COEN 178 Intro to Database Systems Winter2021

Lab 6 (100 pts)

Objectives: Learn

- To practice writing triggers to enforce integrity constraints between tables.
- Using PHP and Oracle SQL to build a Web application. PHP generates the HTML forms necessary to get the user input, accesses the database and generates the output.

Submit: A sql file with your modified trigger from ex. 1, and the completed triggers from 2 & 4. Also submit a text file with the answers to the questions in 2-5

Demo: The webpage in part 2

Part 1

In this part, you will practice writing a few triggers to enforce business rules among the table data.

Create the following tables for Bank Database.

```
Create table BANKCUST_6 (custno VARCHAR(5) Primary Key,custname
VARCHAR(20),street VARCHAR(30), city VARCHAR(20));

Create table ACCOUNTS_6 (AccountNo VARCHAR(5) Primary Key,accountType
VARCHAR(10), amount NUMBER(10,2), custno varchar(5),
CONSTRAINT accounts_fkey FOREIGN Key (custno)REFERENCES BANKCUST_6(custno));

Create table TOTALS_6 (custno VARCHAR(5), totalAmount Number(10,2),
CONSTRAINT totals_fkey FOREIGN Key (custno)REFERENCES BANKCUST_6(custno));
```

Exercise 1 (10 pts)

In this exercise, you will write a trigger display the data that is inserted into **Bankcust_6** table. This trigger is not really very useful, but just a warm up exercise to write a trigger and see if it fires correctly, as a preparation to write the triggers in the subsequent exercises.

a) **At SQL PROMPT type set serveroutput on;**

b) **Create the following trigger (either run from a text file or copy and paste it at SQL prompt)**

```
CREATE or REPLACE TRIGGER display_customer_trig
   AFTER  INSERT on BankCust_6
   FOR EACH ROW
BEGIN
      DBMS_OUTPUT.PUT_LINE('From Trigger '||'Customer NO:
'||:new.custno||' Customer Name: '||:new.custname);
END;
/
show errors;
```

c) Insert the following values into BANKCUST_6 table.

```
insert into BANKCUST_6 values('c1','Smith','32 Lincoln st','SJ');
insert into BANKCUST_6 values('c2','Jones','44 Benton st','SJ');
insert into BANKCUST_6 values('c3','Peters','12 palm st','SFO');
insert into BANKCUST_6 values('c20','Chen','20 san felipo','LA');
insert into BANKCUST_6 values('c33','Williams',' 11 cherry
Ave','SFO');
```

Did your trigger work?

Modify the trigger so that it displays the city as well.

Exercise 2 (20 pts)

a) We will disable the display_customer_trig using the alter trigger statement. Use the statement (use the trigger name).

```
Alter trigger trigger_name disable
```

b) We will now write a trigger which fires after inserting a row in the Accounts_6 table. The trigger should enter the custno and the amount into the TOTALS_6 table as follows:

- If the custno is already in Totals_6 table, adds the new amount to the existing one.

- If the custno is not in Totals_6 table, adds a new row for this new customer.

The Totals_6 table should give us the total amount in all the accounts owned by each customer.

c) Complete the code for the trigger given below, following the comments

```
Create Or Replace Trigger Acct_Cust_Trig
AFTER INSERT ON Accounts_6
FOR EACH ROW
BEGIN
/*If the custno is already in the Totals_6 table, the update will
succeed */
      update totals_6
      set totalAmount = totalAmount + :new.amount
      where custno = :new.custno;

/*If the custno is not in the Totals_6 table, we insert a row into
Totals_6 table. Complete the missing part in te subquery */
insert into totals_6 (select :new.custno, :new.amount from dual
      where not exists (select * from TOTALS_6 where custno= ));

END;
/
```

Make sure that your trigger compiles without any errors.

d) Delete if there is any data in the Accounts_6 and Totals_6 tables.

e) Insert the following data into Accounts_6 table.

```
insert into ACCOUNTS_6 values('a1523','checking',2000.00,'c1');
insert into ACCOUNTS_6 values('a2134','saving',5000.00,'c1');
insert into ACCOUNTS_6 values('a4378','checking',1000.00,'c2');
insert into ACCOUNTS_6 values('a5363','saving',8000.00,'c2');
insert into ACCOUNTS_6 values('a7236','checking',500.00,'c33');
insert into ACCOUNTS_6 values('a8577','checking',150.00,'c20');
```

Did your trigger work? How did you check?

Show the data in Totals_6 table.

What is the amount for the customer, ‘c1’?

Does the total amount for ‘c1’ agree with the amounts for that customer in the Accounts_6 table?

Exercise 3 (15 pts)

If your trigger is working correctly and updating the total amount for a customer every time a new Account for that customer is created, let us try the following query from SQL prompt.

```
update Accounts_6
set amount = 1000
where accountno = 'a1523';
```

If the above query successfully ran, check the Totals_6 table.

What is the amount for the customer, ‘c1’?

Does the amount in Totals_6 table for ‘c1’ agree with the total of amounts in all the accounts for ‘c1’ in Accounts_6 table?

Exercise 4 (20 pts)

We will modify our trigger Acct_Cust_Trig to fire after inserting as well as updating data in the Accounts_6 table.

```
Create Or Replace Trigger Acct_Cust_Trig
AFTER INSERT OR UPDATE ON Accounts_6
FOR EACH ROW
BEGIN
      If inserting then
            update totals_6
            set totalAmount = totalAmount + :new.amount
            where custno = :new.custno;

            insert into totals_6 (select :new.custno, :new.amount from
dual
      where not exists ( write your complete query from Question 2);
      END IF;
      if updating then
      /* If we are updating we want to correctly set the totalAmount
            to the new amount that may be >= or < old amount
            Complete the query */

            update totals_6
            set totalAmount = totalAmount +
            where custno = :new.custno;
      end if;
END;
/
Show Errors;
```

a) Complete and Compile your trigger.

b) Now delete all rows from Accounts_6 table and Totals_6 table.

c) Insert the following data into Accounts_6 table.

```
insert into ACCOUNTS_6 values('a1523','checking',2000.00,'c1');
insert into ACCOUNTS_6 values('a2134','saving',5000.00,'c1');
insert into ACCOUNTS_6 values('a4378','checking',1000.00,'c2');
insert into ACCOUNTS_6 values('a5363','saving',8000.00,'c2');
insert into ACCOUNTS_6 values('a7236','checking',500.00,'c33');
insert into ACCOUNTS_6 values('a8577','checking',150.00,'c20');
```

d) Show the data in Totals_6 table.

What is the amount for the customer, 'c1'?

e) Run this query

```
update Accounts_6

set amount = 1000

where accountno = 'a1523';
```

f) If the above query successfully ran, check the Totals_6 table.

g) What is the amount for the customer, 'c1'?

h) Does the amount in Totals_6 table for 'c1' agree with the total of amounts in all the accounts for 'c1' in Accounts_6 table?

Exercise 5 (10 pts)

One way to check if a specific column is being updated in a table is to use, if updating (column name).

The following trigger prevents the primary key in the BANKCUST_6 table from being updated.

```
Create Or Replace Trigger NoUpdatePK_trig

After UPDATE ON BANKCUST_6
For each row
BEGIN
 if updating ('custno')  then
      raise_application_error (-20999,'Cannot update a Primary Key');
      End if;
END;
/
show errors;
```

Now, type the following command from SQL prompt:

```
UPDATE BANKCUST_6
Set custno='c99'
Where custno='c1';
```

What is the result?

Is the custno updated?

Part 2 (25 pts)

In this, you will run a PHP program that will create an HTML form to get user's input, connect to your Oracle database tables, fetches and shows the data as given in the query.

Use the file, showSalary_form.php, place the file in the folder where you are required to put the .php files.

You must edit the code below to put your login and password. At the end of the lab session, please feel free to change your password.

For setup the webpage, you need to copy the .php file to your own webpage folder in ECC Linux system. The folder directory should be like this: /webpages/<your own username>/ , you also need to add read permission to the your .php file, so it is allow the people to access your page by browser.

You could get more ideas for setup your own page: https://wiki.helpme.engr.scu.edu/index.php/Webpage

a) **Run the program by typing the URL of the program in the browser window. Give an employee name that you have stored in your AlphaCoEmp table. Check if the program works. You could access the page like this: http://linux.students.engr.scu.edu/~<your_own_username>/ShowSalary_Form.php**

b) **Now, change the code in the showSalary_form.php file, to display the name, salary and title. Try not to display this information from the function (getSalaryFromDB) but from the main program that calls this function.**