被忽略,表现为不复制CHECK约束,GaussDB支持复制CHECK约束。
		对于主键约束名称,在建表时,MySQL所有主键约束名称固定为PRIMARYKEY,GaussDB不支持复制。
		● 对于唯一键约束名称,在建 表时,MySQL支持复制, GaussDB不支持复制。
		 对于CHECK约束名称,在建 表时, MySQL 8.0.16 之前 的版本无CHECK约束信息, GaussDB支持复制。
		对于索引名称,在建表时, MySQL支持复制, GaussDB不支持复制。
		• 在跨sql_mode模式建表时,MySQL受宽松模式和严格模式控制,GaussDB可能存在严格模式失效的情况。例如:源表存在默认值"0000-00-00",在"no_zero_date"严格模式下,GaussDB建表成功,且包含默认值"0000-00-00",严格模式失效;而MySQL建表失败,受严格模式控制。
支持增加子分区语法	ALTER TABLE [IF EXISTS] { table_name [*] ONLY table_name ONLY (table_name)} add_clause; add_clause: ADD {{partition less than item	保留原分区表语法。 不支持下述语法添加多分区: ALTER TABLE table_name ADD PARTITION (partition_definition1, partition_definition1,···);
	<pre>{{partition_tess_than_item partition_test_than partition_list_item PARTITION({partition_tess_than _item partition_start_end_item partition_list_item})}</pre>	仅支持原有添加多分区语法: ALTER TABLE table_name ADD PARTITION (partition_definition1), ADD PARTITION (partition_definition2[y1]), …;

概述	详细语法说明	差异
TRUNCATE子分区语法	ALTER TABLE [IF EXISTS] table_name truncate_clause;	支持子项有差异,对于 truncate_clause: M-Compatibility模式: TRUNCATE PARTITION { { ALL partition_name [,] } FOR (partition_value [,]) } [UPDATE GLOBAL INDEX] MySQL支持: TRUNCATE PARTITION {partition_names ALL}
主键索引名	CREATE TABLE table_name (col_definitine ,PRIMARY KEY [index_name] [USING method] ({ column_name (expression) }[ASC DESC] } [,]) index_parameters [USING method COMMENT 'string'])	GaussDB中的主键指定索引名 后创建的索引名为用户所指定 的索引名,MySQL索引名为 PRIMARY。
删除有依赖的对象	DROP drop_type name CASCADE;	在GaussDB中,删除有依赖的 对象需要加CASCADE, MySQL不需要。
NOT NULL约束不允许 插入NULL值	CREATE TABLE t1 (id int NOT NULL DEFAULT 8); INSERT INTO t1 VALUES(NULL); INSERT INTO t1 VALUES(1), (NULL),(2);	在MySQL宽松模式下,会将 NULL进行类型转换,并成功插 入数据;在MySQL严格模式下 不允许插入NULL值。在 GaussDB不支持此特性,在宽 松模式和严格模式下均不允许 插入NULL值。
CHECK约束生效	CREATE TABLE	CREATE TABLE带CHECK约束的时候,MySQL8.0会生效,MySQL5.7只解析语法但不生效。GaussDB在此功能上同步MySQL8.0版本,且GaussDBCHECK约束可以引用其他列,而MySQL不能。GaussDB一个表中最多只能加32767个CHECK约束。
索引的algorithm和 lock选项不起作用	CREATE INDEX DROP INDEX	M-Compatibility模式的 CREATE/DROP INDEX语句中 INDEX选项algorithm_option 和lock_option目前只在语法上 支持,创建时不报错,但实际 不起作用。

概述	详细语法说明	差异
CREATE TABLE hash 分区和二级分区的存储 与MySQL不同	CREATE TABLE	GaussDB的CREATE TABLE语句中hash分区表和二级分区表所使用的hash函数与MySQL不一致,因此hash分区表和二级分区表的存储与MySQL有区别。
分区表索引	CREATE INDEX	GaussDB的分区表索引分为 LOCAL和GLOBAL两种。 LOCAL索引与某个具体分区绑 定,而GLOBAL索引则对应整 个分区表。 LOCAL和GLOBAL索引的创建
		方法和默认规则具体说明参见《开发指南》中"SQL语法 > SQL语句 > C > CREATE INDEX"章节,例如:在非分区键上创建唯一索引,会默认创建为GLOBAL索引。
		MySQL无GLOBAL索引的概念。在GaussDB中,当分区表索引为GLOBAL索引时,对表分区进行DROP、TRUNCATE、EXCHANGE等操作不会默认更新GLOBAL索引,进而导致GLOBAL索引失效,导致后续语句无法选中该索引。为了避免这种场景,建议用户在使用分区操作语法时在最后显指定UPDATEGLOBAL INDEX子句,或配置全局GUC参数enable_gpi_auto_update为true(推荐),使得在进行分区操作时自定更新GLOBAL索引。
CREATE/ALTER TABLE 语句中分区表为KEY分 区,不支持指定 algorithm。部分分区 定义入参不支持表达 式。	CREATE TABLE、ALTER TABLE	GaussDB的CREATE/ALTER TABLE语句中分区表为KEY分区,不支持指定algorithm。不支持表达式入参的语法: PARTITION BY HASH() PARTITION BY KEY() VALUES LESS THAN()
分区表不支持 LINEAR/KEY hash	CREATE TABLE PARTITION	GaussDB分区表不支持 LINEAR/KEY hash。

概述	详细语法说明	差异
check和 auto_increment语法不 能作用在同一字段	CREATE TABLE	由于MySQL5.7的check字段不 生效,check和 auto_increment同时作用于同 一字段只有auto_increment生 效,但GaussDB报错。
删除存在依赖关系的表	DROP TABLE	GaussDB删除存在依赖的表必 须加上CASCADE才能成功, MySQL不需要。

概述	详细语法说明	差异
新增外键约束、修改外 键约束的参考字段、被 参考字段	CREATE TABLE、ALTER TABLE	• GaussDB创建外键时支持指定MATCH FULL和MATCH SIMPLE选项,如果用户指定了MATCH PARTIAL选项,会提示报错信息。MySQL中支持指定以上选项,但并不生效,行为与MATCH SIMPLE一致。
		GaussDB创建外键时可以指定ON [UPDATE DELETE] SET DEFAULT选项。MySQL创建外键时指定ON [UPDATE DELETE] SET DEFAULT选项会报错。
		GaussDB创建外键时,必须在被参考表的被参考列上创建唯一索引。MySQL创建外键时,需要在被参考表的被参考列上创建索引,可以不是唯一索引。
		• GaussDB创建外键时,不需要在参考表的参考列上创建索引。MySQL创建外键时,需要在参考表的参考列上创建索引,如果在参考表的参考列上创建索引,如果在参考表的参考列上没有索引,则会自动添加一条对应的索引,在删除外键时,不会删除自动添加的索引。
		• GaussDB的参考表和被参考表可以都是临时表,不可以在临时表和非临时表之间创建外键。MySQL无法使用临时表作为参考表或被参考表。MySQL在创建外键指定被参考表时,不会匹配当前会话已创建的临时表。
		• GaussDB创建外键时可以不 指定被参考表的被参考字段 名,此时会将被参考表中的 主键作为外键的被参考字 段。MySQL中,必须指定 被参考表的被参考字段。
		● GaussDB无论 foreign_key_checks是否关 闭,都可以修改参考字段或 被参考字段的数据类型。

概述	详细语法说明	差异
		MySQL仅当foreign_key_checks为off时,才可以修改参考字段或被参考字段的数据类型。 GaussDB可以删除参考表的参考字段,此时会级联删除相关外键约束。MySQL在删除参考表的参考字段时,会发生删除包含被参考表的模式,且参考表在其他模式中时,foreign_key_checks为on,则删除失败。
表定义相关选项	CREATE TABLE \ ALTER TABLE	● GaussDB不支持以下选项: AVG_ROW_LENGTH、 CHECKSUM、 COMPRESSION、 CONNECTION、DATA DIRECTORY、INDEX DIRECTORY、 DELAY_KEY_WRITE、 ENCRYPTION、 INSERT_METHOD、 KEY_BLOCK_SIZE、 MAX_ROWS、 MIN_ROWS、 PACK_KEYS、 PASSWORD、 STATS_AUTO_RECALC、 STATS_PERSISTENT、 STATS_SAMPLE_PAGES。 ● 以下选项在GaussDB中不报 错,但实际上也不生效: ENGINE、 ROW_FORMAT。
CMK密钥轮转,轮换 加密COLUMN ENCRYPTION KEY的 CLIENT MASTER KEY,对COLUMN ENCRYPTION KEY明文 进行重加密。	ALTER COLUMN ENCRYPTION KEY	M-Compatibility模式不支持全密态。故不支持该语法。

概述	详细语法说明	差异
密态等值查询特性使用 多级加密模型,主密钥 加密列密钥,列密钥加 密数据。本语法用于创 建主密钥对象。	CREATE CLIENT MASTER KEY	M-Compatibility模式不支持全 密态。故不支持该语法。
创建一个列加密密钥, 该密钥可用于加密表中 的指定列。	CREATE COLUMN ENCRYPTION KEY	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,传输密钥 到服务端缓存,只在开 启内存解密逃生通道的 情况下使用。	\send_token	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,传输密钥 到服务端缓存,只在开 启内存解密逃生通道的 情况下使用。	\st	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,销毁服务 端缓存的密钥,只在开 启内存解密逃生通道的 情况下使用。	\clear_token	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,销毁服务 端缓存的密钥,只在开 启内存解密逃生通道的 情况下使用。	\ct	M-Compatibility模式不支持全 密态。故不支持该语法。
在全密态数据库特性 中,用于设置访问外部 密钥管理者的参数。	\key_info KEY_INFO	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,用于开启 三方动态库功能与加载 三方动态库时的参数设 置。	\crypto_module_info MODULE_INFO	M-Compatibility模式不支持全 密态。故不支持该语法。
全密态功能,用于开启 三方动态库功能与加载 三方动态库时的参数设 置。	\cmi MODULE_INFO	M-Compatibility模式不支持全 密态。故不支持该语法。
generated always as 语句不能再引用由 generated always as 生成的列。	Generated Always AS	GaussDB generated always as语句不能再引用由generated always as生成的列,MySQL 可以。

概述	详细语法说明	差异
支持更改表名语法	ALTER TABLE tbl_name RENAME [TO AS =] new_tbl_name;	GaussDB的ALTER RENAME语 法仅支持修改表名称功能操 作,不能耦合其它功能操作。
	或RENAME {TABLE TABLES} tbl_name TO new_tbl_name [, tbl_name2 TO	GaussDB仅旧表名字段支持如 schema.table_name用法;且 新表名与旧表名将属于同一 Schema下。
	new_tbl_name2,];	GaussDB不支持新旧表跨 Schema重命名操作;但如有权 限,则可在当前Schema下修改 其他Schema下表名称。
		GaussDB的RENAME多组表的 语法支持全为本地临时表的重 命名,不支持本地临时表和非 本地临时表组合的场景。
禁用GUC参数 enable_expr_fusion	SET enable_expr_fusion= ON	M-Compatibility模式暂不支持 GUC参数enable_expr_fusion 打开。

概述	详细语法说明	差异
支持CREATE VIEW AS SELECT语法	CREATE VIEW table_name AS query;	● 当不开启精度传递开关 (m_format_behavior_compat_options不开启 enable_precision_decimal选项)时,针对以下类型,不支持CREATE VIEW view_name AS query语法中query包含计算操作(如函数调用、使用操作符计算),仅允许直接的字段调用(如 SELECT col1 FROM table1)。精度传递开关打开 (m_format_behavior_compat_options开启enable_precision_decimal选项)时可以使用。 - BINARY[(n)] - VARBINARY(n) - CHAR[(n)] - VARCHAR(n) - TIME[(p)] - DATETIME[(p)] - DIMERIC[(p[,s])] - DEC[(p[,s])] - DEC[(p[,s])] - FLOAT4[(p, s)] - FLOAT4[(p, s)] - FLOAT[(p)] - REAL[(p, s)] - FLOAT[(p, s)] - DOUBLE[(p,s)] - DOUBLE PRECISION[(p,s)] - TEXT - TINYTEXT - MEDIUMTEXT - LONGTEXT - BLOB

概述	详细语法说明	差异
		longtext,GaussDB返回 mediumtext。如: MySQL 5.7行为: mysql> CREATE TABLE IF NOT EXISTS tb_1 (id int,col_text2 text); Query OK, 0 rows affected (0.02 sec)
		mysql> CREATE TABLE IF NOT EXISTS tb_2 (id int,col_text2 text); Query OK, 0 rows affected (0.02 sec)
		mysql> CREATE VIEW v1 AS SELECT * FROM (SELECT cast(col_text2 AS char) c37 FROM tb_1) t1 -> UNION ALL SELECT * FROM (SELECT cast(col_text2 as char) c37 FROM tb_2) t2; Query OK, 0 rows affected (0.00 sec)
		mysql> DESC v1; +++++ ++-+++++ Field Type
		++ 1 row in set (0.00 sec) GaussDB行为: mysql_regression=# CREATE TABLE IF NOT EXISTS tb_1 (id
		int,col_text2 text); CREATE TABLE mysql_regression=# CREATE TABLE IF NOT EXISTS tb_2 (id int,col_text2 text); CREATE TABLE mysql_regression=# CREATE VIEW v1 AS SELECT * FROM (SELECT cast(col_text2 AS char) c37 from tb_1) t1
		mysql_regression-# UNION ALL SELECT * FROM (SELECT cast(col_text2 AS char) c37 FROM tb_2) t2; CREATE VIEW mysql_regression=# DESC v1; Field Type Null Key Default Extra
		++ c37 mediumtext YES
		● 使用bitstring常量进行视图 创建时,与MySQL不一

概述	详细语法说明	差异
		致,MySQL转换成 hexstring进行创建, GaussDB使用bitstring直接 创建。由于bitstring常量为 unsigned值,因此GaussDB 创建视图的属性为 unsigned。 – MySQL5.7行为:
		mysql> SELECT version();
		++ version()
		mysql> DROP VIEW IF EXISTS v1; Query OK, 0 rows affected, 1 warning (0.00 sec)
		mysql> CREATE VIEW v1 AS SELECT b'101'/b'101' AS c22; Query OK, 0 rows affected (0.01 sec)
		mysql> DESC v1; +++ +++ Field Type Null Key Default Extra
		++++ +++ c22 decimal(5,4) unsigned YES NULL ++
		1 row in set (0.00 sec)
		mysql> SHOW CREATE VIEW v1; + +
		++ ++ View Create View
		character_set_client collation_connection +
		++
		v1 CREATE ALGORITHM=UNDEFINED DEFINER=`omm`@`%` SQL SECURITY DEFINER VIEW `v1` AS select (0x05 / 0x05) AS `c22` utf8mb4
		utf8mb4_general_ci

概述	详细语法说明	差异
		+ + + 1 row in set (0.00 sec) - GaussDB行力: m_db=# DROP VIEW IF EXISTS v1; DROP VIEW m_db=# CREATE VIEW v1 AS SELECT b'101'/b'101' AS c22; CREATE VIEW m_db=# DESC v1; Field Type Null Key Default Extra
索引名可重名范围	CREATE TABLE CREATE INDEX	MySQL中索引名在一个表下唯一,在不同的表下可以有相同的索引名。M-Compatibility模式中的索引名在同一个SCHEMA下唯一,在同一的SCHEMA下不可用相同的索引名。在M-Compatibility模式下,针对会自动创建索引的约束和key,也会有相同的规则。

概述	详细语法说明	差异
视图依赖差异	CREATE VIEW、ALTER TABLE	MySQL中视图存储,只记录名 标表的表名、列名、数据一标表的表名。列名、数据一标表的表名。列名、数据一标表的表目标表的唯一标识。因是标识。因此有一种的形型,是一种的一种的的的。 一种的一种的一种的的的。 一种的一种的的的,是一种的一种的的的。 一种的一种的的的,是一种的一种的的的。 一种的一种的一种的一种的一种的一种的一种的一种的一种的一种的一种的一种的一种的一
		改,且后续可以查询该视 图。
外键差异	CREATE TABLE	GaussDB外键约束对类型不敏感,如果主表和从表对应的字段数据类型存在隐式类型转换就可以建成。MySQL外键类型敏感。如果两个表对应的列类型不同外键无法建成。
		COLUMN或CHANGE COLUMN方式修改表列外键所 在列的数据类型或列名等, GaussDB可以。
索引升降序差异	CREATE INDEX	在MySQL 5.7中,ASC DESC 被解析但是被忽略,默认行为 为ASC;在MySQL 8.0及 GaussDB中,ASC DESC被解 析且生效。

概述	详细语法说明	差异
修改视图定义 CREATE OR REPLACE VIEW、ALTER VIEW	MySQL可以对视图的任何属性 进行修改;GaussDB禁止修改 不可更新视图当中的列名、列 类型以及删除列,允许修改可 更新视图列名、列类型以及删 除列。	
		MySQL对嵌套视图的底层视图 执行修改列操作后,只要列名 存在,上层视图就可以正常使 用;GaussDB对嵌套视图中底 层视图做修改列名、列类型以 及删除列操作,会导致上层视 图不可用。
ANALYZE分区语法	ALTER TABLE tbl_name ANALYZE PARTITION {partition_names ALL}	 GaussDB下该语法仅支持分区统计信息收集功能。 MySQL下分区名partition_names 大小写不区分,GaussDB下分区名不带反引号的情况下不区分大小写,带反引号时区分大小写。 GaussDB下执行成功回显ALTER TABLE,执行报错由已有错误码报错信息决定,MySQL下执行结果以表格回显形式显示。

概述	详细语法说明	差异
支持虚拟生成列语法	[GENERATED ALWAYS] AS (generation_expr) [STORED VIRTUAL]	MySQL数据库中虚拟生成 列支持创建索引,GaussDB 数据库中不支持。
		MySQL数据库中虚拟生成 列支持作为分区键, GaussDB数据库中不支持。
		 GaussDB数据库中生成列的 CHECK约束兼容MySQL8.0 数据库的行为,即CHECK约 束检查生效。
		 MySQL数据库中存储生成 列作为分区键时,支持 ALTER TABLE修改存储生成 列,GaussDB数据库中不支 持。
		 MySQL数据库中向可被更 新的视图中更新生成列的数 据时,支持指定关键字 DEFAULT,GaussDB数据库 中不支持。
		MySQL数据库中虚拟生成 列支持IGNORE特性, GaussDB数据库中不支持。
		● 在GaussDB数据库中查询虚拟生成列等价于查询虚拟生成列等价于查询虚拟生成列的表达式(在表达字符集或字符序不一致时,会将表或字符序不一致的类型做资子的列定义的类型做货处理),此行为在建设处理的人方,可以是是一个人,在是不是一个人,在是一个人,在是一个人,是对应对的数据类型与MySQL数据库存在差异。例如:在使用以为FLOAT类型,在自标识表中以为FLOAT类型,在目标或别定列的数据类型可能为及UBLE,与MySQL数据库存在差异。

概述	详细语法说明	差异
通过CREATE TABLE SELECT新建表并插入数据	CREATE TABLE [AS] SELECT	 天支持replace/ignore。 不支持replace/ignore。 如果SELECT列非直接表列让LL且无默认允许NULL且无默认如: create table t1 select unix_timestamp('2008-01-02 09:08:07.3465') as a,创建出来的对应。 如果需要你的方面。 如果有GUC参照的方面。 如果有GUC参照的方面。 如果有GUC参照的方面。 如果有GUC参照的方面。 如果有GUC参照的方面。 如果有GUC参照的方面。 中国大学型中的方面。 中国大学型中域的方面。 大多种种的对应的方面。 使用CREATE TABLE AS SELECT建表,超过的超过超过超过超过超过超过超过超过超过超过超过超过超过超过超过超过超过超过
ALTER TABLE tabname;	ALTER TABLE tabname;	GaussDB中不支持tablename 为空。
分区键暂不支持key的 column_list为空	CREATE TABLE PARTITION	GaussDB中分区键暂不支持key 的column_list为空。

概述	详细语法说明	差异
UTF8字符集编码最大 长度不同,导致创建 表/视图的字段长度产 生差异	CREATE TABLE [AS] SELECT; CREATE VIEW [AS] SELECT	● 当MySQL的字符集为 utf8,GaussDB的字符集为 utf8(utf8mb4)时,由于 MySQL的 UTF8编码最大 为3字节,GaussDB的utf8(utf8mb4)的编码最大为 4字节,开启guc参数 m_format_behavior_comp at_options = 'enable_precision_decimal' 时,create table as (ctas)和create view as (cvas)创建文本类型时(包括二进制文本)可能存在差异: 由于ctas和cvas的场景会依赖字符集的最长字节长度,例如某一节点返回的最大字节长度GaussDB与MySQL均为1024,那么最终返回的字符长度,MySQL为341(1024/3),GaussDB为256(1024/4)。如: MySQL5.7行为: mysql、CREATE TABLE t1 AS SELECT (case when true then min(521.2312) else GROUP_CONCAT(115.0414) end) res1; Query OK, 1 row affected (0.06 sec) Records: 1 Duplicates: 0 Warnings: 0 mysql〉DESC t1; ++

概述	详细语法说明	差异
		++ res1 varchar(256) YES

概述	详细语法说明	差异
指定字段的默认值	F细语法说明 CREATE TABLE、ALTER TABLE	● MySQL5.7指定默认值时, 仅支持不带括号形式的默认 值。MySQL8.0与GaussDB 支持带括号形式的默认值。 GaussDB m_db=# DROP TABLE IF EXISTS t1, t2; DROP TABLE m_db=# CREATE TABLE t1(a DATETIME DEFAULT NOW()); CREATE TABLE m_db=# CREATE TABLE t2(a DATETIME DEFAULT (NOW())); CREATE TABLE MySQL5.7 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.04 sec) mysql> CREATE TABLE t1(a
		DATETIME DEFAULT NOW()); Query OK, 0 rows affected (0.04 sec) mysql> CREATE TABLE t2(a DATETIME DEFAULT (NOW())); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(NOW()))' at line 1
		MySQL8.0 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.17 sec)
		mysql> CREATE TABLE t1(a DATETIME DEFAULT NOW()); Query OK, 0 rows affected (0.19 sec) mysql> CREATE TABLE t2(a
		DATETIME DEFAULT (NOW())); Query OK, 0 rows affected (0.20 sec)
		 MySQL给BLOB、TEXT、 JSON数据类型指定默认值 时必须给默认值添加括号, GaussDB给上述数据类型指 定默认值时可以不添加括 号。
		GaussDB指定默认值时不会校验默认值是否溢出; MySQL指定不带括号形式的默认值时会校验是否溢

概述	详细语法说明	差异
		出,指定带括号形式的默认 值时不会校验是否溢出。
		● GaussDB支持使用DATE/ TIME/TIMESTAMP开头的时间常量给字段指定默认值, MySQL使用DATE/TIME/ TIMESTAMP开头的时间常量给字段指定默认值时必须在默认值外添加括号。
		GaussDB m_db=# DROP TABLE IF EXISTS t1, t2; DROP TABLE m_db=# CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); CREATE TABLE m_db=# CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); CREATE TABLE
		MySQL5.7 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.02 sec)
		mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); ERROR 1067 (42000): Invalid default value for 'a' mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(TIMESTAMP '2000-01-01 00:00:00'))' at line 1
		MySQL8.0 mysql> DROP TABLE IF EXISTS t1, t2; Query OK, 0 rows affected (0.14 sec)
		mysql> CREATE TABLE t1(a TIMESTAMP DEFAULT TIMESTAMP '2000-01-01 00:00:00'); ERROR 1067 (42000): Invalid default value for 'a' mysql> CREATE TABLE t2(a TIMESTAMP DEFAULT (TIMESTAMP '2000-01-01 00:00:00')); Query OK, 0 rows affected (0.19 sec)

4.3.8.4 DML

表 4-57 DML 语法兼容介绍

序号	概述	详细语 法说明	差异
1	DELETE支持从多个表中删除数据	DELET E	 在多表删除执行过程中,若发现将要删除的元组被其他会话并发修改,会取该条会话匹配中所有元组的最新值重新进行匹配,若依然满足条件再对这条元组进行删除。这个过程中MySQL对所有目标表的删除是一致的,而GaussDB仅会对涉及并发更新的目标表的元组进行重新匹配,可能产生数据不一致。 多表操作语法中目标表和范围表的校验规则与MySQL会存在差异,设置GUC兼容性参数m_format_dev_version为's2'后,检验规则保持一致。
2	DELETE支持ORDER BY 和LIMIT	DELET E	-
3	DELETE支持从指定分区 (或子分区)删除数据	DELET E	-
4	UPDATE支持从多个表中 更新数据	UPDAT E	在多表更新执行过程中,若发现将要 更新的元组被其他会话并发修改,会 取该条会话匹配中所有元组的最新值 重新进行匹配,若依然满足条件再对 这条元组进行更新。这个过程中 MySQL对所有目标表的更新是一致 的,而GaussDB仅会对涉及并发更新 的目标表的元组进行重新匹配,可能 产生数据不一致。
5	UPDATE支持ORDER BY 和LIMIT	UPDAT E	-
6	SELECT INTO语法	SELECT	 GaussDB可以使用SELECT INTO根据查询结果创建一个新表,MySQL不支持。 GaussDB的SELECT INTO语法不支持将多个查询进行集合运算后的结果作为查询结果。

序号	概述	详细语 法说明	差异
7	REPLACE INTO语法	REPLA	● MySQL不受严格模式和宽松模式的影响,可向表中插入时间0值,即: mysql> CREATE TABLE test(f1 TIMESTAMP NOT NULL, f2 DATETIME NOT NULL, f3 DATE NOT NULL); Query OK, 1 row affected (0.00 sec) mysql> REPLACE INTO test VALUES(f1, f2, f3); Query OK, 1 row affected (0.00 sec) mysql> SELECT * FROM test; ++ f1
8	SELECT支持指定多分区 查询	SELECT	-
9	UPDATE支持指定多分区 更新	UPDAT E	-

序号	概述	详细语 法说明	差异
10	LOAD DATA导入数据功能	LOAD DATA	在使用LOAD DATA导入数据功能时, GaussDB与MySQL相比有如下差异:
			● LOAD DATA语法执行结果与M*严 格模式一致,宽松模式暂未适配。
			IGNORE与LOCAL参数功能仅为当 导入数据与表中数据存在冲突时, 忽略当前冲突行数据功能和当文件 中字段数小于指定表中列数时自动 为其余列填充默认值功能,其余功 能暂未适配。
			指定LOCAL关键字,且文件路径为相对路径时,文件从二进制目录下搜索;不指定LOCAL关键字,且文件路径为相对路径时,文件从数据目录下搜索。
			● LOAD DATA仅支持从服务端导入文件。
			● [(col_name_or_user_var [, col_name_or_user_var])]指定列 参数不支持重复指定列。
			● [FIELDS TERMINATED BY 'string'] 指定换行符不能与[LINES TERMINATED BY 'string']分隔符相 同。
			执行LOAD DATA语法写入表中的数据若无法转换为表中数据类型格式时报错。
			LOAD DATA SET表达式中不支持指 定列名计算。
			● LOAD DATA只能用于表,不能用于 视图。
			 windows下的文件与linux环境下文件默认换行符存在差异,LOADDATA无法识别此场景会报错,建议用户导入时检查导入文件行尾换行符。
			● 设置 m_format_behavior_compat_optio ns参数中包含值: enable_load_data_remote_transm ission后,指定local参数支持从客 户端读取文件数据并导入到数据库 表中。

序号	概述	详细语 法说明	差异
11	INSERT支持VALUES引用 列语法	INSERT INTO tabna me VALUE S(1,2,3) ON DUPLI CATE KEY UPDAT E b = VALUE S(column_na me)	GaussDB的ON DUPLICATE KEY UPDATE子句中的VALUES()不支持表 名.列名格式,MySQL支持。
12	LIMIT限制差异	DELET E SELECT UPDAT E	各个语句的limit子项与MySQL的limit 项当前存在差异。 GaussDB中limit参数最大值为BIG INT 类型限制(超过 9223372036854775807报错)。在 MySQL中,limit最大值为unsigned LONGLONG类型限制(超过 18446744073709551615 报错)。 limit可以设置小数值,实际执行时四 舍五入。MySQL不能取小数。 GaussDB的delete语句中,不允许limit 0。MySQL在delete语句中允许limit 0。

序号	概述	详细语 法说明	差异
13	反斜杠(\)用法差异	INSERT	反斜杠(\)的用法在GaussDB和MySQL中都可以由参数控制但当前默认用法不同:
			MySQL中使用参数 NO_BACKSLASH_ESCAPES控制字符 串和标识符中的反斜杠(\)被解析为普通字符还是转义字符,默认反斜杠字符(\)作为字符串和标识符中的转义字符。设置set sql_mode='NO_BACKSLASH_ESCAPE S';可以禁用反斜杠字符(\)作为字符串和标识符中的转义字符。 GaussDB中使用参数
			standard_conforming_strings控制字符串和标识符中的反斜杠(\)被解析为普通字符还是转义字符。默认值为on,在普通字符串文本中按照SQL标准把反斜杠(\)当普通文本。使用set standard_conforming_strings=off;将反斜杠字符(\)作为字符串和标识符中的转义字符。
14	插入值少于字段数目 时,MySQL报错, GaussDB补充空值。	INSERT	GaussDB不指定列的列表时,如果插入值少于字段数目,默认按建表时的字段顺序赋值。字段上有非空约束时报错,没有非空约束时,如果指定了默认值则缺省部分补充默认值,若未指定默认值则补充空。
15	ORDER BY中排序的列必须包括在结果集的列中。	SELECT	在GaussDB中,在与GROUP BY子句一起使用的情况下,ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。在与DISTINCT关键字一起使用的情况下,ORDER BY中排序的列必须包括在SELECT语句所检索的结果集的列中。
16	不允许对约束字段用 ON DUPLICATE KEY UPDATE 进行修改。	INSERT	-
17	SELECT结果允许存在重 复列名。	SELECT	-
18	NATURAL JOIN与 MySQL有差异。	SELECT	在GaussDB中,NATURAL [[LEFT RIGHT] OUTER] JOIN允许不指定LEFT RIGHT,不指定时NATURAL OUTER JOIN为NATURAL JOIN。允许连续使用多次JOIN。

序号	概述	详细语 法说明	差异
19	外键数据类型是 timestamp/datetime 时,update/delete外表 报错。	UPDAT E/ DELET E	外键数据类型是timestamp/datetime 时,update/delete外表报错,MySQL 成功。
20	nature join和using兼容。	SELECT	 GaussDB join的顺序严格按照从左往右,MySQL可能会调整顺序。 GaussDB和MySQL在natural join与using时均不允许左表或右表参与join的字段出现歧义(一般由左或右临时表中重名字段造成)。因为两者join的顺序有差别,故行为上可能有差别。 GaussDB的行为:
			NATURAL JOIN t3; ERROR 1052 (23000): Column 'a' in from clause is ambiguous
21	with clause兼容 MySQL8.0版本	SELECT	-
		INSERT	
		UPDAT E、 DELET E	
22	join兼容	SELECT	GaussDB join不支持使用逗号","的 连接方式,MySQL支持。 GaussDB不支持use index for join。

序号	概述	详细语 法说明	差异
	概述 SELECT语句显示的列名		● 为了使SELECT语句显示的列名与MySQL一致,需要打开列名回显控制开关: SET m_format_behavior_compat_options = 'select_column_name' ● 不设置此配置项时: - SELECT系统函数: 回显为系统函数名。 - SELECT系统函数: 回显为系统函数名。 - SELECT表达式: 回显为字column?。 - SELECT布尔值: 回显为bool。 ● 设置此配置项时,列名回显为全部的函数或表达式输入。 - 对于普通注释,MySQL客户端会将注释忽略,gsql客户端和pymysql不会忽略。 - 对于如/*!形式开头的注释,MySQL服务端会将其转为可执行语句,M兼容暂不支持识别此类注释,因此作为普通注释处理。 - 对于包含目后面无空格的表达式,M兼容不支持识别此类注释,因此作为普通注释处理。 - 对于包含目后面无空格的表达式,M兼容有等以字符,只有在设置了m_format_behavior_compat_options为enable_escape_string后才会显示转义字符本身,比如"SELECT"abc\tdef";" M兼容在未开启上述设置时显示为abc\tdef。m_db=# SET m_format_behavior_compat_options='select_column_name,enable_escape_string'; SET m_db=# SELECT "abc\tdef";
			m_db=# SELECT "abc\tdef"; abc def abc def (1 row) m_db=# SET m_format_behavior_compat_options='sel ect_column_name'; SET m_db=# SELECT "abc\tdef";

序号 概述	详细语 法说明	差异
		abc\tdef (1 row) - 列名超过63个字符时,会截断后面部分。 - 表达式最后的部分为注释时,则不会显示最后的注释以及与注释相连的空格。 m_db=# SELECT 123 /* 456 */; 123 123 (1 row) - 表达式为布尔值时,无论输入大小写,回显为TRUE或FALSE。 m_db=# SELECT true; TRUE t (1 row) - 表达式为null时,无论输入大小写,回显为NULL。 m_db=# SELECT null; NULL (1 row) - 表达式包含-时,会将全部的输入作为列名输出。 m_db=# SELECT (+-+1); (+-+1)1 (1 row) m_db=# SELECT -true; -true1 (1 row) m_db=# SELECT -true; -1 (1 row)

序号	概述	详细语 法说明	差异
24	SELECT导出文件(into outfile)	SELECT INTO OUFIL E	SELECT INTO OUTFILE语法,导出文件中FLOAT、DOUBLE、REAL类型的值显示精度和MySQL存在差异,不影响COPY导入和导入后的值。
25	UPDATE/INSERT/ REPLACE SET指定模 式名、表名	UPDAT E/ INSERT / REPLA CE SET	 SELECT语句指定投影列时,MySQL 支持模式名.表别名.列名的三段式用法,GaussDB不支持。 m_db=# CREATE SCHEMA test; CREATE SCHEMA m_db=# CREATE TABLE test.t1(a int); CREATE TABLE m_db=# SELECT test.alias1.a FROM t1 alias1; ERROR: invalid reference to FROM-clause entry for table "alias1" LINE 1: select test.alias1.a from t1 alias1;
26	UPDATE SET执行顺序与 MySQL存在差异	UPDAT E SET	MySQL中,UPDATE SET的顺序是从前往后依次UPDATE,前面UPDATE的结果会影响后面的结果,且允许多次设置同一列;GaussDB中为先取出原来的所有相关的数据,再一次性UPDATE,且不允许多次设置同一列,二者存在差异。设置GUC兼容性参数m_format_dev_version为's2'后,仅在单表场景可保持与MySQL行为一致,既支持同一列设置多次,且引用更新后的结果。

序号	概述	详细语 法说明	差异
27	IGNORE特性	UPDAT E/ DELET E/ INSERT	MySQL数据库和GaussDB执行过程的 差异,因此产生的WARNING条数和 WARNING信息可能存在不同。
28	SHOW COLUMNS语法	SHOW	 ● 用户权限验证与MySQL存在差异。 - GaussDB中需要拥有指定表所在Schema的USAGE权限,同时还需要拥有指定表的任意表级权限或列级权限,仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 - MySQL中需要拥有指定表的任意表级权限或列级权限,仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 ● LIKE和WHERE子句中涉及到字符串比较操作时,Field、Collation、Null、Extra、Privileges字段使用字符集utf8mb4、字符序utf8mb4_general_ci,Type、Key、Default、Comment字段使用字符集utf8mb4_bin。 ● GaussDB中建议用户在WHERE子句中,不要对返回字段以外的列进行选择,否则可能会出现非预期的报错。一种,不要对返回字段以外的列进行选择,否则可能会出现非预期的报错。一种,不要对该回字段以外的列进行选择,否则可能会出现非预期的报告。 → 预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE 'b'='pri'; ERROR: Column "b" does not exist. LINE 1: SHOW FULL COLUMNS FROM t02 WHERE 'b'='pri'; - 非预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE 'c'='pri'; - 非预期报错 m_db=# SHOW FULL COLUMNS FROM t02 WHERE 'c'='pri'; ERROR: input of anonymous composite types is not implemented LINE 1: SHOW FULL COLUMNS FROM t02 WHERE 'c'='pri';

序号	概述	详细语 法说明	差异
29	SHOW CREATE DATABASE语法	SHOW	用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定Schema的USAGE权限。 MySQL中需要拥有任意库级权限(除GRANT OPTION和USAGE)、任意表级权限(除GRANT OPTION)或任意列级权限。
30	SHOW CREATE TABLE 语法	SHOW	 用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定表所在Schema的USAGE权限和指定表的任意表级权限。 MySQL中需要拥有指定表的任意表级权限(除GRANTOPTION)。 返回的建表语句与MySQL存在差异。 GaussDB中索引以CREATEINDEX语句的形式返回。MySQL中表的索引在CREATETABLE语句中返回。主要因为GaussDB中CREATEINDEX语法支持的可选参数范围与CREATETABLE语法中创建索引不同,因此某些索引无法在CREATETABLE语句中创建。 GaussDB中CREATETABLE语句中创建。 GaussDB中CREATETABLE语句中创建。 GaussDB中CREATETABLE语句中创度。 GaussDB中CREATETABLE语句中创度。 GaussDB中CREATETABLE语句中创度。 证成以做了语法适配,实际不生效,因此在返回的建表语句中不予显示。 设置兼容性参数m_format_dev_version为's2'后,返回的建表语句才兼容MySQL。兼容的内容包括:列注释位置变更、未完全更大多数的内容包括:列注释位置变更、全局临时表ONCOMMIT选项位置实更、全局临时表ONCOMMIT选项位置变更、主键与电中约束位置变更、主键与束中的USINGINDEXTABLESPACE选项不再显示以及索引注释位置变更。

序号	概述	详细语 法说明	差异
31	SHOW CREATE VIEW语法	SHOW	 用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定视图所在Schema的USAGE权限和指定视图的任意表级权限。 MySQL中需要拥有指定视图的表级SELECT和表级SHOW VIEW权限。 返回的视图创建语句与MySQL存在差异。以SELECT*FROMtbl_name形式创建的视图,GaussDB中*不会被展开,而MySQL中会展开。 返回结果中的character_set_client和collation_connection字段与MySQL存在差异。 MySQL中显示视图创建时系统变量character_set_client和collation_connection的会话值 GaussDB中未记录相关元数据,显示为NULL。
32	SHOW PROCESSLIST语 法	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.processlist视图内字段内容与大小写保持一致,MySQL中可能存在差异。 GaussDB中用户只能访问自己的线程信息,拥有SYSADMIN权限的用户可以访问所有用户的线程信息。 MySQL中用户只能访问自己的线程信息,拥有PROCESS权限的用户可以访问所有用户的线程信息。
33	SHOW [STORAGE] ENGINES	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.engines视图内字段内容与大小写保持一致,MySQL中可能存在差异。因为MySQL与GaussDB的存储引擎不同,所以该指令查询的结果不同。
34	SHOW [SESSION] STATUS	SHOW	GaussDB中该命令的查询结果中的字段内容和大小写与information_schema.session_status视图内字段内容与大小写保持一致,MySQL中可能存在差异。GaussDB中当前仅支持Threads_connected和Uptime。

序号	概述	详细语 法说明	差异
35	SHOW [GLOBAL] STATUS	SHOW	GaussDB中该命令的查询结果中的字 段内容和大小写与 information_schema.global_status视 图内字段内容与大小写保持一致, MySQL中可能存在差异。GaussDB中 当前仅支持Threads_connected和 Uptime。
36	SHOW INDEX	SHOW	 用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定 SCHEMA的USAGE权限和指定表的任意表级权限或者任意列级权限。 MySQL中需要拥有指定表的任意表级权限(除GRANTOPTION)或者任意列级权限。 GaussDB中临时表存储于独立的临时Schema中,在使用FROM或INdb_name条件来展示指定临时表索引信息时,须指明db_name为临时表所在的Schema才能展示临时表索引信息,否则会提示不存在该临时表,这一点和Mysql在部分情况下存在差异。 GaussDB中,查询结果中Table、Index_type、Index_comment字段使用字符集utf8mb4、字符序utf8mb4_bin;字段Key_name,Column_name、Collation、Null、Comment字段使用字符集utf8mb4、字符序utf8mb4、字符序utf8mb4_general_ci。
37	SHOW SESSION VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与 information_schema.session_variable s视图内字段内容及大小写保持一致,与MySQL可能存在差异。 GaussDB中对查询结果中字段使用 LIKE和WHERE进行选择时,排序规则与 information_schema.session_variable s视图内对应字段保持一致。

序号	概述	详细语 法说明	差异
38	SHOW GLOBAL VARIABLES	SHOW	GaussDB中查询结果中字段内容及大小写与 information_schema.global_variables 视图内字段内容及大小写保持一致,与MySQL可能存在差异。 GaussDB中对查询结果中字段使用 LIKE和WHERE进行选择时,排序规则与 information_schema.global_variables 视图内对应字段保持一致。
39	SHOW CHARACTER SET	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.character_sets视图内字段内容及大小写保持一致,与MySQL可能存在差异。 GaussDB中对查询结果中字段使用LIKE和WHERE进行选择时,排序规则与information_schema.character_sets视图内对应字段保持一致。
40	SHOW COLLATION	SHOW	GaussDB中查询结果中字段内容及大小写与information_schema.collations视图内字段内容及大小写保持一致,与MySQL可能存在差异。 GaussDB中对查询结果中字段使用LIKE和WHERE进行选择时,排序规则与information_schema.collations视图内对应字段保持一致。
41	EXCEPT语法	SELECT	-
42	SELECT支持 STRAIGHT_JOIN语法	SELECT	GaussDB中多表关联JOIN场景下生成 的执行计划,与MySQL可能存在差 异。

序号	概述	详细语 法说明	差异
43	SHOW TABLES	SHOW	 LIKE行为存在差异,具体请参见操作符章节的"LIKE"。 WHERE表达式行为存在差异,具体行为请参见GaussDB数据库的"WHERE表达式"。 GaussDB中:表和数据库的权限需要分开赋予用户,查询的数据库必须是用户在SHOW SCHEMAS上可以查询到,不能仅仅有表的权限,必须还需要有数据库的权限。MySQL中只要拥有表权限即可访问。 GaussDB中:校验逻辑中优先校验Schema是否存在,后校验当前用户是否对Schema具有权限,与MySQL存在差异。 GaussDB中:查询结果中字段使用字符集utf8mb4、字符序utf8mb4、字符序utf8mb4_bin。 GaussDB中:LIKE子句中,当目标database是information_schema时,pattern被转为小写再进行匹配。在Mysql8.0中,当目标database是information_schema时,pattern被转为大写再进行匹配。有例实引息不同,并可以表述的概率

序号	概述	详细语 法说明	差异
44	SHOW TABLE STATUS	SHOW	GaussDB中:该语法展示数据依赖 information_schema下的tables视图。MySQL中tables指定的是表。
			GaussDB中: 表和数据库的权限需要分开赋予用户,查询的数据库必须是用户在SHOW SCHEMAS上可以查询到,不能仅仅有表的权限,必须还需要有数据库的权限。 MySQL中只要拥有表权限即可访问。
			 GaussDB中:校验逻辑中优先校验 Schema是否存在,后校验当前用户 是否对Schema具有权限,与 MySQL存在差异。
			 GaussDB中:对查询结果中字段使用LIKE和WHERE进行选择时,排序规则与information_schema.tables视图内对应字段保持一致。
			● GaussDB中: LIKE子句中,当目标 database是information_schema 时,pattern被转为小写再进行匹配。在Mysql8.0中,当目标 database是information_schema 时,pattern被转为大写再进行匹配。
45	GROUP BY 后支持WITH ROLLUP	SELECT	GaussDB支持既有WITH ROLLUP又有 ORDER BY写法,MySQL不支持。
46	支持SQL Mode中 ONLY_FULL_GROUP_BY 选项	SELECT	SELECT列表中非聚合函数列与GROUP BY字段不一致时,非聚合函数列都需要出现GROUP BY列表或WHERE列表中,且WHERE子句中的列需要等于某一常量时,不报错。其中WHERE子句中的列,GaussDB支持入参数为1的函数列表达式,MySQL不支持函数列表达式。 GaussDB只支持GROUP BY后字段为正整数。
47	HAVING语法	SELECT	GaussDB的HAVING必须且只能引用GROUP BY子句中的列或聚合函数中使用的列。MySQL支持对此行为的扩展,并允许HAVING引用列表中的SELECT列和外部子查询中的列。

序号	概述	详细语 法说明	差异
48	使用SELECT查询系统参数、用户变量	SELECT @varia ble \ SELECT @@var iable	 MySQL支持查询用户变量时,不添加具体的变量名(即SELECT @),GaussDB不支持。 MySQL的行为: mysql> SELECT @; ++ @

序号	概述	详细语 法说明	差异
49	子查询	SELECT	● GaussDB不支持子查询结果包含多列,包含多列时执行会报错; MySQL支持子查询包含多列。 MySQL的行为: mysql> SELECT row(1,2) = (SELECT 1,2); +

序号	概述	详细语 法说明	差异
			greatest
			(1 row)

序号	概述	详细语 法说明	差异
50	SHOW DATABASES	SHOW	GaussDB中:查询结果中字段使用字符集utf8mb4、字符序utf8mb4_bin。
51	SELECT后跟行表达式	SELECT	MySQL不支持SELECT后跟行表达式,GaussDB支持SELECT后跟行表达式。 MySQL的行为: mysql> SELECT row(1,2); ERROR 1241 (21000): Operand should contain 1 column(s) GaussDB的行为: m_db=# SELECT row(1,2); row(1,2)

序号 概述	详细语 法说明	差异
52 SELECT视图查询、子查 询或UNION涉及 NUMERIC转TIME/ DATETIME进位差异	SELECT	SELECT部分场景下输出TIME/DATETIME类型结果与MySQL存在差异。 差异场景: 视图查询、子查询或UNION; 涉及NUMERIC转TIME/DATETIME。 差异行为: GaussDB的SELECT行为统一,NUMERIC转TIME/DATETIME类型时,只对最大精度位(6)作进位处理。MySQL的视图查询、子查询和UNION场景,对实际的结果精度位作进位处理。MySQL的行为:简单查询,只对最大精度位6作进位,所以输出11:11:00.00002 mysql> SELECT maketime(11, 11, 2.2/time '08:30:23.01'); +

序号	概述	详细语 法说明	差异
53	SELECT在数值类型和子 查询日期与时间函数的 运算处理差异	SELECT	SELECT在数值类型和子查询日期与时间函数的运算场景,GUC参数m_format_behavior_compat_options开启enable_precision_decimal选项,GaussDB会先将函数返回的日期与时间转换为数值类型,然后按照数值类型运算,结果也为数值类型。MySQL在子查询条件查询、分组查询等场景会截断日期与时间函数返回值。MySQL的行为:mysql> select 1.5688 * (select ADDDATE('2020-10-20', interval 1 day) where true group by 1 having true); +

54	SELECT在嵌套子查询时 unsigned类型差异		
	unsigneu夫主左升	SELECT	SELECT在嵌套子查询时unsigned类型不会被覆盖,与MySQL5.7存在差异。 MySQL5.7的行为: mysql> drop table if exists t1; Query OK, 0 rows affected (0.02 sec) mysql> create table t1 (-> c10 real(10, 4) zerofill ->); Query OK, 0 rows affected (0.03 sec) mysql> insert into t1 values(123.45); Query OK, 1 row affected (0.00 sec) mysql> desc t1; ++ Field Type

序号	概述	详细语 法说明	差异
			INSERT 0 1 test=# DESC t1_sub_1; Field Type Null Key Default Extra+
55	SELECT FOR SHRAE/FOR UPDATE/ LOCK IN SHRAE MODE	SELECT	 ● GaussDB不支持FOR SHARE/FOR UPDATE/LOCK IN SHARE MODE 子句和UNION/EXCEPT/DISTINCT/GROUP BY/HAVING子句一起使用,MySQL 5.7部分支持(FOR SHARE/EXCEPT语法不支持),MySQL8.0均支持。 ● 当将锁子句与LEFT/RIGHT[OUTER] JOIN子句连用时,LEFT JOIN不支持给右表上锁,RIGHT JOIN不支持给左表上锁;MySQL可以给JOIN两侧的表同时上锁。

序号	概述	详细语 法说明	差异
56	SELECT语法支持范围	SELECT	 GaussDB指定from子句中的表别名时,支持带字段名称。MySQL5.7不支持指定表别名时带字段名称,MySQL8.0仅支持给子查询指定表别名时带字段名称。
			GaussDB m_db=# DROP TABLE IF EXISTS t1; DROP TABLE m_db=# CREATE TABLE t1(a INT, b INT); CREATE TABLE m_db=# INSERT INTO t1 VALUES(1,2); INSERT 0 1 m_db=# SELECT * FROM t1 t2(a, b); a b+ 1 2 (1 row)
			m_db=# SELECT * FROM (SELECT * FROM t1) t2(a, b); a b+ 1 2 (1 row)
			MySQL5.7 mysql> DROP TABLE IF EXISTS t1; Query OK, 0 rows affected, 1 warning (0.00 sec)
			mysql> CREATE TABLE t1(a INT, b INT); Query OK, 0 rows affected (0.03 sec)
			mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.01 sec)
			mysql> SELECT * FROM t1 t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1 mysql> SELECT * FROM (SELECT * FROM t1) t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1
			MySQL8.0 mysql> DROP TABLE IF EXISTS t1; Query OK, 0 rows affected (0.10 sec)
			mysql> CREATE TABLE t1(a INT, b INT); Query OK, 0 rows affected (0.18 sec)
			mysql> INSERT INTO t1 VALUES(1,2); Query OK, 1 row affected (0.03 sec)
			mysql> SELECT * FROM t1 t2(a, b); ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '(a, b)' at line 1 mysql> SELECT * FROM (SELECT * FROM t1)

序号	概述	详细语 法说明	差异
			t2(a, b); ++

序号	概述	详细语 法说明	差异
57	不携带ORDER BY子句的 UNION、GROUP BY等 语句在数据合并或聚合的时候,由于使用执行器算子存在差异,不保证输出数据顺序一致。	SELECT	以GROUP BY场景为例,使用hashagg 算子时和原始顺序不同,建议需要保 证数据顺序的场景添加ORDER BY子 句。数据初始化 DROP TABLE IF EXISTS test; CREATE TABLE test(id INT); INSERT INTO test VALUES (1),(2),(3),(4),(5); GaussDB不开精度传递,id顺序为(1 3 2 4 5) m_db=# SET m_format_behavior_compat_options= "; SET m_db=# SELECT /*+ use_hash_agg*/ id, pi() FROM test GROUP BY 1,2; id

4.3.8.5 DCL

表 4-58 DCL 语法兼容介绍

序号	概述	详细语法说明	差异
1	SET NAMES指定 COLLATE子句	SET [SESSION LOCAL] NAMES {'charset_name' [COLLATE 'collation_name'] DEFAULT};	GaussDB中SQL_ASCII库下 暂不支持指定 charset_name与数据库字 符集不同。具体请参考 《M-Compatibility开发指 南》中"SQL参考 > SQL 语法 > SQL语句 > S > SET"章节。 不指定字符集时,MySQL 会报错但GaussDB不报 错。
2	支持DESCRIBE语句	{DESCRIBE DESC} tbl_name [col_name wild]	 用户权限验证与MySQL存在差异。 GaussDB中需要拥有指定表所在Schema的USAGE权限,同时还需要拥有指定或规模,仅显示拥有区域和限,仅显示拥有SELECT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 MySQL中需要拥有指定表的权限或列级权限,就是是CT、INSERT、UPDATE、REFERENCES和COMMENT权限的列信息。 模糊匹配时涉及到字符串比较操作时,Field字段使用字符集utf8mb4、字符或utf8mb4_general_ci。

序号	概述	详细语法说明	差异
3	START TRANSACTION 支持 开启一致性读快照	START TRANSACTION [{ ISOLATION LEVEL { READ COMMITTED SERIALIZABLE REPEATABLE READ WRITE READ ONLY } WITH CONSISTENT SNAPSHOT } [,]];	 在MySQL中,可重复读隔离级别下的事务,只有在执行第一个SELECT语句后才开始快照读。在GaussDB中,事务一旦开启,不仅第一个SELECT语句会进行快照读,第一个执行的DDL、DML或DCL语句也会建立事务的一致性读快照 GaussDB的STARTTANSACTION支持设置多次隔离级别/事务访问模式/开启一致性快照,覆盖生效。

序号	概述	详细语法说明	差异
4	SET设置用户变量	SET @var_name := expr	MySQL用户变量名支持使用转义。GaussDB用户变量名,GaussDB用户变量名,要自己的形形。一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一

序号	概述	详细语法说明	差异
			as @var_name := expr is not supported. expr在GaussDB中可以为聚集函数,在MySQL中不支持。
5	SET设置系统参数	SET [SESSION @@SESSION. @@ LOCAL @@LOCAL.] {config_parameter { TO = } { expr DEFAULT } FROM CURRENT }};	 config_parameter为BOOLEAN类型系统参数时: 参数值直接设置为字符串形式的'1'/'0'、'true'/'false'时,M-Compatibility设置成功,MySQL设置失败。 参数值设置为子查询的结果,当查询结果为'true'/'false'、非整数类型1/0时,M-Compatibility设置成功,MySQL设置大败;当查询结果为NULL时,M-Compatibility设置失败,MySQL设置大败,MySQL设置成功。

4.3.8.6 其他语句

表 4-59 其他语法兼容介绍

序号	概述	详细语法说明	差异	
1	事务相关语法	数据库默认隔离 级别	M-Compatibility默认隔离级别为READ COMMITTED,MySQL默认隔离级别为 REPEATABLE READ。	
			M-Compatibility隔离级别只有READ COMMITTED REPEATABLE READ生效。	
2	事务相关语法	事务嵌套	M-Compatibility中嵌套事务不会自动提 交,MySQL会自动提交。	

序号	概述	详细语法说明	差异
3	事务相关语法	自动提交	M-Compatibility使用GaussDB存储,继承GaussDB事务机制,事务中执行DDL,DCL不会自动提交。MySQL在DDL、DCL、管理类语句,锁相关语句会自动提交。
4	事务相关语法	报错后需 rollback	M-Compatibility事务中报错,需要执行 rollback,MySQL无限制。
5	事务相关语法	锁机制	M-Compatibility锁机制只能在事务块中 使用,MySQL无限制。
6	锁机制	锁机制	MySQL获取read锁后,当前会话无法 进行写操作,M-Compatibility获取 read锁后,当前会话可以进行写操 作。
			● MySQL给表上锁后,读取其他表报 错,M-Compatibility无限制。
			MySQL同一会话中获取同一个表的 锁,会自动释放上一个锁,并提交事 务,M-Compatibility无该机制。
			M-Compatibility中LOCK TABLE只能在一个事务块的内部有用,且无UNLOCK TABLE命令,锁总是在事务结束时释放。
7	PBE	PBE	● 重复创建同名的PREPARE语句,M-Compatibility会报已经存在的错误,需要先删除已有statement,MySQL会覆盖旧的statement。
			 M-Compatibility和MySQL在SQL语句 执行过程中对异常场景的报错阶段不 同,例如解析层、执行层等;而 PREPARE语句对预备语句只处理到解 析层。因此PBE下对于异常场景,报错 位置在PREPARE阶段还是EXECUTE阶 段,M-Compatibility和MySQL存在可 能差异。

4.3.8.7 用户与权限

概述

在M-Compatibility中,用户与权限管控相关的行为、语法整体沿用GaussDB的机制,暂不同步MySQL。

用户与权限的行为与GaussDB保持一致,具体行为说明请参见《开发指南》中的"数据库安全 > 用户及权限"章节。

用户与权限的语法在原有GaussDB的基础上,裁剪了部分语法,具体语法说明请参见《M-Compatibility开发指南》中的"SQL参考 > SQL语法 > SQL语句"章节。M-Compatibility与GaussDB的语法差异请参见表4-60。

M-Compatibility创建USER时会自动创建与USER同名的Schema, MySQL不创建。

表 4-60 M-Compatibility 与 GaussDB 的语法差异

序号	语法说明	概述	M-Compatibility与 GaussDB的差异
1	CREATE ROLE	创建一个角色。	在M-Compatibility中,
2	CREATE USER	创建一个用户。	不支持指定涉及以下关键 字的选项:ENCRYPTED、
3	CREATE GROUP	创建一个新用户组。 CREATE GROUP是 CREATE ROLE的别名,不 推荐使用。	UNENCRYPTED、 RESOURCE POOL、PERM SPACE、TEMP SPACE、 SPILL SPACE。
4	ALTER ROLE	修改角色属性。	
5	ALTER UER	修改用户属性。	
6	ALTER GROUP	修改一个用户组的属性。	-
7	DROP ROLE	删除角色。	-
8	DROP USER	删除用户。	-
9	DROP GROUP	删除用户组。	-
10	DROP OWNED	删除一个数据库角色所拥 有的数据库对象。	-
11	REASSIGN OWNED	修改数据库对象的属主。	M-Compatibility中不支持 该语法。
12	GRANT	对角色和用户进行授权操 作。	M-Compatibility中不支持 授予或回收函数、存储过
13	REVOKE	用于撤销一个或多个角色 的权限。	程、表空间、DATABASE LINK等对象的权限。
14	ALTER DEFAULT PRIVILEGES	设置应用于将来创建的对象的权限(这不会影响分配到已有对象中的权限)。	M-Compatibility中不支持 该语法。

差异说明

• 语法格式差异

M-Compatibility的授权语法请参见《M-Compatibility开发指南》中的"SQL参考 > SQL语法 > SQL语句 > G > GRANT"章节,MySQL中的授权语法如下:

-- 全局级、数据库级、表级、存储过程级赋权语法 GRANT

```
priv_type [(column_list)]
    [, priv_type [(column_list)]] ...
   ON [object_type] priv_level
   TO user [auth_option] [, user [auth_option]] ...
   [REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
   [WITH {GRANT OPTION | resource_option} ...]
-- 用户代理赋权语法
GRANT PROXY ON user
   TO user [, user] ...
   [WITH GRANT OPTION]
object_type: {
   TABLÉ
  FUNCTION
 | PROCEDURE
priv_level: {
 db_name.*
 db_name.tbl_name
  tbl_name
 | db_name.routine_name
user:
   'user_name'@'host_name'
auth_option: {
   IDENTIFIED BY 'auth_string'
 | IDENTIFIED WITH auth_plugin
 | IDENTIFIED WITH auth_plugin BY 'auth_string'
 | IDENTIFIED WITH auth_plugin AS 'auth_string'
| IDENTIFIED BY PASSWORD 'auth_string'
tls_option: {
   SSL
 | X509
 CIPHER 'cipher'
 | ISSUER 'issuer'
| SUBJECT 'subject'
resource_option: {
 | MAX_QUERIES_PER_HOUR count
  MAX_UPDATES_PER_HOUR count
 MAX_CONNECTIONS_PER_HOUR count
| MAX_USER_CONNECTIONS count
```

• 赋权类型差异

MySQL支持的赋权类型如下:

表 4-61 MySQL 支持的赋权类型

权限类型	释义及权限级别
ALL [PRIVILEGES]	授予指定访问级别的所有权限,除了 GRANT OPTION和 PROXY。
ALTER	启用ALTER TABLE。级别:全局、数 据库、表。

权限类型	释义及权限级别
ALTER ROUTINE	允许更改或删除存储过程。级别:全 局、数据库、例程。
CREATE	启用数据库和表创建。级别:全局、 数据库、表。
CREATE ROUTINE	启用存储过程创建。级别:全局、数 据库。
CREATE TABLESPACE	允许创建、更改或删除表空间和日志 文件组。级别:全局。
CREATE TEMPORARY TABLES	启用CREATE TEMPORARY TABLE。 级别:全局、数据库。
CREATE USER	启用CREATE USER、 DROP USER、 RENAME USER和REVOKE ALL PRIVILEGES。级别:全局。
CREATE VIEW	允许创建或更改视图。级别:全局、 数据库、表。
DELETE	启用DELETE. 级别:全局、数据库、 表。
DROP	允许删除数据库、表和视图。级别: 全局、数据库、表。
EVENT	启用定时任务。级别:全局、数据 库。
EXECUTE	使用户能够执行存储过程。级别:全局、数据库、存储过程。
FILE	使用户能够使服务器读取或写入文 件。级别:全局。
GRANT OPTION	允许向其他账户授予权限或从其他账 户删除权限。级别:全局、数据库、 表、存储过程、代理。
INDEX	允许创建或删除索引。级别:全局、 数据库、表。
INSERT	启用INSERT。级别:全局、数据库、 表、列。
LOCK TABLES	在具有SELECT权限的表上启用LOCK TABLES 。级别:全局、数据库。
PROCESS	使用户能够通过SHOW PROCESSLIST 查看所有正在运行的线程. 级别:全 局。
PROXY	启用用户代理。级别: 从用户到用 户。

权限类型	释义及权限级别
REFERENCES	启用外键创建。级别:全局、数据 库、表、列。
RELOAD	启用FLUSH操作的使用。级别:全 局。
REPLICATION CLIENT	使用户能够查询源服务器或副本服务器的位置。级别:全局。
REPLICATION SLAVE	允许副本从源读取二进制日志。级 别:全局。
SELECT	启用使用 SELECT 。级别:全局、数据 库、表、列。
SHOW DATABASES	启用SHOW DATABASES以显示所有数据库。级别:全局。
SHOW VIEW	启用SHOW CREATE VIEW。级别: 全局、数据库、表。
SHUTDOWN	启用mysqladmin shutdown 的使用。级别:全局。
SUPER	启用其他管理操作,例如 CHANGE MASTER TO、 KILL、 PURGE BINARY LOGS、 SET GLOBAL和 mysqladmin debug命令。级别:全 局。
TRIGGER	启用触发器操作。级别:全局、数据 库、表。
UPDATE	启用UPDATE。 级别:全局、数据库、表、列。
USAGE	等价于"没有特权"。

M-Compatibility以级别划分支持以下权限:

表 4-62 M-Compatibility 支持的赋权类型

授权对象	支持授予的权限
模式	CREATE、USAGE、ALTER、DROP、 COMMENT
表、视图	SELECT、INSERT、UPDATE、 DELETE、TRUNCATE、 REFERENCES、TRIGGER、ALTER、 DROP、COMMENT、INDEX、 VACUUM

授权对象	支持授予的权限
列	SELECT、INSERT、UPDATE、 REFERENCES、COMMENT
序列	SELECT、USAGE、UPDATE、 ALTER、DROP、COMMENT

- MySQL中通过'dbname.*'表示模式层级的授权对象;在M-Compatibility中,使用'{DATABASE | SCHEMA} dbname'表示模式层级的授权对象。
- MySQL中用户名为两部分: 用户名@主机名; M-Compatibility当前仅支持用户名。
- MySQL支持在GRANT赋权语法中修改用户验证,安全连接,资源参数属性,即 auth_option、tls_option和resource option; M-Compatibility赋权语法中不支持 以上特性,需使用CREATE USER、ALTER USER设置用户相关属性。
- MySQL支持用户代理赋权,GRANT PROXY ON主要用于对多个用户进行统一的权限管理。MySQL5.7未提供角色机制,而在MySQL8.0和M-Compatibility中都提供了角色机制。角色能满足用户对于多个用户权限统一管控的目标,可以替代GRANT PROXY ON。
- M-Compatibility拥有public的概念,所用用户都拥有public的权限,部分系统表、系统视图可供所有用户查询。用户可以对public所拥有的权限进行grant和revoke; MySQL中,新创建的用户只拥有全局的usage权限,这个权限很小,几乎为0,只有连接数据库和查询information_schema数据库的权限。
- M-Compatibility中,对象的所有者缺省具有该对象上的所有权限,出于安全考虑 所有者可以舍弃部分权限,但ALTER、DROP、COMMENT、INDEX、VACUUM以 及对象的可再授予权限属于所有者固有的权限,隐式拥有; MySQL中,没有 owner的概念,即使用户创建了表,如果没赋予用户对应权限,那么用户也不能 对其创建的表进行IUD等操作。
- 在MySQL中, USAGE实际上表示无权限,所用用户都拥有该权限,当执行revoke 或grant usage时,实际上不会进行任何修改;在M-Compatibility中,USAGE权 限如下:
 - 对于模式,USAGE允许访问包含在指定模式中的对象,若没有该权限,则只能看到这些对象的名称。
 - 对于序列,USAGE允许使用nextval函数 。
- 在M-Compatibility中,支持给用户设置管理员角色,包括系统管理员(SYSADMIN)、安全管理员(CREATEROLE)、审计管理员(AUDITADMIN)、监控管理员(MONADMIN)、运维管理员(OPRADMIN)、安全策略管理员(POLADMIN)。默认情况下拥有SYSADMIN属性的系统管理员,具备系统最高权限。三权分立后,系统管理员将不再具有CREATEROLE属性(安全管理员)和AUDITADMIN属性(审计管理员)能力,即不再拥有创建角色和用户的权限,也不再拥有查看和维护数据库审计日志的权限;在MySQL中,不支持该用户设置管理员角色,也没有三权分立相关设计。
- 在M-Compatibility中,可以给用户赋予ANY权限,表示用户能够在非系统模式下 拥有对应的权限,包括CREATE ANY TABLE、SELECT ANY TABLE、CREATE ANY INDEX等;在MySQL中,不支持ANY权限的赋予。
- MySQL中提供SHOW GRANTS查询用户权限; M-Compatibility中,可以通过gsql客户端元命令'\l+'、'\dn+'、'\dp'查询权限信息,也可以通过查询pg_namespace、pg_class、pg_attribute等系统表的权限相关字段查询权限信息。

- MySQL中数据库、表、列被删除时,相关的授权信息在系统表中依然保留,如果 重新创建同名对象用户依然拥有权限; M-Compatibility中当数据库、表、列被删 除时,相关的授权信息会被删除,在重新创建同名对象后需要重新授权。
- MySQL在授予数据库层级的权限时,支持'_'和'%'对数据库名进行模糊匹配; M-Compatibility不支持对象名模糊匹配, '_'或'%'等特殊字符被识别为普通字符。
- MySQL中,GRANT语句中指定用户不存在时默认会创建该账户(此特性已在 MySQL8.0中移除);M-Compatibility不支持给未创建用户赋权。

4.3.8.8 系统表和系统视图

表 4-63 M-Compatibility 与 GaussDB 的系统表或系统视图的差异

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
1	information_schema. columns	generation_expression	该字段输出结果因涉及M-Compatibility与MySQL的表达式的字符串拼接逻辑的不同而存在差异。
2	information_schema. columns	data_type	该字段输出结果因涉及M-Compatibility的数据类型format_type输出,目前未修改,与MySQL存在差异。
3	information_schema. columns	column_type	该字段输出结果因涉及M- Compatibility的数据类型 format_type输出,目前未 修改,与MySQL存在差 异。
4	information_schema. tables	engine	M-Compatibility中: • ENGINE对齐 information_schema.e ngines数据。 • 部分系统表ENGINE为空。 • 在默认为ASTORE表且 未指定STORAGE_TYPE 时ENGINE为空。
5	information_schema. tables	version	M-Compatibility中不支持 该字段。
6	information_schema. tables	row_format	M-Compatibility中不支持 该字段。

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
7	information_schema. tables	avg_row_length	M-Compatibility下表示使 用数据文件除以所有元组 数(包括活元组和死元 组)的结果。表中没有元 组,值为null。
8	information_schema. tables	max_data_length	M-Compatibility中不支持 该字段。
9	information_schema. tables	data_free	M-Compatibility中表示死 元组在总元组中的比例乘 以数据文件大小。如果表 中没有元组,则为null。
10	information_schema. tables	check_time	M-Compatibility中不支持 该字段。
11	information_schema. tables	create_time	在M-Compatibility下,此字段与MySQL行为表现有差异,对于创建视图的情形MySQL中该字段置null,M-Compatibility则显示实际的创建表时间。数据库自带的表,视图设置null。
12	information_schema. tables	update_time	M-Compatibility数据库自 带的表,视图设置null。
13	information_schema. tables	table_collation	在M-Compatibility下,此字段与MySQL行为表现有差异。如果表指定的是视图,则为null。如果指定的表在创建时,未使用COLLATE子句指定列的排序规则,则为null。
14	information_schema. statistics	collation	M-Compatibility只有值 A、D,不会是NULL。
15	information_schema. statistics	packed	M-Compatibility中不支持 该字段。
16	information_schema. statistics	sub_part	M-Compatibility中不支持 该字段。
17	information_schema. statistics	comment	M-Compatibility中不支持 该字段。
18	information_schema. partitions	subpartition_name	M-Compatibility中,如果 分区不是子分区,则为 null。

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
19	information_schema. partitions	subpartition_ordinal_p osition	M-Compatibility中,如果 分区不是子分区,则为 null。
20	information_schema. partitions	partition_method	M-Compatibility中, 分区策略。如果分区不是 一级分区,则为null。 • 'r': 范围分区。 • 'i': 间隔分区。 • 'l': list分区。 • 'h': hash分区。
21	information_schema. partitions	subpartition_method	M-Compatibility中, 子分区策略。如果分区不 是二级分区,则为null。 • 'r': 范围分区。 • 'i': 间隔分区。 • 'l': list分区。 • 'h': hash分区。
22	information_schema. partitions	partition_description	M-Compatibility中,是区 分一级分区和二级分区 的。
23	information_schema. partitions	partition_expression	M-Compatibility中不支持 该字段。
24	information_schema. partitions	subpartition_expressio n	M-Compatibility中不支持 该字段。
25	information_schema. partitions	data_length	M-Compatibility中不支持 该字段。
26	information_schema. partitions	max_data_length	M-Compatibility中不支持 该字段。
27	information_schema. partitions	index_length	M-Compatibility中不支持 该字段。
28	information_schema. partitions	data_free	M-Compatibility中不支持 该字段。
29	information_schema. partitions	create_time	M-Compatibility中不支持 该字段。
30	information_schema. partitions	update_time	M-Compatibility中不支持 该字段。

序号	系统表或系统视图	差异列	M-Compatibility与 MySQL的差异
31	information_schema. partitions	check_time	M-Compatibility中不支持 该字段。
32	information_schema. partitions	checksum	M-Compatibility中不支持 该字段。
33	information_schema. partitions	partition_comment	M-Compatibility中不支持 该字段。
34	information_schema. partitions	nodegroup	M-Compatibility中不支持 该字段。

□ 说明

- 视图中对于整型的类型回显,不支持指定精度范围。如MySQL的bigint(1), M-Compatibility下对应的是bigint类型, MySQL中bigint(21) unsigned, 在M-Compatibility下对应的是bigint unsigned类型。
- MySQL中int类型,在M-Compatibility中是integer类型。
- m_schema.columns_priv视图的Column_priv字段、m_schema.tables_priv视图的
 Table_priv,Column_priv字段、m_schema.procs_priv视图的Routine_type,Proc_priv字段、
 m_schema.proc视图的
 type,language,sql_data_access,is_deterministic,security_type,sql_mode字段、
 m_schema.func视图的type字段均不支持,在此版本中不予显示。
- 由于information_schema.tables、information_schema.statistics中table_rows、avg_row_length、data_length、data_free、index_length、cardinality基于统计信息获取,查看前请先执行ANALYZE,更新统计信息后再查看(如果数据库中更新数据,建议延迟执行ANALYZE)。
- information_schema.statistics包含的索引列需要是创建索引中索引列是完整的表列,如果索引列是表达式,不在这个视图中。
- information_schema.partitions 中table_row, avg_row_length基于统计信息获取,查看前请 先执行ANALYZE,更新统计信息后再查看(如果数据库中更新数据,建议延迟执行 ANALYZE)。
- information_schema.partitions中一级分区和二级分区是分开呈现。
- 对于支持的grantee字段,MySQL的格式是'*user_name*'@'*host_name*',在M-Compatibility数据库,是被授予权限的用户或角色的名称。
- 对于支持的host字段,在M-Compatibility数据库,返回当前节点的hostname。
- m_schema.tables_priv、information_schema.user_privileges、information_schema.schema_privileges, information_schema.table_privileges、information_schema.column_privileges、m_schema.columns_priv、m_schema.func、m_schema.procs_priv 在MySQL下需要授权后才能查看视图内容,M-Compatibility数据库可以根据默认权限查看到对应的内容。如对于表t1,在MySQL下需要先对t1给对应的用户授权,才能在权限视图中看到对应的权限信息,M-Compatibility数据库下则可以直接在视图中看到t1表相关的权限信息。
- m_schema中的系统视图,但在MySQL是系统表。
- information_schema.views中的VIEW_DEFINITION以及information_schema.routines中的 ROUTINE_DEFINITION不做字符序控制。
- 《 M-Compatibility开发指南》中"Schema"章节所列举的字符类型的视图字段,字符集使用utf8mb4,字符序使用utf8mb4_bin或utf8mb4_general_ci,字符序优先级为《 M-Compatibility开发指南》中"SQL参考>字符集与字符序>字符集和字符序合并规则"所描述的"支持字符序的数据类型的列"的优先级,和MySQL存在差异。

4.3.9 驱动

4.3.9.1 ODBC

4.3.9.1.1 ODBC 接口参考

获取参数描述信息

SQLDescribeParam接口是ODBC API中的一个函数,用于获取与预处理SQL语句(如调用SQLPrepare)相关参数的描述信息。它可以返回参数的类型、大小、是否允许NULL值等元数据,这对于动态构建SQL语句和绑定参数非常有用。

原型

SQLRETURN SQLDescribeParam(
SQLHSTMT StatementHandle,
SQLUSMALLINT ParameterNumber,
SQLSMALLINT *DataTypePtr,
SQLULEN *ParameterSizePtr,
SQLSMALLINT *DecimalDigitsPtr,
SQLSMALLINT *NullablePtr);

表 4-64 SQLDescribeParam 参数说明

参数名	参数说明	差异
StatementHand le	语句句柄。	-
ParameterNum ber	参数序号,起始为1,依次递增。	-
DataTypePtr	指向返回参数数据类型的指针。	MySQL ODBC对于任意类型均返回SQL_VARCHAR。
		GaussDB ODBC的会根据内 核返回的不同类型判断返回 给应用相应的DataType类 型。
ParameterSizeP tr	指向返回参数大小的指针。	MySQL ODBC若允许ODBC 驱动程序使用更大的数据包 进行数据传输,则返回 24M,否则返回255。
		GaussDB ODBC根据实际类型返回参数大小。
DecimalDigitsPt r	指向返回参数十进制位数的指针。	-

参数名	参数说明	差异
NullablePtr	指向返回参数是否允许NULL值的 指针。	MySQL ODBC直接返回 SQL_NULLABLE_UNKNOW N。 GaussDB ODBC直接返回 SQL_NULLABLE。

4.3.10 计划外应用无损透明

4.3.10.1 流程控制函数

表 4-65 流程控制函数列表

MySQL数据库	支持计划外ALT
IF()	支持
IFNULL()	支持
NULLIF()	支持

4.3.10.2 日期和时间函数

表 4-66 日期与和时间函数列表

MySQL数据库	支持计划外ALT
ADDDATE()	支持
ADDTIME()	支持
CONVERT_TZ()	支持
CURDATE()	支持
CURRENT_DATE()/CURRENT_DATE	支持
CURRENT_TIME()/CURRENT_TIME	支持
CURRENT_TIMESTAMP()/ CURRENT_TIMESTAMP	支持
CURTIME()	支持
DATE()	支持
DATE_ADD()	支持
DATE_FORMAT()	支持

MySQL数据库	支持计划外ALT
DATE_SUB()	支持
DATEDIFF()	支持
DAY()	支持
DAYNAME()	支持
DAYOFMONTH()	支持
DAYOFWEEK()	支持
DAYOFYEAR()	支持
EXTRACT()	支持
FROM_DAYS()	支持
FROM_UNIXTIME()	支持
GET_FORMAT()	支持
HOUR()	支持
LAST_DAY()	支持
LOCALTIME()/LOCALTIME	支持
LOCALTIMESTAMP/ LOCALTIMESTAMP()	支持
MAKEDATE()	支持
MAKETIME()	支持
MICROSECOND()	支持
MINUTE()	支持
MONTH()	支持
MONTHNAME()	支持
NOW()	支持
PERIOD_ADD()	支持
PERIOD_DIFF()	支持
QUARTER()	支持
SEC_TO_TIME()	支持
SECOND()	支持
STR_TO_DATE()	支持
SUBDATE()	支持

MySQL数据库	支持计划外ALT
SUBTIME()	支持
SYSDATE()	支持
TIME()	支持
TIME_FORMAT()	支持
TIME_TO_SEC()	支持
TIMEDIFF()	支持
TIMESTAMP()	支持
TIMESTAMPADD()	支持
TIMESTAMPDIFF()	支持
TO_DAYS()	支持
TO_SECONDS()	支持
UNIX_TIMESTAMP()	支持
UTC_DATE()	支持
UTC_TIME()	支持
UTC_TIMESTAMP()	支持
WEEK()	支持
WEEKDAY()	支持
WEEKOFYEAR()	支持
YEAR()	支持
YEARWEEK()	支持

4.3.10.3 字符串函数

表 4-67 字符串函数列表

MySQL数据库	支持计划外ALT
ASCII()	支持
BIT_LENGTH()	支持
CHAR_LENGTH()	支持
CHARACTER_LENGTH()	支持
CONCAT()	支持

MySQL数据库	支持计划外ALT
CONCAT_WS()	支持
HEX()	支持
LENGTH()	支持
LPAD()	支持
MD5()	支持
RANDOM_BYTES()	支持
REPEAT()	支持
REPLACE()	支持
RPAD()	支持
SHA()/SHA1()	支持
SHA2()	支持
SPACE()	支持
STRCMP()	支持
FIND_IN_SET()	支持
LCASE()	支持
LEFT()	支持
LOWER()	支持
LTRIM()	支持
REVERSE()	支持
RIGHT()	支持
RTRIM()	支持
SUBSTR()	支持
SUBSTRING()	支持
SUBSTRING_INDEX()	支持
TRIM()	支持
UCASE()	支持
UPPER()	支持
UNHEX()	支持
FIELD()	支持
COMPRESS()	支持

MySQL数据库	支持计划外ALT
UNCOMPRESS()	支持
UNCOMPRESS_LENGTH()	支持
EXPORT_SET()	支持
POSITION()	支持
LOCATE()	支持
CHAR()	支持
ELT()	支持
FORMAT()	支持
BIN()	支持
MAKE_SET()	支持
TO_BASE64()	支持
FROM_BASE64()	支持
ORD()	支持
MID()	支持
QUOTE()	支持
INSERT()	支持
INSTR()	支持

4.3.10.4 强制转换函数

表 4-68 强制转换函数列表

MySQL数据库	支持计划外ALT
CAST()	支持
CONVERT()	支持

4.3.10.5 加密函数

表 4-69 加密函数列表

MySQL数据库	支持计划外ALT
AES_DECRYPT()	支持

MySQL数据库	支持计划外ALT
AES_ENCRYPT()	支持
PASSWORD()	支持

4.3.10.6 比较函数

表 4-70 比较函数列表

MySQL数据库	支持计划外ALT
COALESCE()	支持
INTERVAL()	支持
GREATEST()	支持
LEAST()	支持
ISNULL()	支持

4.3.10.7 聚合函数

表 4-71 聚合函数列表

MySQL数据库	支持计划外ALT
AVG()	支持
BIT_AND()	支持
BIT_OR()	支持
BIT_XOR()	支持
COUNT()	支持
GROUP_CONCAT()	支持
MAX()	支持
MIN()	支持
SUM()	支持
STD()	支持

4.3.10.8 JSON 函数

表 4-72 JSON 函数列表

MySQL数据库	支持计划外ALT
JSON_APPEND()	支持
JSON_ARRAY()	支持
JSON_ARRAY_APPEND()	支持
JSON_ARRAY_INSERT()	支持
JSON_CONTAINS()	支持
JSON_CONTAINS_PATH()	支持
JSON_DEPTH()	支持
JSON_EXTRACT()	支持
JSON_INSERT()	支持
JSON_KEYS()	支持
JSON_LENGTH()	支持
JSON_MERGE()	支持
JSON_MERGE_PATCH()	支持
JSON_MERGE_PRESERVE()	支持
JSON_OBJECT()	支持
JSON_QUOTE()	支持
JSON_REMOVE()	支持
JSON_REPLACE()	支持
JSON_SEARCH()	支持
JSON_SET()	支持
JSON_TYPE()	支持
JSON_UNQUOTE()	支持
JSON_VALID()	支持

4.3.10.9 窗口函数

表 4-73 窗口函数列表

MySQL数据库	支持计划外ALT
LAG()	支持
LEAD()	支持
ROW_NUMBER()	支持

4.3.10.10 数字操作函数

表 4-74 数字操作函数列表

MySQL数据库	支持计划外ALT
ABS()	支持
ACOS()	支持
ASIN()	支持
ATAN()	支持
ATAN2()	支持
CEILING()	支持
COS()	支持
DEGREES()	支持
EXP()	支持
FLOOR()	支持
LN()	支持
LOG()	支持
LOG10()	支持
LOG2()	支持
PI()	支持
POW()	支持
POWER()	支持
RAND()	支持
SIGN()	支持
SIN()	支持

MySQL数据库	支持计划外ALT
SQRT()	支持
TAN()	支持
TRUNCATE()	支持
CEIL()	支持
CRC32()	支持
CONV()	支持

4.3.10.11 网络地址函数

表 4-75 网络地址函数列表

MySQL数据库	支持计划外ALT
INET_ATON()	支持
INET_NTOA()	支持
INET6_ATON()	支持
INET6_NTOA()	支持
IS_IPV6()	支持
IS_IPV4()	支持

4.3.10.12 其他函数

表 4-76 其他函数列表

MySQL数据库	支持计划外ALT
DATABASE()	支持
UUID()	支持
UUID_SHORT()	支持
ANY_VALUE()	支持
SLEEP()	支持
COLLATION()	支持
FOUND_ROWS()	支持
ROW_COUNT()	支持

MySQL数据库	支持计划外ALT
SYSTEM_USER()	支持
DEFAULT()	支持
BENCHMARK()	支持