SQL Lab-5

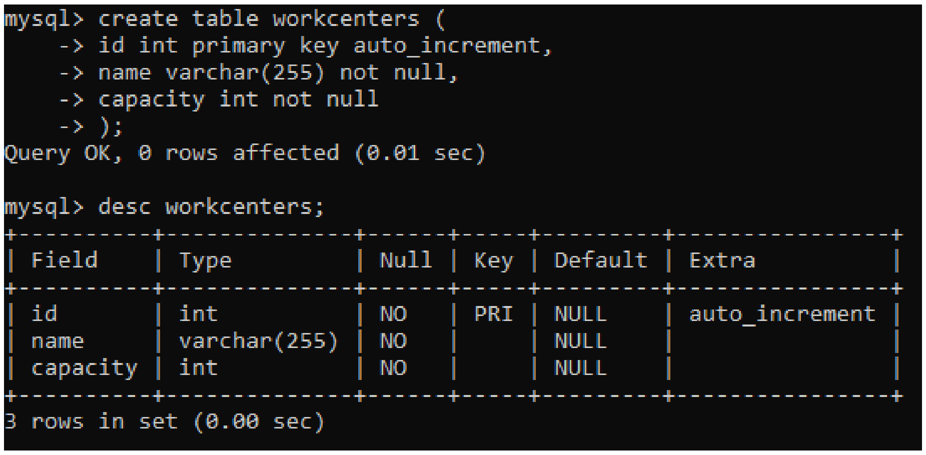
1. Create a table Workcenters with the following data

Column	Datatype	Constraint
id	int	Primary key,autoincrement
name	Varchar(255)	Not null
capacity	int	Not null

Create a table WorkcenterStats with the following data

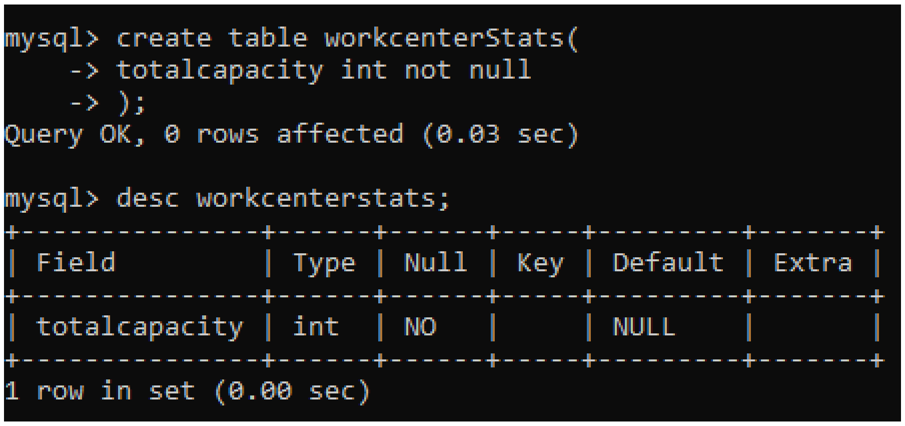
Column	Datatype	Constraint
totalcapacity	int	Not Null

Write a trigger which updates the total capacity in the WorkCenterStats table before a new work center is inserted into the WorkCenter table based on the following condition:
If the table WorkCenterStats has a row, the trigger adds the new capacity to the totalCapacity column.
Otherwise, it inserts a new row into the WorkCenterStats table with the new capacity in the totalcapacity column.
Test the trigger by inserting new rows into the WorkCenters table
Ans:
Tables creation-
create table workcenters (
id int primary key auto_increment,
name varchar(255) not null,
capacity int not null
);



```
mysql> create table workcenters (
    -> id int primary key auto_increment,
    -> name varchar(255) not null,
    -> capacity int not null
    -> );
Query OK, 0 rows affected (0.01 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)
```

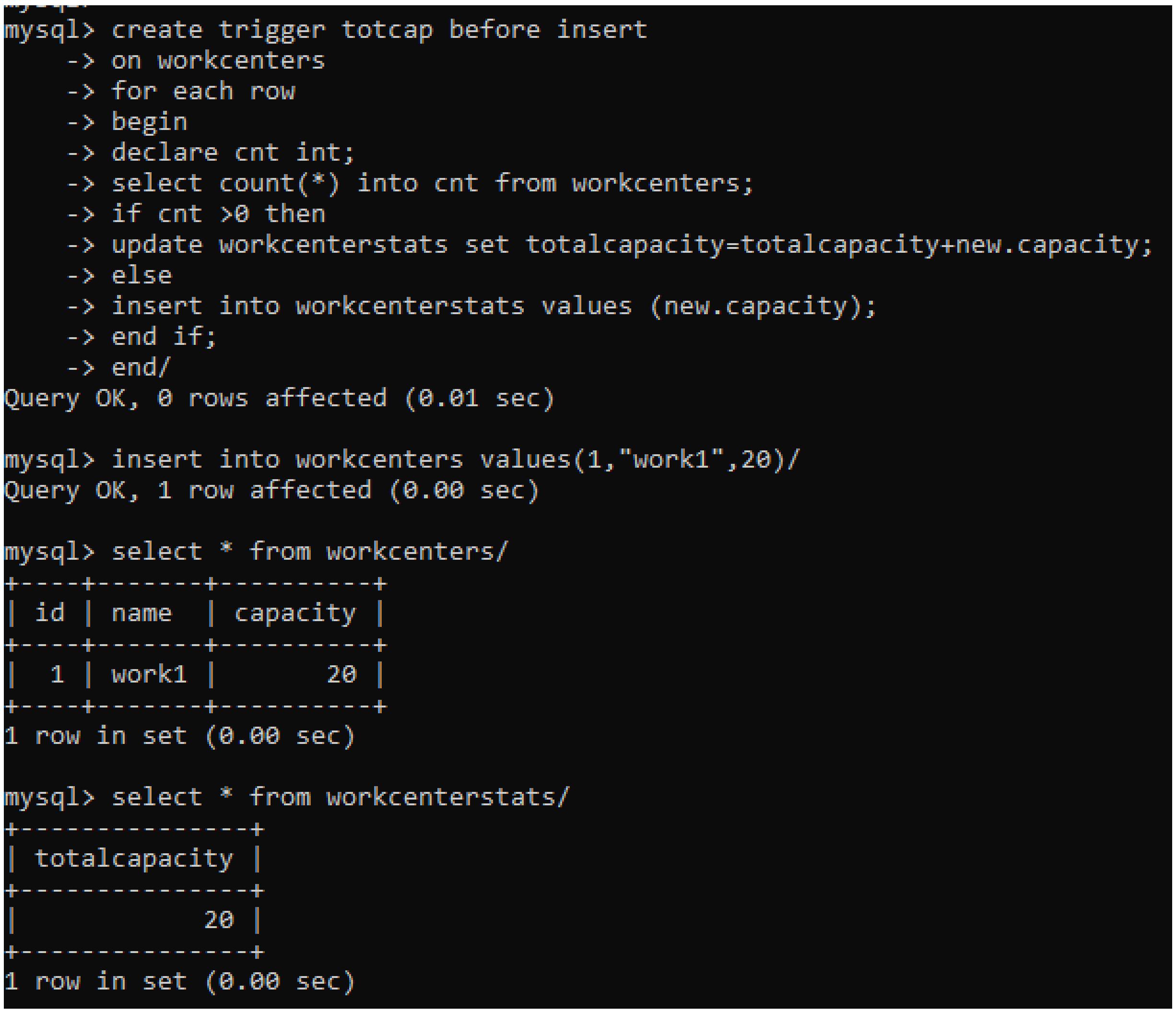
create table workcenterStats(
totalcapacity int not null
);



```
mysql> create table workcenterStats(
    -> totalcapacity int not null
    -> );
Query OK, 0 rows affected (0.03 sec)

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

Trigger-
create trigger totcap before insert
on workcenters
for each row
begin
declare cnt int;
select count(*) into cnt from workcenters;
if cnt >0 then
update workcenterstats set totalcapacity=totalcapacity+new.capacity;
else
insert into workcenterstats values (new.capacity);
end if;
end/

insert-
insert into workcenters values(1,”work1”,20);



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

mysql> insert into workcenters values(1,"work1",20)/
Query OK, 1 row affected (0.00 sec)

mysql> select * from workcenters/
+----+-------+----------+
| id | name  | capacity |
+----+-------+----------+
|  1 | work1 |       20 |
+----+-------+----------+
1 row in set (0.00 sec)

mysql> select * from workcenterstats/
+---------------+
| totalcapacity |
+---------------+
|            20 |
+---------------+
1 row in set (0.00 sec)
```

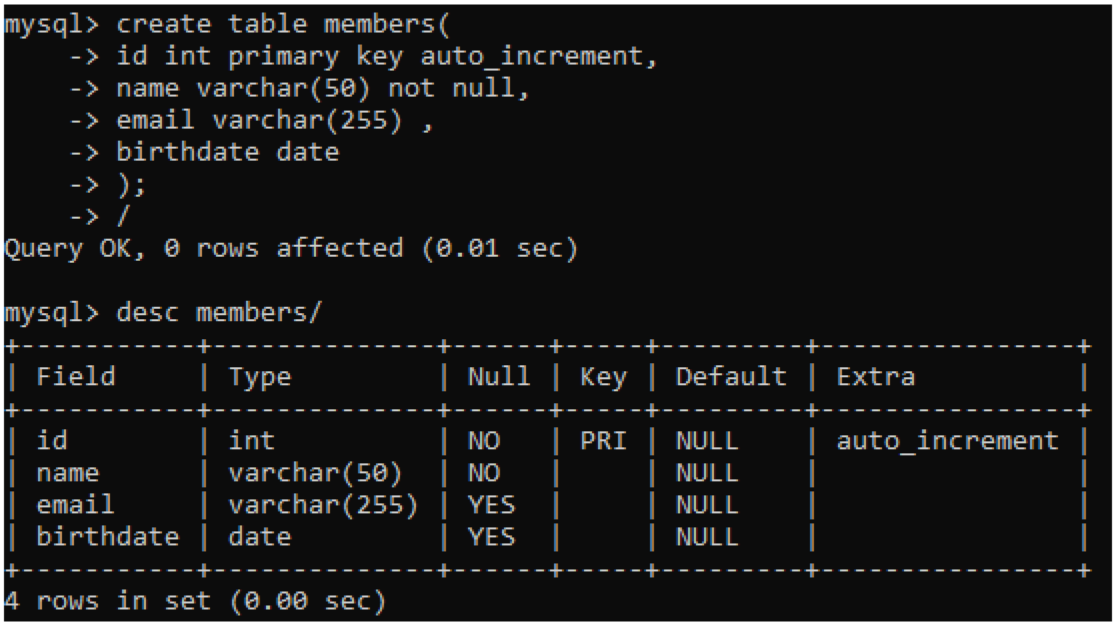
2. Create a table Members with the following data

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

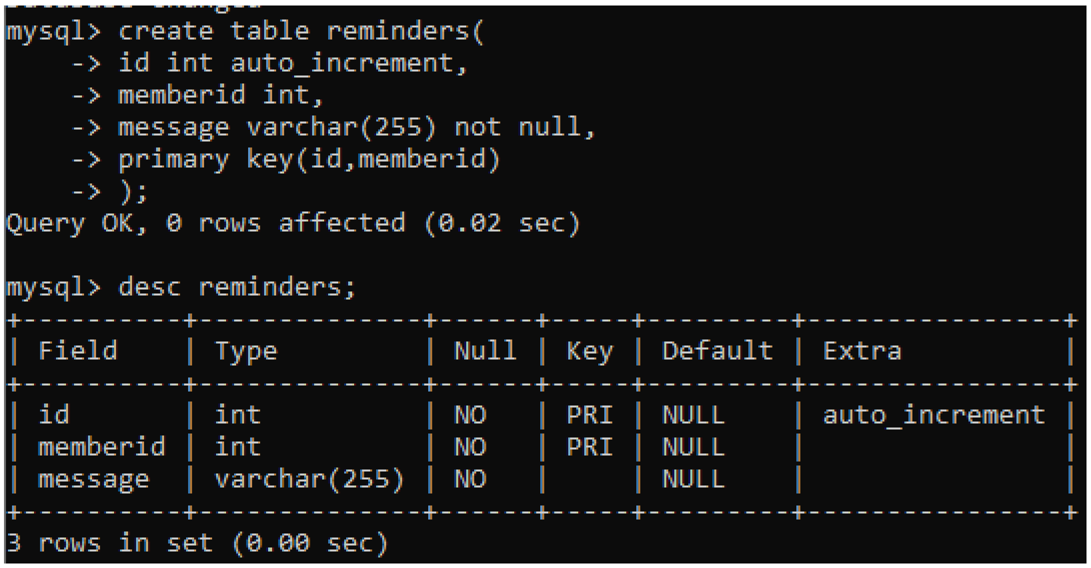
Column	Datatype	Constraint
id	int	Autoincrement Primary key
memberId	int	Primary Key
Message	Varchar(255)	Not Null

Create an AFTER INSERT trigger that inserts a reminder into the reminders table if the birth date of the member is NULL..
Ans:
Tables creation-
create table members(
id int primary key auto_increment,
name varchar(50) not null,
email varchar(255) ,
birthdate date
);



```
mysql> create table members(
    -> id int primary key auto_increment,
    -> name varchar(50) not null,
    -> email varchar(255) ,
    -> birthdate date
    -> );
    -> /
Query OK, 0 rows affected (0.01 sec)

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)
```

create table reminders(
id int auto_increment,
memberid int,
message varchar(255) not null,
primary key(id,memberid)
);



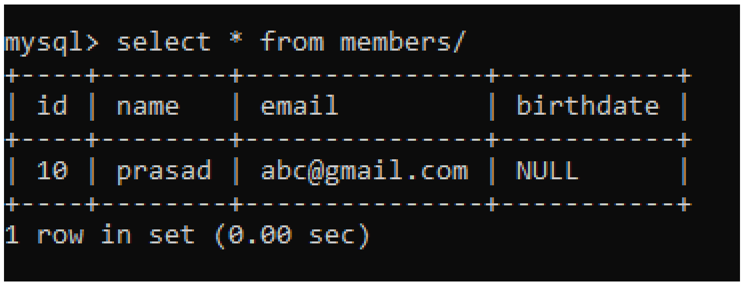
```
mysql> create table reminders(
    -> id int auto_increment,
    -> memberid int,
    -> message varchar(255) not null,
    -> primary key(id,memberid)
    -> );
Query OK, 0 rows affected (0.02 sec)

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

Trigger creation-

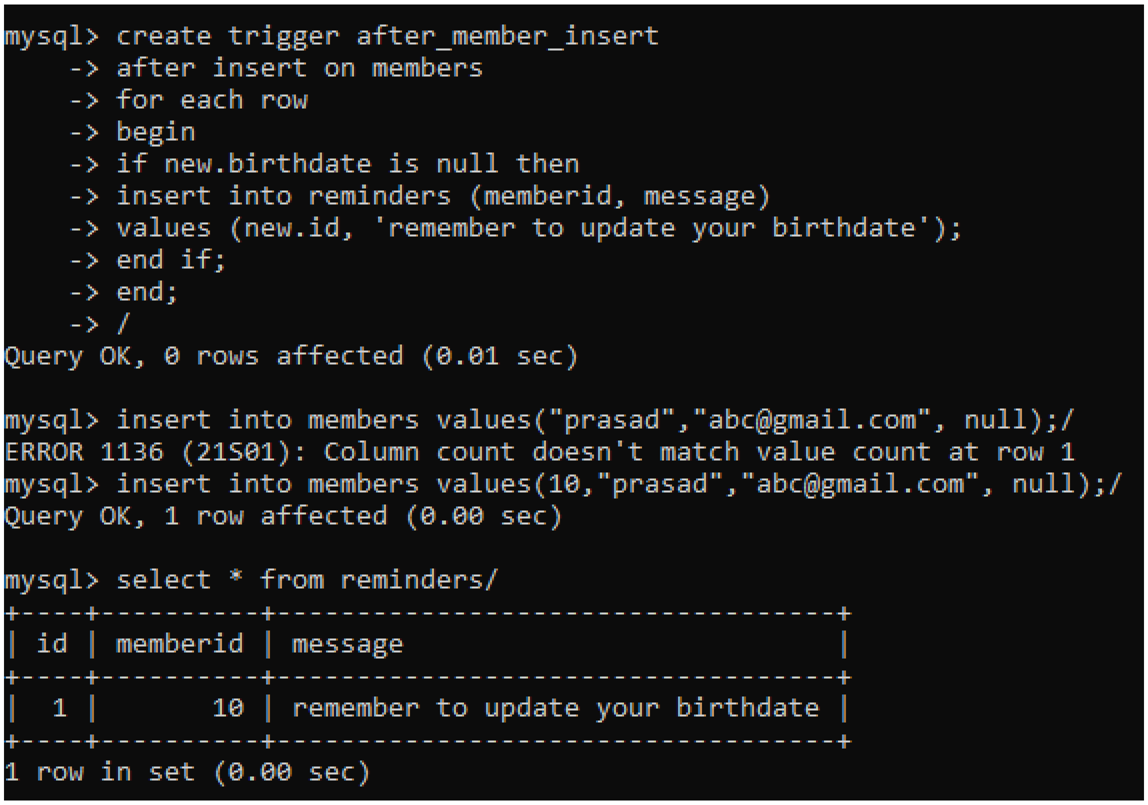
```
create trigger after_member_insert
after insert on members
for each row
begin
if new.birthdate is null then
insert into reminders (memberid, message)
values (new.id, 'remember to update your birthdate');
end if;
end/
```

insert-

insert into members values(10,“prasad”,”abc@gmail.com”, null);



```
mysql> select * from members/
+----+--------+---------------+-----------+
| id | name   | email         | birthdate |
+----+--------+---------------+-----------+
| 10 | prasad | abc@gmail.com | NULL      |
+----+--------+---------------+-----------+
1 row in set (0.00 sec)
```



```
mysql> create trigger after_member_insert
    -> after insert on members
    -> for each row
    -> begin
    -> if new.birthdate is null then
    -> insert into reminders (memberid, message)
    -> values (new.id, 'remember to update your birthdate');
    -> end if;
    -> end;
    -> /
Query OK, 0 rows affected (0.01 sec)

mysql> insert into members values("prasad","abc@gmail.com", null);/
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> insert into members values(10,"prasad","abc@gmail.com", null);/
Query OK, 1 row affected (0.00 sec)

mysql> select * from reminders/
+----+----------+-----------------------------------+
| id | memberid | message                           |
+----+----------+-----------------------------------+
|  1 |       10 | remember to update your birthdate |
+----+----------+-----------------------------------+
1 row in set (0.00 sec)
```

3. 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

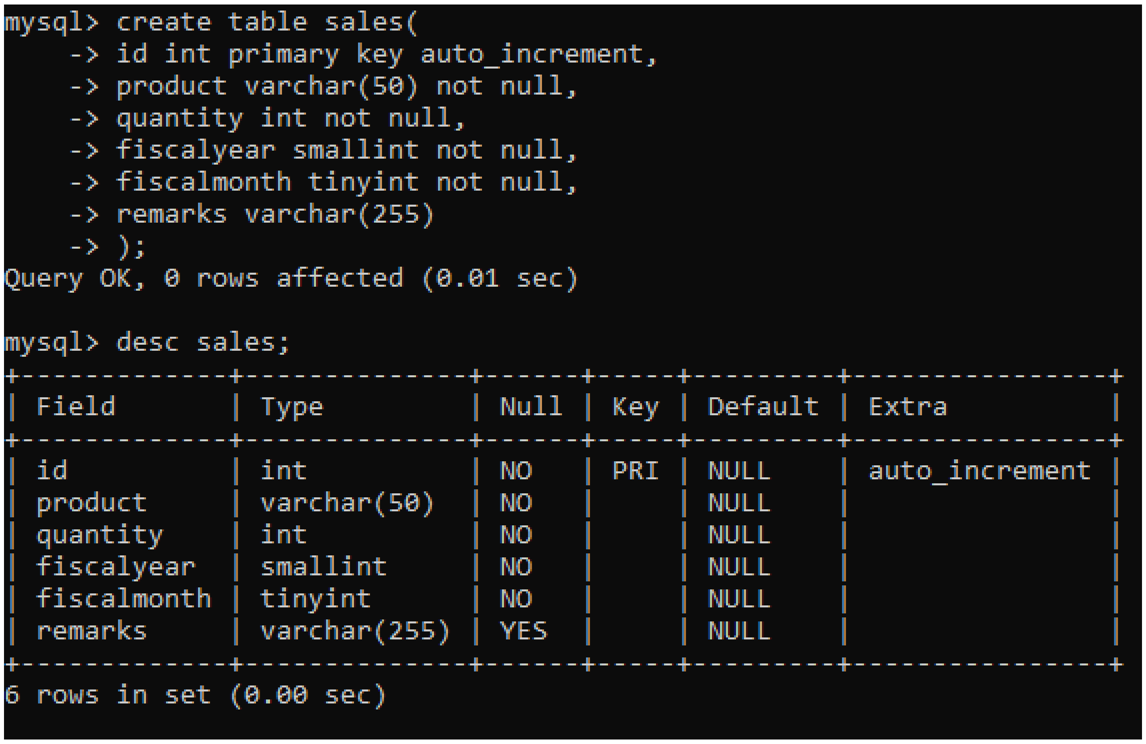
1. '2003 Harley-Davidson Eagle Drag Bike',120, 2020,1
2. '1969 Corvair Monza', 150,2020,1
3. '1970 Plymouth Hemi Cuda', 200,2020,1

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"

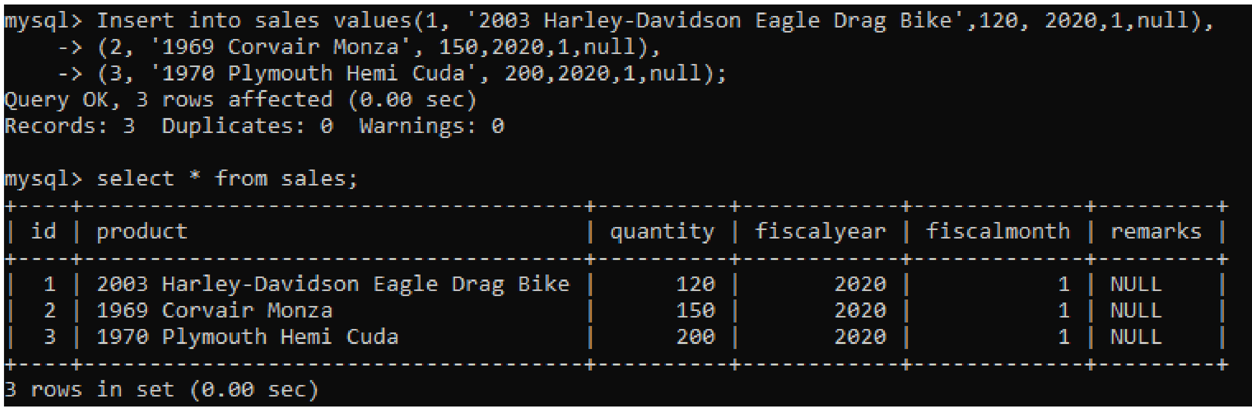
Update the row and check with different values

Ans:
Table creation-
create table sales(
id int primary key auto_increment,
product varchar(50) not null,
quantity int not null,
fiscalyear smallint not null,
fiscalmonth tinyint not null,
remarks varchar(255)
);



```
mysql> create table sales(
    -> id int primary key auto_increment,
    -> product varchar(50) not null,
    -> quantity int not null,
    -> fiscalyear smallint not null,
    -> fiscalmonth tinyint not null,
    -> remarks 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)
```

Insert-
Insert into sales values(1, '2003 Harley-Davidson Eagle Drag Bike',120, 2020,1,null),
(2, '1969 Corvair Monza', 150,2020,1,null),
(3, '1970 Plymouth Hemi Cuda', 200,2020,1,null);



```
mysql> Insert into sales values(1, '2003 Harley-Davidson Eagle Drag Bike',120, 2020,1,null),
    -> (2, '1969 Corvair Monza', 150,2020,1,null),
    -> (3, '1970 Plymouth Hemi Cuda', 200,2020,1,null);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from sales;
+----+--------------------------------------+----------+------------+-------------+---------+
| id | product                              | quantity | fiscalyear | fiscalmonth | remarks |
+----+--------------------------------------+----------+------------+-------------+---------+
|  1 | 2003 Harley-Davidson 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)
```

Trigger creation-

create trigger saleslimit before update

on sales

for each row

begin

if new.quantity>(old.quantity*3) then

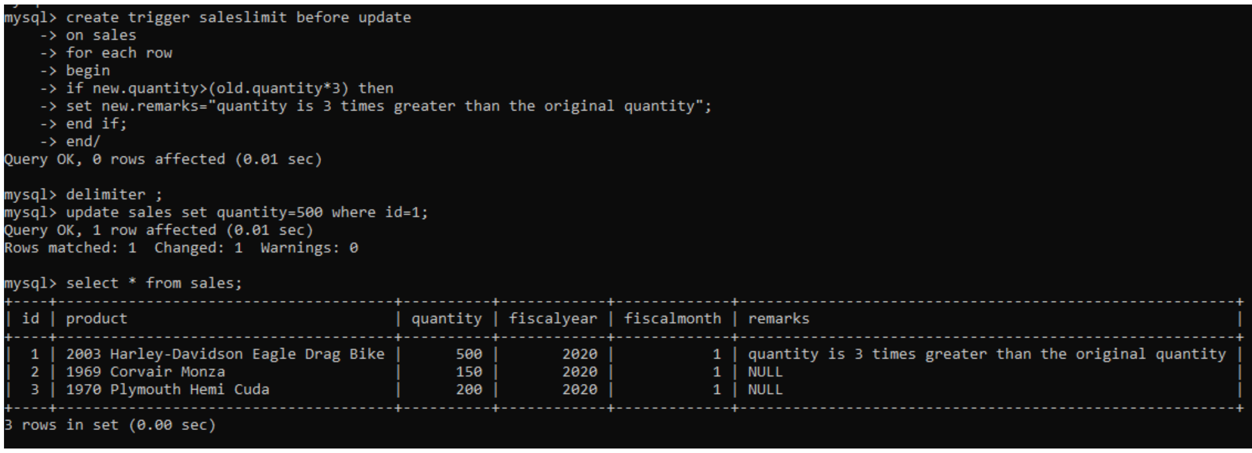
set new.remarks=”quantity is 3 times greater than the original quantity”;

end if;

end/

update –

update sales set quantity=500 where id=1;



```
mysql> create trigger saleslimit before update
    -> on sales
    -> for each row
    -> begin
    -> if new.quantity>(old.quantity*3) then
    -> set new.remarks="quantity is 3 times greater than the original quantity";
    -> end if;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> update sales set quantity=500 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 Harley-Davidson Eagle Drag Bike |      500 |       2020 |           1 | quantity is 3 times greater than the original quantity |
|  2 | 1969 Corvair Monza                   |      150 |       2020 |           1 | NULL                                                    |
|  3 | 1970 Plymouth Hemi Cuda              |      200 |       2020 |           1 | NULL                                                    |
+----+--------------------------------------+----------+------------+-------------+---------------------------------------------------------+
3 rows in set (0.00 sec)
```

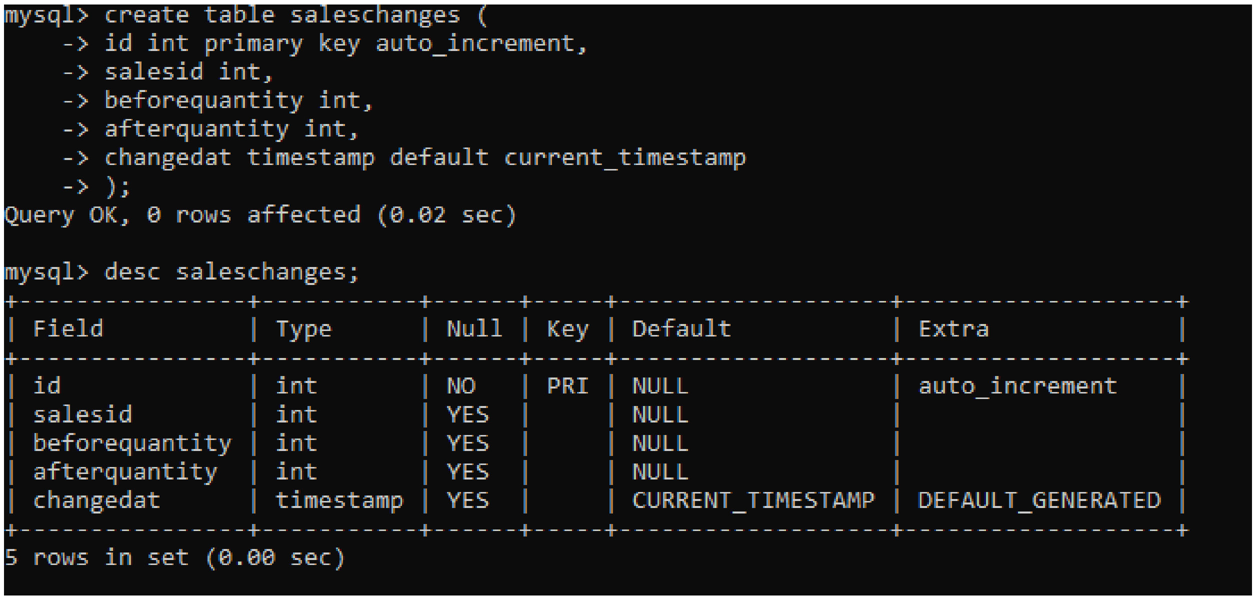
4. Create a table SalesChanges with the following data

Column	Datatype	Constraint
id	int	Autoincrement Primary key
salesid	int	
beforequantity	int	
afterquantity	int	
changedAt	Timestamp	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.
Ans:

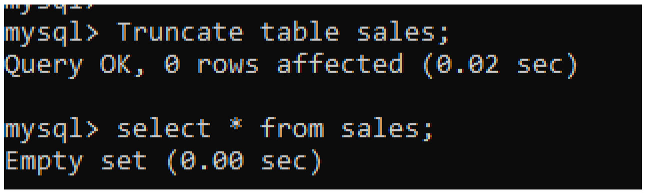
Table creation-

```
create table saleschanges (
id int primary key auto_increment,
salesid int,
beforequantity int,
afterquantity int,
changedat timestamp default current_timestamp
);
```


```
mysql> create table saleschanges (
    -> id int primary key auto_increment,
    -> salesid int,
    -> beforequantity int,
    -> afterquantity int,
    -> changedat timestamp default current_timestamp
    -> );
Query OK, 0 rows affected (0.02 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)
```

Deleting existing from sales-

Truncate table sales;

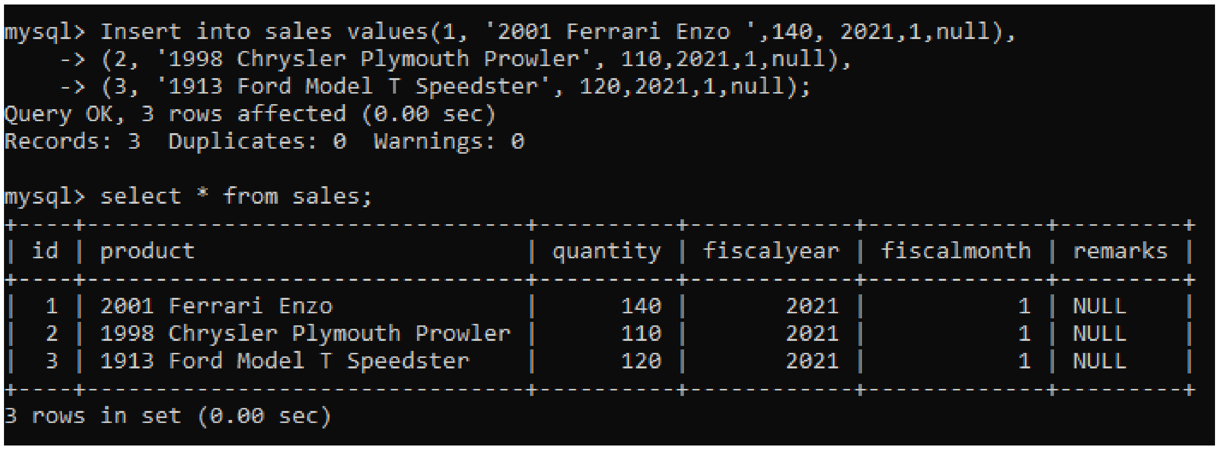

```
mysql> Truncate table sales;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from sales;
Empty set (0.00 sec)
```

Insert into sales table-

Insert into sales values(1, '2001 Ferrari Enzo ',140, 2021,1,null),

(2, '1998 Chrysler Plymouth Prowler', 110,2021,1,null),

(3, '1913 Ford Model T Speedster', 120,2021,1,null);

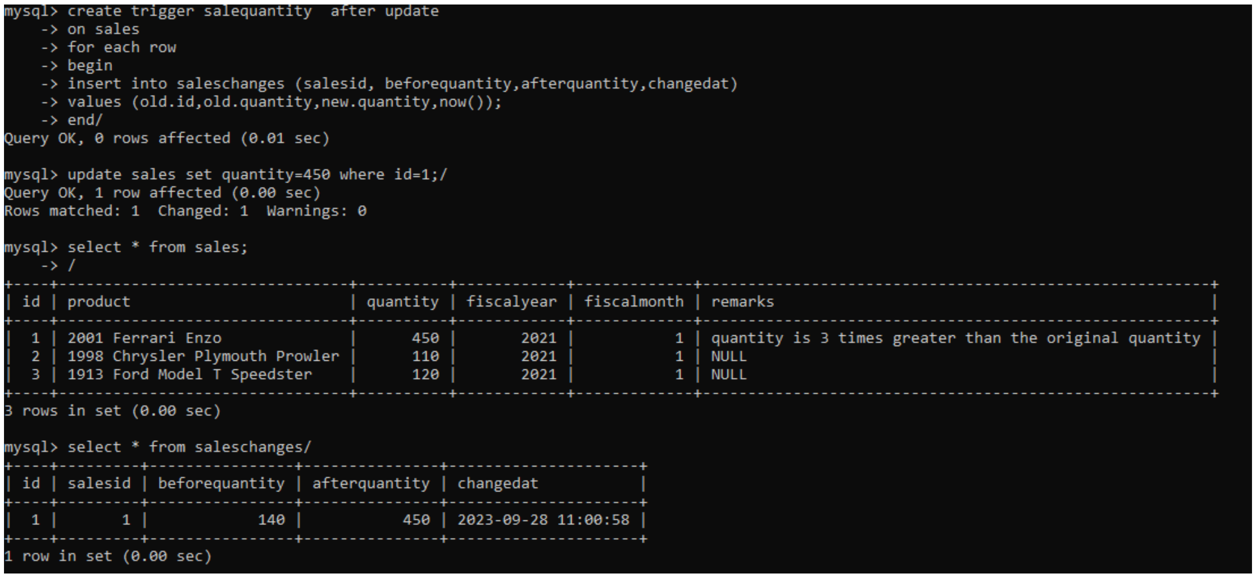

```
mysql> Insert into sales values(1, '2001 Ferrari Enzo ',140, 2021,1,null),
    -> (2, '1998 Chrysler Plymouth Prowler', 110,2021,1,null),
    -> (3, '1913 Ford Model T Speedster', 120,2021,1,null);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from sales;
+----+--------------------------------+----------+------------+-------------+---------+
| id | product                        | quantity | fiscalyear | fiscalmonth | remarks |
+----+--------------------------------+----------+------------+-------------+---------+
|  1 | 2001 Ferrari Enzo              |      140 |       2021 |           1 | NULL    |
|  2 | 1998 Chrysler Plymouth Prowler |      110 |       2021 |           1 | NULL    |
|  3 | 1913 Ford Model T Speedster    |      120 |       2021 |           1 | NULL    |
+----+--------------------------------+----------+------------+-------------+---------+
3 rows in set (0.00 sec)
```

Trigger creation-

```
create trigger salequantity  after update
on sales
for each row
begin
insert into saleschanges (salesid, beforequantity,afterquantity,changedat)
values (old.id,old.quantity,new.quantity,now());
end/
```

update-

```
update sales set quantity=450 where id=1;
```



```
mysql> create trigger salequantity  after update
    -> on sales
    -> for each row
    -> begin
    -> insert into saleschanges (salesid, beforequantity,afterquantity,changedat)
    -> values (old.id,old.quantity,new.quantity,now());
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> update sales set quantity=450 where id=1;/
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sales;
    -> /
+----+-------------------------------+----------+------------+-------------+------------------------------------------------------+
| id | product                       | quantity | fiscalyear | fiscalmonth | remarks                                              |
+----+-------------------------------+----------+------------+-------------+------------------------------------------------------+
|  1 | 2001 Ferrari Enzo             |      450 |       2021 |           1 | quantity is 3 times greater than the original quantity |
|  2 | 1998 Chrysler Plymouth Prowler |      110 |       2021 |           1 | NULL                                                 |
|  3 | 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 |       1 |            140 |           450 | 2023-09-28 11:00:58 |
+----+---------+----------------+---------------+---------------------+
1 row in set (0.00 sec)
```

5. 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,'2000-01-01',60000
3. 1076,'2000-01-01',70000

Create a table SalaryArchives with the following data

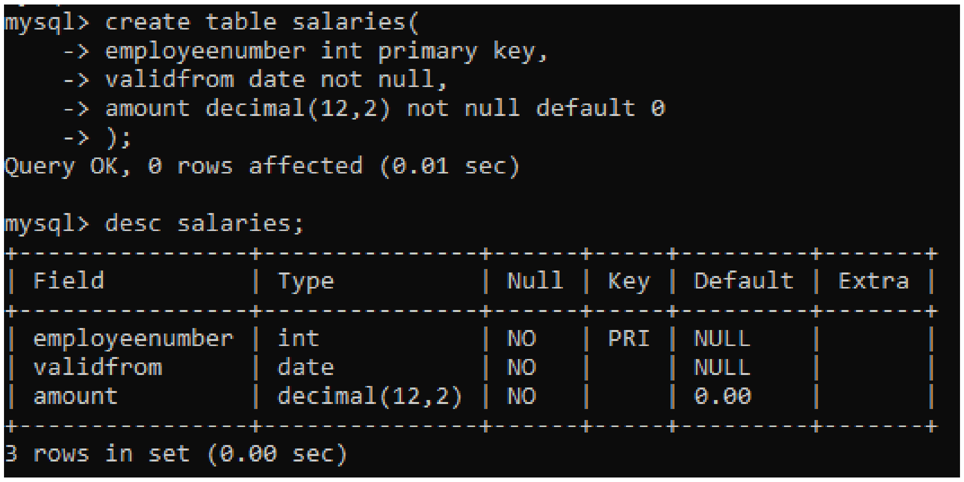
Column	Datatype	Constraint
id	int	Primary Key autoincrement
employeenumber	int	
validFrom	Date	Not Null
amount	Decimal(12,2)	Not Null Default 0
Deletedat	Timestamp	Default now()

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
Ans:

Table creation-

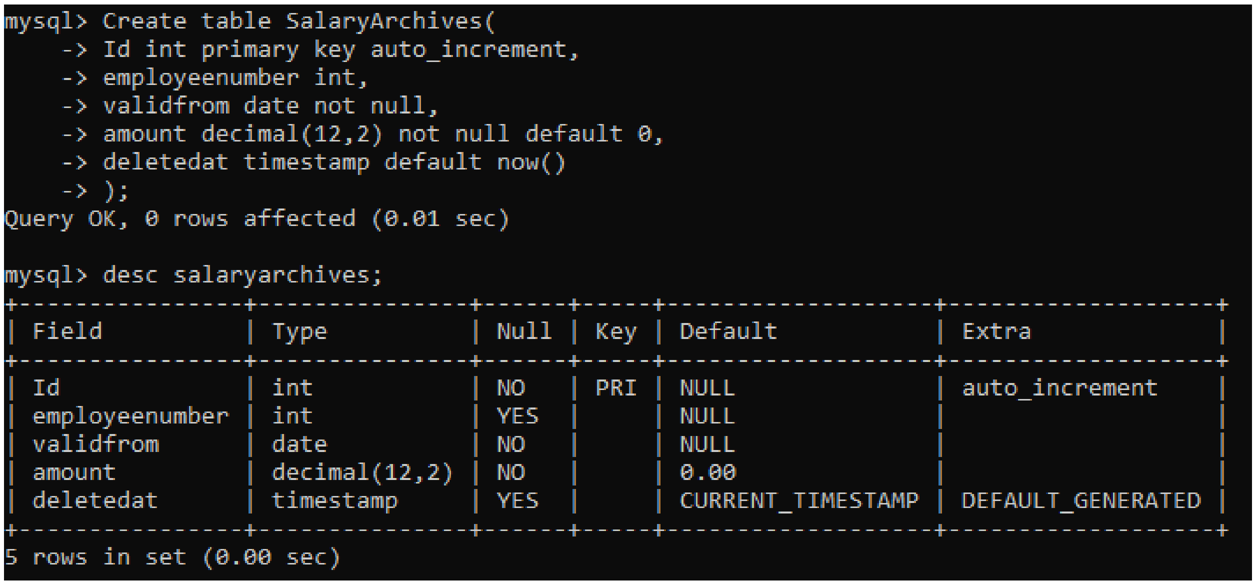
```
create table salaries(
employeenumber int primary key,
validfrom date not null,
amount decimal(12,2) not null default 0
);
```



```
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 (0.01 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)
```

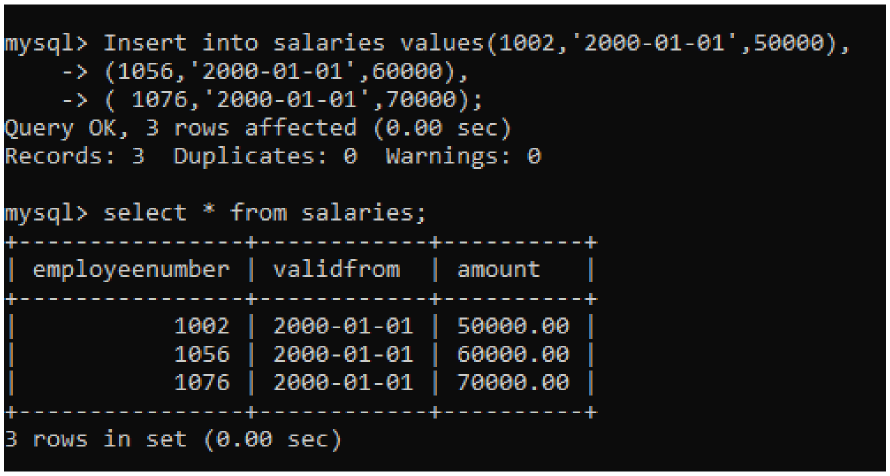
Create table SalaryArchives(

Id int primary key auto_increment,
employeenumber int,
validfrom date not null,
amount decimal(12,2) not null default 0,
deletedat timestamp default now()
);



```
mysql> Create table SalaryArchives(
    -> Id int primary key auto_increment,
    -> employeenumber int,
    -> validfrom date not null,
    -> amount decimal(12,2) not null default 0,
    -> deletedat timestamp default now()
    -> );
Query OK, 0 rows affected (0.01 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              |                   |
| deletedat      | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+----------------+---------------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```

Insert into salaries-
Insert into salaries values(1002,'2000-01-01',50000),
(1056,'2000-01-01',60000),
(1076,'2000-01-01',70000);

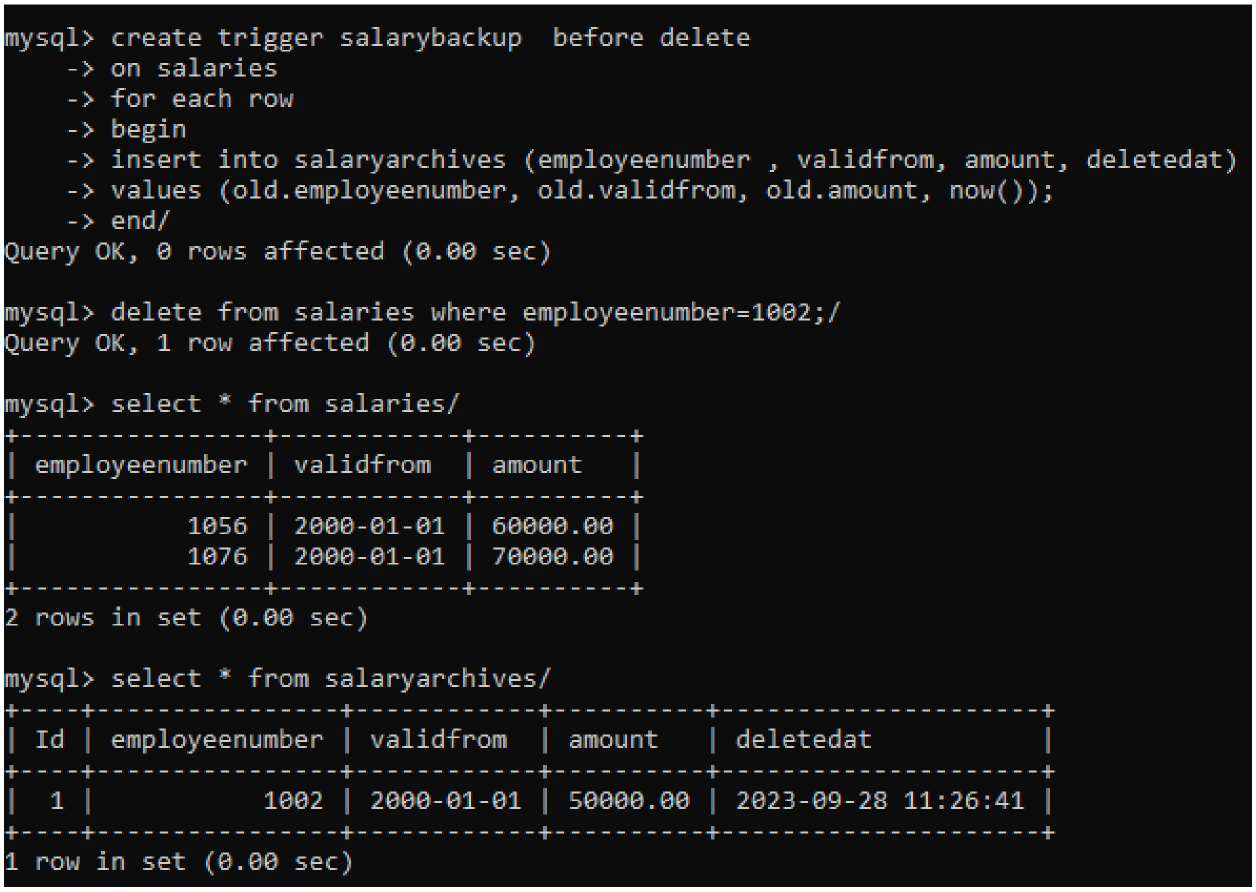


```
mysql> Insert into salaries values(1002,'2000-01-01',50000),
    -> (1056,'2000-01-01',60000),
    -> ( 1076,'2000-01-01',70000);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

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)
```

Trigger creation-

create trigger salarybackup before delete
on salaries
for each row
begin
insert into salaryarchives (employeenumber , validfrom, amount, deletedat)
values (old.employeenumber, old.validfrom, old.amount, now());
end/

delete-
delete from salaries where employeenumber=1002;/



```
mysql> create trigger salarybackup  before delete
    -> on salaries
    -> for each row
    -> begin
    -> insert into salaryarchives (employeenumber , validfrom, amount, deletedat)
    -> values (old.employeenumber, old.validfrom, old.amount, now());
    -> end/
Query OK, 0 rows affected (0.00 sec)

mysql> delete from salaries where employeenumber=1002;/
Query OK, 1 row affected (0.00 sec)

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

mysql> select * from salaryarchives/
+----+----------------+------------+----------+---------------------+
| Id | employeenumber | validfrom  | amount   | deletedat           |
+----+----------------+------------+----------+---------------------+
|  1 |           1002 | 2000-01-01 | 50000.00 | 2023-09-28 11:26:41 |
+----+----------------+------------+----------+---------------------+
1 row in set (0.00 sec)
```

6. 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

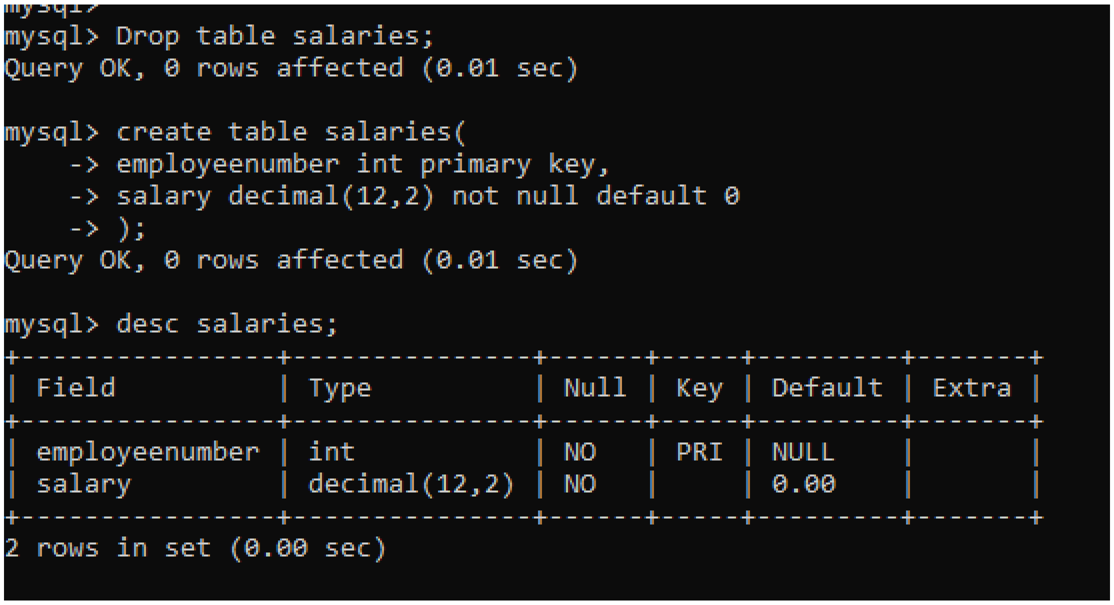
Column	Datatype	Constraint
total	Decimal(15,2)	Not Null

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)

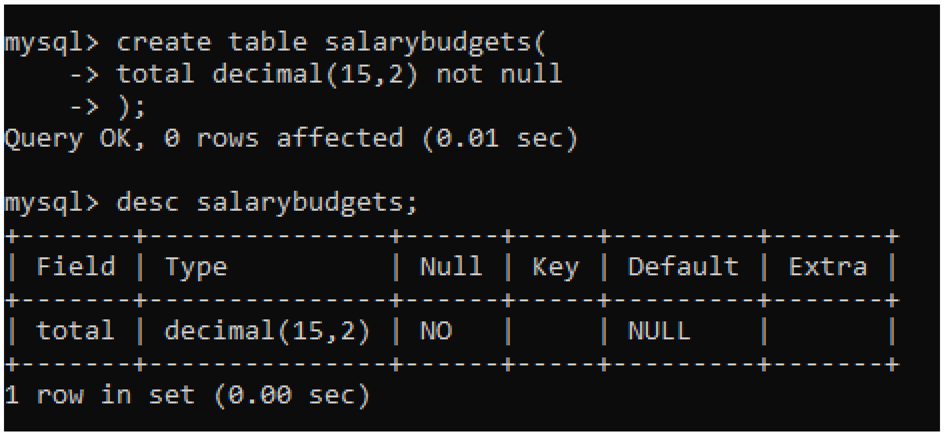
Test the trigger by deleting the rows from the salaries table

Ans:
Drop salaries-
```
Drop table salaries;
```

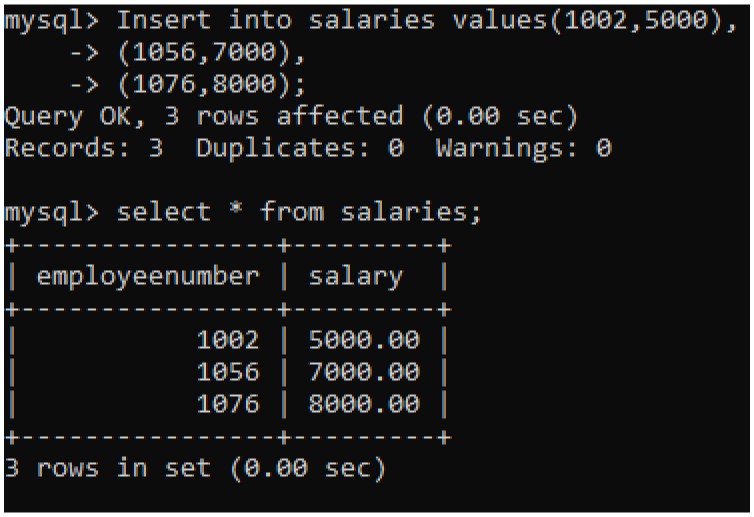
Table creation-
```
create table salaries(
employeenumber int primary key,
salary decimal(12,2) not null default 0
);
```


```
mysql>
mysql> Drop table salaries;
Query OK, 0 rows affected (0.01 sec)

mysql> create table salaries(
    -> employeenumber int primary key,
    -> salary decimal(12,2) not null default 0
    -> );
Query OK, 0 rows affected (0.01 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)
```

create table salarybudgets(
total decimal(15,2) not null
);

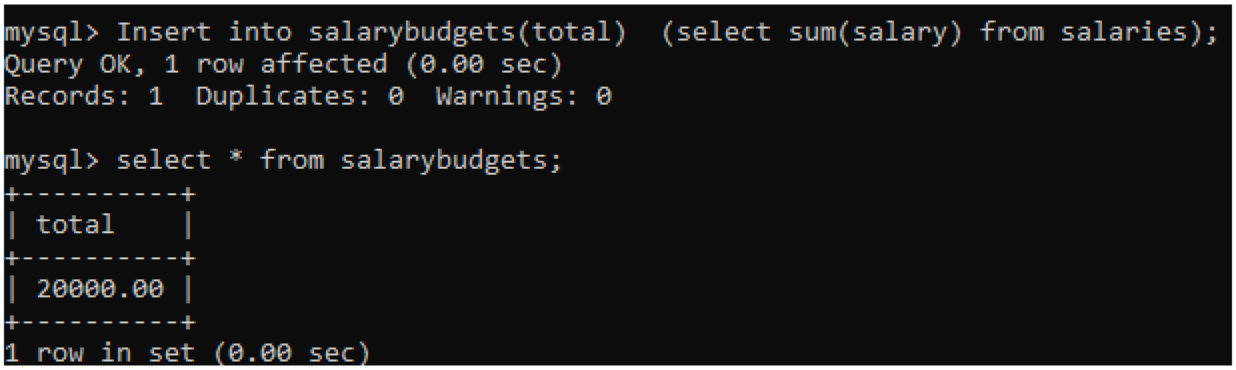

```
mysql> create table salarybudgets(
    -> total decimal(15,2) not null
    -> );
Query OK, 0 rows affected (0.01 sec)

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

Insert –
Insert into salaries values(1002,5000),
(1056,7000),
(1076,8000);


```
mysql> Insert into salaries values(1002,5000),
    -> (1056,7000),
    -> (1076,8000);
Query OK, 3 rows affected (0.00 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)
```

Insert into salarybudgets-

Insert into salarybudgets(total) (select sum(salary) from salaries);

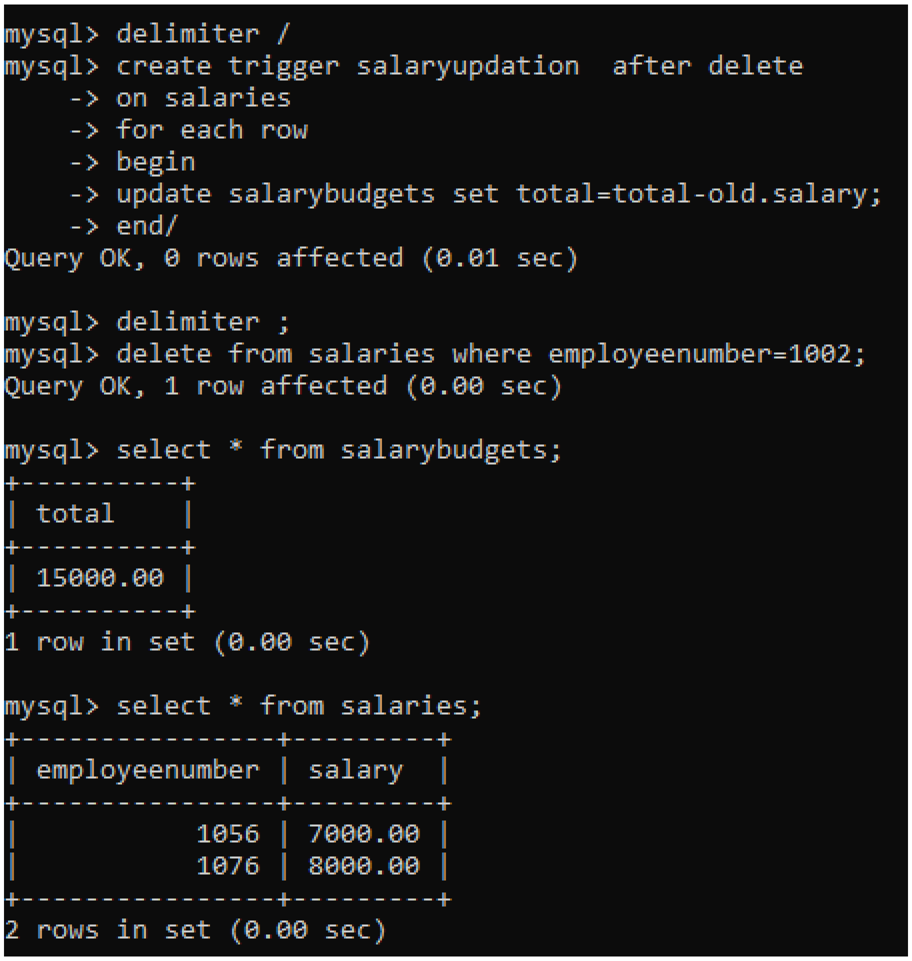

```
mysql> Insert into salarybudgets(total)  (select sum(salary) from salaries);
Query OK, 1 row affected (0.00 sec)
Records: 1  Duplicates: 0  Warnings: 0

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

Trigger creation-

```
create trigger salaryupdation  after delete
on salaries
for each row
begin
update salarybudgets set total=total-old.salary;
end/
```

delete-

delete from salaries where employeenumber=1002;



```
mysql> delimiter /
mysql> create trigger salaryupdation  after delete
    -> on salaries
    -> for each row
    -> begin
    -> update salarybudgets set total=total-old.salary;
    -> end/
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> delete from salaries where employeenumber=1002;
Query OK, 1 row affected (0.00 sec)

mysql> select * from salarybudgets;
+----------+
| total    |
+----------+
| 15000.00 |
+----------+
1 row in set (0.00 sec)

mysql> select * from salaries;
+----------------+---------+
| employeenumber | salary  |
+----------------+---------+
|           1056 | 7000.00 |
|           1076 | 8000.00 |
+----------------+---------+
2 rows in set (0.00 sec)
```