Trigger Example:

To start with, we will be using the CUSTOMERS table we had created and used in the previous chapters:

Select \* from customers;

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

+----+----------+-----+-----------+----------+

The following program creates a **row level** trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:

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;

/

When the above code is executed at SQL prompt, it produces the following result:

Trigger created.

Here following two points are important and should be noted carefully:

* OLD and NEW references are not available for table level triggers, rather you can use them for record 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.
* Above trigger has been written in such a way that it will fire before any DELETE or INSERT or UPDATE operation on the table, but you can write your trigger on a single or multiple operations, for example BEFORE DELETE, which will fire whenever a record will be deleted using DELETE operation on the table.

Triggering a Trigger

Let us perform some DML operations on the CUSTOMERS table. Here is one INSERT statement, which will create a new record in the table:

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (7, 'Kriti', 22, 'HP', 7500.00 );

When a record is created in CUSTOMERS table, above create trigger **display\_salary\_changes** will be fired and it will display the following result:

Old salary:

New salary: 7500

Salary difference:

Because this is a new record so old salary is not available and above result is coming as null. Now, let us perform one more DML operation on the CUSTOMERS table. Here is one UPDATE statement, which will update an existing record in the table:

UPDATE customers

SET salary = salary + 500

WHERE id = 2;

When a record is updated in CUSTOMERS table, above create trigger **display\_salary\_changes** will be fired and it will display the following result:

Old salary: 1500

New salary: 2000

Salary difference: 500

# ORACLE/PLSQL: DROP TRIGGER STATEMENT

Learn how to use the **DROP TRIGGER statement** to drop a trigger in Oracle with syntax and examples.

## DESCRIPTION

Once you have created a trigger in Oracle, you might find that you need to remove it from the database. You can do this with the DROP TRIGGER statement.

## SYNTAX

The syntax to a **drop a trigger** in Oracle in Oracle/PLSQL is:

DROP TRIGGER trigger\_name;

### PARAMETERS OR ARGUMENTS

trigger\_name is the name of the trigger that you wish to drop.

## NOTE

* See also how to create [**AFTER DELETE**](http://www.techonthenet.com/oracle/triggers/after_delete.php), [**AFTER INSERT**](http://www.techonthenet.com/oracle/triggers/after_insert.php), [**AFTER UPDATE**](http://www.techonthenet.com/oracle/triggers/after_update.php), [**BEFORE DELETE**](http://www.techonthenet.com/oracle/triggers/before_delete.php), [**BEFORE INSERT**](http://www.techonthenet.com/oracle/triggers/before_insert.php), and [**BEFORE UPDATE**](http://www.techonthenet.com/oracle/triggers/before_update.php) triggers.

## EXAMPLE

Let's look at an example of how to drop a trigger in Oracle.

For example:

DROP TRIGGER orders\_before\_insert;

This example uses the ALTER TRIGGER statement to drop the trigger called orders\_before\_insert.

http://www.techonthenet.com/oracle/triggers/