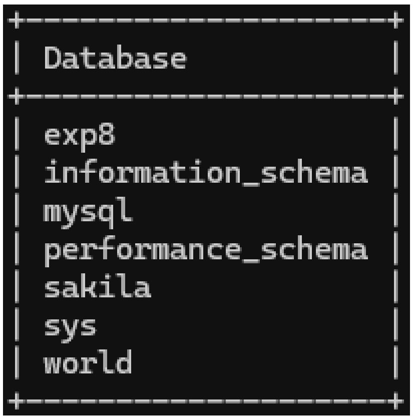
Assignment 8

```
Name : Ravish Ranjan
Course : MCA
Semester : 1st semester
```

Creating DB and Tables

Creating the database

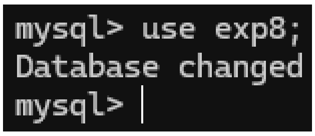
```
create database exp8;
```



```
+--------------------+
| Database           |
+--------------------+
| exp8               |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
```

Using the database

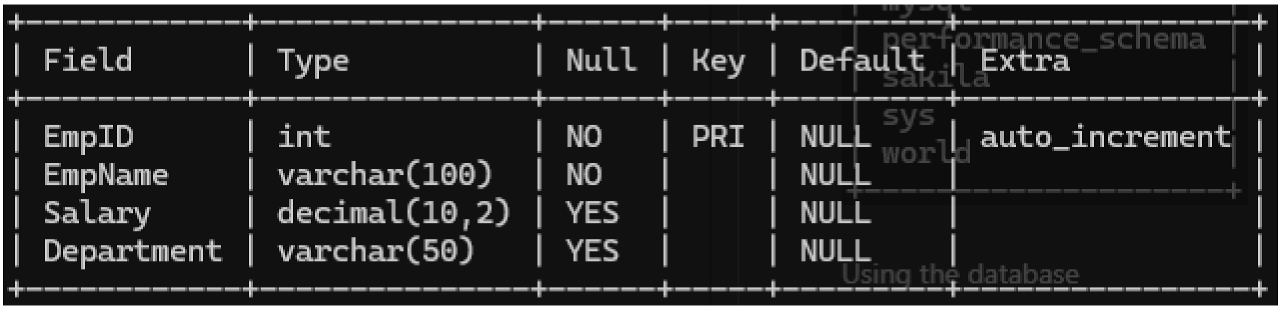
```
use exp8;
```



```
mysql> use exp8;
Database changed
mysql>
```

Creating table Employee

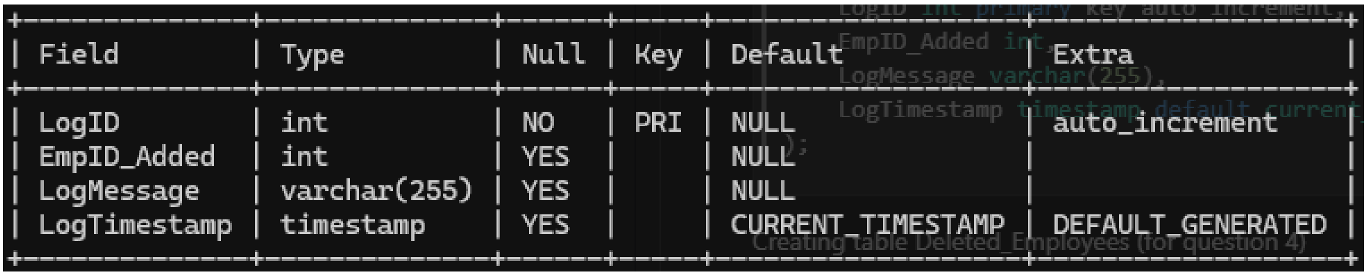
```
create table Employee (
    EmpID int primary key auto_increment,
    EmpName varchar(100) not null,
    Salary decimal(10, 2),
    Department varchar(50)
);
```



```
+------------+---------------+------+-----+---------+----------------+
| Field      | Type          | Null | Key | Default | Extra          |
+------------+---------------+------+-----+---------+----------------+
| EmpID      | int           | NO   | PRI | NULL    | auto_increment |
| EmpName    | varchar(100)  | NO   |     | NULL    |                |
| Salary     | decimal(10,2) | YES  |     | NULL    |                |
| Department | varchar(50)   | YES  |     | NULL    |                |
+------------+---------------+------+-----+---------+----------------+
```

Creating table Employee_Log (for question 2)

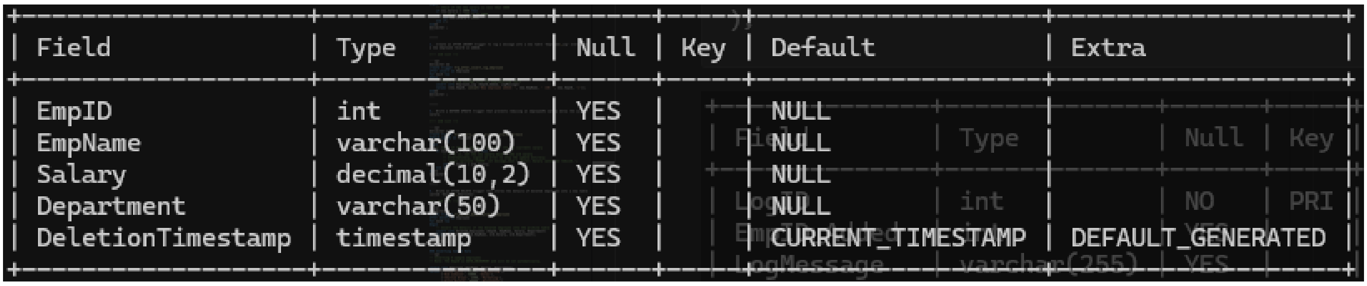
```
create table Employee_Log (
    LogID int primary key auto_increment,
    EmpID_Added int,
    LogMessage varchar(255),
    LogTimestamp timestamp default current_timestamp
);
```



```
+--------------+--------------+------+-----+-------------------+-------------------+
| Field        | Type         | Null | Key | Default           | Extra             |
+--------------+--------------+------+-----+-------------------+-------------------+
| LogID        | int          | NO   | PRI | NULL              | auto_increment    |
| EmpID_Added  | int          | YES  |     | NULL              |                   |
| LogMessage   | varchar(255) | YES  |     | NULL              |                   |
| LogTimestamp | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+--------------+--------------+------+-----+-------------------+-------------------+
```

Creating table Deleted_Employees (for question 4)

```
create table Deleted_Employees (
    EmpID int,
    EmpName varchar(100),
    Salary decimal(10, 2),
    Department varchar(50),
    DeletionTimestamp timestamp default current_timestamp
);
```

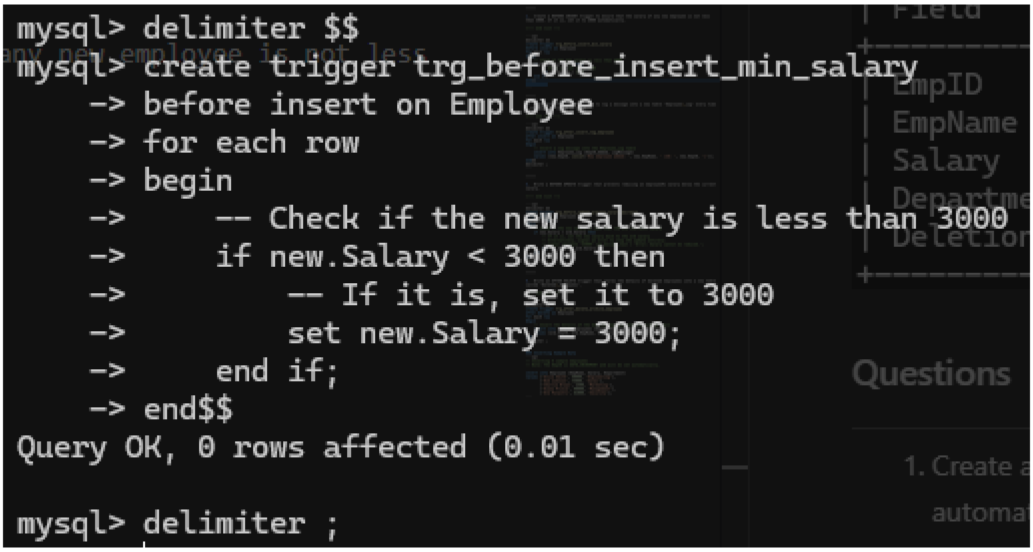


```
+-------------------+---------------+------+-----+-------------------+-------------------+
| Field             | Type          | Null | Key | Default           | Extra             |
+-------------------+---------------+------+-----+-------------------+-------------------+
| EmpID             | int           | YES  |     | NULL              |                   |
| EmpName           | varchar(100)  | YES  |     | NULL              |                   |
| Salary            | decimal(10,2) | YES  |     | NULL              |                   |
| Department        | varchar(50)   | YES  |     | NULL              |                   |
| DeletionTimestamp | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-------------------+---------------+------+-----+-------------------+-------------------+
```

Questions

1. Create a BEFORE INSERT trigger to ensure that the salary of any new employee is not less than 3000. If it is, set it to 3000 automatically.

```
delimiter $$
create trigger trg_before_insert_min_salary
before insert on Employee
for each row
begin
    -- Check if the new salary is less than 3000
    if new.Salary < 3000 then
        -- If it is, set it to 3000
        set new.Salary = 3000;
    end if;
end$$
delimiter ;
```

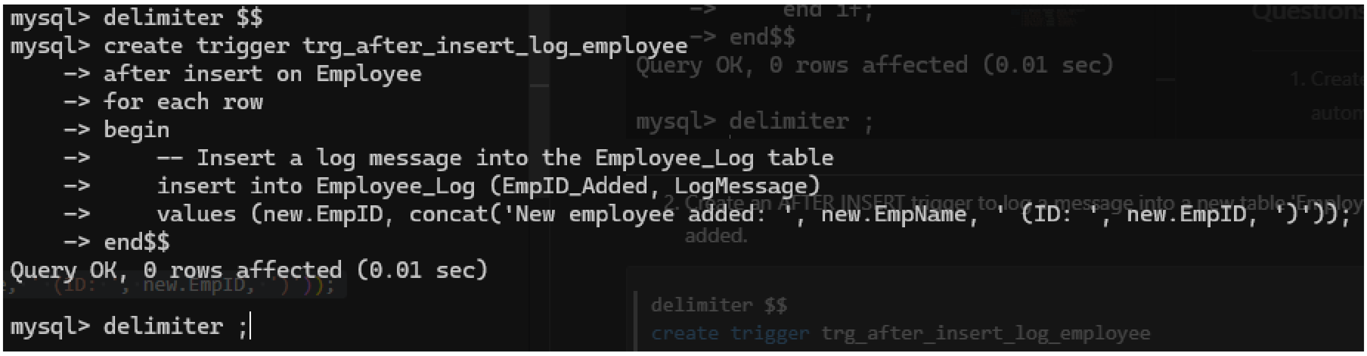


```
mysql> delimiter $$
mysql> create trigger trg_before_insert_min_salary
    -> before insert on Employee
    -> for each row
    -> begin
    ->     -- Check if the new salary is less than 3000
    ->     if new.Salary < 3000 then
    ->         -- If it is, set it to 3000
    ->         set new.Salary = 3000;
    ->     end if;
    -> end$$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

---

2. Create an AFTER INSERT trigger to log a message into a new table 'Employee_Log' every time a new employee record is added.

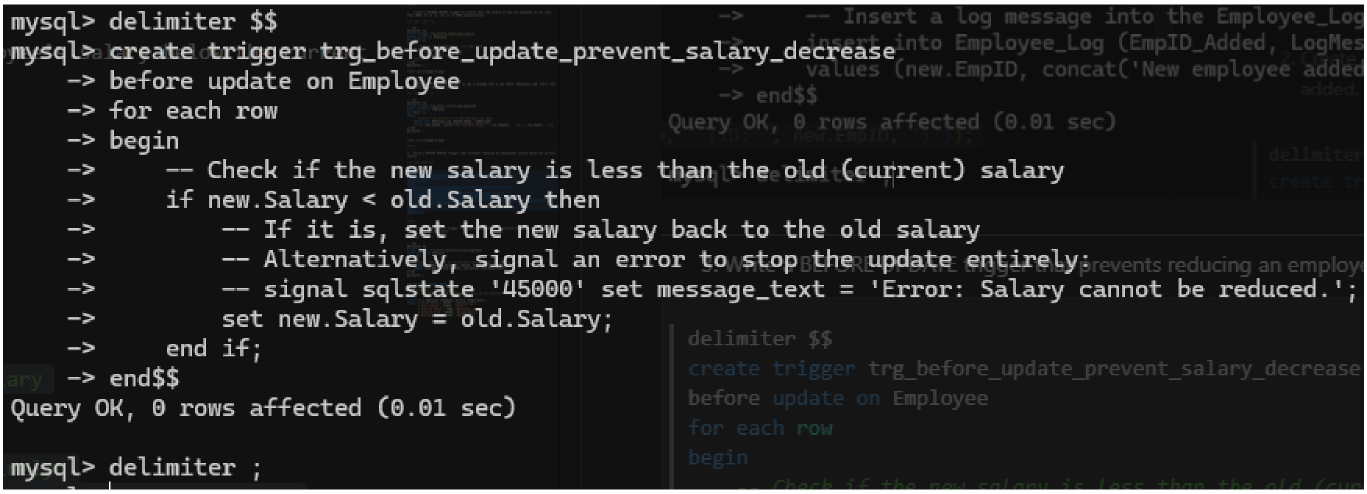
```
delimiter $$
create trigger trg_after_insert_log_employee
after insert on Employee
for each row
begin
    -- Insert a log message into the Employee_Log table
    insert into Employee_Log (EmpID_Added, LogMessage)
    values (new.EmpID, concat('New employee added: ', new.EmpName, ' (ID: ',
new.EmpID, ')'));
end$$
delimiter ;
```



```
mysql> delimiter $$
mysql> create trigger trg_after_insert_log_employee
    -> after insert on Employee
    -> for each row
    -> begin
    ->     -- Insert a log message into the Employee_Log table
    ->     insert into Employee_Log (EmpID_Added, LogMessage)
    ->     values (new.EmpID, concat('New employee added: ', new.EmpName, ' (ID: ', new.EmpID, ')'));
    -> end$$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

3. Write a BEFORE UPDATE trigger that prevents reducing an employee's salary below the current salary.

```
delimiter $$
create trigger trg_before_update_prevent_salary_decrease
before update on Employee
for each row
begin
    -- Check if the new salary is less than the old (current) salary
    if new.Salary < old.Salary then
        -- If it is, set the new salary back to the old salary
        -- Alternatively, signal an error to stop the update entirely:
        -- signal sqlstate '45000' set message_text = 'Error: Salary cannot be reduced.';
        set new.Salary = old.Salary;
    end if;
end$$
delimiter ;
```

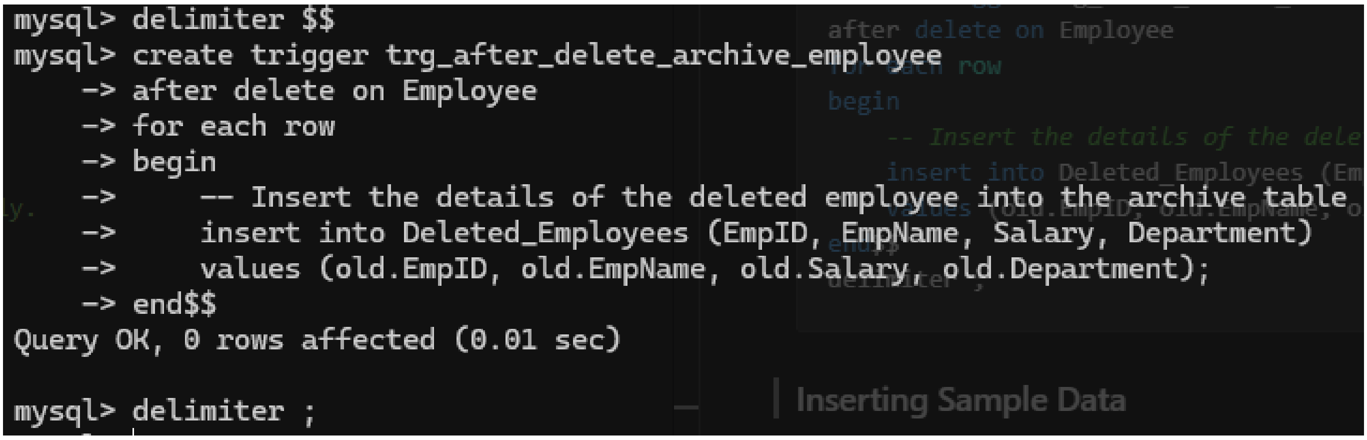


```
mysql> delimiter $$
mysql> create trigger trg_before_update_prevent_salary_decrease
    -> before update on Employee
    -> for each row
    -> begin
    ->     -- Check if the new salary is less than the old (current) salary
    ->     if new.Salary < old.Salary then
    ->         -- If it is, set the new salary back to the old salary
    ->         -- Alternatively, signal an error to stop the update entirely:
    ->         -- signal sqlstate '45000' set message_text = 'Error: Salary cannot be reduced.';
    ->         set new.Salary = old.Salary;
    ->     end if;
    -> end$$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

4. Write an AFTER DELETE trigger that stores the details of deleted employees into a new table called 'Deleted_Employees'.

```
delimiter $$
create trigger trg_after_delete_archive_employee
after delete on Employee
```

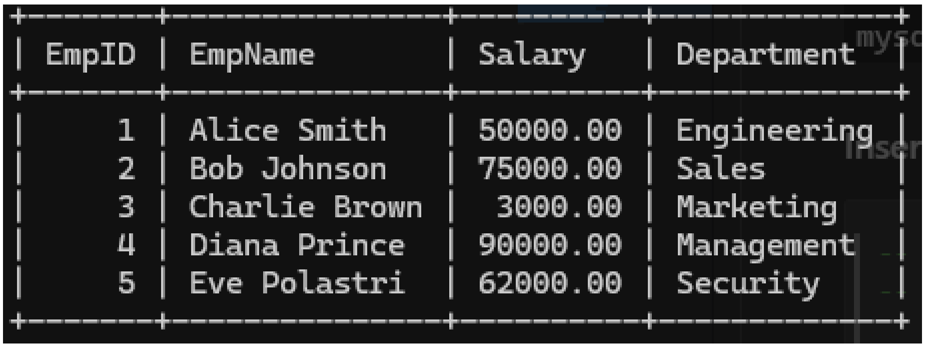
```
for each row
begin
    -- Insert the details of the deleted employee into the archive table
    insert into Deleted_Employees (EmpID, EmpName, Salary, Department)
    values (old.EmpID, old.EmpName, old.Salary, old.Department);
end$$
delimiter ;
```



```
mysql> delimiter $$
mysql> create trigger trg_after_delete_archive_employee
    -> after delete on Employee
    -> for each row
    -> begin
    ->     -- Insert the details of the deleted employee into the archive table
    ->     insert into Deleted_Employees (EmpID, EmpName, Salary, Department)
    ->     values (old.EmpID, old.EmpName, old.Salary, old.Department);
    -> end$$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

Inserting Sample Data

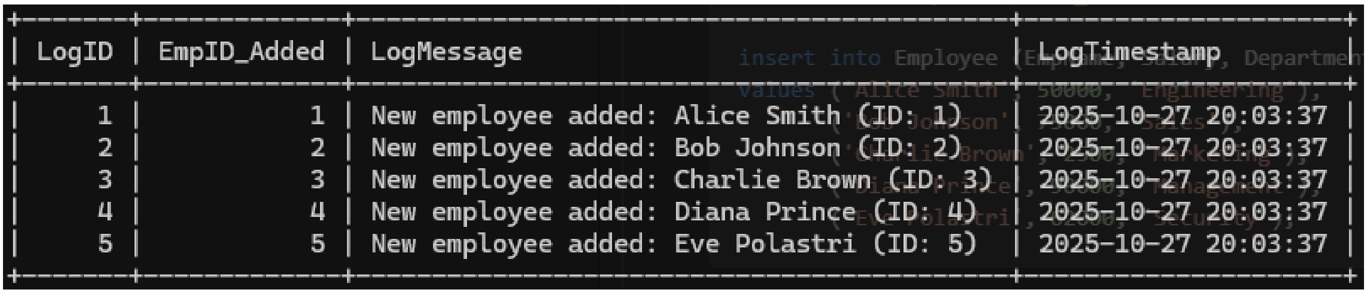
```
-- Inserting 5 sample employees
-- Note: The EmpID is AUTO_INCREMENT and will be set automatically.

insert into Employee (EmpName, Salary, Department)
values ('Alice Smith', 50000, 'Engineering'),
       ('Bob Johnson', 75000, 'Sales'),
       ('Charlie Brown', 2500, 'Marketing'),
       ('Diana Prince', 90000, 'Management'),
       ('Eve Polastri', 62000, 'Security');
```



```
+-------+---------------+----------+-------------+
| EmpID | EmpName       | Salary   | Department  |
+-------+---------------+----------+-------------+
|     1 | Alice Smith   | 50000.00 | Engineering |
|     2 | Bob Johnson   | 75000.00 | Sales       |
|     3 | Charlie Brown |  3000.00 | Marketing   |
|     4 | Diana Prince  | 90000.00 | Management  |
|     5 | Eve Polastri  | 62000.00 | Security    |
+-------+---------------+----------+-------------+
```

Employee Insertion Logs

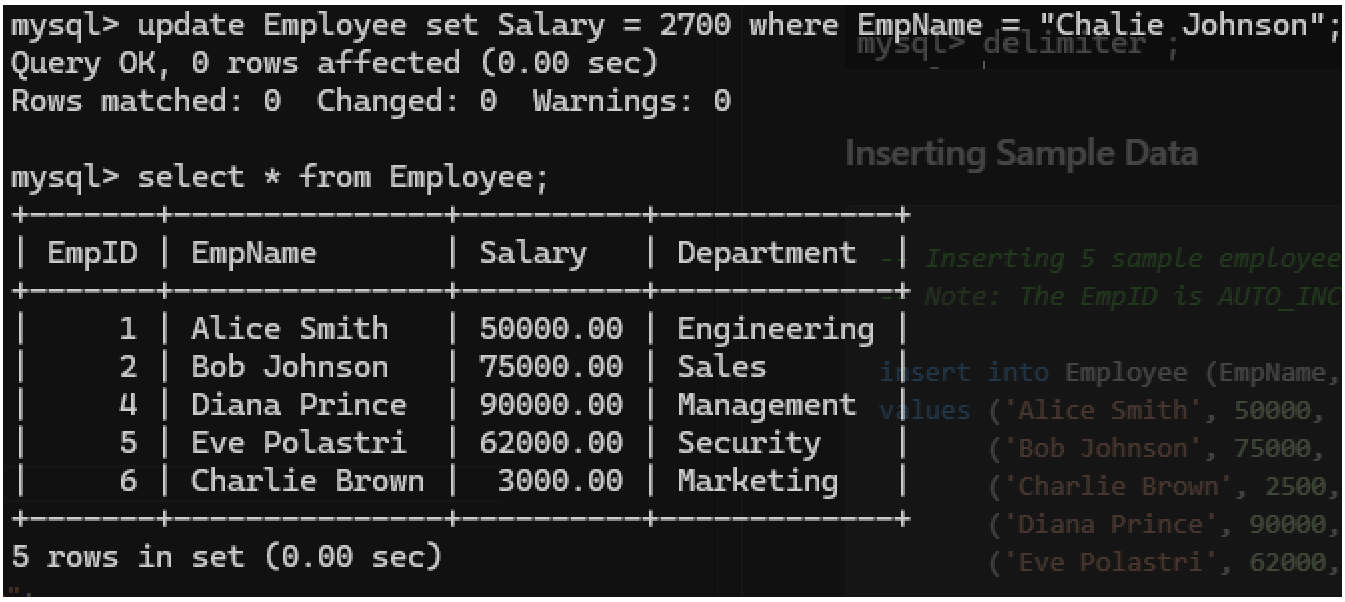
```
select * from Employee_Log;
```



```
+-------+-------------+------------------------------------------+---------------------+
| LogID | EmpID_Added | LogMessage                               | LogTimestamp        |
+-------+-------------+------------------------------------------+---------------------+
|     1 |           1 | New employee added: Alice Smith (ID: 1)   | 2025-10-27 20:03:37 |
|     2 |           2 | New employee added: Bob Johnson (ID: 2)   | 2025-10-27 20:03:37 |
|     3 |           3 | New employee added: Charlie Brown (ID: 3) | 2025-10-27 20:03:37 |
|     4 |           4 | New employee added: Diana Prince (ID: 4)  | 2025-10-27 20:03:37 |
|     5 |           5 | New employee added: Eve Polastri (ID: 5)  | 2025-10-27 20:03:37 |
+-------+-------------+------------------------------------------+---------------------+
```

Demonstrating update trigger

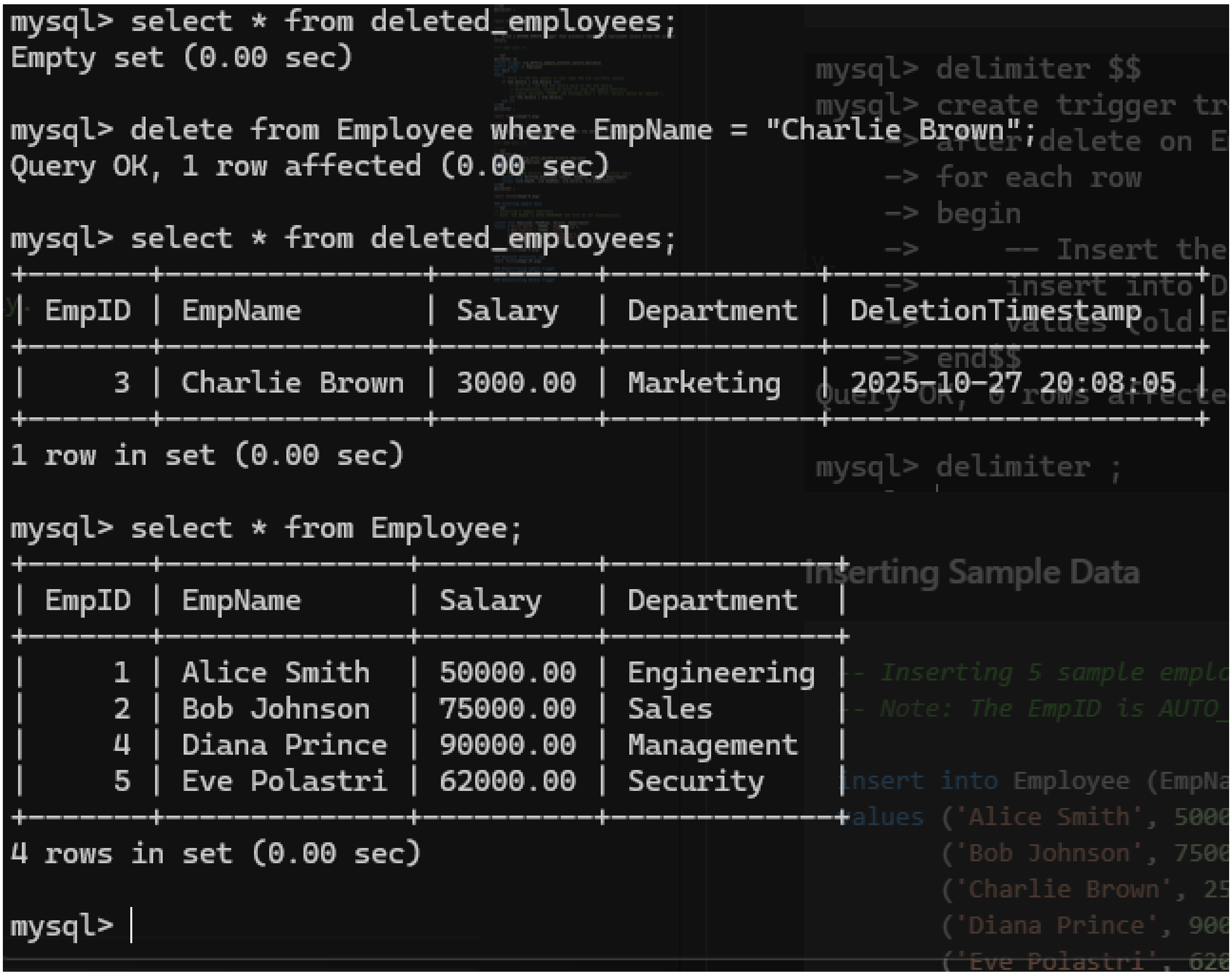
```
update Employee set Salary = 2700 where EmpName = "Chalie Johnson";
select * from Employee;
```



```
mysql> update Employee set Salary = 2700 where EmpName = "Chalie Johnson";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> select * from Employee;
+-------+---------------+----------+-------------+
| EmpID | EmpName       | Salary   | Department  |
+-------+---------------+----------+-------------+
|     1 | Alice Smith   | 50000.00 | Engineering |
|     2 | Bob Johnson   | 75000.00 | Sales       |
|     4 | Diana Prince  | 90000.00 | Management  |
|     5 | Eve Polastri  | 62000.00 | Security    |
|     6 | Charlie Brown |  3000.00 | Marketing   |
+-------+---------------+----------+-------------+
5 rows in set (0.00 sec)
```

Demonstrating delete trigger

```
select * from deleted_employees;
delete from EMployee where EmpName = "Charlie Johnson";
select * from deleted_emploees;
select * from Employee;
```



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

mysql> delete from Employee where EmpName = "Charlie Brown";
Query OK, 1 row affected (0.00 sec)

mysql> select * from deleted_employees;
+-------+---------------+---------+------------+---------------------+
| EmpID | EmpName       | Salary  | Department | DeletionTimestamp   |
+-------+---------------+---------+------------+---------------------+
|     3 | Charlie Brown | 3000.00 | Marketing  | 2025-10-27 20:08:05 |
+-------+---------------+---------+------------+---------------------+
1 row in set (0.00 sec)

mysql> select * from Employee;
+-------+--------------+----------+-------------+
| EmpID | EmpName      | Salary   | Department  |
+-------+--------------+----------+-------------+
|     1 | Alice Smith  | 50000.00 | Engineering |
|     2 | Bob Johnson  | 75000.00 | Sales       |
|     4 | Diana Prince | 90000.00 | Management  |
|     5 | Eve Polastri | 62000.00 | Security    |
+-------+--------------+----------+-------------+
4 rows in set (0.00 sec)

mysql>
```