Triggers

This tutorial uses the relational schema in `T07.sql`.

1. Suppose that no employee can be both an engineer and a manager. Create two triggers to enforce this constraint on the `Engineers` and `Managers` tables. The triggers should run before insert or update and *prevent* changes (*i.e.*, no insertion and no update) when the condition is not met (*i.e.*, when an employee is about to be both engineer and manager).

2. Suppose that we pay every engineers working on a project \$100 per hour worked. Since every project has a budget, the total number of hours worked by every engineer multiplied by 100 cannot exceed the project budget. Create a trigger to enforce this constraint such that when an insert or update is performed on `Works` table that violates this constraint, the number of hours worked by the engineer is set to the maximum allowable for that project.

3. As each work how has a type, we have an additional constraint that for a given work, the amount of time spent on the work cannot exceed the maximum hours for that particular type of work. Create a trigger to restrict `Works` table such that the hours worked cannot exceed the maximum hours for the given type. Whenever we want to insert or update such that the hours worked exceed the maximum hours, we set the hours worked to the maximum hours.

4. Consider a case where we have a default work type. For simplicity, we let `wid = 0` to be the default work type. As this is the default, we can neither modify nor delete this work type. Create a trigger to prevent modification or deletion of the default work type. The trigger should raise notice that some users are trying to modify or delete this default work type. Furthermore, the trigger should not raise any notice when some users are trying to modify or delete other type of work.