

1.Create a trigger that display the count of rows after inserting a new record t a employee

```
create database triggers;
```

```
use triggers;
```

```
create table employee (  
    emp_id int PRIMARY KEY AUTO_INCREMENT,  
    empname varchar(100) NOT NULL,  
    salary DECIMAL(10, 2) NOT NULL,  
    practice varchar(50),  
    location varchar(100)  
);
```

```
create table employeecount (  
    empcount int DEFAULT 0  
);
```

```
insert into employeecount (empcount) values (0);
```

```
delimiter $$
```

```
create trigger after_employee_insert  
after insert on employee  
for each row  
begin  
    update employeecount  
    set empcount = empcount + 1;  
end $$;  
delimiter ;
```

```
insert into employee (empname, salary, practice, location)  
values ('Abi', 60000, 'TN', 'chennai'),  
('Keerthu', 67000, 'SL', 'banglore'),  
('Ruthu', 23000, 'TN', 'chennai');
```

```
insert into employee (empname, salary, practice, location)  
values ('jaya', 70000, 'TN', 'kochi');  
set SQL_SAFE_UPDATES = 0;
```

```
select * from employeecount;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	empcount			
▶	5			

2.Create a trigger that display the count of employees whose salary get updated .

```
create table employeesalarycount (  
    salary_count int default 0  
);
```

```
insert into employeesalarycount (salary_count) values (0);
```

```
delimiter $$  
create trigger salary_update  
after update on employee  
for each row  
begin  
    if old.salary != new.salary then  
        update employeesalarycount  
        set salary_count = salary_count + 1;  
    end if;  
end $$;  
delimiter ;
```

```
update employee  
set salary = 65000  
where emp_id = 5;
```

```
select * from employeesalarycount;
```

Result Grid		Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
	salary_count			
▶	2			

3.create a view return employee in chennai location and tn practice

create view chennai\_TN as

select \*

from employee

where location = 'Chennai' and practice = 'TN';

select \* from chennai\_TN;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	emp_id	empname	salary	practice	location
▶	4	Abi	60000.00	TN	chennai
	6	Ruthu	23000.00	TN	chennai

4.create a view that display the details of product from electronics price>9000

create table electronics (

e\_id int PRIMARY KEY AUTO\_INCREMENT,

name varchar(100) not null,

price int not null

);

ALTER TABLE electronics

MODIFY COLUMN price INT NOT NULL;

create view electronic as

select \*

from electronics

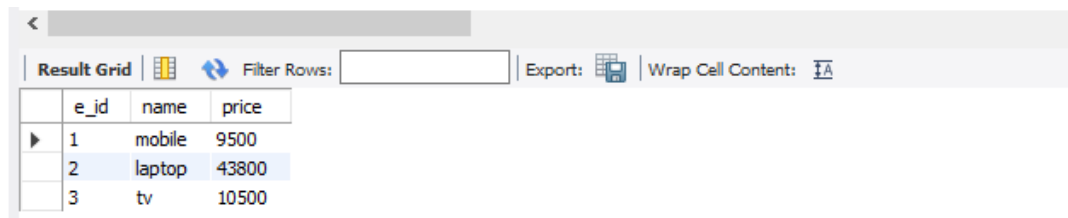
where price > 9000;

truncate table electronics;

insert into electronics (name, price)

values

```
('mobile', 9500),  
('laptop', 43800),  
('tv', 10500),  
('iron box', 1200);  
select * from electronic;
```



The screenshot shows a database query result interface. At the top, there is a toolbar with a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' button. Below the toolbar is a table with three columns: 'e\_id', 'name', and 'price'. The table contains three rows of data: a mobile phone with ID 1 and price 9500, a laptop with ID 2 and price 43800, and a TV with ID 3 and price 10500. The second row (laptop) is highlighted in blue.

	e_id	name	price
▶	1	mobile	9500
	2	laptop	43800
	3	tv	10500