CS 328 - Week 6 Lab Exercise - 2025-02-27/28

Deadline

for the Thursday lab: due by 11:59 pm on 2025-02-27

for the Friday lab: due by the end of lab

Purpose

To practice writing and testing a PL/SQL trigger.

How to submit

Submit your files for this lab using `~st10/328submit` on nrs-projects, each time entering a lab number of **86**.

Requirements

- For this particular lab, you may work either in pairs or individually.
- IF you work in pairs, you are expected to pair program:
  - This means **two** people working at **ONE** computer, one typing ("driving"), one saying what to type ("navigating"),

    while **BOTH** are looking at the **shared** computer screen and **discussing** issues along the way.
- If you pair program, make sure **BOTH** of your names appear in each file submitted.
- When you are done, before you leave lab, somehow e-mail or copy the lab exercise files so that **BOTH** of you have copies, and **BOTH** of you should submit these files using `~st10/328submit` on nrs-projects, with a lab number of **86**.
- You are expected to follow the style standards from the posted "CS 328 SQL and PL/SQL Coding Standards so far" (at https://nrs-projects.humboldt.edu/~st10/s24cs328/328-sql-plsql-coding-standards.pdf).

Lab set-up

- On nrs-projects, if the driver has not previously executed **`set-up-ex-tbls.sql`** in their Oracle account, they should do so, so that they have the tables `empl`, `dept`, and `customer` in their database.
  - If needed, they can get a copy of this script using:

    ```
    cp ~st10/set-up-ex-tbls.sql .    # don't forget the blank and period!
    ```
- Copy over the following "starter" file into your working directory, that you will then add to for today's lab exercise:

  ```
  cp ~st10/lab6-start.sql lab6.sql
  ```
- In this SQL script **`lab6.sql`**:
  - In the opening comment, add your name(s) where requested, and adjust the last-modified date appropriately.

– Also add your names in the first **`prompt`** command where requested.

New table **`dept_changes`**

An organization has decided it would like to keep track, over time, of its trends in changing department names.

In your script `lab6.sql`, you will see that there are commands to drop and create a table named `dept_changes` that has four attributes:

- `dept_num`, of type `char(3)`
- `change_date`, of type `date`
- `prev_dept_name`, of type `varchar2(15)`
- `next_dept_name`, of type `varchar2(15)`

You will also see that:

- the pair of attributes `dept_num` and `change_date` are its primary key
- `dept_num` as a foreign key referencing `dept`

(The first time you run this, the `drop table` command should give an error complaining that there is not a table with this name to drop. You will not get this error on subsequent runs of your script.)

Problem 1 - trigger **`log_name_changes`**

In your script `lab6.sql`, after the comment:

```
/*===
   Problem 1 - create trigger log_name_changes
===*/
```

write a PL/SQL trigger `log_name_changes` that meets the following requirements:

- Look in the posted SQL script **`328lect06-1.sql`** at the version of the trigger **`empl_trig`** that we created during class on Monday.
  – Create an opening comment block for your trigger that has a **`trigger:`** part and **`purpose:`** part in the same style that you see here.
  – Follow that with the PL/SQL code creating your trigger.
- It should fire after each update to the `dept` table, for each row updated.
- *If* the firing update changed a department's name, it should insert a new row into the table `dept_changes` containing:
  – the updated department's department number,
  – the current date (use `sysdate` for this),
  – the previous name of the updated department, and
  – the department name after the update.

  – (If the update changed something else -- for example, the department location -- then this trigger will *not* make any changes to the table `dept_changes`. That is, it will simply do **nothing** in that case.)

- Remember to follow your PL/SQL trigger with:

```
/
show errors
```

Problem 2 - testing your trigger

You should see a comment `/*=== TESTING trigger log_name_changes ===*/`
that is then followed by statements exercising and testing your trigger.

Note that it first does a `commit;` of the current state of the database, and that it ends with a `rollback;` after the tests -- you should not have to re-run `set-up-ex-tbls.sql` as you are debugging your trigger.

After you have written your trigger, run your script and carefully look at the results of this testing part. Do its results suggest that your trigger is working correctly?

If you see any problems, debug your trigger until you believe these tests are passing.

When successful, your resulting **`lab6-out.txt`** should show that your trigger successfully compiled, and that its tests passed.

BEFORE you leave lab:

Make sure that you **both** have copies of the files:

- **`lab6.sql`** and **`lab6-out.txt`**

...and you BOTH submit these two files using `~st10/328submit` on
nrs-projects, with a lab number of **86**.

How the navigator can get files `lab6.sql` and `lab6-out.txt`:

(for a driver with username *dr12*, and a navigator with username *na89* - replace these with your *actual* usernames when you actually do this)

These may be in a directory that is harder for the navigator to make a copy from than `public_html`.

For example -- they might be in a directory `328lab6` that is **not** a sub-directory of `public_html`, but is instead a subdirectory of the driver's home directory *~dr12*.

Here is an approach for this:

- The **driver** *dr12* should *temporarily* make the directory with these files world-readable and -executable, and these files world-readable:

```
chmod 755 .          # notice the space and the period!
chmod 644 lab6.sql lab6-out.txt
```

- Now the **navigator** can copy these into a directory of their choice -- assuming the navigator is within the directory they want to copy into:

```
cp ~dr12/328lab6/* .  # notice the space and the period!
```

- The **driver** and **navigator** should **BOTH** then **protect** these files:

```
chmod 600 lab6.sql lab6-out.txt
```

...and both can protect the directory containing them:

```
chmod 700 .          # notice the space and the period!
```

...and **both** can now submit these using `~st10/328submit` from the directory containing these files.