

Homework 7. Triggers

For this exercise please use an updated version of movies database. Both table generation and data insertion statements are provided. Create your database and pre-populate it with data.

The triggers are expected to be written similarly to the lecture examples, but you can experiment by adding your own extensions.

1. Create a trigger that fires each time a net worth of a movie executive becomes greater than 10,000,000 and record the names of all such people into table `rich_execs`. Test your trigger by inserting one tuple with net worth < 10,000,000, and one with a greater one. Check which of the two was recorded in `rich_execs` (AFTER UPDATE trigger).
2. Ensure that only movie execs with net worth at least 10,000,000 can be inserted as studio presidents (BEFORE UPDATE OR INSERT trigger ON table `studio`). Test your trigger by trying to insert a violating exec while updating `studio` table.
3. Create a statement-level audit trigger on table `movieexec`, which records all the changes of networth field. An example of the audit trigger is given in the lecture link. Update networth of all execs by adding 100,000 to all not-null values. Check that your trigger generates a single entry in the audit table.
4. Create view `MGM` which gives all movie titles and years of movies produced by MGM. Implement INSTEAD OF insert trigger on `MGM` updateable view.

Sample syntax for INSTEAD OF triggers for Postgre 9.1 can be seen here:

<https://vibhorkumar.wordpress.com/2011/10/28/instead-of-trigger/>

Submit your code for all tables that you created, functions and triggers in a single file `h7.sql`. The code should run on `cdf` against PostgreSQL 9.1.