

# openGauss SQL兼容性需求check-in评审报告

---

## 评审纪要

---

时间：2022年9月19日

特性名称：openGauss兼容MySQL日期处理函数

特性责任人：

评审人：

## 设计checklist

---

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
GUC参数	功能	新增GUC参数适配项	①guc.cpp里需要新增guc参数的相应代码，包括参数的校验/赋值等功能函数；②如果要加入postgres.conf，需要修改postgresql_single.conf.sample 路径：src/common/backend/utis/misc/ 一般以注释方式加入，如果非常确认使用某个默认值，也可以以非注释方式加入 ③如果要允许gs_guc工具设置，请修改cluster_guc.conf 路径：src/bin/g_s_guc/cluster_guc.conf	否	
升级	兼容性	版本号变更	src/common/backend/utis/init/globals.cpp : GRAND_VERSION_NUM	否	
升级	兼容性	前向兼容性	1) 涉及系统表修改的特性，必须提供系统表升级脚本和回滚脚本，修改版本号文件 2) 对于涉及修改持久化数据（如日志）的特性，必须考虑新老版本共存时的兼容性场景，必要情况下，需要增加版本号以进行识别 3) 对于涉及修改执行态数据格式（如syscache结构）的特性，必须考虑新老版本共存时的兼容性场景，必要情况下，需要增加版本号以进行识别	否	
升级	兼容性	验证升级正常	涉及到升级时需要验证如下场景： 1. 集群环境安装：该步骤会验证集群环境安装是否正常 2. 集群环境升级：从老的版本（1.1.0版本）升级到最新版本是否正常	否	
公共数据结构	功能	hashtable	hashtable使用注意事项 ①在使用hash表时，entry必须封装成一个结构体，第一个成员必须是hash key 否则会造成查找失败！！ ②封装的结构体必须只有两个成员：key+value，即value需要封装起来 否则hash表在search的时候会core！！ ③hash表在创建的时候，entry的size一定要正确 否则在enter的时候会造成写越界，不会报错但是其它变量内存会被踩，不易发现！！	否	
公共数据结构	功能	禁止使用commandTag判断Query的类型	需要对SELECT\INSERT\UPDATE\DELETE\MERGE INTO等query类型进行识别判断时，应该使用CmdType进行判断或新增标识。禁止使用commandTag作为判断标识，因为commandTag在带returning*语句执行中变化。	否	
系统函数	兼容性	系统表升级是否正确	通过对比initdb和升级后的db中pg_proc每一列的值，避免出现升级和非升级不一致的情况	否	

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
系统函数	兼容性	新增数据类型适配FENCED函数	新增数据类型如果是FENCED函数使用，需要修改UDFArgumentHandler函数，把数据类型添加到对应收发分支。	否	
系统函数	功能	NULL值处理	对于非strict函数允许接收NULL入参，在函数逻辑中应对NULL值进行细致的处理	是	是
前向兼容性	兼容性	Node相关数据结构删减	禁止对Node相关数据结构（如Query、RangeTblEntry）数据成员删减	否	
兼容性	兼容性	兼容性设计原则	目前支持兼容性PG,A,B,C;对于每一个兼容点要求： 1、如果该兼容点为非冲突兼容点，应该是各个兼容模式下均支持，不需要开关控制 2、如果该兼容性点为冲突兼容点，应该由兼容开关控制。	是	是
查询解析	兼容性	添加关键字场景	为了尽可能的防止移进规约冲突，最大化的用关键字作为对象名，关键字分有4类：非保留关键字、列名关键字、类型函数关键字、保留关键字。他们的约束逐渐加强，比如保留关键字不可以用做表名。添加关键字需要考虑如下： 1. 添加新的关键字最好先作为非保留关键字添加 2. 如果要加其他关键字类型那么需要考虑是否有前向兼容问题。比如排查客户场景是否有用该关键字作为对象名。 3. 关键字还需要加入kwlist.h文件中。	否	
查询解析	兼容性	Gram.y中处理语法冲突	1、合理使用a_expr/b_expr/c_expr使用场景 为了尽可能防止冲突，划分了a_expr/b_expr/c_expr，比如BETWEEN a_expr AND b_expr中 AND属于a_expr范畴，b_expr中不含AND。否则就无法区分BETWEEN之后是Bool判断还是范围。c_expr是a_expr和b_expr的交集。所以 尽量用a_expr这样支持的范围最广。如果用b_expr/c_expr 那需要考虑潜在的用户问题场景（参考BETWEEN...AND...例子）并在手册中写清楚约束。	否	
其他	功能	新增已有的数据结构成员适配序列化与反序列化	注意适配： src/common/backend/nodes/copyfuncs.cpp src/common/backend/nodes/equalfuncs.cpp src/common/backend/nodes/outfuncs.cpp src/common/backend/nodes/readfuncs.cpp	否	

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
公共子系统	内存定位定界	局部变量内存释放，非程序结尾位置，释放进行置空；全局内存变量释放，无论什么位置均需要置空	防止内存使用出现use-after-free模式，内存越界	否	
公共子系统	内存定位定界	对于使用数组、下标访问内存操作场景，是否提前判断下标符合内存空间要求	防止出现heap overflow场景	否	
公共子系统	内存定位定界	内存分配报错日志中，是否需要新增信息，便于分析报错原因	内存报错场景下，快速定位内存报错原因	否	
资源负载管理	兼容性	只能新增系统表参数，不可以删除已有的参数	前向兼容	否	
资源负载管理	功能	新增系统表参数的默认初始化数值要合理，不影响已有的功能	前向兼容	否	
HA用例	功能	ha check	1.HA相关设计实现需要严格系统的测试。 2.需要保证最简单的功能验证，包括搭建主备环境，确认主备可以正常同步，以及跑ha check用例，确认所有用例可以跑过。 3.需要完成各种故障场景的验证，设计详细的故障库，确保覆盖各个场景。	否	
安全	未公开接口	是否新增或修改函数、视图、系统表	新增或修改函数、视图、系统表需刷新产品文档，考虑权限控制	是	是



```

set b_compatibility_mode = true;
-- 边界值, 坏值
select timestampdiff(microsecond, '0000-1-1 00:00:00', '0000-1-2');
select timestampdiff(second, '0000-1-1 00:00:00', '0000-1-2');
select timestampdiff(minute, '0000-1-1 00:00:00', '0000-1-2');
select timestampdiff(hour, '0000-1-1 00:00:00', '0000-1-2');
select timestampdiff(day, '9999-11-12 00:00:00', '9999-11-11');
select timestampdiff(day, '10000-1-1 00:00:00', '9999-12-31');
select timestampdiff(year, '9999-12-31', '0000-12-31');
select timestampdiff(quarter, '9999-12-31', '0000-12-31');
select timestampdiff(month, '9999-12-31', '0000-12-31');
select timestampdiff(week, '9999-12-31', '0000-12-31');
select timestampdiff(microsecond, time'-838:59:59', time'838:59:59');
select timestampdiff(second, time'-838:59:59', time'838:59:59');
select timestampdiff(minute, time'-838:59:59', time'838:59:59');
select timestampdiff(hour, time'-838:59:59', time'838:59:59');
select timestampdiff(day, time'-838:59:59', time'838:59:59');
select timestampdiff(month, time'-838:59:59', time'838:59:59');
select timestampdiff(quarter, time'-838:59:59', time'838:59:59');
select timestampdiff(year, time'-838:59:59', time'838:59:59');
select timestampdiff(microsecond, '0000-0-1', '0000-1-2');
select timestampdiff(microsecond, '9999-0-1', '0000-1-2');
-- 混合类型
select timestampdiff(microsecond, '2022-9-12', time'12:12:12');
select timestampdiff(microsecond, time'12:12:12', '2022-9-12');
select timestampdiff(microsecond, datetime'2022-9-12 12:12:12.999999',
time'12:12:12.999999');
select timestampdiff(microsecond, time'12:12:12.999999', datetime'2022-9-12
12:12:12.999999');
select timestampdiff(microsecond, date'2022-9-12', time'12:12:12.999999');
select timestampdiff(microsecond, time'12:12:12.999999', date'2022-9-12');
set b_compatibility_mode = false;

-- test convert_tz
-- 双字母, 第一次转换超[1970-1-1 00:00:01.000000, 2038-01-19 03:14:07.999999], 结果不
转换
select convert_tz('1970-1-1 00:00:01.000000', 'MET', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'CCT', 'GMT');
select convert_tz('2038-01-19 03:14:07.999999', 'EDT', 'GMT');
-- 双字母正常转换
select convert_tz('1234-1-1 00:00:00', 'MET', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'GMT', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'EDT', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'CDT', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'EST', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'MDT', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'MST', 'GMT');
select convert_tz('2038-01-19 03:14:07.999999', 'JST', 'GMT');
select convert_tz('2038-01-19 03:14:07.999999', 'KST', 'GMT');
select convert_tz('2038-01-19 03:14:07.999999', 'CCT', 'GMT');
-- 双数字, 第一次转换超[1970-1-1 00:00:01.000000, 2038-01-19 03:14:07.999999], 结果不
转换
select convert_tz('1970-1-1 00:00:01.000000', '+1:00', '+0:00');
select convert_tz('2038-01-19 03:14:07.999999', '-1:00', '+0:00');
-- 双数字正常转换

```

```

select convert_tz('1970-1-1 00:00:01.000000', '-1:00', '+0:00');
select convert_tz('1970-1-1 00:00:01.000000', '-2:00', '+0:00');
select convert_tz('1970-1-1 00:00:01.000000', '-10:00', '+0:00');
select convert_tz('2038-01-19 03:14:07.999999', '+1:00', '+0:00');
select convert_tz('2038-01-19 03:14:07.999999', '+2:00', '+0:00');
select convert_tz('2038-01-19 03:14:07.999999', '+10:00', '+0:00');
-- 字母数字混合
select convert_tz('1970-1-1 00:00:01.000000', '-1:00', 'GMT');
select convert_tz('1970-1-1 00:00:01.000000', 'GMT', '+3:00');
select convert_tz('2038-01-19 03:14:07.999999', '+2:00', 'GMT');
select convert_tz('2038-01-19 03:14:07.999999', 'GMT', '+7:00');

-- test adddate(date,INTERVAL expr unit)
-- 边界值, 坏值
select adddate('0001-01-01',INTERVAL 1 day);
select adddate('0001-02-28',INTERVAL 1 day);
select adddate('0004-02-29',INTERVAL 1 day);
select adddate('0001-12-31',INTERVAL 1 month);
select adddate('9999-12-30',INTERVAL 1 day);
select adddate('0000-01-01',INTERVAL 1 day);
select adddate(2001-01-01,INTERVAL 1 day);
select adddate('9999-12-12',INTERVAL 1 year);
-- 进位
select adddate('0001-01-31',INTERVAL 1 day);
select adddate('0001-2-28',INTERVAL 1 day);
select adddate('0001-12-31',INTERVAL 1 day);
select adddate('2021-12-31',INTERVAL 1 day);
select adddate('9998-12-31',INTERVAL 1 day);
select adddate('0001-12-31',INTERVAL 1 month);
select adddate('1999-12-31',INTERVAL 1 day);
-- 不同类型输入
select adddate(date'001-01-01',INTERVAL 1 day);
select adddate(datetime'001-01-01 12:21:21',INTERVAL 1 day);
select adddate(time'1:1:1',INTERVAL 1 day);
select adddate('2001-12-12',INTERVAL 1 day);
select adddate('2001-12-12 12:12:12',INTERVAL 1 day);

-- test date_add(date,INTERVAL expr unit)
-- 边界值, 坏值
select date_add('0001-01-01',INTERVAL 1 day);
select date_add('0001-02-28',INTERVAL 1 day);
select date_add('0004-02-29',INTERVAL 1 day);
select date_add('0001-12-31',INTERVAL 1 month);
select date_add('9999-12-30',INTERVAL 1 day);
select date_add('0000-01-01',INTERVAL 1 day);
select date_add(2001-01-01,INTERVAL 1 day);
select date_add('9999-12-12',INTERVAL 1 year);
-- 进位
select date_add('0001-01-31',INTERVAL 1 day);
select date_add('0001-2-28',INTERVAL 1 day);
select date_add('0001-12-31',INTERVAL 1 day);
select date_add('2021-12-31',INTERVAL 1 day);
select date_add('9998-12-31',INTERVAL 1 day);
select date_add('0001-12-31',INTERVAL 1 month);
select date_add('1999-12-31',INTERVAL 1 day);

```

```

-- 不同类型输入
select date_add(date'001-01-01',INTERVAL 1 day);
select date_add(datetime'001-01-01 12:21:21',INTERVAL 1 day);
select date_add(time'1:1:1',INTERVAL 1 day);
select date_add('2001-12-12',INTERVAL 1 day);
select date_add('2001-12-12 12:12:12',INTERVAL 1 day);

-- test date_sub(date,INTERVAL expr unit)
-- 边界值, 坏值
select date_sub('0001-01-01',INTERVAL '-1' day);
select date_sub('0001-02-28',INTERVAL '-1' day);
select date_sub('0004-02-29',INTERVAL '-1' day);
select date_sub('0001-12-31',INTERVAL '-1' month);
select date_sub('9999-12-30',INTERVAL '-1' day);
select date_sub('0000-01-01',INTERVAL '-1' day);
select date_sub(2001-01-01,INTERVAL '-1' day);
select date_sub('9999-12-12',INTERVAL '-1' year);
-- 进位
select date_sub('0001-01-31',INTERVAL '-1' day);
select date_sub('0001-2-28',INTERVAL '-1' day);
select date_sub('0001-12-31',INTERVAL '-1' day);
select date_sub('2021-12-31',INTERVAL '-1' day);
select date_sub('9998-12-31',INTERVAL '-1' day);
select date_sub('0001-12-31',INTERVAL '-1' month);
select date_sub('1999-12-31',INTERVAL '-1' day);
-- 不同类型输入
select date_sub(date'001-01-01',INTERVAL '-1' day);
select date_sub(datetime'001-01-01 12:21:21',INTERVAL '-1' day);
select date_sub(time'1:1:1',INTERVAL '-1' day);
select date_sub('2001-12-12',INTERVAL '-1' day);
select date_sub('2001-12-12 12:12:12',INTERVAL '-1' day);

-- test addtime(expr1,expr2)
-- 边界值, 坏值
select addtime('0000-01-01 0:0:0','20:20:30');
select addtime('0000-01-01 00:00:00.000000','00:00:00.000000');
select addtime('9999-12-31 23:59:59.999999','0:0:0');
select addtime(0000-01-01 00:00:00.000000,'00:00:00.000000');
select addtime('0000-01-01 00:00:00.000000',00:00:00.000000);
select addtime('0000-01-01 00:00:00.000000','2001-01-01');
-- 进位
select addtime('2000-12-31 23:59:59','0:0:1');
select addtime('2000-12-31 23:59:59.9','0:0:0.1');
select addtime('2000-1-31 23:59:59.9','0:0:0.1');
select addtime('2000-1-31 23:59:59','0:0:1');
-- 不同类型输入
select addtime('11:22:33','10:20:30');
select addtime('2020-03-04 11:22:33','10:20:30');
select addtime(datetime'2020-03-04 11:22:33','10:20:30');
select addtime(time'11:22:33','10:20:30');
select addtime(datetime'2020-03-04 11:22:33',time'10:20:30');
select addtime('11:22:33',time'10:20:30');

-- test adddate(date/datetime/time,days)
-- 边界值, 坏值

```



```

select adddate('0001-01-01',999);
select adddate('0001-02-28',1000000);
select adddate('0004-02-29',1);
select adddate('9999-12-30',1);
select adddate('9999-12-31',0);
select adddate('9999-12-31',1);
select adddate('10000-12-31',1);
select adddate('0000-01-01',-1);
select adddate(2001-01-01,1);
select adddate('12:12:12',0.5);
select adddate('12:12:12',45);
select adddate(time'132:122:12',-55);
select adddate('132:122:12',5);
select adddate(time'ssssss',5);
select adddate('ssssss',5);
-- 进位
select adddate('0001-01-31',1);
select adddate('0001-2-28',1);
select adddate('0001-12-31',1);
select adddate('2021-12-31',1);
select adddate('9998-12-31',1);
-- 不同类型输入
select adddate(date'0001-01-01',1);
select adddate('0001-01-01',1);
select adddate(datetime'0001-01-01 12:21:21',1);
select adddate('0001-01-01 12:21:21',1);
select adddate(time'1:1:1',1);
select adddate(time'1:1:1',30);
select adddate(time'1:1:1',-30);
select adddate(time'1:1:1',-100);
select adddate('1:1:1',1);
select adddate('2001-12-12',1);
select adddate('2001-12-12 12:12:12',1);

-- test adddate(date/datetime/time,INTERVAL expr unit)
-- 边界值, 坏值
select adddate('0001-01-01',INTERVAL 1 day);
select adddate('0001-02-28',INTERVAL 1 day);
select adddate('0004-02-29',INTERVAL 1 day);
select adddate('0001-12-31',INTERVAL 1 month);
select adddate('9999-12-30',INTERVAL 1 day);
select adddate('0000-01-01',INTERVAL 1 day);
select adddate(2001-01-01,INTERVAL 1 day);
select adddate('9999-12-12',INTERVAL 1 year);
select adddate('0000-00-00',INTERVAL 1 day);
select adddate('1000000-01-00',INTERVAL 1 day);
select adddate('0002-03-01',INTERVAL '-3' year);
select adddate(time'12:12:12',INTERVAL 1 year);
select adddate('12:12:12',INTERVAL 1 year);
-- 进位
select adddate('0001-01-31',INTERVAL 1 day);
select adddate('0001-2-28',INTERVAL 1 day);
select adddate('0001-12-31',INTERVAL 1 day);
select adddate('2021-12-31',INTERVAL 1 day);
select adddate('9998-12-31',INTERVAL 1 day);

```

```

select adddate('0001-12-31',INTERVAL 1 month);
select adddate('1999-12-31',INTERVAL 1 day);
-- 不同类型输入
select adddate(date'001-01-01',INTERVAL 1 day);
select adddate(datetime'001-01-01 12:21:21',INTERVAL 1 day);
select adddate(time'1:1:1',INTERVAL 1 day);
select adddate('2001-12-12',INTERVAL 1 day);
select adddate('2001-12-12 12:12:12',INTERVAL 1 day);

-- test date_add(date,INTERVAL expr unit)
-- 边界值, 坏值
select date_add('0001-01-01',INTERVAL 1 day);
select date_add('0001-02-28',INTERVAL 1 day);
select date_add('0004-02-29',INTERVAL 1 day);
select date_add('0001-01-01',1);
select date_add('0001-02-28',1);
select date_add('0004-02-29',1);
select date_add('0001-12-31',INTERVAL 1 month);
select date_add('9999-12-30',1);
select date_add('0000-01-01',1);
select date_add(2001-01-01,1);
select date_add('9999-12-12',INTERVAL 1 year);
select date_add('9999-12-12 12:12:12',INTERVAL 1 day);
select date_add(time'12:12:12',1);
select date_add('12:12:12',1);
select date_add('133:131:45',5);
-- 进位
select date_add('0001-01-31',INTERVAL 1 day);
select date_add('0001-2-28',INTERVAL 1 day);
select date_add('0001-12-31',INTERVAL 1 day);
select date_add('2021-12-31',INTERVAL 1 day);
select date_add('9998-12-31',INTERVAL 1 day);
select date_add('0001-01-31',INTERVAL 1 day);
select date_add('0001-2-28',1);
select date_add('0001-12-31',1);
select date_add('2021-12-31',1);
select date_add('9998-12-31',1);
select date_add('0001-12-31',INTERVAL 1 month);
select date_add('1999-12-31',INTERVAL 1 day);
-- 不同类型输入
select date_add(date'001-01-01',INTERVAL 1 day);
select date_add(datetime'001-01-01 12:21:21',INTERVAL 1 day);
select date_add(time'1:1:1',INTERVAL 1 day);
select date_add('2001-12-12',INTERVAL 1 day);
select date_add('2001-12-12 12:12:12',INTERVAL 1 day);

-- test date_sub(date,INTERVAL expr unit)
-- 边界值, 坏值
select date_sub('0001-01-01',INTERVAL '-1' day);
select date_sub('0001-02-28',INTERVAL '-1' day);
select date_sub('0004-02-29',INTERVAL '-1' day);
select date_sub('0001-12-31',INTERVAL '-1' month);
select date_sub('9999-12-30',INTERVAL '-1' day);
select date_sub('0000-01-01',INTERVAL '-1' day);
select date_sub(2001-01-01,INTERVAL '-1' day);

```

```

select date_sub('9999-12-12',INTERVAL '-1' year);
select date_sub('2001-01-01',1);
select date_sub('0001-02-28',1);
select date_sub('9999-12-30',-1);
select date_sub('0000-01-01',-1);
select date_sub(2001-01-01,-1);
-- 进位
select date_sub('0001-01-31',INTERVAL '-1' day);
select date_sub('0001-2-28',INTERVAL '-1' day);
select date_sub('0001-12-31',INTERVAL '-1' day);
select date_sub('2021-12-31',INTERVAL '-1' day);
select date_sub('9998-12-31',INTERVAL '-1' day);
select date_sub('0001-01-31',-1);
select date_sub('0001-2-28',-1);
select date_sub('0001-12-31',-1);
select date_sub('2021-12-31',-1);
select date_sub('9998-12-31',-1);
select date_sub('1999-12-31',-1);
-- 不同类型输入
select date_sub(date'001-01-01',INTERVAL '-1' day);
select date_sub(date'001-01-01',-1);
select date_sub(datetime'001-01-01 12:21:21',INTERVAL '-1' day);
select date_sub(time'1:1:1',INTERVAL '-1' day);
select date_sub('2001-12-12',INTERVAL '-1' day);
select date_sub('2001-12-12 12:12:12',INTERVAL '-1' day);

-- test addtime(expr1,expr2)
-- 边界值, 坏值
select addtime('0000-01-01 0:0:0','20:20:30');
select addtime('0000-01-01 0:0:0',1);
select addtime('0000-01-01 00:00:00.000000','00:00:00.000000');
select addtime('0000-01-01 00:00:00.000000',1);
select addtime('9999-12-31 23:59:59.999999','0:0:0');
select addtime('9999-12-31 23:59:59.999999',1);
select addtime(0000-01-01 00:00:00.000000,1);
select addtime('0000-01-01 00:00:00.000000','00:00:00.000000');
select addtime('0000-01-01 00:00:00.000000','2001-01-01');
-- 进位
select addtime('2000-12-31 23:59:59','0:0:1');
select addtime('2000-12-31 23:59:59.9','0:0:0.1');
select addtime('2000-1-31 23:59:59.9','0:0:0.1');
select addtime('2000-1-31 23:59:59','0:0:1');
-- 不同类型输入
select addtime('11:22:33','10:20:30');
select addtime('11:22:33',999);
select addtime('2020-03-04 11:22:33','10:20:30');
select addtime('2020-03-04 11:22:33',1);
select addtime(datetime'2020-03-04 11:22:33','10:20:30');
select addtime(datetime'2020-03-04 11:22:33',0);
select addtime(time'11:22:33','10:20:30');
select addtime(datetime'2020-03-04 11:22:33',time'10:20:30');
select addtime('11:22:33',time'10:20:30');

```

## 回归测试验证

测试集	结论
fastcheck	通过
memcheck	通过
hacheck	

## Fastcheck

```

case_sensitive_test/single_node_triggers .... ok          TIME TAKEN: [ 0.202]
parallel group (2 tests): case_sensitive_test/alter_table case_sensitive_test/reindex_internal
case_sensitive_test/alter_table .... ok          TIME TAKEN: [ 1.988]
case_sensitive_test/reindex_internal .... ok        TIME TAKEN: [ 2.008]
test case_sensitive_test/drop .... ok             TIME TAKEN: [ 0.703]
test case_sensitive_test/clean_env .... ok        TIME TAKEN: [ 0.703]
test case_sensitive_test/role_test .... ok        TIME TAKEN: [ 2.909]
parallel group (3 tests): test_mysql_prepare db_b_rename_user_test export_set
test_mysql_prepare .... ok                       TIME TAKEN: [ 2.810]
db_b_rename_user_test .... ok                   TIME TAKEN: [ 4.114]
export_set .... ok                             TIME TAKEN: [ 5.418]
test test_mysql_operator .... ok                TIME TAKEN: [ 17.855]
test case_sensitive_test/role_test .... ok        TIME TAKEN: [ 3.010]
parallel group (10 tests): json_extract json_array json_object json_array_append json_keys json_contains_path json_unquote json_quote json_contains json_search
json_array .... ok                             TIME TAKEN: [ 4.417]
json_object .... ok                           TIME TAKEN: [ 6.624]
json_quote .... ok                           TIME TAKEN: [ 12.845]
json_contains .... ok                         TIME TAKEN: [ 14.651]
json_contains_path .... ok                   TIME TAKEN: [ 10.236]
json_extract .... ok                         TIME TAKEN: [ 2.910]
json_unquote .... ok                         TIME TAKEN: [ 11.540]
json_keys .... ok                           TIME TAKEN: [ 3.329]
json_search .... ok                         TIME TAKEN: [ 16.957]
json_array_append .... ok                   TIME TAKEN: [ 7.726]
test db_b_date_time_functions3 .... ok          TIME TAKEN: [ 4.814]
test dummy .... ok                           TIME START: [2022-09-17 12:47:04]. TIME TAKEN: [ 0.101] %CPU: [ 0.39] %SCPU: [ 0.00] %MEM: [ 0.00] MEM: [1.14
B]

===== shutting down postmaster                    =====
stop postmaster now!

=====
All 195 tests passed.

```

## Memcheck

本地代码

```

test db_b_date_time_functions3 .... ok          TIME TAKEN: [ 14.045]
test dummy .... ok                           TIME START: [2022-09-17 13:49:58]. TIME TAKEN: [
B]

===== shutting down postmaster                    =====
stop postmaster now!

=====
All 195 tests passed.
Total Time: 1744.945405s
=====

```

Suppressions used:

count	bytes	template
154	7493	regression_main
35248	3524800	get_node_info_name

- #0 0xbb7693 in \_\_interceptor\_memcpy ../../../../libsanitizer/asan/asan\_interceptors.cc
- #1 0x5728bff in memcpy\_s (/home/zql/sda/openGauss-server/contrib/dolphin/tmp\_check/in
- #2 0x4069c7f in heap\_fill\_tuple(tupleDesc\*, unsigned long\*, bool const\*, char\*, unsig
- #3 0x4072813 in heap\_form\_tuple(tupleDesc\*, unsigned long\*, bool\*) /home/zql/sda/open
- #4 0x2f909af in CreateTrigger(CreateTrigStmnt\*, char const\*, unsigned int, unsigned in
- #5 0xffffea0439c9b in createForeignKeyTriggers /home/zql/sda/openGauss-server/contrib/
- #6 0xffffea0431a8f in ATAddForeignKeyConstraint /home/zql/sda/openGauss-server/contrib

官方代码

```
#0 0xa1bc33 in __interceptor_memcpy ../../../../libsanitizer
#1 0x520e1ef in memcpy_s (/home/atwww/sda/openGauss-server/
#2 0x3ebd3ff in heap_fill_tuple(tupleDesc*, unsigned long*,
rver/src/gausskernel/storage/access/common/heaptuple.cpp:240
#3 0x3ec5f93 in heap_form_tuple(tupleDesc*, unsigned long*,
#4 0x2deaf33 in CreateTrigger(CreateTrigStmt*, char const*,
rc/gausskernel/optimizer/commands/trigger.cpp:800
```

## hacheck

# 代码检视结论

## 编码规范检查

### 编程规范

请对照社区安全编码规范进行排查：

<https://gitee.com/opengauss/security/blob/master/guide/SecureCoding.md>



[https://gitee.com/opengauss/security/blob/master/guide/SecureCompile\(C&C++\).md](https://gitee.com/opengauss/security/blob/master/guide/SecureCompile(C&C++).md)

## 内存使用排查

覆盖率：87.5%

### LCOV - code coverage report

Current view: top level	Hit	Total	Coverage
Test: dev3 increment.info	Lines: 232	265	87.5 %
Date: 2022-09-17 00:26:37	Functions: 342	598	57.2 %

Directory	Line Coverage	Functions
/home/atwww/sda/openGauss-server/contrib/dolphin/plugin_parser	 100.0 % 15 / 15	67.0 % 61 / 91
/home/atwww/sda/openGauss-server/contrib/dolphin/plugin_utils/adit	 86.8 % 217 / 250	55.4 % 281 / 507

Generated by: LCOV version 1.14

## 鲲鹏平台乱序排查

编号	排查依据	排查结果
1	是否为多线程并发场景？	否
2	是否涉及线程间共享数据？	否
3	是否未对共享数据加锁保护？	否
4	线程间共享数据是否存在无锁同步模式？	否
5	是否 未 在正确的位置插入合适的内存屏障	否

## 代码检视意见

提出人	意见	答复详情

## 遗留问题

---

## 测试建议

---