

Creating PL/SQL Triggers:

Triggers are stored programs, which are automatically **executed or fired** when **some event occurs**.

Triggers are 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).

Syntax to create a Trigger:

```
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;
```

Here:

- CREATE [OR REPLACE] TRIGGER trigger_name: It creates or replaces an existing trigger with the trigger_name.
- {BEFORE | AFTER | INSTEAD OF} : This specifies when the trigger would be executed. The INSTEAD OF clause is used for creating trigger on a view.
- {INSERT [OR] | UPDATE [OR] | DELETE}: This specifies the DML operation.
- [OF col_name]: This specifies the column name that would be updated.
- [ON table_name]: This specifies the name of the table associated with the trigger.
- [REFERENCING OLD AS o NEW AS n]: This allows you to refer new and old values for various DML statements, like INSERT, UPDATE, and DELETE.
- [FOR EACH ROW]: This specifies a row level trigger, i.e., the trigger would be executed for each row being affected.
Otherwise the trigger will execute just once when the SQL statement is executed, which is called a table level trigger.
- WHEN (condition): This provides a condition for rows for which the trigger would fire. This clause is valid only for row level triggers.

Example 1 :

```
CREATE OR REPLACE TRIGGER courseNameChecker
BEFORE INSERT ON courses_tbl FOR EACH ROW
```

```
DECLARE
    rowcount int;
BEGIN
    select count(*) into rowcount from courses_tbl where course_name =:
new.course_name;
    IF rowcount<>0 THEN
        raise_application_error(-20001, 'Course is already present');
    END IF;
END;
```

```
Insert into courses_tbl (course_number, course_name, course_hours)values(203,
'Geometry',45);
```

Important pointers:

OLD and NEW references are used for record level triggers; these are not available for table level triggers.

If you want to query the table in the same trigger, then you should use the AFTER keyword, because triggers can query the table or change it again only after the initial changes are applied and the table is back in a consistent state.

In case of a constraint and a trigger, the trigger is always executed first.

In case of multiple triggers with the same logic, the most recent trigger gets executed.

Deleting PL/SQL Triggers:

If you want to remove your created trigger from the database, you should use the following syntax.

Syntax:

```
DROP TRIGGER trigger_name;
```