

PLSQL

TRIGGERS AND VIEW

1.TRIGGERS:

```
use practice;
```

```
select * from employee;
```

```
create table employee(
```

```
  ID INT auto_increment primary key,
```

```
  employeeName varchar(50),
```

```
  employeeDep varchar(50),
```

```
  Location varchar(50)
```

```
);
```

```
Create table tn_employee_count(
```

```
  id Int auto_increment primary key,
```

```
  updated_at datetime not null,
```

```
  tn_count int not null
```

```
);
```

```
select*from tn_employee_count;
```

```
insert into employee
```

```
values('121','Alex','Software Engineer','TN'),
```

```
('122','Mahi','Software Engineer','Hyd');
```

```
insert into employee (ID ,employeeName,employeeDep,Location)
```

```
values('127','lakshman','Finance','TN');
```

```
select *from employee;
```

```
DELIMITER $$
```

```
CREATE TRIGGER after_emp_inc
```

```
after insert on employee
```

```
for each row
```

```
begin
```

```
declare tn_count INT;
```

```
select count(*)into tn_count from employee where Location ='TN';
```

```
insert into tn_employee_count (updated_at,tn_count)
```

```
values(now(),tn_count);
```

```
END ;
```

```
delimiter ;
```

```
select * from tn_employee_count;
```

OUTPUT

Navigator: PLSQL Practice Trigger and view* Views

SCHEMAS

Filter objects

- new_schema
- practice
 - Tables
 - electronic_products
 - employee
 - employee_view
 - productdetails
 - Views
 - Stored Procedures
 - Functions
- products
- sakila
- sys
- world

Administration Schemas

```
4  
5 • select * from employee;  
6  
7 • create table employee(  
8   ID INT auto_increment primary key,  
9   employeeName varchar(50),  
10  employeeDeo varchar(50),
```

Result Grid

ID	employeeName	employeeDep	Location
121	Alex	Software Engineer	TN
122	Mahi	Software Engineer	Hyd
NULL	NULL	NULL	NULL

Navigator: PLSQL Practice Trigger and view* Views

SCHEMAS

Filter objects

- new_schema
- practice
 - Tables
 - electronic_products
 - employee
 - employee_view
 - productdetails
 - Views
 - Stored Procedures
 - Functions
- products
- sakila
- sys
- world

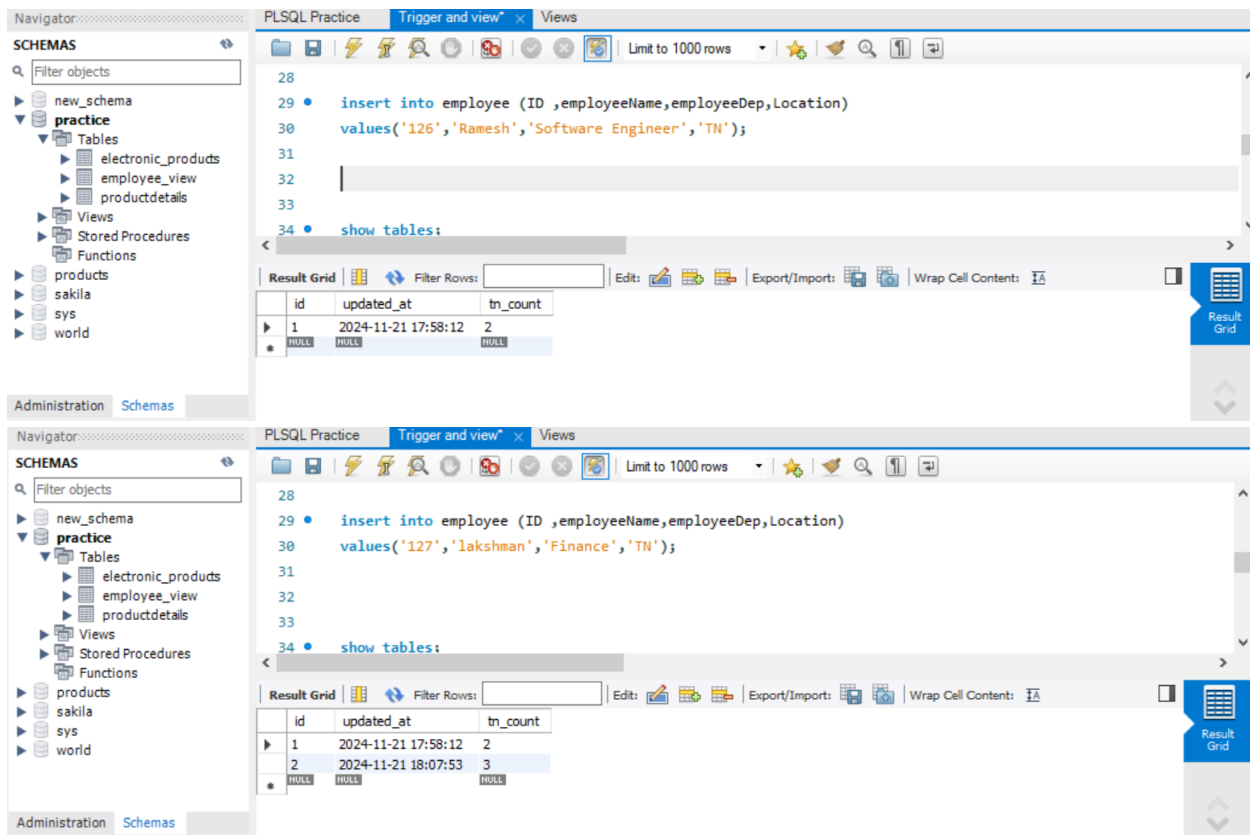
Administration Schemas

```
27  
28  
29 • insert into employee (ID ,employeeName,employeeDep,Location)  
30 values('126','Ramesh','Software Engineer','TN');  
31  
32  
33
```

Result Grid

ID	employeeName	employeeDep	Location
121	Alex	Software Engineer	TN
122	Mahi	Software Engineer	Hyd
126	Ramesh	Software Engineer	TN
NULL	NULL	NULL	NULL

employee 5 x Apply Revert



2.VIEWS

```

create table employee_view(
ID INT auto_increment primary key,
employeeName varchar(50),
employeeDep varchar(50),
Location varchar(50),
Team varchar(50)
);

```

```

select *from employee_view;

```

```

insert into employee_view
values('121','Alex','Software Engineer','Chennai','TN'),

```

```
('122','Mahi','Software Engineer','Siruser','TN'),  
('123','Yuva','HR','Banglore','KA'),  
('124','Tito','HR','Chennai','TN');
```

```
create view Chennai_TN_view  
as (select*from employee_view where Location='Chennai' and Team ='TN');
```

```
select * from Chennai_TN_view;
```

```
/*
```

Create a view That displays product from electronics and price>1000

```
*/
```

```
/*
```

Create a view return a employee details from chennai location and TN team

```
*/
```

```
use practice;
```

```
create table Electronic_products(
```

```
product_id numeric,
```

```
product_name varchar(20),
```

```
qty numeric,
```

```
price numeric
```

```
);
```

```
insert into Electronic_products values(101,'Pen',10,100),(102,'Chart paper',20,35);
```

```
create view Product_filter_view
```

```
as(select*from Electronic_products where price>50);
```

```
select *from product_filter_view;
```

The screenshot displays a PL/SQL IDE interface. On the left, a 'SCHEMAS' tree shows a 'practice' schema containing tables like 'electronic_products', 'employee_view', and 'productdetails'. The main editor window, titled 'PLSQL Practice', shows a SQL script with an 'insert into employee_view' statement. Below the script, the 'Result Grid' displays the following data:

ID	employeeName	employeeDep	Location	Team
121	Alex	Software Engineer	Chennai	TN
122	Mahi	Software Engineer	Siruser	TN
123	Yuva	HR	Banglore	KA
124	Tito	HR	Chennai	TN
NULL	NULL	NULL	NULL	NULL

Navigation

SCHEMAS

Filter objects

new_schema

practice

Tables

Views

Stored Procedures

Functions

products

sakila

sys

world

PLSQL Practice

Trigger and view

Views

Limit to 1000 rows

52 • insert into Electronic_products values(101,'Pen',10,100),(102,'Chart paper',20,35);

53

54 • create view Product_filter_view

55 as(select*from Electronic_products where price>50);

56

57 • select *from product_filter_view;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	product_id	product_name	qty	price
▶	101	Pen	10	100

Administration Schemas