





MariaDB Server

MDEV-26433

assertion: table->get_ref_count() == 0 in dict0dict.cc line 1915

▼ Details

Type:	 Bug
Status:	CLOSED (View Workflow)
Priority:	 Blocker
Resolution:	Fixed
Affects Version/s:	10.7.0, 10.5, 10.6, 10.7, 10.8, 10.9
Fix Version/s:	10.5.17 , 10.6.9 , 10.7.5 , (2)
Component/s:	Data Definition - Temporary , (1) Storage Engine - InnoDB
Labels:	None
Environment:	Linux version 5.13.0-1-MANJARO (builduser@LEGION) (gcc (GCC) 11.1.0, GNU ld (GNU Binutils) 2.36.1) #1 SMP PREEMPT Mon Jun 7 06:16:10 UTC 2021 x86_64

▼ Description

step to reproduce:

```
CREATE TEMPORARY TABLE v0 ( v1 TEXT ( 15 ) CHAR SET BINARY NOT NULL NOT NULL UNIQUE
SELECT SQL_CALC_FOUND_ROWS * FROM v0 WHERE v1 IN ( SELECT v3 FROM v0 ) LIMIT 16 ;
DROP PROCEDURE v0 ;
CREATE TABLE v4 ( v6 INT , v5 INT DEFAULT 27 ) ;
ROLLBACK TO SAVEPOINT v4 ;
INSERT INTO v4 VALUES ( + 84 , + 32 , 48 ) ;
```

report (compiled with ASAN):

```
Server version: 10.7.0-MariaDB
key_buffer_size=134217728
read_buffer_size=131072
max_used_connections=1
max_threads=153
thread_count=1
It is possible that mysqld could use up to
key_buffer_size + (read_buffer_size + sort_buffer_size)*max_threads = 467956 K
Hope that's ok; if not, decrease some variables in the equation.

Thread pointer: 0x62b0000bd218
```

```
Attempting backtrace. You can use the following information to find out
where mysqld died. If you see no messages after this, something went
terribly wrong...
stack_bottom = 0x7fb65ba7c850 thread_stack 0x5fc00
sanitizer_common/sanitizer_common_interceptors.inc:4203(__interceptor_backtrace
mysys/stacktrace.c:213(my_print_stacktrace)[0x55df46af9747]
sql/signal_handler.cc:222(handle_fatal_signal)[0x55df45ac1120]
sigaction.c:0(__restore_rt)[0x7fb67ad12870]
```

gdb bt:

```
Using host libthread_db library "/usr/lib/libthread_db.so.1".
Core was generated by `/usr/local/mysql/bin//mysqld --port 10018 --datadir=/hom
Program terminated with signal SIGABRT, Aborted.
#0  0x00007fb67ad0f808 in pthread_kill () from /usr/lib/libpthread.so.0
#1  0x000055df45ac106b in handle_fatal_signal (sig=<optimized out>) at /experim
#2  <signal handler called>
#3  0x00007fb67a7f1d22 in raise () from /usr/lib/libc.so.6
#4  0x00007fb67a7db862 in abort () from /usr/lib/libc.so.6
#5  0x000055df44f00246 in ut_dbg_assertion_failed (expr=expr@entry=0x55df4700bd
#6  0x000055df44f2296e in dict_sys_t::remove (this=<optimized out>, this@entry=
#7  0x000055df463d955f in ha_innobase::delete_table (this=<optimized out>, name
#8  0x000055df45ad0145 in hton_drop_table (hton=<optimized out>, path=<optimize
#9  0x000055df4594c2c8 in THD::rm_temporary_table (this=<optimized out>, base=<
#10 0x000055df4594d237 in THD::free_tmp_table_share (this=<optimized out>, shar
#11 0x000055df459513f5 in THD::drop_temporary_table (this=0x62b0000bd218, table
#12 0x000055df451da338 in drop_open_table (thd=<optimized out>, table=<optimize
#13 0x000055df452837f5 in select_create::abort_result_set (this=0x6290000890c8)
#14 0x000055df454f48a5 in handle_select (thd=thd@entry=0x62b0000bd218, lex=lex@
#15 0x000055df455d8e27 in Sql_cmd_create_table_like::execute (this=<optimized o
```

▼ Issue Links

relates to

✓ [MDEV-17805](#) Do not add temporary tables to dict_sys->table_hash


⚡ **STALLED**

links to

■ [CVE-2022-32082](#)

▼ Activity



✓  Alice Sherepa added a comment - 2021-08-27 13:58

Thanks for the report!

Repeatable on 10.5, 10.6

```
--source include/have_innodb.inc
```

```
CREATE TEMPORARY TABLE t1 (i TEXT(15) NOT NULL UNIQUE CHECK (i)) engine=innodb
REPLACE SELECT NULL AS a;
```

10.5 87ff4ba7c874ccb8a5b1105

```
2021-08-27 15:56:14 0x7f41c2918700 InnoDB: Assertion failure in file /10.
InnoDB: Failing assertion: table->get_ref_count() == 0
InnoDB: We intentionally generate a memory trap.
InnoDB: Submit a detailed bug report to https://jira.mariadb.org/
InnoDB: If you get repeated assertion failures or crashes, even
InnoDB: immediately after the mysqld startup, there may be
InnoDB: corruption in the InnoDB tablespace. Please refer to
InnoDB: https://mariadb.com/kb/en/library/innodb-recovery-modes/
InnoDB: about forcing recovery.
210827 15:56:14 [ERROR] mysqld got signal 6 ;
```

```
linux/raise.c:51(__GI_raise)[0x7f41d1cef18b]
stdlib/abort.c:81(__GI_abort)[0x7f41d1cce859]
ut/ut0dbg.cc:60(_sub_D_00099_0)[0x555e28c31207]
dict/dict0dict.cc:1916(dict_sys_t::remove(dict_table_t*, bool, bool))[0x55
row/row0mysql.cc:3370(row_drop_table_for_mysql(char const*, trx_t*, enum_s
handler/ha_innodb.cc:13302(ha_innobase::delete_table(char const*, enum_sql
handler/ha_innodb.cc:13430(ha_innobase::delete_table(char const*)) [0x555e2
```

✓  Marko Mäkelä added a comment - 2021-08-27 14:43

InnoDB is only the messenger here. I checked `./mtr --rr` of the described SQL preceded with the following:

```
--source include/have_innodb.inc
set default_storage_engine=innodb;
```

In `rr replay` on an AMD64 system, I executed

```
continue
# SIGABRT here
```

```

frame 4
watch -l table->n_ref_count
disable 1
break ha_innodb::create
run
disable 2
enable 1
continue
...

```

to find all modifications of the reference-count. After the table was created, I see the reference-count being incremented in:

```

#3 0x00005607bd1a4638 in handler::ha_open (this=0x7f7720178bd0, table_arg
    test_if_locked=test_if_locked@entry=4114, mem_root=mem_root@entry=0x0,
#4 0x00005607bcfe3145 in open_table_from_share (thd=thd@entry=0x7f7720001
    outparam=0x7f77201787c8, is_create_table=false, partitions_to_open=0x0
#5 0x00005607bd101da8 in THD::open_temporary_table (this=this@entry=0x7f7
#6 0x00005607bd1039d8 in THD::create_and_open_tmp_table (this=this@entry=
    open_internal_tables=open_internal_tables@entry=false) at /mariadb/10.
#7 0x00005607bcfa6ca0 in create_table_impl (thd=thd@entry=0x7f7720001a28,
    @0x7f7720015070: {str = 0x7f7720015738 "test", length = 4}, orig_tab
    @0x7f7720015080: {str = 0x7f7720015020 "v0", length = 2}, path=@0x7f
    create_info=0x7f7740b5fd20, alter_info=0x7f7740b5fc30, create_table_mo
#8 0x00005607bcfa6f38 in mysql_create_table_no_lock (thd=thd@entry=0x7f77
    table_name=0x7f7720015080, create_info=0x7f7740b5fd20, alter_info=0x7f
#9 0x00005607bce8ca00 in select_create::create_table_from_items (this=thi
    at /mariadb/10.6/sql/sql_insert.cc:4498
#10 0x00005607bce8cfb5 in select_create::prepare (this=0x7f7720016a38, _va
#11 0x00005607bcf389f4 in JOIN::prepare (this=this@entry=0x7f7720016b80, t
    skip_order_by=skip_order_by@entry=false, group_init=<optimized out>, h
#12 0x00005607bcf50945 in mysql_select (thd=thd@entry=0x7f7720001a28, tabl

```

and

```

#3 0x00005607bd1a4638 in handler::ha_open (this=this@entry=0x7f77200183a0, ta
    mem_root=mem_root@entry=0x7f7720007660, partitions_to_open=0x0) at /mariad
#4 0x00005607bd1a4a79 in handler::clone (this=this@entry=0x7f7720178bd0, name
#5 0x00005607bd4cc359 in ha_innodb::clone (this=0x7f7720178bd0, name=<optim
#6 0x00005607bd1adcb8 in handler::create_lookup_handler (this=this@entry=0x7f
#7 0x00005607bd1af084 in handler::prepare_for_insert (this=0x7f7720178bd0, do
#8 0x00005607bce8d344 in select_create::prepare (this=0x7f7720016a38, _values
#9 0x00005607bcf389f4 in JOIN::prepare (this=this@entry=0x7f7720016b80, table
    skip_order_by=skip_order_by@entry=false, group_init=<optimized out>, havin
#10 0x00005607bcf50945 in mysql_select (thd=thd@entry=0x7f7720001a28, tables=0

```

```

conds=0x0, og_num=0, order=0x0, group=0x0, having=0x0, proc_param=0x0, sel
#11 0x00005607bcf50be9 in handle_select (thd=thd@entry=0x7f7720001a28, lex=lex
#12 0x00005607bcfa90cf in Sql_cmd_create_table_like::execute (this=0x7f7720014
#13 0x00005607bcd826d in mysql_execute_command (thd=thd@entry=0x7f7720001a28,
#14 0x00005607bcd9299 in mysql_parse (thd=thd@entry=0x7f7720001a28, rawbuf=<o
#15 0x00005607bcd8b7fc in dispatch_command (command=command@entry=COM_QUERY, t
packet@entry=0x7f7720087d49 "CREATE TEMPORARY TABLE v0 ( v1 TEXT ( 15 ) CH
blocking=blocking@entry=true) at /mariadb/10.6/sql/sql_parse.cc:1896

```

After this, the SQL layer will close only one of the 2 handles that it opened:

```

#4 0x00005607bd1a4464 in handler::ha_close (this=0x7f7720178bd0) at /mariadb/
#5 0x00005607bcfd9958 in closefrm (table=table@entry=0x7f77201787c8) at /mari
#6 0x00005607bd101fe7 in THD::close_temporary_table (this=this@entry=0x7f7720
#7 0x00005607bd102b12 in THD::free_temporary_table (this=this@entry=0x7f77200
#8 0x00005607bd10385e in THD::drop_temporary_table (this=0x7f7720001a28, tabl
#9 0x00005607bce47e56 in drop_open_table (thd=0x7f7720001a28, table=0x7f77201
#10 0x00005607bce869be in select_create::abort_result_set (this=0x7f7720016a38
#11 0x00005607bcf50c40 in handle_select (thd=thd@entry=0x7f7720001a28, lex=lex
#12 0x00005607bcfa90cf in Sql_cmd_create_table_like::execute (this=0x7f7720014
#13 0x00005607bcd826d in mysql_execute_command (thd=thd@entry=0x7f7720001a28,
#14 0x00005607bcd9299 in mysql_parse (thd=thd@entry=0x7f7720001a28, rawbuf=<o
#15 0x00005607bcd8b7fc in dispatch_command (command=command@entry=COM_QUERY, t
packet@entry=0x7f7720087d49 "CREATE TEMPORARY TABLE v0 ( v1 TEXT ( 15 ) CH
blocking=blocking@entry=true) at /mariadb/10.6/sql/sql_parse.cc:1896

```

The assertion will fail, because the table is being dropped even though a handle remains open:


```

#4 0x00005607bd4f2776 in ha_innobase::delete_table (this=<optimized out>, nam
#5 0x00005607bd1a42e2 in hton_drop_table (hton=<optimized out>, path=<optimiz
#6 0x00005607bd101ad7 in THD::rm_temporary_table (this=<optimized out>, base=
#7 0x00005607bd102175 in THD::free_tmp_table_share (this=this@entry=0x7f77200
#8 0x00005607bd1038df in THD::drop_temporary_table (this=0x7f7720001a28, tabl
#9 0x00005607bce47e56 in drop_open_table (thd=0x7f7720001a28, table=0x7f77201
#10 0x00005607bce869be in select_create::abort_result_set (this=0x7f7720016a38
#11 0x00005607bcf50c40 in handle_select (thd=thd@entry=0x7f7720001a28, lex=lex
#12 0x00005607bcfa90cf in Sql_cmd_create_table_like::execute (this=0x7f7720014
#13 0x00005607bcd826d in mysql_execute_command (thd=thd@entry=0x7f7720001a28,
#14 0x00005607bcd9299 in mysql_parse (thd=thd@entry=0x7f7720001a28, rawbuf=<o
#15 0x00005607bcd8b7fc in dispatch_command (command=command@entry=COM_QUERY, t
packet@entry=0x7f7720087d49 "CREATE TEMPORARY TABLE v0 ( v1 TEXT ( 15 ) CH
blocking=blocking@entry=true) at /mariadb/10.6/sql/sql_parse.cc:1896

```

Related note: It would be great if the SQL layer did not force storage engines to memorize temporary tables by a generated name string ([MDEV-17805](#)).

If I remember correctly, it was in MariaDB 10.3 where we started to allow multiple handles to a temporary table in a query. It may be that older InnoDB versions are missing an equivalent assertion.

-
- ✓  [Oleksandr Byelkin](#) added a comment - 2022-07-06 12:27

Original test suite has an error:

```
CREATE TEMPORARY TABLE v0 ( v1 TEXT ( 15 ) CHAR SET BINARY NOT NULL NOT NULL U
main.test                                     [ fail ]
```


```
Test ended at 2022-07-06 14:26:26
```

```
CURRENT_TEST: main.test
```


```
mysqltest: At line 1: query 'CREATE TEMPORARY TABLE v0 ( v1 TEXT ( 15 ) CHAR S
```

-
- ✓  [Mingli-Yu](#) added a comment - 2022-07-07 08:06

Does the version 10.8.3 have the issue? Thanks!

-
- ✓  [Oleksandr Byelkin](#) added a comment - 2022-07-08 08:27

[Mingli-Yu](#) very probably.


-
- ✓  [Oleksandr Byelkin](#) added a comment - 2022-07-08 09:40

```
commit 5a48ba6ac493d760934172c4733840729355a01b (HEAD -> bb-10.5-MDEV-26433, o
Author: Oleksandr Byelkin <sanja@mariadb.com>
```

```
Date:   Fri Jul 8 11:38:45 2022 +0200
```

```
MDEV-26433 assertion: table->get_ref_count() == 0 in dict0dict.cc line 191
```

```
Close handlers in THD::drop_temporary_table.
```


-
- ✓  [Sergei Golubchik](#) added a comment - 2022-07-12 08:16

5a48ba6ac493d is ok to push

Assignee:

 Oleksandr Byelkin

Reporter:

 Jingzhou Fu

Votes:

0 Vote for this issue

Watchers:

7 Start watching this issue

▼ Dates

Created:

2021-08-19 04:32

Updated:

2022-07-12 09:37

Resolved:

2022-07-12 09:36

▼ Git Integration



Error rendering 'com.xiplink.jira.git.jira_git_plugin:git-issue-webpanel'. Please contact your Jira administrators.