Module 4 – Introduction to DBMS (Lab Exercises)

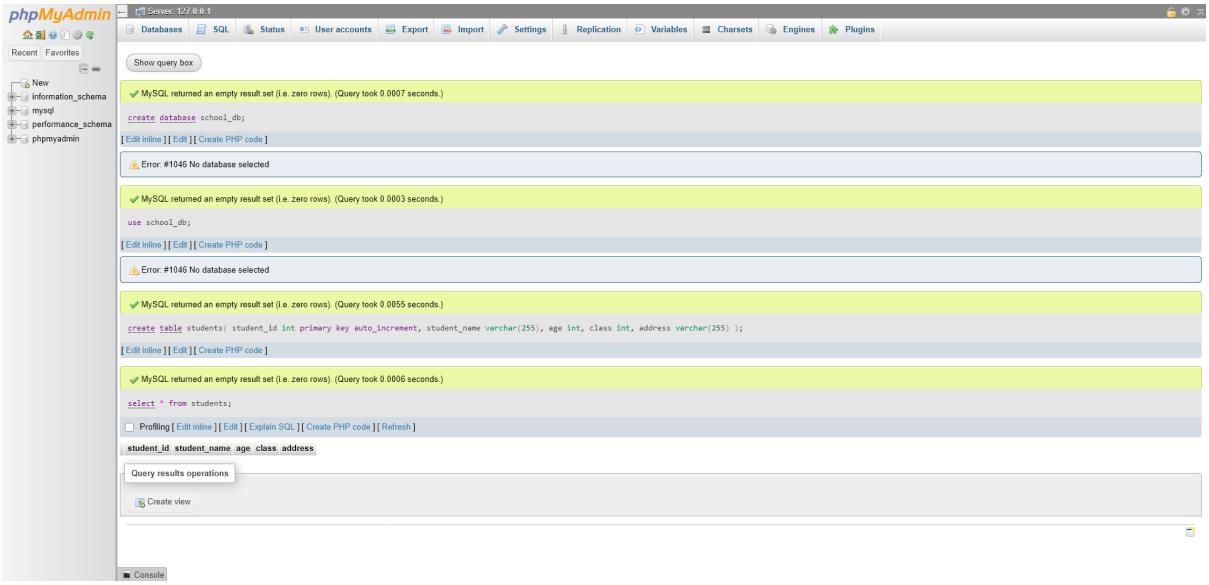
Shivam Gangwani

June 27, 2025

1 Introduction to SQL

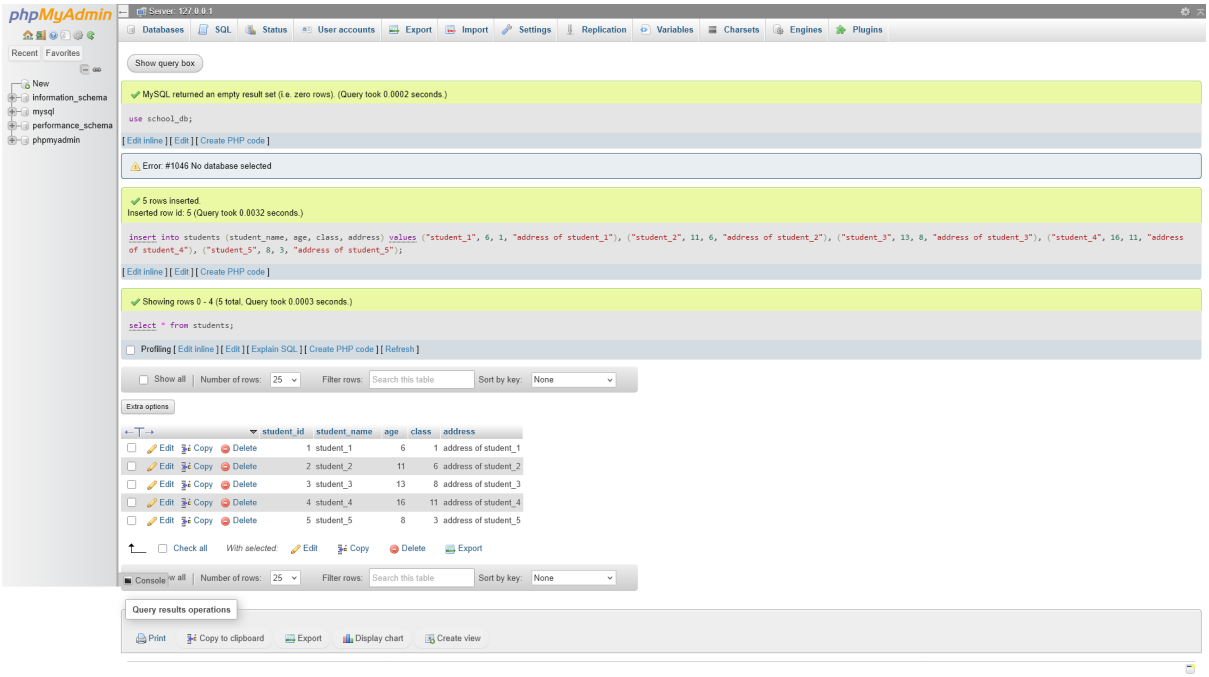
1.1 Create a new database named **school_db** and a table called **students** with the following columns: **student_id**, **student_name**, **age**, **class**, and **address**.

```sql
create database school_db;
use school_db;

create table students(
    student_id int primary key auto_increment,
    student_name varchar(255),
    age int,
    class int,
    address varchar(255)
);

select *
from students;
```



1.2 Insert five records into the **students** table and retrieve all records using the **SELECT** statement.

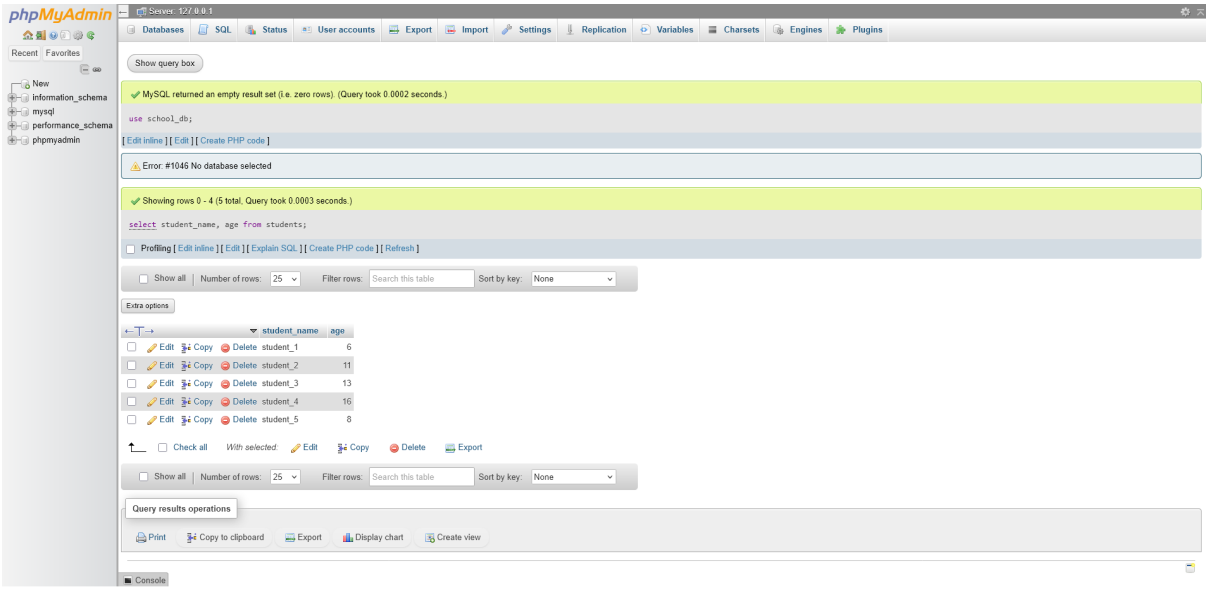
```sql
use school_db;

insert into students (student_name, age, class, address) values
    ("student_1", 6, 1, "address of student_1"),
    ("student_2", 11, 6, "address of student_2"),
    ("student_3", 13, 8, "address of student_3"),
    ("student_4", 16, 11, "address of student_4"),
    ("student_5", 8, 3, "address of student_5");

select *
from students;
```



2 SQL Syntax

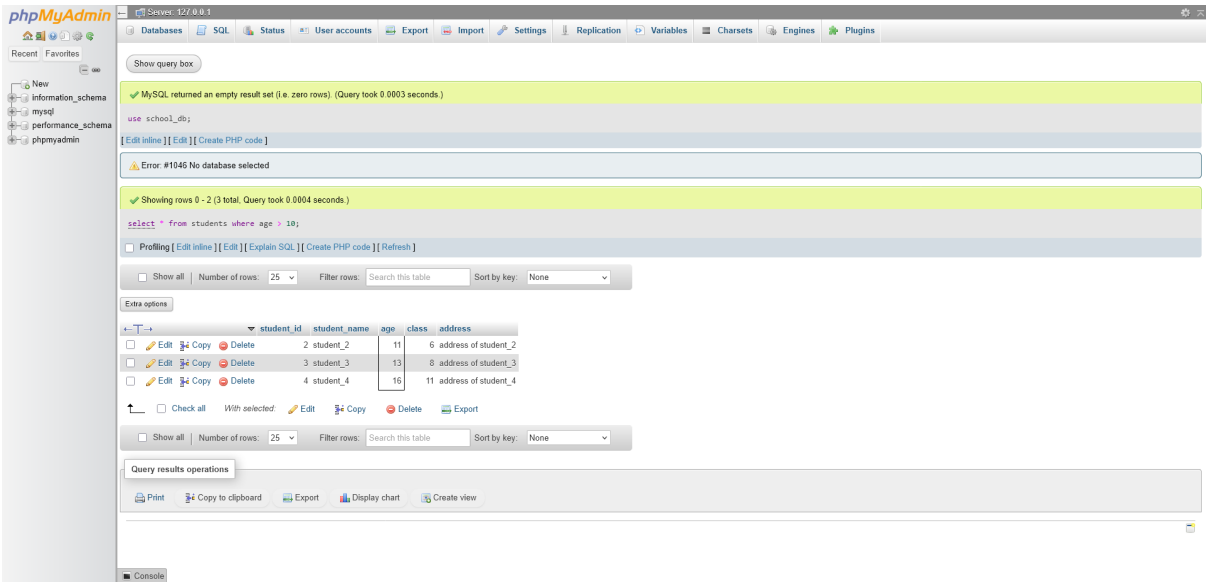
2.1 Write SQL queries to retrieve specific columns (**student_name** and **age**) from the **students** table.

```sql
use school_db;

select student_name, age
from students;
```



2.2 Write SQL queries to retrieve all students whose age is greater than 10.

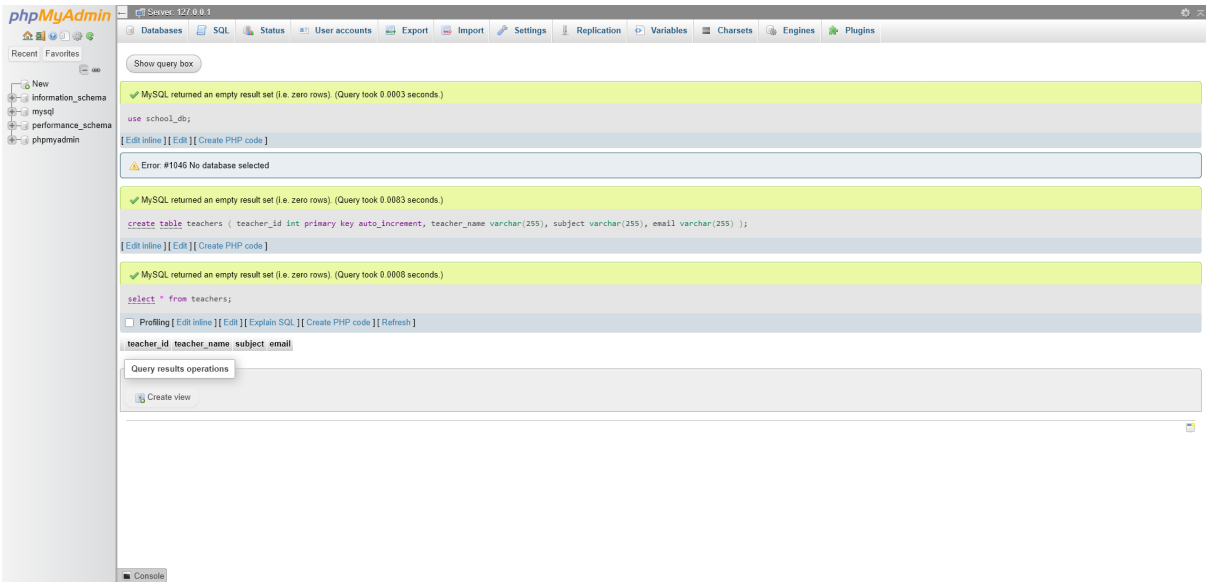
```sql
use school_db;

select *
from students
where age > 10;
```



3 SQL Constraints

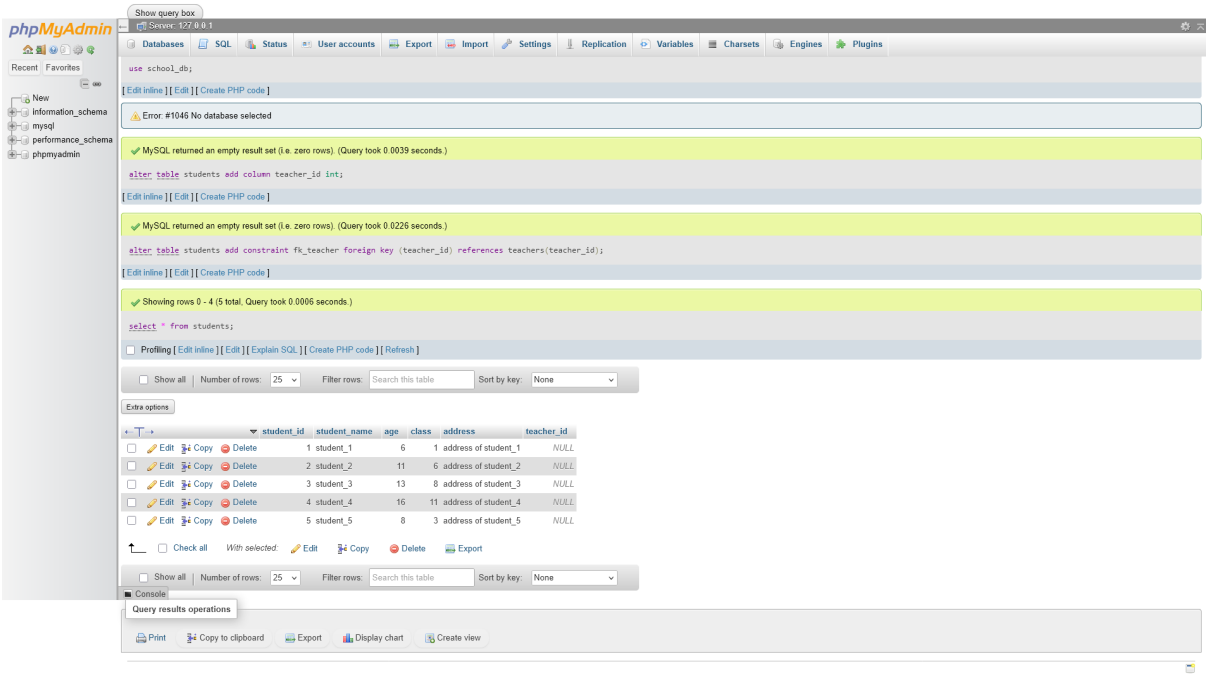
3.1 Create a table **teachers** with the following columns: **teacher_id** (Primary Key), **teacher_name** (NOT NULL), **subject** (NOT NULL), and **email** (UNIQUE).

```sql
use school_db;

create table teachers (
    teacher_id int primary key auto_increment,
    teacher_name varchar(255),
    subject varchar(255),
    email varchar(255)
);

select *
from teachers;
```



3.2 Implement a **FOREIGN KEY** constraint to relate the **teacher_id** from the **teachers** table with the **students** table.

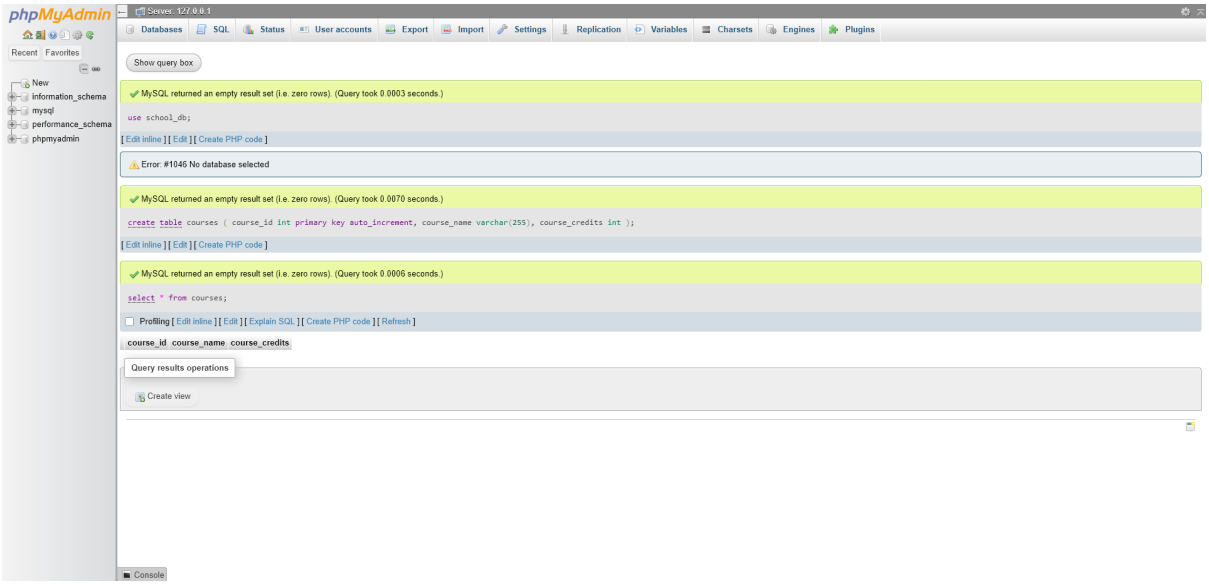
```sql
use school_db;

alter table students
add column teacher_id int;

alter table students
add constraint fk_teacher
foreign key (teacher_id) references teachers(teacher_id);

select *
from students;
```



4 Main SQL Commands and Sub-commands (DDL)

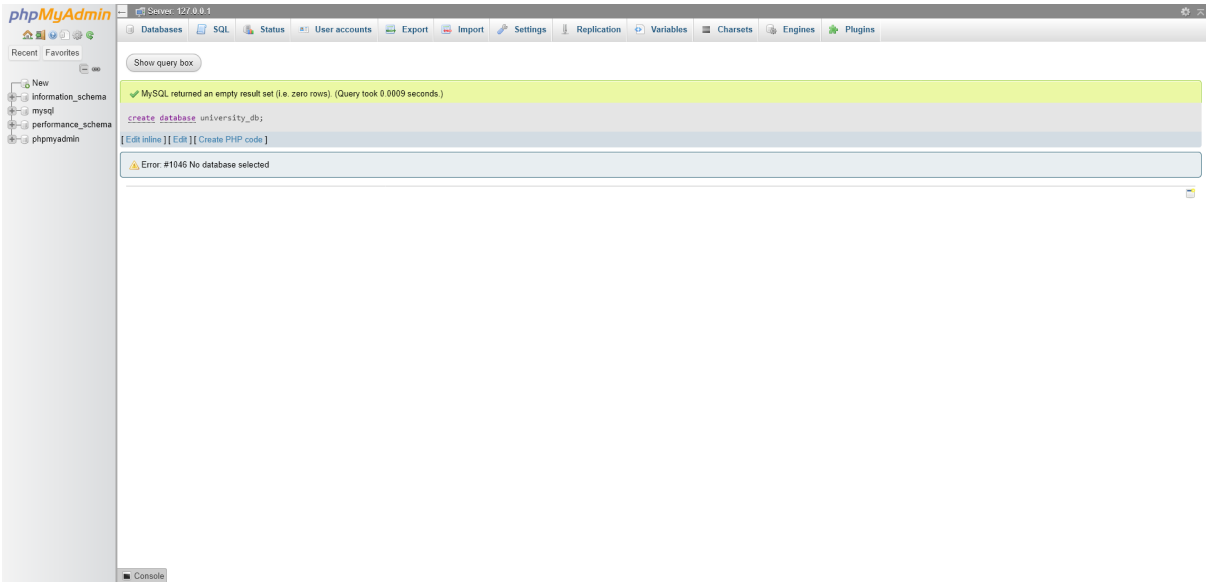
4.1 Create a table **courses** with columns: **course_id**, **course_name**, and **course_credits**. Set the **course_id** as the primary key.

```sql
use school_db;

create table courses (
    course_id int primary key auto_increment,
    course_name varchar(255),
    course_credits int
);

select *
from courses;
```



4.2 Use the **CREATE** command to create a database **university_db**.

```sql
create database university_db;
```

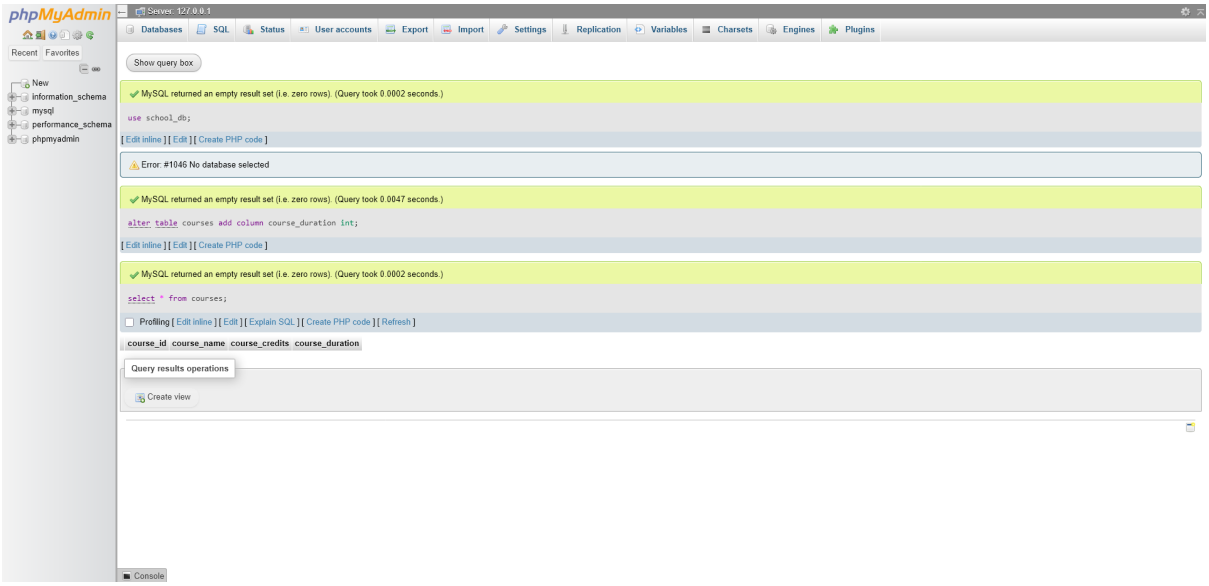


5 ALTER Command

5.1 Modify the **courses** table by adding a column **course_duration** using the **ALTER** command.

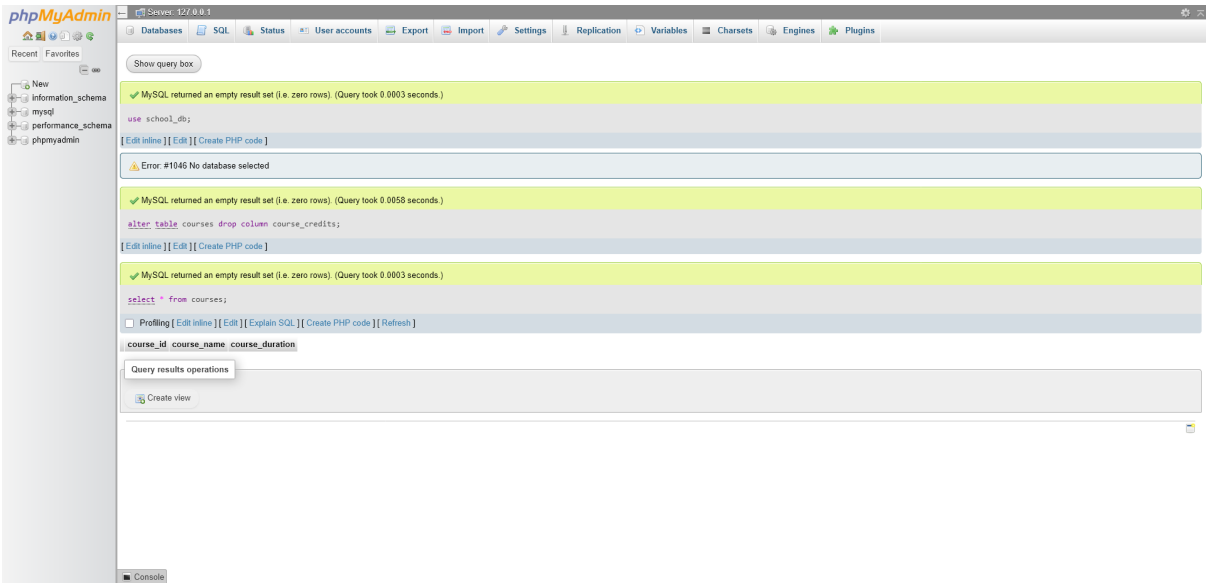
```sql
use school_db;

alter table courses
add column course_duration int;
```

```
5
6 select *
7 from courses;
```



5.2 Drop the `course_credits` column from the `courses` table.

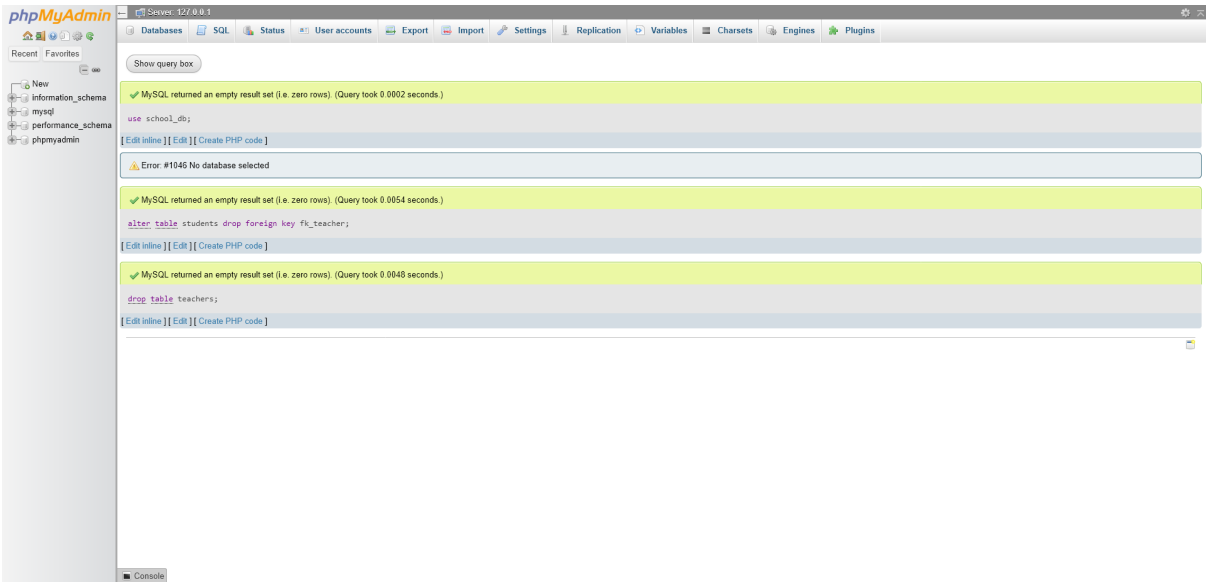
```
1 use school_db;
2
3 alter table courses
4 drop column course_credits;
5
6 select *
7 from courses;
```



6 DROP Command

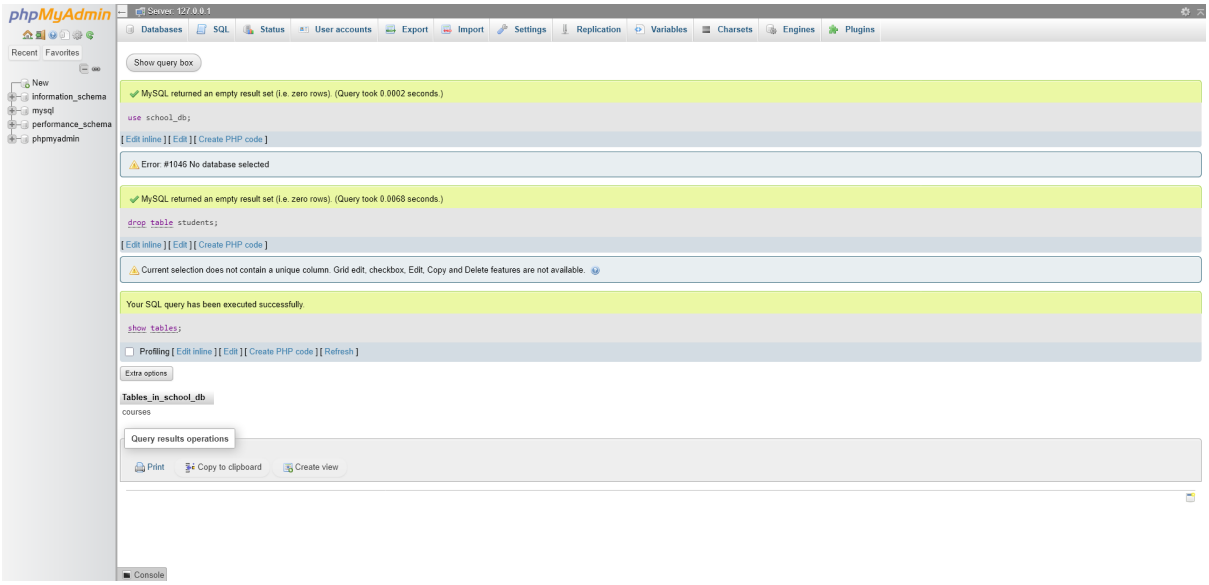
6.1 Drop the `teachers` table from the `school_db` database.

```
use school_db;

alter table students
drop foreign key fk_teacher;

drop table teachers;
```



6.2 Drop the `students` table from the `school_db` database and verify that the table has been removed.

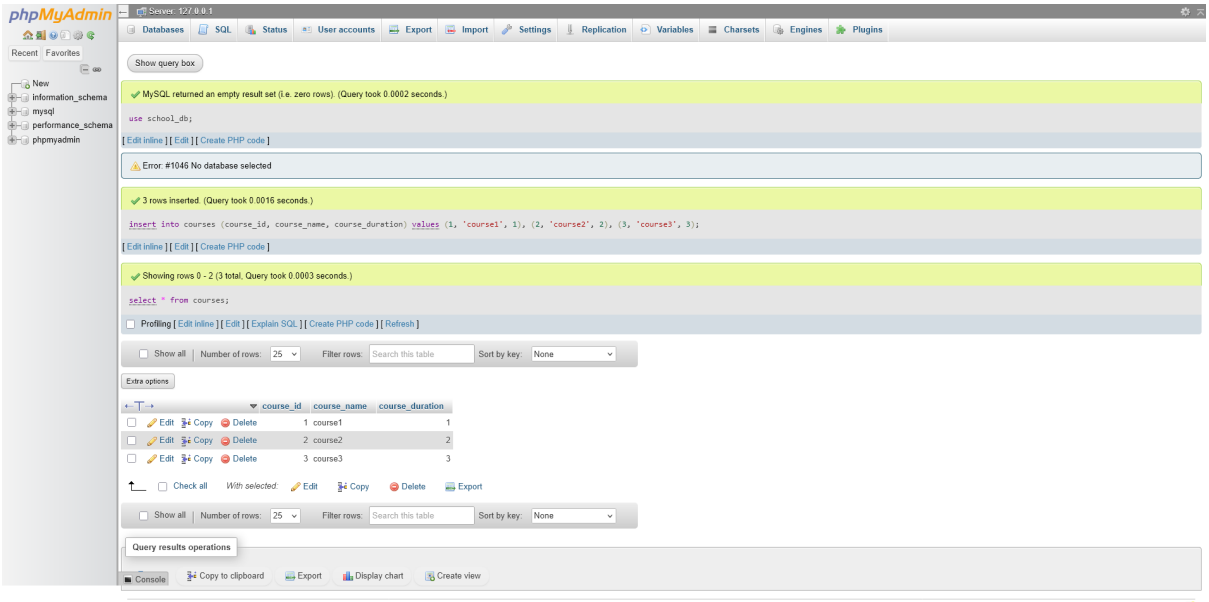
```
use school_db;

drop table students;

show tables;
```



7 Data Manipulation Language (DML)

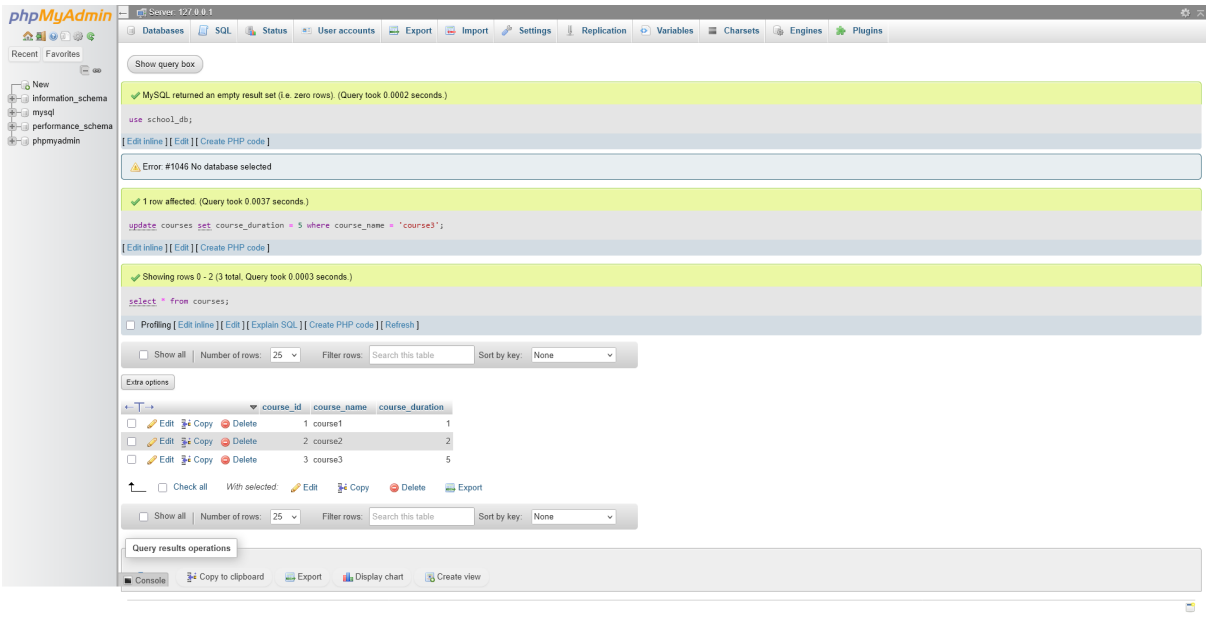
7.1 Insert three records into the **courses** table using the **INSERT** command.

```sql
use school_db;

insert into courses (course_id, course_name, course_duration) values
    (1, 'course1', 1),
    (2, 'course2', 2),
    (3, 'course3', 3);

select *
from courses;
```



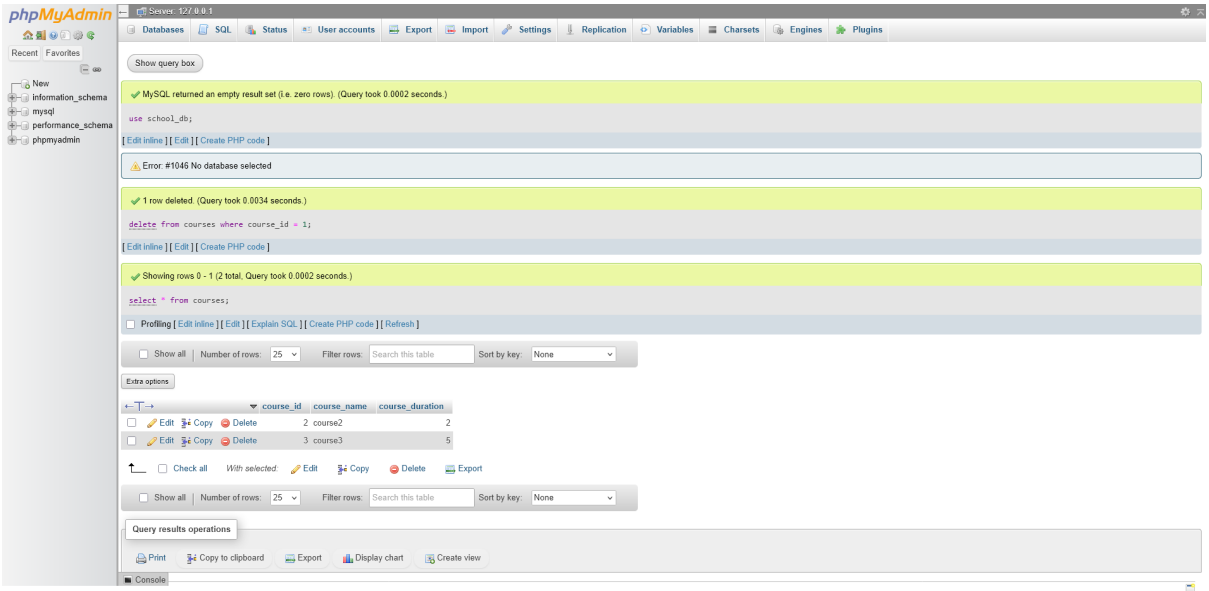
7.2 Update the course duration of a specific course using the **UPDATE** command.

```sql
use school_db;

update courses
set course_duration = 5
where course_name = 'course3';

select *
from courses;
```



7.3 Delete a course with a specific **course_id** from the **courses** table using the **DELETE** command.

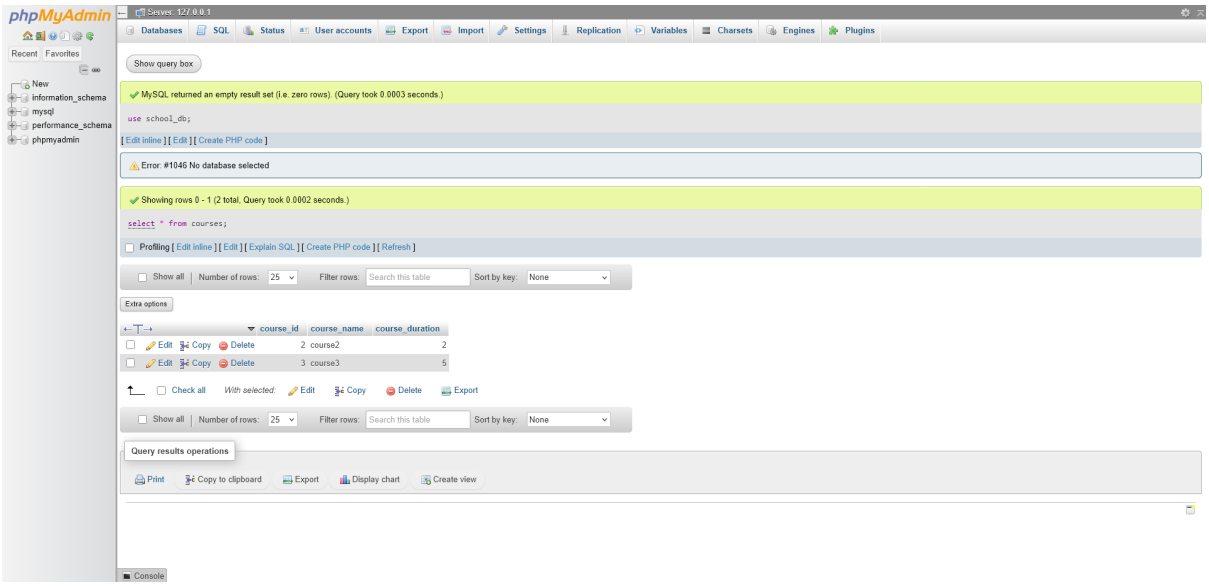
```sql
use school_db;

delete from courses
where course_id = 1;

select *
from courses;
```



8 Data Query Language (DQL)

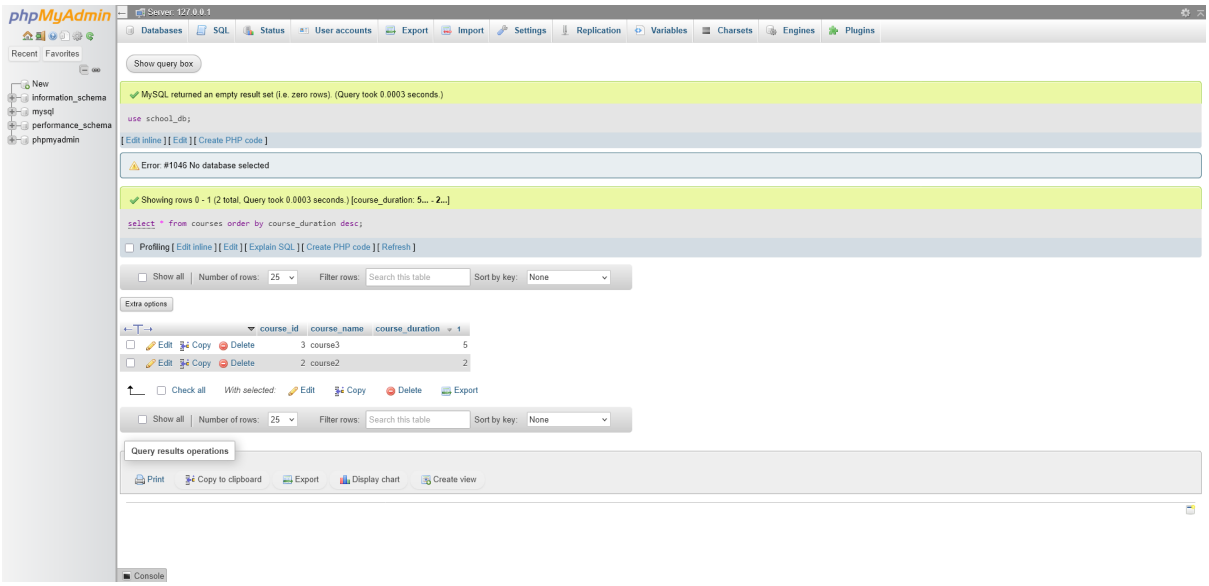
8.1 Retrieve all courses from the **courses** table using the **SELECT** statement.

```
use school_db;

select *
from courses;
```



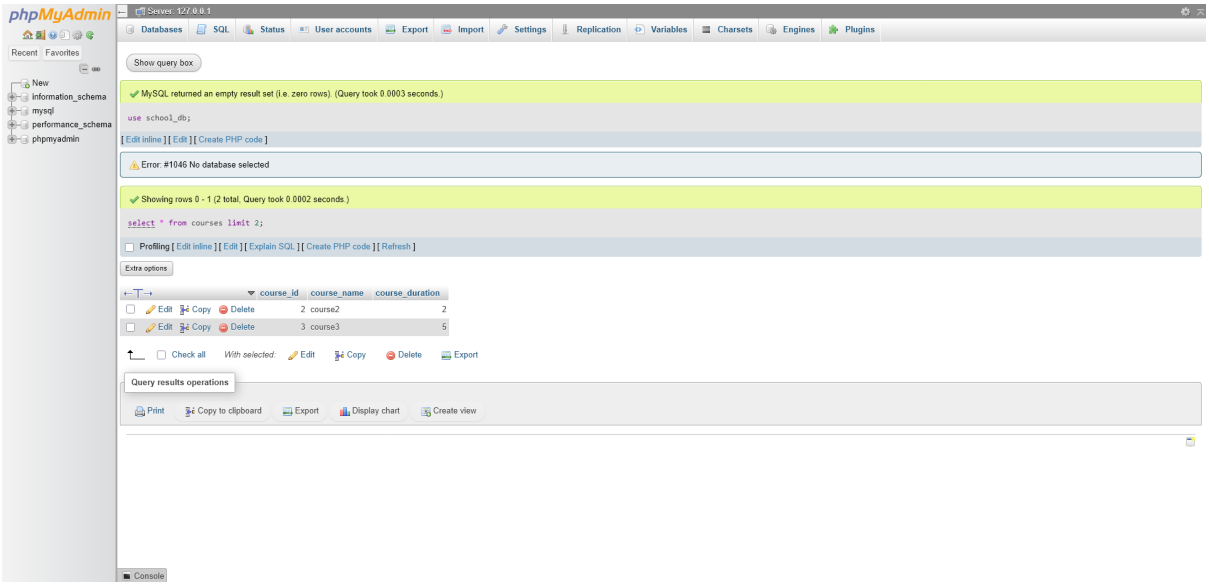
8.2 Sort the courses based on **course_duration** in descending order using **ORDER BY**.

```sql
use school_db;

select *
from courses
order by course_duration desc;
```



8.3 Limit the results of the **SELECT** query to show only the top two courses using **LIMIT**.

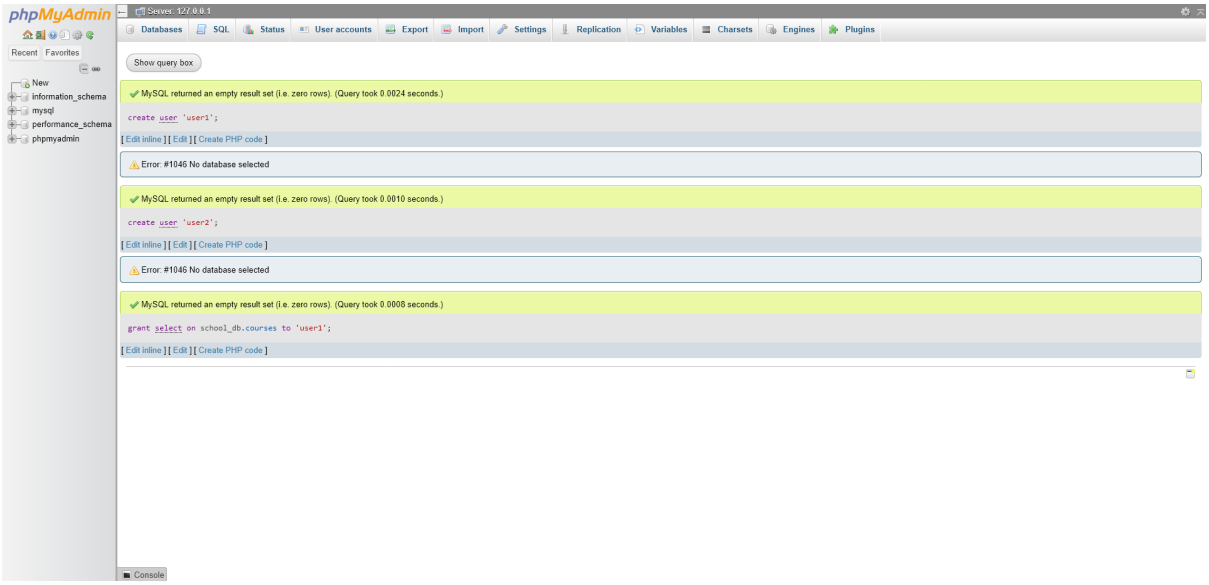
```sql
use school_db;

select *
from courses limit 2;
```



9 Data Control Language (DCL)

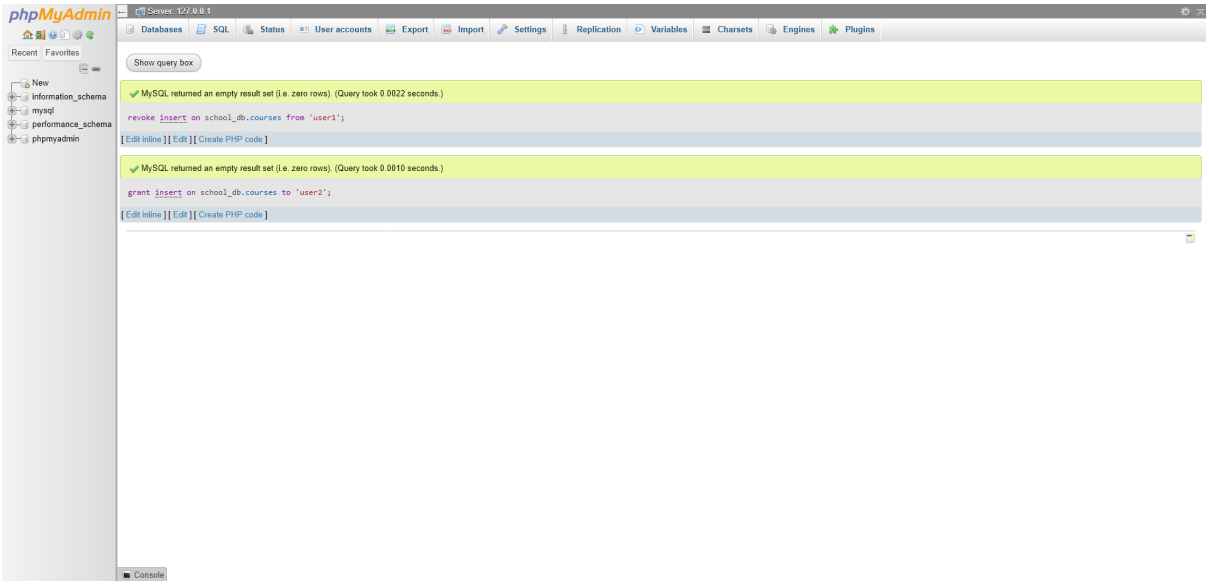
9.1 Create two new users **user1** and **user2** and grant **user1** permission to **SELECT** from the **courses** table.

```
create user 'user1';

create user 'user2';

grant select on school_db.courses
to 'user1';
```



9.2 Revoke the **INSERT** permission from **user1** and give it to **user2**.

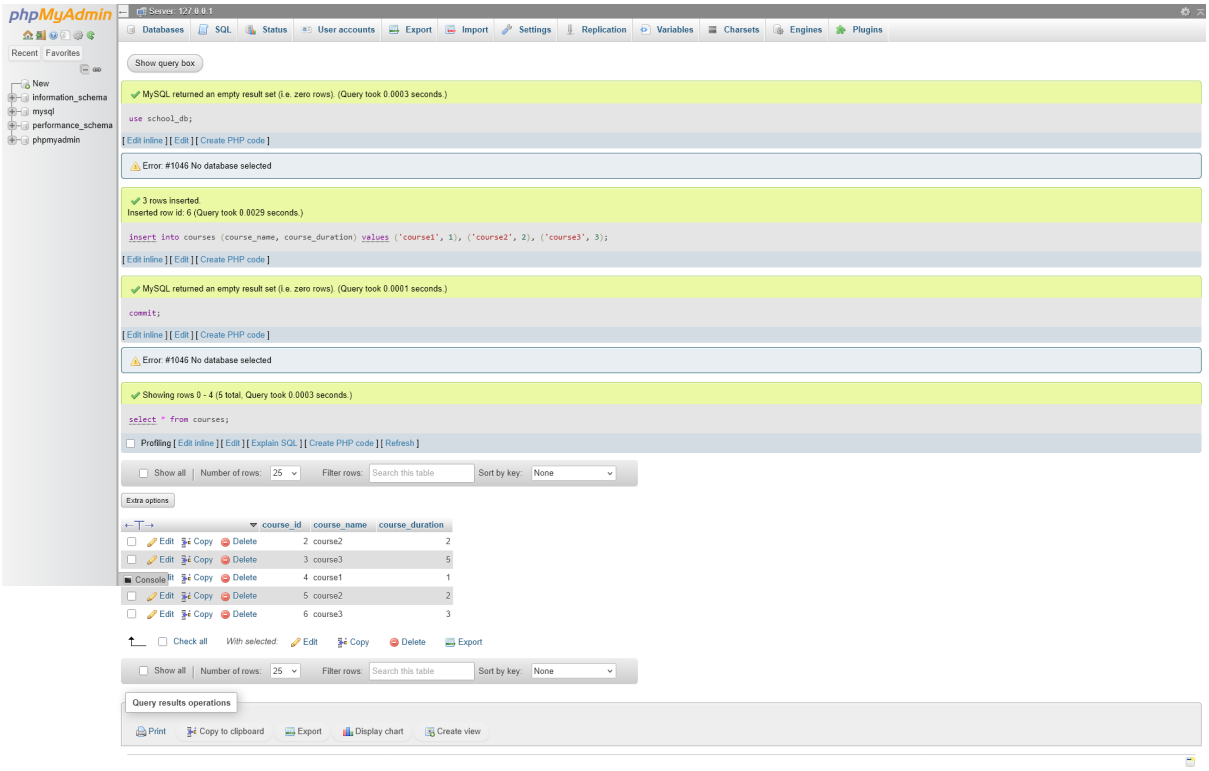
```
revoke insert on school_db.courses
from 'user1';

grant insert on school_db.courses
to 'user2';
```



10 Transaction Control Language (TCL)

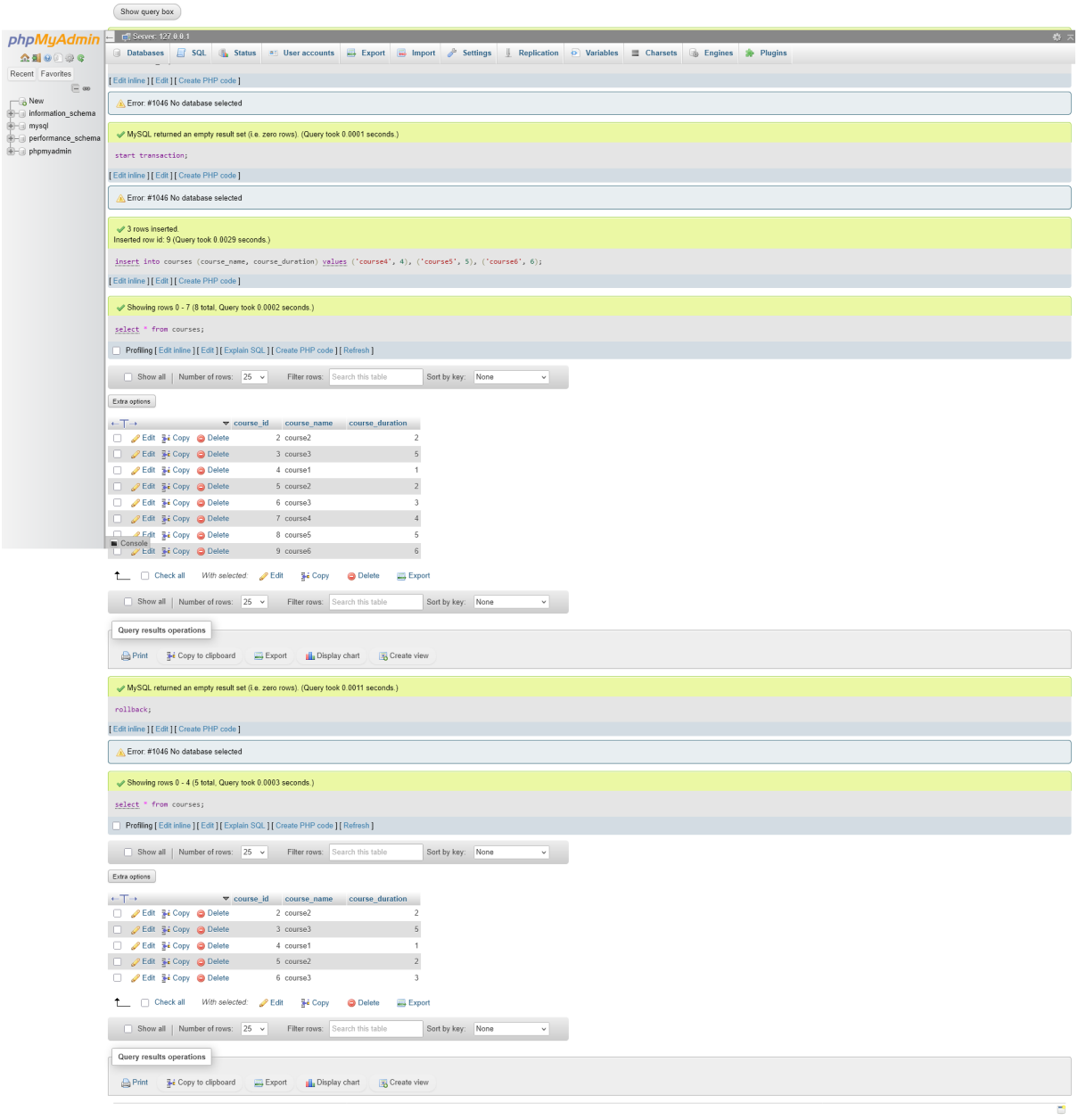
10.1 Insert a few rows into the **courses** table and use **COMMIT** to save the changes.

```sql
use school_db;

insert into courses (course_name, course_duration) values
    ('course1', 1),
    ('course2', 2),
    ('course3', 3);

commit;

select * from courses;
```



10.2 Insert additional rows, then use **ROLLBACK** to undo the last insert operation.

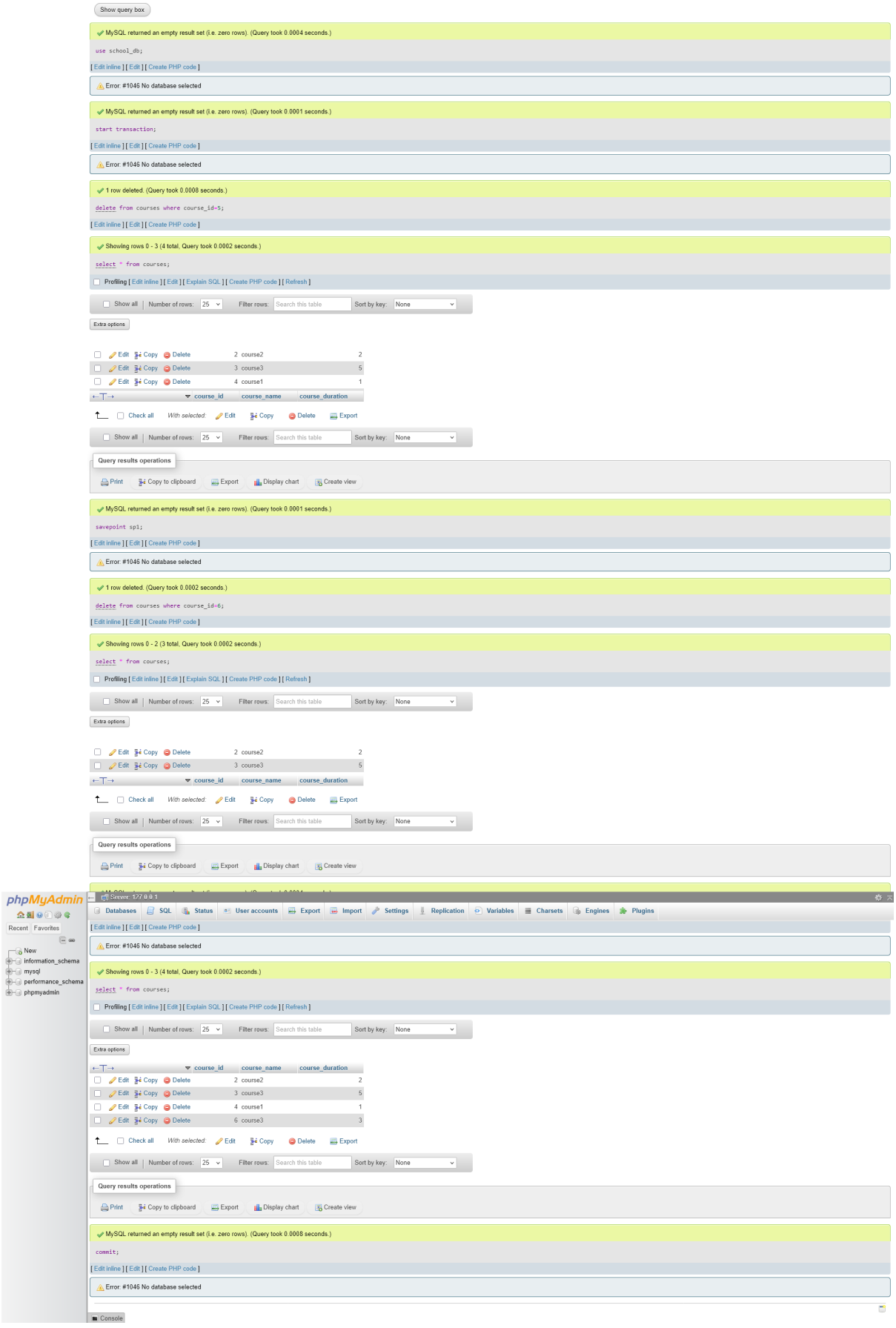
```
use school_db;

start transaction;

insert into courses (course_name, course_duration) values
    ('course4', 4),
    ('course5', 5),
    ('course6', 6);

select * from courses;

rollback;

select * from courses;
```



10.3 Create a **SAVEPOINT** before updating the **courses** table, and use it to roll back specific changes.

```
use school_db;

start transaction;

delete from courses
where course_id=5;

select * from courses;

savepoint sp1;
```

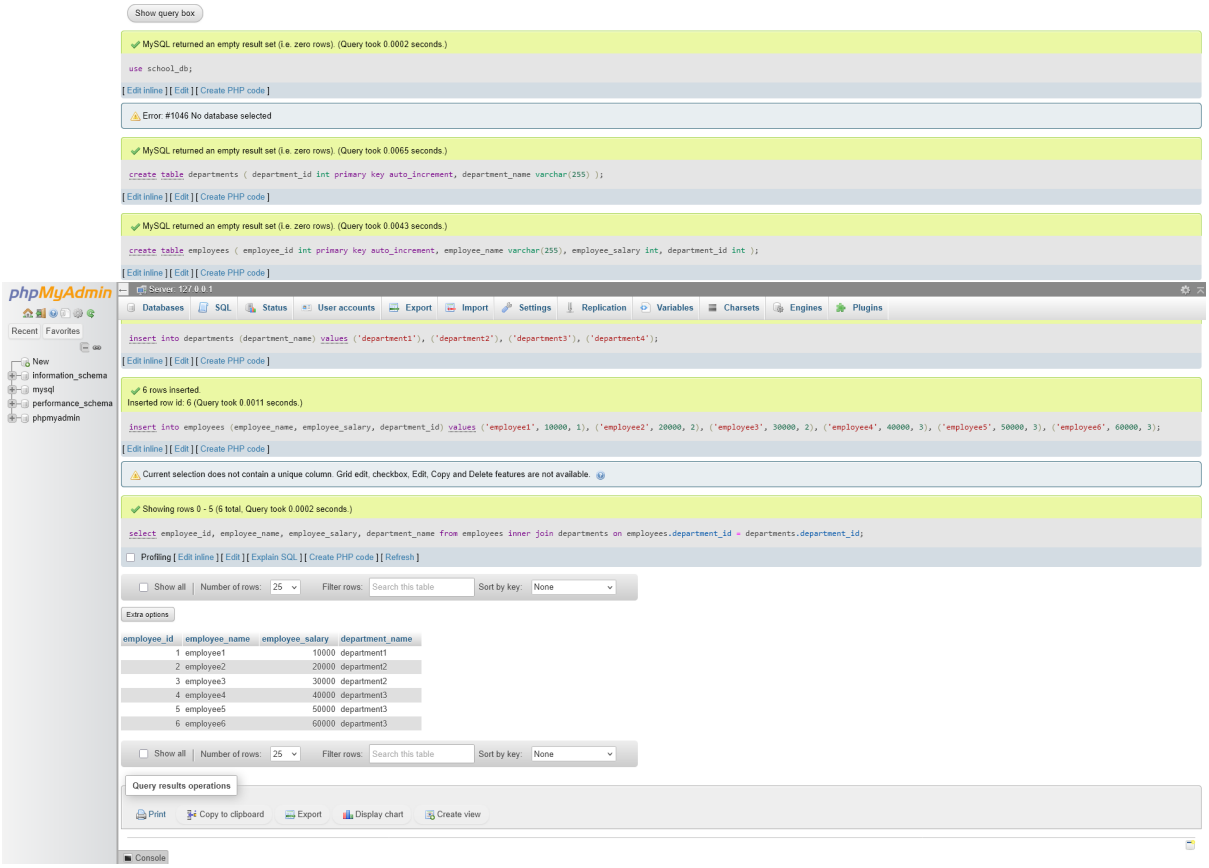
```
11
12 delete from courses
13 where course_id=6;
14
15 select * from courses;
16
17 rollback to savepoint sp1;
18
19 select * from courses;
20
21 commit;
```


Show query box
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)
use school_db;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)
start transaction;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
1 row deleted. (Query took 0.0008 seconds.)
delete from courses where course_id=5;
[Edit inline] [Edit] [Create PHP code]
Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)
select * from courses;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
Edit Copy Delete 4 course1 1
course_id course_name course_duration
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)
savepoint sp1;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
1 row deleted. (Query took 0.0002 seconds.)
delete from courses where course_id=6;
[Edit inline] [Edit] [Create PHP code]
Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)
select * from courses;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
course_id course_name course_duration
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
phpMyAdmin
Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins
Recent Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)
select * from courses;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
course_id course_name course_duration
Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
Edit Copy Delete 4 course1 1
Edit Copy Delete 6 course3 3
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)
commit;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
Console

11 SQL Joins

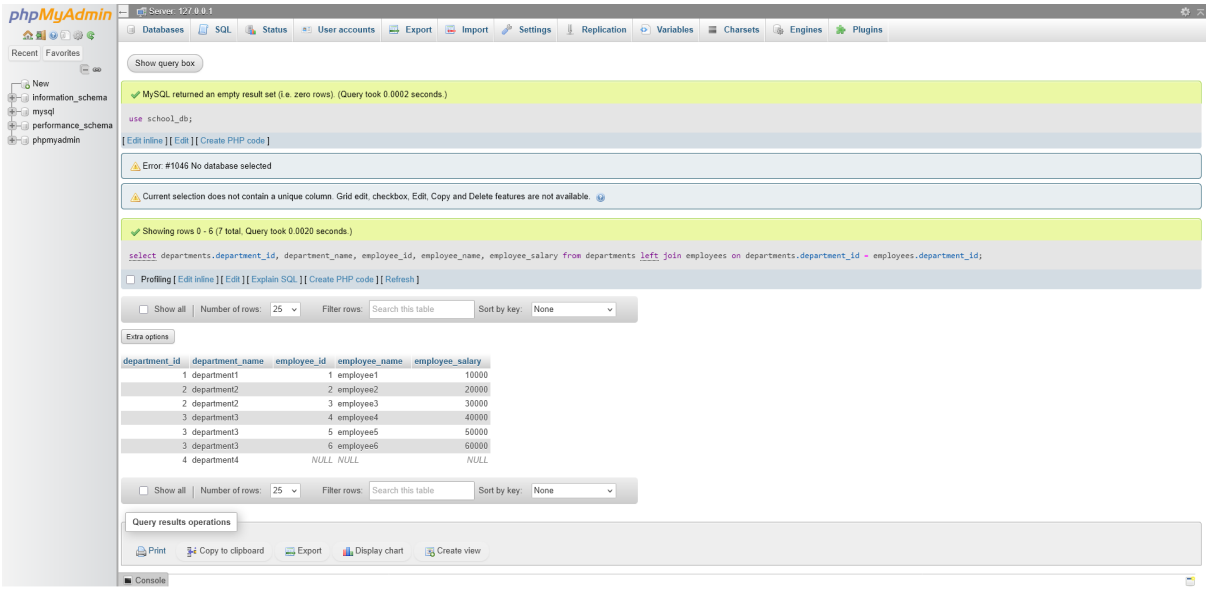
11.1 Create two tables: **departments** and **employees**. Perform an **INNER JOIN** to display employees along with their respective departments.

```sql
use school_db;

create table departments (
    department_id int primary key auto_increment,
    department_name varchar(255)
);

create table employees (
    employee_id int primary key auto_increment,
    employee_name varchar(255),
    employee_salary int,
    department_id int
);

insert into departments (department_name) values
    ('department1'),
    ('department2'),
    ('department3'),
    ('department4');

insert into employees (employee_name, employee_salary, department_id
    ) values
    ('employee1', 10000, 1),
    ('employee2', 20000, 2),
    ('employee3', 30000, 2),
    ('employee4', 40000, 3),
    ('employee5', 50000, 3),
    ('employee6', 60000, 3);

select employee_id, employee_name, employee_salary, department_name
from employees
inner join departments
on employees.department_id = departments.department_id;
```



11.2 Use a `LEFT JOIN` to show all departments, even those without employees.

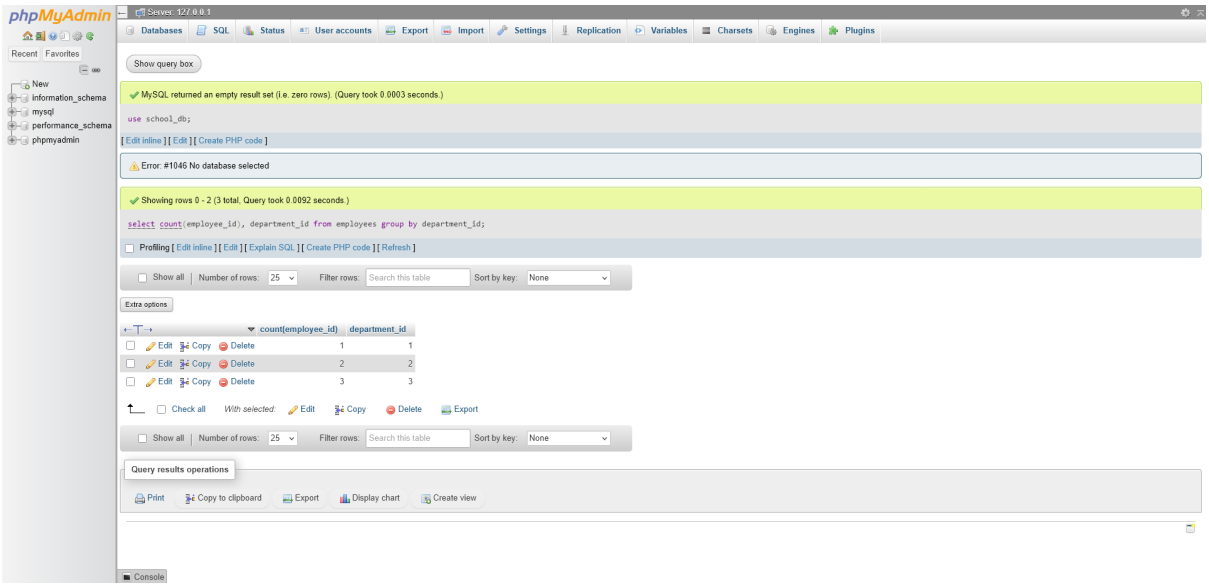
```sql
use school_db;

select departments.department_id, department_name, employee_id,
   employee_name, employee_salary
from departments
left join employees
on departments.department_id = employees.department_id;
```



12 SQL Group By

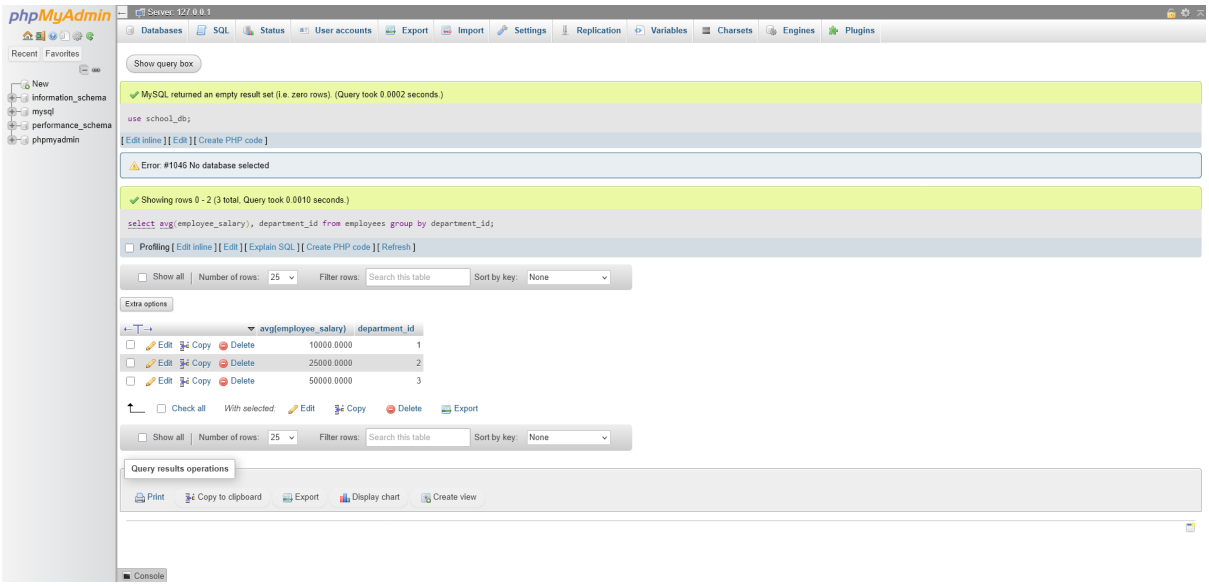
12.1 Group employees by department and count the number of employees in each department using `GROUP BY`.

```
use school_db;

select count(employee_id), department_id
from employees
group by department_id;
```



12.2 Use the `AVG` aggregate function to find the average salary of employees in each department.

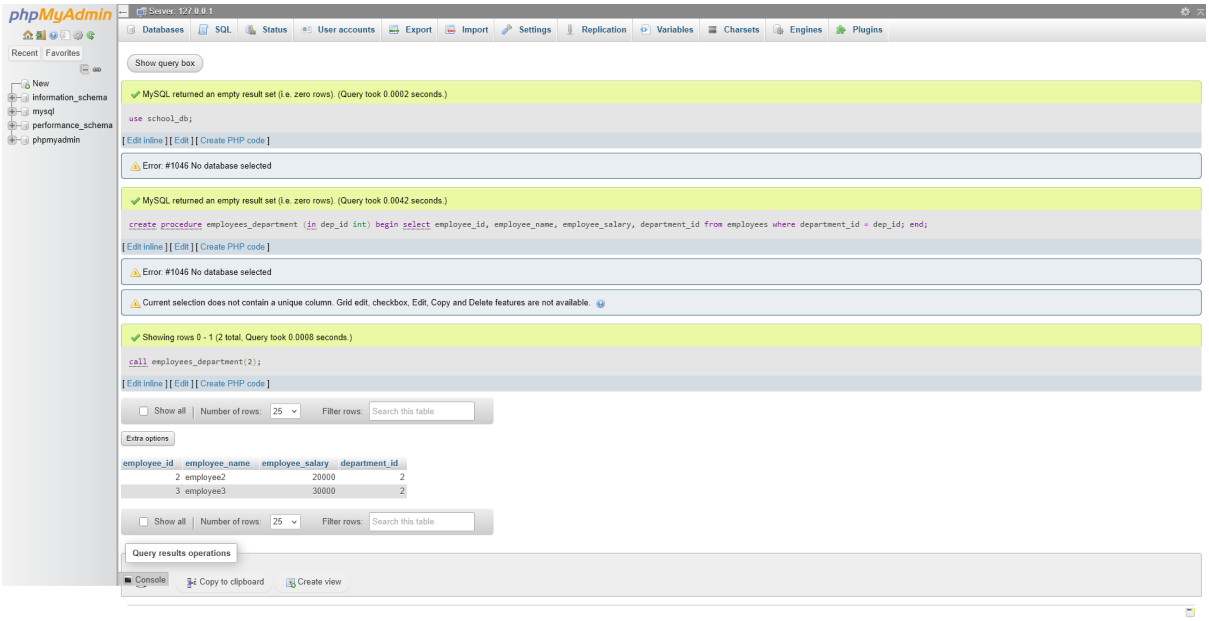
```
use school_db;

select avg(employee_salary), department_id
from employees
group by department_id;
```



13 SQL Stored Procedure

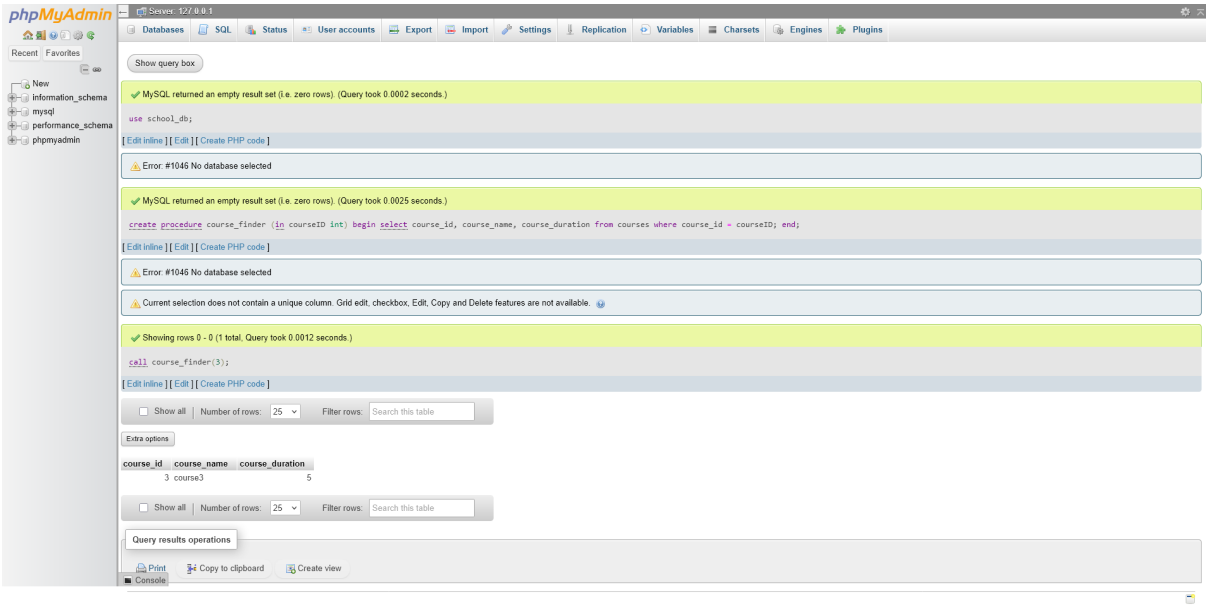
13.1 Write a stored procedure to retrieve all employees from the **employees** table based on department.

```
use school_db;

delimiter $$

create procedure employees_department (in dep_id int)
begin
    select employee_id, employee_name, employee_salary,
        department_id
    from employees
    where department_id = dep_id;
end $$

delimiter ;


call employees_department(2);
```



13.2 Write a stored procedure that accepts **course_id** as input and returns the course details.

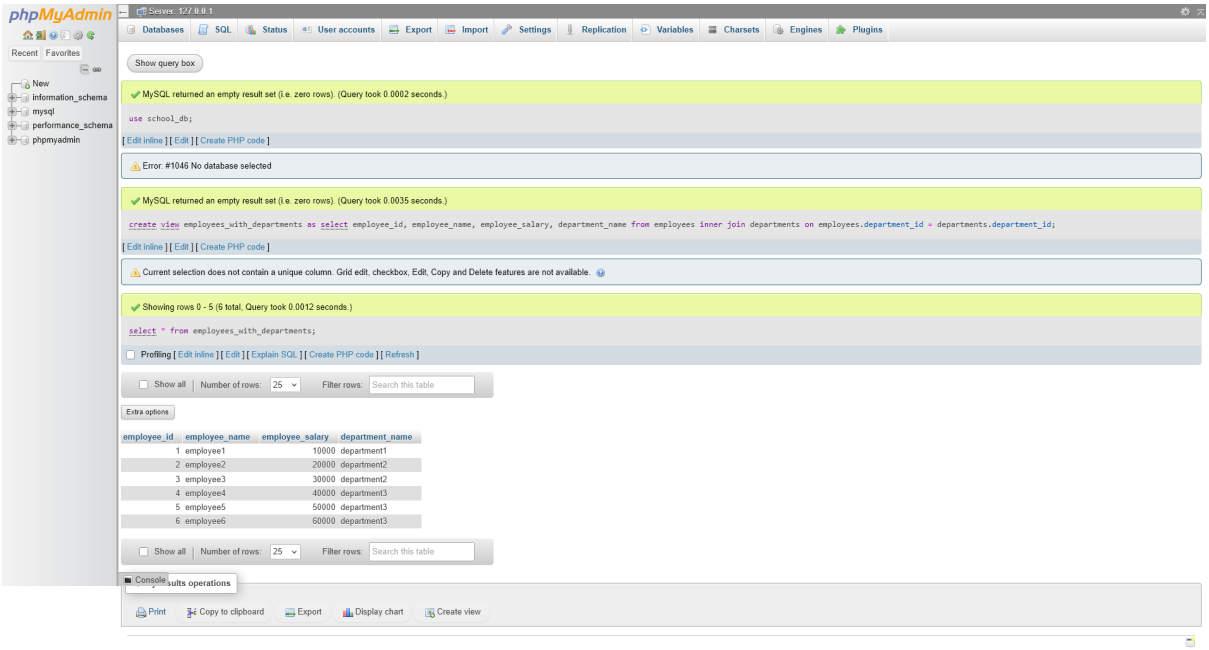
```
use school_db;

delimiter $$

create procedure course_finder (in courseID int)
begin
    select course_id, course_name, course_duration
    from courses
    where course_id = courseID;
end $$

delimiter ;


call course_finder(3);
```



14 SQL View

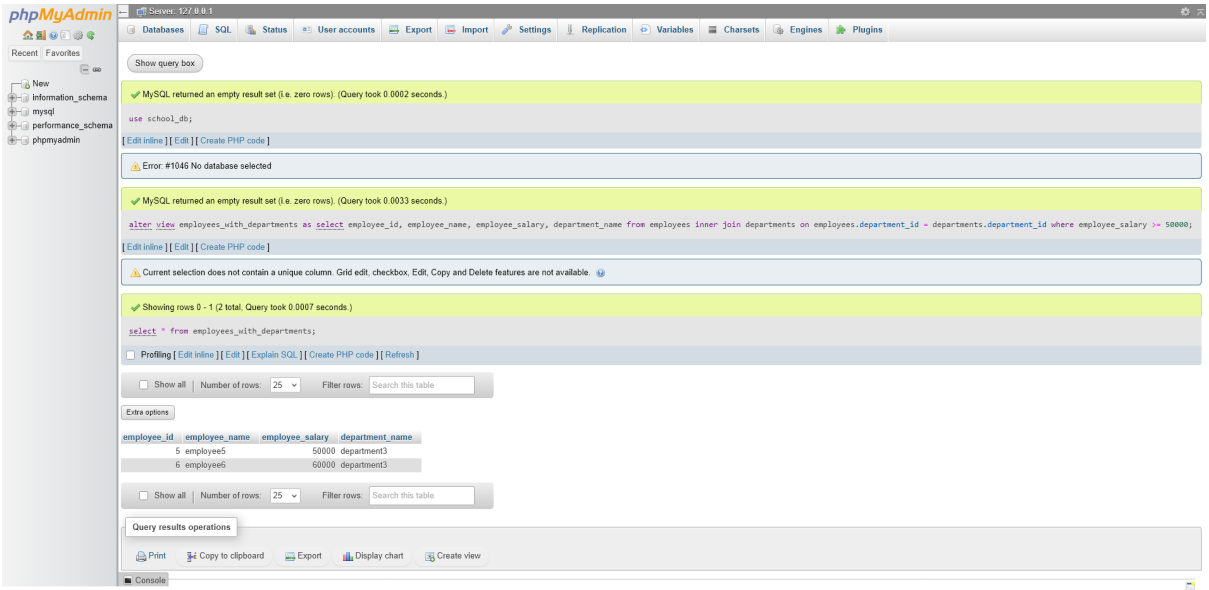
14.1 Create a view to show all employees along with their department names.

```sql
use school_db;

create view employees_with_departments as
select employee_id, employee_name, employee_salary, department_name
from employees
inner join departments
on employees.department_id = departments.department_id;


select * from employees_with_departments;
```



14.2 Modify the view to exclude employees whose salaries are below $50,000.

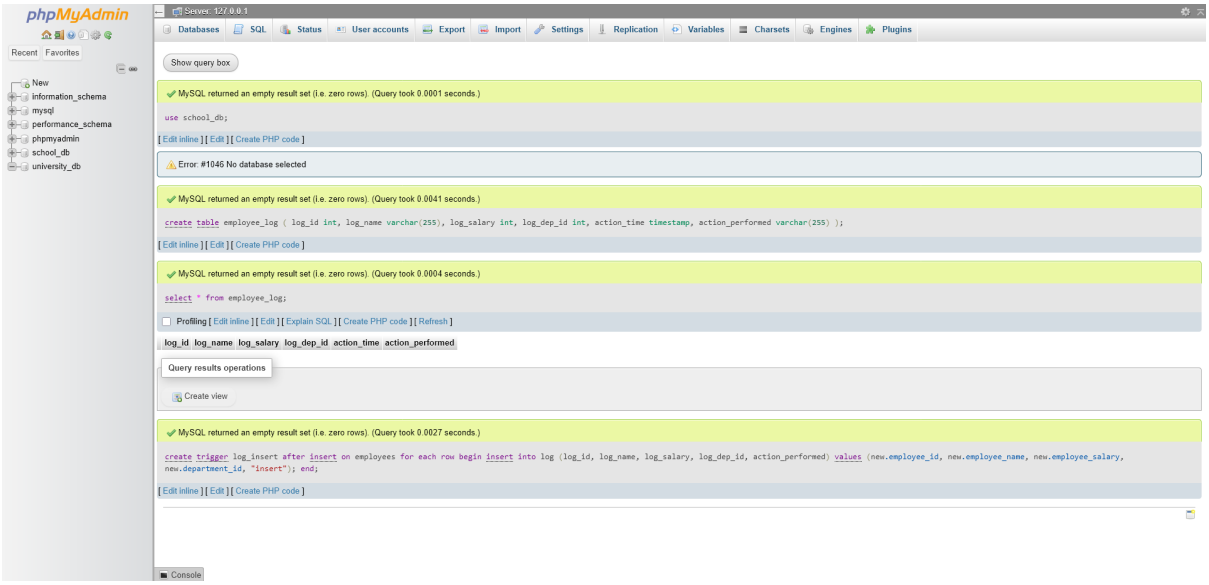
```
use school_db;

alter view employees_with_departments as
select employee_id, employee_name, employee_salary, department_name
from employees
inner join departments
on employees.department_id = departments.department_id
where employee_salary >= 50000;


select * from employees_with_departments;
```



15 SQL Triggers

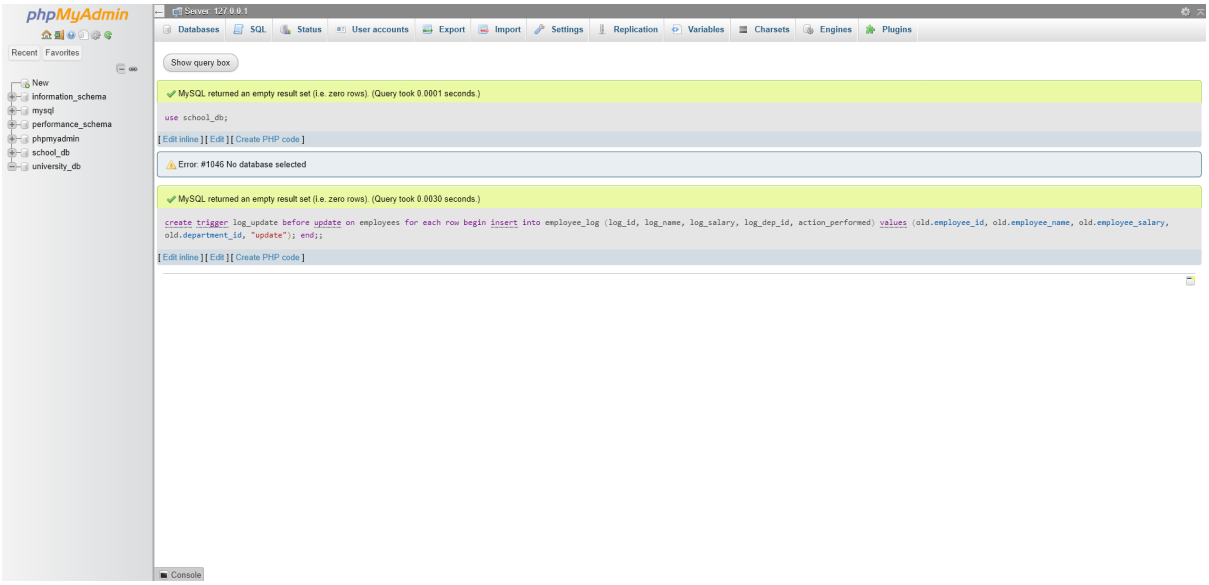
15.1 Create a trigger to automatically log changes to the **employees** table when a new employee is added.

```sql
use school_db;

create table employee_log (
    log_id int,
    log_name varchar(255),
    log_salary int,
    log_dep_id int,
    action_time timestamp,
    action_performed varchar(255)
);

select * from employee_log;

delimiter $$
create trigger log_insert
after insert on employees
for each row
begin
    insert into log (log_id, log_name, log_salary, log_dep_id,
        action_performed)
    values (new.employee_id, new.employee_name, new.employee_salary,
         new.department_id, "insert");
end $$
delimiter ;
```



15.2 Create a trigger to update the `last_modified` timestamp whenever an employee record is updated.

```
use school_db;

delimiter $$
create trigger log_update
before update on employees
for each row
begin
    insert into employee_log (log_id, log_name, log_salary,
        log_dep_id, action_performed)
    values (old.employee_id, old.employee_name, old.employee_salary,
        old.department_id, "update");
end; $$
delimiter ;
```



16 Introduction to PL/SQL

16.1 Write a PL/SQL block to print the total number of **employees** from the employees table.

4/4_16_1.png

16.2 Create a PL/SQL block that calculates the total sales from an **orders** table.

4/4_16_2.png

17 PL/SQL Control Structures

17.1 Write a PL/SQL block using an `IF-THEN` condition to check the department of an employee.

4/4_17_1.png

17.2 Use a `FOR LOOP` to iterate through employee records and display their names.

4/4_17_2.png

18 SQL Cursors

18.1 Write a PL/SQL block using an explicit cursor to retrieve and display employee details.

4/4_18_1.png

18.2 Create a cursor to retrieve all courses and display them one by one.

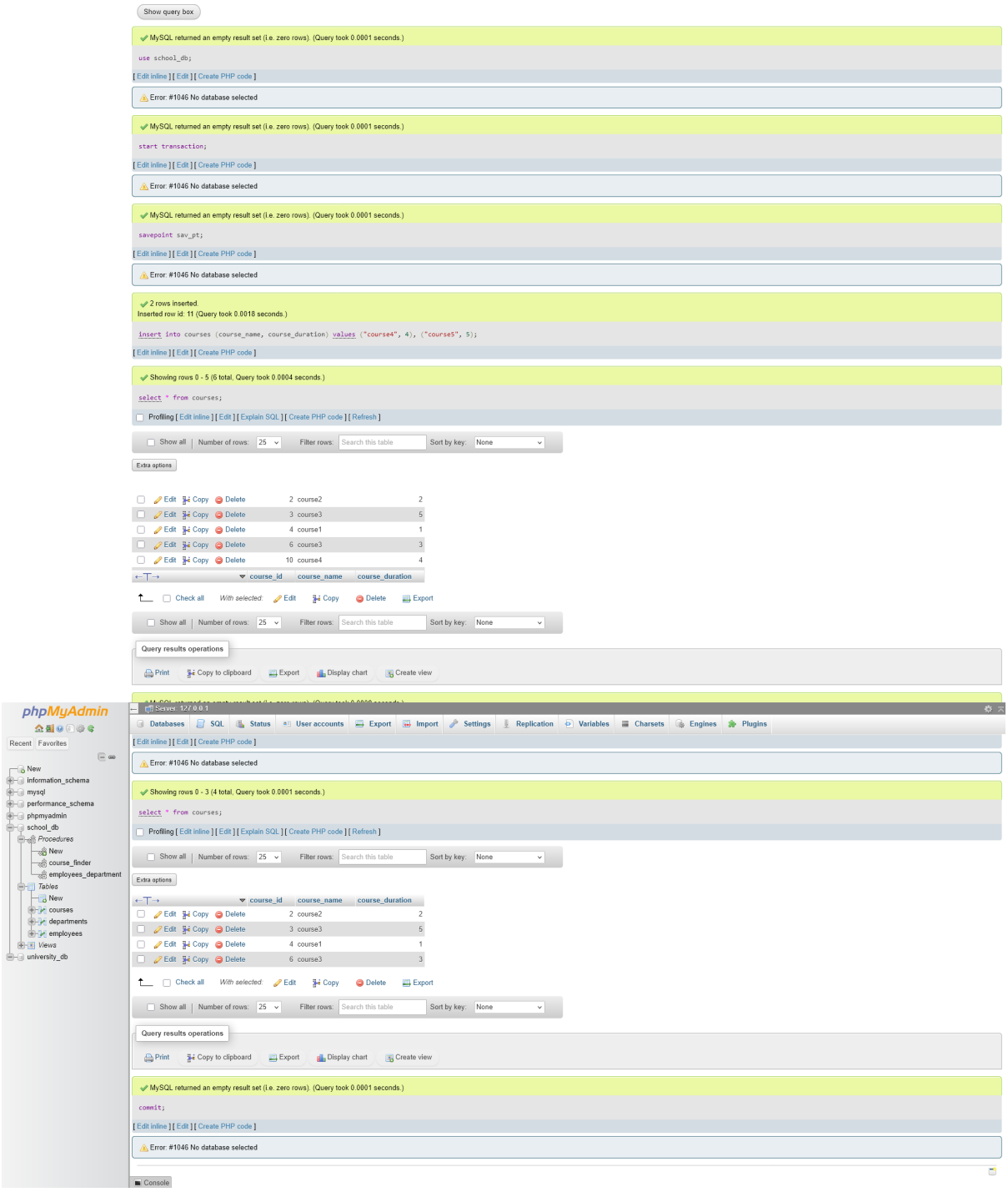
4/4_18_2.png

19 Rollback and Commit Savepoint

19.1 Perform a transaction where you create a savepoint, insert records, then rollback to the savepoint?

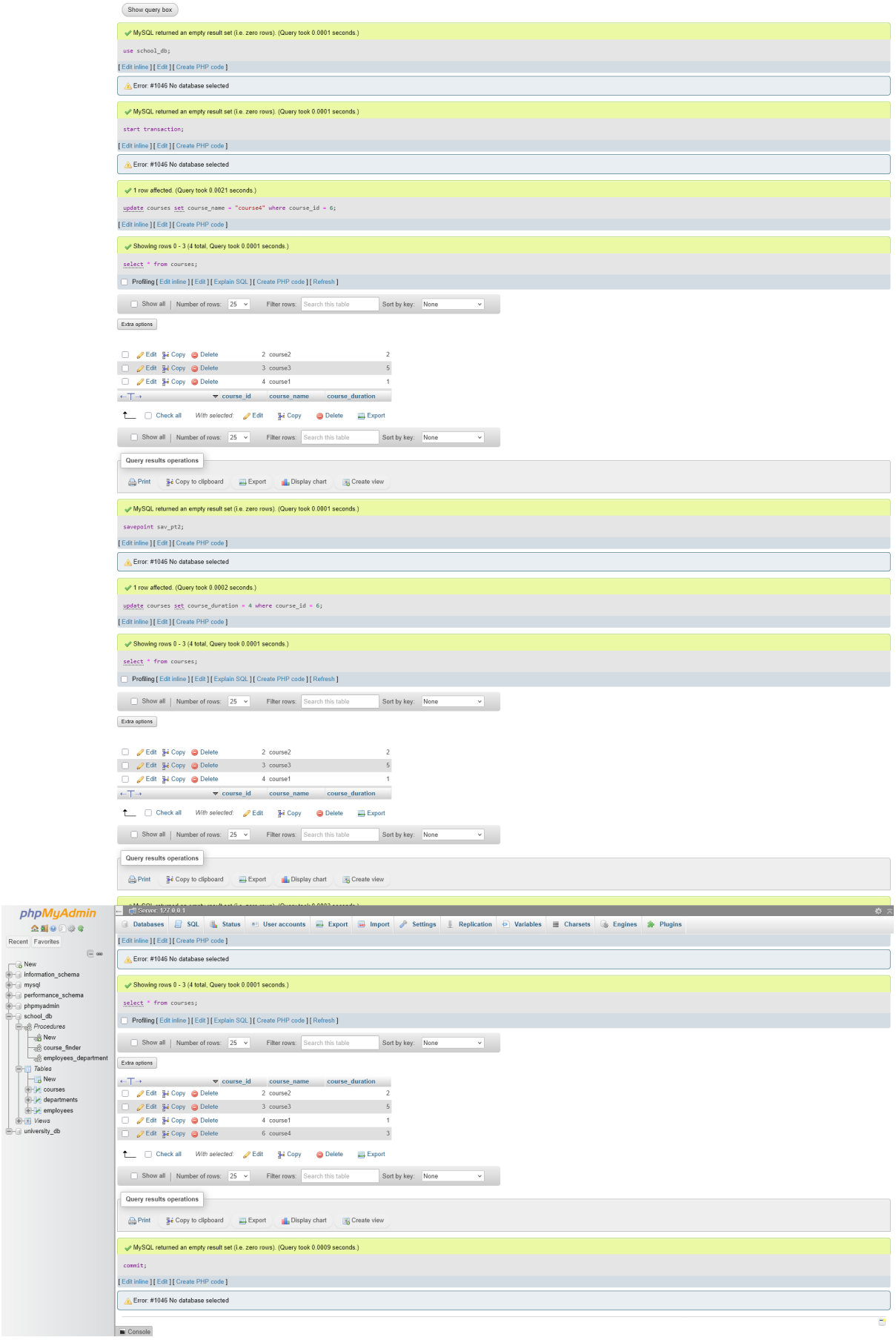
```
use school_db;

start transaction;

savepoint sav_pt;

insert into courses (course_name, course_duration) values
    ("course4", 4),
    ("course5", 5);

```

```
11 select * from courses;
12
13 rollback to savepoint sav_pt;
14
15 select * from courses;
16
17 commit;
```



19.2 Commit part of a transaction after using a savepoint and then rollback the remaining changes.

```
use school_db;

start transaction;

update courses
set course_name = "course4"
where course_id = 6;

select * from courses;

savepoint sav_pt2;

update courses
set course_duration = 4
where course_id = 6;

select * from courses;

rollback to sav_pt2;

select * from courses;

commit;
```


Show query box
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)
use school_db;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)
start transaction;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
1 row affected. (Query took 0.0021 seconds.)
update courses set course_name = "course4" where course_id = 6;
[Edit inline] [Edit] [Create PHP code]
Showing rows 0 - 3 (4 total, Query took 0.0001 seconds.)
select * from courses;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
Edit Copy Delete 4 course1 1
course_id course_name course_duration
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)
savepoint sav_pt2;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
1 row affected. (Query took 0.0002 seconds.)
update courses set course_duration = 4 where course_id = 6;
[Edit inline] [Edit] [Create PHP code]
Showing rows 0 - 3 (4 total, Query took 0.0001 seconds.)
select * from courses;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
Edit Copy Delete 4 course1 1
course_id course_name course_duration
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
phpMyAdmin
Recent Favorites
New
Information_schema
mysql
performance_schema
phpmyadmin
school_db
Procedures
New
course_finder
employees_department
Tables
New
courses
departments
employees
Views
university_db
Server: 127.0.0.1
Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
Showing rows 0 - 3 (4 total, Query took 0.0001 seconds.)
select * from courses;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
course_id course_name course_duration
Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
Edit Copy Delete 4 course1 1
Edit Copy Delete 6 course4 3
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)
commit;
[Edit inline] [Edit] [Create PHP code]
Error: #1046 No database selected
Console