

Name: Saravanakumar K  
Date: 21.11.2024

Department: Impact Training  
Task: PL/SQL

**Create a trigger that displays the count of row fire after inserting a new record to a product table (Before/after insert).**

```
create database MODEL;
```

```
use MODEL;
```

```
CREATE TABLE trigger_table(
```

```
Product_id int,
```

```
Product_name varchar(200),
```

```
Quantity int not null,
```

```
Price decimal (10,2)
```

```
);
```

```
CREATE TABLE trigger_value (
```

```
Product_id int,
```

```
Product_name varchar(200),
```

```
Quantity int not null,
```

```
Price decimal (10,2)
```

```
);
```

```
DELIMITER $$
```

```
CREATE TRIGGER after_insert_value
```

```
AFTER INSERT ON trigger_value
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO trigger_table (Product_id, Product_name, Quantity ,  
Price )
```

```
    VALUES (new.Product_id, new.Product_name, new.Quantity, new.Price);
```

```
END $$
```

Name: Saravanakumar K

Date: 21.11.2024

Department: Impact Training

Task: PL/SQL

DELIMITER ;

-- Validate

INSERT INTO trigger\_value (Product\_id, Product\_name, Quantity, Price)

VALUES ( 1, 'Memory Card', 50, 450.00);

select count(\*) from trigger\_value;

CREATE TRIGGER before\_insert\_value

BEFORE INSERT ON trigger\_value

FOR EACH ROW

BEGIN

INSERT INTO trigger\_table (Product\_id, Product\_name, Quantity ,Price )

values (new.Product\_id, new.Product\_name, new.Quantity, new.Price)

END \$\$

DELIMITER ;

INSERT INTO trigger\_value (Product\_id, Product\_name, Quantity , Price)

VALUES (2, 'iphone', 25, 75000.00);

select \* from trigger\_value;

SELECT COUNT(\*) FROM trigger\_table;

**OUTPUT:**

Name: Saravanakumar K

Date: 21.11.2024

Department: Impact Training

Task: PL/SQL

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL code:

```
46 • INSERT INTO trigger_value (Product_id, Product_name, Quantity , Price)
47 • VALUES (2, 'iphone', 25, 75000.00);
48
49 • select * from trigger_value;
50 • SELECT COUNT(*) FROM trigger_table;
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays the following data:

COUNT(*)
2

On the right side of the result grid, there are buttons for 'Result Grid' and 'Form Editor'. At the bottom right, there is a 'Read Only' status indicator.

**Create a trigger that display the count of employees whose salary got updated (after update):**

Use MODEL;

CREATE TABLE employee\_table(

Employee\_id int primary key ,

Employee\_name varchar (50),

salary decimal (10 , 2),

DOJ date

);

CREATE TABLE employee(

Employee\_id int primary key,

Employee\_name varchar (50),

salary decimal (10 , 2),

DOJ date

Name: Saravanakumar K

Date: 21.11.2024

Department: Impact Training

Task: PL/SQL

);

insert into employee\_table value (1,'ALLEN', 45000.00, '2021-11-21'),

(2,'TONY', 60000.00, '2022-01-22'),

(3,'STEVE', 28000.00, '2022-08-22'),

(4,'BARREN', 25000.00, '2023-02-18'),

(5,'NATASHA', 20000.00, '2023-08-26'),

(6,'WANDA',15000.00, '2024-11-13');

Select \* from employee\_table;

DELIMITER \$\$

CREATE TRIGGER Employee\_Trigger

AFTER UPDATE ON employee\_table

FOR EACH ROW

BEGIN

INSERT INTO employee (Employee\_id, Employee\_name, salary, DOJ)

VALUES (Old.Employee\_id, Old.Employee\_name, New.salary, Old.DOJ);

END \$\$

DELIMITER ;

drop trigger Employee\_Trigger;

Select \* from employee;

-- Validate

select \* from employee\_table;

update employee\_table

set Salary = 35000.00

where Employee\_id= 5;

select count(\*) from employee;

**OUTPUT:**

Name: Saravanakumar K

Date: 21.11.2024

Department: Impact Training

Task: PL/SQL

The screenshot shows a SQL Developer window titled 'Query 1' with a tab 'SQL File 3'. The SQL code in the editor is as follows:

```
85 DELIMITER ;
86
87 • drop trigger Employee_Trigger;
88
89 • Select * from employee;
90 -- Validate
```

Below the code editor is the 'Result Grid' tab, which displays the results of the 'Select \* from employee;' query. The grid has four columns: 'Employee\_id', 'Employee\_name', 'salary', and 'DOJ'. The data is as follows:

Employee_id	Employee_name	salary	DOJ
5	NATASHA	35000.00	2023-08-26
6	WANDA	15000.00	2024-11-13
NULL	NULL	NULL	NULL

The status bar at the bottom indicates 'employee 38 x'.

The screenshot shows a SQL Developer window titled 'Query 1' with a tab 'SQL File 3'. The SQL code in the editor is as follows:

```
92 • update employee_table
93 set Salary = 35000.00
94 where Employee_id= 1;
95
96 • select count(*) from employee;
```

Below the code editor is the 'Result Grid' tab, which displays the results of the 'select count(\*) from employee;' query. The grid has one column: 'count(\*)'. The data is as follows:

count(*)
2

The status bar at the bottom indicates 'Result 39 x'.

**Create a view that displays the details of the products from electronics where the price > 40000.**

CREATE TABLE Product\_table(

Product\_id int,

Product\_name varchar(60),

category varchar(50),

Name: Saravanakumar K

Date: 21.11.2024

Department: Impact Training

Task: PL/SQL

Price decimal (10,2)

);

insert into product\_table values (1, 'Laptop', 'Electronics', 75000.00 ),

(2, 'Mobile', 'Electronics', 33000.00 ),

(3, 'iPhone', 'Electronics', 95000.00 );

create view view\_product as select product\_id,product\_name,category,price from product\_table

where category='Electronics' and price>=40000;

select \* from view\_product;

## OUTPUT:

The screenshot displays the SQL Developer interface. The top pane shows the following SQL code:

```
77 (2, 'Mobile', 'Electronics', 25000.00 ),
78 (3, 'iPhone', 'Electronics', 95000.00 );
79 • create view view_product as select product_id,product_name,category,price
80   where category='Electronics' and price>=40000;
81 • select * from view_product;
82
```

The bottom pane shows the 'Result Grid' with the following data:

	product_id	product_name	category	price
▶	1	Laptop	Electronics	40000.00
	3	iPhone	Electronics	95000.00