

Database - Bug #11928

Rolling back a transacton yields NPE if inside the transaction a local temporary table was filled and dropped

10/01/2026 07:01 AM - Alexandru Lungu

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:	Alexandru Lungu	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			

History

#1 - 10/01/2026 07:02 AM - Alexandru Lungu

- Assignee set to Alexandru Lungu

Problem

With the fwd-h2 page store (MV_STORE=FALSE), rolling back a transaction fails with a NullPointerException when, inside that open transaction, a local temporary table was first given rows and then dropped with DROP TABLE ... TRANSACTIONAL.

The TRANSACTIONAL keyword stops the DROP from committing the open transaction, as intended. But the undo log still holds the row-level undo records of the dropped table. On ROLLBACK (or a rollback to a savepoint taken before those rows) H2 replays them against the removed PageStoreTable, whose database reference is already null:

```
org.h2.jdbc.JdbcSQLException: General error:
"java.lang.NullPointerException: Cannot invoke "org.h2.engine.Database.getNextModificationDataId()" because
e "this.database" is null"; SQL statement:
ROLLBACK [50000-200]
Caused by: java.lang.NullPointerException: Cannot invoke
"org.h2.engine.Database.getNextModificationDataId()"
because "this.database" is null
    at org.h2.pagestore.db.PageStoreTable.removeRow(PageStoreTable.java:315)
    at org.h2.engine.UndoLogRecord.undo(UndoLogRecord.java:100)
    at org.h2.engine.Session.rollbackTo(Session.java:879)
    at org.h2.engine.Session.rollback(Session.java:855)
    at org.h2.command.dml.TransactionCommand.update(TransactionCommand.java:49
```

The session then can't roll back at all. The JDBC connection fails the same way even on close().

Minimal reproducer

Plain JDBC against fwd-h2-1.65-trunk.jar:

```
create table app (id int primary key);
-- autocommit off
insert into app values (1);
create local temporary table t (id int primary key) transactional;
insert into t values (1);
drop table if exists t cascade transactional;
rollback; -- NPE in PageStoreTable.removeRow
```

Results:

- **Without the insert into t:** the rollback succeeds.
- **Plain DROP TABLE (no TRANSACTIONAL):** no NPE, but the drop implicitly commits the open transaction, so the app row survives the rollback. That is wrong too.
- **Same steps on PostgreSQL, SQL Server and MariaDB:** each works (transactional DDL, or a temp-table drop that neither commits nor breaks the rollback).

Impact on FWD (#8388)

The intra-session dirty share (8388a) keeps uncommitted changes in `__temp__<table>_0` local temporary tables. These are created, indexed and dropped inside the application transaction:

- `Dialect.getTransientDdlPostfix()` returns " transactional" on H2.
- `TransientDatabaseManager.endTxScope` drops the tables of every rolled-back scope, the same rule on all dialects.

When a sub-transaction is undone together with its parent (no outcome reported, e.g. UNDO of an outer block), the drop runs while the rows are still in the open transaction. The parent's rollback then hits the NPE and FWD reports unable to rollback current transaction (`StopConditionException`).