

Triggers

Mengxi Chen, Min Huang, Tao Lin



Goals and Progress

- Support execute create and drop trigger statements
 - Update data structure for create and drop statement to support trigger
 - Convert Postgres parsing tree to Peloton statement for trigger
 - Update data structure create and drop for plan to support trigger
 - Convert state to plan for trigger
 - Data structures for Trigger and TriggerList and related operations
 - Store trigger information into the catalog and support updates
 - Update DataTable with TriggerList information
-
- Work Finished
 - Work in Progress

Goals and Progress

- Evaluate the predicates in the trigger (Put/get predicates in catalog)
- Prevent infinite recursion of triggers
- Invoke real UDFs and send contexts
- Trigger operations for 12 kinds of triggers
 - One complete: Before insert for each row
 - The other 11 kinds: (before, after) × (insert, delete, update) × (row, statement)

- Work Finished
- Work in Progress

Correctness Test

- PostgreSQL parser tests
 - Create trigger test
 - Drop trigger test
- Create tests
 - Creating trigger
 - Creating trigger in catalog
- Drop tests
 - Dropping trigger
- Trigger tests

Quality Assessment

- Infinite recursion of triggers
 - When insert a tuple, insert a tuple...
 - “It is the trigger programmer's responsibility to avoid infinite recursion in such scenarios.” ([PostgreSQL: Documentation: 9.6: Overview of Trigger Behavior](#))
 - PostgreSQL: `pg_trigger_depth()`
 - We: Forbid cascading triggers

Quality Assessment

- Put/get predicates in catalog
 - Serialize and deserialize AbstractExpression?
 - Keep raw SQL text in catalog and parse every time?
 - Only support the simplest?

```
CREATE TRIGGER check_update
  BEFORE UPDATE ON accounts
  FOR EACH ROW
  WHEN (OLD.balance IS DISTINCT FROM NEW.balance)
  EXECUTE PROCEDURE check_account_update();
```

```
{DISTINCTEXPR :opno 96 :opfuncid 65 :opresulttype 16 :opreset
false :opcollid 0 :inputcollid 0 :args ({VAR :varno 1
:varattno 1 :vartype 23 :vartypmod -1 :varcollid 0
:varlevelsup 0 :varnoold 1 :varoattno 1 :location 76} {VAR
:varno 2 :varattno 1 :vartype 23 :vartypmod -1 :varcollid 0
:varlevelsup 0 :varnoold 2 :varoattno 1 :location 105})
:location 88}
```



<https://surrealcomics.wordpress.com/2011/07/12/comic-book-drawings-the-laughing-city/>

Benchmark Results

- TPC-C (run on my Laptop; 1 warehouse; 1 terminal)
- Peloton without trigger (master branch):
6040 requests/s (core dumped when inserted into OORDER)
- Peloton with trigger:
5656 requests/s (core dumped when inserted into OORDER)

Future Work

- Finish work in progress
 - Put/get predicates in catalog
 - Invoke real UDF and send contexts
 - Support different kinds of triggers
- Enable/disable triggers
- Support other kinds of triggers if Peloton support view
 - INSTEAD OF
 - DEFERRABLE

Additional Slides

Benchmark Results

- Benchmark is not supported because of grammar difference
- Benchmark:

```
CREATE OR REPLACE TRIGGER user_seq_tr
    BEFORE INSERT ON useracct FOR EACH ROW
    WHEN (NEW.user_id IS NULL OR NEW.user_id = 0)
    BEGIN SELECT user_seq.NEXTVAL INTO :NEW.user_id FROM dual;END;;
```

- PostgreSQL:

```
CREATE TRIGGER check_update
    BEFORE UPDATE ON accounts
    FOR EACH ROW
    EXECUTE PROCEDURE check_account_update();
```