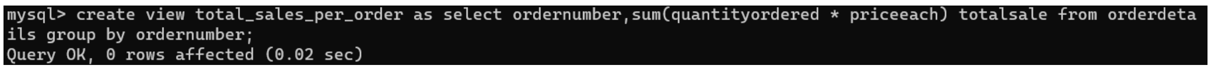
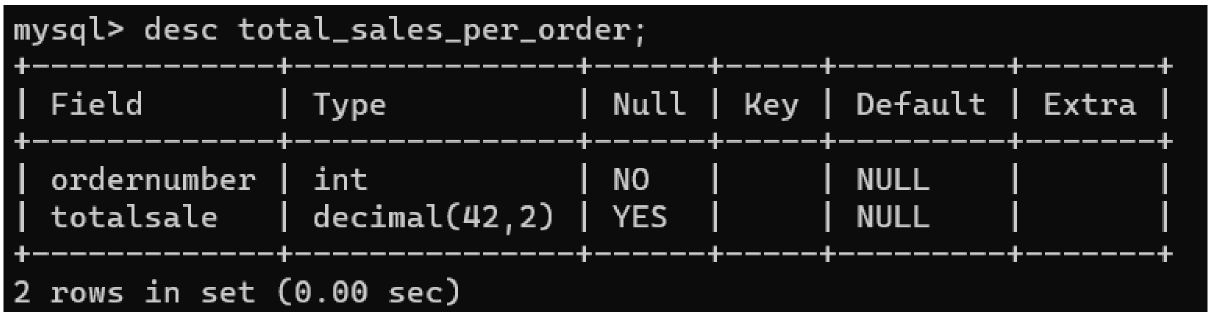
LAB 4

Q1. Create a view that represents total sales per order from the orders table.



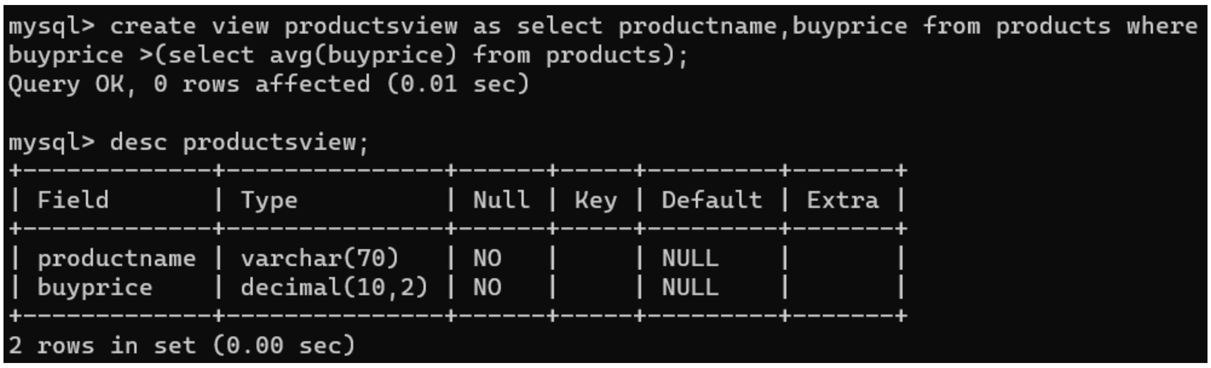
```
mysql> create view total_sales_per_order as select ordernumber,sum(quantityordered * priceeach) totalsale from orderdeta
ils group by ordernumber;
Query OK, 0 rows affected (0.02 sec)
```



```
mysql> desc total_sales_per_order;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| ordernumber | int           | NO   |     | NULL    |       |
| totalsale   | decimal(42,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Q2. Create a view that contains products whose buy prices are higher than the

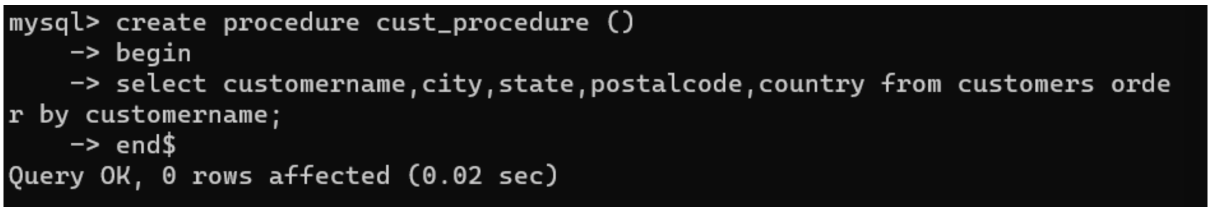
average price of all products.



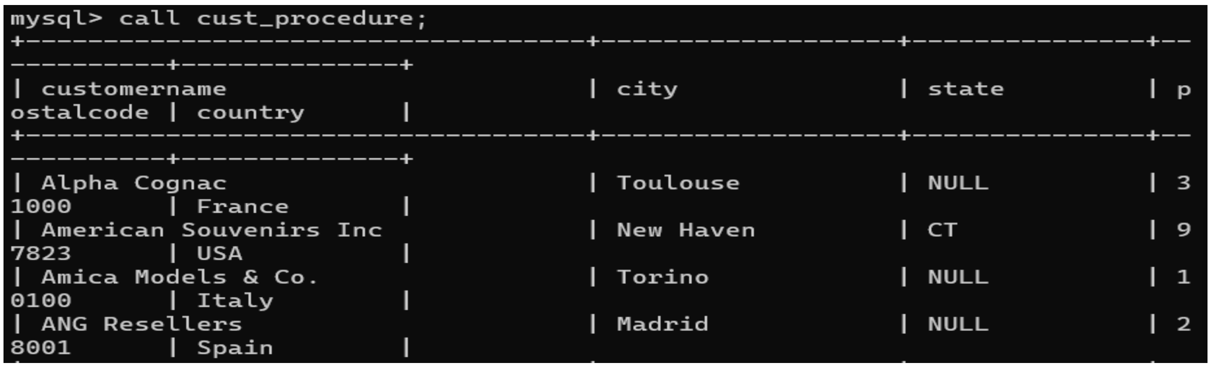
```
mysql> create view productsview as select productname,buyprice from products where
buyprice >(select avg(buyprice) from products);
Query OK, 0 rows affected (0.01 sec)

mysql> desc productsview;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| productname | varchar(70)   | NO   |     | NULL    |       |
| buyprice    | decimal(10,2) | NO   |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Q3. create a procedure to select the name, city, state, postalcode and country

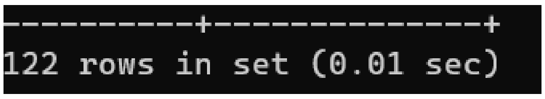
from the customers table in the alphabetical order of name.



```
mysql> create procedure cust_procedure ()
    -> begin
    -> select customername,city,state,postalcode,country from customers orde
r by customername;
    -> end$
Query OK, 0 rows affected (0.02 sec)
```



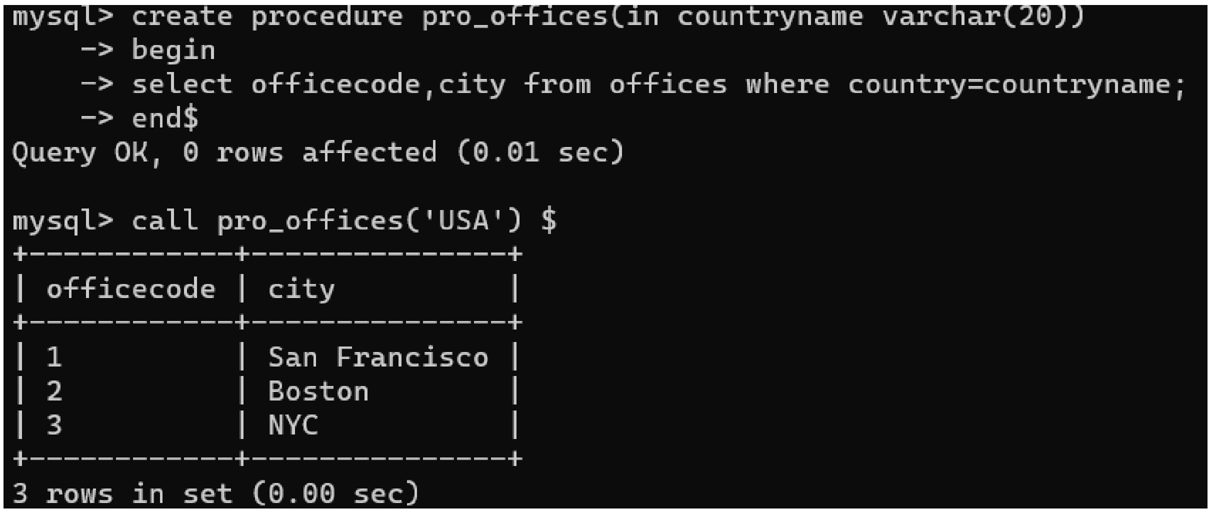
```
mysql> call cust_procedure;
+------------------------------------+-------------------+----------------+--
----------+--------------+
| customername                       | city              | state          | p
ostalcode | country      |
+------------------------------------+-------------------+----------------+--
----------+--------------+
| Alpha Cognac                       | Toulouse          | NULL           | 3
1000       | France       |
| American Souvenirs Inc             | New Haven         | CT             | 9
7823       | USA          |
| Amica Models & Co.                 | Torino            | NULL           | 1
0100       | Italy        |
| ANG Resellers                      | Madrid            | NULL           | 2
8001       | Spain        |
```



```
----------+--------------+
122 rows in set (0.01 sec)
```

Q4. Create a stored procedure that finds all offices that locate in a country

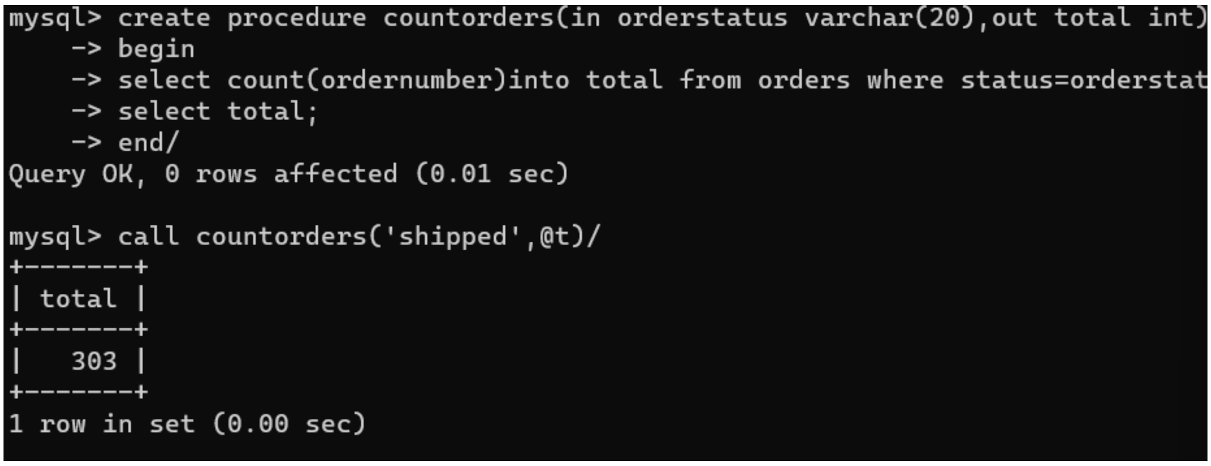
specified by the input parameter countryName.



```
mysql> create procedure pro_offices(in countryname varchar(20))
    -> begin
    -> select officecode,city from offices where country=countryname;
    -> end$
Query OK, 0 rows affected (0.01 sec)

mysql> call pro_offices('USA') $
+------------+---------------+
| officecode | city          |
+------------+---------------+
| 1          | San Francisco |
| 2          | Boston        |
| 3          | NYC           |
+------------+---------------+
3 rows in set (0.00 sec)
```

Q5. Create a stored procedure to find the number of orders that already

shipped by passing the orderstatus into the procedure



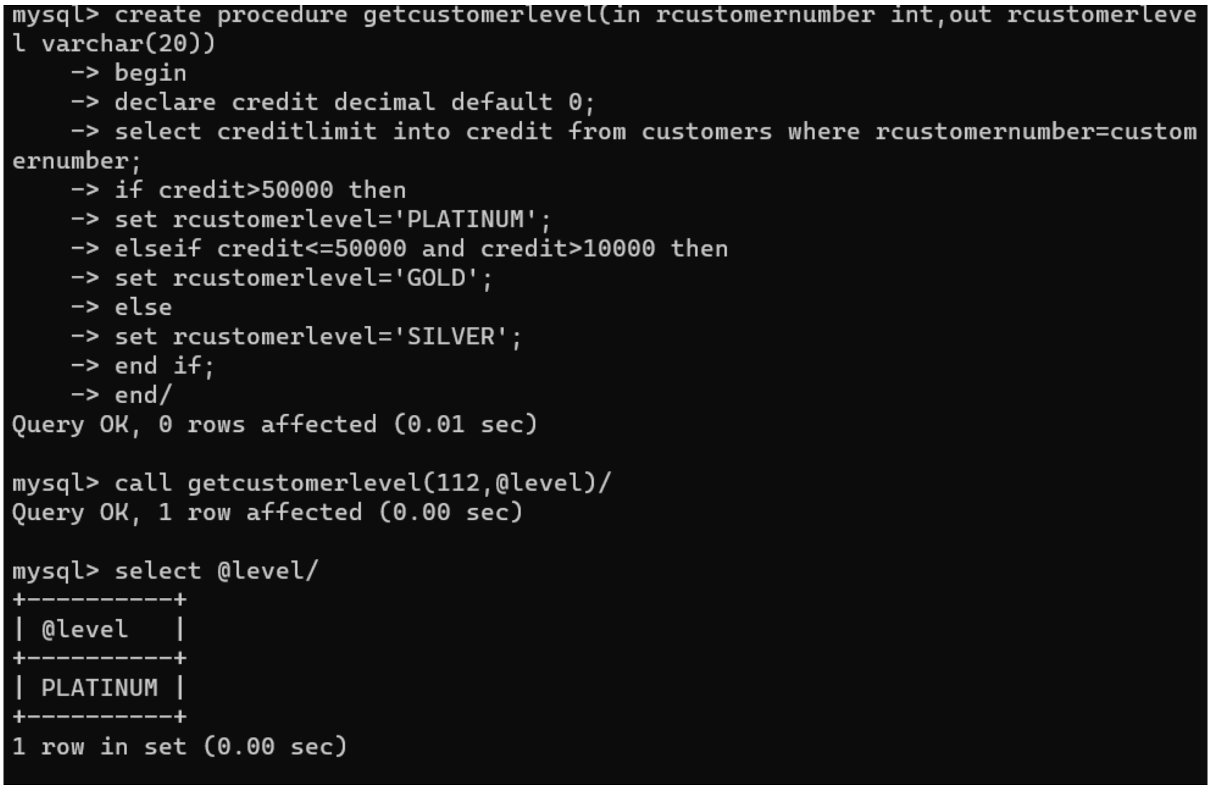
```
mysql> create procedure countorders(in orderstatus varchar(20),out total int)
    -> begin
    -> select count(ordernumber)into total from orders where status=orderstat
    -> select total;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call countorders('shipped',@t)/
+-------+
| total |
+-------+
|   303 |
+-------+
1 row in set (0.00 sec)
```

Q6. Create a stored procedure using if statement which inputs the customernumber and selects the creditlimit and displays the customerlevelbased on the following condition

 If the credit is greater than 50,000, the level of the customer is PLATINUM.

 If the credit is less than or equal 50,000 and greater than 10,000, then the level of customer is GOLD.

 Otherwise, the level of the customer is SILVER.



```
mysql> create procedure getcustomerlevel(in rcustomernumber int,out rcustomerleve
l varchar(20))
    -> begin
    -> declare credit decimal default 0;
    -> select creditlimit into credit from customers where rcustomernumber=custom
ernumber;
    -> if credit>50000 then
    -> set rcustomerlevel='PLATINUM';
    -> elseif credit<=50000 and credit>10000 then
    -> set rcustomerlevel='GOLD';
    -> else
    -> set rcustomerlevel='SILVER';
    -> end if;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call getcustomerlevel(112,@level)/
Query OK, 1 row affected (0.00 sec)

mysql> select @level/
+----------+
| @level   |
+----------+
| PLATINUM |
+----------+
1 row in set (0.00 sec)
```

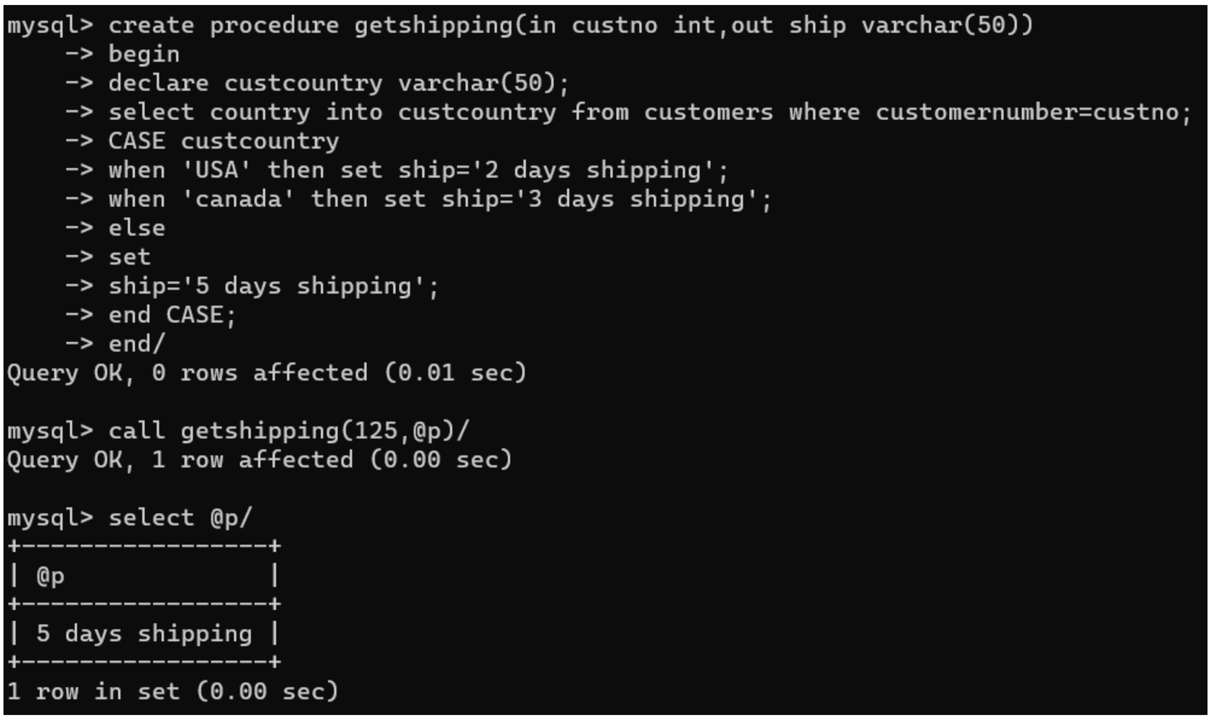
Q7. Create a stored procedure using case which inputs the customernumber and selects the country and displays the shipping time based on the following condition

 If the customer locates in USA , the shipping time is 2-day shipping .

 If the customer locates in Canada , the shipping time is 3-day

shipping .

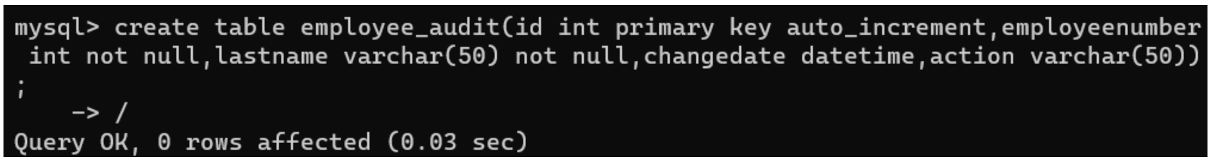
 The customers from other countries have 5-day shipping



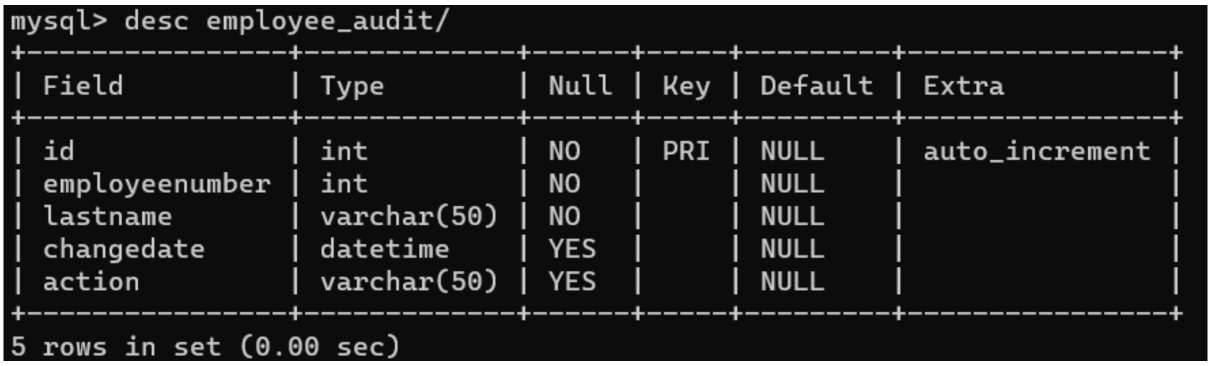
```
mysql> create procedure getshipping(in custno int,out ship varchar(50))
    -> begin
    -> declare custcountry varchar(50);
    -> select country into custcountry from customers where customernumber=custno;
    -> CASE custcountry
    -> when 'USA' then set ship='2 days shipping';
    -> when 'canada' then set ship='3 days shipping';
    -> else
    -> set
    -> ship='5 days shipping';
    -> end CASE;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> call getshipping(125,@p)/
Query OK, 1 row affected (0.00 sec)

mysql> select @p/
+-----------------+
| @p              |
+-----------------+
| 5 days shipping |
+-----------------+
1 row in set (0.00 sec)
```

Q8. Create a table employees_audit with the following data

Column	Datatype	Constraint
id	int	Primarykey,autoincrement
employeenumber	int	Not null
lastname	Varchar(50)	Not null
changedat	datetime	
action	Varchar(50)	

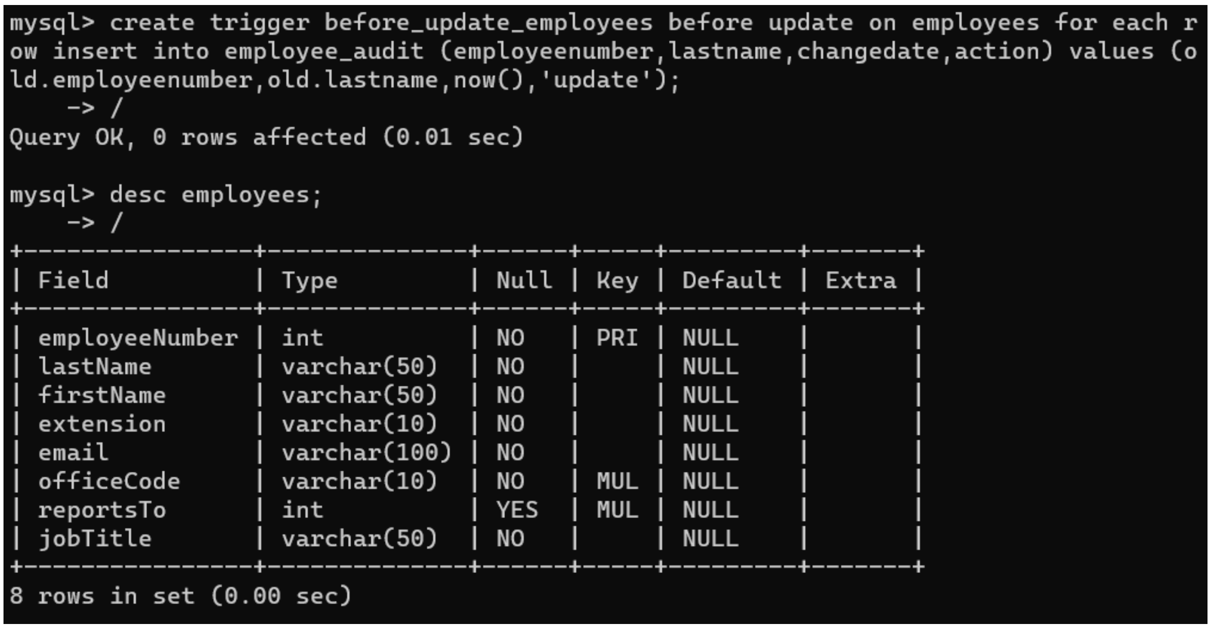


```
mysql> create table employee_audit(id int primary key auto_increment,employeenumber
 int not null,lastname varchar(50) not null,changedate datetime,action varchar(50))
;
    -> /
Query OK, 0 rows affected (0.03 sec)
```



```
mysql> desc employee_audit/
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| id             | int         | NO   | PRI | NULL    | auto_increment |
| employeenumber | int         | NO   |     | NULL    |                |
| lastname       | varchar(50) | NO   |     | NULL    |                |
| changedate     | datetime    | YES  |     | NULL    |                |
| action         | varchar(50) | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

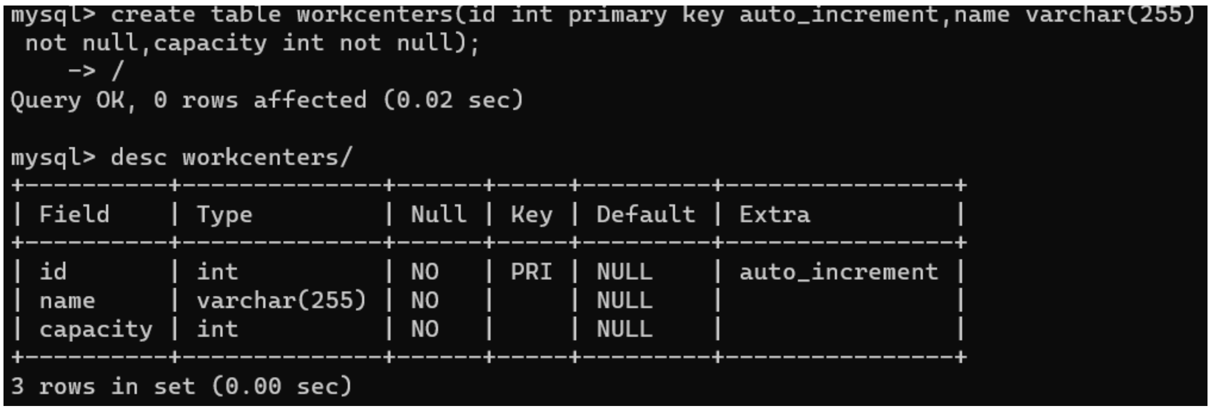
Q9. Create a trigger which will insert into the employees_audit table before updating the employees table.action should be set as “update”,employeenumber and lastname should be set with the old value and changedat should be set with the current date and time. Update rows in the employees table and check the employees_audit table.



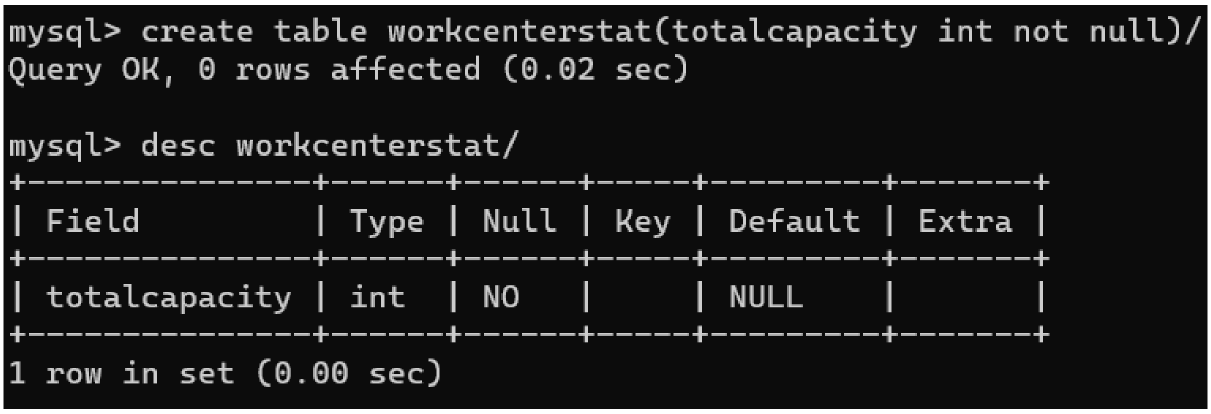
```
mysql> create trigger before_update_employees before update on employees for each r
ow insert into employee_audit (employeenumber,lastname,changedate,action) values (o
ld.employeenumber,old.lastname,now(),'update');
    -> /
Query OK, 0 rows affected (0.01 sec)

mysql> desc employees;
    -> /
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| employeeNumber | int          | NO   | PRI | NULL    |       |
| lastName       | varchar(50)  | NO   |     | NULL    |       |
| firstName      | varchar(50)  | NO   |     | NULL    |       |
| extension      | varchar(10)  | NO   |     | NULL    |       |
| email          | varchar(100) | NO   |     | NULL    |       |
| officeCode     | varchar(10)  | NO   | MUL | NULL    |       |
| reportsTo      | int          | YES  | MUL | NULL    |       |
| jobTitle       | varchar(50)  | NO   |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

LAB 5

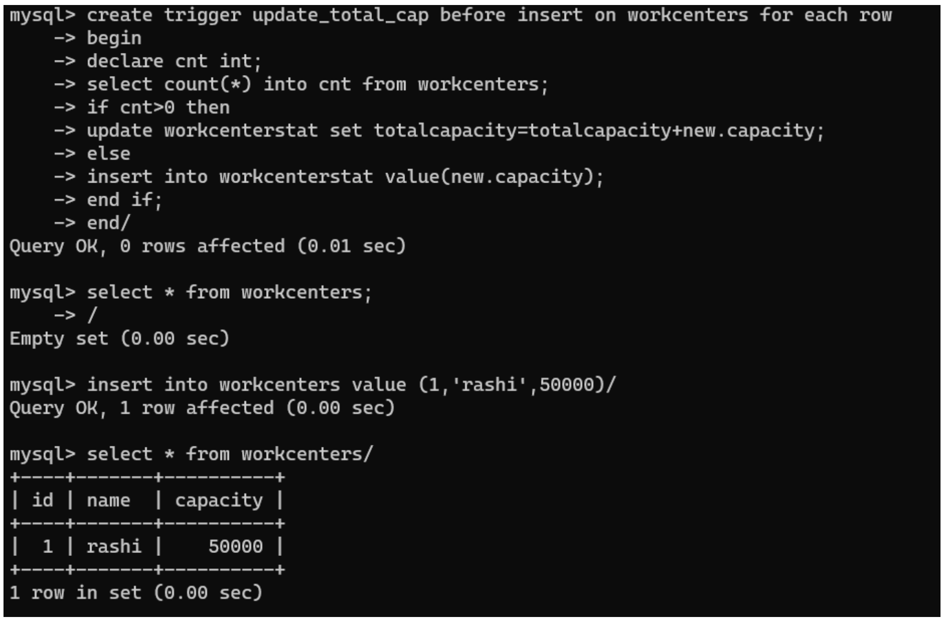
Q.1. Create a table Workcenters with the following data.



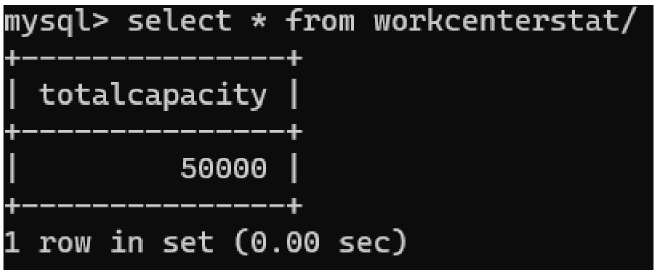
```
mysql> create table workcenters(id int primary key auto_increment,name varchar(255)
 not null,capacity int not null);
    -> /
Query OK, 0 rows affected (0.02 sec)

mysql> desc workcenters/
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int          | NO   | PRI | NULL    | auto_increment |
| name     | varchar(255) | NO   |     | NULL    |                |
| capacity | int          | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```



```
mysql> create table workcenterstat(totalcapacity int not null)/
Query OK, 0 rows affected (0.02 sec)

mysql> desc workcenterstat/
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| totalcapacity | int  | NO   |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
1 row in set (0.00 sec)
```



```
mysql> create trigger update_total_cap before insert on workcenters for each row
    -> begin
    -> declare cnt int;
    -> select count(*) into cnt from workcenters;
    -> if cnt>0 then
    -> update workcenterstat set totalcapacity=totalcapacity+new.capacity;
    -> else
    -> insert into workcenterstat value(new.capacity);
    -> end if;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> select * from workcenters;
    -> /
Empty set (0.00 sec)

mysql> insert into workcenters value (1,'rashi',50000)/
Query OK, 1 row affected (0.00 sec)

mysql> select * from workcenters/
+----+-------+----------+
| id | name  | capacity |
+----+-------+----------+
|  1 | rashi |    50000 |
+----+-------+----------+
1 row in set (0.00 sec)
```



```
mysql> select * from workcenterstat/
+---------------+
| totalcapacity |
+---------------+
|         50000 |
+---------------+
1 row in set (0.00 sec)
```

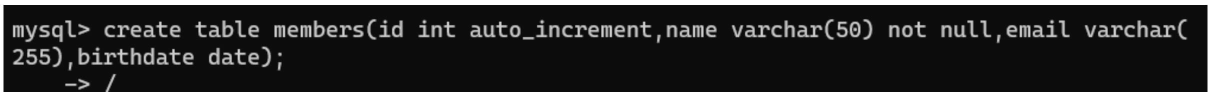
Q2. Create a table Members with the following data

Column	Datatype	Constraint
id	int	Autoincrement Primary key
name	Varchar(50)	Not Null
email	Varchar(255)	
birthdate	date	

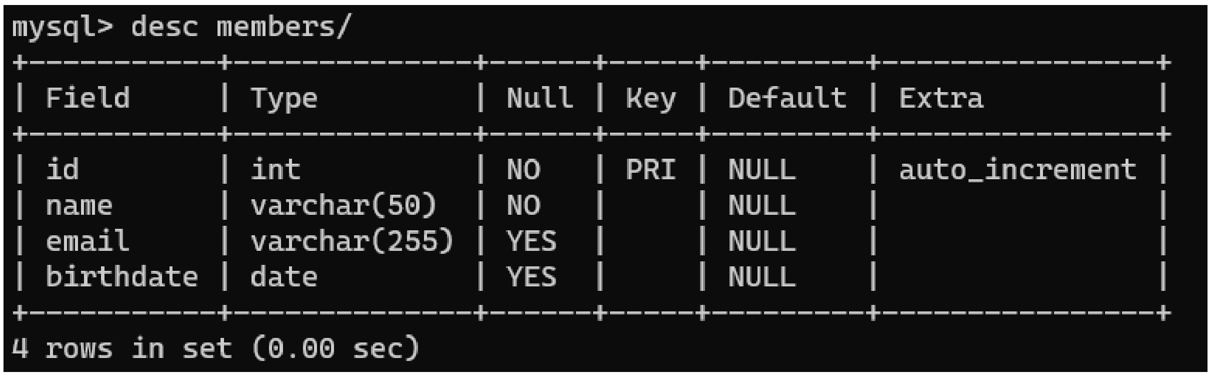
Create a table Reminders with the following data

Column	Datatype	Constraint
id	int	Autoincrement Primary key
memberId	int	Primary Key
Message	Varchar(255)	Not Null

MEMBERS TABLE:

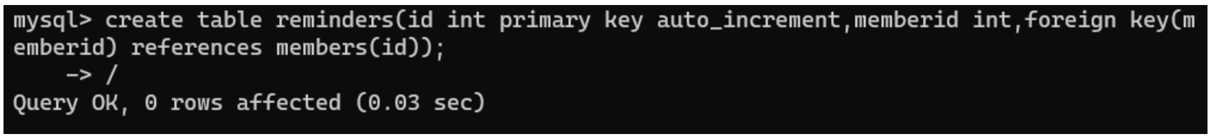


```
mysql> create table members(id int auto_increment,name varchar(50) not null,email varchar(
255),birthdate date);
    -> /
```

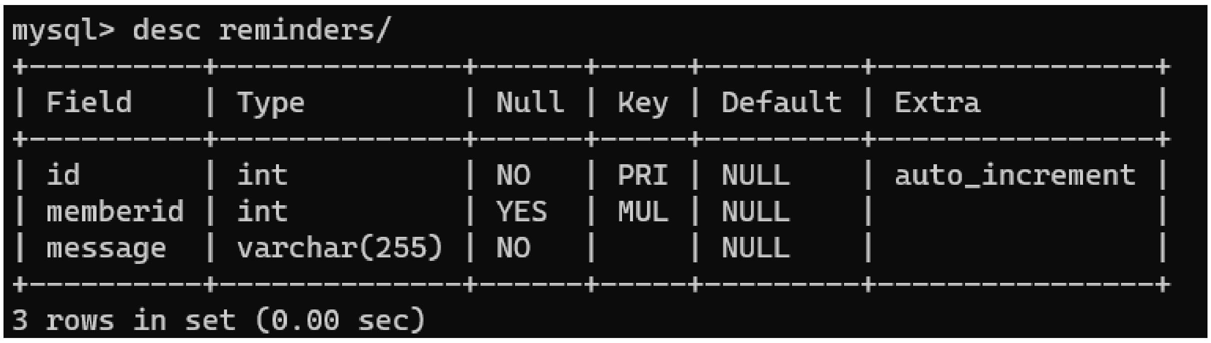


```
mysql> desc members/
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| id        | int          | NO   | PRI | NULL    | auto_increment |
| name      | varchar(50)  | NO   |     | NULL    |                |
| email     | varchar(255) | YES  |     | NULL    |                |
| birthdate | date         | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

REMINDERS TABLE:

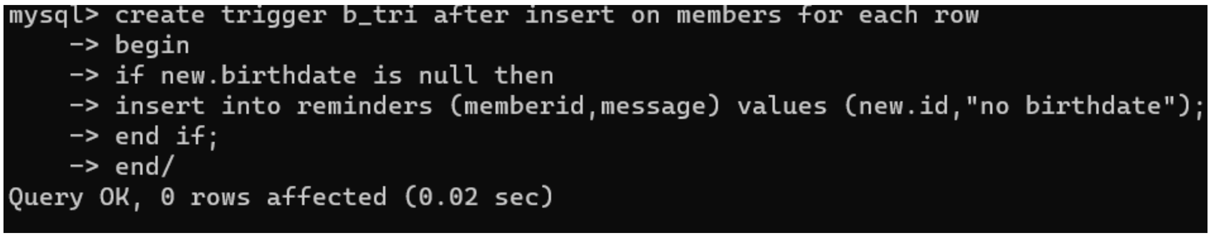


```
mysql> create table reminders(id int primary key auto_increment,memberid int,foreign key(m
emberid) references members(id));
    -> /
Query OK, 0 rows affected (0.03 sec)
```



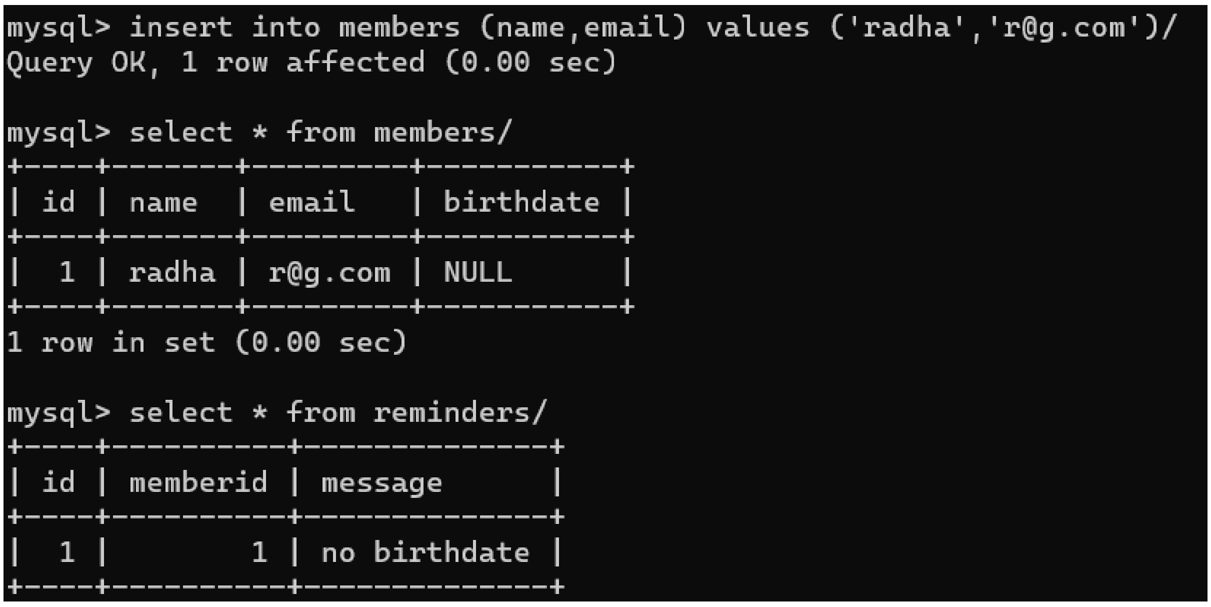
```
mysql> desc reminders/
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int          | NO   | PRI | NULL    | auto_increment |
| memberid | int          | YES  | MUL | NULL    |                |
| message  | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

CREATE TRIGGER:

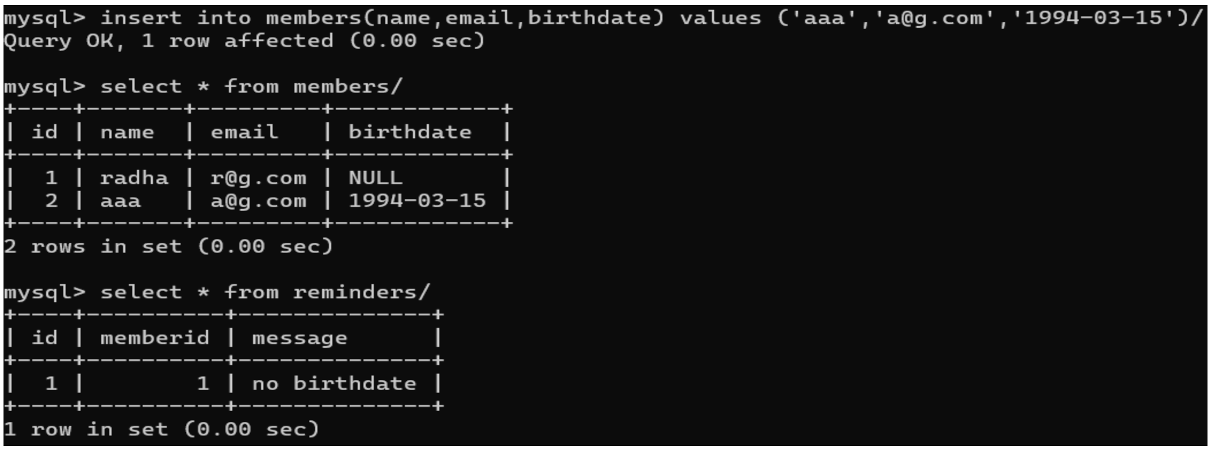


```
mysql> create trigger b_tri after insert on members for each row
    -> begin
    -> if new.birthdate is null then
    -> insert into reminders (memberid,message) values (new.id,"no birthdate");
    -> end if;
    -> end/
Query OK, 0 rows affected (0.02 sec)
```

INSERT VALUES IN MEMBERS table:



```
mysql> insert into members (name,email) values ('radha','r@g.com')/
Query OK, 1 row affected (0.00 sec)

mysql> select * from members/
+----+-------+---------+-----------+
| id | name  | email   | birthdate |
+----+-------+---------+-----------+
|  1 | radha | r@g.com | NULL      |
+----+-------+---------+-----------+
1 row in set (0.00 sec)

mysql> select * from reminders/
+----+----------+--------------+
| id | memberid | message      |
+----+----------+--------------+
|  1 |        1 | no birthdate |
+----+----------+--------------+
```


```
mysql> insert into members(name,email,birthdate) values ('aaa','a@g.com','1994-03-15')/
Query OK, 1 row affected (0.00 sec)

mysql> select * from members/
+----+-------+---------+------------+
| id | name  | email   | birthdate  |
+----+-------+---------+------------+
|  1 | radha | r@g.com | NULL       |
|  2 | aaa   | a@g.com | 1994-03-15 |
+----+-------+---------+------------+
2 rows in set (0.00 sec)

mysql> select * from reminders/
+----+----------+--------------+
| id | memberid | message      |
+----+----------+--------------+
|  1 |        1 | no birthdate |
+----+----------+--------------+
1 row in set (0.00 sec)
```

Q3. Create a table Sales with the following data

Column	Datatype	Constraint
Id	int	Autoincrement Primary key
Product	Varchar(50)	Not Null
Quantity	Int	Not Null
fiscalYear	smallint	Not Null
fiscalMonth	Tinyint	Not Null
Remarks	Varchar(255)	

INSERT 3 rows in the columns product, quantity, fiscalYear, fiscalMonth the following

VALUES

1. ‘2003 Harley-Davidson Eagle Drag Bike’,120, 2020,1

2. ‘1969 Corvair Monza’, 150,2020,1

3.’1970 Plymouth Hemi Cuda’, 200,2020,1

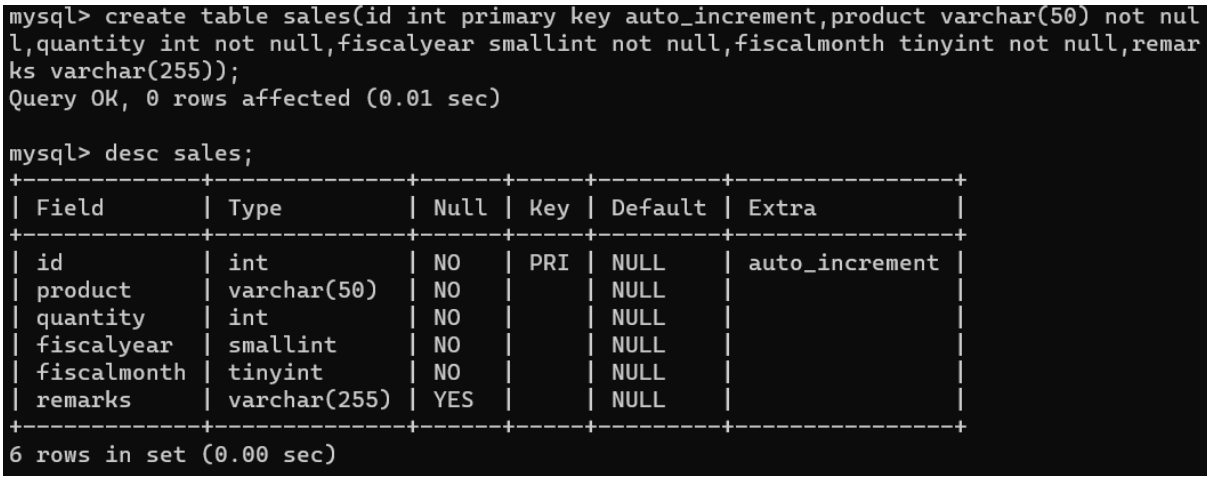
Create a before update trigger which does the following

If the value in the quantity column is updated to a new value that is 3 times greater than

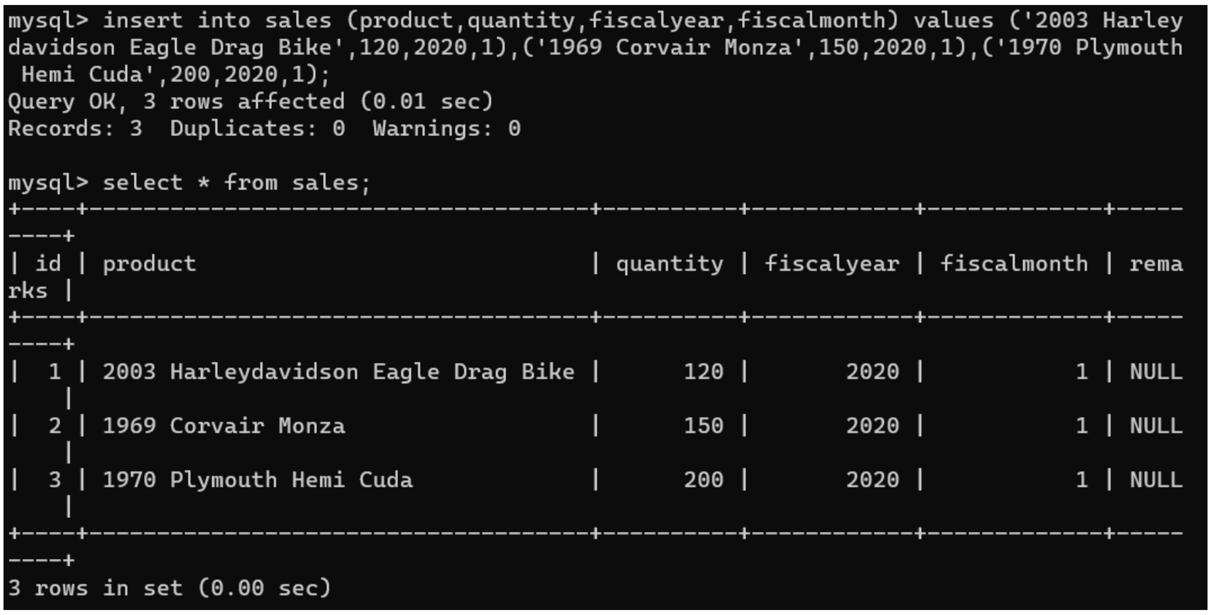
the current value, the remarks column of that row should be updated with a message

“New quantity cannot be 3 times greater than the current quantity”;

Update the row and check with different values.

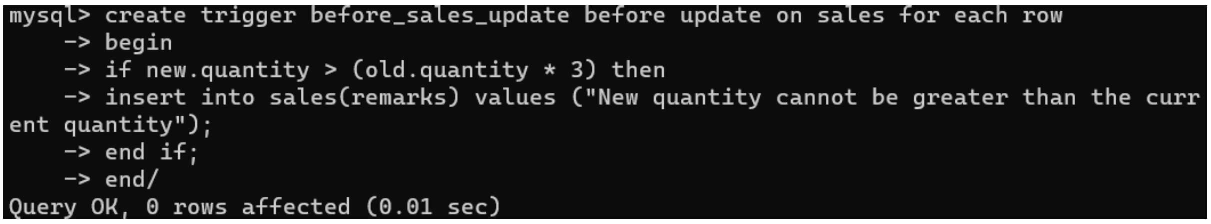


```
mysql> create table sales(id int primary key auto_increment,product varchar(50) not nul
l,quantity int not null,fiscalyear smallint not null,fiscalmonth tinyint not null,remar
ks varchar(255));
Query OK, 0 rows affected (0.01 sec)

mysql> desc sales;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| product     | varchar(50)  | NO   |     | NULL    |                |
| quantity    | int          | NO   |     | NULL    |                |
| fiscalyear  | smallint     | NO   |     | NULL    |                |
| fiscalmonth | tinyint      | NO   |     | NULL    |                |
| remarks     | varchar(255) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```



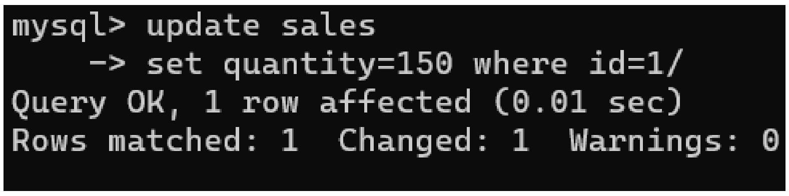
```
mysql> insert into sales (product,quantity,fiscalyear,fiscalmonth) values ('2003 Harley
davidson Eagle Drag Bike',120,2020,1),('1969 Corvair Monza',150,2020,1),('1970 Plymouth
 Hemi Cuda',200,2020,1);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from sales;
+----+-----------------------------------+----------+------------+-------------+-----
----+
| id | product                           | quantity | fiscalyear | fiscalmonth | rema
rks |
+----+-----------------------------------+----------+------------+-------------+-----
----+
|  1 | 2003 Harleydavidson Eagle Drag Bike |      120 |       2020 |           1 | NULL
    |
|  2 | 1969 Corvair Monza                |      150 |       2020 |           1 | NULL
    |
|  3 | 1970 Plymouth Hemi Cuda           |      200 |       2020 |           1 | NULL
    |
+----+-----------------------------------+----------+------------+-------------+-----
----+
3 rows in set (0.00 sec)
```

Create TRIGGER:

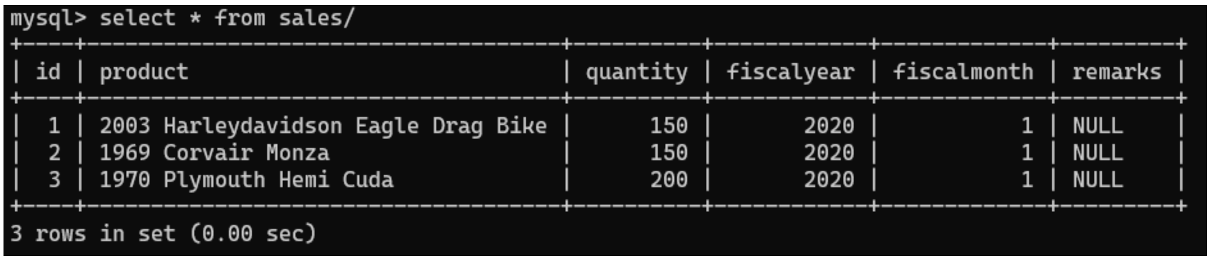


```
mysql> create trigger before_sales_update before update on sales for each row
    -> begin
    -> if new.quantity > (old.quantity * 3) then
    -> insert into sales(remarks) values ("New quantity cannot be greater than the curr
ent quantity");
    -> end if;
    -> end/
Query OK, 0 rows affected (0.01 sec)
```

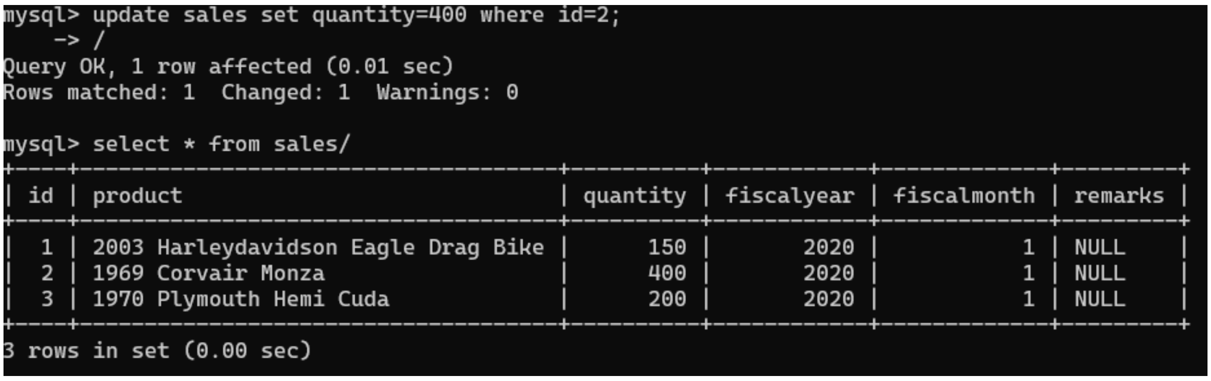
Update sales table:



```
mysql> update sales
    -> set quantity=150 where id=1/
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```



```
mysql> select * from sales/
+----+-------------------------------------+----------+------------+-------------+---------+
| id | product                             | quantity | fiscalyear | fiscalmonth | remarks |
+----+-------------------------------------+----------+------------+-------------+---------+
|  1 | 2003 Harleydavidson Eagle Drag Bike |      150 |       2020 |           1 | NULL    |
|  2 | 1969 Corvair Monza                  |      150 |       2020 |           1 | NULL    |
|  3 | 1970 Plymouth Hemi Cuda             |      200 |       2020 |           1 | NULL    |
+----+-------------------------------------+----------+------------+-------------+---------+
3 rows in set (0.00 sec)
```



```
mysql> update sales set quantity=400 where id=2;
    -> /
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sales/
+----+-------------------------------------+----------+------------+-------------+---------+
| id | product                             | quantity | fiscalyear | fiscalmonth | remarks |
+----+-------------------------------------+----------+------------+-------------+---------+
|  1 | 2003 Harleydavidson Eagle Drag Bike |      150 |       2020 |           1 | NULL    |
|  2 | 1969 Corvair Monza                  |      400 |       2020 |           1 | NULL    |
|  3 | 1970 Plymouth Hemi Cuda             |      200 |       2020 |           1 | NULL    |
+----+-------------------------------------+----------+------------+-------------+---------+
3 rows in set (0.00 sec)
```

Q4. Create a table SalesChanges with the following data

Column	Datatype	Constraint
id	int	Autoincrement Primary key
salesid	int	
beforequantity	int	
afterquantity	int	
change	date	default current_timestamp

Delete the existing rows in the Sales table

INSERT 3 rows in the columns product, quantity, fiscalYear, fiscalMonth the following

VALUES

1. ‘2001 Ferrari Enzo’,140, 2021,1

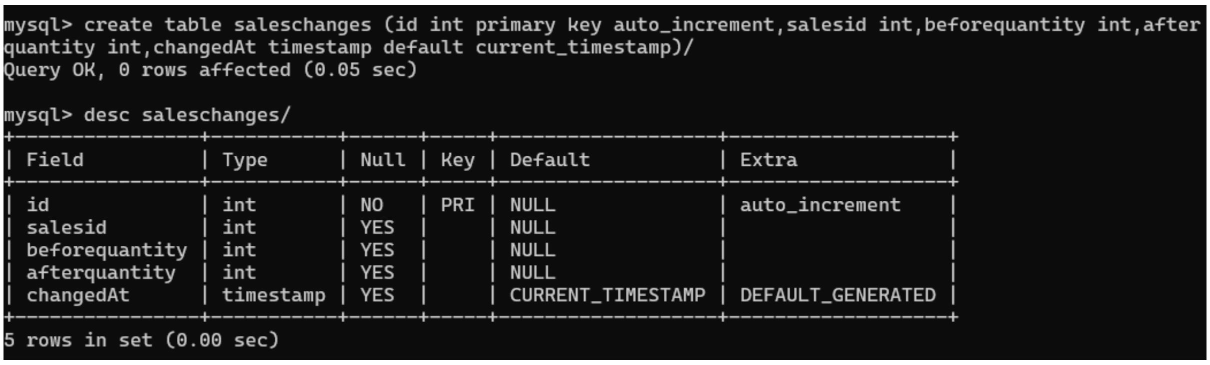
2. '1998 Chrysler Plymouth Prowler’, 110,2021,1

3. ‘1913 Ford Model T Speedster’, 120,2021,1

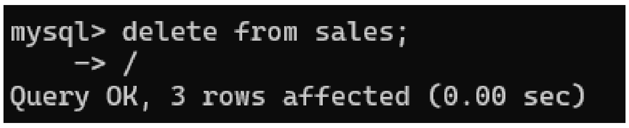
Create an after update trigger which does the following

When the value in the quantity column of sales table is updated to a new value then

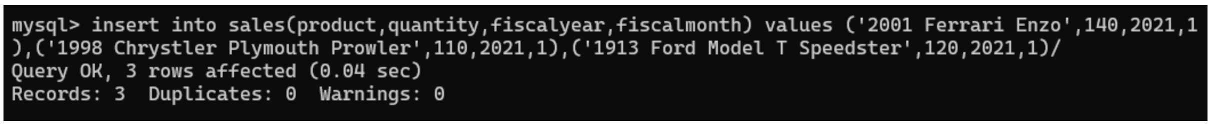
insert a new row to log the changes in the SalesChanges table otherwise do not insert.



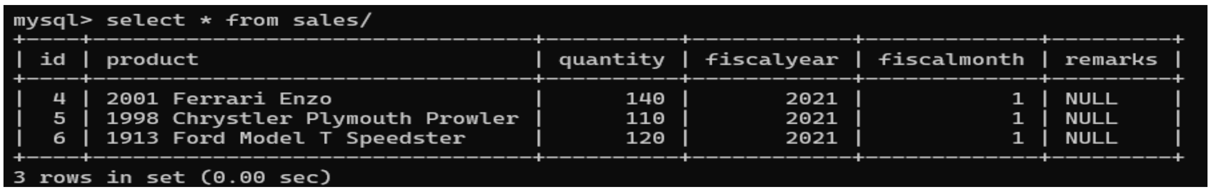
```
mysql> create table saleschanges (id int primary key auto_increment,salesid int,beforequantity int,after
quantity int,changedAt timestamp default current_timestamp)/
Query OK, 0 rows affected (0.05 sec)

mysql> desc saleschanges/
+----------------+-----------+------+-----+-------------------+-------------------+
| Field          | Type      | Null | Key | Default           | Extra             |
+----------------+-----------+------+-----+-------------------+-------------------+
| id             | int       | NO   | PRI | NULL              | auto_increment    |
| salesid        | int       | YES  |     | NULL              |                   |
| beforequantity | int       | YES  |     | NULL              |                   |
| afterquantity  | int       | YES  |     | NULL              |                   |
| changedAt      | timestamp | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+----------------+-----------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```



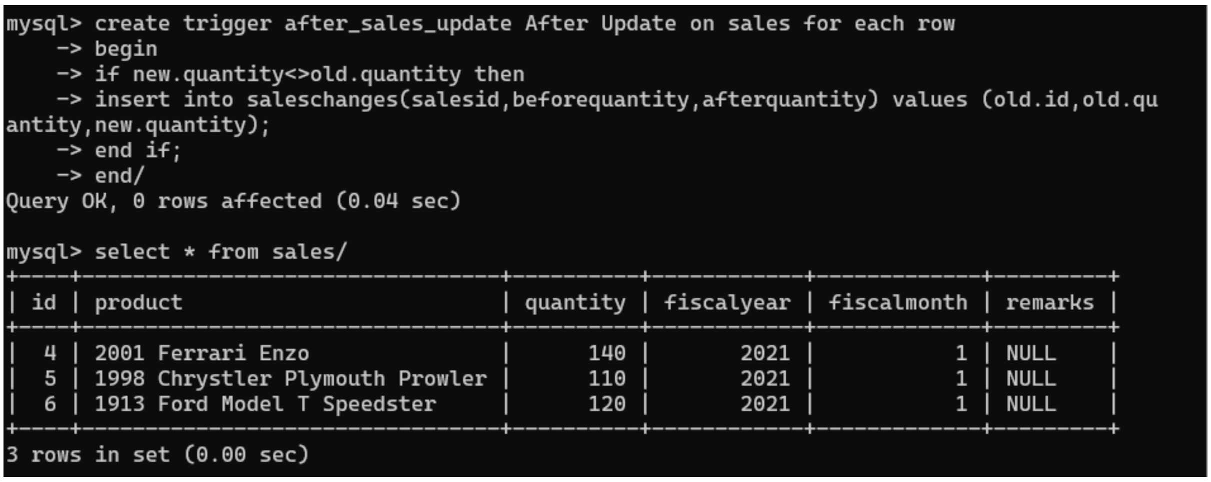
```
mysql> delete from sales;
    -> /
Query OK, 3 rows affected (0.00 sec)
```



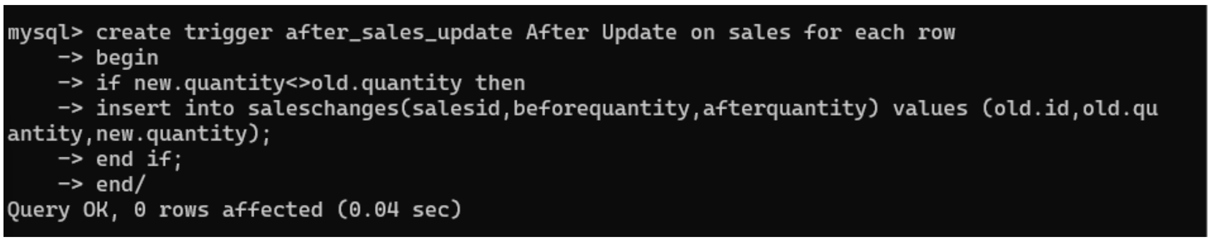
```
mysql> insert into sales(product,quantity,fiscalyear,fiscalmonth) values ('2001 Ferrari Enzo',140,2021,1
),('1998 Chrystler Plymouth Prowler',110,2021,1),('1913 Ford Model T Speedster',120,2021,1)/
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0
```



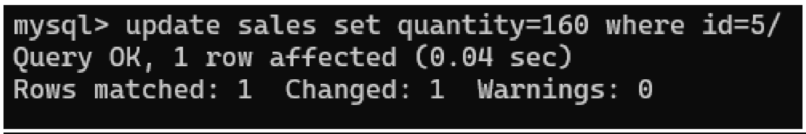
```
mysql> select * from sales/
+----+--------------------------------+----------+------------+-------------+---------+
| id | product                        | quantity | fiscalyear | fiscalmonth | remarks |
+----+--------------------------------+----------+------------+-------------+---------+
|  4 | 2001 Ferrari Enzo              |      140 |       2021 |           1 | NULL    |
|  5 | 1998 Chrystler Plymouth Prowler |      110 |       2021 |           1 | NULL    |
|  6 | 1913 Ford Model T Speedster    |      120 |       2021 |           1 | NULL    |
+----+--------------------------------+----------+------------+-------------+---------+
3 rows in set (0.00 sec)
```



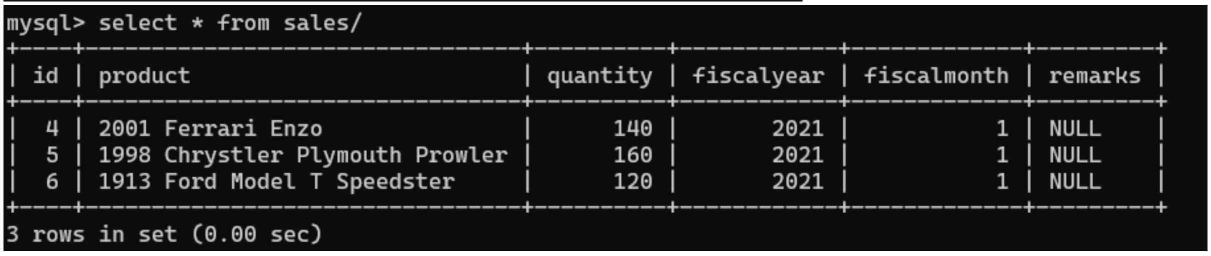
```
mysql> create trigger after_sales_update After Update on sales for each row
    -> begin
    -> if new.quantity<>old.quantity then
    -> insert into saleschanges(salesid,beforequantity,afterquantity) values (old.id,old.qu
antity,new.quantity);
    -> end if;
    -> end/
Query OK, 0 rows affected (0.04 sec)

mysql> select * from sales/
+----+--------------------------------+----------+------------+-------------+---------+
| id | product                        | quantity | fiscalyear | fiscalmonth | remarks |
+----+--------------------------------+----------+------------+-------------+---------+
|  4 | 2001 Ferrari Enzo              |      140 |       2021 |           1 | NULL    |
|  5 | 1998 Chrystler Plymouth Prowler |      110 |       2021 |           1 | NULL    |
|  6 | 1913 Ford Model T Speedster    |      120 |       2021 |           1 | NULL    |
+----+--------------------------------+----------+------------+-------------+---------+
3 rows in set (0.00 sec)
```



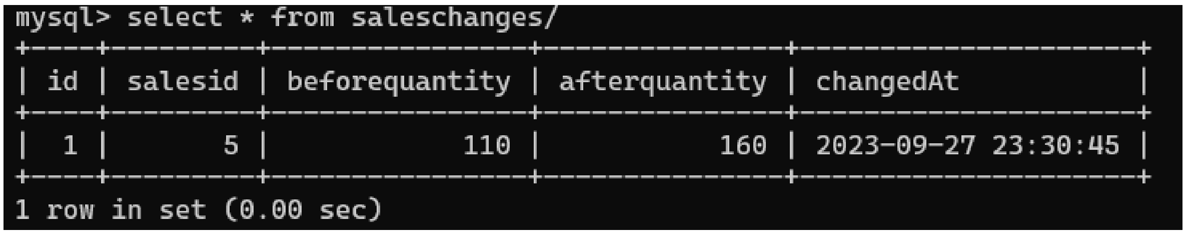
```
mysql> create trigger after_sales_update After Update on sales for each row
    -> begin
    -> if new.quantity<>old.quantity then
    -> insert into saleschanges(salesid,beforequantity,afterquantity) values (old.id,old.qu
antity,new.quantity);
    -> end if;
    -> end/
Query OK, 0 rows affected (0.04 sec)
```



```
mysql> update sales set quantity=160 where id=5/
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```



```
mysql> select * from sales/
+----+--------------------------------+----------+------------+-------------+---------+
| id | product                        | quantity | fiscalyear | fiscalmonth | remarks |
+----+--------------------------------+----------+------------+-------------+---------+
|  4 | 2001 Ferrari Enzo              |      140 |       2021 |           1 | NULL    |
|  5 | 1998 Chrystler Plymouth Prowler |      160 |       2021 |           1 | NULL    |
|  6 | 1913 Ford Model T Speedster    |      120 |       2021 |           1 | NULL    |
+----+--------------------------------+----------+------------+-------------+---------+
3 rows in set (0.00 sec)
```



```
mysql> select * from saleschanges/
+----+---------+----------------+---------------+---------------------+
| id | salesid | beforequantity | afterquantity | changedAt           |
+----+---------+----------------+---------------+---------------------+
|  1 |       5 |            110 |           160 | 2023-09-27 23:30:45 |
+----+---------+----------------+---------------+---------------------+
1 row in set (0.00 sec)
```

Q5. Create a table Salaries with the following data

Column	Datatype	Constraint
Employeenumber	int	Primary Key
validFrom	Date	Not Null
amount	Decimal(12,2)	Not Null Default 0

INSERT 3 rows in the table the following VALUES

1. 1002,’2000-01-01’,50000

2. 1056,’39;2000-01-01’,60000

3. 1076,'2000-01-01',70000

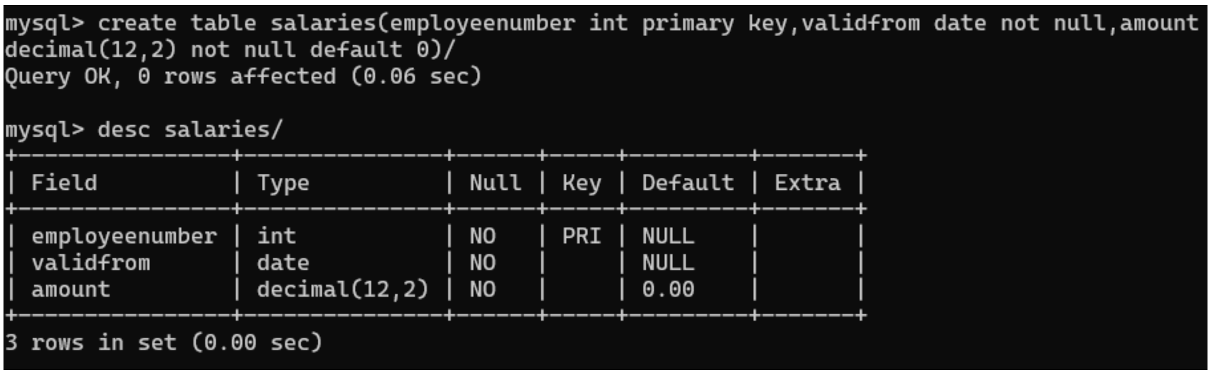
Create a table SalaryArchives with the following data

Column	Datatype	Constraint
id	int	Primary Key autoincrement
employeenumber	int	
validFrom	Date	Not Null
amount	Decimal(12,2)	Not Null Default 0
Delete	date	TimestampDefault now()

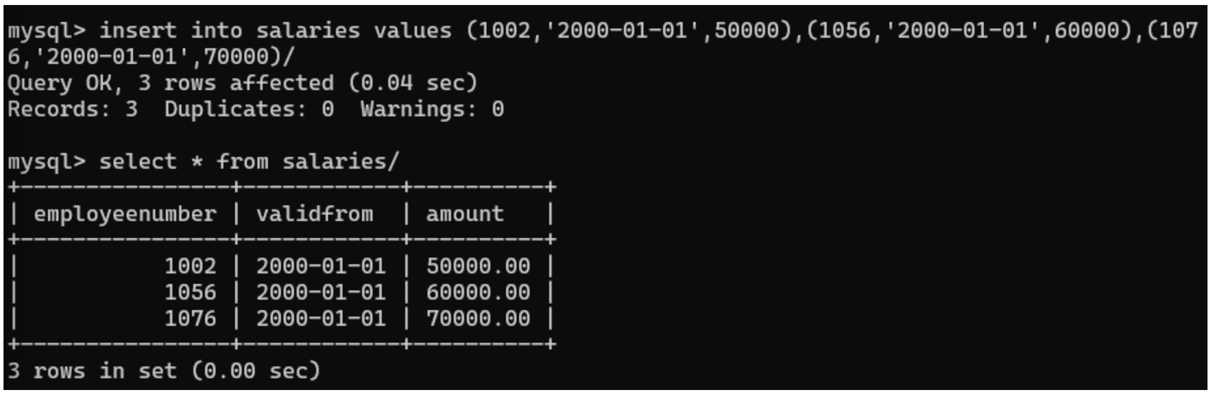
Create a BEFORE DELETE trigger that inserts a new row into the SalaryArchives table

before a row from the Salaries table is deleted.

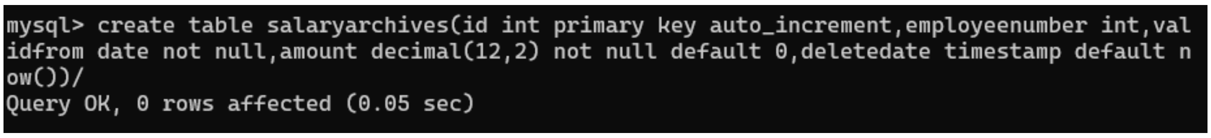
Test the trigger by deleting the rows in the salaries table.



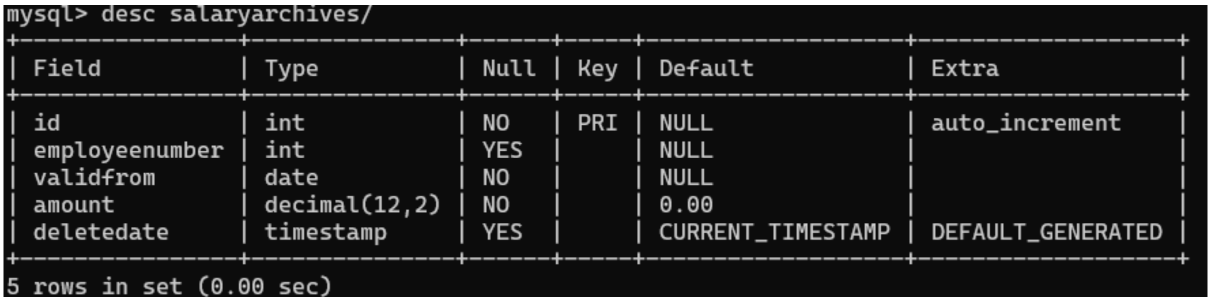
```
mysql> create table salaries(employeenumber int primary key,validfrom date not null,amount
decimal(12,2) not null default 0)/
Query OK, 0 rows affected (0.06 sec)

mysql> desc salaries/
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| employeenumber | int           | NO   | PRI | NULL    |       |
| validfrom      | date          | NO   |     | NULL    |       |
| amount         | decimal(12,2) | NO   |     | 0.00    |       |
+----------------+---------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```



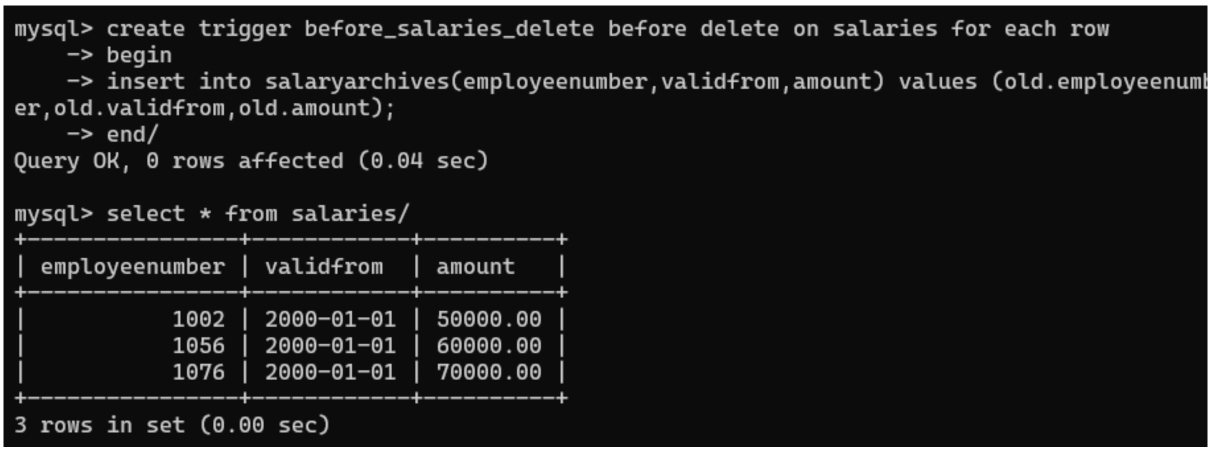
```
mysql> insert into salaries values (1002,'2000-01-01',50000),(1056,'2000-01-01',60000),(107
6,'2000-01-01',70000)/
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from salaries/
+----------------+------------+----------+
| employeenumber | validfrom  | amount   |
+----------------+------------+----------+
|           1002 | 2000-01-01 | 50000.00 |
|           1056 | 2000-01-01 | 60000.00 |
|           1076 | 2000-01-01 | 70000.00 |
+----------------+------------+----------+
3 rows in set (0.00 sec)
```



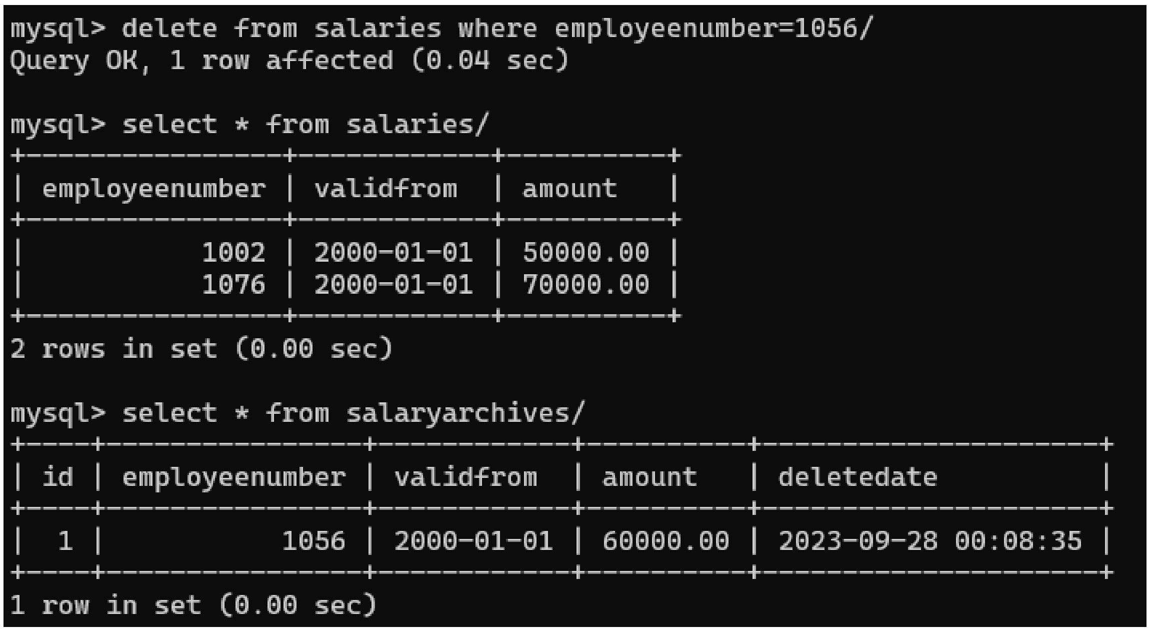
```
mysql> create table salaryarchives(id int primary key auto_increment,employeenumber int,val
idfrom date not null,amount decimal(12,2) not null default 0,deletedate timestamp default n
ow())/
Query OK, 0 rows affected (0.05 sec)
```



```
mysql> desc salaryarchives/
+----------------+---------------+------+-----+-------------------+-------------------+
| Field          | Type          | Null | Key | Default           | Extra             |
+----------------+---------------+------+-----+-------------------+-------------------+
| id             | int           | NO   | PRI | NULL              | auto_increment    |
| employeenumber | int           | YES  |     | NULL              |                   |
| validfrom      | date          | NO   |     | NULL              |                   |
| amount         | decimal(12,2) | NO   |     | 0.00              |                   |
| deletedate     | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+----------------+---------------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```



```
mysql> create trigger before_salaries_delete before delete on salaries for each row
    -> begin
    -> insert into salaryarchives(employeenumber,validfrom,amount) values (old.employeenumb
er,old.validfrom,old.amount);
    -> end/
Query OK, 0 rows affected (0.04 sec)

mysql> select * from salaries/
+----------------+------------+----------+
| employeenumber | validfrom  | amount   |
+----------------+------------+----------+
|           1002 | 2000-01-01 | 50000.00 |
|           1056 | 2000-01-01 | 60000.00 |
|           1076 | 2000-01-01 | 70000.00 |
+----------------+------------+----------+
3 rows in set (0.00 sec)
```



```
mysql> delete from salaries where employeenumber=1056/
Query OK, 1 row affected (0.04 sec)

mysql> select * from salaries/
+----------------+------------+----------+
| employeenumber | validfrom  | amount   |
+----------------+------------+----------+
|           1002 | 2000-01-01 | 50000.00 |
|           1076 | 2000-01-01 | 70000.00 |
+----------------+------------+----------+
2 rows in set (0.00 sec)

mysql> select * from salaryarchives/
+----+----------------+------------+----------+---------------------+
| id | employeenumber | validfrom  | amount   | deletedate          |
+----+----------------+------------+----------+---------------------+
|  1 |           1056 | 2000-01-01 | 60000.00 | 2023-09-28 00:08:35 |
+----+----------------+------------+----------+---------------------+
1 row in set (0.00 sec)
```

Q6. Drop the table salaries .Create a table Salaries with the following data

Column	Datatype	Constraint
employeenumber	int	Primary Key
salary	Decimal(12,2)	Not Null Default 0

INSERT 3 rows in the table the following VALUES

1. 1002,5000
2. 1056,,7000
3. 1076,8000

Create a table SalaryBudgets with the following data

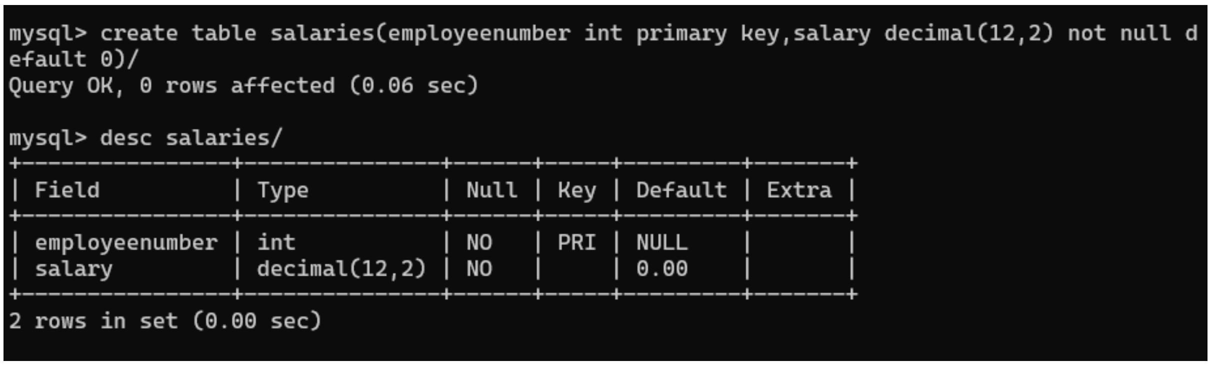
Column	Datatype	Constraint
total	Decimal(15,2)	Not Null

Insert a row into the SalaryBudgets table which is the sum of the values in the salary column of the Salaries table

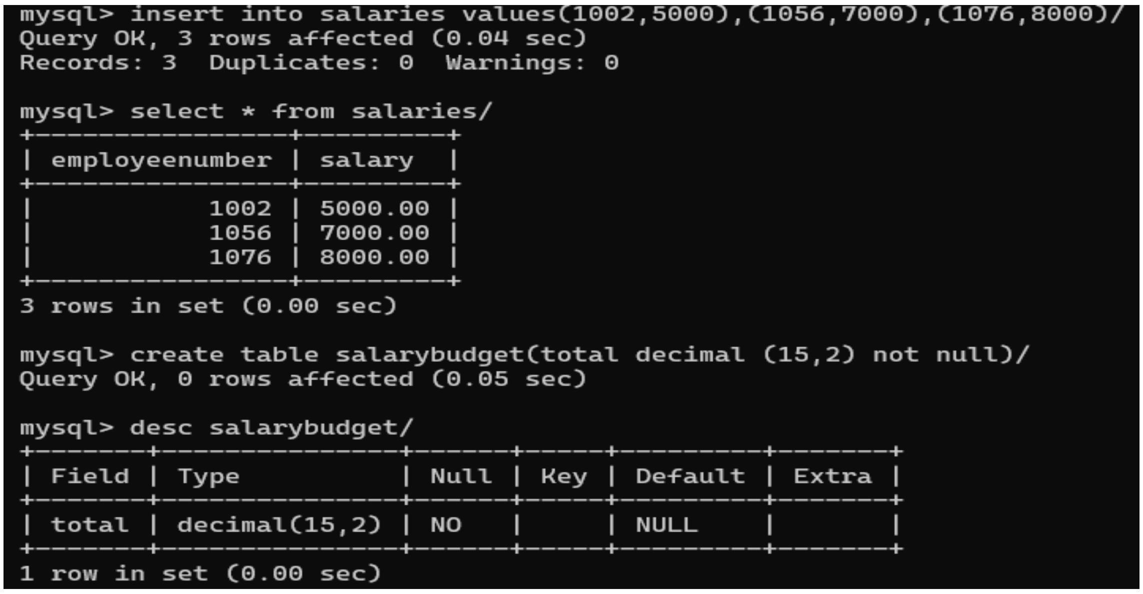
Create an AFTER DELETE trigger updates the total salary in the SalaryBudgets table after a row is deleted from the Salaries table (totalsalary should be updated by subtracting the salary of the row that is deleted from totalsalary column)

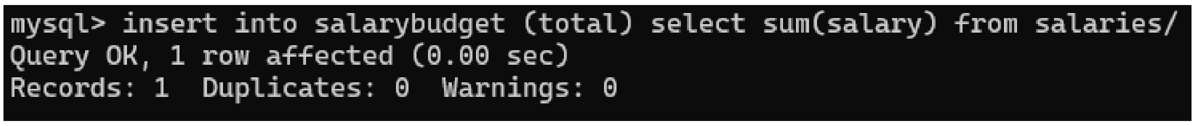
Test the trigger by deleting the rows from the salaries table

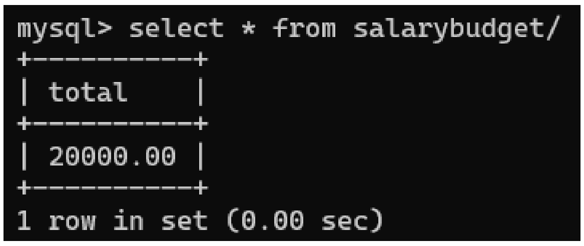
Salaries table:



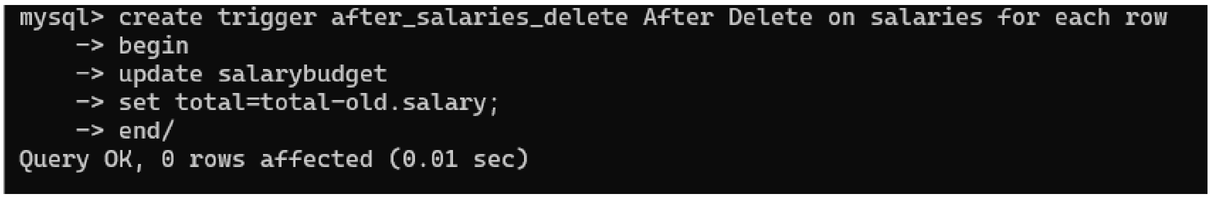
```
mysql> create table salaries(employeenumber int primary key,salary decimal(12,2) not null d
efault 0)/
Query OK, 0 rows affected (0.06 sec)

mysql> desc salaries/
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| employeenumber | int           | NO   | PRI | NULL    |       |
| salary         | decimal(12,2) | NO   |     | 0.00    |       |
+----------------+---------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```


```
mysql> insert into salaries values(1002,5000),(1056,7000),(1076,8000)/
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from salaries/
+----------------+---------+
| employeenumber | salary  |
+----------------+---------+
|           1002 | 5000.00 |
|           1056 | 7000.00 |
|           1076 | 8000.00 |
+----------------+---------+
3 rows in set (0.00 sec)

mysql> create table salarybudget(total decimal (15,2) not null)/
Query OK, 0 rows affected (0.05 sec)

mysql> desc salarybudget/
+-------+---------------+------+-----+---------+-------+
| Field | Type          | Null | Key | Default | Extra |
+-------+---------------+------+-----+---------+-------+
| total | decimal(15,2) | NO   |     | NULL    |       |
+-------+---------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

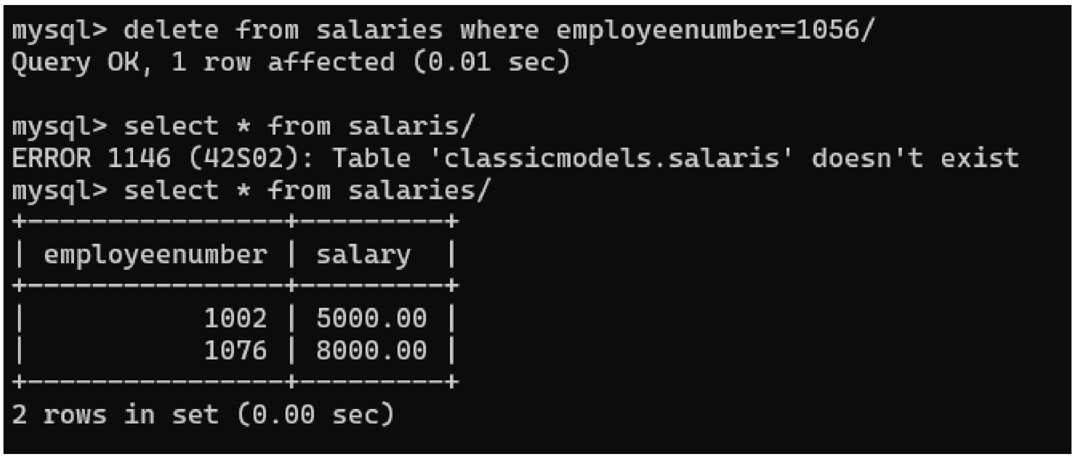

```
mysql> insert into salarybudget (total) select sum(salary) from salaries/
Query OK, 1 row affected (0.00 sec)
Records: 1  Duplicates: 0  Warnings: 0
```


```
mysql> select * from salarybudget/
+----------+
| total    |
+----------+
| 20000.00 |
+----------+
1 row in set (0.00 sec)
```

Create TRIGGER:

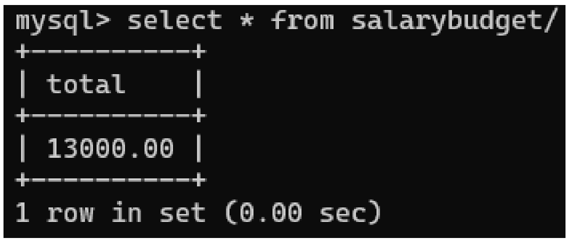

```
mysql> create trigger after_salaries_delete After Delete on salaries for each row
    -> begin
    -> update salarybudget
    -> set total=total-old.salary;
    -> end/
Query OK, 0 rows affected (0.01 sec)
```

Delete from salaries:



```
mysql> delete from salaries where employeenumber=1056/
Query OK, 1 row affected (0.01 sec)

mysql> select * from salaris/
ERROR 1146 (42S02): Table 'classicmodels.salaris' doesn't exist
mysql> select * from salaries/
+----------------+---------+
| employeenumber | salary  |
+----------------+---------+
|           1002 | 5000.00 |
|           1076 | 8000.00 |
+----------------+---------+
2 rows in set (0.00 sec)
```

Total salary :



```
mysql> select * from salarybudget/
+----------+
| total    |
+----------+
| 13000.00 |
+----------+
1 row in set (0.00 sec)
```