

PostgreSQL Triggers Lab I

Tom Kelliher, CS 417

The purpose of this lab is for you to gain some understanding of how triggers and stored procedures are used in PostgreSQL to implement integrity constraints. You will create a PL/pgSQL function which will implement two semantic constraints for a payroll database. The constraints are: No salary should be negative and the salary cap (maximum sum of the salaries) is \$1,000.

1. Download `triggerLab1.sql` from the course web site to one of your directories on phoenix.
2. Open this file in an editor. Notice that it consists of three parts: a clean-up section; a section which creates the payroll table, PL/pgSQL function, and trigger; and a section which tests the trigger function.
3. You will be writing the code for `VerifySalary()`. Carefully read the comments above this function so that you're clear on what needs to be done.
4. Write the code for `VerifySalary()`.
5. When you're ready to run the code, run `psql` to open your personal database and execute the SQL code in your file. The output from the run should be very similar to:

```
kelliher=> \i triggerLab1.sql
psql:triggerLab1.sql:5: ERROR:  relation "employee" does not exist
psql:triggerLab1.sql:6: ERROR:  function verify() does not exist
psql:triggerLab1.sql:7: ERROR:  table "employee" does not exist
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
psql:triggerLab1.sql:60: ERROR:  Tom assigned a negative salary.
  name | salary
-----+-----
(0 rows)

INSERT 0 1
INSERT 0 1
```

(Continued on next page.)

```
psql:triggerLab1.sql:70: NOTICE: Tom reached the salary cap.  
UPDATE 1  
 name | salary  
-----+-----  
 Jill |    895  
 Tom  |    105  
(2 rows)
```

The **ERROR** messages will be present only on the first run. Subsequently, they will be replaced with **DROP** messages.