

# **Module 5 – Introduction to DBMS**

(lab-ex)

-BARAD VIPUL

# Introduction to SQL

Lab 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.

```
CREATE TABLE students(student_id int PRIMARY KEY AUTO_INCREMENT, student_name varchar(100), age int, class int, address varchar(200))
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0037 seconds.)

```
CREATE TABLE students(student_id int PRIMARY KEY AUTO_INCREMENT, student_name varchar(100), age int, class int, address varchar(200));
```

[Edit inline](#) | [Edit](#) | [Create PHP code](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	student_id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 2	student_name	varchar(100)	utf8mb4_general_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 3	age	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 4	class	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 5	address	varchar(200)	utf8mb4_general_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

## Lab 2: Insert five records into the students table and retrieve all records using the SELECT statement.





















Show query box

✓ 5 rows inserted.

Inserted row id: 5 (Query took 0.0046 seconds.)

```
INSERT into students(student_id,student_name,age,class,address) VALUES (null,'harsh',22,'3','rajkot'),(null,'anuradha',19,'1','gondal'),(null,'vedisha',22,'3','morbi'),(null,'sayan',23,'2','bhavnagar'),(null,'nibhay',20,'2','godhara');
```


[\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Create PHP code \]](#)

			student_id	student_name	age	class	address
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	harsh	22	3 rajkot
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	anuradha	19	1 gondal
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	vedisha	22	3 morbi
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	sayan	23	2 bhavnagar
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	nibhay	20	2 godhara
	<input type="checkbox"/> Check all	With selected:		 Edit	 Copy	 Delete	 Export

# SQL Syntax

Lab 1: Write SQL queries to retrieve specific columns (student\_name and age) from the students table.

```
SELECT student_name,age FROM `students`
```
















 Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

```
SELECT student_name,age FROM `students`;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows:  Filter rows:  Sort by key:

Extra options

				student_name	age
<input type="checkbox"/>	 Edit	 Copy	 Delete	harsh	22
<input type="checkbox"/>	 Edit	 Copy	 Delete	anuradha	19
<input type="checkbox"/>	 Edit	 Copy	 Delete	vedisha	22
<input type="checkbox"/>	 Edit	 Copy	 Delete	sayan	23
<input type="checkbox"/>	 Edit	 Copy	 Delete	nibhay	20

- Lab 2: Write SQL queries to retrieve all students whose age is greater than 10.

SELECT age FROM `students` WHERE age>10

Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

```
SELECT age FROM `students` WHERE age>10;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows:  Filter rows:  Sort by key:

Extra options

	age
<input type="checkbox"/> Edit  Copy  Delete	22
<input type="checkbox"/> Edit  Copy  Delete	19
<input type="checkbox"/> Edit  Copy  Delete	22
<input type="checkbox"/> Edit  Copy  Delete	23
<input type="checkbox"/> Edit  Copy  Delete	20

☐ Check all With selected: Edit Copy Delete Export

# SQL Constraints

- Lab 1: Create a table teachers with the following columns: teacher\_id (Primary Key), teacher\_name (NOT NULL), subject (NOT NULL), and email (UNIQUE).

```
CREATE TABLE teachers(teacher_id int Primary Key  
AUTO_INCREMENT,teacher_name varchar(100) NOT  
NULL,subject varchar(100) NOT NULL, email  
varchar(100) UNIQUE)
```

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0011 seconds.)

```
CREATE TABLE teachers(teacher_id int Primary Key AUTO_INCREMENT,teacher_name varchar(100) NOT NULL,subject varchar(100) NOT NULL, email varchar(100) UNIQUE);
```

[ Edit inline ] [ Edit ] [ Create PHP code ]



Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 teacher_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 teacher_name	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 subject	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 email	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext  
 Add to central columns Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after email Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit  Rename  Drop	PRIMARY	BTREE	Yes	No	teacher_id	0	A	No	
Edit  Rename  Drop	email	BTREE	Yes	No	email	0	A	Yes	

- Lab 2: Implement a FOREIGN KEY constraint to relate the teacher\_id from the teachers table with the students table.

```
ALTER TABLE students ADD teacher_id int;  
ALTER TABLE students ADD FOREIGN KEY  
(teacher_id) REFERENCES teachers(teacher_id);
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
ALTER TABLE students ADD teacher_id int;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
ALTER TABLE students ADD FOREIGN KEY (teacher_id) REFERENCES teachers(teacher_id);
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	student_id	int(11)		No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	student_name	varchar(100) utf8mb4_general_ci	Yes	NULL				Change  Drop  More
<input type="checkbox"/>	3	age	int(11)	Yes	NULL				Change  Drop  More
<input type="checkbox"/>	4	class	int(11)	Yes	NULL				Change  Drop  More
<input type="checkbox"/>	5	address	varchar(200) utf8mb4_general_ci	Yes	NULL				Change  Drop  More
<input type="checkbox"/>	6	teacher_id	int(11)	Yes	NULL				Change  Drop  More

☐ Check all
 With selected:
 Browse
 Change
 Drop
 Primary
 Unique
 Index
 Spatial
 f

Add to central columns
 Remove from central columns

Print
 Propose table structure
 Track table
 Move columns
 Normalize

Add
 
 column(s)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit  Rename  Drop	PRIMARY	BTREE	Yes	No	student_id	5	A	No	
Edit  Rename  Drop	teacher_id	BTREE	No	No	teacher_id	2	A	Yes	

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

```
SELECT * FROM `students`
```

☐ Profiling
 [ [Edit inline](#) ]
 [ [Edit](#) ]
 [ [Explain SQL](#) ]
 [ [Create PHP code](#) ]
 [ [Refresh](#) ]

☐ Show all
 Number of rows: 
 Filter rows: 
 Sort by key:

Extra options

student\_id

student\_name

age

class

address

teacher\_id

<input type="checkbox"/>	Edit  Copy  Delete	1	harsh	22	3	rajkot	NULL
<input type="checkbox"/>	Edit  Copy  Delete	2	anuradha	19	1	gondal	NULL
<input type="checkbox"/>	Edit  Copy  Delete	3	vedisha	22	3	morbi	NULL
<input type="checkbox"/>	Edit  Copy  Delete	4	sayan	23	2	bhavnagar	NULL
<input type="checkbox"/>	Edit  Copy  Delete	5	nibhay	20	2	godhara	NULL

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export



# Main SQL Commands and Sub-commands (DDL)

- Lab 1: Create a table courses with columns: course\_id, course\_name, and course\_credits. Set the course\_id as the primary key.

```
CREATE TABLE courses (course_id int PRIMARY KEY, course_name varchar(100), course_credits varchar(100))
```

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
CREATE TABLE courses (course_id int PRIMARY KEY, course_name varchar(100), course_credits varchar(100));
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

The screenshot displays the phpMyAdmin web interface. On the left is a sidebar with a database tree showing 'school\_db' and its tables: 'courses', 'students', and 'teachers'. The main panel is titled 'Table: courses' and shows the 'Table structure' view. It lists three columns: 'course\_id' (int(11), PRIMARY KEY), 'course\_name' (varchar(100)), and 'course\_credits' (varchar(100)). Below the column list are various table management options like 'Check all', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. At the bottom, the 'Indexes' section shows a PRIMARY index on the 'course\_id' column.

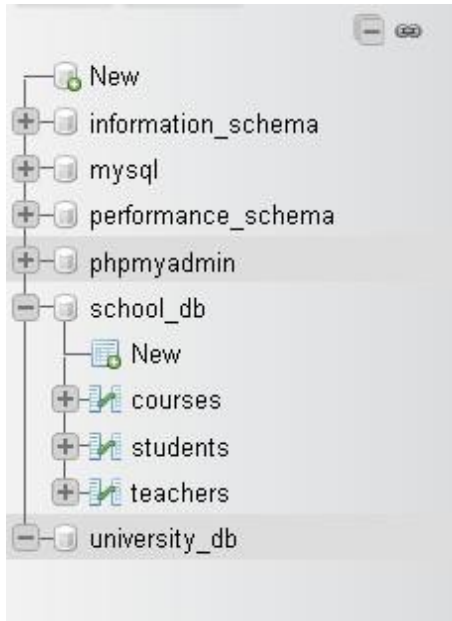
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	course_id	int(11)			No	None			Change Drop More
2	course_name	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	course_credits	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	course_id	0	A	No	

- Lab 2: Use the CREATE command to create a database university\_db.

CREATE DATABASE university\_db



# ALTER Command

- Lab 1: Modify the courses table by adding a column `course_duration` using the ALTER command.

ALTER TABLE courses add course\_duration varchar(100)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0011 seconds.)

```
ALTER TABLE courses add course_duration varchar(100);
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]



Table structure



Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>course_id</b>	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	2 <b>course_name</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	3 <b>course_credits</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	4 <b>course_duration</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More



☐ Check all

With selected:

Browse

Change

Drop

Primary

Unique

Index

Spatial

Fulltext



Add to central columns



Remove from central columns

- Lab 2: Drop the `course_credits` column from the `courses` table.

**ALTER TABLE** `courses` **DROP** `course_credits`

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0013 seconds.)

```
ALTER TABLE courses DROP course_credits;
```

[\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Create PHP code \]](#)

Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Tr

Table structure
Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>course_id</b>	int(11)			No	None			Change  Drop  More
<input type="checkbox"/> 2	<b>course_name</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 3	<b>course_duration</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More

☐ Check all
With selected:
Browse
 Change
 Drop
 Primary
 Unique
 Index
 Spatial
 Fulltext

Add to central columns
 Remove from central columns

# DROP Command

- Lab 1: Drop the teachers table from the school\_db database.
- Lab 2: Drop the students table from the school\_db database and verify that the table has been removed.

First we have to remove constraint key from tables

For that first we have to know what is name of key and for that go to `information_schema` database then go to `KEY_COLUMN_USAGE`

And find 'students' in `table_name` column and 'teacher\_id' in `column_name` column.

This is where you can see the name of the constraint key name in the `constraint_name` column.

```
ALTER TABLE students DROP CONSTRAINT  
students_ibfk_1;
```

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0013 seconds.)

```
ALTER TABLE students DROP CONSTRAINT students_ibfk_1;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

drop TABLE teachers;  
drop TABLE students

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
drop TABLE teachers;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

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

```
drop TABLE students;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]










The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a tree view of databases: information\_schema, mysql, performance\_schema, phpmyadmin, school\_db, and university\_db. The 'school\_db' database is selected, and the 'courses' table is highlighted. The main panel displays the 'Structure' tab for the 'courses' table. It shows a single column named 'courses' of type 'InnoDB' with 'utf8mb4\_general\_ci' collation, a size of 16.0 KiB, and no overhead. Below the table structure, there are options to 'Check all' and a 'With selected:' dropdown menu. The top navigation bar includes links for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, and Events.






# Data Manipulation Language (DML)

the INSERT command.

```
INSERT INTO courses
(course_id,course_name,course_duration) VALUES
(1001,'soft dev','8 months'),(1002,'web dev','6
months'),(1003,'flutter','10 months')
```

Extra options

				course_id	course_name	course_duration
<input type="checkbox"/>	 Edit	 Copy	 Delete	1001	soft dev	8 months
<input type="checkbox"/>	 Edit	 Copy	 Delete	1002	web dev	6 months
<input type="checkbox"/>	 Edit	 Copy	 Delete	1003	flutter	10 months

 ☐ Check all    With selected:  Edit     Copy     Delete     Export

- Lab 2: Update the course duration of a specific course using the UPDATE command.

UPDATE courses SET course\_duration = '9 months'  
WHERE course\_id = 1003;

☐ Show all | Number of rows: 25 ▾ | Filter rows:  | Sort by

Extra options

				course_id	course_name	course_duration
<input type="checkbox"/>	Edit	Copy	Delete	1001	soft dev	8 months
<input type="checkbox"/>	Edit	Copy	Delete	1002	web dev	6 months
<input type="checkbox"/>	Edit	Copy	Delete	1003	flutter	9 months

☐ Check all | With selected: Edit Copy Delete Export



- Lab 3: Delete a course with a specific course\_id from the courses table using the DELETE command.

delete from courses where course\_id=1002

Show all		Number of rows: 23	Filter rows: Search this table	Sort by key: No
Extra options				
		course_id	course_name	course_duration
<input type="checkbox"/>	Edit  Copy  Delete	1001	soft dev	8 months
<input type="checkbox"/>	Edit  Copy  Delete	1003	flutter	9 months
	<input type="checkbox"/> Check all	With selected:  Edit  Copy  Delete  Export		

# Data Query Language (DQL)

- Lab 1: Retrieve all courses from the courses table using the SELECT statement.

```
SELECT * FROM `courses`
```

The screenshot shows a database management interface with a top navigation bar containing tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, and Operations. The SQL tab is active, displaying a green status bar that reads "Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)". Below this, the SQL query `SELECT * FROM `courses`;` is entered. A toolbar offers options: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. A control bar includes a "Show all" checkbox, "Number of rows" set to 25, a "Filter rows" search box, and a "Sort by key" dropdown set to "None". An "Extra options" button is also present. The results are shown in a table with columns `course_id`, `course_name`, and `course_duration`. Two rows are visible: (1001, soft dev, 8 months) and (1003, flutter, 9 months). Each row has checkboxes and icons for Edit, Copy, and Delete. At the bottom, a summary bar shows "Check all" and "With selected:" options, along with Edit, Copy, Delete, and Export actions. A second control bar at the very bottom mirrors the one above.

	course_id	course_name	course_duration
<input type="checkbox"/>	1001	soft dev	8 months
<input type="checkbox"/>	1003	flutter	9 months

- Lab 2: Sort the courses based on course\_duration in descending order using ORDER BY.

```
SELECT * FROM `courses` ORDER BY course_duration  
DESC
```

✓ Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.) [course\_duration: 9 MONTHS... - 8 MONTHS...]

```
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:  Sort by key: None

Extra options

			course_id	course_name	course_duration
<input type="checkbox"/>	Edit	Copy	Delete	1003 flutter	9 months
<input type="checkbox"/>	Edit	Copy	Delete	1001 soft dev	8 months

☐ Check all With selected: Edit Copy Delete Export

- Lab 3: Limit the results of the SELECT query to show only the top two courses using LIMIT.

Adding one entry just for more convenience

☐ Show all | Number of rows: 25 ▼ | Filter rows:  | Sort by k

Extra options

				course_id	course_name	course_duration
<input type="checkbox"/>	Edit	Copy	Delete	1001	soft dev	8 months
<input type="checkbox"/>	Edit	Copy	Delete	1002	web dev	5 months
<input type="checkbox"/>	Edit	Copy	Delete	1003	flutter	9 months

☐ Check all | With selected: Edit Copy Delete Export

SELECT \* FROM courses LIMIT 2

SELECT \* FROM courses LIMIT 2;

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

Extra options

				course_id	course_name	course_duration
<input type="checkbox"/>	Edit	Copy	Delete	1001	soft dev	8 months
<input type="checkbox"/>	Edit	Copy	Delete	1002	web dev	5 months

☐ Check all | With selected: Edit Copy Delete Export

Query results operations

# Data Control Language (DCL)

- Lab 1: Create two new users user1 and user2 and grant user1 permission to SELECT from the courses table.

```
CREATE USER 'user1'@'localhost' IDENTIFIED BY  
    'password1';
```

```
CREATE USER 'user2'@'localhost' IDENTIFIED BY  
    'password2';
```









```
GRANT SELECT ON school_db.courses TO  
    'user1'@'localhost';
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0066 seconds.)

```
GRANT SELECT ON school_db.courses TO 'user1'@'localhost';
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

## Users having access to "school\_db.courses"

	User name	Host name	Type	Privileges	Grant	Action
<input type="checkbox"/>	root	127.0.0.1	global	ALL PRIVILEGES	Yes	 <a href="#">Edit privileges</a>  <a href="#">Export</a>
<input type="checkbox"/>	root	:::1	global	ALL PRIVILEGES	Yes	 <a href="#">Edit privileges</a>  <a href="#">Export</a>
<input type="checkbox"/>	root	localhost	global	ALL PRIVILEGES	Yes	 <a href="#">Edit privileges</a>  <a href="#">Export</a>
<input type="checkbox"/>	user1	localhost	table-specific	USAGE	No	 <a href="#">Edit privileges</a>  <a href="#">Export</a>

 ☐ [Check all](#) With selected:  [Export](#)

..

Edit privileges: User account 'user1'@'localhost' - Databases school\_db - Table courses

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	
<div>course_id</div> <div>course_name</div> <div>course_duration</div>	<div>course_id</div> <div>course_name</div> <div>course_duration</div>	<div>course_id</div> <div>course_name</div> <div>course_duration</div>	<div>course_id</div> <div>course_name</div> <div>course_duration</div>	<div><input type="checkbox"/> DELETE</div> <div><input type="checkbox"/> CREATE</div> <div><input type="checkbox"/> DROP</div> <div><input type="checkbox"/> GRANT</div> <div><input type="checkbox"/> INDEX</div> <div><input type="checkbox"/> ALTER</div> <div><input checked="" type="checkbox"/> CREATE VIEW</div> <div><input type="checkbox"/> SHOW VIEW</div> <div><input type="checkbox"/> TRIGGER</div> <div><input type="checkbox"/> DELETE HISTORY</div>
Select all	Select all	Select all	Select all	
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	

- Lab 2: Revoke the INSERT permission from user1 and give it to user2.

```
REVOKE INSERT ON school_db.courses FROM  
    'user1'@'localhost';  
GRANT INSERT ON school_db.courses TO  
    'user2'@'localhost';  
FLUSH PRIVILEGES;
```

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0059 seconds.)

```
REVOKE INSERT ON school_db.courses FROM 'user1'@'localhost';
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)

```
GRANT INSERT ON school_db.courses TO 'user2'@'localhost';
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0018 seconds.)

```
FLUSH PRIVILEGES;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

**Users having access to "school\_db.courses"**

	User name	Host name	Type	Privileges	Grant	Action
<input type="checkbox"/>	root	127.0.0.1	global	ALL PRIVILEGES	Yes	Edit privileges            Export
<input type="checkbox"/>	root	::1	global	ALL PRIVILEGES	Yes	Edit privileges            Export
<input type="checkbox"/>	root	localhost	global	ALL PRIVILEGES	Yes	Edit privileges            Export
<input type="checkbox"/>	user1	localhost	table-specific	USAGE	No	Edit privileges            Export
<input type="checkbox"/>	user2	localhost	table-specific	USAGE	No	Edit privileges            Export

☐ Check all
 With selected: Export

New

Edit privileges: User account `'user2'@'localhost'` - Databases `school_db` - Table `courses`

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	<input type="checkbox"/> DELETE
<div>course_id course_name course_duration</div>	<div>course_id course_name course_duration</div>	<div>course_id course_name course_duration</div>	<div>course_id course_name course_duration</div>	<div><input type="checkbox"/> CREATE <input type="checkbox"/> DROP <input type="checkbox"/> GRANT <input type="checkbox"/> INDEX <input type="checkbox"/> ALTER <input type="checkbox"/> CREATE VIEW <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> TRIGGER <input type="checkbox"/> DELETE HISTORY</div>
Select all Or <input type="checkbox"/> None	Select all Or <input type="checkbox"/> None	Select all Or <input type="checkbox"/> None	Select all Or <input type="checkbox"/> None	



# Transaction Control Language (TCL)

- Lab 1: Insert a few rows into the courses table and use COMMIT to save the changes.

START TRANSACTION;

```
INSERT INTO school_db.courses (course_id, course_name, course_duration) VALUES (1004, 'digi marketing','9 months'), (1005,'frontend','4 months');
```

COMMIT;

Show query box

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

START TRANSACTION;

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 2 rows inserted. (Query took 0.0035 seconds.)

INSERT INTO school\_db.courses (course\_id, course\_name, course\_duration) VALUES (1004, 'digi marketing','9 months'), (1005,'frontend','4 months');

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0018 seconds.)

COMMIT;
















[ Edit inline ] [ Edit ] [ Create PHP code ]

```
SELECT * FROM `courses`
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows:  Filter rows:  Sort by key:

Extra options

				course_id	course_name	course_duration
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>
				1001	soft dev	8 months
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>
				1002	web dev	5 months
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>
				1003	flutter	9 months
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>
				1004	digi marketing	9 months
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>
				1005	frontend	4 months

 ☐ Check all With selected:  [Edit](#)  [Copy](#)  [Delete](#)  [Export](#)

- Lab 2: Insert additional rows, then use ROLLBACK to undo the last insert operation.

START TRANSACTION;

INSERT INTO school\_db.courses (course\_id, course\_name, course\_duration) VALUES (1006, 'digi marketing','6 months'), (1007,'backend','12 months');

ROLLBACK;

---

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

START TRANSACTION;

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

✓ 2 rows inserted. (Query took 0.0023 seconds.)

INSERT INTO school\_db.courses (course\_id, course\_name, course\_duration) VALUES (1006, 'digi marketing','6 months'), (1007,'backend','12 months');

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0014 seconds.)

ROLLBACK;

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

---

- Lab 3: Create a SAVEPOINT before updating the courses table, and use it to roll back specific changes.

START TRANSACTION;

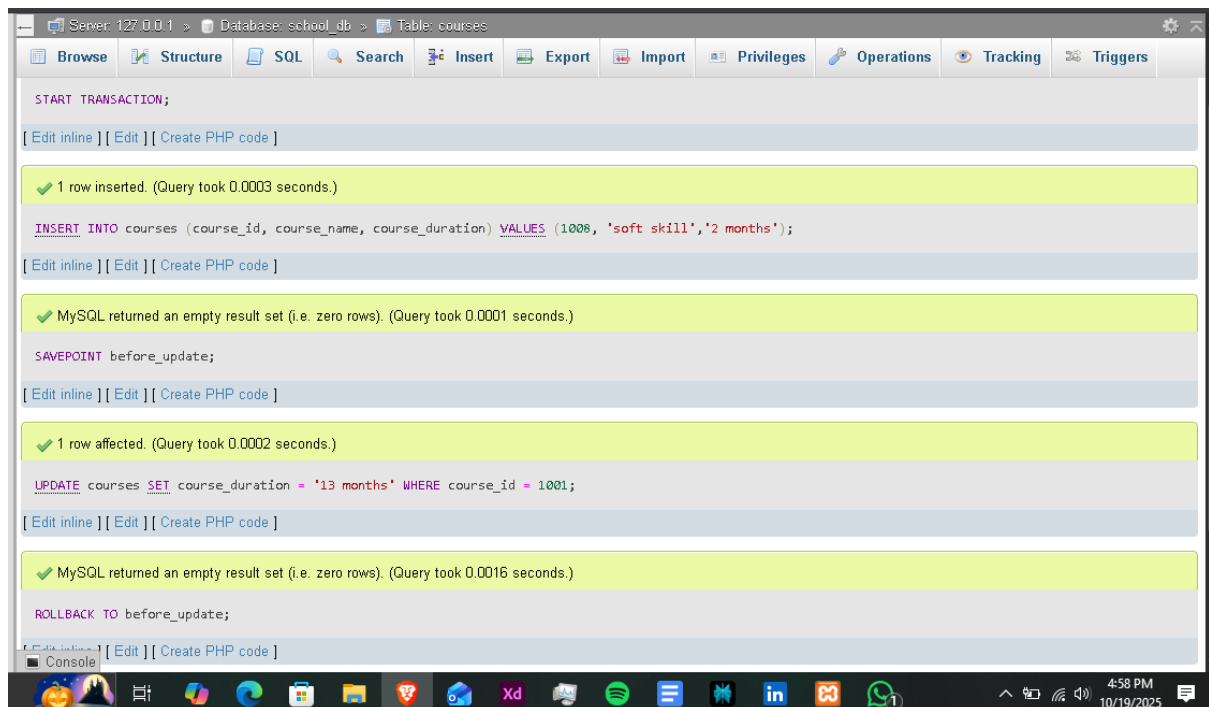
INSERT INTO courses (course\_id, course\_name, course\_duration) VALUES (1008, 'soft skill', '2 months');

SAVEPOINT before\_update;

UPDATE courses SET course\_duration = '13 months'  
WHERE course\_id = 1001;

ROLLBACK TO before\_update;

COMMIT;



SELECT \* FROM `courses`

☐ Profiling

[\[ Edit inline \]](#)

[\[ Edit \]](#)

[\[ Explain SQL \]](#)

[\[ Create PHP code \]](#)

[\[ Refresh \]](#)

☐ Show all

Number of rows: 25

Filter rows:

Sort by key: None

Extra options

←

T

→

			course_id	course_name	course_duration	
<input type="checkbox"/>	Edit	Copy	Delete	1001	soft dev	8 months
<input type="checkbox"/>	Edit	Copy	Delete	1002	web dev	5 months
<input type="checkbox"/>	Edit	Copy	Delete	1003	flutter	9 months
<input type="checkbox"/>	Edit	Copy	Delete	1004	digi marketing	9 months
<input type="checkbox"/>	Edit	Copy	Delete	1005	frontend	4 months
<input type="checkbox"/>	Edit	Copy	Delete	1008	soft skill	2 months

↑

☐ Check all

With selected:

Edit

Copy

Delete

Export

# SQL Joins

- Lab 1: Create two tables: departments and employees. Perform an INNER JOIN to display employees along with their respective departments.

Extra options

emp_id	emp_name	dept_name
101	Rahul	HR
102	Anita	IT
103	Vijay	Finance

☐ Show all

Number of rows: 25 ▼

Filter rows:

Sort by key: None ▼

Query results operations

```
ALTER TABLE employee ADD CONSTRAINT fk_dept
FOREIGN KEY (dept_id) REFERENCES
departments(dept_id);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	<b>emp_id</b>	int(11)		No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	<b>emp_name</b>	varchar(100)	utf8mb4_general_ci	No	None			Change  Drop  More
<input type="checkbox"/>	3	<b>dept_id</b>	int(11)		No	None			Change  Drop  More
<input type="checkbox"/>	4	<b>salary</b>	int(11)		No	None			Change  Drop  More

☐ Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext  
 Add to control columns    Remove from control columns

```
SELECT
    employee.emp_id,employee.emp_name,departments.dept
    _name FROM employee
INNER JOIN departments
ON employee.dept_id = departments.dept_id;
```

- Lab 2: Use a LEFT JOIN to show all departments, even those without employees.

SELECT

departments.dept\_id,departments.dept\_name,employee.emp\_id,employee.emp\_name FROM departments

LEFT JOIN employee

ON departments.dept\_id = employee.dept\_id;



# SQL Group By

- Lab 1: Group employees by department and count the number of employees in each department using GROUP BY.

```
SELECT dept_id,COUNT(emp_id) AS total_emp FROM  
employee GROUP BY dept_id
```

☐ Show all | Number of rows: 25 ▼ | Filter rows:

Extra options

dept_id	total_employees
1	1
2	1
3	1

☐ Show all | Number of rows: 25 ▼ | Filter rows:










Query results operations






Print Copy to clipboard Export Display chart Create view

- Lab 2: Use the AVG aggregate function to find the average salary of employees in each department.

```
SELECT dept_id,AVG(salary) AS avg_salary FROM  
employee GROUP by dept_id
```

Extra options

				dept_id	average_salary
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	30000.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	40000.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	35000.000000

 ☐ Check all    With selected:  Edit     Copy     Delete     Export

☐ Show all    Number of rows: 25 ▼    Filter rows:

# SQL Stored Procedure

- Lab 1: Write a stored procedure to retrieve all employees from the employees table based on department.

DELIMITER \$\$

```
CREATE PROCEDURE GetEmps(IN dept INT)
BEGIN
    SELECT emp_id, emp_name, salary, dept_id
    FROM employee
    WHERE dept_id = dept;
END $$
```

DELIMITER ;

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0059 seconds.)

```
CREATE PROCEDURE GetEmps(IN dept INT) BEGIN SELECT emp_id, emp_name, salary, dept_id FROM employee WHERE dept_id = dept; END;
```

[\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Create PHP code \]](#)







- Lab 2: Write a stored procedure that accepts `course_id` as input and returns the course details.

```
DELIMITER //
```

```
CREATE PROCEDURE GetCourseDetails(IN course INT)
BEGIN
    SELECT *
    FROM courses
    WHERE course_id = course;
END //
```

```
DELIMITER ;
```










Routines 

<input type="checkbox"/> Check all	 Export	 Drop
Name	Type	Returns
<input type="checkbox"/> GetCourseDetails	PROCEDURE	 Edit  Execute  Export  Drop

☐ 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

			emp_id	emp_name	dept_id	salary
<input type="checkbox"/>				101	Rahul	1 30000.00
<input type="checkbox"/>				102	Anita	2 40000.00
<input type="checkbox"/>				103	Vijay	3 35000.00

 ☐ Check all | With selected:  Edit  Copy  Delete  Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None



Show all

Number of rows:

25



Filter rows:

Search this table

Extra options

emp_id	emp_name	dept_id	salary
101	Rahul	1	30000.00

# SQL View

- Lab 1: Create a view to show all employees along with their department names.

```
CREATE VIEW emp_dept_view AS
SELECT employee.emp_id, employee.emp_name,
       departments.dept_name
FROM employee
INNER JOIN departments
ON employee.dept_id = departments.dept_id;
```

Extra options

				emp_id	emp_name	dept_name
<input type="checkbox"/>	Edit	Copy	Delete	101	Rahul	HR
<input type="checkbox"/>	Edit	Copy	Delete	102	Anita	IT
<input type="checkbox"/>	Edit	Copy	Delete	103	Vijay	Finance
↑	<input type="checkbox"/> Check all	With selected:	Edit	Copy	Delete	Export





- Lab 2: Modify the view to exclude employees whose salaries are below \$50,000.

```
CREATE VIEW emp_dept_view_less_50k+ AS
SELECT employee.emp_id, employee.emp_name,
       departments.dept_name, employee.salary
FROM employee
INNER JOIN departments
ON employee.dept_id = departments.dept_id
WHERE employee.salary >= 50000;
```

<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None
Extra options						
		emp_id	emp_name	dept_id	salary	
<input type="checkbox"/>	Edit	Copy	Delete	101	Rahul	1 30000.00
<input type="checkbox"/>	Edit	Copy	Delete	102	Anita	2 40000.00
<input type="checkbox"/>	Edit	Copy	Delete	103	Vijay	3 35000.00
	<input type="checkbox"/> Check all	With selected:		Edit	Copy	Delete  Export

# SQL Triggers

- Lab 1: Create a trigger to automatically log changes to the employees table when a new employee is added.

```
CREATE TABLE employee_log  
(  
    log_id INT AUTO_INCREMENT PRIMARY KEY,  
    emp_id INT, emp_name VARCHAR(200), dept_id INT,  
    action_time DATETIME, action_description  
    VARCHAR(200)  
);
```

DELIMITER \$\$

```
CREATE TRIGGER employee_add
AFTER INSERT ON employee
FOR EACH ROW
BEGIN
    INSERT INTO employee_log (emp_id, emp_name,
        dept_id, action_time, action_description)
    VALUES (NEW.emp_id, NEW.emp_name,
        NEW.dept_id, NOW(), 'new employee added');
END $$
```

DELIMITER ;

Show query box

✓ MySQL returned an empty result set (i.e. zero rows) (Query took 0.0048 seconds.)

```
CREATE TRIGGER after_employee_insert AFTER INSERT ON employees FOR EACH ROW BEGIN INSERT INTO employee_log (emp_id, action, action_date) VALUES (NEW.emp_id, 'INSERT', NOW()); END;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]



Lab 2: Create a trigger to update the last\_modified timestamp whenever an employee record is updated.

```
DELIMITER $$
```

```
CREATE TRIGGER emp_modification  
BEFORE UPDATE ON employee  
FOR EACH ROW
```

```
BEGIN  
  INSERT INTO employee_log (emp_id, emp_name,  
    dept_id, salary, action_time, action_description)  
  VALUES (NEW.emp_id, NEW.emp_name,  
    NEW.dept_id, NEW.salary, NOW(), 'UPDATED');
```

```
END$$
```

```
DELIMITER ;
```

For checking

```
update employee set salary=10000 where emp_id=2
```

phpMyAdmin

Recent Favorites

- New
- company\_db
  - Procedures
  - Tables
    - New
    - departments
    - employees
    - employee\_log
  - Views
    - New
    - employee\_department\_vie
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test
- tops

Server: 127.0.0.1 » Database: company\_db

Structure SQL Search Query Export Import Operations Privileges

## Triggers

☐ Check all [Export](#) [Drop](#)

Name	Table	Time	Event	
<input type="checkbox"/> after_employee_insert	employees	AFTER	INSERT	<a href="#">Edit</a> <a href="#">Export</a> <a href="#">Drop</a>



# Introduction to PL/SQL

- Lab 1: Write a PL/SQL block to print the total number of employees from the employees table.

```
DELIMITER $$
```

```
CREATE PROCEDURE total_emp()
```

```
BEGIN
```

```
    DECLARE total_employees INT;
```

```
    SELECT COUNT(emp_id) AS total_employees FROM  
employee;
```

```
END$$
```

```
DELIMITER ;
```

Routines

☐ Check all

Name	Type	Returns
<input type="checkbox"/> GetEmployeesByDepartment	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/> GetTotalEmployees	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>





- Lab 2: Create a PL/SQL block that calculates the total sales from an orders table.

## Customer table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	cust_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	cust_name	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
3	buying_time	datetime			No	None			Change Drop More
4	total_bill	int(11)			No	None			Change Drop More

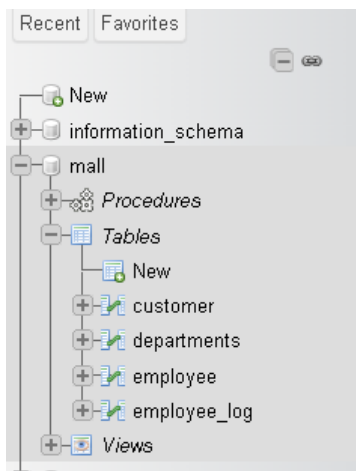
☐ Check all    With selected: ☐ Browse ☐ Change ☐ Drop ☐ Primary ☐ Unique ☐ Index ☐ Spatial ☐ Fulltext  
☐ Add to central columns ☐ Remove from central columns

DELIMITER \$\$

```
CREATE PROCEDURE total_sale()
BEGIN
```

```
    SELECT SUM(total_bill) AS total_sale FROM customer;
END$$
```

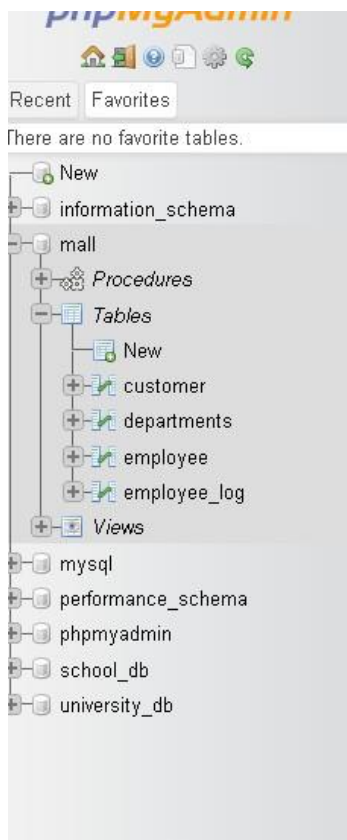
DELIMITER ;



## Routines

☐ Check all

	Name	Type	Returns
<input type="checkbox"/>	GetEmps	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/>	total_emp	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/>	total_sale	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>



Structure SQL Search Query Export Import

✓ Your SQL query has been executed successfully.  
1 row affected by the last statement inside the procedure.

```
CALL `total_sale`();
```

Execution results of routine `total\_sale`

total_sale
9500

## Routines

☐ Check all

	Name	Type	Returns
<input type="checkbox"/>	GetEmps	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/>	total_emp	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/>	total_sale	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>

# PL/SQL Control Structures

- Lab 1: Write a PL/SQL block using an IF-THEN condition to check the department of an employee.

```
DELIMITER $$
```

```
CREATE PROCEDURE CheckDeptByEmpId(IN emp INT)  
BEGIN
```

```
    DECLARE temp_name VARCHAR(100);
```

```
    SELECT d.dept_name INTO temp_name  
    FROM employee e  
    JOIN departments d ON e.dept_id = d.dept_id  
    WHERE e.emp_id = emp;
```

```
    IF temp_name IS NOT NULL THEN
```

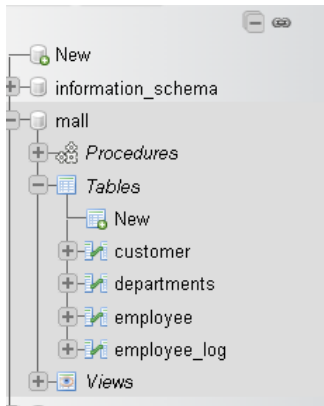
```
        SELECT CONCAT('Employee is in ', temp_name) AS  
Message;
```

```
    ELSE
```

```
        SELECT 'Employee id is invalid ' AS Message;  
    END IF;
```

```
END $$
```

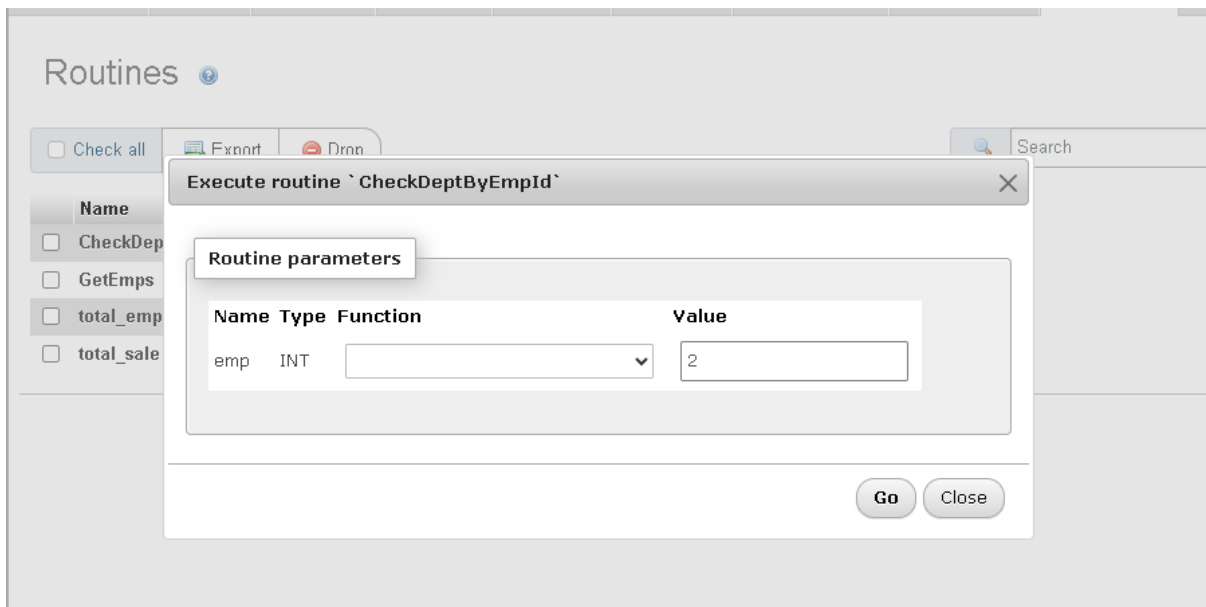
```
DELIMITER ;
```



## Routines

☐ Check all

Name	Type	Returns
<input type="checkbox"/> CheckDeptByEmpId	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/> GetEmps	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/> total_emp	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>
<input type="checkbox"/> total_sale	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>



Recent

Favorites

New

information\_schema

mall
 

Procedures

Tables
 

New

customer

departments

employee

employee\_log

Views

mysql

performance\_schema

phpmyadmin

school\_db

university\_db

Structure

SQL

Search

Query

Export

Import

Operations

✓

Your SQL query has been executed successfully.

1 row affected by the last statement inside the procedure.

```
SET @p0='2'; CALL `CheckDeptByEmpId` (@p0);
```

Execution results of routine `CheckDeptByEmpId`

Message

Employee is in anya

Routines

Check all

Export

Drop

	Name	Type	Returns				
<input type="checkbox"/>	CheckDeptByEmpId	PROCEDURE		Edit	Execute	Export	Drop
<input type="checkbox"/>	GetEmps	PROCEDURE		Edit	Execute	Export	Drop
<input type="checkbox"/>	total_emp	PROCEDURE		Edit	Execute	Export	Drop
<input type="checkbox"/>	total_sale	PROCEDURE		Edit	Execute	Export	Drop

Recent

Favorites

New

information\_schema

mall
 

Procedures

Tables
 

New

customer

departments

employee

employee\_log

Views

mysql

performance\_schema

phpmyadmin

school\_db

university\_db

Structure

SQL

Search

Query

Export

Import

Operations

✓

Your SQL query has been executed successfully.

1 row affected by the last statement inside the procedure.

```
SET @p0='5'; CALL `CheckDeptByEmpId` (@p0);
```

Execution results of routine `CheckDeptByEmpId`

Message

Employee is in bata

Routines

Check all

Export

Drop

	Name	Type	Returns				
<input type="checkbox"/>	CheckDeptByEmpId	PROCEDURE		Edit	Execute	Export	Drop
<input type="checkbox"/>	GetEmps	PROCEDURE		Edit	Execute	Export	Drop
<input type="checkbox"/>	total_emp	PROCEDURE		Edit	Execute	Export	Drop
<input type="checkbox"/>	total_sale	PROCEDURE		Edit	Execute	Export	Drop

- Lab 2: Use a FOR LOOP to iterate through employee records and display their names.

```
DELIMITER $$
```

```
CREATE PROCEDURE DisplayEmployeeNames()
```

```
BEGIN
```

```
    DECLARE total INT DEFAULT 0;
```

```
    DECLARE counter INT DEFAULT 1;
```

```
    DECLARE empName VARCHAR(100);
```

```
    SELECT COUNT(*) INTO total FROM employee;
```

```
    WHILE counter <= total DO
```

```
        SELECT emp_name INTO empName
```

```
        FROM employee
```

```
        WHERE emp_id = counter;
```

```
        SELECT CONCAT(counter, '-> ', empName) AS  
'Employee name';
```






```
        SET counter = counter + 1;
```

```
    END WHILE;
```

```
END $$
```

```
DELIMITER ;
```

phpMyAdmin

Recent Favorites

New

information\_schema

mall





















- Procedures
- Tables
  - New
  - customer
  - departments
  - employee
  - employee\_log
- Views

Server: 127.0.0.1 > Database: mall

StructureSQLSearchQueryExportImportOperations

Routines

☐ Check all

	Name	Type	Returns
<input type="checkbox"/>	CheckDeptByEmpId	PROCEDURE	 Edit  Execute  Export  Drop
<input type="checkbox"/>	DisplayEmployeeNames	PROCEDURE	 Edit  Execute  Export  Drop
<input type="checkbox"/>	GetEmps	PROCEDURE	 Edit  Execute  Export  Drop
<input type="checkbox"/>	total_emp	PROCEDURE	 Edit  Execute  Export  Drop
<input type="checkbox"/>	total_sale	PROCEDURE	 Edit  Execute  Export  Drop



phpMyAdmin

Recent

Favorites

New

information\_schema

mall

- Procedures
- Tables
  - New
  - customer
  - departments
  - employee
  - employee\_log
- Views

- mysql
- performance\_schema
- phpmyadmin
- school\_db
- university\_db

Server: 127.0.0.1 > Database: mall

Structure

SQL

Search

Query

Export

Import

Operations

Privileges

Employee name

6-> megumin

Employee name

7-> maomao

Employee name

8-> lloyd

Routines

☐ Check all

Export

Drop

	Name	Type	Returns
<input type="checkbox"/>	CheckDeptByEmpld	PROCEDURE	Edit  Execute  Export  Drop
<input type="checkbox"/>	GetEmps	PROCEDURE	Edit  Execute  Export  Drop
<input type="checkbox"/>	total_emp	PROCEDURE	Edit  Execute  Export  Drop
<input type="checkbox"/>	total_sale	PROCEDURE	Edit  Execute  Export  Drop
<input type="checkbox"/>	DisplayEmployeeNames	PROCEDURE	Edit  Execute  Export  Drop

Console

# SQL Cursors

- Lab 1: Write a PL/SQL block using an explicit cursor to retrieve and display employee details.

```
DELIMITER $$
```

```
CREATE PROCEDURE ShowEmployeeDetails()
```

```
BEGIN
```

```
    DECLARE h_id INT;
```

```
    DECLARE h_name VARCHAR(100);
```

```
    DECLARE h_dept INT;
```

```
    DECLARE h_salary INT;
```

```
    DECLARE done INT DEFAULT 0;
```

```
    DECLARE emp_cursor CURSOR FOR
```

```
        SELECT emp_id, emp_name, dept_id, salary FROM  
employee;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND  
SET done = 1;
```

```
    OPEN emp_cursor;
```

```
    read_loop: LOOP
```

```
        FETCH emp_cursor INTO h_id, h_name, h_dept,  
h_salary;
```

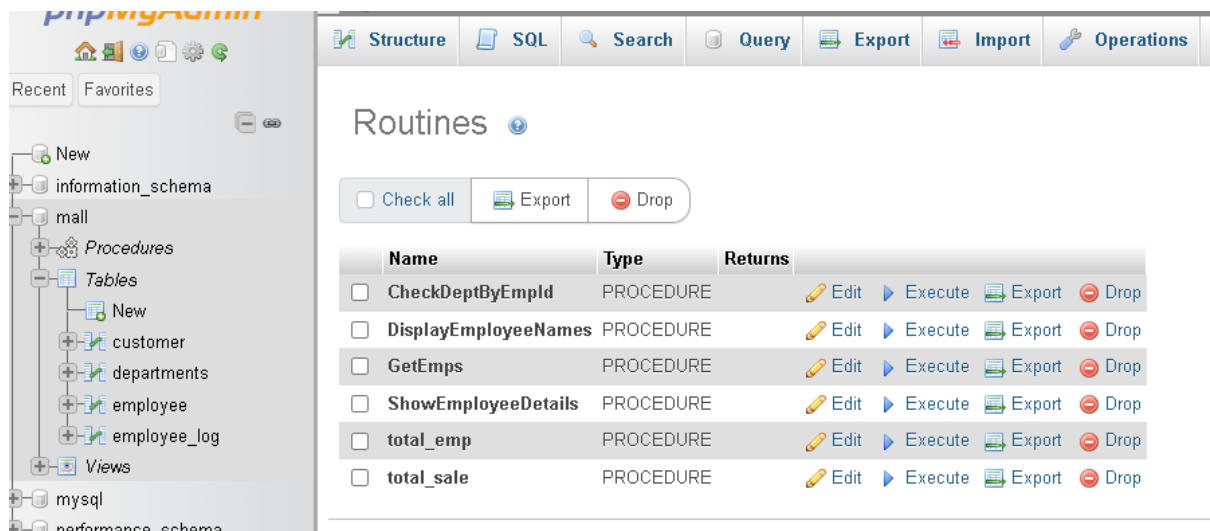
```
        IF done = 1 THEN
```

```
LEAVE read_loop;  
END IF;
```

```
SELECT CONCAT('Emp ID : ', h_id, ', name: ', h_name,  
, dept id: ', h_dept, ', salary: ', h_salary) AS  
Employee_details;  
END LOOP;
```

```
CLOSE emp_cursor;  
END $$
```

```
DELIMITER ;
```





phpMyAdmin

Recent

Favorites

New

information\_schema

mall

- Procedures
- Tables
  - New
  - customer
  - departments
  - employee
  - employee\_log
- Views

mysql

performance\_schema

phpmyadmin

school\_db

university\_db

Server: 127.0.0.1 > Database: mall

Structure

SQL

Search

Query

Export

Import

Operations

Employee\_details

Emp ID : 6, name: megumin, dept id: 2, salary: 10000

Employee\_details

Emp ID : 7, name: maomao, dept id: 2, salary: 90000

Employee\_details

Emp ID : 8, name: lloyd, dept id: 3, salary: 200000

Routines

☐ Check all

Export

Drop

	Name	Type	Returns	
<input type="checkbox"/>	CheckDeptByEmpId	PROCEDURE	Edit  Execute  Export  Drop	
<input type="checkbox"/>	DisplayEmployeeNames	PROCEDURE	Edit  Execute  Export  Drop	
<input type="checkbox"/>	GetEmps	PROCEDURE	Edit  Execute  Export  Drop	
<input type="checkbox"/>	ShowEmployeeDetails	PROCEDURE	Edit  Execute  Export  Drop	
<input type="checkbox"/>	total_emp	PROCEDURE	Edit  Execute  Export  Drop	
<input type="checkbox"/>	total_sale	PROCEDURE	Edit  Execute  Export  Drop	

Console

- Lab 2: Create a cursor to retrieve all courses and display them one by one.

```
DELIMITER $$
```

```
CREATE PROCEDURE ShowCourses()
```

```
BEGIN
```

```
    DECLARE c_id INT;
```

```
    DECLARE c_name VARCHAR(100);
```

```
    DECLARE c_duration VARCHAR(50);
```

```
    DECLARE done INT DEFAULT 0;
```

```
    DECLARE course_cursor CURSOR FOR
```

```
        SELECT course_id, course_name, course_duration
```

```
FROM courses;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND
```

```
SET done = 1;
```

```
    OPEN course_cursor;
```

```
read_loop: LOOP
```

```
    FETCH course_cursor INTO c_id, c_name, c_duration;
```

```
    IF done = 1 THEN
```

```
        LEAVE read_loop;
```

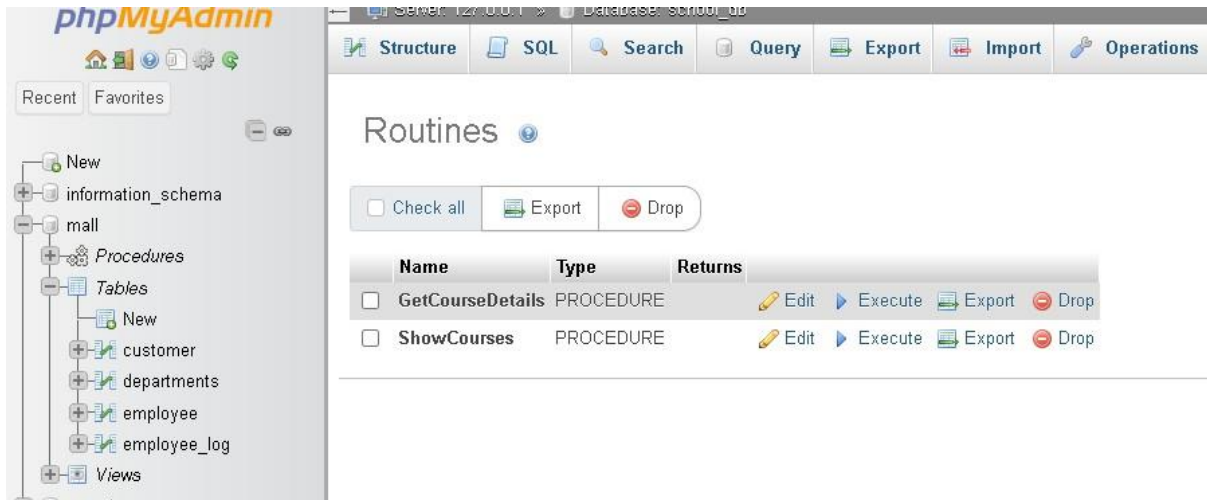
```
    END IF;
```

```
        SELECT CONCAT('course id : ', c_id, ', name: ',  
c_name, ', duration: ', c_duration) AS course_details;
```

```
    END LOOP;
```

```
CLOSE course_cursor;  
END $$
```

DELIMITER ;



it\_vie

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25  Filter rows:  Sort by key: None

Extra options

<div><div><div>←</div><div>T</div><div>→</div></div></div>						order_id	customer_id	order_amount	order_date	
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>	1	101	500.00	2025-12-01
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>	2	102	1200.50	2025-12-05
<input type="checkbox"/>		<a href="#">Edit</a>		<a href="#">Copy</a>		<a href="#">Delete</a>	3	101	750.25	2025-12-10

 ☐ Check all    With selected:  [Edit](#)     [Copy](#)     [Delete](#)     [Export](#)

☐ Show all | Number of rows: 25  Filter rows:  Sort by key: None

Query results operations



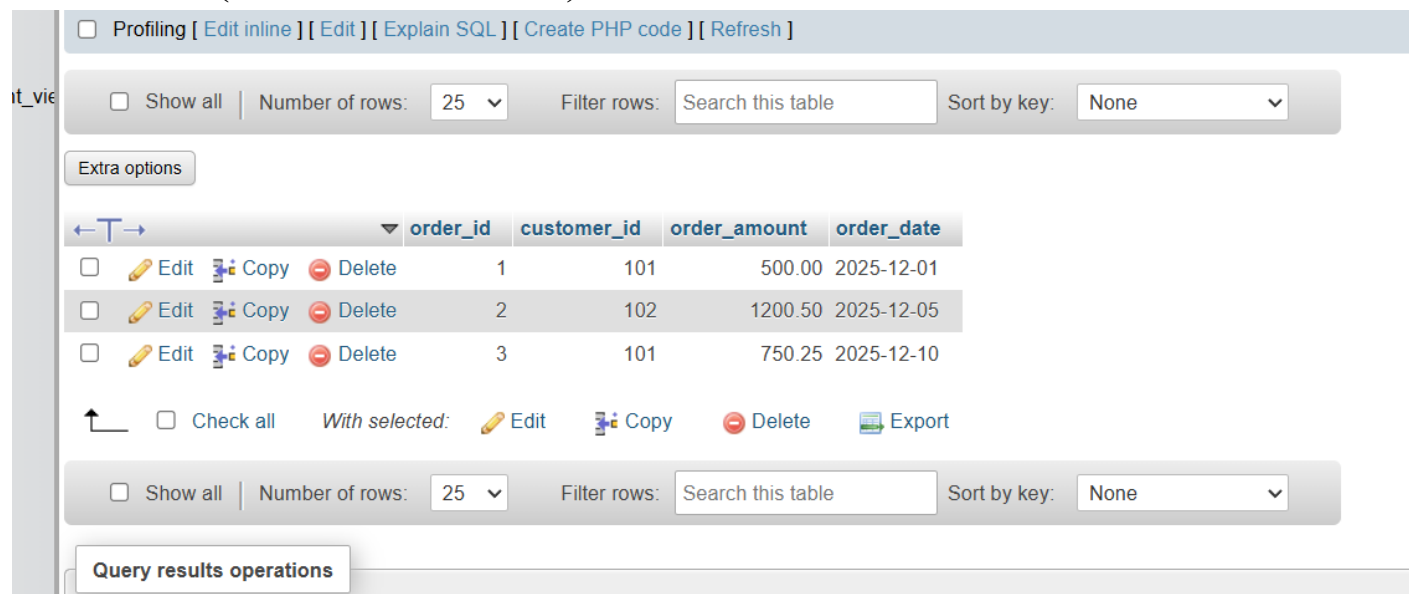
# Rollback and Commit Savepoint

- Lab 1: perform a transaction where you create a savepoint, insert records, then rollback to the savepoint.

START TRANSACTION;

```
INSERT INTO employee (emp_name, dept_id, salary)
VALUES ('vasu', 3, 70000);
SAVEPOINT after_first_insert;
```

```
INSERT INTO employee (emp_name, dept_id, salary)
VALUES ('khwab', 1, 52000);
```



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

			order_id	customer_id	order_amount	order_date
<input type="checkbox"/>	Edit	Copy	Delete	1	101	500.00 2025-12-01
<input type="checkbox"/>	Edit	Copy	Delete	2	102	1200.50 2025-12-05
<input type="checkbox"/>	Edit	Copy	Delete	3	101	750.25 2025-12-10

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

```
ROLLBACK TO after_first_insert;
```

```
COMMIT;
```

- Lab 2: Commit part of a transaction after using a savepoint and then rollback the remaining changes.

```
START TRANSACTION;
```

```
INSERT INTO courses (course_id, course_name,  
course_duration)  
VALUES (1009, 'cloud basics', '6 months');
```

```
SAVEPOINT after_first_course;
```

```
INSERT INTO courses (course_id, course_name,  
course_duration)  
VALUES (1010, 'data analytics', '7 months');
```

```
RELEASE SAVEPOINT after_first_course;  
COMMIT;
```

```
START TRANSACTION;
```

```
INSERT INTO courses (course_id, course_name,  
course_duration)  
VALUES (1011, 'ai ml', '8 months');
```

```
SAVEPOINT after_new_course;
```

```
INSERT INTO courses (course_id, course_name,  
course_duration)  
VALUES (1012, 'seo', '5 months');
```

The screenshot displays a database management tool interface. On the left is a sidebar with a tree view of the database structure, including 'company\_db', 'Procedures', 'Tables' (with sub-items like 'departments', 'employees', 'employee\_log', 'orders'), 'Views' (with 'employee\_department\_view'), and various system schemas like 'information\_schema', 'mysql', 'performance\_schema', 'phpmyadmin', 'test', and 'tops'.

The main area shows a sequence of SQL queries and their results:

- Query 1:** `INSERT INTO employees (emp_id, emp_name, dept_id, salary) VALUES (108, 'Amit', 1, 60000);` Result: 1 row inserted. (Query took 0.0007 seconds.)
- Query 2:** `INSERT INTO employees (emp_id, emp_name, dept_id, salary) VALUES (109, 'Priya', 2, 65000);` Result: 1 row inserted. (Query took 0.0039 seconds.)
- Query 3:** `SELECT * FROM `orders` WHERE 1;` Result: Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

Below the queries is a table view for the 'orders' table. It includes a 'Profiling' checkbox and a control bar with 'Show all', 'Number of rows' (set to 25), 'Filter rows' (search box), and 'Sort by key' (set to None). The table data is as follows:

	order_id	customer_id	order_amount	order_date
<input type="checkbox"/> Edit Copy Delete	1	101	500.00	2025-12-01
<input type="checkbox"/> Edit Copy Delete	2	102	1200.50	2025-12-05
<input type="checkbox"/> Edit Copy Delete	3	101	750.25	2025-12-10

At the bottom, there are 'Check all', 'With selected:' options, and buttons for 'Edit', 'Copy', 'Delete', and 'Export'. A 'Console' tab is also visible at the very bottom.

ROLLBACK TO after\_new\_course;

COMMIT;