



KIU

SEN4303

Advanced Database Systems

Assignment-1

Student ID: 11366

Student Name: B.R.C. Dilshan

Submitted Date: 12/27/2025

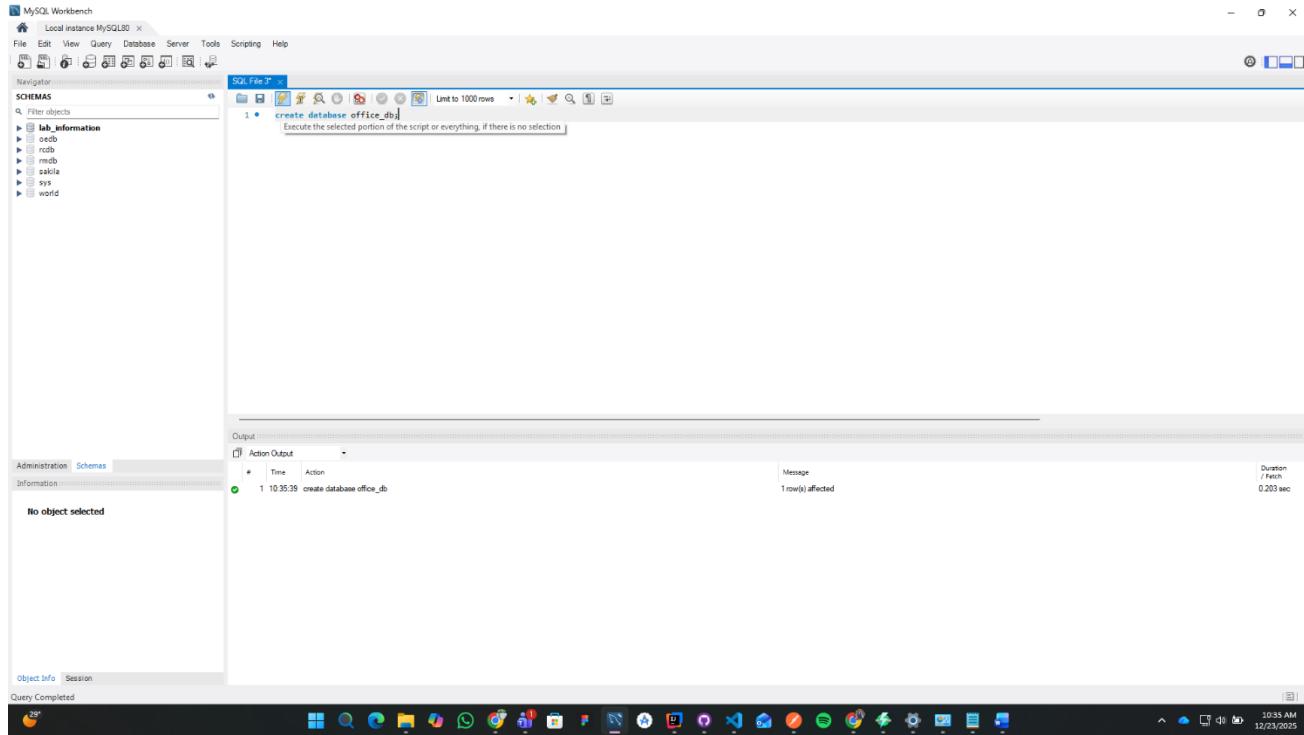
Table of Contents

1. Database and Tables.....	3
2. Constraints	7
3. Triggers	11
4. Stored Procedures.....	16
5. Testing	20
6. Database Support for API Endpoints	32

1. Database and Tables

Create database called office_db.

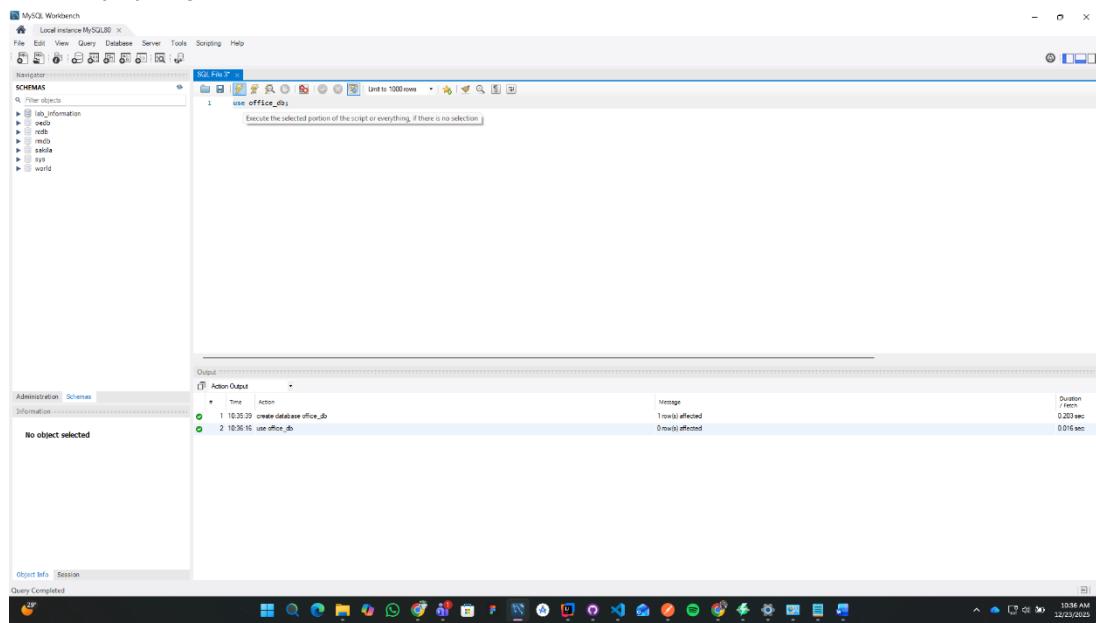
- *Sql query:* create database office_db;



The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, the command `create database office_db;` is entered. The Output tab shows the execution results: a single row was affected, and the duration was 0.203 sec. The status bar at the bottom indicates the time is 10:35 AM on 12/23/2025.

Create tables.

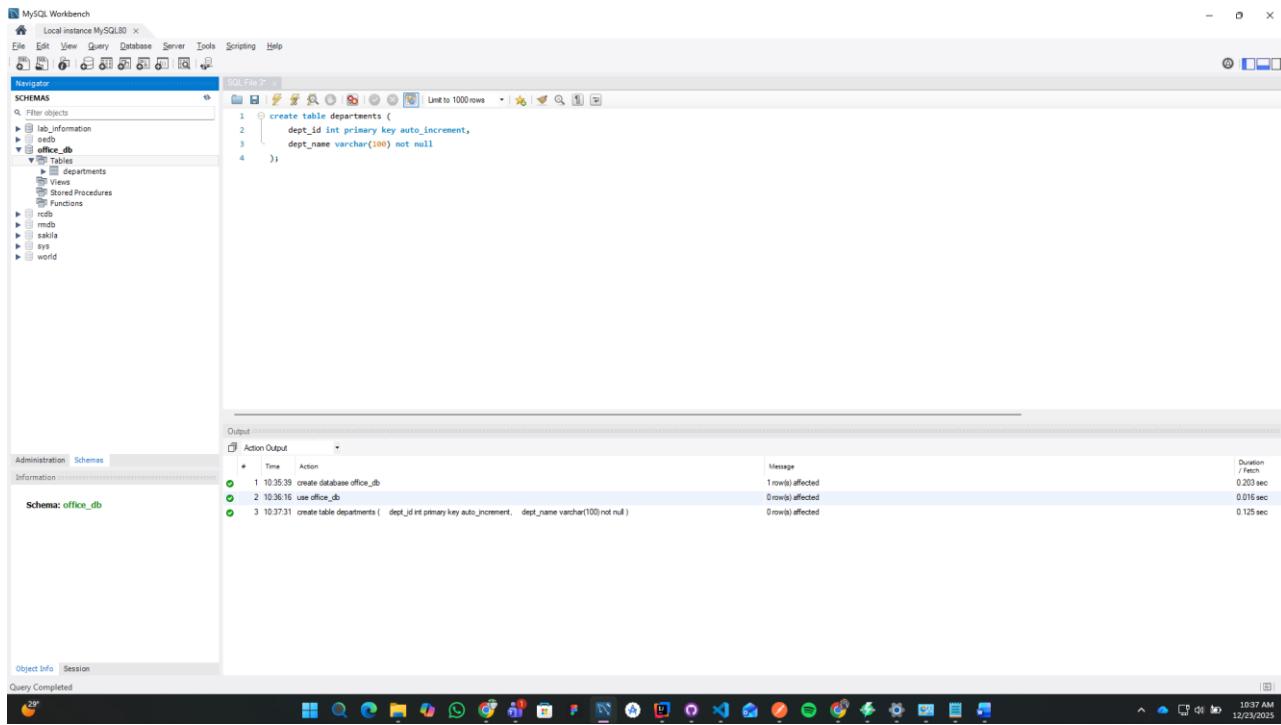
- *Sql query:* use office_db;



The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, the command `use office_db;` is entered. The Output tab shows the execution results: two rows were affected, and the duration was 0.016 sec. The status bar at the bottom indicates the time is 10:35 AM on 12/23/2025.

➤ Department table

- *Sql query:* create table departments (
dept_id int primary key auto_increment,
dept_name varchar(100) not null
);



The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left lists databases: 'innodb', 'mysql', 'office_db' (selected), 'performance_schema', 'rdsdb', 'rmbdb', 'sakila', 'sys', and 'world'. The 'SQL' pane contains the SQL code for creating the 'departments' table. The 'Output' pane shows the execution log with three entries: 1. '10:35:39 create database office_db' (Message: 1row(s) affected, Duration: 0.203 sec). 2. '10:36:16 use office_db' (Message: 0row(s) affected, Duration: 0.016 sec). 3. '10:37:31 create table departments (dept_id int primary key auto_increment, dept_name varchar(100) not null)' (Message: 0row(s) affected, Duration: 0.129 sec). The status bar at the bottom right shows the time as 10:37 AM and the date as 12/23/2025.

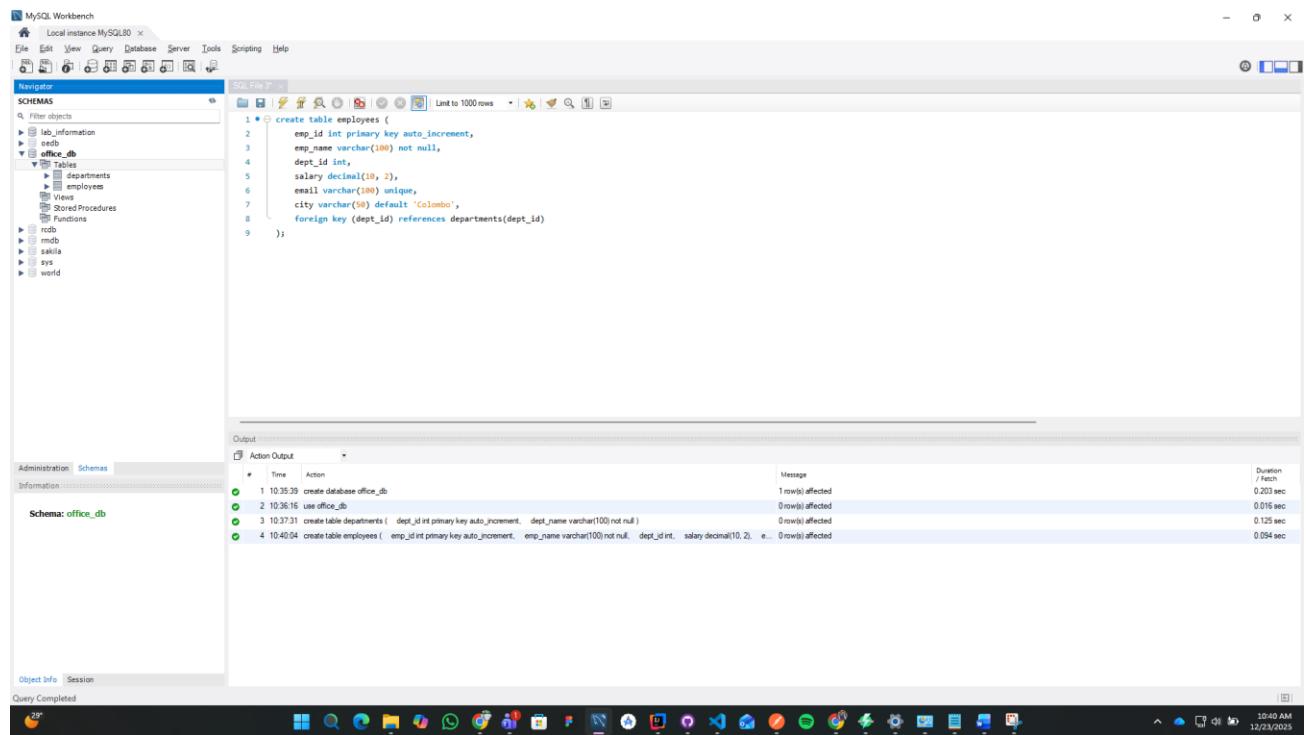
```
1 create table departments (
2     dept_id int primary key auto_increment,
3     dept_name varchar(100) not null
4 );
```

#	Time	Action	Message	Duration
1	10:35:39	create database office_db	1row(s) affected	0.203 sec
2	10:36:16	use office_db	0row(s) affected	0.016 sec
3	10:37:31	create table departments (dept_id int primary key auto_increment, dept_name varchar(100) not null)	0row(s) affected	0.129 sec

➤ Employee table

- *Sql query:*

```
create table employees (
    emp_id int primary key auto_increment,
    emp_name varchar(100) not null,
    dept_id int,
    salary decimal(10, 2),
    email varchar(100) unique,
    city varchar(50) default 'Colombo',
    foreign key (dept_id) references departments(dept_id)
);
```



The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the database structure, including the 'office_db' schema which contains the 'departments' and 'employees' tables. The 'SQL' tab in the center contains the SQL code for creating the 'employees' table. The 'Output' pane at the bottom shows the execution log with four entries: 1. create database office_db (1 row(s) affected, 0.203 sec), 2. use office_db (0 row(s) affected, 0.016 sec), 3. create table departments (0 row(s) affected, 0.125 sec), and 4. create table employees (0 row(s) affected, 0.059 sec). The status bar at the bottom right indicates the session is 'Query Completed' at 10:40 AM on 12/23/2025.

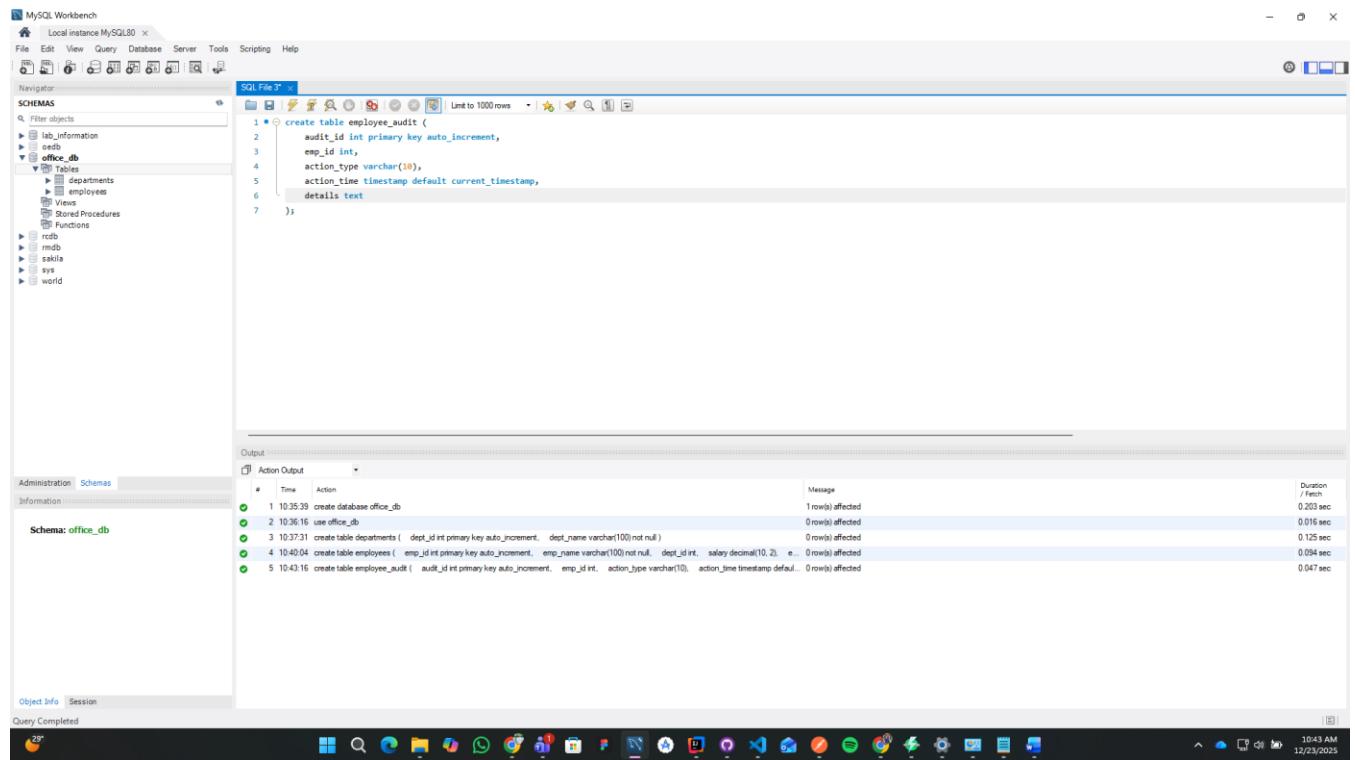
```
1 • 1 create table employees (
2     emp_id int primary key auto_increment,
3     emp_name varchar(100) not null,
4     dept_id int,
5     salary decimal(10, 2),
6     email varchar(100) unique,
7     city varchar(50) default 'Colombo',
8     foreign key (dept_id) references departments(dept_id)
9 );
```

Action	Time	Message	Duration
create database office_db	1 10:35:39	1 row(s) affected	/ Fetch 0.203 sec
use office_db	2 10:36:16	0 row(s) affected	0.016 sec
create table departments (dept_id int primary key auto_increment, dept_name varchar(100) not null)	3 10:36:31	0 row(s) affected	0.125 sec
create table employees (emp_id int primary key auto_increment, emp_name varchar(100) not null, dept_id int, salary decimal(10, 2), email varchar(100) unique, city varchar(50) default 'Colombo', foreign key (dept_id) references departments(dept_id))	4 10:40:04	0 row(s) affected	0.059 sec

➤ Employee_audit table

- *Sql query:*

```
create table employee_audit (
    audit_id int primary key auto_increment,
    emp_id int,
    action_type varchar(10),
    action_time timestamp default current_timestamp,
    details text
);
```



The screenshot shows the MySQL Workbench interface. The left sidebar (Navigator) shows the database structure for 'office_db', including 'Tables' like 'departments', 'employees', and 'employees'. The central area (SQL File Tab) contains the SQL code for creating the 'employee_audit' table. The bottom section (Output) displays the execution log with five entries, each showing a successful execution of a SQL command. The log includes columns for Action, Time, Message, and Duration.

Action	Time	Message	Duration
create database office_db	1 10:35:39	1 row(s) affected	0.203 sec
use office_db	2 10:36:16	0 row(s) affected	0.016 sec
create table departments (dept_id int primary key auto_increment, dept_name varchar(100) not null)	3 10:37:31	0 row(s) affected	0.125 sec
create table employees (emp_id int primary key auto_increment, emp_name varchar(100) not null, dept_id int, salary decimal(10,2), e...	4 10:40:04	0 row(s) affected	0.094 sec
create table employee_audit (audit_id int primary key auto_increment, emp_id int, action_type varchar(10), action_time timestamp default...	5 10:43:16	0 row(s) affected	0.047 sec

Sql file:



create_database.sql

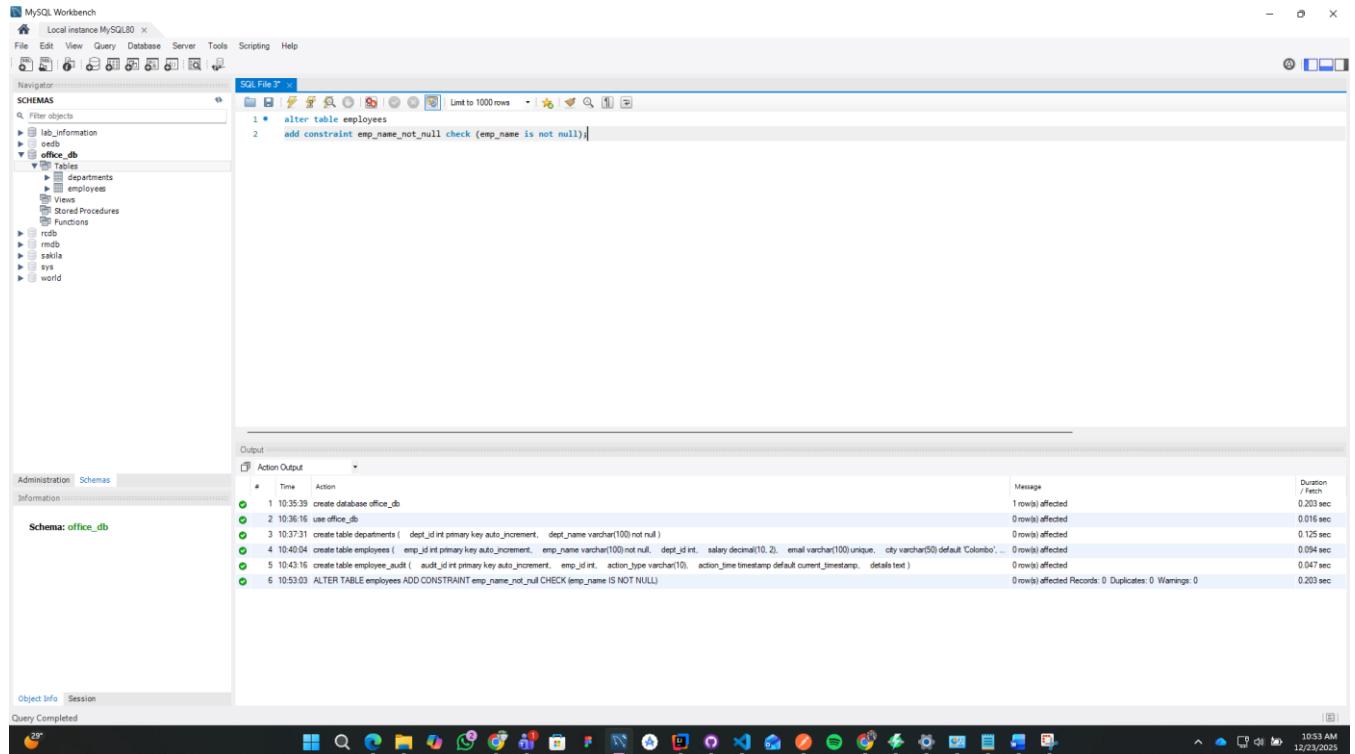
2. Constraints

- emp_name must NOT be NULL.

(This was also done during the creation of the employees table.)

- *Sql query:* alter table employees

```
add constraint emp_name_not_null check (emp_name is not null);
```



- emp_id should be the PRIMARY KEY.

(This was done during the creation of the employees table.)

- *Sql query:* create table employees (

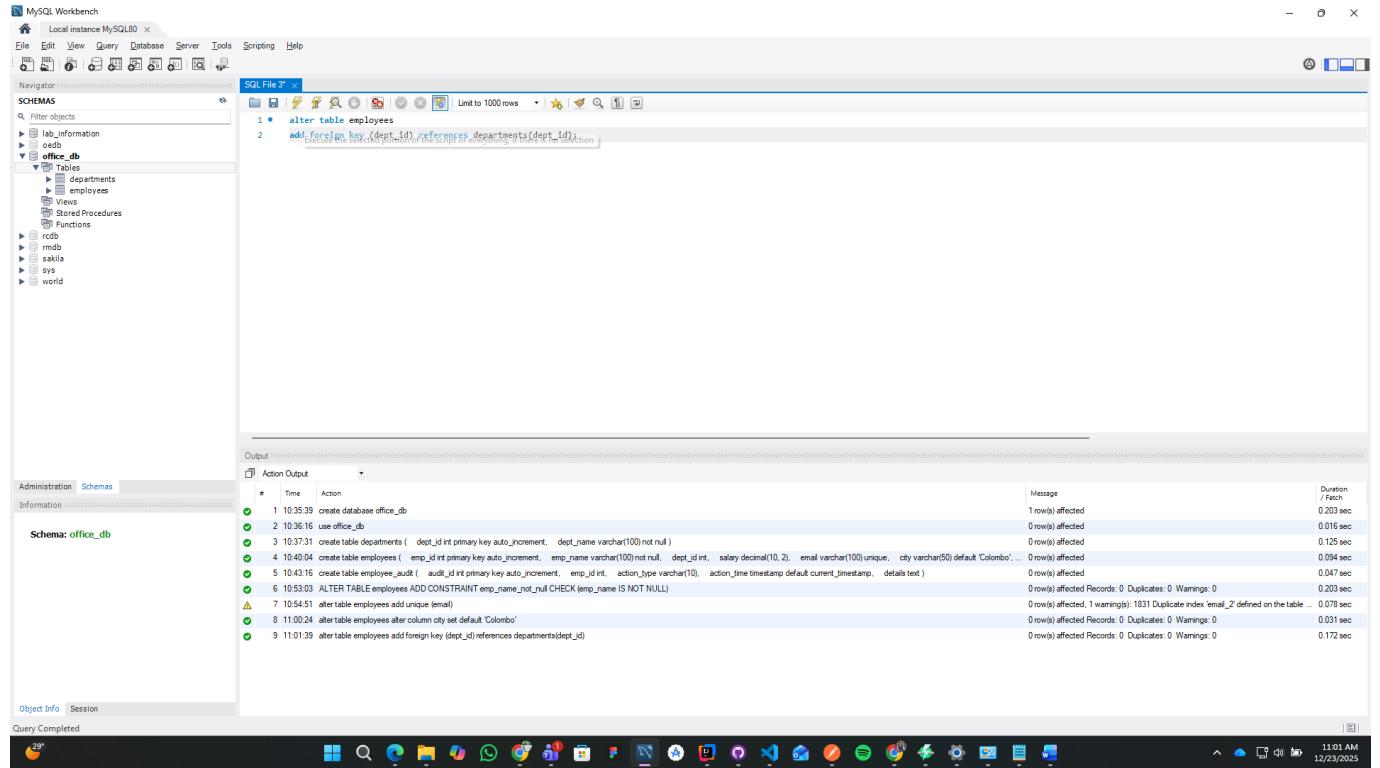
```
emp_id int primary key auto_increment,  
emp_name varchar(100) not null,  
dept_id int,  
salary decimal(10, 2),  
email varchar(100) unique,  
city varchar(50) default 'Colombo',  
foreign key (dept_id) references departments(dept_id)  
);
```

- department in employees should reference dept_id in departments.

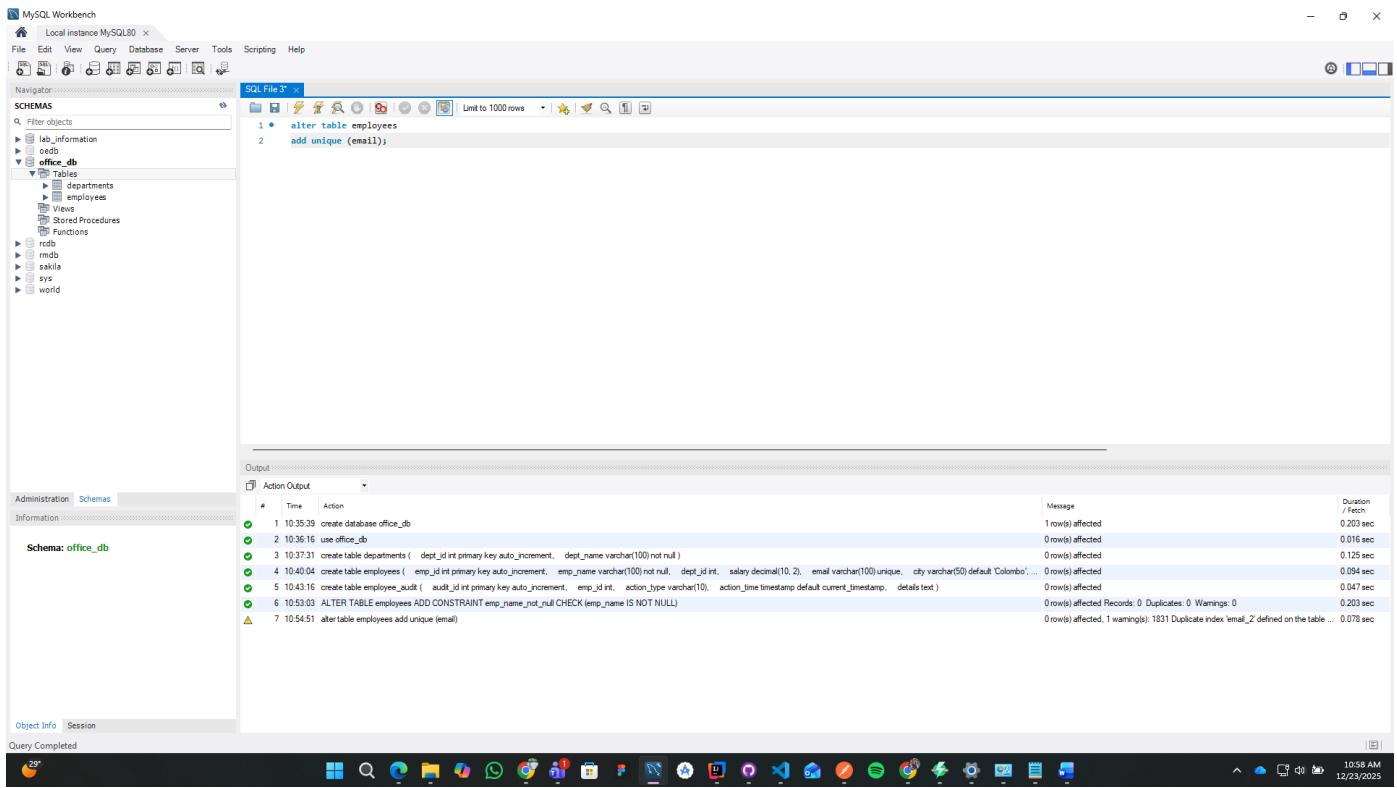
- *Sql query:*

alter table employees

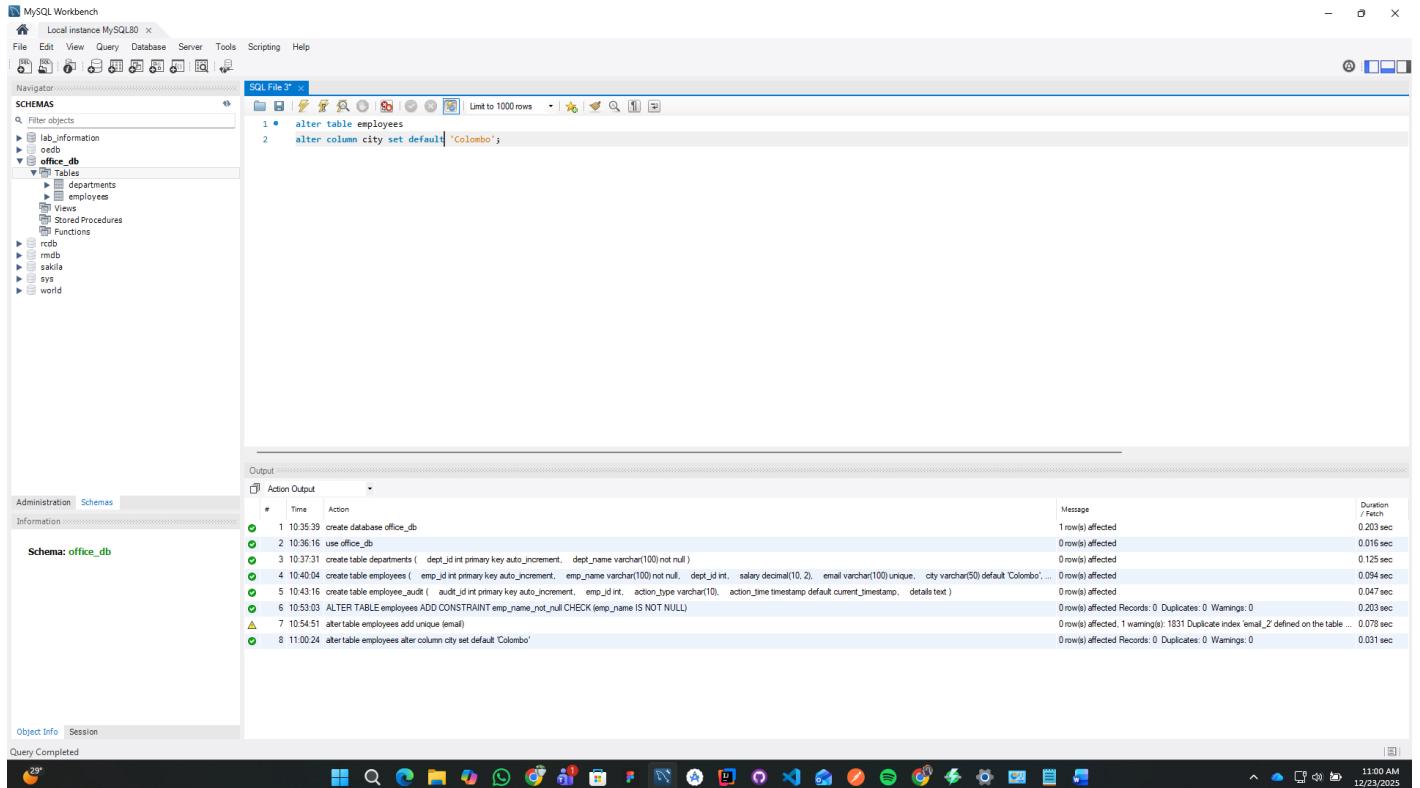
add foreign key (dept_id) references departments(dept_id);



- Add a UNIQUE constraint on a sample column (e.g., email).
(This was also done during the creation of the employees table.)
 - *Sql query:* alter table employees
add unique (email);



- Add a DEFAULT value for a column (e.g., city = 'Colombo').
(This was also done during the creation of the employees table.)
 - *Sql query:* alter table employees
alter column city set default 'Colombo';



The screenshot shows the MySQL Workbench interface. The SQL Editor pane contains the following SQL code:

```

1 • alter table employees
2   alter column city set default 'Colombo';

```

The Output pane displays the execution log:

#	Time	Action	Message	Duration / Fetch
1	10:35:39	create database office_db	1 row(s) affected	0.203 sec
2	10:36:16	use office_db	0 row(s) affected	0.016 sec
3	10:37:31	create table departments (_dept_id int primary key auto_increment, _dept_name varchar(100) not null)	0 row(s) affected	0.125 sec
4	10:40:04	create table employees (_emp_id int primary key auto_increment, _emp_name varchar(100) not null, _dept_id int, _salary decimal(10, 2), _email varchar(100) unique, _city varchar(50) default 'Colombo', _hire_date date, _last_update timestamp default current_timestamp, _details text)	0 row(s) affected	0.094 sec
5	10:43:16	create table employee_audit (_audit_id int primary key auto_increment, _emp_id int, _action_type varchar(10), _action_time timestamp default current_timestamp, _details text)	0 row(s) affected	0.047 sec
6	10:53:00	ALTER TABLE employees ADD CONSTRAINT emp_name_no_null CHECK (emp_name IS NOT NULL)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.203 sec
7	10:54:51	alter table employees add unique email	0 row(s) affected, 1 warning(s): 1831 Duplicate index email_2 defined on the table ...	0.078 sec
8	11:00:24	alter table employees alter column city set default 'Colombo'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Sql file:



add_constraints.sql

3. Triggers

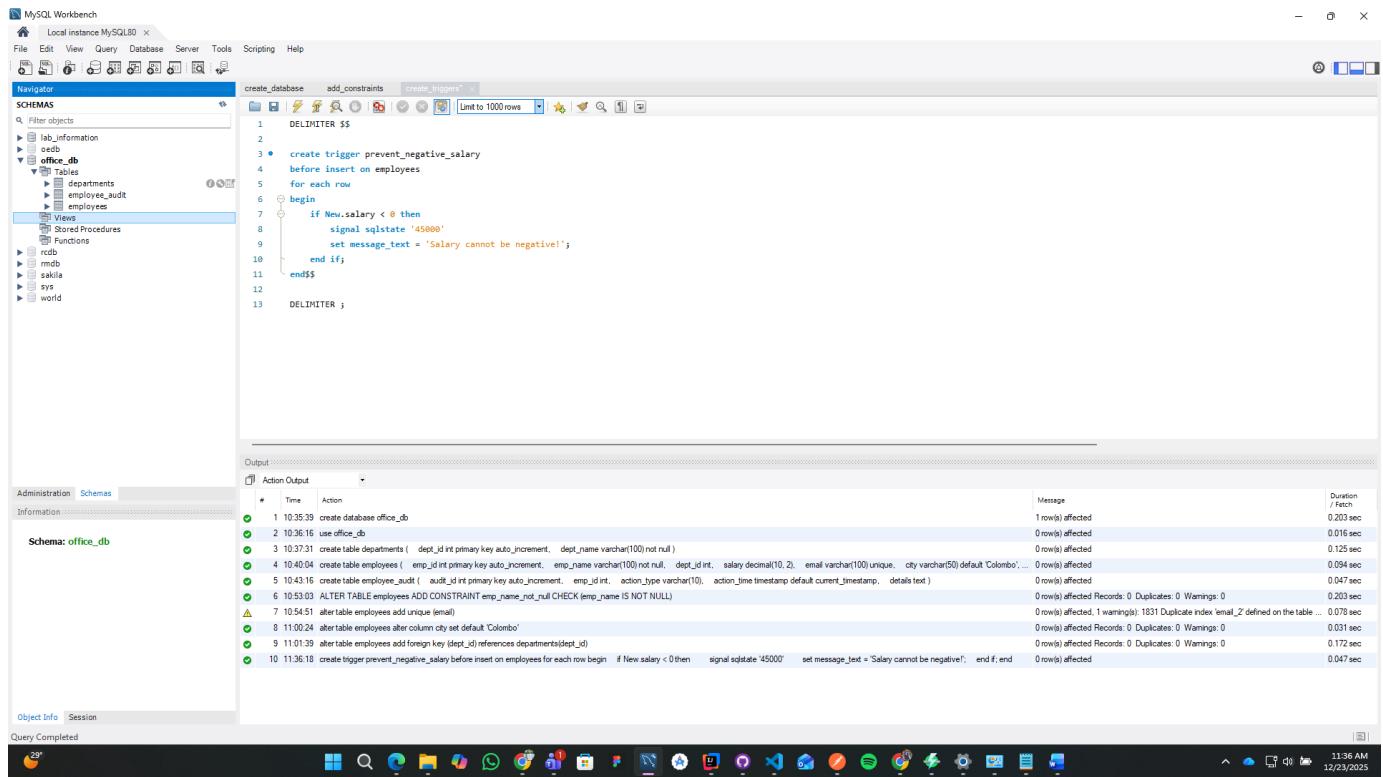
- Create a trigger to prevent inserting negative salaries.

- *Sql query (for insert) :*

```
DELIMITER $$
```

```
create trigger prevent_negative_salary
before insert on employees
for each row
begin
    if New.salary < 0 then
        signal sqlstate '45000'
        set message_text = 'Salary cannot be negative!';
    end if;
end$$
```

```
DELIMITER ;
```



```
DELIMITER $$

create trigger prevent_negative_salary
before insert on employees
for each row
begin
    if New.salary < 0 then
        signal sqlstate '45000'
        set message_text = 'Salary cannot be negative!';
    end if;
end$$

DELIMITER ;
```

Output:

#	Time	Action	Message	Duration / Fetch
1	10:36:39	create database office_db	1 row(s) affected	0.203 sec
2	10:36:16	use office_db	0 row(s) affected	0.016 sec
3	10:37:31	create table departments (dept_id int primary key auto_increment, dept_name varchar(100) not null)	0 row(s) affected	0.125 sec
4	10:40:04	create table employees (emp_id int primary key auto_increment, emp_name varchar(100) not null, dept_id int, salary decimal(10, 2), email varchar(100) unique, city varchar(50) default 'Colombo', hire_date date, manager_id int, job_id int, department_id int, hire_date timestamp default current_timestamp, details text)	0 row(s) affected	0.094 sec
5	10:43:16	create table employee_audit (audit_id int primary key auto_increment, emp_id int, action_type varchar(10), action_time timestamp default current_timestamp, details text)	0 row(s) affected	0.047 sec
6	10:53:03	ALTER TABLE employees ADD CONSTRAINT emp_name_not_null CHECK (emp_name IS NOT NULL)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.203 sec
7	10:54:51	alter table employees add unique (email)	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'email_2' defined on the table ...	0.078 sec
8	11:00:24	alter table employees alter column city set default 'Colombo'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
9	11:01:39	alter table employees add foreign key (dept_id) references departments(dept_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.172 sec
10	11:36:18	create trigger prevent_negative_salary before insert on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative!'; end if; end	0 row(s) affected	0.047 sec

- *Sql query (for update) :*

DELIMITER \$\$

```
create trigger prevent_negative_salary_update
before update on employees
for each row
begin
    if New.salary < 0 then
        signal sqlstate '45000'
        set message_text = 'Salary cannot be negative!';
    end if;
end$$
```

DELIMITER ;

```
DELIMITER $$

create trigger prevent_negative_salary_update
before update on employees
for each row
begin
    if New.salary < 0 then
        signal sqlstate '45000'
        set message_text = 'Salary cannot be negative!';
    end if;
end$$

DELIMITER ;
```

#	Time	Action	Message	Duration / Fetch
1	10:35:39	create database office_db	1 row(s) affected	0.203 sec
2	10:36:16	use office_db	0 rows(s) affected	0.016 sec
3	10:37:31	create table departments (dept_id int primary key auto_increment, dept_name varchar(100) not null)	0 rows(s) affected	0.125 sec
4	10:40:04	create table employees (emp_id int primary key auto_increment, emp_name varchar(100) not null, dept_id int, salary decimal(10,2), email varchar(100) unique, city varchar(50) default 'Colombo', hire_date date, details text)	0 rows(s) affected	0.094 sec
5	10:43:16	create table employee_audit (audit_id int primary key auto_increment, emp_id int, action_type varchar(10), action_time timestamp default current_timestamp, details text)	0 rows(s) affected	0.047 sec
6	10:53:03	ALTER TABLE employees ADD CONSTRAINT emp_name_not_null CHECK (emp_name IS NOT NULL)	0 rows(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.203 sec
7	10:54:51	alter table employees add unique (email)	0 rows(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
8	11:00:24	alter table employees alter column city set default 'Colombo'	0 rows(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
9	11:01:39	alter table employees add foreign key (dept_id) references departments(dept_id)	0 rows(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.172 sec
10	11:36:18	create trigger prevent_negative_salary before insert on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative!'; end if; end	0 rows(s) affected	0.047 sec
11	11:38:53	create trigger prevent_negative_salary_update before update on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative!'; end if; end	0 rows(s) affected	0.016 sec

- Create triggers to log every INSERT, UPDATE, and DELETE on employees into employee_audit.

1. Insert

- *Sql query:*

DELIMITER \$\$

```
create trigger audit_insert
after insert on employees
for each row
begin
    insert into employee_audit (emp_id, action_type, details)
    values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ',
    New.emp_name, ', Salary: ', New.salary));
end$$
```

DELIMITER ;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigation: Local instance MySQL80

SCHEMAS: Schemas: lab_information, office_db, rcdb, rmdb, skills, srs, world

Tables: office_db.employees, office_db.departments, office_db.employee_audit, office_db.foreign_keys, office_db.triggers

Triggers: office_db.prevent_negative_salary, office_db.audit_insert, office_db.prevent_negative_salary_update

Views: Stored Procedures: Functions: rcdb, rmdb, skills, srs

create_triggers*: create_database, add_constraints, create_triggers*

SQL File 7*

```
1  DELIMITER $$  
2  
3  create trigger audit_insert  
4  after insert on employees  
5  for each row  
6  begin  
7      insert into employee_audit (emp_id, action_type, details)  
8      values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ',  
9      New.emp_name, ', Salary: ', New.salary));  
10  
11  end$$  
12  
13  DELIMITER ;
```

Output:

Action	Time	Message	Duration
7 10:54:51 alter table employees add unique (email)	10:54:51	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'email_2' defined on the table...	0.078 sec
8 11:00:24 alter table employees alter column city set default 'Colombo'	11:00:24	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
9 11:01:39 alter table employees add foreign key (dept_id) references departments(dept_id)	11:01:39	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.172 sec
10 11:36:18 create trigger prevent_negative_salary before insert on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative!'; end if; end	11:36:18	0 row(s) affected	0.047 sec
11 11:38:53 create trigger prevent_negative_salary_update before update on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative!'; end if; end	11:38:53	0 row(s) affected	0.016 sec
12 11:54:41 create trigger audit_insert before update on employees for each row begin insert into employee_audit (emp_id,action_type,details) values (New.emp_id,'UPDATE',CONCAT('Updated employee: ',New.emp_name, ', Salary: ', New.salary)); end	11:54:41	0 row(s) affected	0.062 sec
13 11:55:39 create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id,action_type,details) values (New.emp_id,'INSERT',CONCAT('Inserted employee: ',New.emp_name, ', Salary: ', New.salary)); end	11:55:39	Error Code: 1359. Trigger already exists	0.047 sec
14 11:58:02 DROP TRIGGER audit_insert	11:58:02	0 row(s) affected	0.031 sec
15 11:59:24 create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id,action_type,details) values (New.emp_id,'INSERT',CONCAT('Inserted employee: ',New.emp_name, ', Salary: ', New.salary)); end	11:59:24	0 row(s) affected	0.031 sec
16 12:01:20 create trigger audit_update before update on employees for each row begin insert into employee_audit (emp_id,action_type,details) values (NEW.emp_id,'UPDATE',CONCAT('Updated employee: ',New.emp_name, ', Salary: ', New.salary)); end	12:01:20	0 row(s) affected	0.016 sec
17 12:08:11 DROP TRIGGER audit_update	12:08:11	0 row(s) affected	0.016 sec
18 12:08:19 DROP TRIGGER audit_update	12:08:19	0 row(s) affected	0.016 sec
19 12:09:37 create trigger audit_insert after insert on employees for each row begin insert into employee_audit (emp_id,action_type,details) values (New.emp_id,'INSERT',CONCAT('Inserted employee: ',New.emp_name, ', Salary: ', New.salary)); end	12:09:37	0 row(s) affected	0.031 sec

Object Info Session

Query Completed

30°C Sunny

12:10 PM 12/23/2025

2. Update

- *Sql query:*

DELIMITER \$\$

```
create trigger audit_update
after update on employees
for each row
begin
    insert into employee_audit (emp_id, action_type, details)
    values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ',
    NEW.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary));
end$$
```

DELIMITER ;

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

office_db

Tables

Triggers

Views

Stored Procedures

Functions

rdbo

rodb

saikila

sys

world

create_database add_constraints create_triggers SQL File 7*

1 DELIMITER \$\$
2 Execute the selected portion of the script or everything, if there is no selection.
3
4 create trigger audit_update
5 after update on employees
6 for each row
7 begin
8 insert into employee_audit (emp_id, action_type, details)
9 values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', NEW.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary));
10 end\$\$
11 DELIMITER ;

Output

#	Time	Action	Message	Duration / Fetch
8	11:00:24	alter table employees alter column city set default 'Colombo'	0 rows affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
9	11:01:39	alter table employees add foreign key (dept_id) references departments(dept_id)	0 rows affected Records: 0 Duplicates: 0 Warnings: 0	0.172 sec
10	11:36:18	create trigger prevent_negative_salary before insert on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative'; end if; end	0 rows affected	0.047 sec
11	11:38:53	create trigger prevent_negative_salary_update before update on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative'; end if; end	0 rows affected	0.016 sec
12	11:54:41	create trigger audit_insert before update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary)); end	0 rows affected	0.062 sec
13	11:55:39	create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary)); end	Error Code: 1359. Trigger already exists	0.047 sec
14	11:58:02	DROP TRIGGER audit_insert	0 rows affected	0.031 sec
15	11:59:24	create trigger audit_update before insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary)); end	0 rows affected	0.031 sec
16	12:01:20	create trigger audit_update before update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', NEW.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary)); end	0 rows affected	0.016 sec
17	12:08:11	DROP TRIGGER audit_update	0 rows affected	0.016 sec
18	12:08:19	DROP TRIGGER audit_update	0 rows affected	0.016 sec
19	12:09:26	create trigger audit_insert after insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary)); end	0 rows affected	0.031 sec
20	12:11:24	create trigger audit_update after update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', NEW.emp_name, ', Old Salary: ', OLD.salary, ', New Salary: ', NEW.salary)); end	0 rows affected	0.016 sec

Object Info Session

Query Completed

30°C Sunny

12:11 PM 12/23/2025

3. Delete

- *Sql query:*

DELIMITER \$\$

```
create trigger audit_delete
after delete on employees
for each row
begin
    insert into employee_audit (emp_id, action_type, details)
    values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ',
    OLD.emp_name, ', Salary: ', OLD.salary));
end$$
```

DELIMITER ;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure. The 'Tables' node for the 'office_db' schema contains the 'employee_audit' table and several triggers: 'prevent_negative_salary', 'audit_insert', 'prevent_negative_salary_update', 'audit_update', and 'audit_delete'.
- SQL Editor:** The SQL pane contains the trigger creation script:

```
1  DELIMITER $$

2
3  create trigger audit_delete
4  after delete on employees
5  for each row
6  begin
7      insert into employee_audit (emp_id, action_type, details)
8      values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ',
9      OLD.emp_name, ', Salary: ', OLD.salary));
10 end$$

11 DELIMITER ;
```
- Output:** The results pane shows the execution of the script with 21 rows of log output. The log includes actions like creating triggers, inserting into the audit table, and dropping triggers, with details on time, action, message, and duration.

Sql file:



create_triggers.sql

4. Stored Procedures

- Procedure to retrieve all employees.

- *Sql query:*

```
DELIMITER $$
```

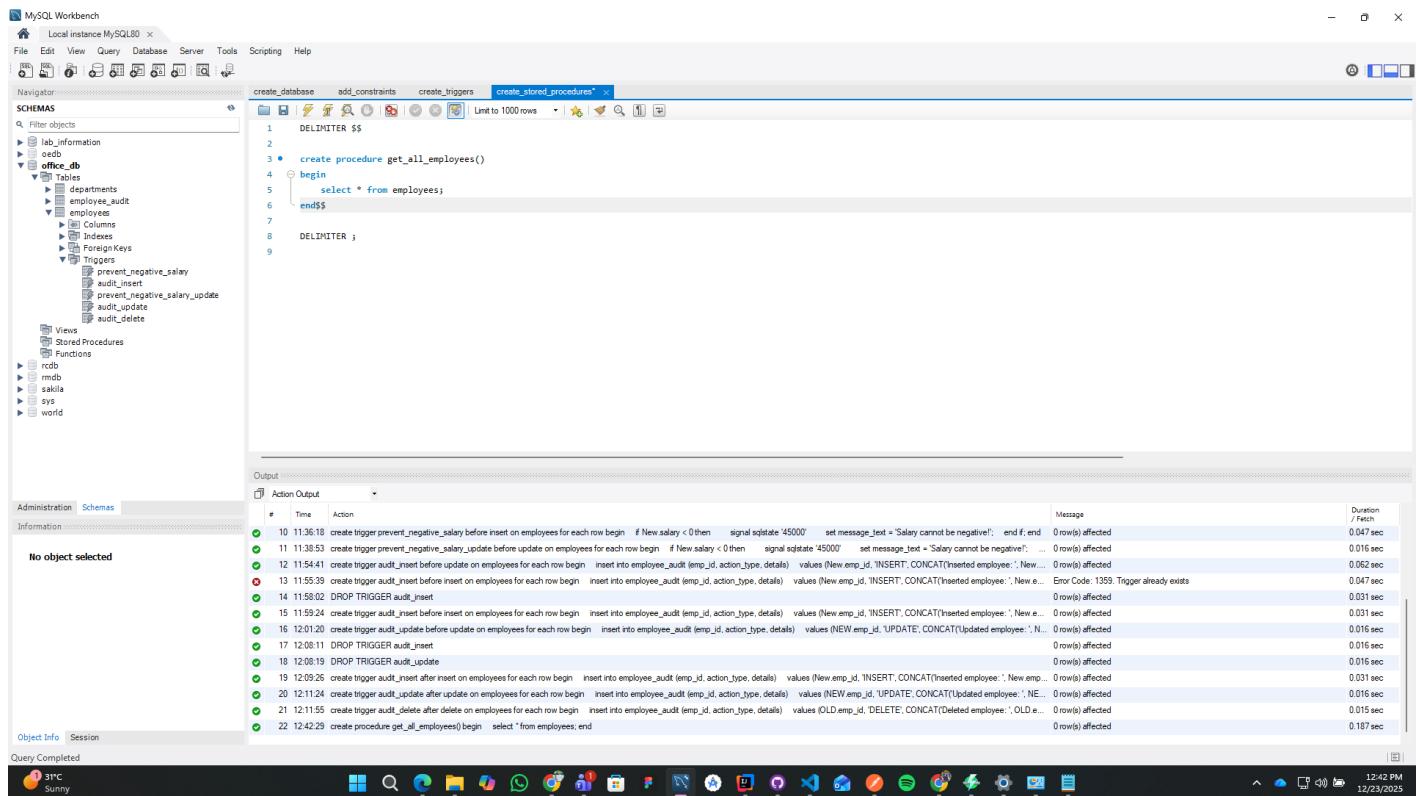
```
create procedure get_all_employees()
```

```
begin
```

```
    select * from employees;
```

```
end$$
```

```
DELIMITER ;
```



The screenshot shows the MySQL Workbench interface with the 'create_stored_procedures' tab selected. The code editor contains the following SQL script:

```
DELIMITER $$

create procedure get_all_employees()
begin
    select * from employees;
end$$

DELIMITER ;
```

The 'Schemas' tree on the left shows the 'lab_information' and 'office_db' databases. The 'office_db' database is expanded, showing its tables (departments, employee_audit, employees, locations, and indexes), foreign keys, and triggers (prevent_negative_salary, audit_insert, prevent_negative_salary_update, audit_update, audit_delete). The 'Output' pane at the bottom displays the execution log with 22 entries, showing the creation of triggers and the stored procedure. The status bar at the bottom right indicates the date and time as 12/23/2025 12:42 PM.

➤ Procedure to get an employee by ID (IN parameter).

- *Sql query:*

```
DELIMITER $$
```

```
create procedure get_employee_by_id(IN id INT)
begin
    select * from employees where emp_id = id;
end$$
```

```
DELIMITER ;
```

The screenshot shows the MySQL Workbench interface with the 'create_stored_procedures' tab selected. In the central pane, the SQL code for the stored procedure is displayed:

```
DELIMITER $$

3 • create procedure get_employee_by_id(IN id INT)
4 begin
5     select * from employees where emp_id = id;
6 end$$
7
8 DELIMITER ;
```

The 'Information' pane at the bottom shows the execution history of the session, including the creation of triggers and the stored procedure itself. The log entries are as follows:

Time	Action	Message	Duration
11:38:53	create trigger prevent_negative_salary_update before update on employees for each row begin if New.salary < 0 then signal sqlstate '45000' set message_text = 'Salary cannot be negative'; end if	0 rows(a) affected	0.016 sec
11:54:41	create trigger audit_insert before update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id, ' at ', NOW()))	0 rows(a) affected	0.062 sec
13:55:39	create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id, ' at ', NOW()))	Error Code: 1359. Trigger already exists	0.047 sec
14:11:58:02	DROP TRIGGER audit_insert	0 rows(a) affected	0.031 sec
15:15:24	create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id, ' at ', NOW()))	0 rows(a) affected	0.031 sec
16:12:01:20	create trigger audit_update before update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', NEW.emp_id, ' at ', NOW()))	0 rows(a) affected	0.016 sec
17:12:08:11	DROP TRIGGER audit_update	0 rows(a) affected	0.016 sec
18:12:08:19	DROP TRIGGER audit_update	0 rows(a) affected	0.016 sec
19:12:09:26	create trigger audit_insert after insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id, ' at ', NOW()))	0 rows(a) affected	0.031 sec
20:12:11:24	create trigger audit_update after update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', NEW.emp_id, ' at ', NOW()))	0 rows(a) affected	0.016 sec
21:12:11:55	create trigger audit_delete after delete on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ', OLD.emp_id, ' at ', NOW()))	0 rows(a) affected	0.015 sec
22:12:42:29	create procedure get_all_employees begin select * from employees; end	0 rows(a) affected	0.187 sec
23:12:50:23	create procedure get_employee_by_id(IN id INT) begin select * from employees where emp_id = id; end	0 rows(a) affected	0.016 sec

- Procedure to count employees by department (OUT parameter).

- *Sql query:*

DELIMITER \$\$

```
create procedure count_employees_by_dept(IN d_id INT, OUT cnt INT)
begin
    select COUNT(*) into cnt from employees where dept_id = d_id;
end$$
```

DELIMITER ;

The screenshot shows the MySQL Workbench interface with the 'create_stored_procedures' tab selected. The code pane contains the following SQL:

```
DELIMITER $$

create procedure count_employees_by_dept(IN d_id INT, OUT cnt INT)
begin
    select COUNT(*) into cnt from employees where dept_id = d_id;
end$$
```

The 'Output' pane shows the execution of this code and other related triggers. The 'Action Output' table lists the following events:

#	Time	Action	Message	Duration / Fetch
12	11:54:41	create trigger audit_insert before update on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id))	0 rows affected
13	11:55:39	create trigger audit_insert before insert on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id))	Error Code: 1359. Trigger already exists
14	11:58:02	DROP TRIGGER audit_insert		0 rows affected
15	11:59:24	create trigger audit_insert before insert on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id))	0 rows affected
16	12:01:20	create trigger audit_update before update on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', OLD.emp_id))	0 rows affected
17	12:08:11	DROP TRIGGER audit_insert		0 rows affected
18	12:08:11	DROP TRIGGER audit_update		0 rows affected
19	12:09:26	create trigger audit_insert after insert on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.emp_id))	0 rows affected
20	12:11:24	create trigger audit_update after update on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', OLD.emp_id))	0 rows affected
21	12:11:55	create trigger audit_delete after delete on employees for each row begin	insert into employee_audit (emp_id, action_type, details) values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ', OLD.emp_id))	0 rows affected
22	12:42:29	create procedure get_all_employees() begin select * from employees; end		0 rows affected
23	12:50:23	create procedure get_employee_by_id(IN id INT) begin select * from employees where emp_id = id; end		0 rows affected
24	12:52:01	create procedure count_employees_by_dept(IN d_id INT, OUT cnt INT) begin select COUNT(*) into cnt from employees where dept_id = d_id; end		0 rows affected

The status bar at the bottom right shows the date and time: 12/23/2025 12:52 PM.

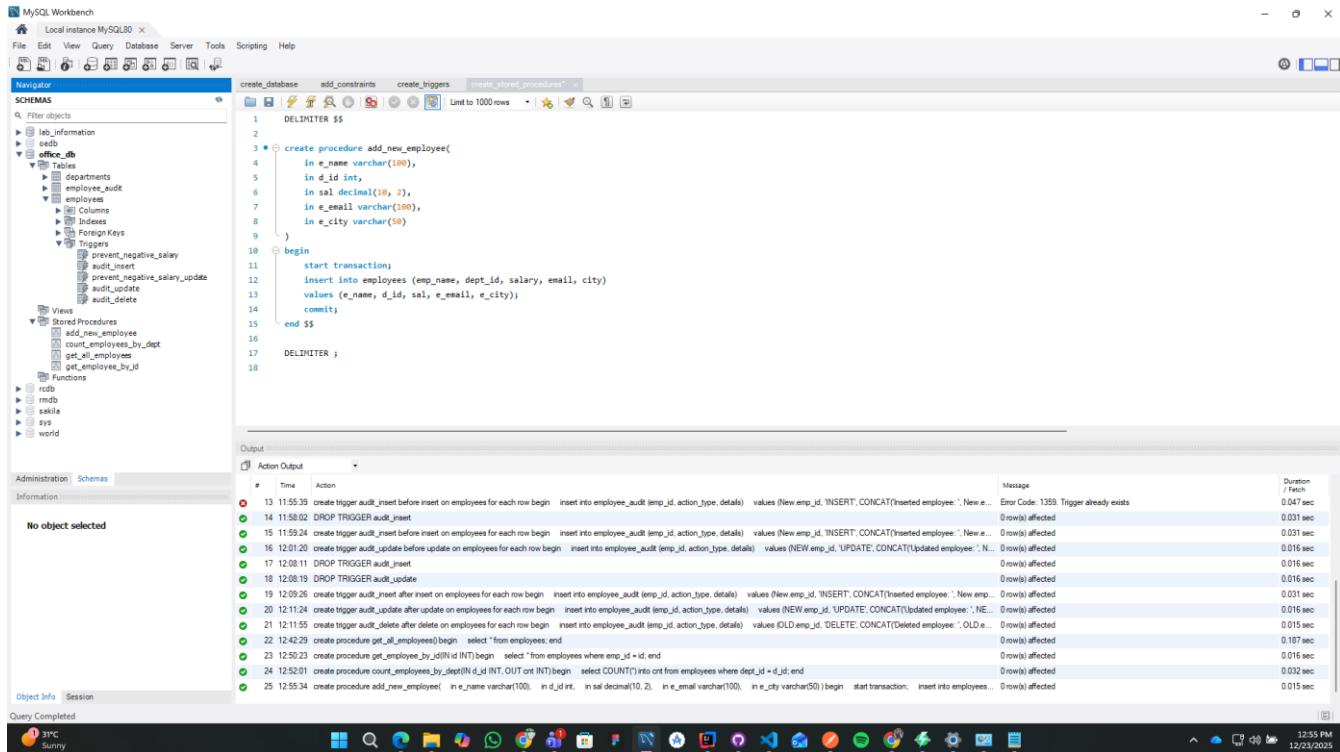
- Optional: procedure to add a new employee using a transaction that also logs the action.

- *Sql query:*

DELIMITER \$\$

```
create procedure add_new_employee(
    in e_name varchar(100),
    in d_id int,
    in sal decimal(10, 2),
    in e_email varchar(100),
    in e_city varchar(50)
)
begin
    start transaction;
    insert into employees (emp_name, dept_id, salary, email, city)
    values (e_name, d_id, sal, e_email, e_city);
    commit;
end $$
```

DELIMITER ;



The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. In the 'Navigator' pane, under the 'employees' schema, the 'Triggers' and 'Procedures' sections are visible. The 'Procedures' section contains a single procedure named 'add_new_employee'. The code for this procedure is displayed in the main editor area:

```
DELIMITER $$

create procedure add_new_employee(
    in e_name varchar(100),
    in d_id int,
    in sal decimal(10, 2),
    in e_email varchar(100),
    in e_city varchar(50)
)
begin
    start transaction;
    insert into employees (emp_name, dept_id, salary, email, city)
    values (e_name, d_id, sal, e_email, e_city);
    commit;
end $$
```

Below the code, the 'Action Output' pane shows the execution history of the procedure. The log entries are as follows:

#	Time	Action	Message	Duration
13	11:55:39	create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.e_name))	Error Code: 1359. Trigger already exists	0.047 sec
14	11:58:02	DROP TRIGGER audit_insert	0 rows(s) affected	0.031 sec
15	11:59:24	create trigger audit_insert before insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.e_name))	0 rows(s) affected	0.031 sec
16	12:01:20	create trigger audit_update before update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', New.e_name))	0 rows(s) affected	0.016 sec
17	12:08:11	DROP TRIGGER audit_insert	0 rows(s) affected	0.016 sec
18	12:08:19	DROP TRIGGER audit_update	0 rows(s) affected	0.016 sec
19	12:09:26	create trigger audit_insert after insert on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (New.emp_id, 'INSERT', CONCAT('Inserted employee: ', New.e_name))	0 rows(s) affected	0.031 sec
20	12:11:24	create trigger audit_update after update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', New.e_name))	0 rows(s) affected	0.016 sec
21	12:11:55	create trigger audit_delete after delete on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ', Old.e_name))	0 rows(s) affected	0.015 sec
22	12:42:29	create procedure get_all_employees begin select * from employees; end	0 rows(s) affected	0.187 sec
23	12:50:23	create procedure get_employee_by_id(IN id INT) begin select * from employees where emp_id = id; end	0 rows(s) affected	0.016 sec
24	12:52:01	create procedure count_employees_by_dept(IN d_id INT, OUT count INT) begin select COUNT(*) into count from employees where dept_id = d_id; end	0 rows(s) affected	0.032 sec
25	12:55:34	create procedure add_new_employee(in e_name varchar(100), in d_id int, in sal decimal(10, 2), in e_email varchar(100), in e_city varchar(50)) begin start transaction; insert into employees (emp_name, dept_id, salary, email, city) values (e_name, d_id, sal, e_email, e_city); commit; end	0 rows(s) affected	0.015 sec

Sql file:



5. Testing

- Insert sample employees and departments.

1. Insert departments.

- *Sql query:*

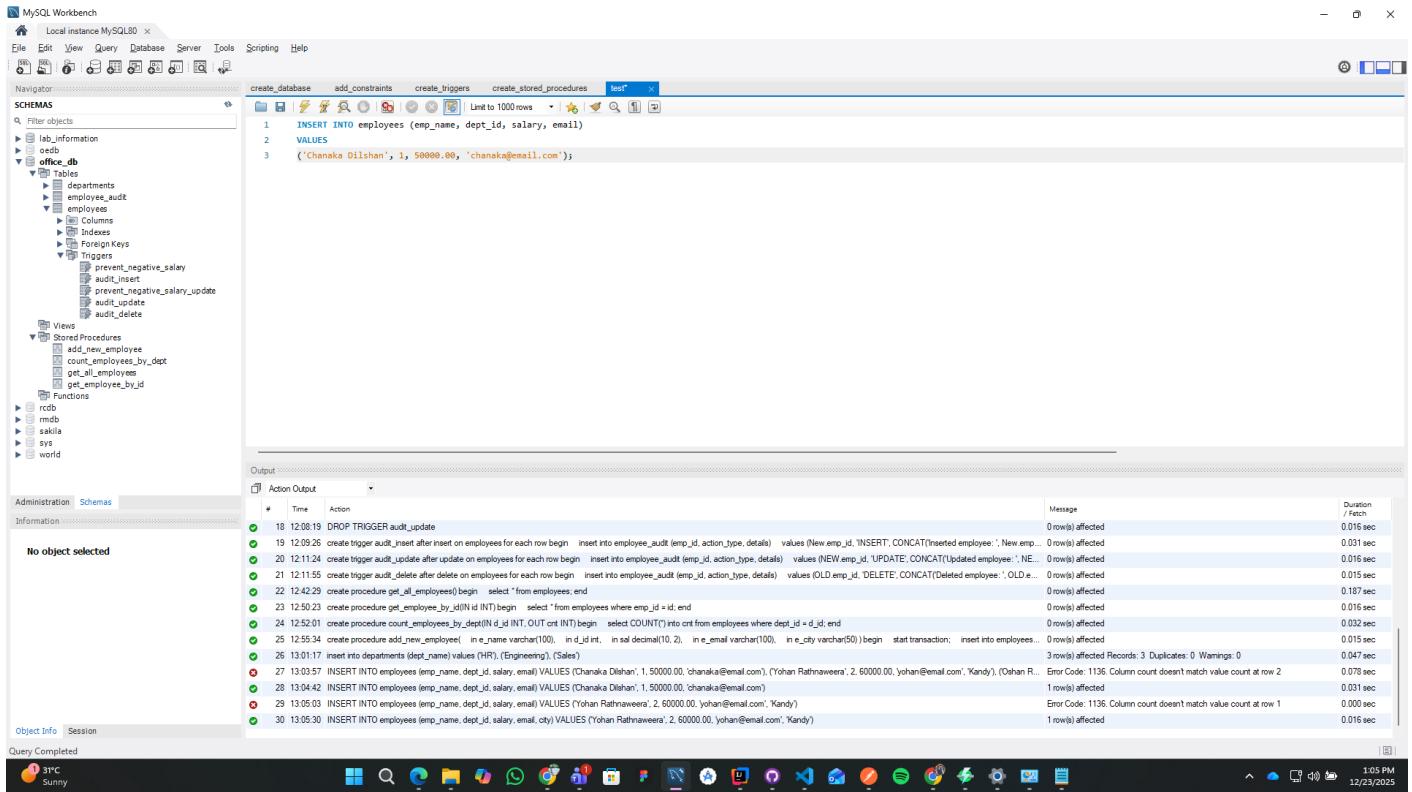
```
insert into departments (dept_name) values ('HR'), ('Engineering'), ('Sales');
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Navigator' with the 'test' schema selected. Under 'Tables', there are 'departments', 'employee_audit', and 'employees'. Under 'Triggers', there are 'prevent_negative_salary', 'audit_insert', 'audit_update', and 'audit_delete'. Under 'Stored Procedures', there are 'add_new_employee', 'count_all_employees_by_dept', 'get_all_employees', and 'get_employee_by_id'. The 'Output' tab shows the execution log with 26 entries, including the insertion of departments and the creation of various triggers and procedures. The status bar at the bottom shows the date and time as 12/23/2025 10:11 PM.

2. Insert employees.

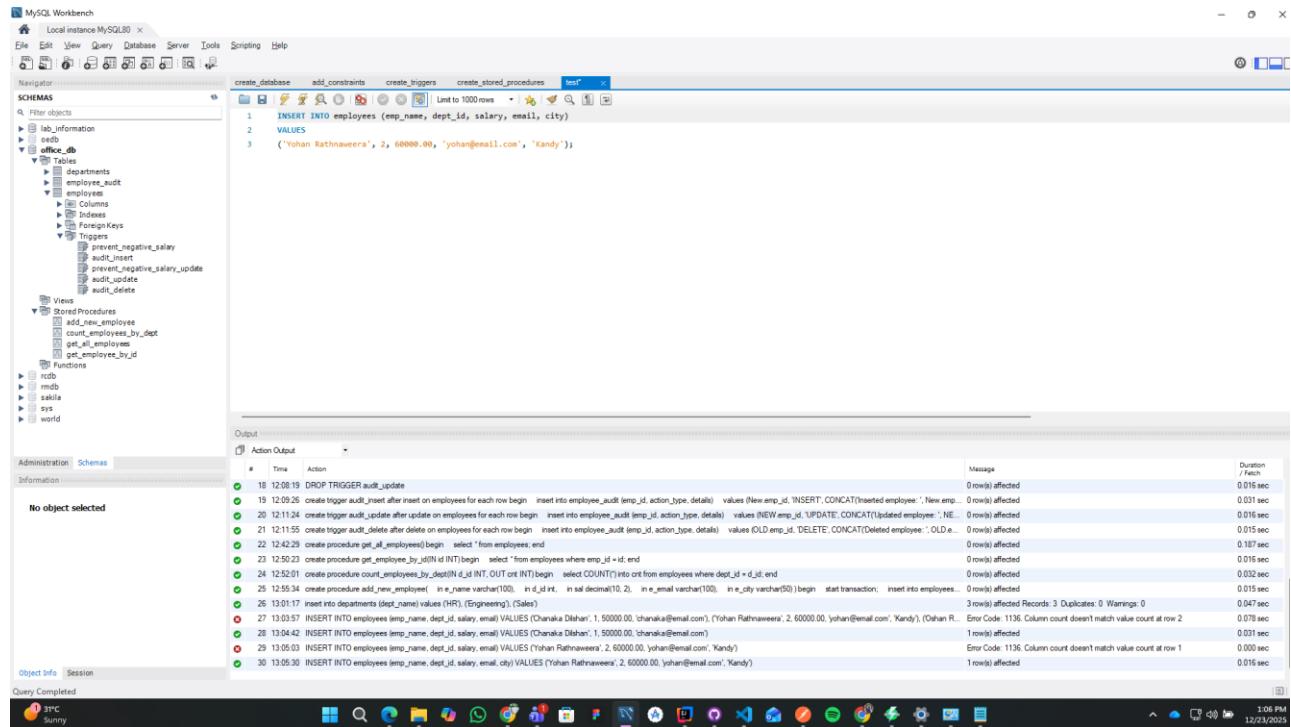
- *Sql query:*

```
INSERT INTO employees (emp_name, dept_id, salary, email)
VALUES
('Chanaka Dilshan', 1, 50000.00, 'chanaka@email.com');
```



- *Sql query:*

```
INSERT INTO employees (emp_name, dept_id, salary, email, city)
VALUES
('Yohan Rathnaweera', 2, 60000.00, 'yohan@email.com', 'Kandy');
```



MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- lab_information
- oedb
- office_db
- Views
- Tables
- Triggers
- Views
- Stored Procedures
- Functions
- rdm
- rmb
- sakila
- sys
- world

create_database add_constraints create_triggers create_stored_procedures test

1. INSERT INTO employees (emp_name, dept_id, salary, email, city)

2. VALUES

3. ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy');

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	12:08:19	CREATE TRIGGER audit_update	0 rows(a) affected	0.016 sec
19	12:09:20	CREATE TRIGGER audit_insert	0 rows(a) affected	0.015 sec
20	12:11:24	CREATE TRIGGER audit_update after update on employees for each row begin	0 rows(a) affected	0.016 sec
21	12:11:55	CREATE TRIGGER audit_delete after delete on employees for each row begin	0 rows(a) affected	0.015 sec
22	12:42:29	CREATE PROCEDURE get_all_employees begin select * from employees; end	0 rows(a) affected	0.187 sec
23	12:50:23	CREATE PROCEDURE get_employee_by_dept(d_id INT) begin select * from employees where emp_id = d_id; end	0 rows(a) affected	0.016 sec
24	12:52:01	CREATE PROCEDURE count_employee_by_dept(d_id INT, OUT cnt INT) begin select COUNT(*) into cnt from employees where dept_id = d_id; end	0 rows(a) affected	0.032 sec
25	12:55:34	CREATE PROCEDURE add_new_employee (i_name varchar(100), i_d_id int, i_salary decimal(10, 2), i_email varchar(50)) begin start transaction; insert into employees (emp_name, dept_id, salary, email) VALUES (i_name, i_d_id, i_salary, i_email); end	0 rows(a) affected	0.019 sec
26	13:01:17	INSERT INTO departments (dept_name) VALUES ('HR'), ('Engineering'), ('Sales')	3 rows(a) affected Records: 3 Duplicates: 0 Warnings: 0	0.047 sec
27	13:03:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankala Dihani', 1, 50000.00, 'chankala@email.com'), ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy'), ('Oshan Rajapaksha', 3, 55000.00, 'oshan@email.com')	Error Code: 1136. Column count doesn't match value count at row 2	0.078 sec
28	13:04:42	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankala Dihani', 1, 50000.00, 'chankala@email.com')	1 rows(a) affected	0.031 sec
29	13:05:03	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy')	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
30	13:05:30	INSERT INTO employees (emp_name, dept_id, salary, email, city) VALUES ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy')	1 rows(a) affected	0.016 sec
31	13:06:52	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Oshan Rajapaksha', 3, 55000.00, 'oshan@email.com')	1 rows(a) affected	0.006 sec

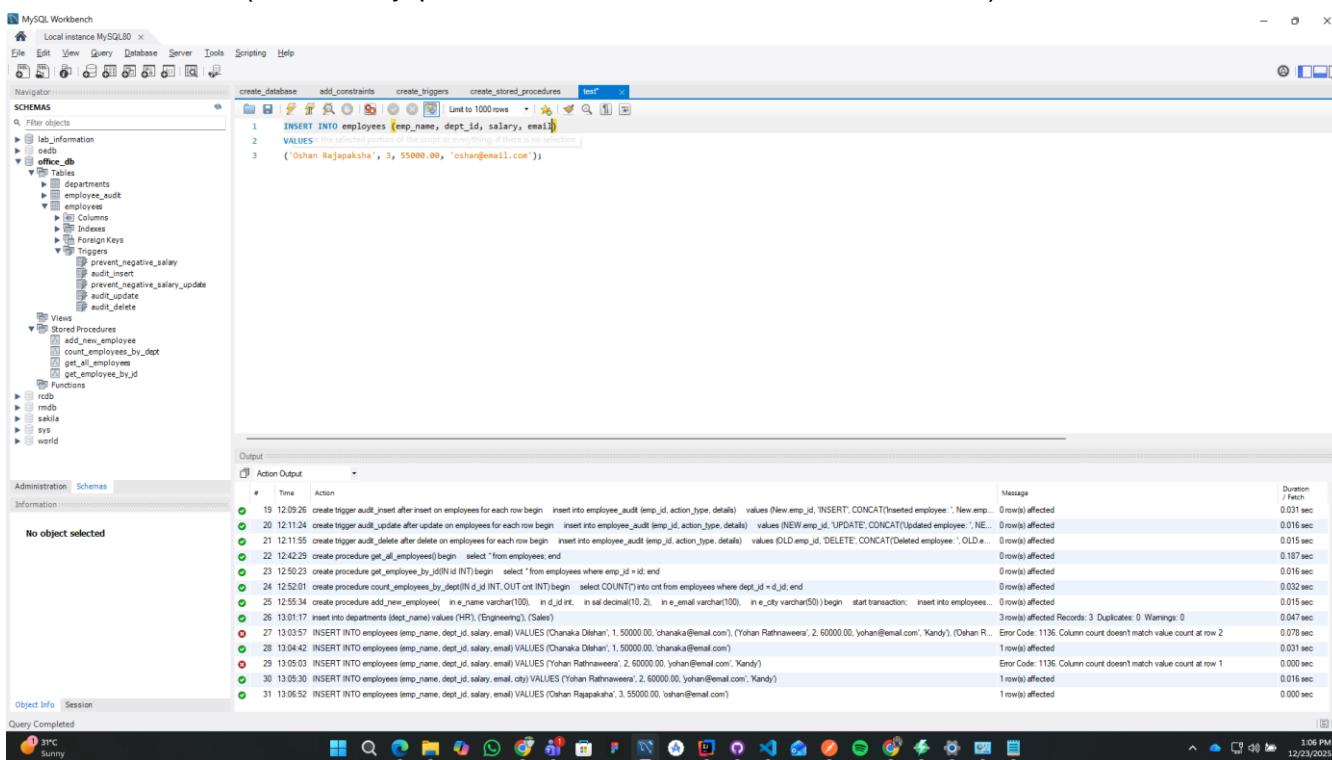
Object Info Session

Query Completed

3°C Sunny 10:04 PM 12/23/2025

• Sql query:

```
INSERT INTO employees (emp_name, dept_id, salary, email)
VALUES
('Oshan Rajapaksha', 3, 55000.00, 'oshan@email.com');
```



MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- lab_information
- oedb
- office_db
- Views
- Tables
- Triggers
- Views
- Stored Procedures
- Functions
- rdm
- rmb
- sakila
- sys
- world

create_database add_constraints create_triggers create_stored_procedures test

1. INSERT INTO employees (emp_name, dept_id, salary, email)

2. VALUES

3. ('Oshan Rajapaksha', 3, 55000.00, 'oshan@email.com'))

Output

Action Output

#	Time	Action	Message	Duration / Fetch
19	12:09:26	CREATE TRIGGER audit_insert after insert on employees for each row begin	0 rows(a) affected	0.031 sec
20	12:11:24	CREATE TRIGGER audit_update after update on employees for each row begin	0 rows(a) affected	0.016 sec
21	12:11:55	CREATE TRIGGER audit_delete after delete on employees for each row begin	0 rows(a) affected	0.015 sec
22	12:42:29	CREATE PROCEDURE get_all_employees begin select * from employees; end	0 rows(a) affected	0.187 sec
23	12:50:23	CREATE PROCEDURE get_employee_by_dept(d_id INT) begin select * from employees where emp_id = d_id; end	0 rows(a) affected	0.016 sec
24	12:52:01	CREATE PROCEDURE count_employee_by_dept(d_id INT, OUT cnt INT) begin select COUNT(*) into cnt from employees where dept_id = d_id; end	0 rows(a) affected	0.032 sec
25	12:55:34	CREATE PROCEDURE add_new_employee (i_name varchar(100), i_d_id int, i_salary decimal(10, 2), i_email varchar(50)) begin start transaction; insert into employees (emp_name, dept_id, salary, email) VALUES (i_name, i_d_id, i_salary, i_email); end	0 rows(a) affected	0.015 sec
26	13:01:17	INSERT INTO departments (dept_name) VALUES ('HR'), ('Engineering'), ('Sales')	3 rows(a) affected Records: 3 Duplicates: 0 Warnings: 0	0.047 sec
27	13:03:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankala Dihani', 1, 50000.00, 'chankala@email.com'), ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy'), ('Oshan Rajapaksha', 3, 55000.00, 'oshan@email.com')	Error Code: 1136. Column count doesn't match value count at row 2	0.078 sec
28	13:04:42	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankala Dihani', 1, 50000.00, 'chankala@email.com')	1 rows(a) affected	0.031 sec
29	13:05:03	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy')	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
30	13:05:30	INSERT INTO employees (emp_name, dept_id, salary, email, city) VALUES ('Yohan Rathnaveera', 2, 60000.00, 'yohan@email.com', 'Kandy')	1 rows(a) affected	0.016 sec
31	13:06:52	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Oshan Rajapaksha', 3, 55000.00, 'oshan@email.com')	1 rows(a) affected	0.006 sec

Object Info Session

Query Completed

3°C Sunny 10:06 PM 12/23/2025

3. Insert negative salary.

- *Sql query:*

```
INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Bob Negative', 1, -1000.00, 'bob@example.com');
```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the database schema, including the 'employees' table and its associated triggers: prevent_negative_salary, audit_insert, audit_update, and audit_delete. The top-right pane shows the SQL editor with the query: `INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Bob Negative', 1, -1000.00, 'bob@example.com');`. The bottom-right pane, titled 'Output', displays the log of events. The log shows the creation of triggers and the execution of the INSERT query, which fails due to a negative salary value. The log entries are as follows:

#	Time	Action	Message	Duration /Fetch
20	12:11:24	create trigger audit_update after update on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (NEW.emp_id, 'UPDATE', CONCAT('Updated employee: ', NEW.emp_name, ' to ', NEW.salary, ' at ', NEW.updated_at)); end	0 row(s) affected	0.01 sec
21	12:11:55	create trigger audit_delete after delete on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ', OLD.emp_name, ' at ', OLD.updated_at)); end	0 row(s) affected	0.015 sec
22	12:42:29	create procedure get_all_employees() begin select * from employees; end	0 row(s) affected	0.187 sec
23	12:50:23	create procedure get_employee_by_id(IN id INT) begin select * from employees where emp_id = id; end	0 row(s) affected	0.016 sec
24	12:52:01	create procedure count_employees_by_dept(IN d_id INT, OUT cnt INT) begin select COUNT(*) into cnt from employees where dept_id = d_id; end	0 row(s) affected	0.032 sec
25	12:55:34	create procedure add_new_employee(IN e_name varchar(100), IN d_id int, IN sal decimal(10, 2), IN e_email varchar(50)) begin start transaction; insert into employees (emp_name, dept_id, salary, email) values (e_name, d_id, sal, e_email); end	0 row(s) affected	0.015 sec
26	13:01:17	insert into departments (dept_name) values ('HR'), ('Engineering'), ('Sales')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.047 sec
27	13:03:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chanaka Dhalan', 1, 50000.00, 'chanaka@email.com'), ('Yohan Rathnaweera', 2, 60000.00, 'yohan@email.com', 'Kandy'), ('Orhan Rajapaksha', 3, 55000.00, 'orhan@email.com')	Error Code: 1136. Column count doesn't match value count at row 2	0.076 sec
28	13:04:42	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chanaka Dhalan', 1, 50000.00, 'chanaka@email.com')	1 row(s) affected	0.031 sec
29	13:05:03	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnaweera', 2, 60000.00, 'yohan@email.com', 'Kandy')	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
30	13:05:30	INSERT INTO employees (emp_name, dept_id, salary, email, city) VALUES ('Yohan Rathnaweera', 2, 60000.00, 'yohan@email.com', 'Kandy')	1 row(s) affected	0.016 sec
31	13:06:52	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Orhan Rajapaksha', 3, 55000.00, 'orhan@email.com')	1 row(s) affected	0.000 sec
32	13:08:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Bob Negative', 1, -1000.00, 'bob@example.com')	Error Code: 1644. Salary cannot be negative!	0.000 sec

4. Update employees

- *Sql query:*

```
UPDATE employees SET salary = 65000.00 WHERE emp_id = 1;
```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS

Filter objects

lab_information

edb

office_db

- Tables
 - departments
 - employees
 - employees_audit
 - Views
 - Stored Procedures
 - Functions
 - rdb
 - imdb
 - sakila
 - sys
 - world

create_database add_constraints create_triggers create_stored_procedures test

1 • UPDATE employees SET salary = 65000.00 WHERE emp_id = 3;

Execute the selected portion of the script or everything, if there is no selection. []

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	21 12:11:55	create trigger audit_delete after delete on employees for each row begin insert into employee_audit (emp_id, action_type, details) values (OLD.emp_id, 'DELETE', CONCAT('Deleted employee: ', OLD.emp_name)) end	0 row(s) affected	0.015 sec
2	22 12:42:29	create procedure get_all_employees() begin select * from employees end	0 row(s) affected	0.187 sec
3	23 12:50:23	create procedure get_employees_by_id(IN id INT) begin select * from employees where emp_id = id end	0 row(s) affected	0.016 sec
4	24 12:52:01	create procedure count_employees_by_dept(IN d_id INT, OUT crt INT) begin select COUNT(*) into crt from employees where dept_id = d_id; end	0 row(s) affected	0.032 sec
5	25 12:55:34	create procedure add_new_employee(IN e_name varchar(100), IN d_id INT, IN e_email varchar(50)) begin start transaction; insert into employees (emp_name, dept_id, e_email) values (e_name, d_id, e_email); end	0 row(s) affected	0.015 sec
6	26 13:01:17	insert into departments (dept_name) values ('H.R.', 'Engineering', 'Sales')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.047 sec
7	27 13:03:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankaa Dihan', 1, 50000.00, 'chankaa@email.com'), ('Yohan Rathnawera', 2, 60000.00, 'yohan@email.com', 'Kandy'), ('Ohan Rajapaksha', 3, 55000.00, 'ohan@email.com')	Error Code: 1136. Column count doesn't match value count at row 2	0.078 sec
8	28 13:04:42	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankaa Dihan', 1, 50000.00, 'chankaa@email.com')	1 row(s) affected	0.031 sec
9	29 13:05:03	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnawera', 2, 60000.00, 'yohan@email.com', 'Kandy')	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
10	30 13:05:30	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnawera', 2, 60000.00, 'yohan@email.com', 'Kandy')	1 row(s) affected	0.016 sec
11	31 13:06:52	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Ohan Rajapaksha', 3, 55000.00, 'ohan@email.com')	1 row(s) affected	0.000 sec
12	32 13:08:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Bob Negative', 1, -1000.00, 'bob@example.com')	Error Code: 1644. Salary cannot be negative!	0.000 sec
13	33 13:09:29	SELECT * FROM office_db.employee_audit LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
14	34 13:11:36	UPDATE employees SET salary = 65000.00 WHERE emp_id = 3;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec

Object Info Session

Query Completed

3°C Sunny 11:12 PM 12/23/2025

5. Delete employee.

- Sql query:

DELETE FROM employees WHERE emp_id = 3;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS

Filter objects

lab_information

edb

office_db

- Tables
 - departments
 - employees
 - employees_audit
 - Views
 - Stored Procedures
 - Functions
 - rdb
 - imdb
 - sakila
 - sys
 - world

create_database add_constraints create_triggers create_stored_procedures test

1 • DELETE FROM employees WHERE emp_id = 3;

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	22 12:42:29	create procedure get_all_employees() begin select * from employees end	0 row(s) affected	0.187 sec
2	23 12:50:23	create procedure get_employees_by_id(IN id INT) begin select * from employees where emp_id = id end	0 row(s) affected	0.016 sec
3	24 12:52:01	create procedure count_employees_by_dept(IN d_id INT, OUT crt INT) begin select COUNT(*) into crt from employees where dept_id = d_id; end	0 row(s) affected	0.032 sec
4	25 12:55:34	create procedure add_new_employee(IN e_name varchar(100), IN d_id INT, IN e_email varchar(50)) begin start transaction; insert into employees (emp_name, dept_id, e_email) values (e_name, d_id, e_email); end	0 row(s) affected	0.015 sec
5	26 13:01:17	insert into departments (dept_name) values ('H.R.', 'Engineering', 'Sales')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.047 sec
6	27 13:03:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankaa Dihan', 1, 50000.00, 'chankaa@email.com'), ('Yohan Rathnawera', 2, 60000.00, 'yohan@email.com', 'Kandy'), ('Ohan Rajapaksha', 3, 55000.00, 'ohan@email.com')	Error Code: 1136. Column count doesn't match value count at row 2	0.078 sec
7	28 13:04:42	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chankaa Dihan', 1, 50000.00, 'chankaa@email.com')	1 row(s) affected	0.031 sec
8	29 13:05:03	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnawera', 2, 60000.00, 'yohan@email.com', 'Kandy')	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
9	30 13:05:30	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Yohan Rathnawera', 2, 60000.00, 'yohan@email.com', 'Kandy')	1 row(s) affected	0.016 sec
10	31 13:06:52	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Ohan Rajapaksha', 3, 55000.00, 'ohan@email.com')	1 row(s) affected	0.000 sec
11	32 13:08:57	INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Bob Negative', 1, -1000.00, 'bob@example.com')	Error Code: 1644. Salary cannot be negative!	0.000 sec
12	33 13:09:29	SELECT * FROM office_db.employee_audit LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
13	34 13:11:36	UPDATE employees SET salary = 65000.00 WHERE emp_id = 3;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
14	35 13:12:33	DELETE FROM employees WHERE emp_id = 3;	1 row(s) affected	0.016 sec

Object Info Session

Query Completed

3°C Sunny 11:12 PM 12/23/2025

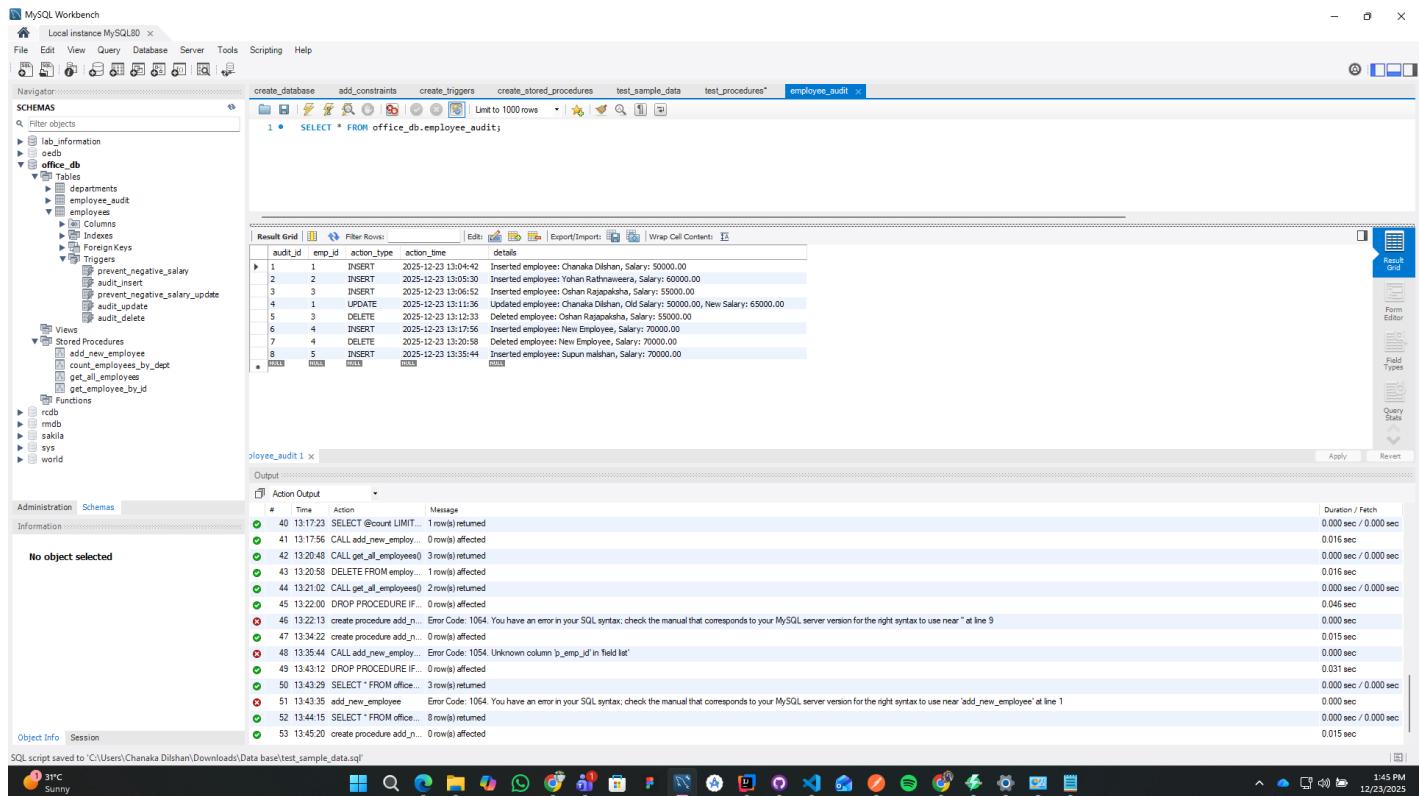
Sql file:



test_sample_data.sql

➤ Test Triggers

All changes made to the employees table are recorded in the employee_audit table by triggers.



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure for the `office_db` schema, including tables like `departments`, `employees`, and `employee_audit`, and triggers like `audit_insert`, `audit_update`, and `audit_delete`.
- Query Editor:** Displays the SQL query: `SELECT * FROM office_db.employee_audit;` and the resulting data in the `employee_audit` table.
- Results:** The `employee_audit` table has 8 rows of data:

audit_id	emp_id	action_type	action_time	details
1	1	INSERT	2025-12-23 13:04:42	Inserted employee: Chanaka Dilshan, Salary: 50000.00
2	2	INSERT	2025-12-23 13:05:30	Inserted employee: Yohan Rathnaweera, Salary: 60000.00
3	3	UPDATE	2025-12-23 13:11:00	Updated employee: Chanaka Dilshan, Old Salary: 50000.00, New Salary: 65000.00
4	1	UPDATE	2025-12-23 13:11:55	Updated employee: Chanaka Dilshan, Old Salary: 65000.00, New Salary: 55000.00
5	3	DELETE	2025-12-23 13:12:33	Deleted employee: Cohan Raspoleksa, Salary: 55000.00
6	4	INSERT	2025-12-23 13:17:56	Inserted employee: New Employee, Salary: 70000.00
7	4	DELETE	2025-12-23 13:20:58	Deleted employee: New Employee, Salary: 70000.00
8	5	INSERT	2025-12-23 13:35:44	Inserted employee: Supun malshan, Salary: 70000.00
- Output:** Shows the log of actions with their times, messages, and durations.
- Object Info:** Shows the SQL script saved to the file: `C:\Users\Chanaka Dilshan\Downloads\Data base\test_sample_data.sql`.

➤ Test Procedure

1. Get all employees.

- *Sql query:*

```
CALL get_all_employees();
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (office_db selected), Tables (departments, employee_audit, employees, triggers), Views, Stored Procedures (get_all_employees, count_employee_by_dept, get_all_employees, get_employee_by_id), Functions.
- Test Procedures Tab:** Shows the query `CALL get_all_employees();` and its execution results.
- Result Grid:**

emp_id	emp_name	dept_id	salary	email	city
1	Chenaka Dilshan	1	65000.00	chenaka@email.com	Colombo
2	Yohan Rathnaveera	2	60000.00	yohan@email.com	Kandy
- Action Output:** Shows the history of database actions with details like time, action, message, duration, and fetch.
- System Bar:** Shows the date (12/23/2025), time (1:15 PM), and weather (31°C, Sunny).

2. Get employee by ID.

- *Sql query:*

```
CALL get_employee_by_id(1);
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Filter objects, SCHEMAS (lab_information, oedb, office_db), Tables (departments, employee_audit, employees, Foreign Keys, Triggers, Views, Stored Procedures, Functions), rcdb, rmdb, sakila, sys, world.
- Query Editor:** Test_procedures* tab, SQL code: `1 CALL get_employee_by_id(1);`, Result Grid showing 1 row: emp_id 1, emp_name Chanaka Dushman, dept_id 1, salary 65000.00, email chanaka@email.com, city Colombo.
- Output:** Action Output table showing the history of 37 database operations, including procedure executions and data insertions.
- System Bar:** Shows the date and time (11:16 PM 12/23/2025) and system icons.

3. Count employees by department.

- *Sql query:*

```
SET @count = 0;
CALL count_employees_by_dept(2, @count);
SELECT @count;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure for 'lab_information' including SCHEMAS, TABLES, and FUNCTIONS.
- Query Editor:** Contains the SQL code:

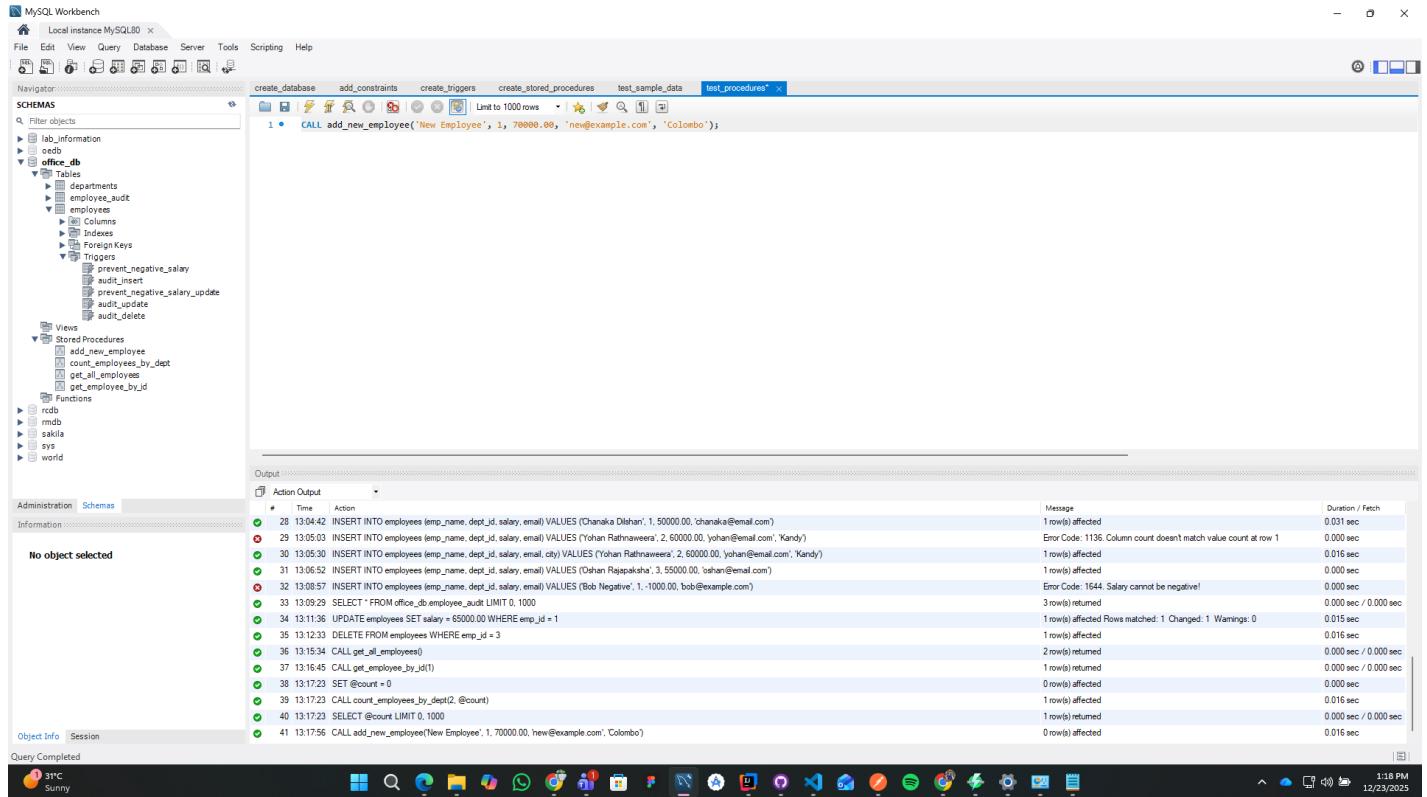
```
1 SET @count = 0;
2 CALL count_employees_by_dept(2, @count);
3 SELECT @count;
```
- Result Grid:** Displays the result of the query:

Result Grid	1
@count	1
- Output:** Shows the execution log with 40 entries, including:
 - Action: INSERT INTO employees (emp_name, dept_id, salary, email) VALUES ('Chanaka Dihan', 1, 50000.00, 'chanaka@email.com'), ('Yohan Rathnaweera', 2, 60000.00, 'yohan@email.com', 'Kandy')
 - Action: CALL count_employees_by_dept(2, @count);
 - Action: SELECT @count;
- Session:** Shows the session environment with the following details:
 - Object Info: Session
 - Query Completed: 1 row(s) returned, 0 rows(s) affected, 0 rows(s) matched, 1 row(s) changed, 0 warnings, 0.079 sec
 - System Information: 31°C, Sunny
 - System Icons: Taskbar icons for various applications like File Explorer, Edge, and File Explorer.

4. Add new employee

- *Sql query:*

```
CALL add_new_employee('Supun malshan', 1, 70000.00,
'supun@email.com', 'Colombo');
```

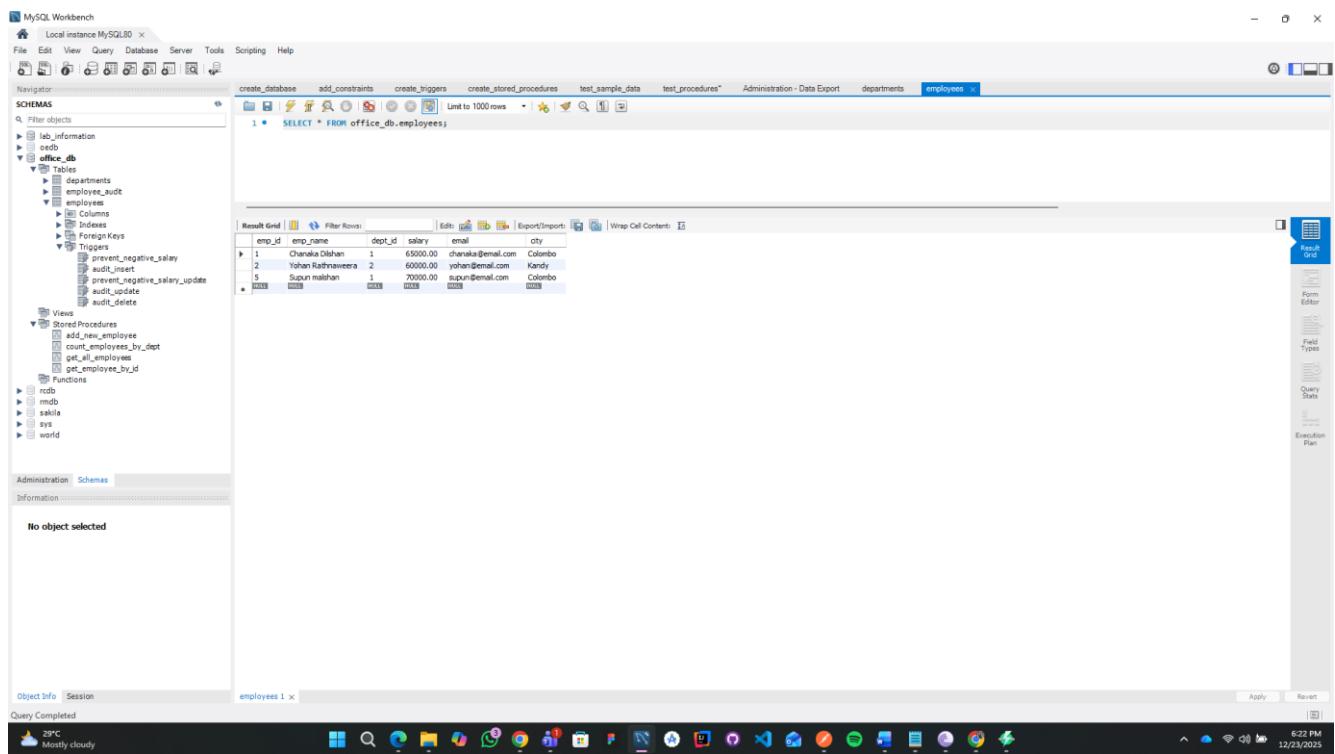


Sql file:



test_procedures.sql

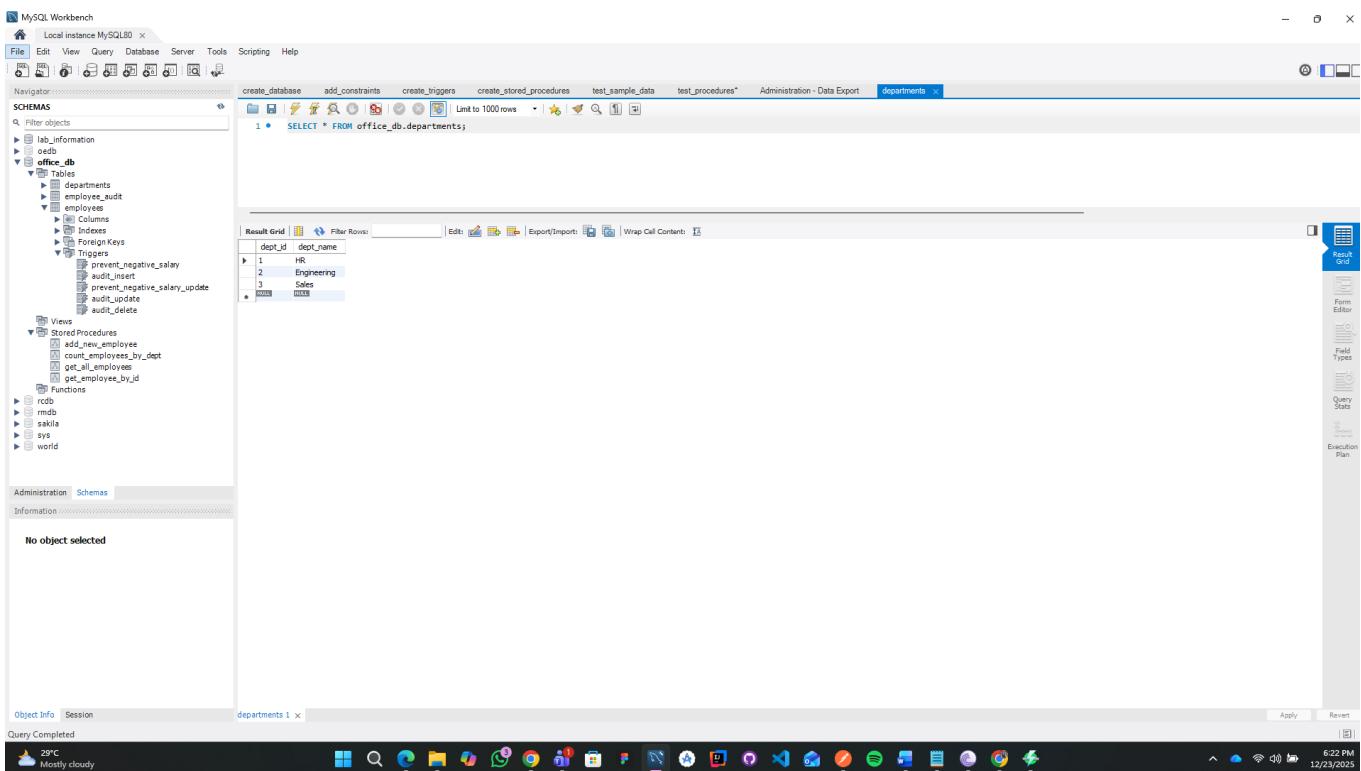
Employee table



The screenshot shows the MySQL Workbench interface with the 'employees' table selected in the 'office_db' schema. The table structure includes columns: emp_id, emp_name, dept_id, salary, email, and city. The data grid shows the following rows:

emp_id	emp_name	dept_id	salary	email	city
1	Chenaka Dshan	1	50000.00	chenaka@email.com	Colombo
2	Yohan Rathnayaka	2	60000.00	yohan@email.com	Kandy
5	Supun Malshan	1	70000.00	supun@email.com	Colombo
6	Samitha	1	65000.00	Samitha@email.com	Colombo

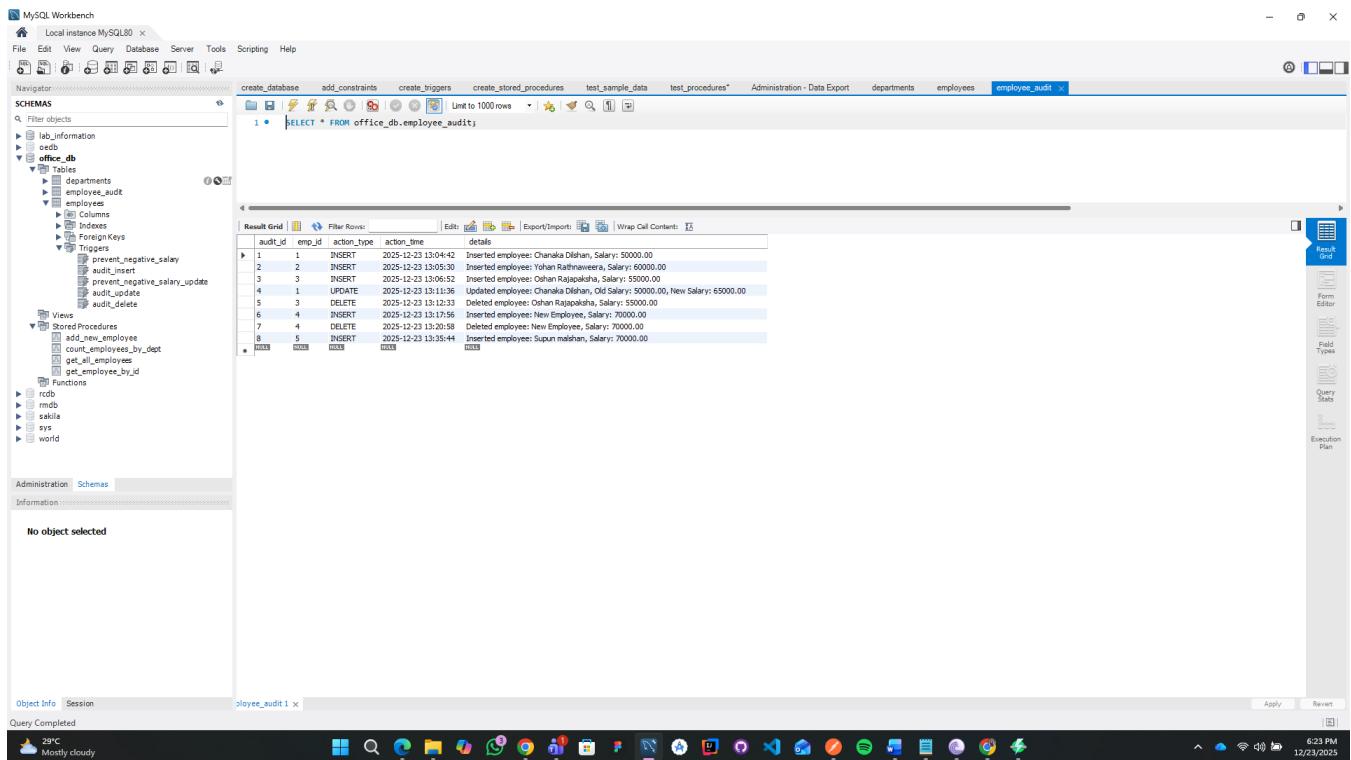
Department table



The screenshot shows the MySQL Workbench interface with the 'departments' table selected in the 'office_db' schema. The table structure includes columns: dept_id and dept_name. The data grid shows the following rows:

dept_id	dept_name
1	HR
2	Engineering
3	Sales
4	Marketing

Employee_audit table



The screenshot shows the MySQL Workbench interface with the 'employee_audit' table selected in the 'Schemas' tree. A query window displays the following data:

audit_id	emp_id	action_type	action_time	details
1	1	INSERT	2025-12-23 13:04:42	Inserted employee: Chanaka Dihshan, Salary: 50000.00
2	2	INSERT	2025-12-23 13:05:30	Inserted employee: Yohan Rathnaweera, Salary: 60000.00
3	3	INSERT	2025-12-23 13:06:52	Inserted employee: Oshan Rajapaksha, Salary: 55000.00
4	1	UPDATE	2025-12-23 13:11:36	Updated employee: Chanaka Dihshan, Old Salary: 50000.00, New Salary: 65000.00
5	3	DELETE	2025-12-23 13:17:56	Deleted employee: Oshan Rajapaksha, Salary: 55000.00
6	4	INSERT	2025-12-23 13:20:58	Inserted employee: New Employee, Salary: 70000.00
7	4	DELETE	2025-12-23 13:20:58	Deleted employee: New Employee, Salary: 70000.00
8	5	INSERT	2025-12-23 13:35:44	Inserted employee: Supun Madhura, Salary: 70000.00

Database file:



6. Database Support for API Endpoints

This SQL database setup is designed to directly support REST API endpoints such as **GET**, **POST**, **PUT**, and **DELETE** operations for an Employee Management System by using structured tables, constraints, triggers, and stored procedures to ensure data integrity, validation, and auditing.

- **GET: api/employee/get-all**

Supported by the get_all_employees stored procedure, which retrieves all employee records from the employees table efficiently.

- **GET: api/employee/get-by-id/{id}**

Managed by the get_employee_by_id stored procedure using an **IN parameter** to fetch details of a specific employee by ID.

- **POST: api/employee/add**

Implemented through the add_new_employee stored procedure, which inserts new employee records within a transaction. Triggers prevent negative salary values and log insert actions into the employee_audit table.

- **PUT: api/employee/update/{id}**

Supports updating employee information in the employees table. Database constraints validate the data, and update triggers automatically record changes in the employee_audit table.

- **DELETE: api/employee/delete/{id}**

Enables deletion of employee records from the employees table. Delete triggers log the action and related details into the employee_audit table for auditing.

- **GET: api/employee/count-by-department/{deptId}**

Supported by the count_employees_by_dept stored procedure using **IN and OUT parameters**, providing employee count summaries per department.

Git hub link - <https://github.com/chanaka-dill/ADS-Assignment-1-SID-11366.git>

Postman link - <https://universal-desert-189598.postman.co/workspace/ADS-Assignment-01~f8183609-0c81-47bf-9798-cd5189958f6a/collection/33855737-225ad55e-212e-42d7-95e7-857d4039f59c?action=share&creator=33855737>