ENSIA 2023-2024

Oracle LAB (6)

Triggers

A database trigger is a stored PL/SQL program which is automatically executed when some events occur. A trigger can be executed in response to any of the following events:

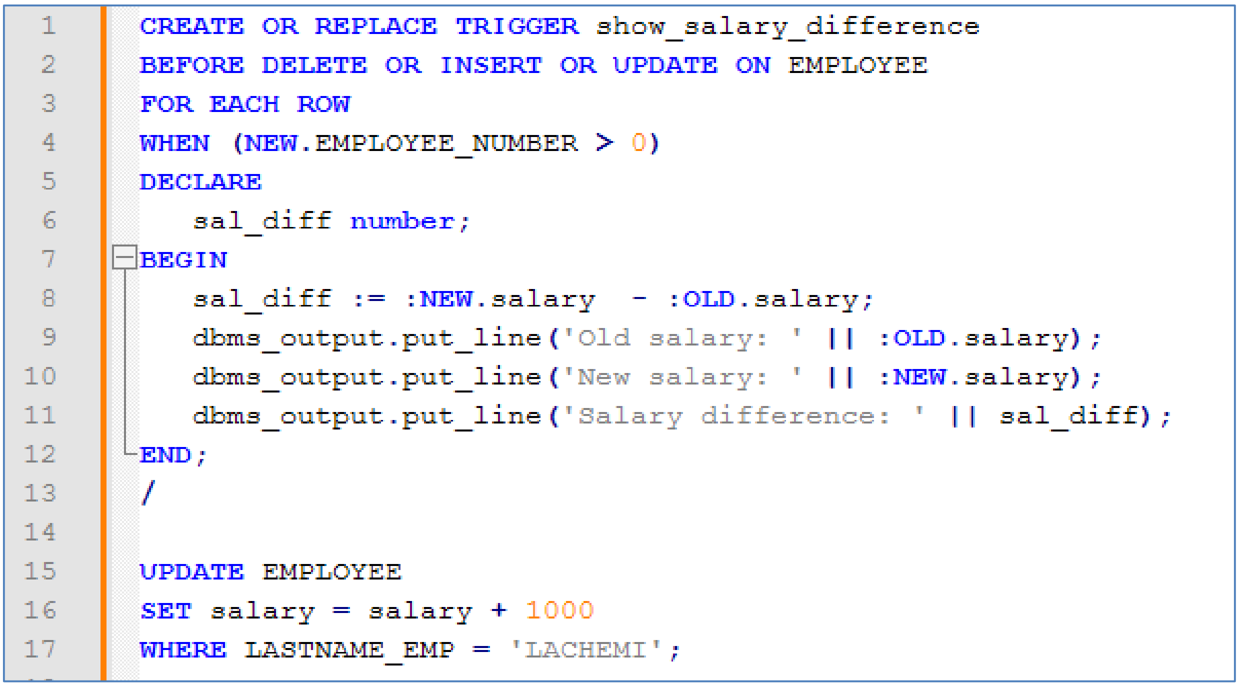
1. A database manipulation (DML) statement like DELETE, INSERT or UPDATE.
2. A database definition (DDL) statement like CREATE, ALTER or DROP.
3. A database operation like SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN. The syntax for creating a trigger is the following:

```
CREATE [OR REPLACE] TRIGGER trigger_name
 {BEFORE | AFTER | INSTEAD OF}
 {INSERT [OR] | UPDATE [OR] | DELETE}
 [OF col_name]
 ON table_name
 [REFERENCING OLD AS o NEW AS n]
 [FOR EACH ROW]
 WHEN (condition)
 BEGIN
 --- Instruction PLSQL
 END;
```

- ***CREATE [OR REPLACE] TRIGGER trigger_name****: to create or overwrite an existing trigger.*
- ***{BEFORE | AFTER | INSTEAD OF}****: the trigger execution time, i.e. before or after updating the table. INSTEAD OF is used to create a trigger on a view.*
- ***{INSERT [OR] | UPDATE [OR] | DELETE}:*** *the operation event that will execute the trigger. It is possible to specify several events by separating them with OR.*
- ***[OF col_name]****: used with the Update operation when we want to execute a trigger only when a specific column is updated.*
- ***[ON table_name]****: the name of the table on which the trigger is defined.*
- ***[REFERENCING OLD AS o NEW AS n]****: It is used to reference the old and new values of the data being changed. By default, it is possible to reference the values as :old.column_name or :new.column_name. The old values cannot be referenced when inserting a record and new values cannot be referenced when deleting a record because they do not exist.*
- ***[FOR EACH ROW]****: specifies if the trigger is executed for each affected row or only once.*
- ***WHEN (condition)****: the trigger is executed only when the affected row satisfies the condition.*

Note: To generate an exception and prevent the program from proceeding, the user can execute the procedure raise_application_error **(Num_Message, 'Message to be displayed')**; Num_Message is between 20000 and 20999.

Example: The following trigger shows the difference in the salary of an employee before and after executing DELETE, INSERT, or UPDATE statements.



```
CREATE OR REPLACE TRIGGER show_salary_difference
BEFORE DELETE OR INSERT OR UPDATE ON EMPLOYEE
FOR EACH ROW
WHEN (NEW.EMPLOYEE_NUMBER > 0)
DECLARE
   sal_diff number;
BEGIN
   sal_diff := :NEW.salary  - :OLD.salary;
   dbms_output.put_line('Old salary: ' || :OLD.salary);
   dbms_output.put_line('New salary: ' || :NEW.salary);
   dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/

UPDATE EMPLOYEE
SET salary = salary + 1000
WHERE LASTNAME_EMP = 'LACHEMI';
```

Questions

Let's assume that the tables from the previous labs are created and filled.

1. Create a trigger that displays "**a new customer is added**" after each insertion of a customer. Do the same for modification and deletion.

2. Create a trigger that displays **"a new model is added to the brand [Brand name]"** after each insertion of a model.

3. Create a trigger that verifies that when an employee's salary is modified, the new value can never be lower than the previous one.

4. Each intervention is handled by one or more employees in different periods. Create a trigger that checks that the intervention period of an employee is included in the intervention period.

5. The administrator wants to have the total number of interventions for each employee. To do this, the administrator adds an attribute: **TOTAL_INTERVENTIONS** in the employee table.

   a. Add the attribute **TOTAL_INTERVENTIONS** in the employee table.
   b. Create a **TOTAL_INTERVENTIONS_TRIGGER** that updates the TOTAL_INTERVENTIONS attribute.

6. The administrator wants to save for each month of the year the total earnings of all the interventions. Each time an intervention is added in a month, either it is the first intervention in that month, so a row is added to the table **Earning (MONTH, YEAR, TOTAL_earnings),** or it is not the first intervention in that month, so the attribute **TOTAL_earnings** is updated. Create a corresponding trigger.