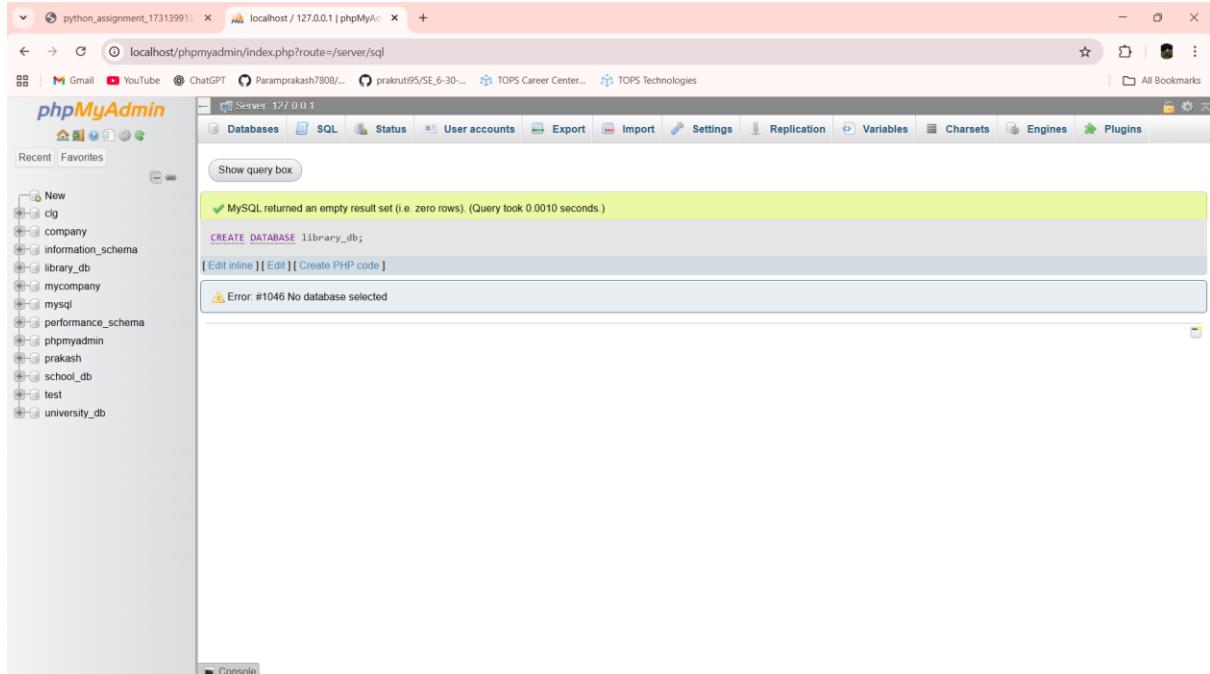


Name: Paramprakash Makwana

Module 4 – Introduction to DBMS(Extra Lab)

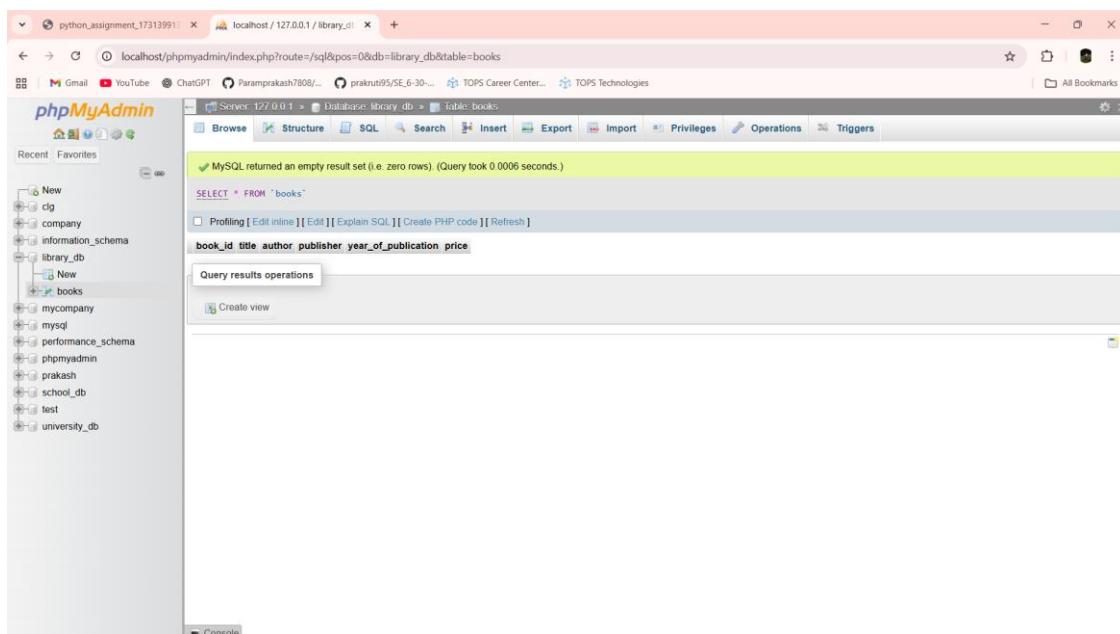
Lab 1.3: Create a database called library_db and a table books with columns: book_id, title, author, publisher, year_of_publication, and price. Insert five records into the table.

Ans:- CREATE DATABASE library_db



The screenshot shows the phpMyAdmin interface on a Windows operating system. The browser tab is titled "python_assignment_17313991" and the address bar shows "localhost / 127.0.0.1 | phpMyAdmin". The left sidebar lists various databases: New, cg, company, information_schema, library_db (selected), mycompany, mysql, performance_schema, phpmyadmin, prakash, school_db, test, university_db. The main query window contains the SQL command: "CREATE DATABASE library_db;". A green status message at the top says "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.)". Below the query, there are "Edit inline", "Edit", and "Create PHP code" buttons. A yellow warning message at the bottom says "Error: #1046 No database selected".

CREATE TABLE books (book_id INT PRIMARY KEY,title VARCHAR(100),author VARCHAR(100),publisher VARCHAR(100),year_of_publication INT,price int);



The screenshot shows the phpMyAdmin interface on a Windows operating system. The browser tab is titled "python_assignment_17313991" and the address bar shows "localhost/phpmyadmin/index.php?route=/library_db/structure&table=books". The left sidebar lists databases: New, cg, company, information_schema, library_db (selected), mycompany, mysql, performance_schema, phpmyadmin, prakash, school_db, test, university_db. The main query window shows the SQL command: "SELECT * FROM `books`;" and a status message "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)". Below the query, there are "Profiling", "Edit inline", "Edit", "Explain SQL", "Create PHP code", and "Refresh" buttons. A "Query results operations" section is visible at the bottom.

INSERT INTO books VALUES

- (1, 'The Alchemist', 'Paulo Coelho', 'HarperCollins', 1988, 399.00),
- (2, 'Clean Code', 'Robert C. Martin', 'Prentice Hall', 2008, 799.00),
- (3, 'Atomic Habits', 'James Clear', 'Penguin', 2018, 499.00),
- (4, 'Think Like a Monk', 'Jay Shetty', 'Simon & Schuster', 2020, 450.00),
- (5, 'Rich Dad Poor Dad', 'Robert Kiyosaki', 'Warner Books', 1997, 350.00);

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'books' table is selected, displaying the following data:

book_id	title	author	publisher	year_of_publication	price
1	The Alchemist	Paulo Coelho	HarperCollins	1988	399
2	Clean Code	Robert C. Martin	Prentice Hall	2008	799
3	Atomic Habits	James Clear	Penguin	2018	499
4	Think Like a Monk	Jay Shetty	Simon & Schuster	2020	450
5	Rich Dad Poor Dad	Robert Kiyosaki	Warner Books	1997	350

Lab 1.4: Create a table members in library_db with columns: member_id, member_name, date_of_membership, and email. Insert five records into this table.

Ans:- CREATE TABLE members (member_id INT PRIMARY KEY, member_name VARCHAR(100), date_of_membership DATE, email VARCHAR(100))

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, including the 'library_db' database which contains the 'books' and 'members' tables. The 'members' table has two rows: 'Member ID' and 'Member Name'. The 'Member ID' row contains the value '1'. The 'Member Name' row contains the value 'Prakash'. The top navigation bar shows the URL as 'localhost/phpmyadmin/index.php?route=/table/sql&db=library_db&table=books'. The main query window displays the SQL command: 'CREATE TABLE members (member_id INT PRIMARY KEY, member_name VARCHAR(100), date_of_membership DATE, email VARCHAR(100));'. A message below the query states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0157 seconds.)'.

INSERT INTO members VALUES

```
(1, 'Ajay Makwana', '2023-01-15', 'ajay@gmail.com'),  
(2, 'Sahil Makwana', '2023-03-10', 'sahil@gmail.com'),  
(3, 'Mayur Makwana', '2023-05-22', 'mayur@gmail.com'),  
(4, 'Darshit Makwana', '2023-07-08', 'darshit@gmail.com'),  
(5, 'Prakash Makwana', '2023-09-18', 'prakash@gmail.com');
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar lists various databases and tables. The 'members' table is selected under the 'library_db' section. The main area displays the data from the 'members' table:

member_id	member_name	date_of_membership	email
1	Ajay Makwana	2023-01-15	ajay@gmail.com
2	Sahil Makwana	2023-03-10	sahil@gmail.com
3	Mayur Makwana	2023-05-22	mayur@gmail.com
4	Darshit Makwana	2023-07-08	darshit@gmail.com
5	Prakash Makwana	2023-09-18	prakash@gmail.com

Lab 2.3: Retrieve all members who joined the library before 2022. Use appropriate SQL syntax with WHERE and ORDER BY.

Ans:- `SELECT * FROM members WHERE date_of_membership < '2022-01-01' ORDER BY date_of_membership`

The screenshot shows the phpMyAdmin interface for a database named 'library_db'. The 'members' table is selected. A query has been run: `SELECT * FROM members WHERE date_of_membership < '2022-01-01' ORDER BY date_of_membership;`. The results show two rows:

member_id	member_name	date_of_membership	email
1	Ajay Makwana	2018-01-15	ajay@gmail.com
2	Sahil Makwana	2020-03-10	sahil@gmail.com

Lab 2.4: Write SQL queries to display the titles of books published by a specific author. Sort the results by year_of_publication in descending order.

Ans:- `SELECT title FROM books WHERE author = 'Robert Kiyosaki' ORDER BY year_of_publication DESC`

The screenshot shows the phpMyAdmin interface for a database named 'library_db'. The 'books' table is selected. A query has been run: `SELECT title FROM books WHERE author = 'Robert Kiyosaki' ORDER BY year_of_publication DESC;`. The results show one row:

title
Rich Dad Poor Dad

Lab 3.3: Add a CHECK constraint to ensure that the price of books in the books table is greater than 0.

Ans:- ALTER TABLE books ADD CONSTRAINT chk_price CHECK (price > 0)

book_id	title	author	publisher	year_of_publication	price
1	The Alchemist	Paulo Coelho	HarperCollins	1988	399
2	Clean Code	Robert C. Martin	Prentice Hall	2008	799
3	Atomic Habits	James Clear	Penguin	2018	499
4	Think Like a Monk	Jay Shetty	Simon & Schuster	2020	450
5	Rich Dad Poor Dad	Robert Kiyosaki	Warner Books	1997	350

Lab 3.4: Modify the members table to add a UNIQUE constraint on the email column, ensuring that each member has a unique email address.

Ans:- ALTER TABLE members ADD CONSTRAINT uq_email UNIQUE (email)

member_id	member_name	date_of_membership	email
1	Ajay Makwana	2018-01-15	ajay@gmail.com
2	Sahil Makwana	2020-03-10	sahil@gmail.com
3	Mayur Makwana	2022-05-22	mayur@gmail.com
4	Darshit Makwana	2023-07-08	darshita@gmail.com
5	Prakash Makwana	2025-09-18	prakash@gmail.com

Lab 4.3: Create a table authors with the following columns: author_id, first_name, last_name, and country. Set author_id as the primary key.

Ans:- CREATE TABLE authors (author_id INT PRIMARY KEY,first_name VARCHAR(50),last_name VARCHAR(50),country VARCHAR(50))

The screenshot shows the phpMyAdmin interface for a database named 'library_db'. On the left, the database structure is visible with tables like 'authors', 'books', and 'members'. In the main area, a query has been run: 'SELECT * FROM `authors`'. The results pane shows a message: 'MySQL returned an empty result set (i.e. zero rows) (Query took 0.0004 seconds.)'. Below this, the table structure is defined with columns: 'author_id', 'first_name', 'last_name', and 'country'. The 'author_id' column is highlighted in blue, indicating it is the primary key.

Lab 4.4: Create a table publishers with columns: publisher_id, publisher_name, contact_number, and address. Set publisher_id as the primary key and contact_number as unique.

Ans:- CREATE TABLE publishers (publisher_id INT PRIMARY KEY,publisher_name VARCHAR(100),contact_number VARCHAR(15) UNIQUE,address VARCHAR(150))

The screenshot shows the phpMyAdmin interface for the same 'library_db' database. A query 'SELECT * FROM `publishers`' has been run, resulting in an empty set of rows. The table structure is defined with columns: 'publisher_id', 'publisher_name', 'contact_number', and 'address'. The 'publisher_id' column is highlighted in blue as the primary key, and the 'contact_number' column is highlighted in red as a unique constraint.

Lab 5.3: Add a new column genre to the books table. Update the genre for all existing records.

Ans:- ALTER TABLE books ADD genre VARCHAR(50)

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'books' table is selected. The table structure includes columns: book_id, title, author, publisher, year_of_publication, price, and genre. The 'genre' column is currently defined as NULL. Five rows of data are listed:

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	399	NULL
2	Clean Code	Robert C. Martin	Prentice Hall	2008	799	NULL
3	Atomic Habits	James Clear	Penguin	2018	499	NULL
4	Think Like a Monk	Jay Shetty	Simon & Schuster	2020	450	NULL
5	Rich Dad Poor Dad	Robert Kiyosaki	Warner Books	1997	350	NULL

UPDATE books SET genre = 'Fiction'

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'books' table is selected. The 'genre' column now contains the value 'Fiction'. The five rows from the previous screenshot now reflect this update:

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	399	Fiction
2	Clean Code	Robert C. Martin	Prentice Hall	2008	799	Fiction
3	Atomic Habits	James Clear	Penguin	2018	499	Fiction
4	Think Like a Monk	Jay Shetty	Simon & Schuster	2020	450	Fiction
5	Rich Dad Poor Dad	Robert Kiyosaki	Warner Books	1997	350	Fiction

Lab 5.4: Modify the members table to increase the length of the email column to 100 characters.

Ans:- ALTER TABLE members MODIFY email VARCHAR(100)

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'members' table is selected. The table structure includes columns: member_id, member_name, date_of_membership, and email. There are 5 rows of data displayed:

	member_id	member_name	date_of_membership	email
<input type="checkbox"/>	1	Ajay Makwana	2018-01-15	ajay@gmail.com
<input type="checkbox"/>	2	Sahil Makwana	2020-03-10	sahil@gmail.com
<input type="checkbox"/>	3	Mayur Makwana	2022-05-22	mayur@gmail.com
<input type="checkbox"/>	4	Darshit Makwana	2023-07-08	darshit@gmail.com
<input type="checkbox"/>	5	Prakash Makwana	2025-09-18	prakash@gmail.com

Lab 6.3: Drop the publishers table from the database after verifying its structure.

Ans:-

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'Structure' tab is selected for the 'publishers' table. The table has four columns: 'publisher_id' (int(11), primary key, auto-increment), 'publisher_name' (varchar(100)), 'contact_number' (varchar(15)), and 'address' (varchar(150)). There are two indexes: 'PRIMARY' (on publisher_id) and 'contact_number'. A confirmation dialog box is open, asking 'Do you really want to execute "DROP TABLE publishers"?'. The 'OK' button is highlighted.

DROP TABLE publishers

The screenshot shows the phpMyAdmin interface with the SQL query 'DROP TABLE publishers' entered in the query editor. A confirmation dialog box is displayed over the interface, asking 'localhost says Do you really want to execute "DROP TABLE publishers"?'. The 'OK' button is highlighted.

Lab 6.4: Create a backup of the members table and then drop the original members table.

Ans:- CREATE TABLE members_backup AS SELECT * FROM members

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'members_backup' table is selected. The table structure is as follows:

member_id	member_name	date_of_membership	email
1	Ajay Makwana	2018-01-15	ajay@gmail.com
2	Sahil Makwana	2020-03-10	sahil@gmail.com
3	Mayur Makwana	2022-05-22	mayur@gmail.com
4	Darshit Makwana	2023-07-08	darshit@gmail.com
5	Prakash Makwana	2025-09-18	prakash@gmail.com

DROP TABLE members

The screenshot shows the phpMyAdmin interface for the 'library_db' database. A confirmation dialog box is displayed, asking if the user really wants to execute the 'DROP TABLE members' query. The dialog has 'OK' and 'Cancel' buttons. The SQL query editor below contains the command: 'DROP TABLE members'. To the right, the table structure for 'members_backup' is shown:

member_id	member_name	date_of_membership	email
-----------	-------------	--------------------	-------

Lab 7.4: Insert three new authors into the authors table, then update the last name of one of the authors.

Ans:- INSERT INTO authors VALUES

(1, 'George', 'Orwell', 'UK'),

(2, 'Jane', 'Austen', 'UK'),

(3, 'Mark', 'Twain', 'USA')

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'authors' table is selected. The table data is as follows:

author_id	first_name	last_name	country
1	George	Orwell	UK
2	Jane	Austen	UK
3	Mark	Twain	USA

UPDATE authors SET last_name = 'Austin' WHERE author_id = 2

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'authors' table is selected. The table data is as follows:

author_id	first_name	last_name	country
1	George	Orwell	UK
2	Jane	Austin	UK
3	Mark	Twain	USA

Lab 7.5: Delete a book from the books table where the price is higher than \$100.

Ans:-

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'books' table is selected. The table data is as follows:

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	99	Fiction
2	Clean Code	Robert C. Martin	Prentice Hall	2008	799	Fiction
3	Atomic Habits	James Clear	Penguin	2018	49	Fiction
4	Think Like a Monk	Jay Shetty	Simon & Schuster	2020	450	Fiction
5	Rich Dad Poor Dad	Robert Kiyosaki	Warner Books	1997	350	Fiction

DELETE FROM books WHERE price > 100

The screenshot shows the phpMyAdmin interface with a confirmation dialog box titled 'localhost says'. The message reads: 'Do you really want to execute "DELETE FROM books WHERE price > 100"?'. There are 'OK' and 'Cancel' buttons. The SQL tab shows the query: 'DELETE FROM books WHERE price > 100'.

python_assignment_17313991 x localhost / 127.0.0.1 / library_db +

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=library_db&table=books

Gmail YouTube ChatGPT Paramprakash7809... prakruti95/SE_6-30... TOPS Career Center... TOPS Technologies All Bookmarks

phpMyAdmin

Recent Favorites

New clg company information_schema library_db New authors books members_backup mycompany mysql performance_schema phpmyadmin prakash school_db test university_db

Server 127.0.0.1 > Database library_db > table books

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 1 (total: 2) Query took 0.0003 seconds.

SELECT * FROM `books`

Profiling | Edit inline | Explain SQL | Create PHP code | Refresh

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

Extra options

	book_id	title	author	publisher	year_of_publication	price	genre
<input type="checkbox"/>	1	The Alchemist	Paulo Coelho	HarperCollins	1988	99	Fiction
<input type="checkbox"/>	3	Atomic Habits	James Clear	Penguin	2018	49	Fiction

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

	book_id	title	author	publisher	year_of_publication	price	genre
<input type="checkbox"/>	1	The Alchemist	Paulo Coelho	HarperCollins	1988	99	Fiction
<input type="checkbox"/>	3	Atomic Habits	James Clear	Penguin	2018	49	Fiction

Lab 8.3: Update the year_of_publication of a book with a specific book_id.

Ans:-

The screenshot shows the phpMyAdmin interface for the library_db database. The 'books' table is selected. In the SQL tab, the following query is entered:

```
UPDATE `books` SET `year_of_publication` = '2018' WHERE `books`.`book_id` = 3;
```

The results pane shows the updated data:

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	99	Fiction
3	Atomic Habits	James Clear	Penguin	2018	49	Fiction

UPDATE books SET year_of_publication = 2021 WHERE book_id = 3

The screenshot shows the phpMyAdmin interface for the library_db database. The 'books' table is selected. The results pane shows the updated data:

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	99	Fiction
3	Atomic Habits	James Clear	Penguin	2021	49	Fiction

Lab 8.4: Increase the price of all books published before 2015 by 10%.

Ans:-

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	99	Fiction
3	Atomic Habits	James Clear	Penguin	2021	49	Fiction

UPDATE books SET price = price + (price * 0.10) WHERE year_of_publication < 2015

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	109	Fiction
3	Atomic Habits	James Clear	Penguin	2021	49	Fiction

Lab 9.3: Remove all members who joined before 2020 from the members table.

Ans:-

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'members_backup' table is selected. The table has columns: member_id, member_name, date_of_membership, and email. The data is as follows:

member_id	member_name	date_of_membership	email
1	Ajay Makwana	2018-01-15	ajay@gmail.com
2	Sahil Makwana	2020-03-10	sahil@gmail.com
3	Mayur Makwana	2022-05-22	mayur@gmail.com
4	Darshil Makwana	2023-07-08	darshil@gmail.com
5	Prakash Makwana	2025-09-18	prakash@gmail.com

DELETE FROM members_backup WHERE date_of_membership < '2020-01-01'

The screenshot shows the phpMyAdmin interface with a confirmation dialog box. The dialog asks if you really want to execute the following SQL query:

```
DELETE FROM members_backup WHERE date_of_membership < '2020-01-01';
```

The dialog has two buttons: 'OK' (highlighted) and 'Cancel'. In the background, the SQL tab shows the same query entered. The results pane on the right displays the structure of the 'members_backup' table:

member_id	member_name	date_of_membership	email
-----------	-------------	--------------------	-------

Showing rows 0 - 3 (total, Query took 0.0002 seconds.)

member_id	member_name	date_of_membership	email
2	Sahil Makwana	2020-03-10	sahil@gmail.com
3	Mayur Makwana	2022-05-22	mayur@gmail.com
4	Darshit Makwana	2023-07-08	darshit@gmail.com
5	Prakash Makwana	2025-09-18	prakash@gmail.com

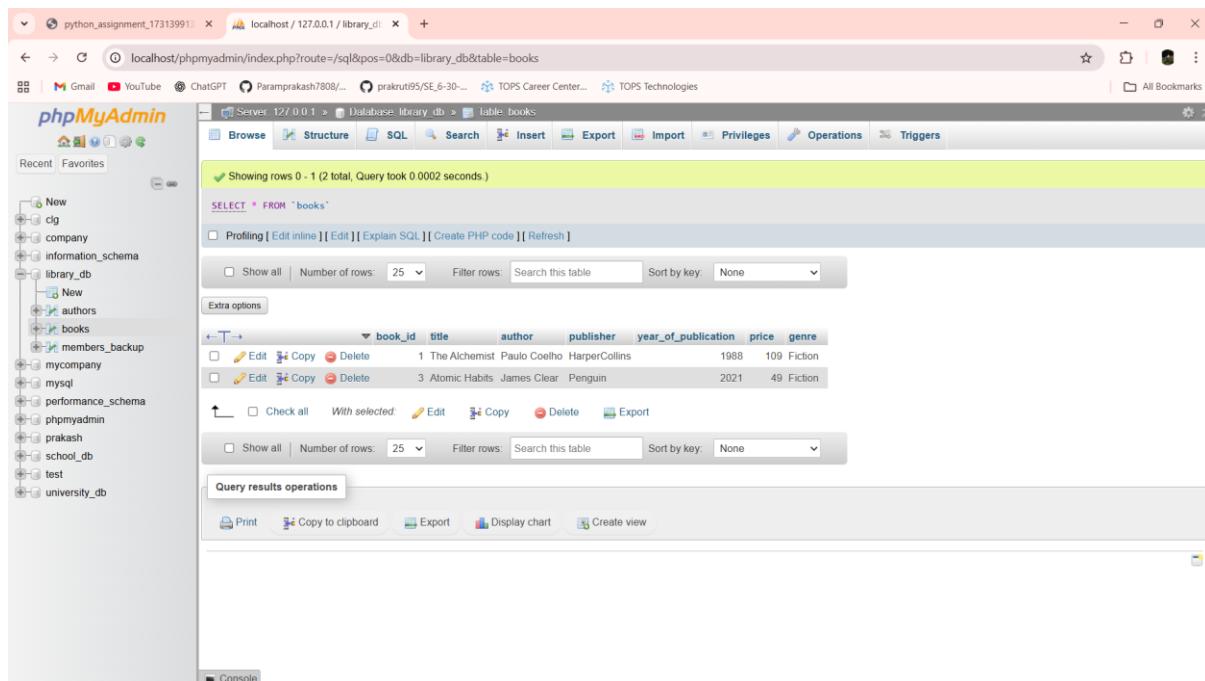
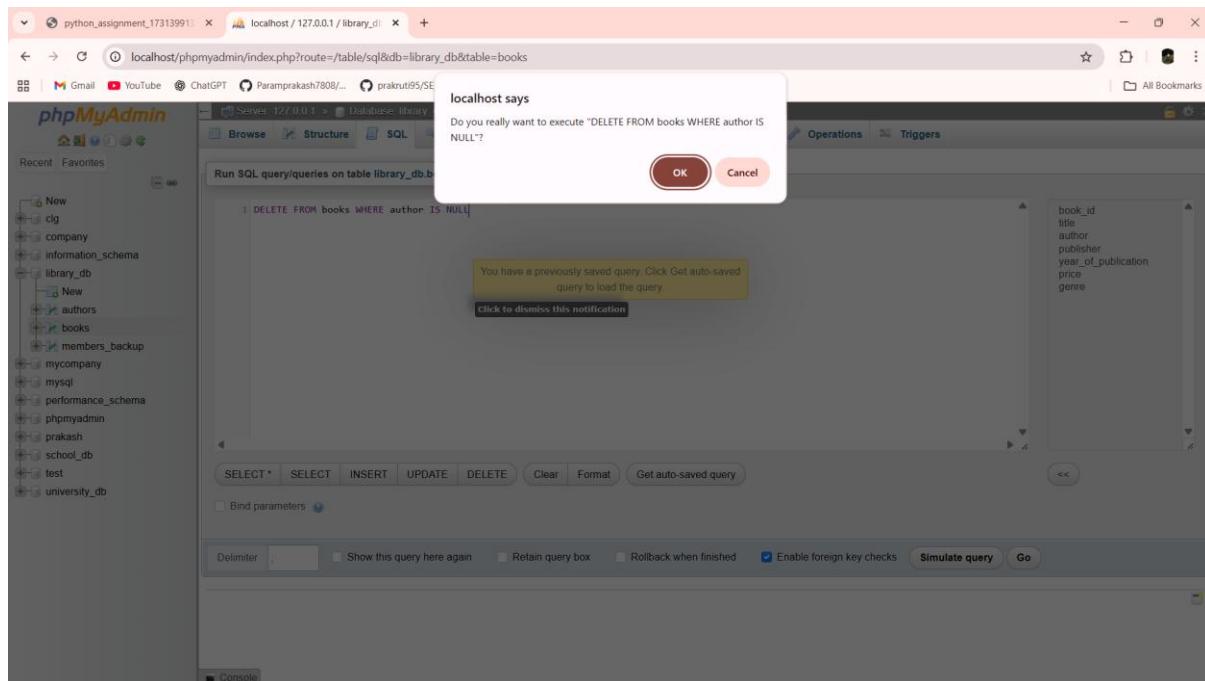
Lab 9.4: Delete all books that have a NULL value in the author column.

Ans:-

Showing rows 0 - 3 (total, Query took 0.0002 seconds.)

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	109	Fiction
2	Java	NULL	Ajay Makwana	2022	199	Fiction
3	Atomic Habits	James Clear	Penguin	2021	49	Fiction

DELETE FROM books WHERE author IS NULL



Lab 10.4: Write a query to retrieve all books with price between \$50 and \$100.

Ans:- SELECT * FROM books WHERE price BETWEEN 50 AND 100

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar lists various databases and their tables. The main area displays the results of a SQL query: 'SELECT * FROM books WHERE price BETWEEN 50 AND 100;'. The results table shows one row of data: book_id 3, title 'Atomic Habits', author 'James Clear', publisher 'Penguin', year_of_publication 2021, price 75, and genre 'Fiction'.

book_id	title	author	publisher	year_of_publication	price	genre
3	Atomic Habits	James Clear	Penguin	2021	75	Fiction

Lab 10.5: Retrieve the list of books sorted by author in ascending order and limit the results to the top 3 entries.

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'books' table is selected. The table structure is displayed with columns: book_id, title, author, publisher, year_of_publication, price, and genre. Two rows are present in the data grid:

book_id	title	author	publisher	year_of_publication	price	genre
1	The Alchemist	Paulo Coelho	HarperCollins	1988	109	Fiction
3	Atomic Habits	James Clear	Penguin	2021	75	Fiction

SELECT * FROM books ORDER BY author ASC LIMIT 1

The screenshot shows the phpMyAdmin interface for the 'library_db' database. A query has been run: `SELECT * FROM books ORDER BY author ASC LIMIT 1;`. The result is displayed in the 'Query results operations' section, showing one row from the 'books' table:

book_id	title	author	publisher	year_of_publication	price	genre
3	Atomic Habits	James Clear	Penguin	2021	75	Fiction

Lab 11.3: Grant SELECT permission to a user named librarian on the books table.

Ans:- GRANT SELECT ON library_db.books TO 'librarian'@'localhost';

The screenshot shows the phpMyAdmin interface for the 'library_db' database. In the 'Privileges' tab, a query has been run to grant SELECT permission on the 'books' table to the user 'librarian'@'localhost'. The results show the grant was successful, and the specific grant command is displayed.

```
SHOW GRANTS FOR 'librarian'@'localhost';
GRANT USAGE ON *.* TO 'librarian'@'localhost' IDENTIFIED BY '';
GRANT SELECT ON `library_db`.`books` TO 'librarian'@'localhost';
```

Lab 11.4: Grant INSERT and UPDATE permissions to the user admin on the members table.

Ans:- GRANT INSERT, UPDATE ON library_db.members TO 'admin'@'localhost';

The screenshot shows the phpMyAdmin interface for the 'library_db' database. In the 'Privileges' tab, a query has been run to grant INSERT and UPDATE permissions on the 'members' table to the user 'admin'@'localhost'. The results show the grant was successful, and the specific grant command is displayed.

```
SHOW GRANTS FOR 'admin'@'localhost';
GRANT USAGE ON *.* TO 'admin'@'localhost' IDENTIFIED BY '';
GRANT INSERT, UPDATE ON `library_db`.`members` TO 'admin'@'localhost';
```

Lab 12.3: Revoke the INSERT privilege from the user librarian on the books table.

Ans:- REVOKE INSERT ON library_db.books FROM 'librarian'@'localhost';

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'Privileges' tab is selected. A query has been run:

```
SHOW GRANTS FOR 'librarian'@'localhost';
```

The results show grants for the 'librarian' user:

```
GRANT USAGE ON *.* TO 'librarian'@'localhost' IDENTIFIED BY '';
GRANT SELECT ON 'library_db'.books TO 'librarian';
```

Below the results, there are options to Print, Copy to clipboard, or Create view.

Lab 12.4: Revoke all permissions from user admin on the members table.

Ans:- REVOKE ALL PRIVILEGES ON library_db.members FROM 'admin'@'localhost';

The screenshot shows the phpMyAdmin interface for the 'library_db' database. The 'Privileges' tab is selected. A query has been run:

```
SHOW GRANTS FOR 'admin'@'localhost';
```

The results show grants for the 'admin' user:

```
GRANT USAGE ON *.* TO 'admin'@'localhost' IDENTIFIED BY '';
```

Below the results, there are options to Print, Copy to clipboard, or Create view.

Lab 13.3: Use COMMIT after inserting multiple records into the books table, then make another insertion and perform a ROLLBACK.

Ans:- INSERT INTO books (title, author, price) VALUES

('Python Basics', 'Ravi Kumar', 280.00),

('Data Structures', 'Anil Mehta', 350.00);

COMMIT;

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar lists databases: information_schema, library_db, mysql, performance_schema, phpmyadmin, and test. The 'books' table under 'library_db' is selected. The main area displays the 'Structure' tab for the 'books' table, which has columns: book_id, title, author, and price. A SQL query at the top shows the current state of the table:

```
SELECT * FROM `books`
```

The table data is as follows:

book_id	title	author	price
1	Database Basics	John Smith	250.00
2	Learning SQL	Alice Brown	300.00
3	Web Development	David Lee	450.00
4	Python Basics	Ravi Kumar	280.00
5	Data Structures	Anil Mehta	350.00

Below the table, there are buttons for Edit, Copy, Delete, and Export. The bottom section contains a 'Query results operations' panel with options like Print, Copy to clipboard, Export, Display chart, and Create view.

```
INSERT INTO books (title, author, price)  
VALUES ('Machine Learning', 'Suresh Patel', 500.00);  
ROLLBACK;
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar lists databases like 'information_schema', 'library_db', 'mysql', etc. The main area displays the 'members' table with the following data:

member_id	member_name	join_date
1	Rahul Patel	2023-01-10
2	Neha Sharma	2024-03-15
3	Amit Verma	2025-02-05

Below the table, there are buttons for 'Edit', 'Copy', and 'Delete' for each row. The top navigation bar includes tabs for 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Triggers'.

Lab 13.4: Set a SAVEPOINT before making updates to the members table, perform some updates, and then roll back to the SAVEPOINT.

Ans:- UPDATE members

```
SET member_name = 'Rahul Shah'
```

```
WHERE member_id = 1;
```

```
SAVEPOINT sp1;
```

UPDATE members

```
SET member_name = 'Neha Verma'
```

```
WHERE member_id = 2;
```

UPDATE members

```
SET member_name = 'Amit Joshi'
```

```
WHERE member_id = 3;
```

```
ROLLBACK TO SAVEPOINT sp1;
```

COMMIT;

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The 'members' table is selected. The table structure includes columns: member_id, member_name, and join_date. There are three records:

member_id	member_name	join_date
1	Rahul Shah	2023-01-10
2	Neha Verma	2024-03-15
3	Amit Joshi	2025-02-05

The 'Edit' link for member_id 2 is highlighted with a red box. The browser address bar shows 'localhost/phpmyadmin/index.php?route=/sql&pos=0&db=library_db&table=members'.

Lab 14.3: Perform an INNER JOIN between books and authors tables to display the title of books and their respective authors' names.

Ans:- `SELECT b.title, a.author_name FROM books b INNER JOIN authors a ON b.author_id = a.author_id`

The screenshot shows the phpMyAdmin interface on a web browser. The left sidebar lists databases: information_schema, library_db, mysql, performance_schema, phpmyadmin, and test. The library_db database is selected. In the main area, under the 'books' table, a query has been run:

```
SELECT b.title, a.author_name FROM books b INNER JOIN authors a ON b.author_id = a.author_id;
```

The results show two rows:

title	author_name
Database Basics	John Smith
Learning SQL	Alice Brown

Below the results, there are 'Query results operations' buttons: Print, Copy to clipboard, Export, Display chart, and Create view.

Lab 14.4: Use a FULL OUTER JOIN to retrieve all records from the books and authors tables, including those with no matching entries in the other table.

Ans:-
SELECT b.title, a.author_name FROM books b LEFT JOIN authors a ON b.author_id = a.author_id

UNION

SELECT b.title, a.author_name FROM books b RIGHT JOIN authors a ON b.author_id = a.author_id

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar lists databases like 'information_schema', 'library_db', 'authors', 'books', 'members', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The 'books' table is selected. The main area displays the results of a UNION query:

```
SELECT b.title, a.author_name FROM books b LEFT JOIN authors a ON b.author_id = a.author_id UNION SELECT b.title, a.author_name FROM books b RIGHT JOIN authors a ON b.author_id = a.author_id;
```

The results table shows the following data:

title	author_name
Database Basics	John Smith
Learning SQL	Alice Brown
Web Development	NULL
Python Basics	NULL
Data Structures	NULL
Machine Learning	NULL
NULL	David Lee

Below the table are buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Lab 15.3: Group books by genre and display the total number of books in each genre.

Ans:- SELECT genre, COUNT(*) AS total_books FROM books GROUP BY genre

Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT genre, COUNT(*) AS total_books FROM books GROUP BY genre;
```

genre	total_books
NULL	3
Education	1
Literature	1
Technology	1

Lab 15.4: Group members by the year they joined and find the number of members who joined each year.

Ans:- SELECT YEAR(join_date) AS join_year, COUNT(*) AS total_members FROM members GROUP BY YEAR(join_date)

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

```
SELECT YEAR(join_date) AS join_year, COUNT(*) AS total_members FROM members GROUP BY YEAR(join_date);
```

join_year	total_members
2022	1
2023	2

Lab 16.3: Write a stored procedure to retrieve all books by a particular author.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE get_books_by_author(IN p_author VARCHAR(100))
```

```
BEGIN
```

```
    SELECT book_id, title, price
```

```
    FROM books
```

```
    WHERE author = p_author;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL get_books_by_author('John Smith');
```

The screenshot shows the phpMyAdmin interface on a Windows operating system. The browser tab is titled "localhost / 127.0.0.1 / library_db". The left sidebar shows the database structure with databases like information_schema, library_db, mysql, performance_schema, phpmyadmin, and test. The main panel is titled "members" under the "library_db" database. The top navigation bar includes "Browse", "Structure", "SQL", "Search", "Insert", "Export", "Import", "Privileges", "Operations", and "Triggers". A message at the top says "Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)". Below it, the SQL query "CALL get_books_by_author('John Smith');" is entered. The results table shows one row: book_id 1, title "Database Basics", and price 250.00. At the bottom, there are "Query results operations" buttons for Print, Copy to clipboard, and Create view.

Lab 16.4: Write a stored procedure that takes book_id as an argument and returns the price of the book.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE get_book_price(IN p_book_id INT)
```

```
BEGIN
```

```
    SELECT price
```

```
    FROM books
```

```
    WHERE book_id = p_book_id;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL get_book_price(1);
```

The screenshot shows the phpMyAdmin interface on a Windows operating system. The title bar indicates the browser window is titled 'python_assignment_17313991' and the address bar shows 'localhost / 127.0.0.1 / library_db'. The left sidebar lists databases: 'information_schema', 'library_db', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. Under 'library_db', there are tables: 'authors', 'books', and 'members'. The main panel shows the 'members' table structure. A query has been run in the SQL tab:

```
CALL get_book_price(1);
```

The results of the query are displayed in the 'Query results operations' section:

price
250.00

At the bottom of the interface, there is a 'Console' tab.

Lab 17.3: Create a view to show only the title, author, and price of books from the books table.

Ans:- CREATE VIEW view_books_basic AS SELECT title, author, price FROM books

SELECT * FROM view_books_basic;

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar shows the database structure with tables 'books' and 'members' under the 'library_db' schema. The main area displays the results of a query:

```
SELECT * FROM view_books_basic;
```

The results table shows the following data:

	title	author	price
<input type="checkbox"/>	Database Basics	John Smith	250.00
<input type="checkbox"/>	Learning SQL	Alice Brown	300.00
<input type="checkbox"/>	Web Development	David Lee	450.00
<input type="checkbox"/>	Python Basics	Ravi Kumar	280.00
<input type="checkbox"/>	Data Structures	Anil Mehta	350.00
<input type="checkbox"/>	Machine Learning	Suresh Patel	500.00

Lab 17.4: Create a view to display members who joined before 2020.

Ans:- CREATE VIEW view_members_before_2020 AS SELECT member_id, member_name, join_date FROM members WHERE join_date < '2020-01-01'

SELECT * FROM view_members_before_2020;

The screenshot shows the phpMyAdmin interface on a Windows desktop. The browser title is "localhost / 127.0.0.1 / library_db". The left sidebar shows the database structure for 'library_db', including tables like 'authors', 'books', and 'members', and views like 'view_books_basic' and 'view_members_before_2020'. The main query window contains the SQL command:

```
SELECT * FROM view_members_before_2020;
```

Below the query, the results show a table with four rows:

	member_id	member_name	join_date
1	Old Member		2018-04-10
2	Old Member		2018-04-10
3	Old Member		2018-04-10

At the bottom of the results window, there are operations buttons: Print, Copy to clipboard, Export, Display chart, and Create view.

Lab 18.3: Create a trigger to automatically update the last_modified timestamp of the books table whenever a record is updated.

Ans:- DELIMITER \$\$

CREATE TRIGGER trg_books_last_modified

BEFORE UPDATE ON books

FOR EACH ROW

BEGIN

SET NEW.last_modified = CURRENT_TIMESTAMP;

END\$\$

DELIMITER ;

The screenshot shows the phpMyAdmin interface for the library_db database. The left sidebar shows the database structure with tables like authors, books, and members. The main area displays the books table with the following data:

book_id	title	price	last_modified
1	Database Basics	260.00	2025-12-28 21:27:18
2	Learning SQL	300.00	2025-12-28 21:26:54
3	Web Development	450.00	2025-12-28 21:26:54
4	Python Basics	280.00	2025-12-28 21:26:54
5	Data Structures	350.00	2025-12-28 21:26:54
6	Machine Learning	500.00	2025-12-28 21:26:54

Lab 18.4: Create a trigger that inserts a log entry into a log_changes table whenever a DELETE operation is performed on the books table.

Ans:- DELIMITER \$\$

```
CREATE TRIGGER trg_books_delete_log
```

```
AFTER DELETE ON books
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO log_changes (book_id, action_type)
```

```
    VALUES (OLD.book_id, 'DELETE');
```

```
END$$
```

```
DELIMITER ;
```

The screenshot shows the phpMyAdmin interface for the library_db database. The left sidebar shows the schema structure with tables like authors, books, and log_changes. The main area displays the log_changes table with one row:

log_id	book_id	action_type	action_time
1	2	DELETE	2025-12-28 21:33:59

Below the table, there are buttons for Print, Copy to clipboard, Export, Display chart, and Create view.

Lab 19.3: Write a PL/SQL block to insert a new book into the books table and display a confirmation message.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE insert_book()
```

```
BEGIN
```

```
    INSERT INTO books (title, author, price)
```

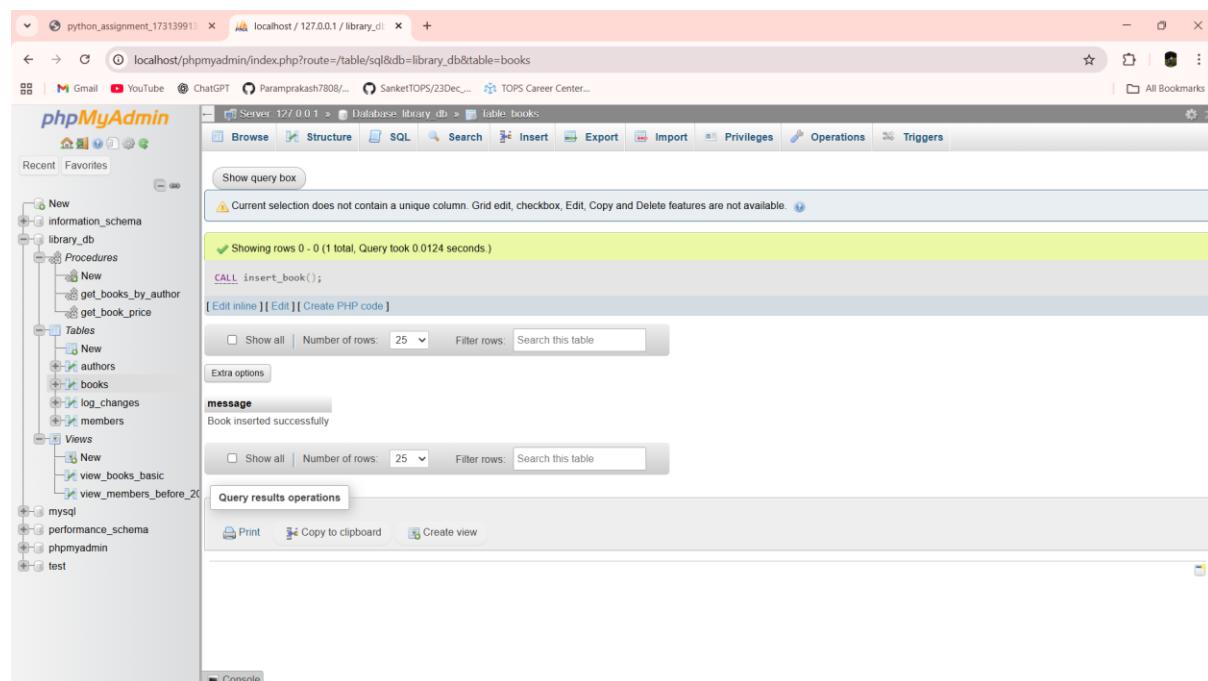
```
        VALUES ('Operating Systems', 'Andrew Tanenbaum', 550.00);
```

```
    SELECT 'Book inserted successfully' AS message;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL insert_book();
```



SELECT * FROM books;

The screenshot shows the phpMyAdmin interface for a database named 'library_db'. The left sidebar displays the database schema with tables like 'books', 'authors', and 'log_changes'. The main area shows the results of the query `SELECT * FROM books;`. The results table has the following data:

book_id	title	author	price	author_id	genre	last_modified
1	Database Basics	John Smith	260.00	1	Education	2025-12-28 21:27:18
3	Web Development	David Lee	450.00	NULL	Literature	2025-12-28 21:26:54
4	Python Basics	Ravi Kumar	280.00	NULL	NULL	2025-12-28 21:26:54
5	Data Structures	Anil Mehta	350.00	NULL	NULL	2025-12-28 21:26:54
6	Machine Learning	Suresh Patel	500.00	NULL	NULL	2025-12-28 21:26:54
7	Operating Systems	Andrew Tanenbaum	550.00	NULL	NULL	2025-12-28 21:36:07

Lab 19.4: Write a PL/SQL block to display the total number of books in the books table.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE total_books()
```

```
BEGIN
```

```
DECLARE total INT;
```

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

```
SELECT total AS total_number_of_books;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL total_books();
```

The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar displays the database structure with tables like 'books', 'authors', 'members', etc. The main query editor window contains the following SQL code:

```
CALL total_books();
```

The results pane shows the output of the procedure:

total_number_of_books
6

Below the results, there are 'Query results operations' buttons for Print, Copy to clipboard, and Create view.

Lab 20.3: Write a PL/SQL block to declare variables for book_id and price, assign values, and display the results.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE show_book_values()
```

```
BEGIN
```

```
    DECLARE v_book_id INT;
```

```
    DECLARE v_price DECIMAL(6,2);
```

```
    SET v_book_id = 101;
```

```
    SET v_price = 450.75;
```

```
    SELECT v_book_id AS book_id, v_price AS price;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL show_book_values();
```

The screenshot shows the phpMyAdmin interface on a web browser. The URL is `localhost / 127.0.0.1 / library_db`. The left sidebar shows the database structure under the `library_db` database, including tables like `books`, `authors`, and `members`. The main panel is titled "Table books" and shows the result of executing the stored procedure `show_book_values()`. The output is:

```
book_id    price
101        450.75
```

Below the results, there are "Query results operations" buttons for Print, Copy to clipboard, and Create view.

Lab 20.4: Write a PL/SQL block using constants and perform arithmetic operations on book prices.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE book_price_calculation()
```

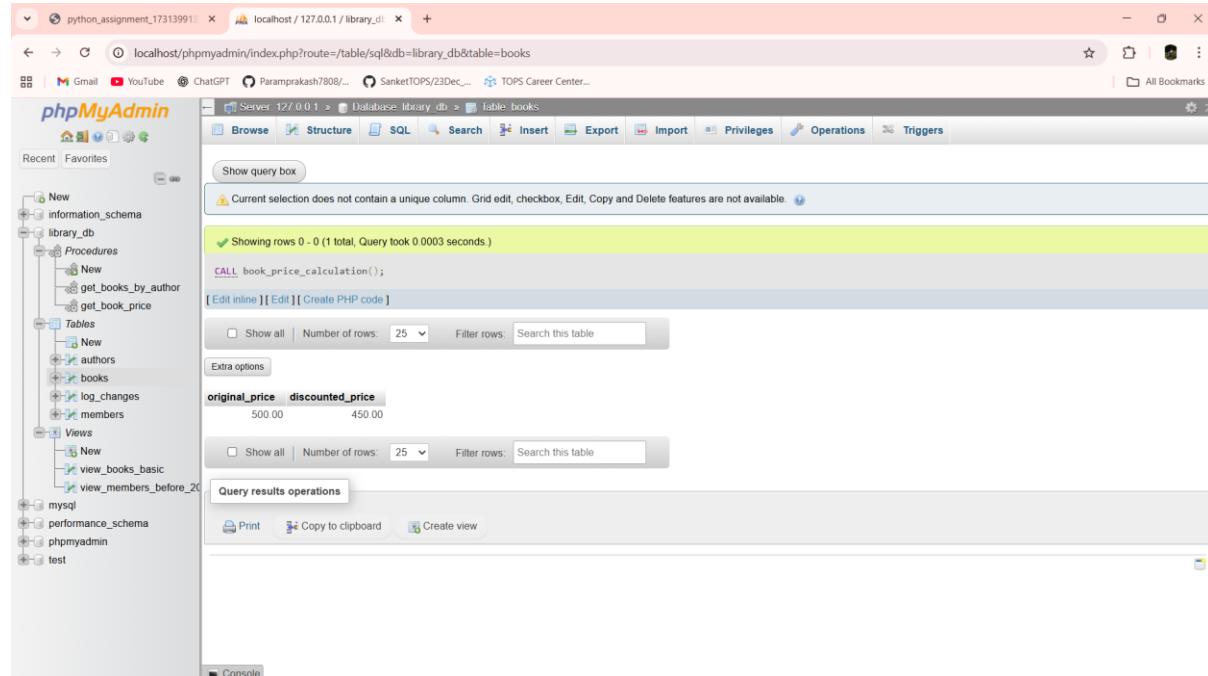
```
BEGIN
```

```
    DECLARE base_price DECIMAL(6,2) DEFAULT 500.00;  
  
    DECLARE discount_rate DECIMAL(4,2) DEFAULT 0.10;  
  
    DECLARE final_price DECIMAL(6,2);  
  
    SET final_price = base_price - (base_price * discount_rate);  
  
    SELECT base_price AS original_price,  
          final_price AS discounted_price;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL book_price_calculation();
```



The screenshot shows the phpMyAdmin interface for a MySQL database named 'library_db'. The left sidebar displays the database schema with tables like 'books', 'authors', 'members', and views like 'view_books_basic' and 'view_members_before_20'. The main panel shows the results of executing the stored procedure 'book_price_calculation()'. The output is:

```
Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)  
CALL book_price_calculation();  
[Edit inline] [Edit] [Create PHP code]  
original_price discounted_price  
500.00 450.00
```

Below the results, there are 'Query results operations' buttons for Print, Copy to clipboard, and Create view.

Lab 21.3: Write a PL/SQL block using IF-THEN-ELSE to check if a book's price is above \$100 and print a message accordingly.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE check_book_price()
```

```
BEGIN
```

```
    DECLARE v_price DECIMAL(6,2);
```

```
    SELECT price INTO v_price
```

```
    FROM books
```

```
    WHERE book_id = 1;
```

```
    IF v_price > 100 THEN
```

```
        SELECT 'Book price is above 100' AS message;
```

```
    ELSE
```

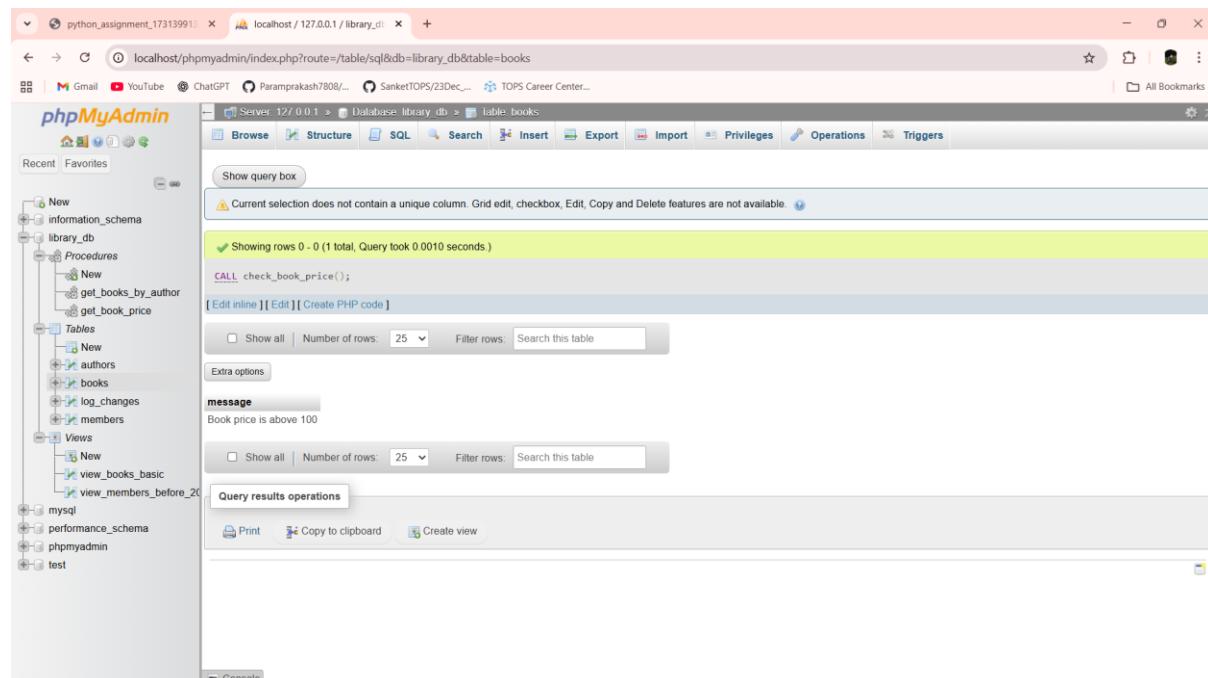
```
        SELECT 'Book price is 100 or below' AS message;
```

```
    END IF;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL check_book_price();
```



Lab 21.4: Use a FOR LOOP in PL/SQL to display the details of all books one by one.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE display_all_books()
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE v_id INT;
    DECLARE v_title VARCHAR(100);
    DECLARE v_price DECIMAL(6,2);
    DECLARE book_cursor CURSOR FOR
        SELECT book_id, title, price FROM books;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN book_cursor;
    read_loop: LOOP
        FETCH book_cursor INTO v_id, v_title, v_price;
        IF done = 1 THEN
            LEAVE read_loop;
        END IF;
        SELECT v_id AS book_id, v_title AS title, v_price AS price;
    END LOOP;
    CLOSE book_cursor;
END$$
```

DELIMITER ;

```
CALL display_all_books();
```

python_assignment_17313991 x localhost / 127.0.0.1 / library_db +

localhost/phpmyadmin/index.php?route=/database/routines&type=PROCEDURE&db=library_db

Gmail YouTube ChatGPT Paramprakash7809... SanketTOPS/23Dec... TOIPS Career Center... All Bookmarks

phpMyAdmin

Recent Favorites

information_schema

library_db

Procedures

New

book_price_calculation

check_book_price

display_all_books

get_books_by_author

get_book_price

insert_book

show_book_values

total_books

Tables

New

authors

books

log_changes

members

Views

New

view_books_basic

view_members_before

mysql

performance_schema

phpmyadmin

test

Structure SQL Search Export Import Operations Privileges Routines Events Triggers Designer

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

CALL `display_all_books`();

Execution results of routine 'display_all_books'

book_id	title	price
1	Database Basics	260.00
3	Web Development	450.00
4	Python Basics	280.00
5	Data Structures	350.00
6	Machine Learning	500.00
7	Operating Systems	550.00

Routines

Check all Export Drop

Search Create new routine

Console

The screenshot shows the phpMyAdmin interface for a database named 'library_db'. In the left sidebar, under 'Procedures', there is a single entry: 'display_all_books'. The main panel displays the execution results of this procedure. A green message bar at the top indicates that the query was successful and one row was affected. Below this, the SQL command 'CALL `display_all_books`();' is shown. The results are presented in a table with columns 'book_id', 'title', and 'price'. The data includes seven books: Database Basics (260.00), Web Development (450.00), Python Basics (280.00), Data Structures (350.00), Machine Learning (500.00), and Operating Systems (550.00). The 'Routines' section at the bottom is currently empty.

Lab 22.3: Write a PL/SQL block using an explicit cursor to fetch and display all records from the members table.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE show_all_members()
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE v_id INT;
    DECLARE v_name VARCHAR(100);
    DECLARE v_join_date DATE;
    DECLARE member_cursor CURSOR FOR
        SELECT member_id, member_name, join_date FROM members;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN member_cursor;
    read_loop: LOOP
        FETCH member_cursor INTO v_id, v_name, v_join_date;
        IF done = 1 THEN
            LEAVE read_loop;
        END IF;
        SELECT v_id AS member_id,
               v_name AS member_name,
               v_join_date AS join_date;
    END LOOP;
    CLOSE member_cursor;
END$$
DELIMITER ;
```

```
CALL show_all_members();
```

python_assignment_17313991 x localhost / 127.0.0.1 / library_db +

localhost/phpmyadmin/index.php?route=/database/routines&type=PROCEDURE&db=library_db

Gmail YouTube ChatGPT Paramprakash7809... SanketTOPS/23Dec... TOPS Career Center... All Bookmarks

phpMyAdmin

Recent Favorites

information_schema

library_db

- Procedures
 - New
 - book_price_calculation
 - check_book_price
 - display_all_books
 - get_books_by_author
 - get_book_price
 - insert_book
 - show_all_members
 - show_book_values
 - total_books
- Tables
 - New
 - authors
 - books
 - log_changes
 - members
- Views
 - New
 - view_books_basic
 - view_members_before

mysqldump performance_schema phpmyadmin test

Server 127.0.0.1 > Database: library_db

Structure SQL Search Export Import Operations Privileges Routines Events Triggers Designer

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

CALL `show_all_members`();

Execution results of routine 'show_all_members'

member_id	member_name	join_date
1	Rahul Shah	2022-05-10
2	Neha Verma	2023-08-15
3	Amit Joshi	2023-11-20
4	Old Member	2018-04-10

Routines

Check all

Name	Type	Returns	Edit	Execute	Export	Drop
book_price_calculation	PROCEDURE		<input type="button" value="Edit"/>	<input type="button" value="Execute"/>	<input type="button" value="Export"/>	<input type="button" value="Drop"/>
check_book_price	PROCEDURE		<input type="button" value="Edit"/>	<input type="button" value="Execute"/>	<input type="button" value="Export"/>	<input type="button" value="Drop"/>
display_all_books	PROCEDURE		<input type="button" value="Edit"/>	<input type="button" value="Execute"/>	<input type="button" value="Export"/>	<input type="button" value="Drop"/>
Console						

Lab 22.4: Create a cursor to retrieve books by a particular author and display their titles.

Ans:- DELIMITER \$\$

```
CREATE PROCEDURE books_by_author(IN p_author VARCHAR(100))
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE v_title VARCHAR(100);
    DECLARE book_cursor CURSOR FOR
        SELECT title FROM books WHERE author = p_author;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN book_cursor;
    read_loop: LOOP
        FETCH book_cursor INTO v_title;
        IF done = 1 THEN
            LEAVE read_loop;
        END IF;
        SELECT v_title AS book_title;
    END LOOP;
    CLOSE book_cursor;
END$$
DELIMITER ;
```

```
CALL books_by_author('John Smith');
```

Screenshot of the phpMyAdmin interface showing the library_db database.

Server: 127.0.0.1 | **Database:** library_db

Structure | **SQL** | **Search** | **Query** | **Export** | **Import** | **Operations** | **Privileges** | **Routines** | **Events** | **Triggers** | **Designer**

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
CALL books_by_author('John Smith');
```

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

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

book_title

Database Basics

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print | Copy to clipboard | Create view

Console

Information Schema

library_db

- Procedures**
 - New
 - book_price_calculation
 - check_book_price
 - display_all_books
 - get_books_by_author
 - get_book_price
 - insert_book
 - show_all_members
 - show_book_values
 - total_books
- Tables**
 - New
 - authors
 - books
 - log_changes
 - members
- Views**
 - New
 - view_books_basic
 - view_members_before

mysql

performance_schema

phpmyadmin

test

Lab 23.3: Perform a transaction that includes inserting a new member, setting a SAVEPOINT, and rolling back to the savepoint after making updates.

Ans:- START TRANSACTION;

```
INSERT INTO members (member_name, join_date)
```

```
VALUES ('Temporary Member', '2024-05-01');
```

```
SAVEPOINT sp_member;
```

```
UPDATE members
```

```
SET member_name = 'Updated Member'
```

```
WHERE member_name = 'Temporary Member';
```

```
ROLLBACK TO SAVEPOINT sp_member;
```

```
COMMIT;
```

```
SELECT * FROM members;
```

member_id	member_name	join_date
1	Rahul Shah	2022-05-10
2	Neha Verma	2023-08-15
3	Amit Joshi	2023-11-20
4	Old Member	2018-04-10
5	Updated Member	2024-05-01

Lab 23.4: Use COMMIT after successfully inserting multiple books into the books table, then use ROLLBACK to undo a set of changes made after a savepoint.

Ans:- START TRANSACTION;

```
INSERT INTO books (title, author, price)
```

```
VALUES
```

```
('Cloud Computing', 'Raj Mehta', 400),
```

```
('Data Mining', 'Anita Rao', 450);
```

```
COMMIT;
```

```
START TRANSACTION;
```

```
SAVEPOINT sp_books;
```

```
UPDATE books SET price = price + 50 WHERE title = 'Cloud Computing';
```

```
DELETE FROM books WHERE title = 'Data Mining';
```

```
ROLLBACK TO SAVEPOINT sp_books;
```

```
COMMIT;
```

```
SELECT * FROM books;
```

Screenshot of the phpMyAdmin interface showing the 'books' table in the 'library_db' database.

The left sidebar shows the database structure:

- Information Schema
- library_db
 - Procedures
 - New
 - books_by_author
 - book_price_calculation
 - check_book_price
 - display_all_books
 - get_books_by_author
 - get_book_price
 - insert_book
 - show_all_members
 - show_book_values
 - total_books
 - Tables
 - New
 - authors
 - books
 - log_changes
 - members
 - Views
 - New
 - view_books_basic
 - view_members_before
- mysql
- performance_schema

The main panel displays the 'books' table data:

book_id	title	author	price	author_id	genre	last_modified
1	Database Basics	John Smith	260.00	1	Education	2025-12-28 21:27:18
3	Web Development	David Lee	450.00	NULL	Literature	2025-12-28 21:26:54
4	Python Basics	Ravi Kumar	280.00	NULL	NULL	2025-12-28 21:26:54
5	Data Structures	Anil Mehta	350.00	NULL	NULL	2025-12-28 21:26:54
6	Machine Learning	Suresh Patel	500.00	NULL	NULL	2025-12-28 21:26:54
7	Operating Systems	Andrew Tanenbaum	550.00	NULL	NULL	2025-12-28 21:36:07
9	Cloud Computing	Raj Mehta	450.00	NULL	NULL	2025-12-28 21:55:36

Below the table, there are buttons for Edit, Copy, Delete, and Export. The 'Query results operations' section includes Print, Copy to clipboard, Export, Display chart, and Create view.