**CSCI N311**

**Project 7: PL/SQL Trigger Programming**

***Objectives***

After completing this lab you should be able to:

* Understand the basic syntax of the PL/SQL programming language.
* Understand what a trigger does.
* Write a trigger to generate an audit trail of changes.

Turn in the PL/SQL source code for your trigger. In addition to this, update and insert some EMPLOYEE rows, then select the data generated in the EMPLOYEE\_LOG table to show that your EMPLOYEE\_TRIGGER works. Capture this work using screen shots.

***Assignment Steps***

1. Create a table for employees (EMPLOYEE). Create the following columns for this table. Define employee\_id as a primary key.

Employee\_id      varchar2(8) not null  
last\_name        varchar2(40)  
first\_name       varchar2(40)  
middle\_initial   varchar2(1)  
dept\_id          varchar2(4)  
title            varchar2(40)  
supervisor\_id    varchar2(8)

1. Create a department table (DEPARTMENT) for our company. This table will have the following columns. Define dept\_id as a primary key.

Dept\_id          varchar2(4) not null  
Division\_name    varchar2(20)  
Department\_name  varchar2(20)

1. Now add referential integrity or foreign key constraints to EMPLOYEE. Since supervisors must be employees, the supervisor ID column will reference the employee ID column. And since we have a table for departments, employee.dept\_id will reference the department.dept\_id. Name these foreign keys FK\_emp\_emp and FK\_emp\_dept.
2. Remove all the data from your EMPLOYEE and DEPARTMENT tables. (The first time you run the script of course no data will be deleted, but on subsequent runs you will need an empty table.)
3. Insert the following data into the department table.

|  |  |  |
| --- | --- | --- |
| **Dept\_id** | **Division\_name** | **Department\_Name** |
| EBX | Pharma | [R&D](file:///C:\Users\Owner\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\FrontPageTempDir\pvw49.htm#NOTE1) |
| CP1 | Corporate | Finance |
| CP2 | Corporate | Marketing |

|  |
| --- |
| [NOTE:](file:///C:\Users\Owner\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\FrontPageTempDir\pvw49.htm#BACK1) Inserting the value "R&D" will cause some problems. The & sign indicates that the following string is a SQL\*Plus variable. You need to run “SET DEFINE OFF” before inserting this row, and run “SET DEFINE ON” after insertion. |

1. Now insert the following data into the employee table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Emp ID** | **Employee Name** | [Dept.](file:///C:\Users\Owner\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\FrontPageTempDir\pvw49.htm#NOTE) | **Title** | **Supervisor** |
| EB001 | Richard Abbott | EBX | Director |  |
| EB002 | Fred Johnson | EBX | Manager | EB001 |
| EB103 | Jan Searle | EBX | Technician | EB002 |
| EB104 | Fred W. Lambert |  | Technician | EB002 |
| CP001 | Judith H. Roche | CP1 | Director |  |
| CP002 | Smithy K. Glaxo | CP2 | Director |  |
| CP003 | Amerigus H. Product | CP1 | Manager | CP001 |
| CP004 | Helmut Bayer | CP1 | Dept. Head | CP003 |
| CP005 | Trent Merck | CP1 | Team Leader | CP004 |
| CP006 | Ted Baxter | CP1 | clerk | CP005 |
| CP007 | John Underling | CP1 | clerk | CP005 |
| CP008 | Susan H. Helper |  | AA | CP003 |
| CP009 | Joe Dilbert | CP1 | clerk | CP005 |

[NOTE:](file:///C:\\Users\\Owner\\AppData\\Local\\Microsoft\\Windows\\Temporary%20Internet%20Files\\FrontPageTempDir\\pvw49.htm" \l "TOP) some dept. values intentionally left null.

1. Insert all the workers from the WORKER table using an INSERT INTO table (columns,...) SELECT statement. For employee\_id, be creative and derive an employee\_id from data found within the worker table (eg. String concatenates, numeric functions, etc.). Assign all these workers into department CP2, with title ‘Salesperson’ and supervisor = CP002.

|  |
| --- |
| Here is one way to implement #7. There would be several ways of doing essentially the same thing. Note however that some "names" in the worker table are not just first and last name. So this would not be a complete solution but good enough for lab5.  INSERT INTO EMPLOYEE (employee\_id, first\_name,  last\_name, dept\_id, title, supervisor\_id)  SELECT substr(name,1,3)||  ltrim(to\_char(nvl(age,1),'009')) as employee\_id,  initcap(substr(name,1,instr(name,' ',1)-1)) as first\_name,  initcap(substr(name,instr(name,' ',1)+1)) as last\_name,  'CP2', 'Salesperson', 'CP002'  FROM WORKER; |

8.    Transactional Control

a)                Commit your changes to the database. Also select COUNT(\*) from employee and department.

b)                Now delete all rows from EMPLOYEE. Select COUNT(\*) from employee. There should be no rows.

c)                 Rollback the deletes from EMPLOYEE. Select COUNT(\*) from employee. The rows should all be back.

9.    Smithy wants to change his name. Update his last name from Glaxo to Beecham. Select just this row to show that it was changed.

1. You are to create a log table that will record all changes that occur to your EMPLOYEE table.

create an EMPLOYEE\_LOG table with columns as shown below:

Name Null? Type

------------------------------- -------- ----

EMPLOYEE\_ID NOT NULL VARCHAR2(8)

PREV\_LAST\_NAME VARCHAR2(40)

PREV\_FIRST\_NAME VARCHAR2(40)

PREV\_MIDDLE\_INITIAL VARCHAR2(1)

PREV\_DEPT\_ID VARCHAR2(4)

PREV\_TITLE VARCHAR2(40)

PREV\_SUPERVISOR VARCHAR2(8)

MOD\_USER VARCHAR2(8)

MOD\_TIMESTAMP DATE

1. Next, create a PL/SQL trigger called EMPLOYEE\_TRIGGER on the EMPLOYEE table. This trigger will be executed after every insert or update.
   1. For inserts, this trigger should put the new EMPLOYEE.EMPLOYEE\_ID value, the user in MOD\_USER, and the system date in MOD\_TIMESTAMP to record the creation of the data into EMPLOYEE\_LOG.
   2. For updates, this trigger should put the previous EMPLOYEE values into the corresponding EMPLOYEE\_LOG column as well as recording the MOD\_USER and MOD\_TIMESTAMP.

***Submission:***

Save your scripts into a .sql file and paste the screen shot into a Word document.

Submit the script file and the Word document through Canvas.