Step1. In file `sql/sql_udf.cc` and function `udf_read_functions_table`. A new_thd is allocated and its m_attachable_trx is nullptr.

```
228   THD *new_thd = new (std::nothrow) THD;
229   if (new_thd == nullptr) {
230     LogErr(ERROR_LEVEL, ER_UDF_CANT_ALLOC_FOR_STRUCTURES);
231     free_root(&mem, MYF(0));
232     delete new_thd;
233     return;
234   }
235   new_thd->thread_stack = (char *)&new_thd;
236   new_thd->store_globals();
237   {
238     LEX_CSTRING db_lex_cstr = {STRING_WITH_LEN(db)};
239     new_thd->set_db(db_lex_cstr);
240   }
241
242   TABLE_LIST tables(db, "func", TL_READ, MDL_SHARED_READ_ONLY);
243
244   if (open_trans_system_tables_for_read(new_thd, &tables)) {
245     DBUG_PRINT("error", ("Can't open udf table"));
246     LogErr(ERROR_LEVEL, ER_UDF_CANT_OPEN_FUNCTION_TABLE);
247     goto end;
248   }
"sql/sql_udf.cc" 949 lines --22%--                                    218,1         23%
```

Step1. new_thd->m_attachable_trx is nullptr

Step2. `udf_read_functions_table` invokes `open_trans_system_tables_for_read`. In file `sql/sql_base.cc` and function `open_trans_system_tables_for_read`:

thd->begin_attachable_ro_transaction() allocates thd->m_attachable_trx.

thd->m_attachable_trx is not nullptr while **thd->m_attachable_trx->m_prev_attachable_trx is nullptr.**

Step3. `open_tables()` failed for whatever reason and thd->end_attachable_transaction() is invoked.

```
10255 bool open_trans_system_tables_for_read(THD *thd, TABLE_LIST *table_list) {
10256   uint counter;
10257   uint flags = MYSQL_OPEN_IGNORE_FLUSH | MYSQL_LOCK_IGNORE_TIMEOUT;
10258
10259   DBUG_TRACE;
10260
10261   assert(!thd->is_attachable_ro_transaction_active());
10262
10263   // Begin attachable transaction.
10264
10265   thd->begin_attachable_ro_transaction();
10266
10267   // Open tables.
10268
10269   if (open_tables(thd, &table_list, &counter, flags)) {
10270     thd->end_attachable_transaction();
10271     return true;
10272   }
"sql/sql_base.cc" 10413 lines --98%--                                 10242,1       98%
```

Step2. Alloc thd->m_attachable_trx

Step3. Failed to open_tables() for whatever reason.

Step4. In file `sql/sql_class.cc` and function `end_attachable_transaction()`:

As shown in Step2, m_attachable_trx->m_prev_attachable_trx is nullptr.

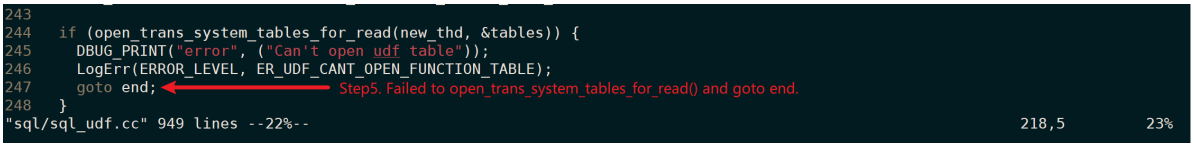
m_attachable_trx is reset to nullptr.

```
1848 void THD::begin_attachable_ro_transaction() {
1849   m_attachable_trx = new Attachable_trx(this, m_attachable_trx);
1850 }
1851
1852 void THD::end_attachable_transaction() {
1853   Attachable_trx *prev_trx = m_attachable_trx->get_prev_attachable_trx();
1854   delete m_attachable_trx;
1855   // Restore attachable transaction which was active before we started
1856   // the one which just has ended. NULL in most cases.
1857   m_attachable_trx = prev_trx;
1858 }
"sql/sql_class.cc" 2911 lines --62%--                                 1828,1        63%
```

Step4. prev_trx is nullptr and thd->m_attachable_trx is reset to nullptr.

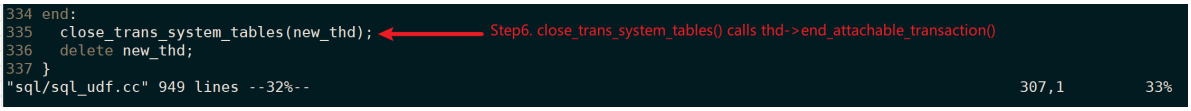
Step5. In file `sql/sql_udf.cc` and function `udf_read_functions_table`:

Failed to open_trans_system_tables_for_read() and goto end.



Step6. In fie `sql/sql_udf.cc` and function `udf_read_functions_table`:

close_trans_system_tables() invokes thd->end_attachable_transaction() and new_thd ends attachable transaction for the second time.



Step7. In file `sql/sql_class.cc` and function `end_attachable_transaction`: