

# Module 4 – Introduction to DBMS (Lab Exercises)

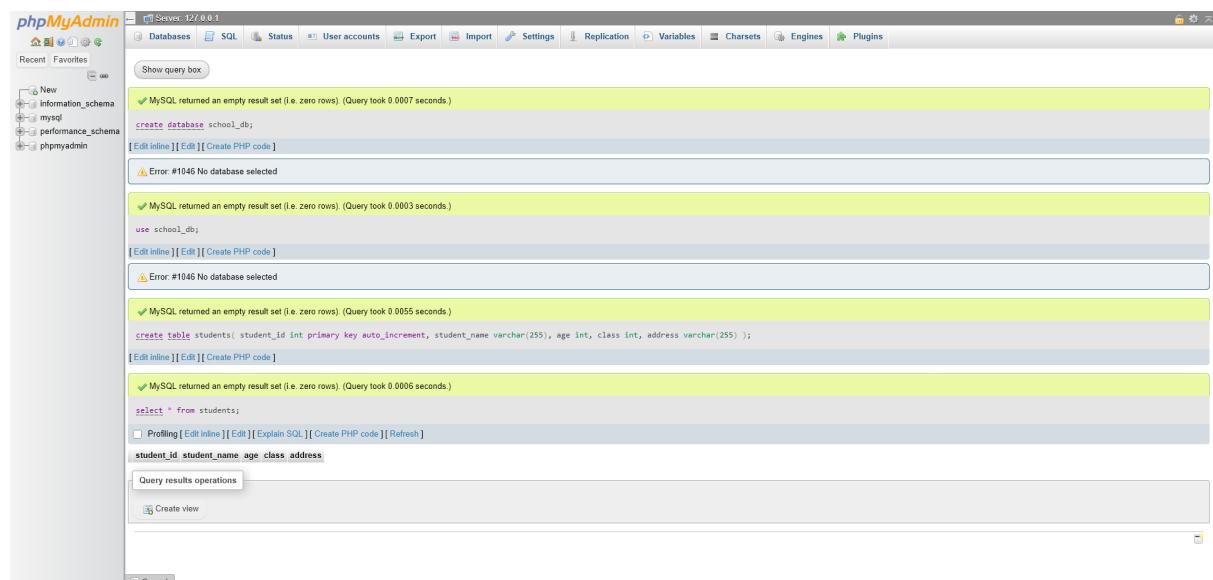
Shivam Gangwani

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**.

```
1 create database school_db;
2 use school_db;
3
4 create table students(
5     student_id int primary key auto_increment,
6     student_name varchar(255),
7     age int,
8     class int,
9     address varchar(255)
10);
11
12 select *
13 from students;
```



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

```

1 use school_db;
2
3 insert into students (student_name, age, class, address) values
4     ("student_1", 6, 1, "address of student_1"),
5     ("student_2", 11, 6, "address of student_2"),
6     ("student_3", 13, 8, "address of student_3"),
7     ("student_4", 16, 11, "address of student_4"),
8     ("student_5", 8, 3, "address of student_5");
9
10 select *
11 from students;
```

The screenshot shows the phpMyAdmin interface. In the left sidebar, there's a tree view of databases: New, information\_schema, mysql, performance\_schema, and phpmyadmin. The main area shows a query editor with the following content:

```
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;
```

Below the query, the results pane displays:

- 5 rows inserted.
- Inserted row id: 5 (Query took 0.0032 seconds.)
- Showing rows 0 - 4 (5 total. Query took 0.0003 seconds.)

	student_id	student_name	age	class	address
1	student_1	6	1	address of student_1	
2	student_2	11	6	address of student_2	
3	student_3	13	8	address of student_3	
4	student_4	16	11	address of student_4	
5	student_5	8	3	address of student_5	

## 2 SQL Syntax

### 2.1 Write SQL queries to retrieve specific columns (**student\_name** and **age**) from the **students** table.

```

1 use school_db;
2
3 select student_name, age
4 from students;
```

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0002 seconds)

```
use school_db;
```

Error #1046 No database selected

Showing rows 0 - 4 (5 total). Query took 0.0003 seconds.

	student_name	age
<input type="checkbox"/>	student_1	6
<input type="checkbox"/>	student_2	11
<input type="checkbox"/>	student_3	13
<input type="checkbox"/>	student_4	16
<input type="checkbox"/>	student_5	8

Profile [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

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

Console

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

```
1 use school_db;
2
3 select *
4 from students
5 where age > 10;
```

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0003 seconds)

```
use school_db;
```

Error #1046 No database selected

Showing rows 0 - 2 (3 total). Query took 0.0004 seconds.

	student_id	student_name	age	class	address
<input type="checkbox"/>	2	student_2	11	6	address of student_2
<input type="checkbox"/>	3	student_3	13	8	address of student_3
<input type="checkbox"/>	4	student_4	16	11	address of student_4

Profile [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

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

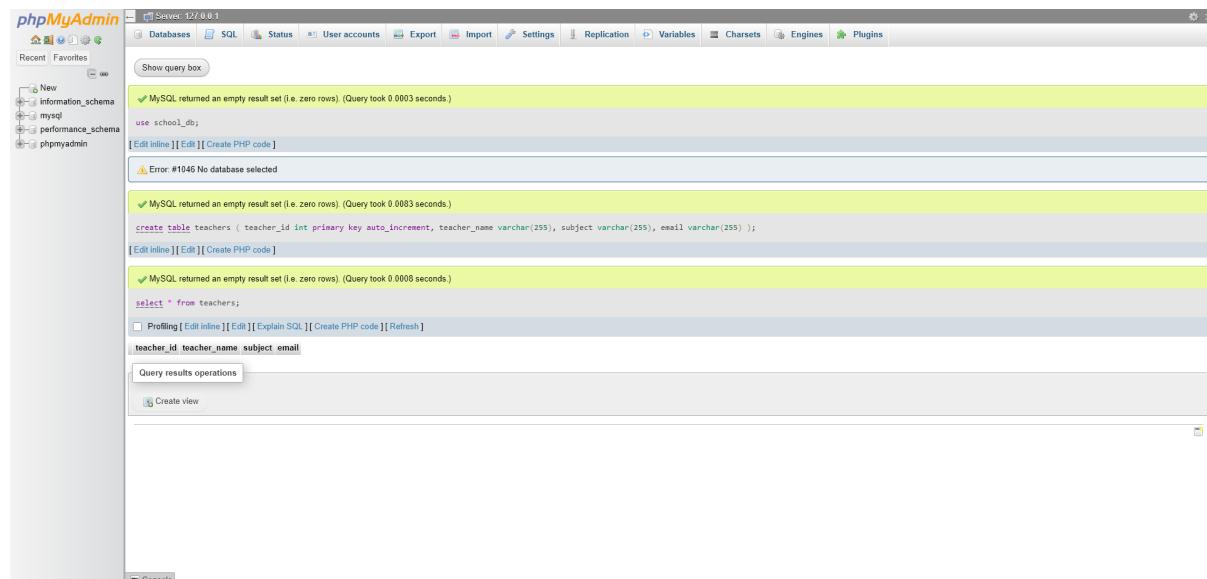
Console

### 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).**

```

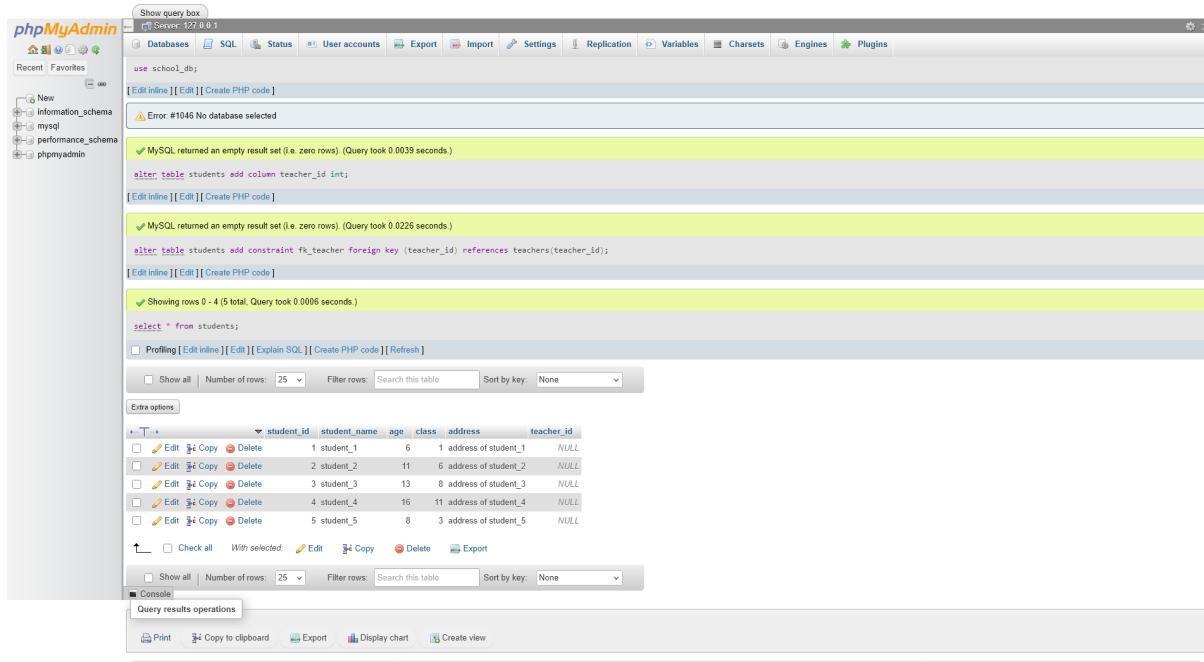
1 use school_db;
2
3 create table teachers (
4     teacher_id int primary key auto_increment,
5     teacher_name varchar(255),
6     subject varchar(255),
7     email varchar(255)
8 );
9
10 select *
11 from teachers;
```



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

```

1 use school_db;
2
3 alter table students
4 add column teacher_id int;
5
6 alter table students
7 add constraint fk_teacher
8 foreign key (teacher_id) references teachers(teacher_id);
9
10 select *
11 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.**

```

1 use school_db;
2
3 create table courses (
4     course_id int primary key auto_increment,
5     course_name varchar(255),
6     course_credits int
7 );
8
9 select *
10 from courses;

```

The screenshot shows the phpMyAdmin interface for MySQL version 127.0.0.1. The left sidebar lists databases: New, information\_schema, mysql, performance\_schema, and phpmyadmin. The 'school\_db' database is selected. The main query window contains the following SQL code:

```

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

```

The results pane shows two green success messages:

- MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds)
- MySQL returned an empty result set (i.e. zero rows). (Query took 0.0070 seconds)

Below the results, there is an error message: "Error: #1046 No database selected".

## 4.2 Use the **CREATE** command to create a database **university\_db**.

```
1 create database university_db;
```

The screenshot shows the phpMyAdmin interface for MySQL version 127.0.0.1. The left sidebar lists databases: New, information\_schema, mysql, performance\_schema, and phpmyadmin. The main query window contains the following SQL code:

```

create database university_db;

```

The results pane shows one green success message:

- MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds)

Below the results, there is an error message: "Error: #1046 No database selected".

## 5 ALTER Command

### 5.1 Modify the **courses** table by adding a column **course\_duration** using the **ALTER** command.

```

1 use school_db;
2
3 alter table courses
4 add column course_duration int;

```

```

5
6 select *
7 from courses;

```

The screenshot shows the phpMyAdmin interface with the following details:

- Left sidebar:** Shows the database structure with databases: information\_schema, mysql, performance\_schema, and phpmyadmin.
- Top menu:** Server 127.0.0.1, Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables,Charsets, Engines, Plugins.
- SQL tab:**
  - Query box: `use school_db;`
  - Result: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)
  - Query box: `select * from courses;`
  - Result: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)
- Bottom:** Query results operations, Create view, Console.

## 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;

```

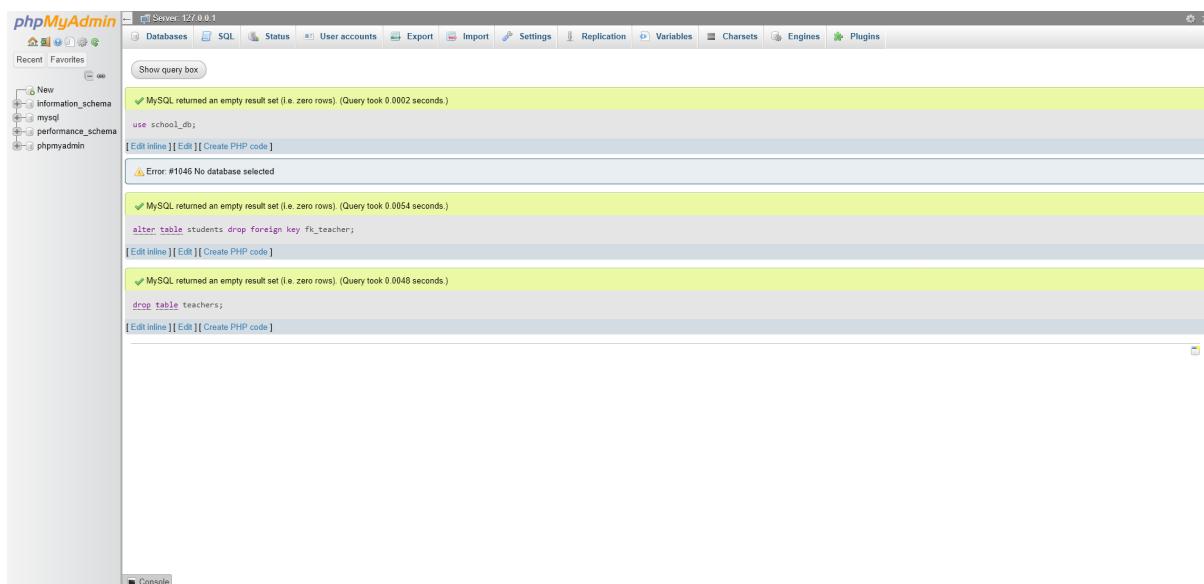
The screenshot shows the phpMyAdmin interface with the following details:

- Left sidebar:** Shows the database structure with databases: information\_schema, mysql, performance\_schema, and phpmyadmin.
- Top menu:** Server 127.0.0.1, Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables,Charsets, Engines, Plugins.
- SQL tab:**
  - Query box: `use school_db;`
  - Result: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)
  - Query box: `alter table courses drop column course_credits;`
  - Result: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0058 seconds.)
  - Query box: `select * from courses;`
  - Result: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)
- Bottom:** Query results operations, Create view, Console.

## 6 DROP Command

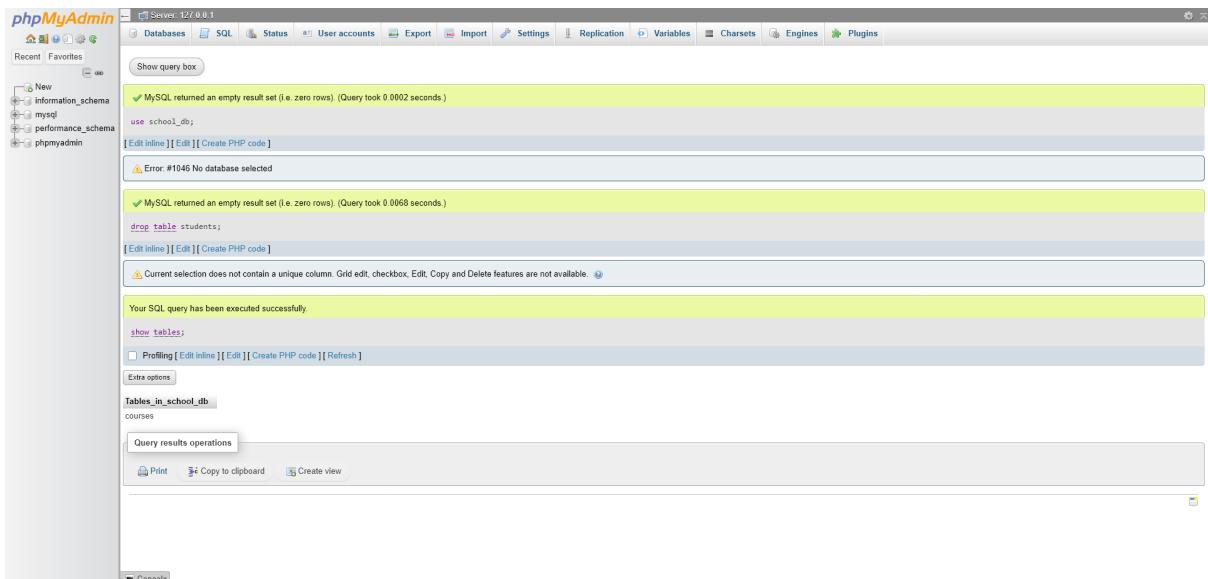
### 6.1 Drop the **teachers** table from the **school\_db** database.

```
1 use school_db;
2
3 alter table students
4 drop foreign key fk_teacher;
5
6 drop table teachers;
```



### 6.2 Drop the **students** table from the **school\_db** database and verify that the table has been removed.

```
1 use school_db;
2
3 drop table students;
4
5 show tables;
```

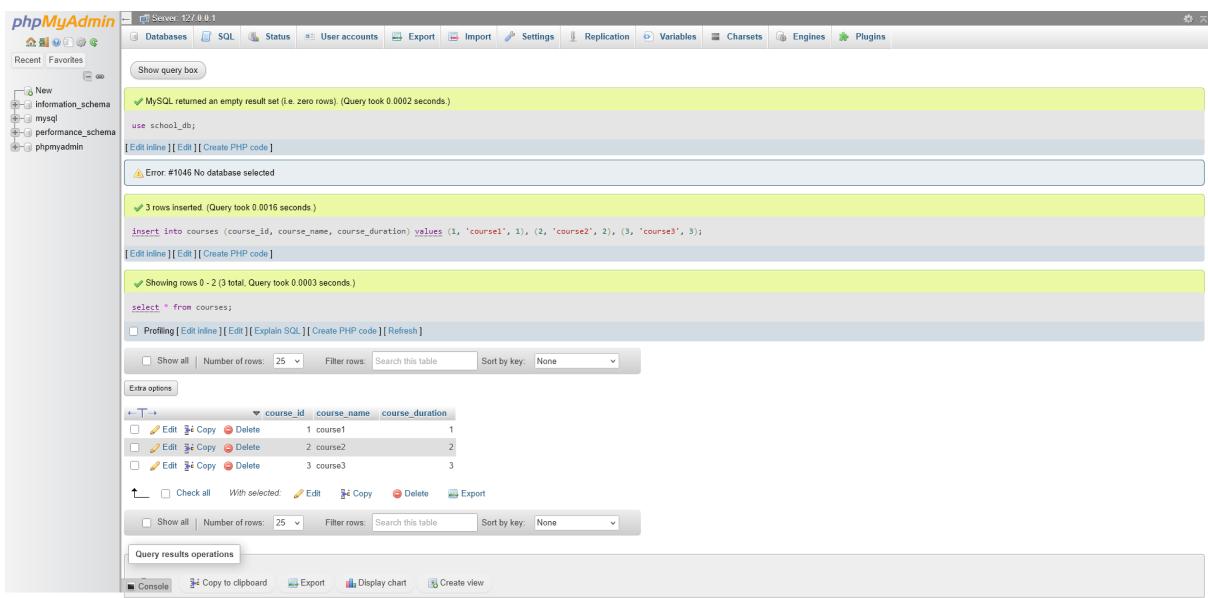


## 7 Data Manipulation Language (DML)

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

```

1 use school_db;
2
3 insert into courses (course_id, course_name, course_duration) values
4     (1, 'course1', 1),
5     (2, 'course2', 2),
6     (3, 'course3', 3);
7
8 select *
9 from courses;
```

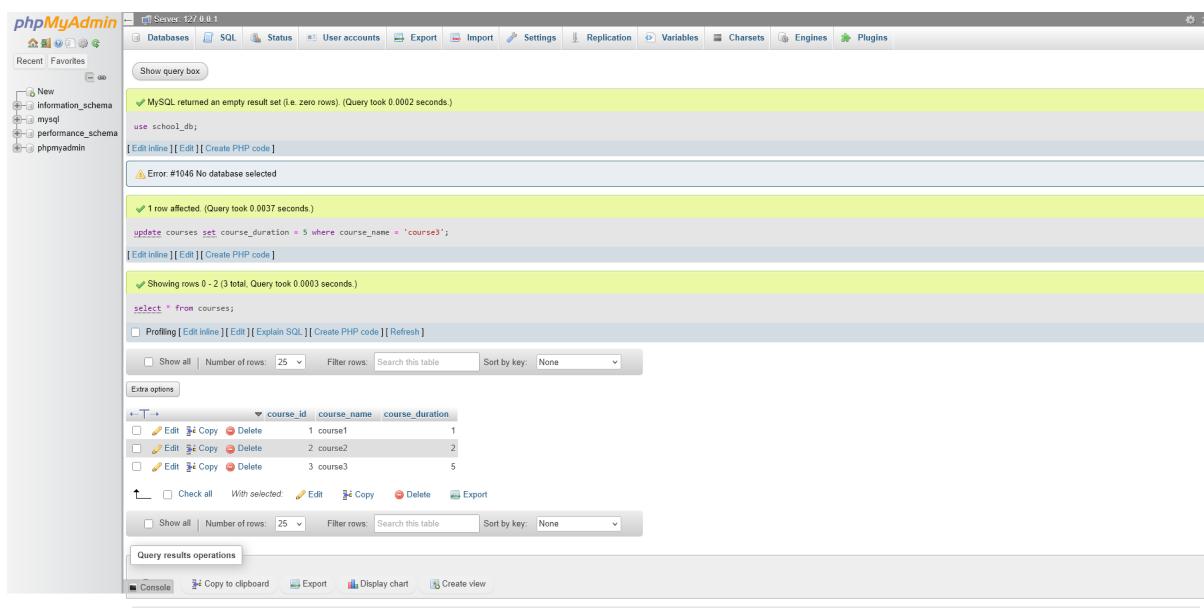


## 7.2 Update the course duration of a specific course using the **UPDATE** command.

```

1 use school_db;
2
3 update courses
4 set course_duration = 5
5 where course_name = 'course3';
6
7 select *
8 from courses;

```

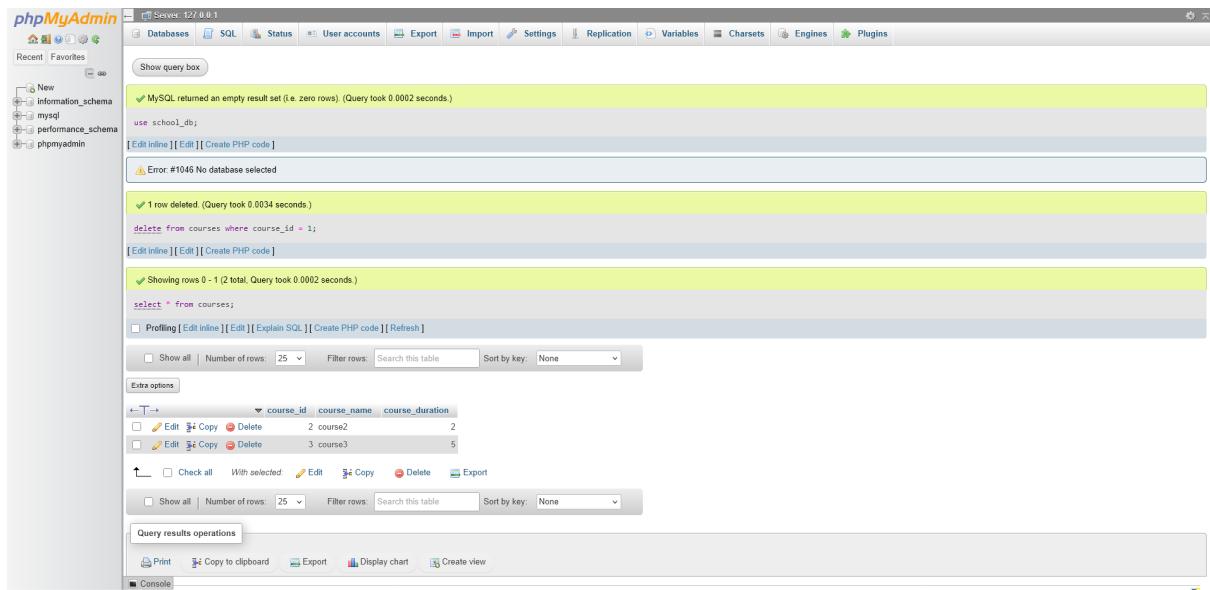


## 7.3 Delete a course with a specific **course\_id** from the **courses** table using the **DELETE** command.

```

1 use school_db;
2
3 delete from courses
4 where course_id = 1;
5
6 select *
7 from courses;

```

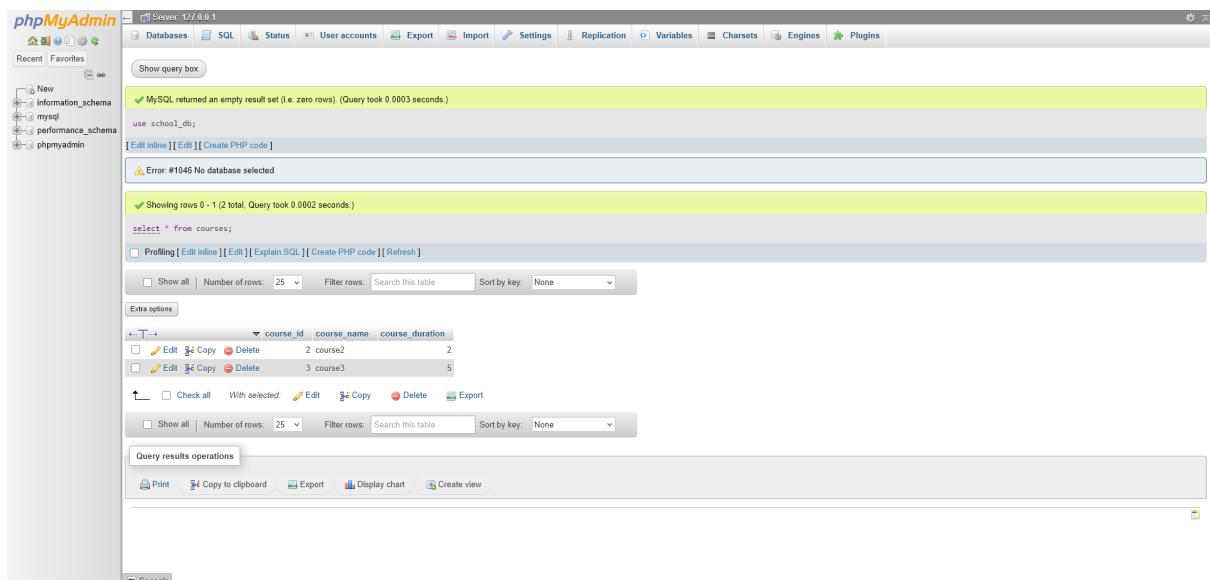


## 8 Data Query Language (DQL)

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

```

1 use school_db;
2
3 select *
4 from courses;
```



### 8.2 Sort the courses based on **course\_duration** in descending order using **ORDER BY**.

```

1 use school_db;
2
3 select *
4   from courses
5   order by course_duration desc;

```

phpMyAdmin - Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication VariablesCharsets Engines Plugins

Show query box

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0003 seconds.)

use school\_db;

Error #1046 No database selected

Showing rows 0 - 1 (2 total. Query took 0.0003 seconds.) [course\_duration: 5... - 2...]

**select** \* **from** courses **order by** course\_duration **desc**;

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
<input type="checkbox"/>	3	course3	5
<input type="checkbox"/>	2	course2	2

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

Console

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

```

1 use school_db;
2
3 select *
4   from courses limit 2;

```

phpMyAdmin - Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication VariablesCharsets Engines Plugins

Show query box

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0003 seconds.)

use school\_db;

Error #1046 No database selected

Showing rows 0 - 1 (2 total. Query took 0.0002 seconds.)

**select** \* **from** courses **limit** 2;

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
<input type="checkbox"/>	2	course2	2
<input type="checkbox"/>	3	course3	5

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

Console

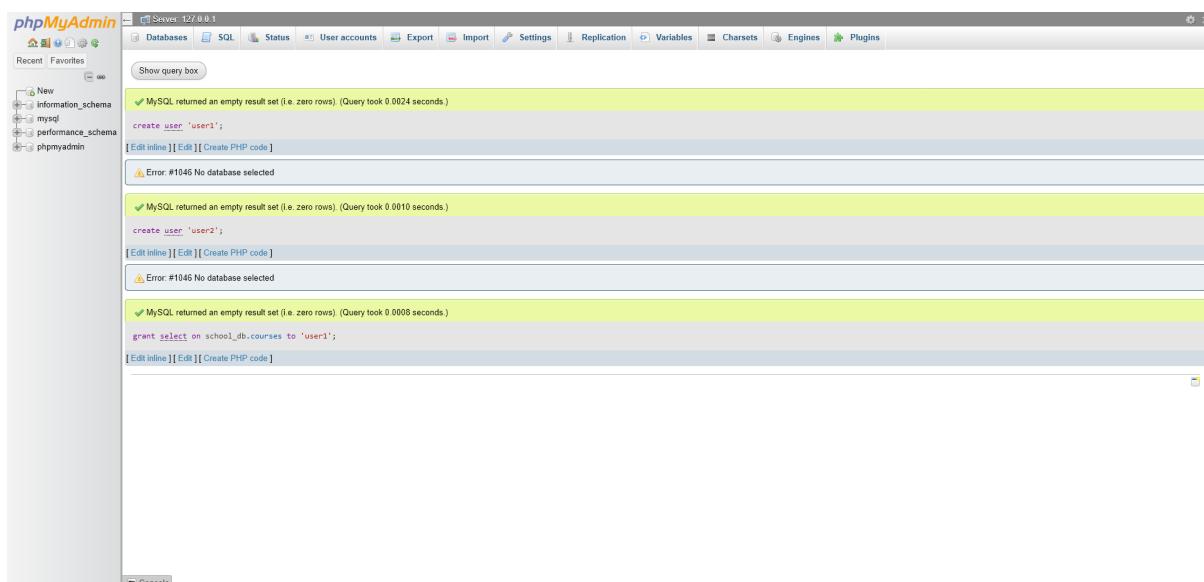
## 9 Data Control Language (DCL)

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

```

1  create user 'user1';
2
3  create user 'user2';
4
5  grant select on school_db.courses
6  to 'user1';

```

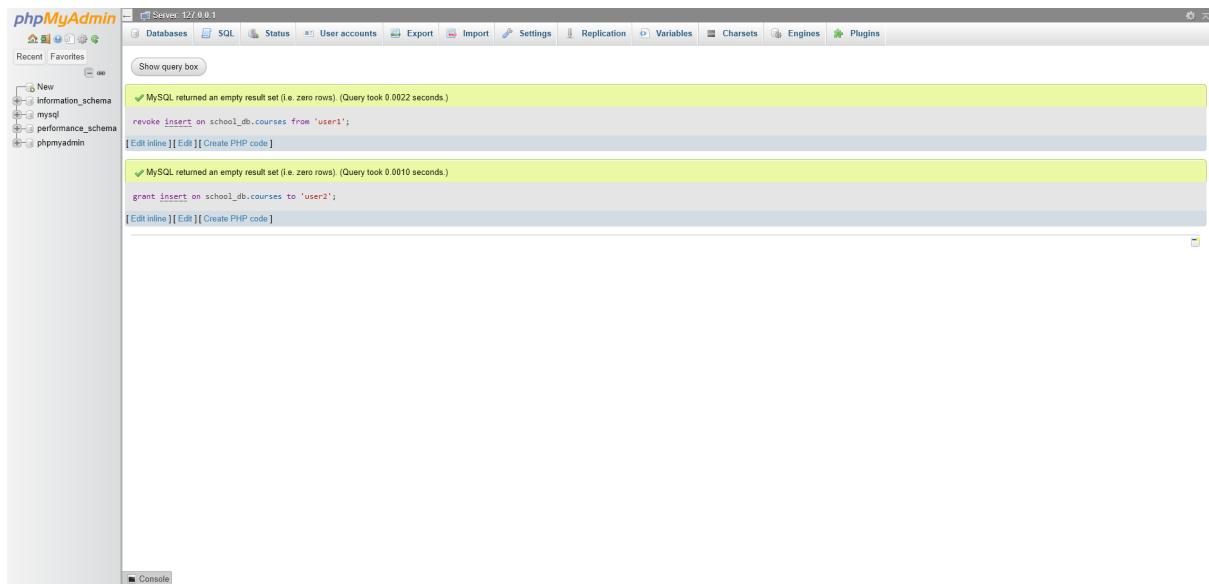


### 9.2 Revoke the **INSERT** permission from **user1** and give it to **user2**.

```

1  revoke insert on school_db.courses
2  from 'user1';
3
4  grant insert on school_db.courses
5  to 'user2';

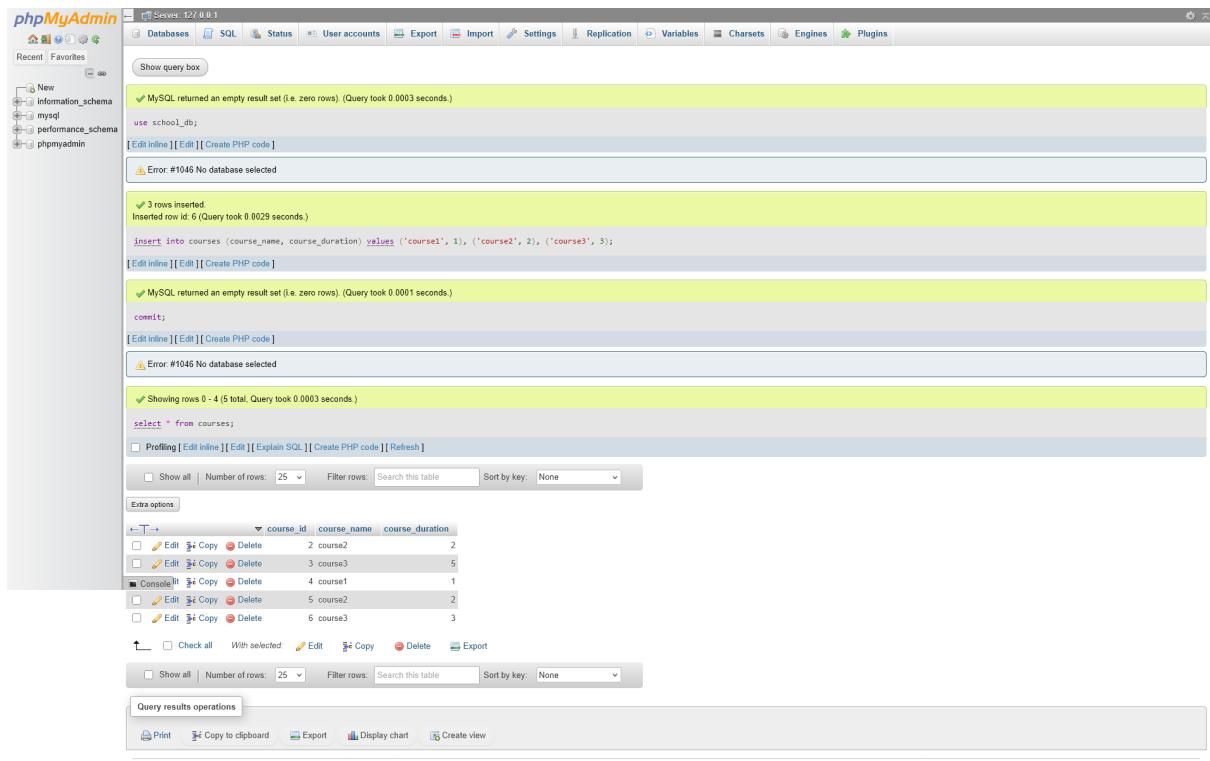
```



## 10 Transaction Control Language (TCL)

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

```
1 use school_db;
2
3 insert into courses (course_name, course_duration) values
4     ('course1', 1),
5     ('course2', 2),
6     ('course3', 3);
7
8 commit;
9
10 select * from courses;
```

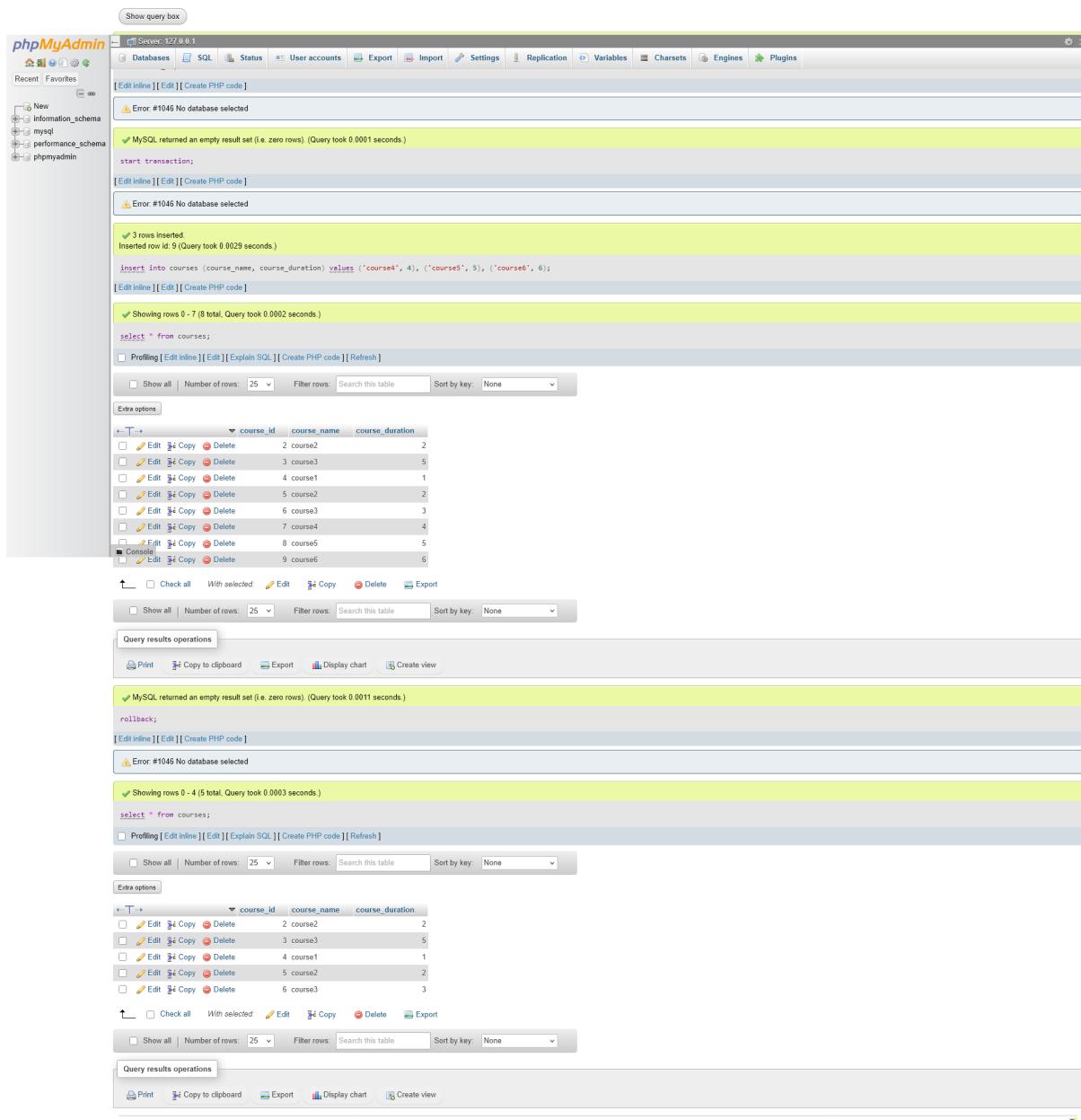


## 10.2 Insert additional rows, then use ROLLBACK to undo the last insert operation.

```

1 use school_db;
2
3 start transaction;
4
5 insert into courses (course_name, course_duration) values
6   ('course4', 4),
7   ('course5', 5),
8   ('course6', 6);
9
10 select * from courses;
11
12 rollback;
13
14 select * from courses;

```



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

```

1 use school_db;
2
3 start transaction;
4
5 delete from courses
6 where course_id=5;
7
8 select * from courses;
9
10 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;
```

Screenshot of the MySQL Workbench interface showing three separate sessions. The first session shows the execution of SQL queries to delete rows from the 'courses' table where course\_id is 5 and 6, and then select all rows from the table. The second session shows the execution of a savepoint command and then delete rows where course\_id is 6. The third session shows the execution of a commit command.

```

use school_db;
start transaction;
delete from courses where course_id=5;
select * from courses;

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds)

use school_db;
Error #1046 No database selected

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds)

start transaction;
Error #1046 No database selected

1 row deleted. (Query took 0.0008 seconds)

delete from courses where course_id=5;
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

Edit Copy Delete 2 course2 2
Edit Copy Delete 3 course3 5
Edit Copy Delete 4 course1 1

course_id course_name course_duration

Up | 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;
Error #1046 No database selected

1 row deleted. (Query took 0.0002 seconds)

delete from courses where course_id=6;
Error #1046 No database selected

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
Edit Copy Delete 4 course1 1

course_id course_name course_duration

Up | 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;
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.

```
1 use school_db;
2
3 create table departments (
4     department_id int primary key auto_increment,
5     department_name varchar(255)
6 );
7
8 create table employees (
9     employee_id int primary key auto_increment,
10    employee_name varchar(255),
11    employee_salary int,
12    department_id int
13 );
14
15 insert into departments (department_name) values
16 ('department1'),
17 ('department2'),
18 ('department3'),
19 ('department4');
20
21 insert into employees (employee_name, employee_salary, department_id
22 ) values
23 ('employee1', 10000, 1),
24 ('employee2', 20000, 2),
25 ('employee3', 30000, 2),
26 ('employee4', 40000, 3),
27 ('employee5', 50000, 3),
28 ('employee6', 60000, 3);
29
30 select employee_id, employee_name, employee_salary, department_name
31 from employees
32 inner join departments
33 on employees.department_id = departments.department_id;
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'school\_db'. The left sidebar lists databases: New, information\_schema, mysql, performance\_schema, and phpmyadmin. The main area shows the SQL tab with the following queries:

```

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;

```

The results pane shows the data inserted into the 'employees' table:

employee_id	employee_name	employee_salary	department_name
1	employee1	10000	department1
2	employee2	20000	department2
3	employee3	30000	department2
4	employee4	40000	department3
5	employee5	50000	department3
6	employee6	60000	department3

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

```

1 use school_db;
2
3 select departments.department_id, department_name, employee_id,
4       employee_name, employee_salary
5   from departments
6  left join employees
7    on departments.department_id = employees.department_id;

```

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Databases:** Information\_schema, mysql, performance\_schema, phpmyadmin
- Query:** MySQL returned an empty result set (i.e. zero rows) (Query took 0.0002 seconds)
- SQL:** `use school_db;`
- PHP:** `[Edit inline] [Edit] [Create PHP code]`
- Warning:** Error #1046 No database selected
- Info:** Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
- Result:** Showing rows 0 - 6 (7 total). Query took 0.0020 seconds.
 

department_id	department_name	employee_id	employee_name	employee_salary
1	department1	1	employee1	10000
2	department2	2	employee2	20000
2	department2	3	employee3	30000
3	department3	4	employee4	40000
3	department3	5	employee5	50000
3	department3	6	employee6	60000
4	department4	NULL	NULL	NULL
- Operations:** Show all, Number of rows: 25, Filter rows, Search this table, Sort by key: None
- Extra options:** Extra options button
- Query results operations:** Print, Copy to clipboard, Export, Display chart, Create view
- Console:** Console button

## 12 SQL Group By

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

```

1 use school_db;
2
3 select count(employee_id), department_id
4 from employees
5 group by department_id;

```

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Databases:** Information\_schema, mysql, performance\_schema, phpmyadmin
- Query:** MySQL returned an empty result set (i.e. zero rows) (Query took 0.0003 seconds)
- SQL:** `use school_db;`
- PHP:** `[Edit inline] [Edit] [Create PHP code]`
- Warning:** Error #1046 No database selected
- Info:** Showing rows 0 - 2 (3 total). Query took 0.0092 seconds.
 

count(employee_id)	department_id
1	1
2	2
3	3
- Operations:** Show all, Number of rows: 25, Filter rows, Search this table, Sort by key: None
- Extra options:** Extra options button
- Query results operations:** Print, Copy to clipboard, Export, Display chart, Create view
- Console:** Console button

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

```

1 use school_db;
2
3 select avg(employee_salary), department_id
4   from employees
5  group by department_id;

```

The screenshot shows the phpMyAdmin interface. In the left sidebar, the database 'school\_db' is selected. The main area contains the following SQL query:

```

use school_db;

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

```

The results pane shows a table with three rows of data:

	avg(employee_salary)	department_id
<input type="checkbox"/>	10000.0000	1
<input type="checkbox"/>	25000.0000	2
<input type="checkbox"/>	50000.0000	3

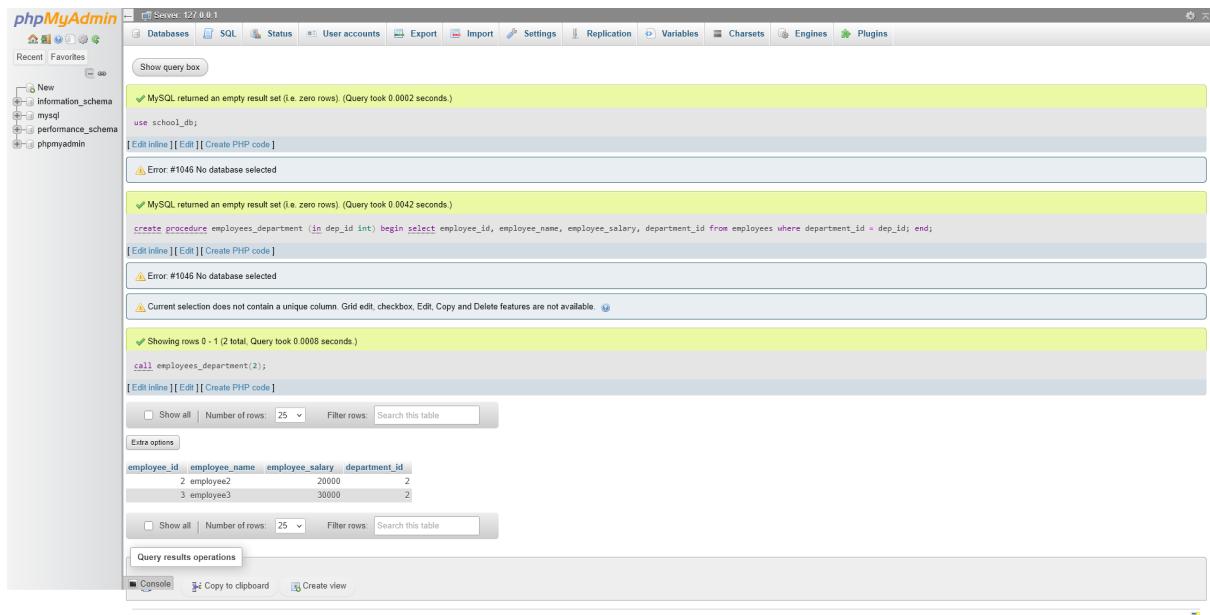
## 13 SQL Stored Procedure

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

```

1 use school_db;
2
3 delimiter $$ 
4
5 create procedure employees_department (in dep_id int)
6 begin
7     select employee_id, employee_name, employee_salary,
8         department_id
9     from employees
10    where department_id = dep_id;
11 end $$ 
12
13 delimiter ;
14
15 call employees_department(2);

```



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

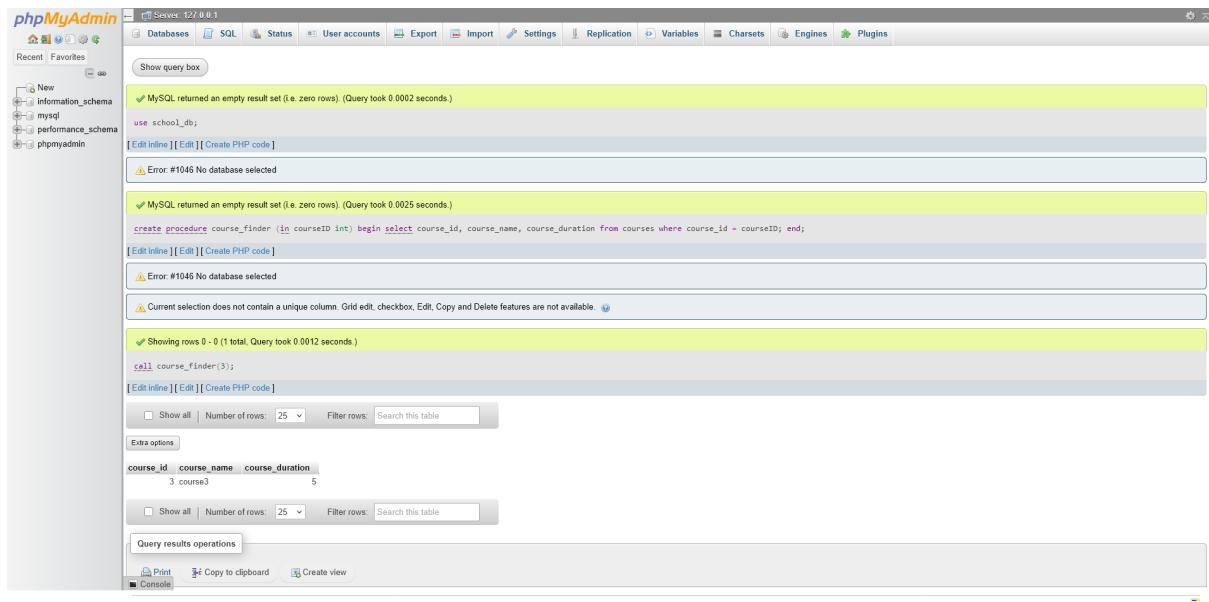
```

1 use school_db;
2
3 delimiter $$

4
5 create procedure course_finder (in courseID int)
6 begin
7     select course_id, course_name, course_duration
8     from courses
9     where course_id = courseID;
10 end $$

11
12 delimiter ;
13
14
15 call course_finder(3);

```



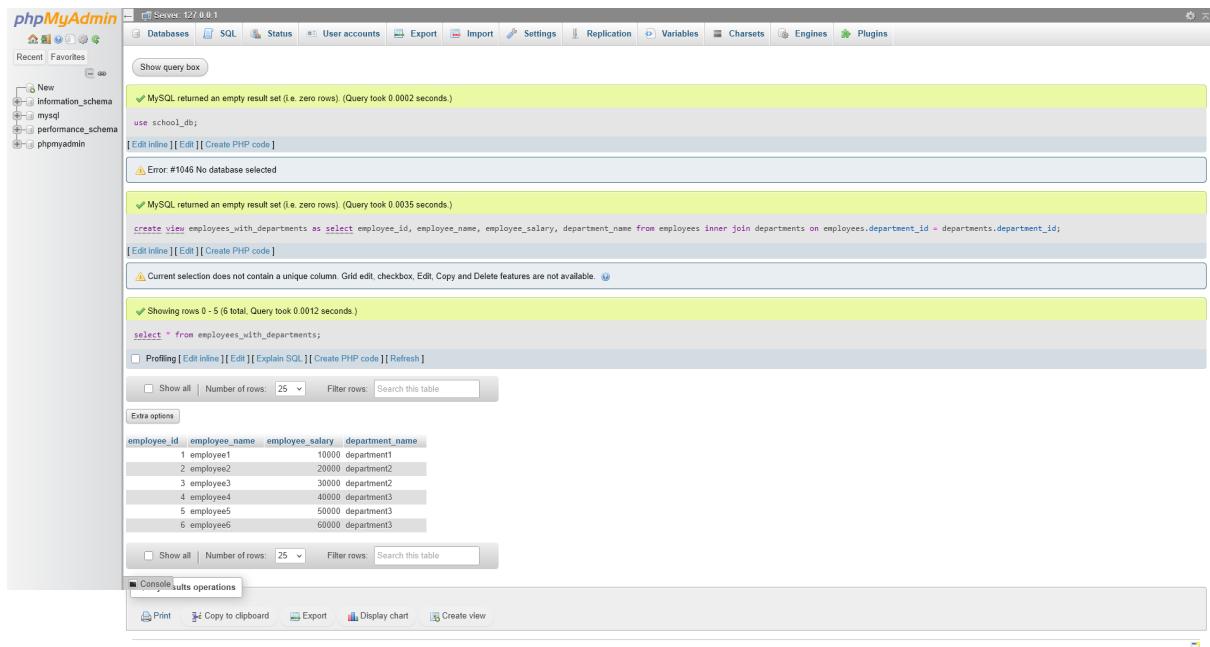
## 14 SQL View

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

```

1 use school_db;
2
3 create view employees_with_departments as
4 select employee_id, employee_name, employee_salary, department_name
5 from employees
6 inner join departments
7 on employees.department_id = departments.department_id;
8
9
10 select * from employees_with_departments;

```

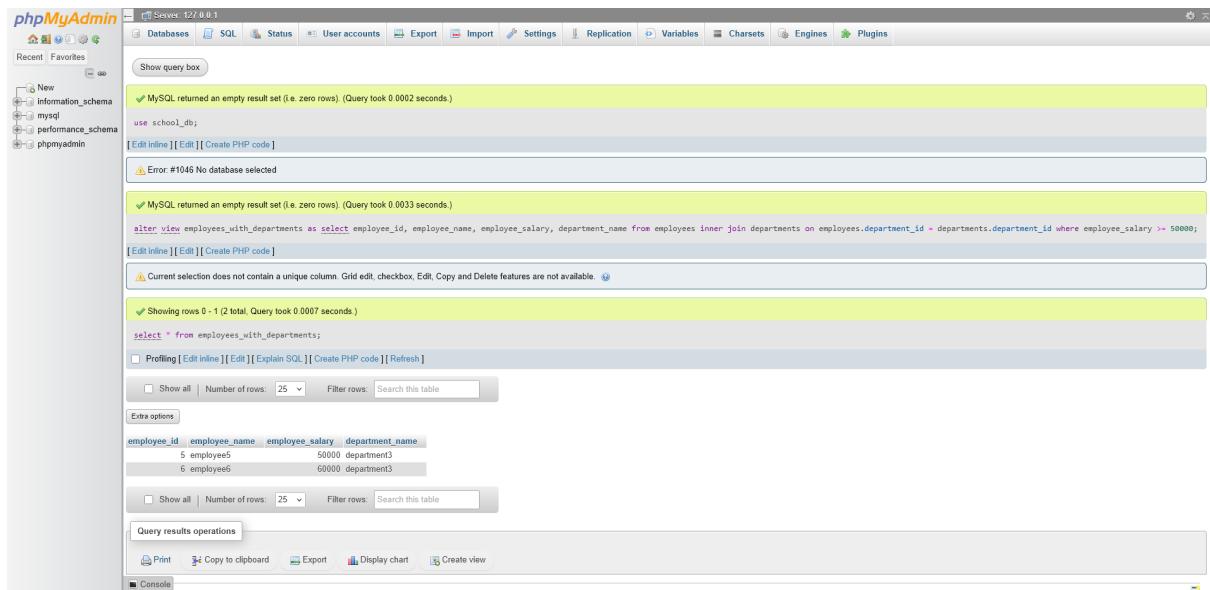


## 14.2 Modify the view to exclude employees whose salaries are below \$50,000.

```

1 use school_db;
2
3 alter view employees_with_departments as
4 select employee_id, employee_name, employee_salary, department_name
5 from employees
6 inner join departments
7 on employees.department_id = departments.department_id
8 where employee_salary >= 50000;
9
10
11 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.

```

1 use school_db;
2
3 create table employee_log (
4     log_id int,
5     log_name varchar(255),
6     log_salary int,
7     log_dep_id int,
8     action_time timestamp,
9     action_performed varchar(255)
10);
11
12 select * from employee_log;
13
14 delimiter $$ 
15 create trigger log_insert
16 after insert on employees
17 for each row
18 begin
19     insert into log (log_id, log_name, log_salary, log_dep_id,
20                     action_performed)
21     values (new.employee_id, new.employee_name, new.employee_salary,
22             new.department_id, "insert");
23 end $$ 
24 delimiter ;

```



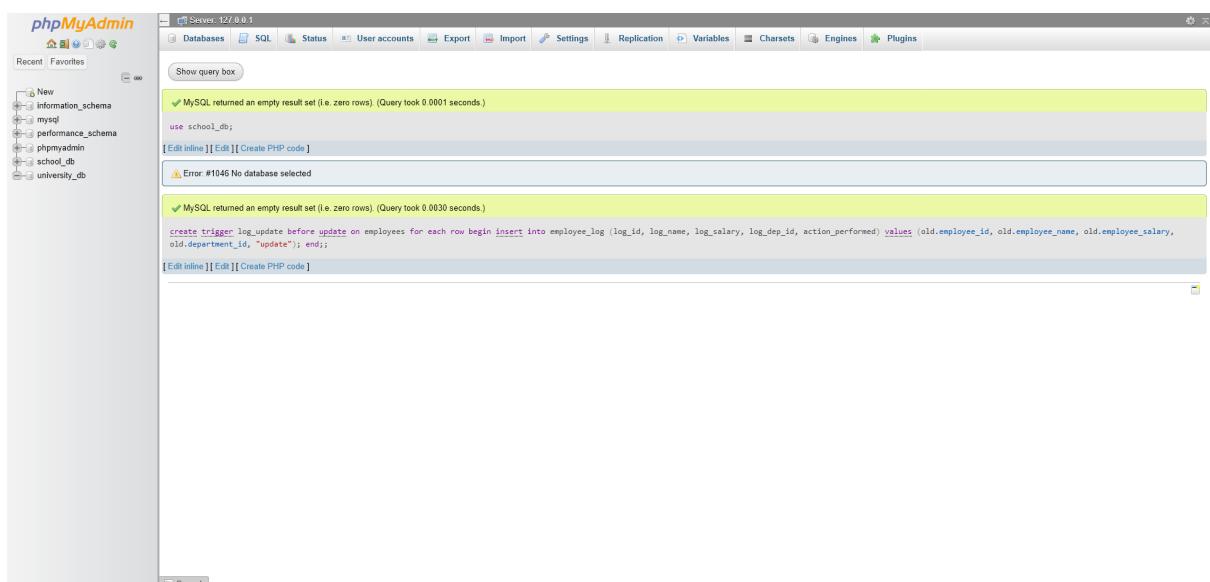
## 15.2 Create a trigger to update the `last_modified` timestamp whenever an employee record is updated.

```

1 use school_db;
2
3 delimiter $$
4 create trigger log_update
5 before update on employees
6 for each row
7 begin
8     insert into employee_log (log_id, log_name, log_salary,
9     log_dep_id, action_performed)
10    values (old.employee_id, old.employee_name, old.employee_salary,
11            old.department_id, "update");
12 end; $$

13 delimiter ;

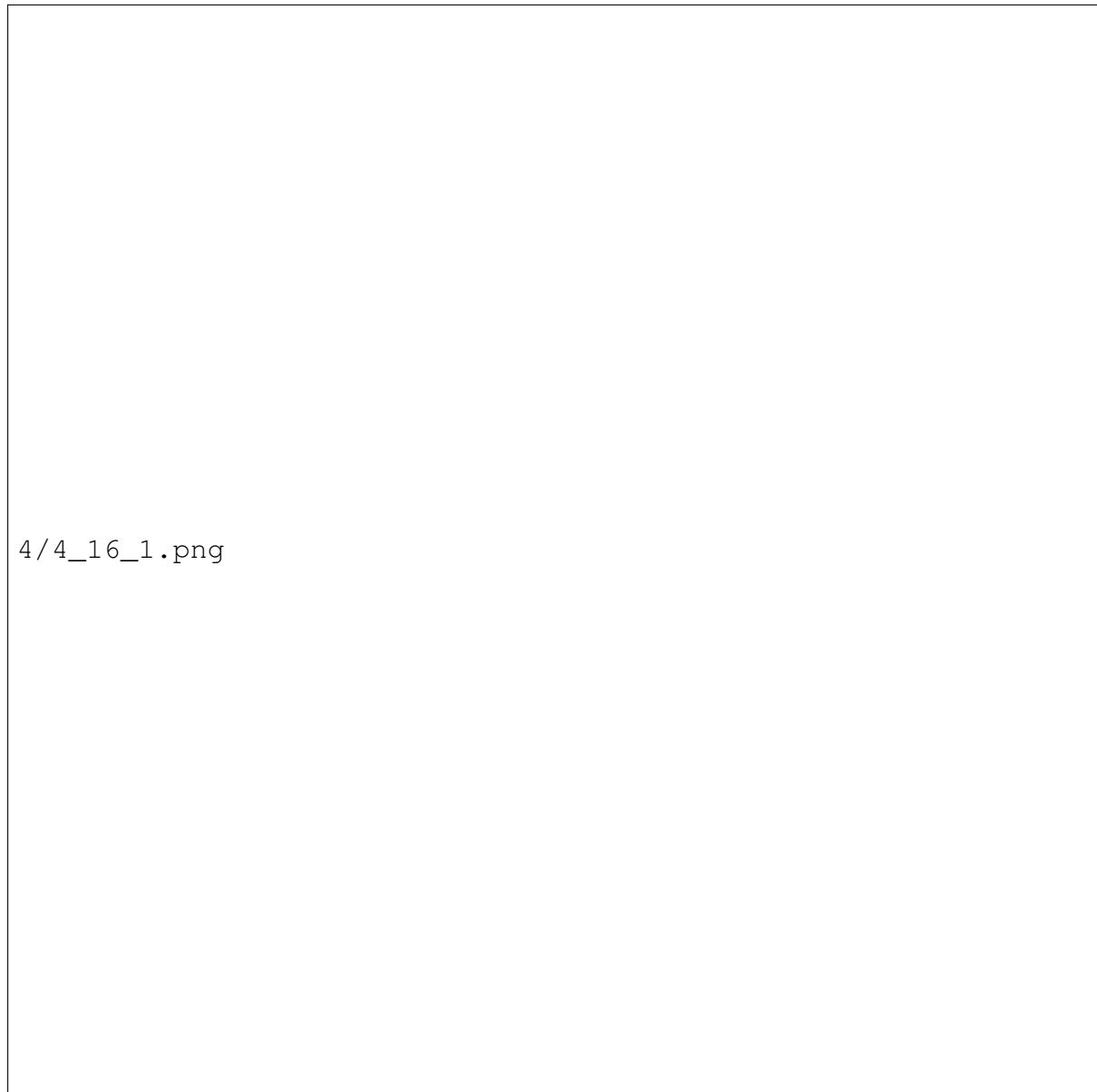
```



## 16 Introduction to PL/SQL

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

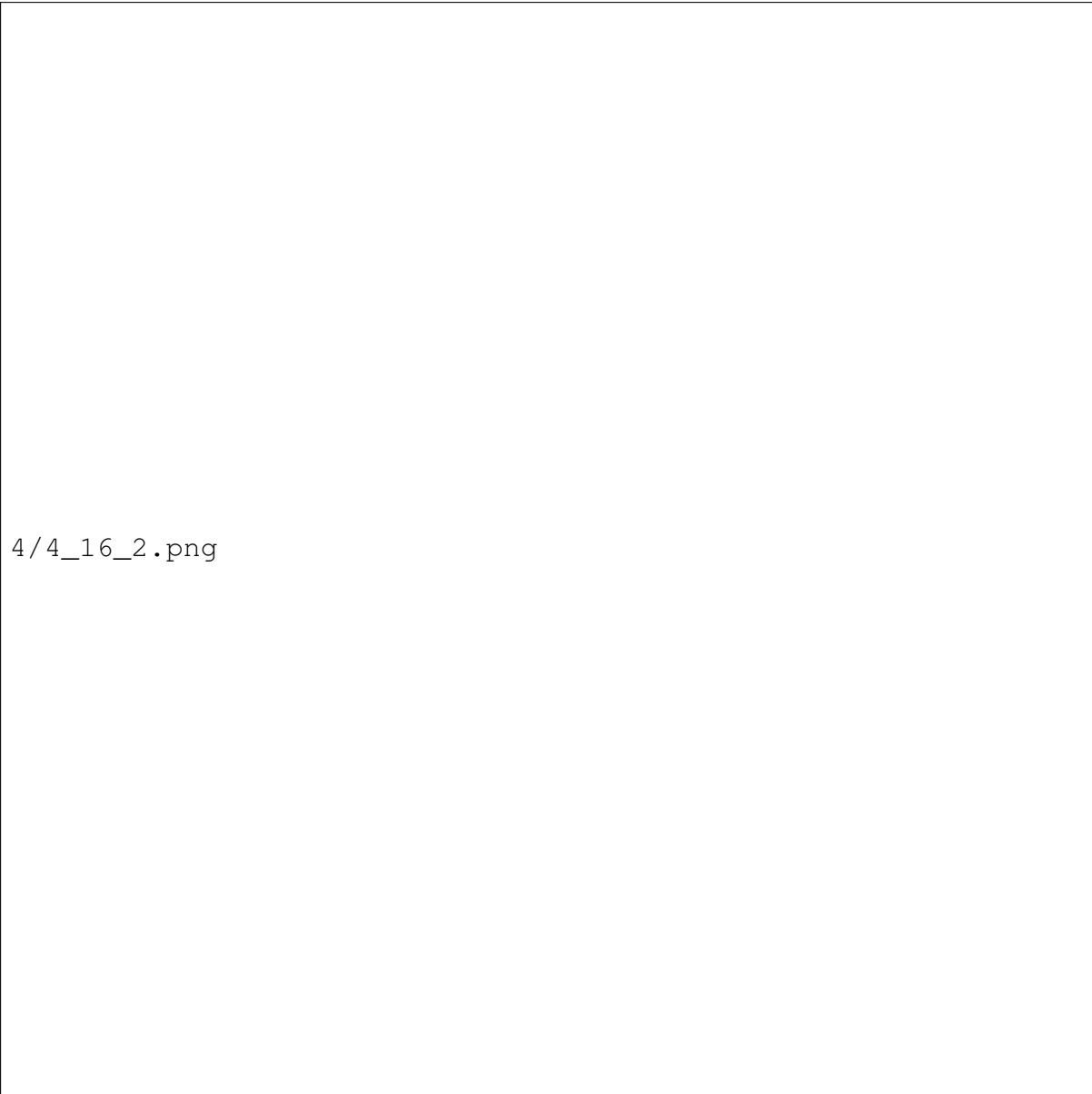
---



4/4\_16\_1.png

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

---

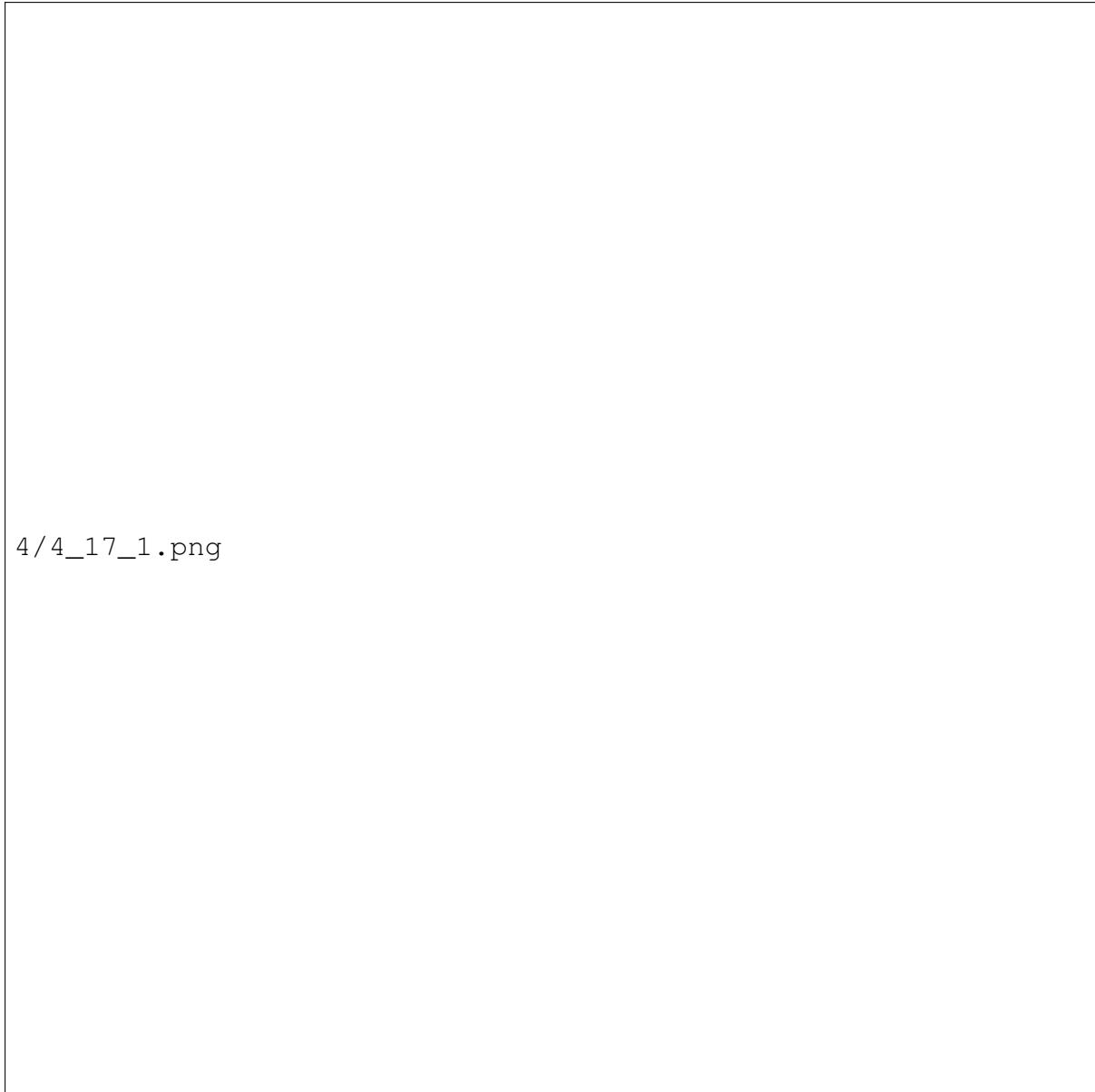


4 / 4 \_16\_2 .png

## 17 PL/SQL Control Structures

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

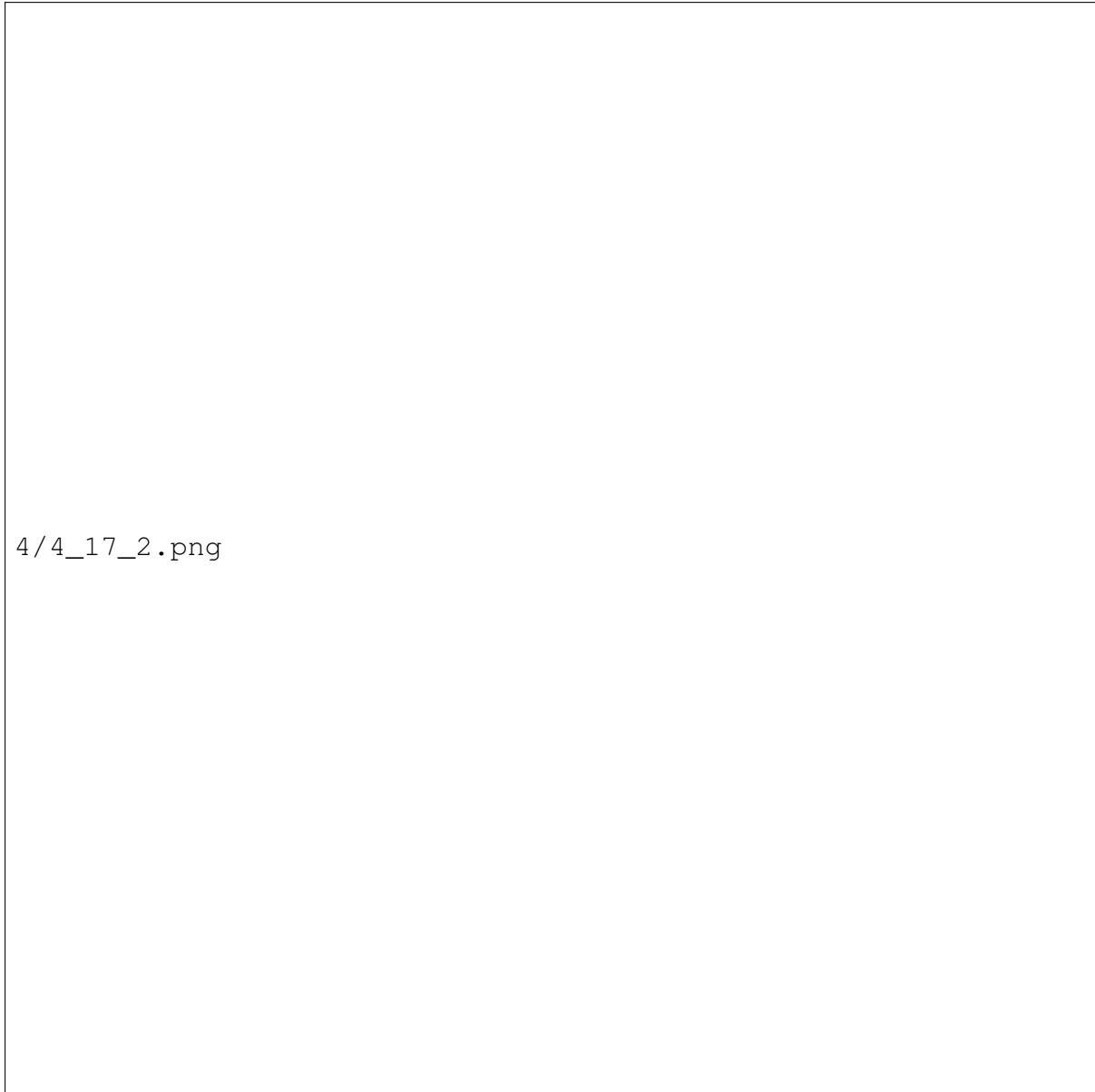
---



4 / 4 \_17\_1 .png

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

---

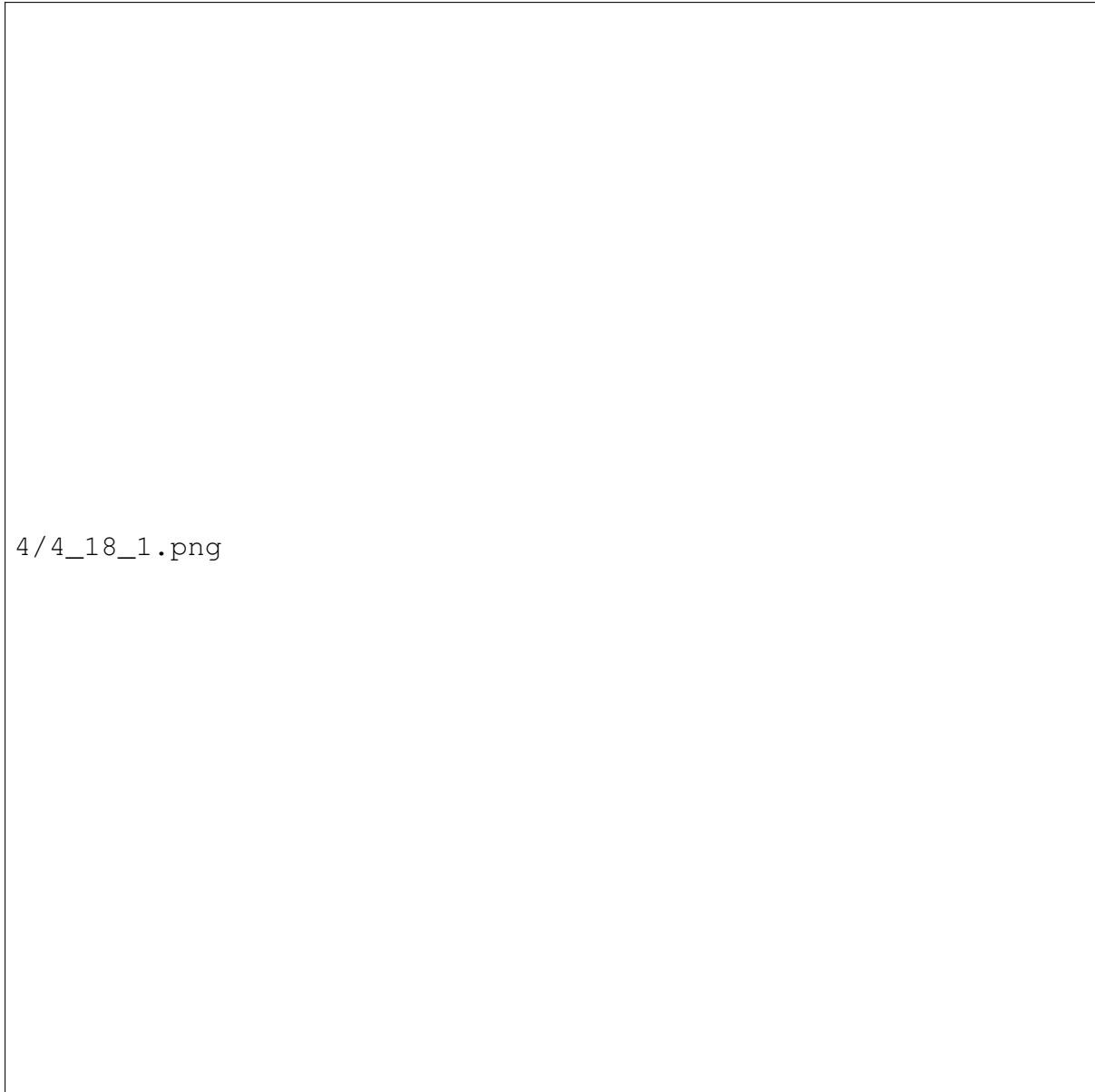


4 / 4 \_17\_2 .png

## 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?

```
1 use school_db;
2
3 start transaction;
4
5 savepoint sav_pt;
6
7 insert into courses (course_name, course_duration) values
8     ("course4", 4),
9     ("course5", 5);
10
```

```

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

```

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

- Query Editor:** Contains the transaction script provided in the code block.
- Results Pane:** Shows three rows of output from the MySQL server, indicating empty result sets and errors related to database selection.
- Table Browser:** Displays the 'courses' table with the following data:
 

course_id	course_name	course_duration
2	course2	2
3	course3	5
4	course1	1
6	course3	3
10	course4	4
- Query Results Operations:** Includes options like Print, Copy to clipboard, Export, Display chart, and Create view.
- phpMyAdmin Screenshot:** Shows the same transaction script being run in phpMyAdmin, resulting in the same empty result sets and database selection errors.

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

```
1 use school_db;
2
3 start transaction;
4
5 update courses
6 set course_name = "course4"
7 where course_id = 6;
8
9 select * from courses;
10
11 savepoint sav_pt2;
12
13 update courses
14 set course_duration = 4
15 where course_id = 6;
16
17 select * from courses;
18
19 rollback to sav_pt2;
20
21 select * from courses;
22
23 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
```

Show query box

```
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
```

Show query box

```
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;
```

Profile [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	course_id	course_name	course_duration
<input type="checkbox"/>				2	course2	2
<input type="checkbox"/>				3	course3	5
<input type="checkbox"/>				4	course1	1

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

Show query box

```
MySQL returned an empty result set (i.e. zero rows) (Query took 0.0001 seconds)

savpoint sav_pt2;
[Edit inline] [Edit] [Create PHP code]

Error #1046 No database selected
```

Show query box

```
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;
```

Profile [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	course_id	course_name	course_duration
<input type="checkbox"/>				2	course2	2
<input type="checkbox"/>				3	course3	5
<input type="checkbox"/>				4	course1	1

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 4.6.6 - MySQL interface

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication VariablesCharsets Engines Plugins

Error #1046 No database selected

Show query box

```
Showing rows 0 - 3 (4 total). Query took 0.0001 seconds.

select * from courses;
```

Profile [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	course_id	course_name	course_duration
<input type="checkbox"/>				2	course2	2
<input type="checkbox"/>				3	course3	5
<input type="checkbox"/>				4	course1	1
<input type="checkbox"/>				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

Show query box

```
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