# Homework #03

## Create a PL/SQL Exercise: Triggers and Constraints

| | |
|---|---|
| 40 pts | A single text file containing the answers to **Part A and Part C** must be submitted via **Circuit**.<br>Self and peer assessment must be submitted through **https://forms.gle/ugtnBwzRPLtkiyADA on Google Form. Peer assessment assignments will be managed** on **Circuit**. |
| Notes: | Use grammatically correct sentences to convey your thoughts when answering questions. One-word answers, "naked" bulleted lists, etc. will receive a score of zero. |

**Overall Instruction**

In this homework, your goal is to brainstorm a real-world problem-solving scenario where PL/SQL **Triggers and Constraints** are needed to provide a solution and formulate a class exercise accordingly.

**This homework consists of four parts:**

- In Part A, you will first answer a few questions to demonstrate your understanding of PL/SQL triggers and constraints.
- In Part B, you will be guided through an example process to create a PL/SQL exercise.
- In Part C, you will need to come up with your own PL/SQL exercise.
- In Part D, which is done through the self and peer assessment stage on the Circuit tool, you will evaluate the exercises created by your classmates and yourself.

**Submission**

- Submit your answers to Part A and Part C in **a single text file** before the Content Submission deadline on Circuit. You can use the template provided to formulate your submission.
- Use **https://forms.gle/ugtnBwzRPLtkiyADA** to conduct self and peer assessment on the Google Form. The Circuit will automatically assign submissions for you to review in the Reviews Stage.

**Grading**

- Your grade for this homework is going to be effort-based. Your instructor and TA will read your submission, self and peer assessment, to determine whether you have created meaningful PL/SQL exercises for each other and whether you have carefully read and evaluated the PL/SQL exercises created by others and yourself.

## Part A (4pts): PL/SQL Triggers and Constraints.

QA.1. (2pts) Create a table **Cust_Audit** table with the following fields and constraints:

- Transaction (this column can only take one of the following values: select, update, delete, insert)
- NumRows (integer)
- Timestamp (DateTime)
- UserID (references the UserID column in the **User** table)
- Description (less than 100 characters).

QA.2. (2pts) Will the trigger fire after the following two statements?

DML Statements:

```
DELETE FROM MEMBER WHERE ZIPCODE=47905;
SELECT FROM MEMBER WHERE ADDRESS IS NULL;
```

Table Information:

```
Member(memberid, lastName, firstname, address, zipcode)
MEMBER_AUDIT (id, Transaction, Timestamp, by_user,
Description)
```

Trigger definition:

```
Create or replace trigger MEM_AUDIT_CHANGES
After update or delete on MEMBER
DECLARE
     l_transaction VARCHAR2(10);
Begin
     l_transaction := CASE
          WHEN UPDATING THEN 'UPDATE'
          WHEN DELETING THEN 'DELETE'
     END;

     Insert into MEMBER_AUDIT (Transaction, Timestamp,
     by_user, Description)
     Values(l_transaction, sysdate, user, 'One row of the
     Member table was ' || l_transaction || 'd.');
END;
```

## Part B: First Experience of Creating a PL/SQL Exercise.

This part demonstrates an example process of creating a PL/SQL exercise using the knowledge of functions and procedures.

*Special thanks to Estevan Becerra, from CNIT372 in Spring 2022. This exercise is adapted from a worked-out example created by Estevan Becerra.*

### Problem Background:

Alex owns a video game store, and one of his daily tasks is to keep track of the information on different video games.

### Overall Problem:

Write a trigger that logs updates/deletes that took place on each row of **GAME_SHOP** table. Assume that there is already an **audits** table.

### A table in Use for this Problem:

Name the table:

- GAME_SHOP

The table contains the following attributes (columns):

- Game_id (number)
- UPC (number)
- Publisher_id (number)
- Release_year (number)

Here are some example rows of the table:

| Game_id | UPC | Publisher_id | Release_year |
|---|---|---|---|
| 1 | 8564 | 4 | 2007 |
| 2 | 9852 | 4 | 2007 |
| 3 | 11063 | 7 | 2006 |
| 4 | 9065 | 15 | 2011 |

Name the table:

- Audits

Table definition:

```
CREATE TABLE audits (
      audit_id         NUMBER GENERATED BY DEFAULT AS
IDENTITY PRIMARY KEY,
      table_name       VARCHAR2(30),
      transaction_name VARCHAR2(10),
      by_user          VARCHAR2(30),
      transaction_date DATE
);
```

## Create a table with dummy data:

Download the **LoadGameShop.txt** in the attachment on the **Circuit** to create the table and import the data. (**Hint**: Due to the limitation of file types on the circuit, remember to change the **.txt** to **.sql** before you use it to import the data into Oracle SQL.)

## Step-by-step Solutions:

```
/* Step 1: Declare a trigger named game_shop_audit_trg that will
fire after each row is updated or deleted in the GAME_SHOP table
*/
```

**Suggested answer:**

```
CREATE OR REPLACE TRIGGER game_shop_audit_trg
    AFTER UPDATE OR DELETE ON GAME_SHOP FOR EACH ROW
BEGIN
     [trigger action]
END;
```

```
/* Step 2: Building on your answer to Step 2, in the trigger
body of game_shop_audit_trg, declare a local variable to store
the transaction type */
```

**Suggested answer:**

```
CREATE OR REPLACE TRIGGER game_shop_audit_trg
    AFTER UPDATE OR DELETE ON GAME_SHOP FOR EACH ROW
DECLARE
   l_transaction VARCHAR2(10);
BEGIN
     [trigger action]
END;
```

```
/* Step 3: Building on your answer to Step 3, in the trigger
body, determine the transaction type and assign the result to
the local variable you created in Step 2. */
```

**Suggested answer:**

```
CREATE OR REPLACE TRIGGER game_shop_audit_trg
    AFTER UPDATE OR DELETE ON GAME_SHOP FOR EACH ROW
DECLARE
   l_transaction VARCHAR2(10);
BEGIN

   l_transaction := CASE
         WHEN UPDATING THEN 'UPDATE'
         WHEN DELETING THEN 'DELETE'
   END;
END;
```

```
/* Step 4: Insert a row into the audit table to log the current
transaction */
CREATE OR REPLACE TRIGGER game_shop_audit_trg
    AFTER UPDATE OR DELETE ON GAME_SHOP FOR EACH ROW
DECLARE
   l_transaction VARCHAR2(10);
BEGIN

     l_transaction := CASE
         WHEN UPDATING THEN 'UPDATE'
         WHEN DELETING THEN 'DELETE'
     END;
     INSERT INTO audits (table_name, transaction_name, by_user,
     transaction_date)
        VALUES('GAME_SHOP', l_transaction, USER, SYSDATE);

END;

/* Step 5: Test the trigger and use SELECT to show the data
inside the table audit*/
DELETE FROM GAME_SHOP WHERE game_id = '1';
SELECT * From audits;
```

**Suggested answer:**

```
1 row deleted

AUDIT_ID  TABLE_NAME     TRANSACTIO BY_USER          TRANSACTI
--------- -------------- ---------- --------------- ----------
1         GAME_SHOP      DELETE     ZHOU922         08-NOV-22
```

## Part C (28 pts): Now It's Your Turn to Create a PL/SQL Exercise Related to Your Interest!

**QC.1. Problem Background (4 pts):** Please provide some context for your PL/SQL exercise with less than 100 words.
*Hints: You can think of any areas you are interested in or any real-world problems that can be solved by this process. Examples include details about video game developers and the day that particular movie theaters sell the most tickets. It can even be used in other fields like finance, healthcare, etc.*

**QC.2. Overall Problem (4 pts):** Please formulate the overall PL/SQL problem you try to solve. The overall problem should be complicated enough to require PL/SQL blocks. There should be **at least 3 steps** needed to solve this problem.

**QC.3. Create a table with dummy data** for solving the problem **(10 pts)**. The table should have at least 5 attributes and at least 100 rows of data.
In your submission, include:

- **Table specification**: table name, attribute name, and data types
- **An SQL query to create and populate your table with dummy data**: You can use the Eagle database as a source for generating dummy data. If you use other online resources, please include the reference and instructions for us to run your query.

**QC.4. List the key steps to solve the overall problem (5 pts).** Each step should have a standalone answer in SQL or PL/SQL. The problem should require **at least 3 steps** to solve.

**QC.5. Please provide the correct answer for each of the steps you listed above. (5 pts)** Please provide the SQL or PL/SQL code for each answer.

## Part D (8 pts): Self and Peer assessment.

After the content submission deadline, you will conduct a self and peer review of this homework.

All reviews need to be submitted through Google Forms. The review for each submission (your self-assessment, as well as each of the three peer assessments) should be submitted separately (4 submissions in total, each worth 2 points).

**Please note: you need to submit your evaluation to Google Forms to receive the self and peer assessment credits.** Links to the Google Form survey are provided in the rubrics on Circuit.