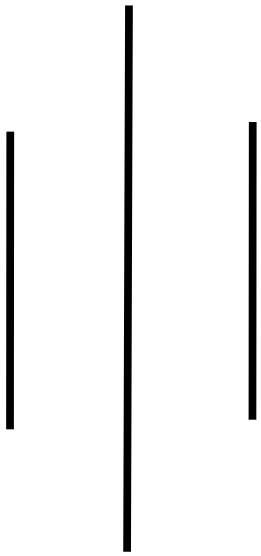
Deerwalk Institute Of Technology
Database Administration



Lab:

Submitted By:
Name: Sagar Giri
Roll No. 205

Submitted To:
Narayan Chalise

Date:

Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events:

- A database manipulation (DML) statement (DELETE, INSERT, or UPDATE).
-
- A database definition (DDL) statement (CREATE, ALTER, or DROP).
-
- A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers could be defined on the table, view, schema, or database with which the event is associated.

Syntax:

```
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)
DECLARE
   Declaration-statements
BEGIN
   Executable-statements
EXCEPTION
   Exception-handling-statements
END;
```

Exercise to calculate salary differences among employees while updating a record:

Lets consider we have following table:

```
SQL> select * from customers;

        ID NAME                      AGE ADDRESS                        SALARY
---------- -------------------- ---------- ------------------------- ----------
       205 Sagar Giri                   22 Chabahil                        4500
       206 Sameer Koirala               22 Koteshwor                       5000
       201 Asim Regmi                   22 Chabahil                        4000
```

Now, we write our trigger and execute:

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON customers
FOR EACH ROW
WHEN (NEW.ID > 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;
/
```

Results | Explain | Describe | Saved SQL | History

```
Trigger created.
```

0.08 seconds

Now when we insert any record a trigger is fired.

```
insert into customers values(216, 'Bimal Gaire', 25, 'Koteshwor', 3000);
```

Results | Explain | Describe | Saved SQL | History

```
Old salary:
New salary: 3000
Salary difference:

1 row(s) inserted.
```

0.02 seconds

Also, while updating:

```
UPDATE customers
SET salary = salary + 1000
WHERE id = 205;
```

Results | Explain | Describe | Saved SQL | History

```
Old salary: 4500
New salary: 5500
Salary difference: 1000

1 row(s) updated.
```

0.03 seconds

Another trigger which change cases of entry in table while updating or inserting.

```
CREATE OR REPLACE TRIGGER change_case
BEFORE INSERT OR UPDATE ON customers
FOR EACH ROW
BEGIN
    :NEW.address := upper(:OLD.address);
END;
/
```

Results | Explain | Describe | Saved SQL | History

```
Trigger created.
```

0.04 seconds