~EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

-Harsh Chauhan

Introduction to SQL

- Lab 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.

```
CREATE DATABASE library_db;


CREATE TABLE books ( book_id int PRIMARY KEY,
    title VARCHAR(255) NOT NULL, author
    VARCHAR(200) NOT NULL, publisher
    VARCHAR(200) NOT NULL, year_of_publication INT,
    price int );

INSERT INTO books (book_id, title, author, publisher,
    year_of_publication, price) VALUES
(1, 'one piece', 'oda', 'shueisha', 1997, 199),
(2, 'the greatest estate developer', 'kim hyunsoo', 'asura
    scans', 2019, 210),
(3, 'spy x family', 'endou tatsuya', 'shueisha', 2019, 150),
(4, 'the apothecary diaries', 'hyuuga natsu', 'square enix
    books', 2011, 180),
(5, 'Hells paradise ', 'kaku yuuji', 'shueisha', 2018, 160);
```

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

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

←T→	book_id	title	author	publisher	year_of_publication	price
Edit Copy Delete	1	one piece	oda	shueisha	1997	199
Edit Copy Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2011	180
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2018	160

Check all *With selected:* Edit Copy Delete Export

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**book_id**	int(11)			No	*None*			Change Drop More
2	**title**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
3	**author**	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
4	**publisher**	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
5	**year_of_publication**	int(11)			Yes	*NULL*			Change Drop More
6	**price**	int(11)			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 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.

CREATE TABLE members (member_id INT PRIMARY KEY, member_name VARCHAR(100) NOT NULL, date_of_membership DATE NOT NULL,email VARCHAR(100) NOT NULL);

INSERT INTO members (member_id, member_name, date_of_membership, email) VALUES
(1, 'harsh', '2023-04-11', 'harsh@email.com'),
(2, 'luffy', '2022-06-22', 'monkey@email.com'),
(3, 'anya', '2024-01-10', 'forger@email.com'),
(4, 'llyod', '2024-03-18', 'fronterra@email.com'),
(5, 'maomao', '2024-05-05', 'hehe@email.com');

Extra options

			member_id	member_name	date_of_membership	email
Edit	Copy	Delete	1	harsh	2023-04-11	harsh@email.com
Edit	Copy	Delete	2	luffy	2022-06-22	monkey@email.com
Edit	Copy	Delete	3	anya	2024-01-10	forger@email.com
Edit	Copy	Delete	4	llyod	2024-03-18	fronterra@email.com
Edit	Copy	Delete	5	maomao	2024-05-05	hehe@email.com

Check all With selected: Edit Copy Delete Export

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**member_id**	int(11)			No	*None*			Change Drop More
2	**member_name**	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
3	**date_of_membership**	date			No	*None*			Change Drop More
4	**email**	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More

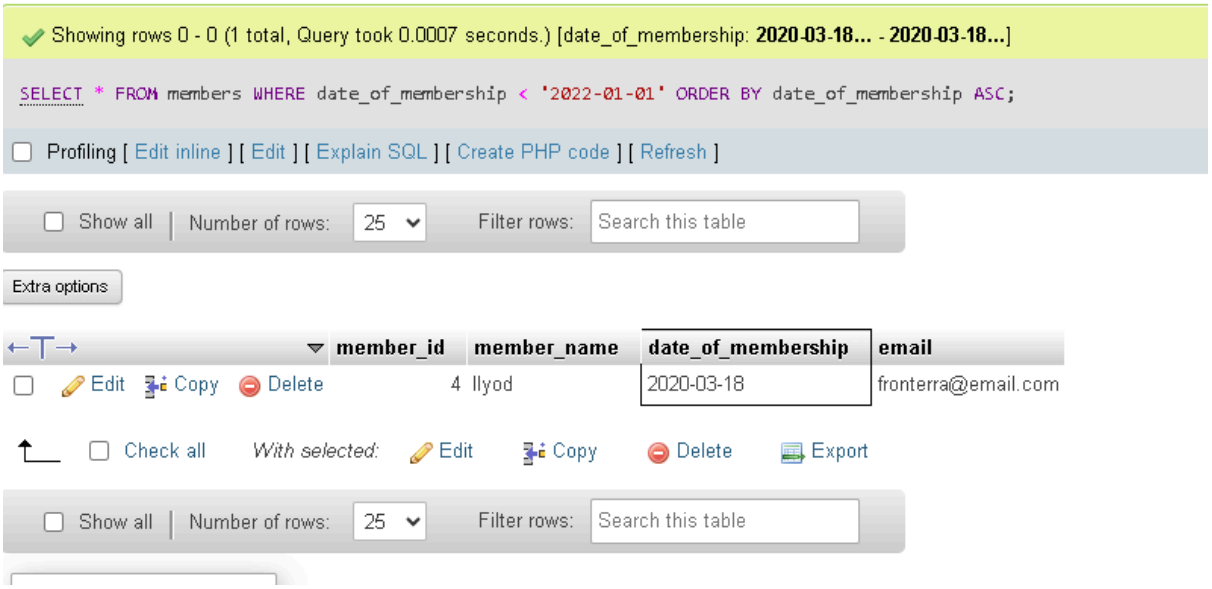
Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Add to central columns Remove from central columns

SQL Syntax

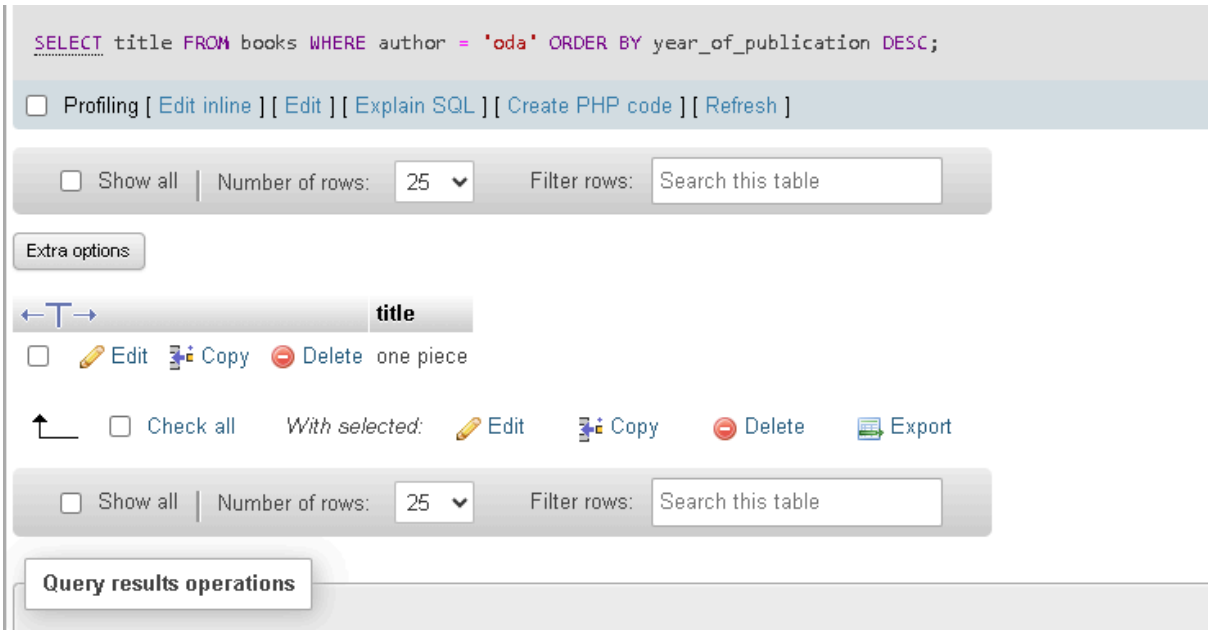
- Lab 3: Retrieve all members who joined the library before 2022. Use appropriate SQL syntax with WHERE and ORDER BY.

SELECT * FROM members WHERE date_of_membership < '2022-01-01' ORDER BY date_of_membership ASC;



- Lab 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.

SELECT title FROM books WHERE author = 'oda' ORDER BY year_of_publication DESC;

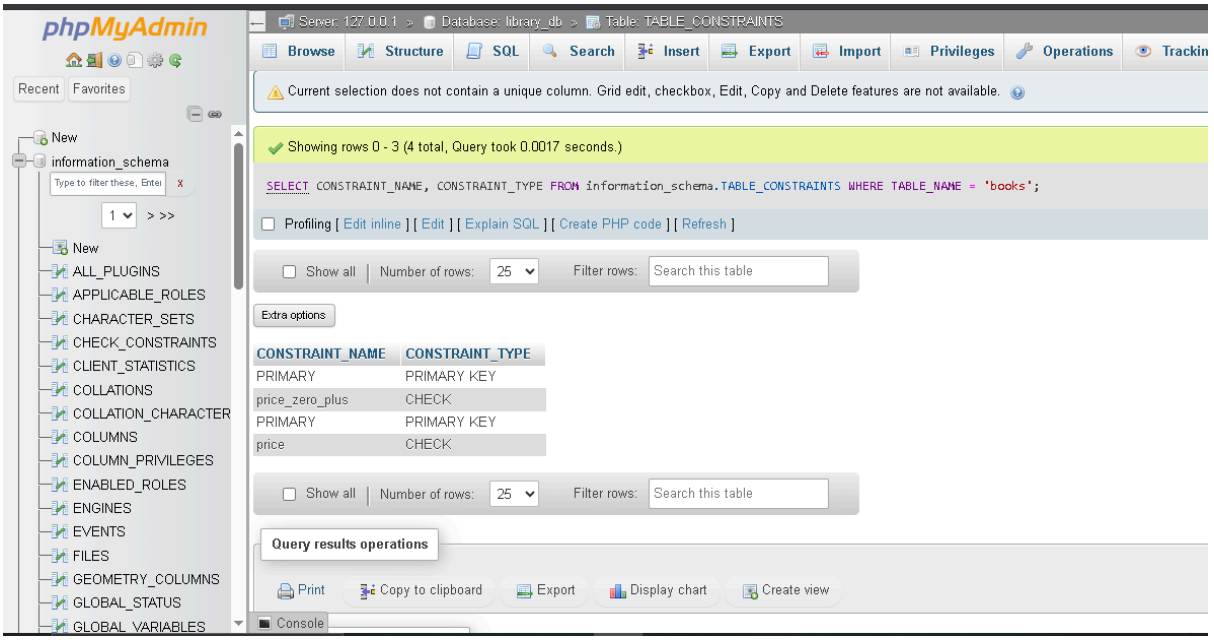


SQL Constraints

- Lab 3: Add a CHECK constraint to ensure that the price of books in the books table is greater than 0.

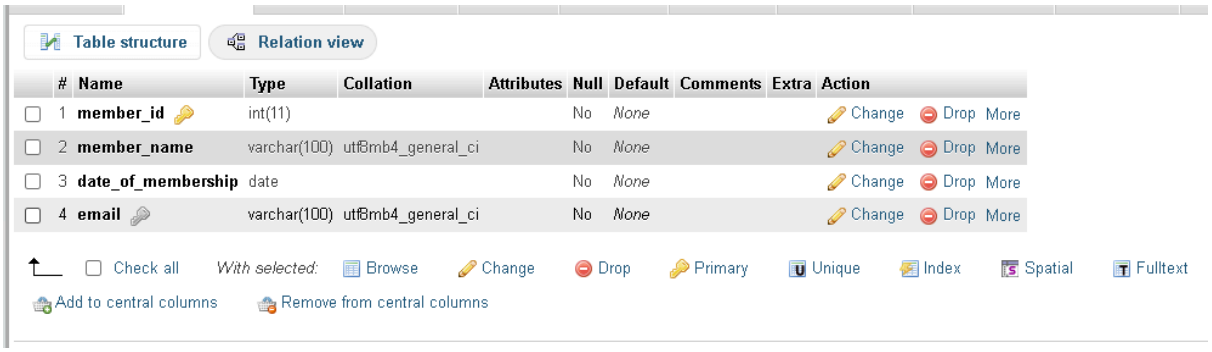
ALTER TABLE books ADD CONSTRAINT price_zero_plus CHECK (price > 0);

SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE FROM information_schema.TABLE_CONSTRAINTS WHERE TABLE_NAME = 'books';



- Lab 4: Modify the members table to add a UNIQUE constraint on the email column, ensuring that each member has a unique email address.

ALTER TABLE members ADD CONSTRAINT unique_email UNIQUE (email);



Main SQL Commands and Sub-commands (DDL)

- Lab 3: Create a table authors with the following columns: author_id, first_name, last_name, and country. Set author_id as the primary key.

CREATE TABLE authors (author_id INT PRIMARY KEY, first_name VARCHAR(100), last_name VARCHAR(100), country VARCHAR(100));

Table structure | Relation view

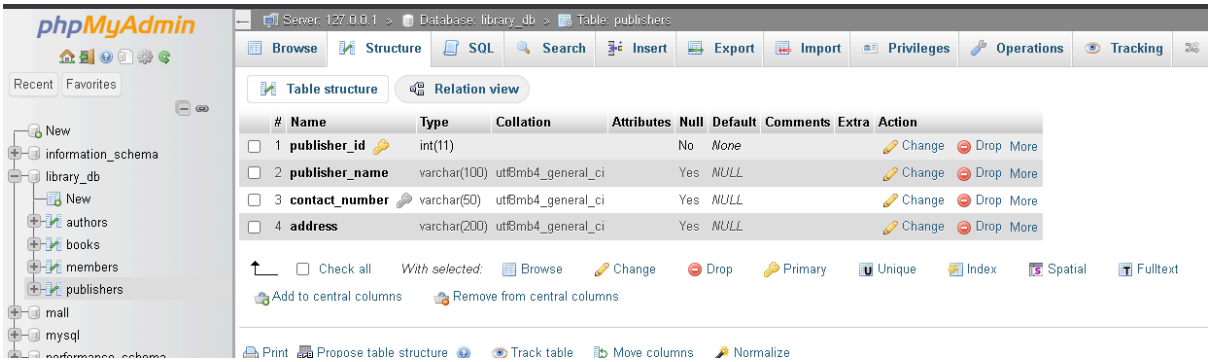
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**author_id**	int(11)			No	*None*			Change Drop More
2	**first_name**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
3	**last_name**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
4	**country**	varchar(100)	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 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.

CREATE TABLE publishers (publisher_id INT PRIMARY KEY, publisher_name VARCHAR(100), contact_number VARCHAR(50) UNIQUE, address VARCHAR(200));



ALTER Command

- Lab 3: Add a new column genre to the books table. Update the genre for all existing records.

```
ALTER TABLE books ADD genre VARCHAR(100);

UPDATE books SET genre = 'Manga' WHERE book_id =
    1;
UPDATE books SET genre = 'Manhwa' WHERE book_id =
    2;
UPDATE books SET genre = 'Manga' WHERE book_id =
    3;
UPDATE books SET genre = 'Manga' WHERE book_id =
    4;
UPDATE books SET genre = 'Manga' WHERE book_id =
    5;
```

Extra options

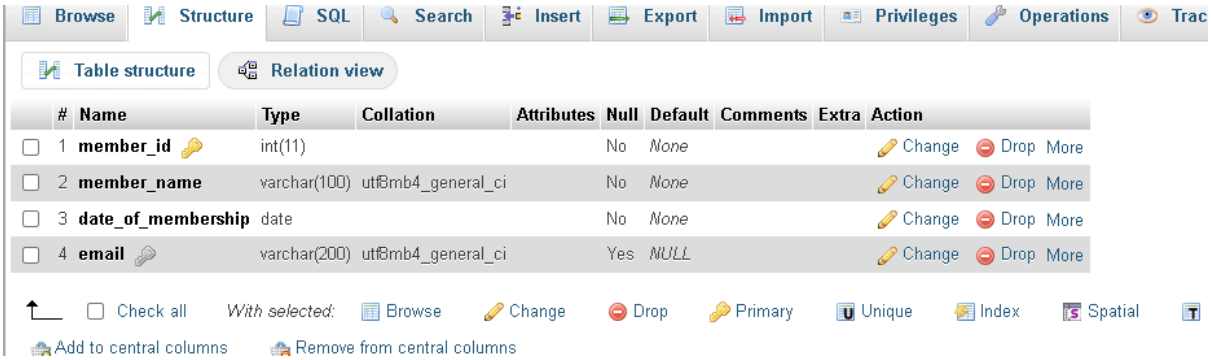
←T→			book_id	title	author	publisher	year_of_publication	price	genre
Edit	Copy	Delete	1	one piece	oda	shueisha	1997	199	Manga
Edit	Copy	Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa
Edit	Copy	Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit	Copy	Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2011	180	Manga
Edit	Copy	Delete	5	Hells paradise	kaku yuuji	shueisha	2018	160	Manga

Check all With selected: Edit Copy Delete Export

- Lab 4: Modify the members table to increase the length of the email column to 100 characters.

Its length is already 100 so i will set it to 200.

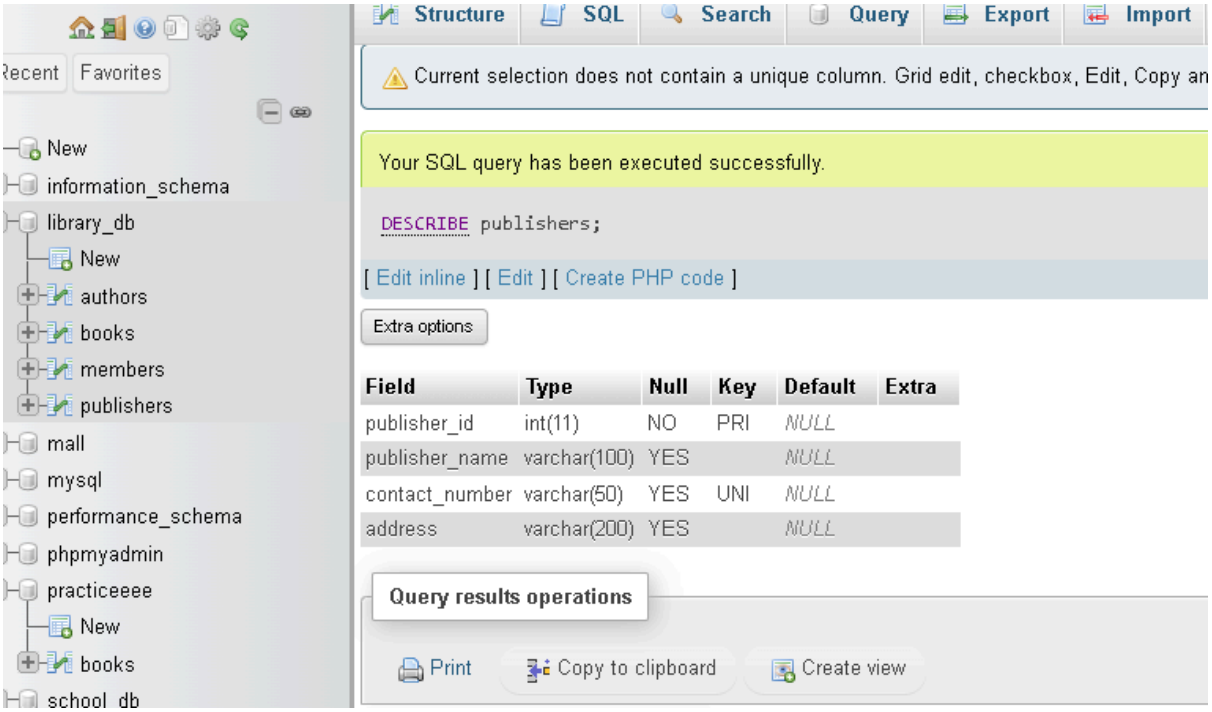
ALTER TABLE members MODIFY email varchar(200)



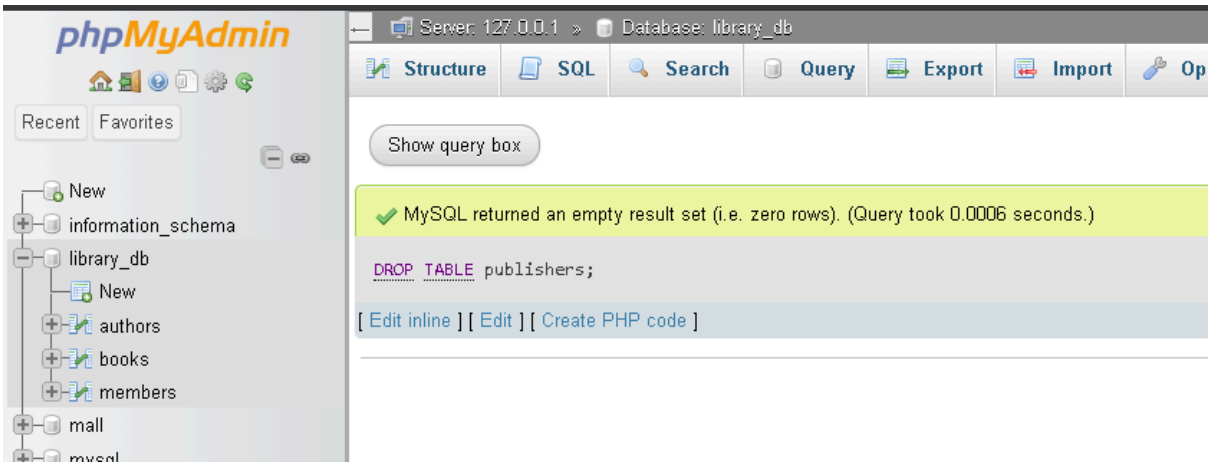
DROP Command

- Lab 3: Drop the publishers table from the database after verifying its structure.

DESCRIBE publishers;

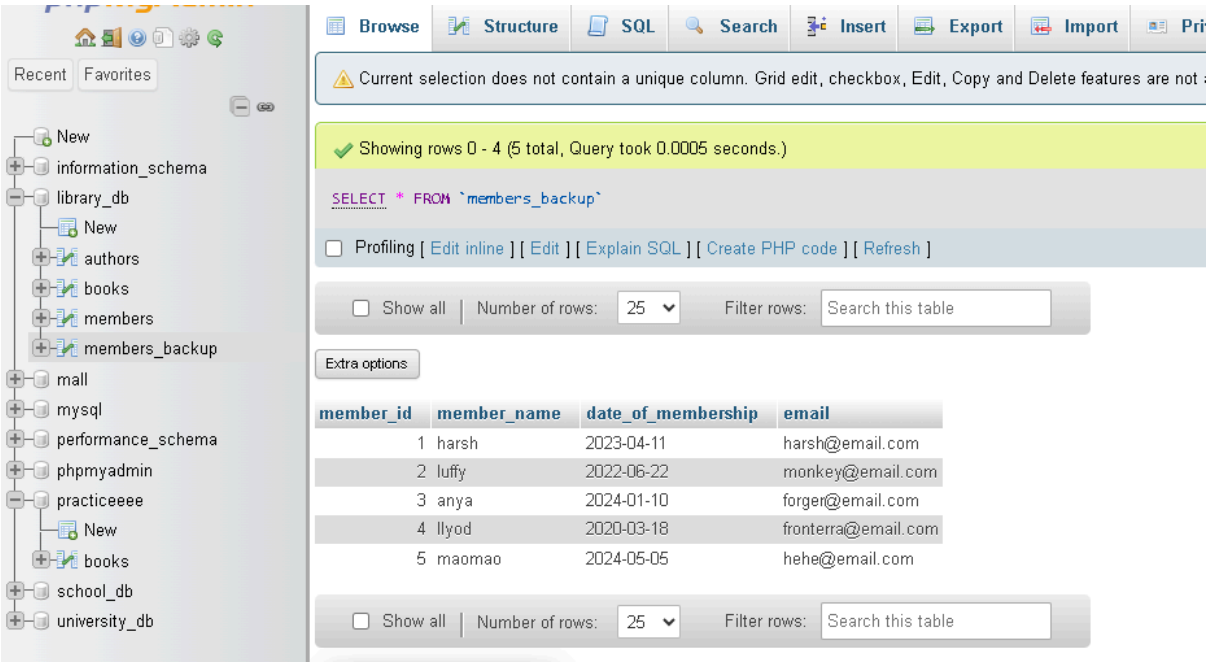


DROP TABLE publishers;

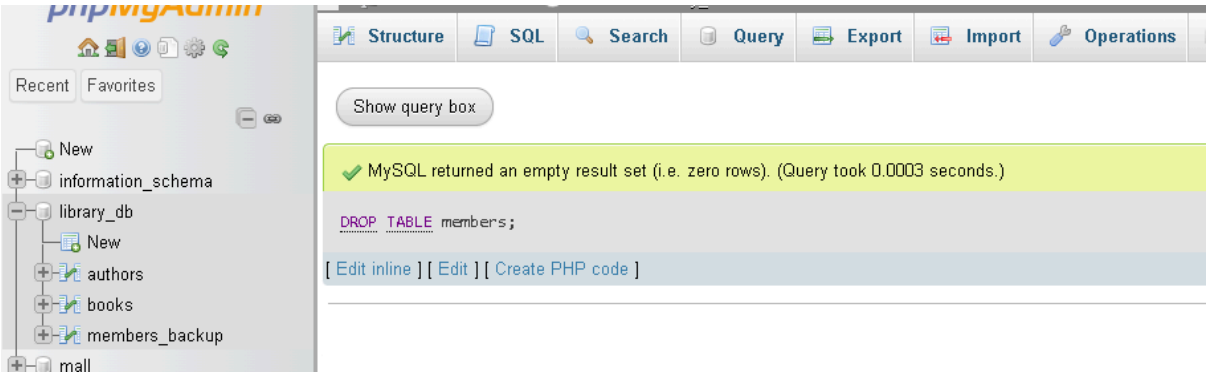


• Lab 4: Create a backup of the members table and then drop the original members table.

CREATE TABLE members_backup AS SELECT * FROM members;



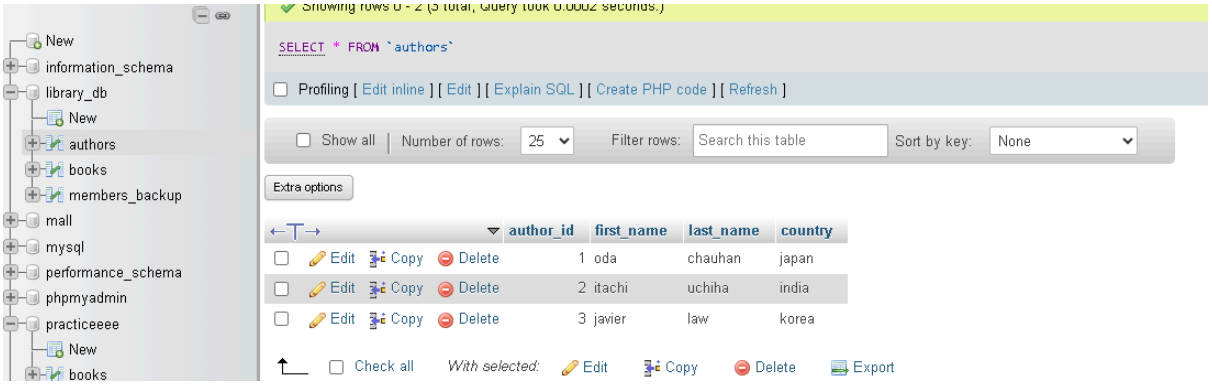
DROP TABLE members;



Data Manipulation Language (DML)

- Lab 4: Insert three new authors into the authors table, then update the last name of one of the authors.

INSERT INTO authors(author_id, first_name, last_name, country)
VALUES(1,'oda','chauhan','japan'),(2,'itachi','uchiha','india'),(3,'javier','law','korea')



UPDATE authors SET last_name = 'monkey' WHERE author_id = 1

- Lab 5: Delete a book from the books table where the price is higher than $100.

DELETE FROM books WHERE price > 100 LIMIT 1;

Extra options

	book_id	title	author	publisher	year_of_publication	price	genre
Edit Copy Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2011	180	Manga
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2018	160	Manga

Check all *With selected:* Edit Copy Delete Export

UPDATE Command

- Lab 3: Update the year_of_publication of a book with a specific book_id.

UPDATE books SET year_of_publication = 2021 WHERE book_id = 4

Extra options

←T→				book_id	title	author	publisher	year_of_publication	price	genre
☐	Edit	Copy	Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa
☐	Edit	Copy	Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
☐	Edit	Copy	Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga
☐	Edit	Copy	Delete	5	Hells paradise	kaku yuuji	shueisha	2018	160	Manga

• Lab 4: Increase the price of all books published before 2015 by 10%.

Bcz i didnt set any book's published year before 2015

UPDATE `books` SET `year_of_publication` = '2012' WHERE `books`.`book_id` = 5;

←T→	book_id	title	author	publisher	year_of_publication	price	genre
Edit Copy Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2012	160	Manga

Check all *With selected:* Edit Copy Delete Export

UPDATE books SET price = price * 1.10 WHERE year_of_publication < 2015

←T→	book_id	title	author	publisher	year_of_publication	price	genre
Edit Copy Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2012	176	Manga

Check all *With selected:* Edit Copy Delete Export

DELETE Command

• Lab 3: Remove all members who joined before 2020 from the members table.

```
CREATE TABLE members (member_id INT PRIMARY
    KEY, member_name VARCHAR(100) NOT NULL,
    date_of_membership DATE NOT NULL,email
    VARCHAR(100) NOT NULL);

INSERT INTO members (member_id, member_name,
    date_of_membership, email) VALUES
(1, 'harsh', '2023-04-11', 'harsh@email.com'),
(2, 'luffy', '2022-06-22', 'monkey@email.com'),
(3, 'anya', '2015-01-10', 'forger@email.com'),
(4, 'llyod', '2024-03-18', 'fronterra@email.com'),
(5, 'maomao', '2019-05-05', 'hehe@email.com');
```

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

Extra options

←T→	member_id	member_name	date_of_membership	email
Edit Copy Delete	1	harsh	2023-04-11	harsh@email.com
Edit Copy Delete	2	luffy	2022-06-22	monkey@email.com
Edit Copy Delete	3	anya	2015-01-10	forger@email.com
Edit Copy Delete	4	llyod	2024-03-18	fronterra@email.com
Edit Copy Delete	5	maomao	2019-05-05	hehe@email.com

Check all *With selected:* Edit Copy Delete Export

DELETE FROM members WHERE date_of_membership < '2020-01-01';

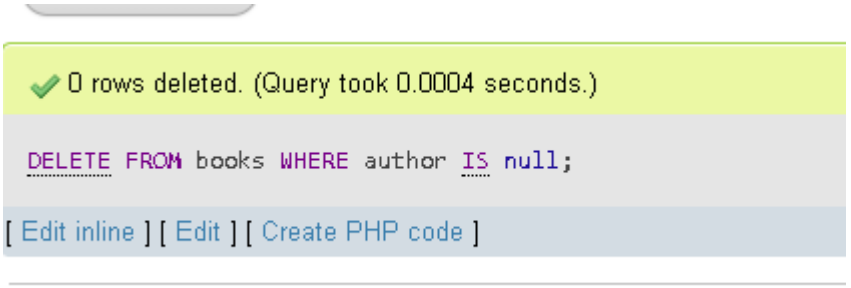
Extra options

				member_id	member_name	date_of_membership	email
☐	Edit	Copy	Delete	1	harsh	2023-04-11	harsh@email.com
☐	Edit	Copy	Delete	2	luffy	2022-06-22	monkey@email.com
☐	Edit	Copy	Delete	4	llyod	2024-03-18	fronterra@email.com

Check all *With selected:* Edit Copy Delete Export

- Lab 4: Delete all books that have a NULL value in the author column.

DELETE FROM books WHERE author IS null



Data Query Language (DQL)

- Lab 4: Write a query to retrieve all books with price between $150 and $200.

SELECT * FROM books WHERE price BETWEEN 150 AND 200

```
SELECT * FROM books WHERE price BETWEEN 150 AND 200;
```

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

	book_id	title	author	publisher	year_of_publication	price	genre
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2012	176	Manga

Check all *With selected:* Edit Copy Delete Export

• Lab 5: Retrieve the list of books sorted by author in ascending order and limit the results to the top 3 entries.

SELECT * FROM books ORDER BY author ASC LIMIT 3

```
SELECT * FROM books ORDER BY author ASC LIMIT 3;
```

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

Extra options

←T→	book_id	title	author 1	publisher	year_of_publication	price	genre
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2012	176	Manga

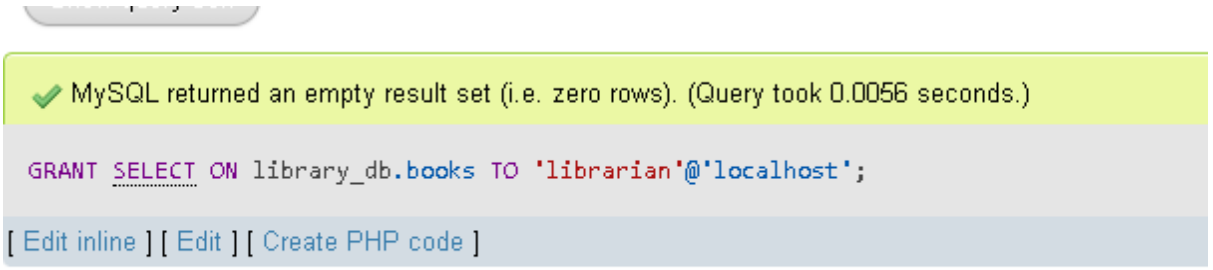
Check all With selected: Edit Copy Delete Export

Data Control Language (DCL)

• Lab 3: Grant SELECT permission to a user named librarian on the books table.

CREATE USER 'librarian'@'localhost' IDENTIFIED BY 'harsh';

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



• Lab 4: Grant INSERT and UPDATE permissions to the user admin on the members table.

GRANT INSERT, UPDATE ON library_db.members TO 'librarian'@'localhost';

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0050 seconds.)

```
GRANT INSERT, UPDATE ON library_db.members TO 'librarian'@'localhost';
```

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

SELECT * FROM `TABLE_PRIVILEGES`

Showing rows 0 - 4 (5 total, Query took 0.0007 seconds.)

```
SELECT * FROM `TABLE_PRIVILEGES`
```

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

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

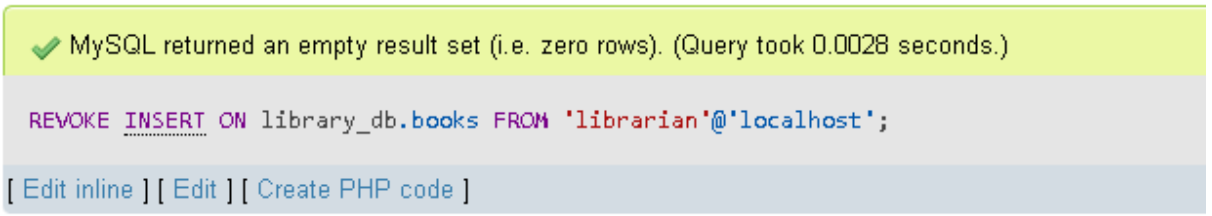
Extra options

GRANTEE	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	PRIVILEGE_TYPE	IS_GRANTABLE
'user2'@'localhost'	def	school_db	courses	INSERT	NO
'librarian'@'localhost'	def	library_db	books	SELECT	NO
'user1'@'localhost'	def	school_db	courses	SELECT	NO
'librarian'@'localhost'	def	library_db	members	INSERT	NO
'librarian'@'localhost'	def	library_db	members	UPDATE	NO

REVOKE Command

- Lab 3: Revoke the INSERT privilege from the user librarian on the books table.

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



• Lab 4: Revoke all permissions from user admin on the members table.

REVOKE ALL PRIVILEGES ON library_db.members FROM 'librarian'@'localhost';

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0022 seconds.)

```
REVOKE ALL PRIVILEGES ON library_db.members FROM 'librarian'@'localhost';
```

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

Transaction Control Language (TCL)

- Lab 3: Use COMMIT after inserting multiple records into the books table, then make another insertion and perform a ROLLBACK.

```
START TRANSACTION;

INSERT INTO books (book_id, title, author, publisher,
    year_of_publication, price, genre)
VALUES(6, 'Blue Lock', 'Muneyuki Kaneshiro',
    'Kodansha', 2018, 220, 'Sports'),
      (7, 'Jujutsu Kaisen', 'Gege Akutami', 'Shueisha', 2018,
    210, 'Supernatural');

COMMIT;

INSERT INTO books (book_id, title, author, publisher,
    year_of_publication, price, genre)
VALUES(8, 'Demon Slayer', 'Koyoharu Gotouge',
    'Shueisha', 2016, 200, 'Action');

ROLLBACK;
```

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

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

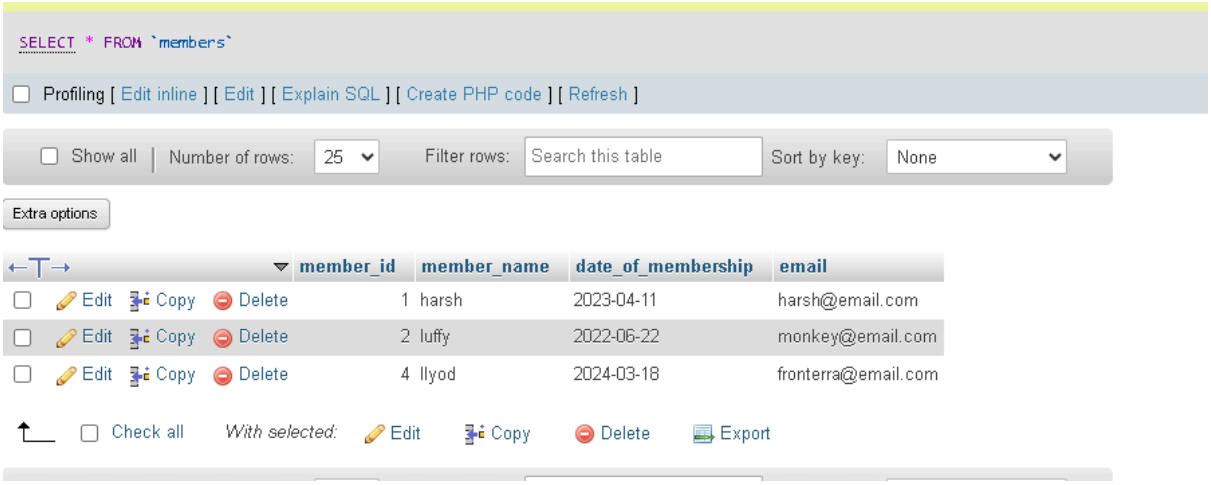
Extra options

←T→	book_id	title	author	publisher	year_of_publication	price	genre
Edit Copy Delete	2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa
Edit Copy Delete	3	spy x family	endou tatsuya	shueisha	2019	150	Manga
Edit Copy Delete	4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga
Edit Copy Delete	5	Hells paradise	kaku yuuji	shueisha	2012	176	Manga
Edit Copy Delete	6	Blue Lock	Muneyuki Kaneshiro	Kodansha	2018	220	Sports
Edit Copy Delete	7	Jujutsu Kaisen	Gege Akutami	Shueisha	2018	210	Supernatural
Edit Copy Delete	8	Demon Slayer	Koyoharu Gotouge	Shueisha	2016	200	Action

Check all *With selected:* Edit Copy Delete Export

- Lab 4: Set a SAVEPOINT before making updates to the members table, perform some updates, and then roll back to the SAVEPOINT.

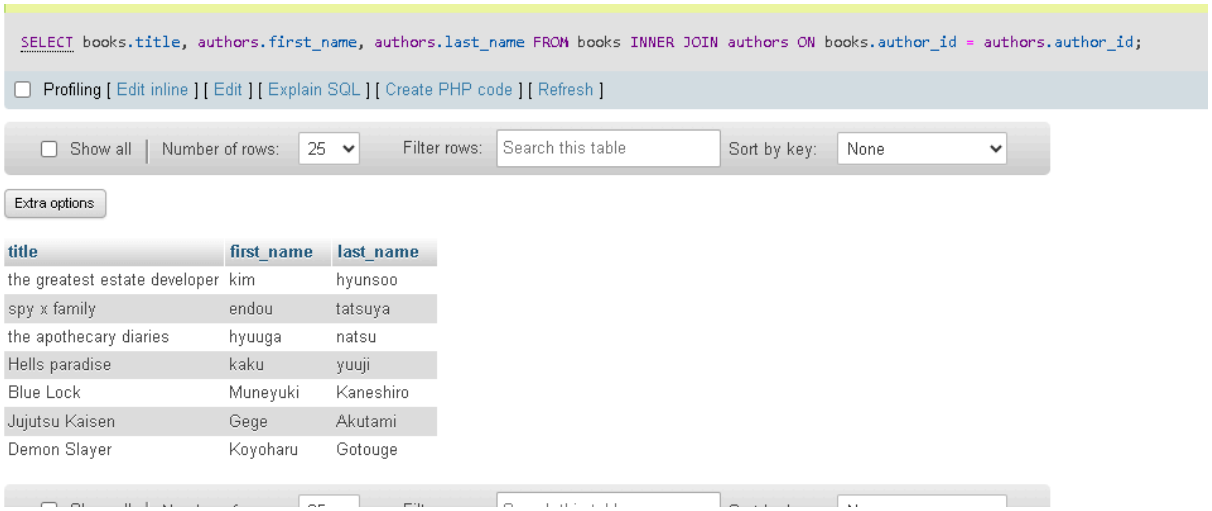
```
START TRANSACTION;

SAVEPOINT before_update;

UPDATE members SET member_name = 'Ash' WHERE
    member_id = 1;
UPDATE members SET member_name = 'Monkey D.
    Luffy' WHERE member_id = 2;

ROLLBACK TO SAVEPOINT before_update;

COMMIT;
```



SQL Joins

- Lab 3: Perform an INNER JOIN between books and authors tables to display the title of books and their respective authors' names.

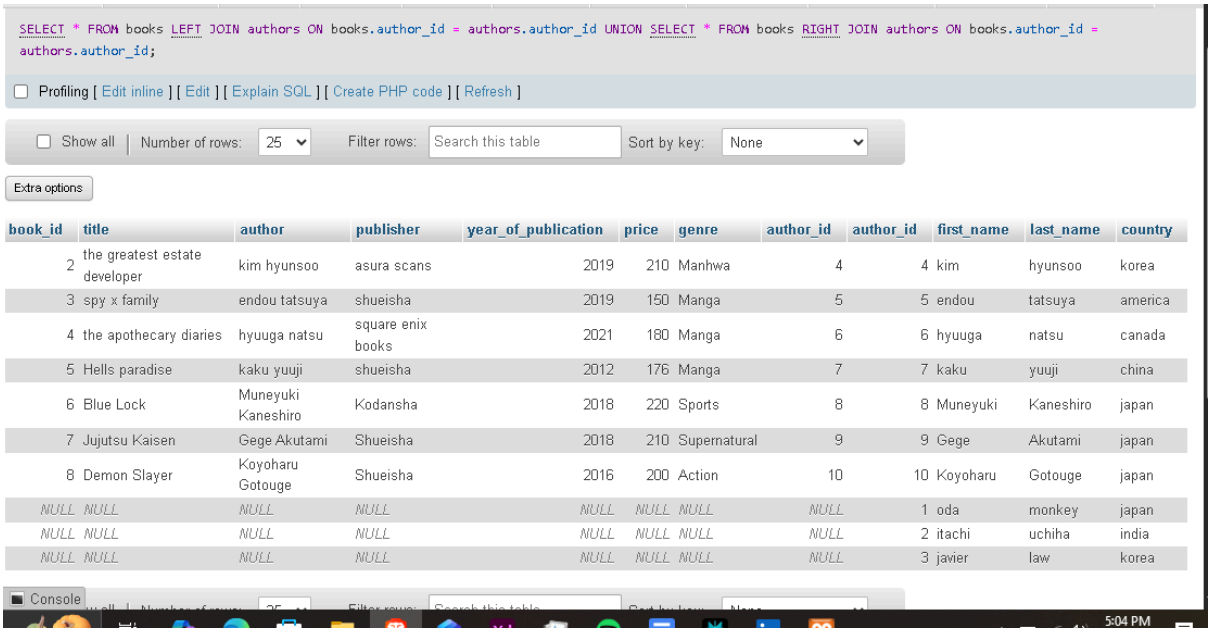
```
SELECT
    books.title,
    authors.first_name,
    authors.last_name
FROM
    books
INNER JOIN
    authors
ON
    books.author_id = authors.author_id;
```



title	first_name	last_name
the greatest estate developer	kim	hyunsoo
spy x family	endou	tatsuya
the apothecary diaries	hyuuga	natsu
Hells paradise	kaku	yuuji
Blue Lock	Muneyuki	Kaneshiro
Jujutsu Kaisen	Gege	Akutami
Demon Slayer	Koyoharu	Gotouge

- Lab 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.

SELECT * FROM books LEFT JOIN authors ON
books.author_id = authors.author_id
UNION
SELECT * FROM books RIGHT JOIN authors ON
books.author_id = authors.author_id



```
SELECT * FROM books LEFT JOIN authors ON books.author_id = authors.author_id UNION SELECT * FROM books RIGHT JOIN authors ON books.author_id =
authors.author_id;
```

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

book_id	title	author	publisher	year_of_publication	price	genre	author_id	author_id	first_name	last_name	country
2	the greatest estate developer	kim hyunsoo	asura scans	2019	210	Manhwa	4	4	kim	hyunsoo	korea
3	spy x family	endou tatsuya	shueisha	2019	150	Manga	5	5	endou	tatsuya	america
4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga	6	6	hyuuga	natsu	canada
5	Hells paradise	kaku yuuji	shueisha	2012	176	Manga	7	7	kaku	yuuji	china
6	Blue Lock	Muneyuki Kaneshiro	Kodansha	2018	220	Sports	8	8	Muneyuki	Kaneshiro	japan
7	Jujutsu Kaisen	Gege Akutami	Shueisha	2018	210	Supernatural	9	9	Gege	Akutami	japan
8	Demon Slayer	Koyoharu Gotouge	Shueisha	2016	200	Action	10	10	Koyoharu	Gotouge	japan
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	oda	monkey	japan
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2	itachi	uchiha	india
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	3	javier	law	korea

Console

5:04 PM

SQL Group By

- Lab 3: Group books by genre and display the total number of books in each genre.

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

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

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

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

Extra options

genre	total_books
Action	1
Manga	3
Manhwa	1
Sports	1
Supernatural	1

- Lab 4: Group members by the year they joined and find the number of members who joined each year.

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

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

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

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

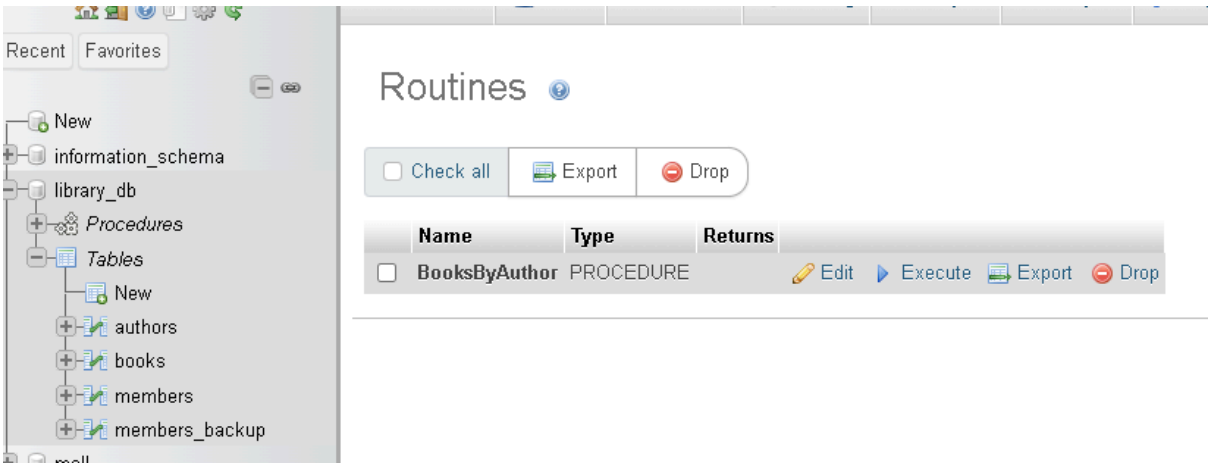
Extra options

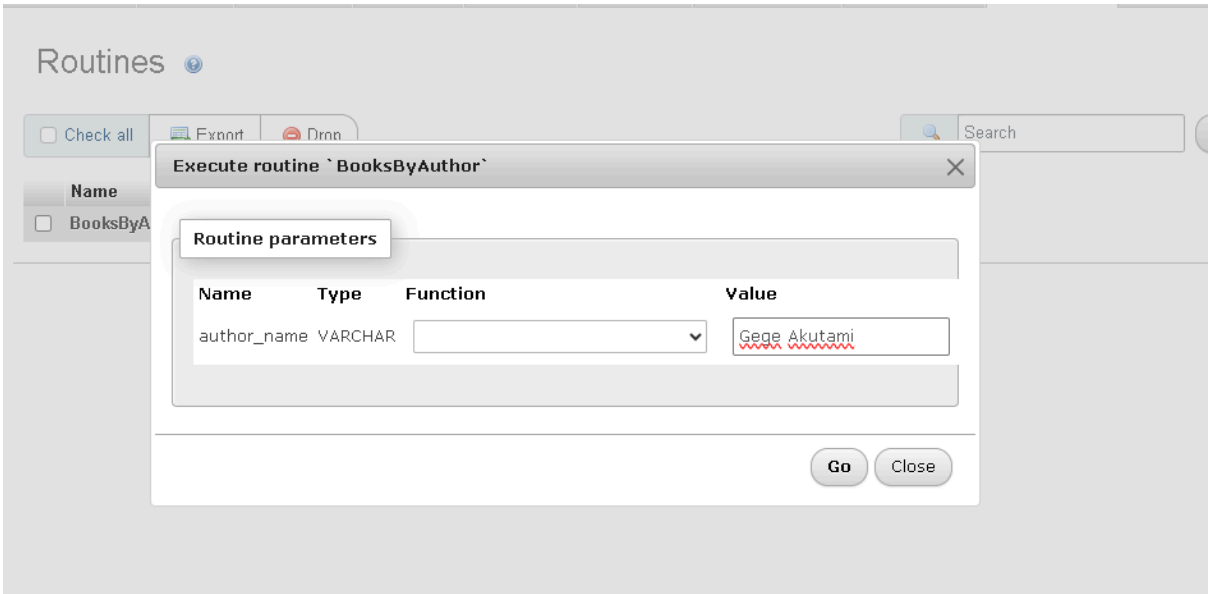
join_year	total_members
2022	1
2023	1
2024	1

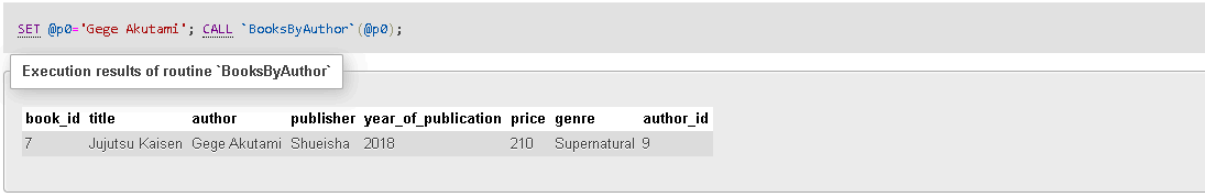
SQL Stored Procedure

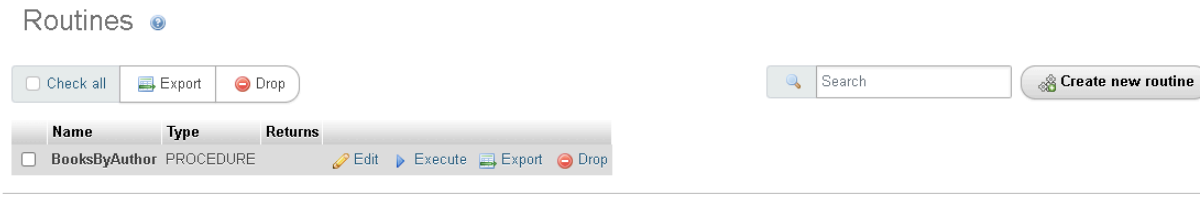
- Lab 3: Write a stored procedure to retrieve all books by a particular author.

```
DELIMITER //

CREATE PROCEDURE BooksByAuthor(IN author_name
    VARCHAR(100))
BEGIN
    SELECT *
    FROM books
    WHERE author = author_name;
END //

DELIMITER ;
```



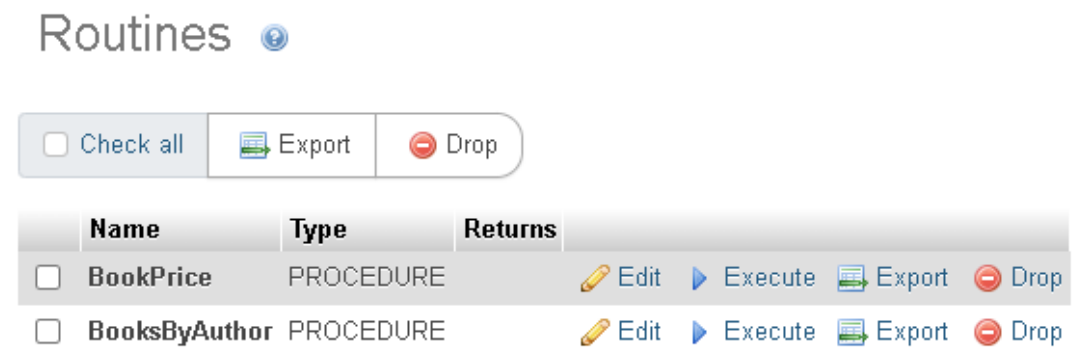

Routines
Check all
Export
Drop
Search
Name
BooksByA
Execute routine `BooksByAuthor`
Routine parameters
Name
Type
Function
Value
author_name
VARCHAR
Gege Akutami
Go
Close


SET @p0='Gege Akutami'; CALL `BooksByAuthor`(@p0);
Execution results of routine `BooksByAuthor`
book_id title author publisher year_of_publication price genre author_id
7 Jujutsu Kaisen Gege Akutami Shueisha 2018 210 Supernatural 9


Routines
Check all
Export
Drop
Search
Create new routine
Name
Type
Returns
BooksByAuthor
PROCEDURE
Edit
Execute
Export
Drop

• Lab 4: Write a stored procedure that takes book_id as an argument and returns the price of the book.

```
DELIMITER //

CREATE PROCEDURE BookPrice(IN id INT)
BEGIN
    SELECT price
    FROM books
    WHERE book_id = id;
END //

DELIMITER ;
```



Execute routine `BookPrice`

Routine parameters

Name	Type	Function	Value
id	INT		3

Go Close

Type Returns

```
SET @p0='3'; CALL `BookPrice`(@p0);
```

Execution results of routine `BookPrice`

price
150

Routines

Check all Export Drop

	Name	Type	Returns	
☐	BookPrice	PROCEDURE		Edit Execute Export Drop
☐	BooksByAuthor	PROCEDURE		Edit Execute Export Drop

SQL View

- Lab 3: Create a view to show only the title, author, and price of books from the books table.

```
CREATE VIEW books_summary AS
SELECT title, author, price
FROM books;
```

```
SELECT * FROM books_summary;
```

```
SELECT * FROM `books_summary`
```

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

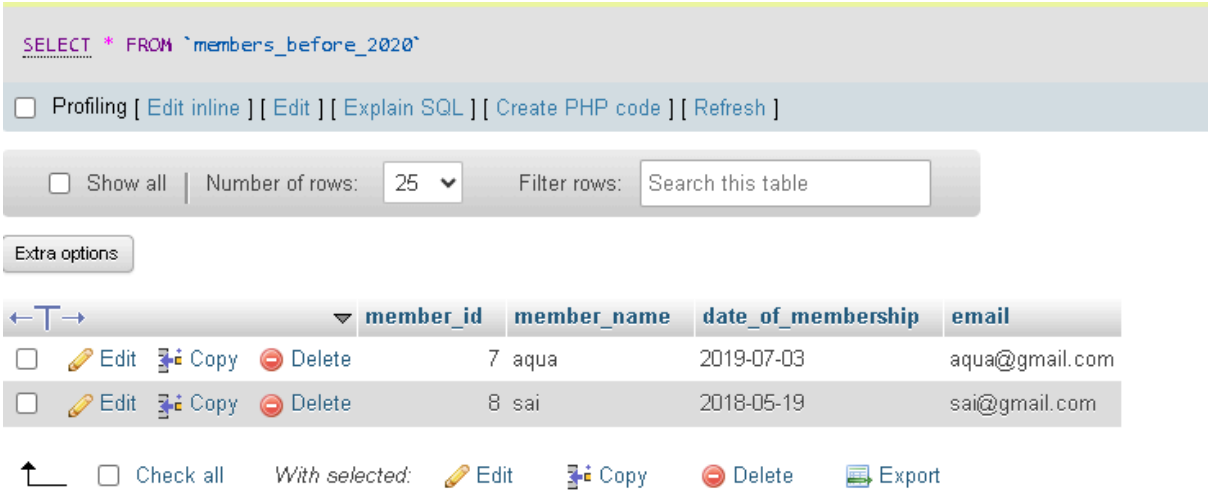
Show all | Number of rows: 25 Filter rows: Search this table

Extra options

←T→			title	author	price
Edit	Copy	Delete	the greatest estate developer	kim hyunsoo	210
Edit	Copy	Delete	spy x family	endou tatsuya	150
Edit	Copy	Delete	the apothecary diaries	hyuuga natsu	180
Edit	Copy	Delete	Hells paradise	kaku yuuji	176
Edit	Copy	Delete	Blue Lock	Muneyuki Kaneshiro	220
Edit	Copy	Delete	Jujutsu Kaisen	Gege Akutami	210
Edit	Copy	Delete	Demon Slayer	Koyoharu Gotouge	200

• Lab 4: Create a view to display members who joined before 2020.

```
CREATE VIEW members_before_2020 AS
SELECT *
FROM members
WHERE YEAR(date_of_membership) < 2020;
```



SQL Trigger

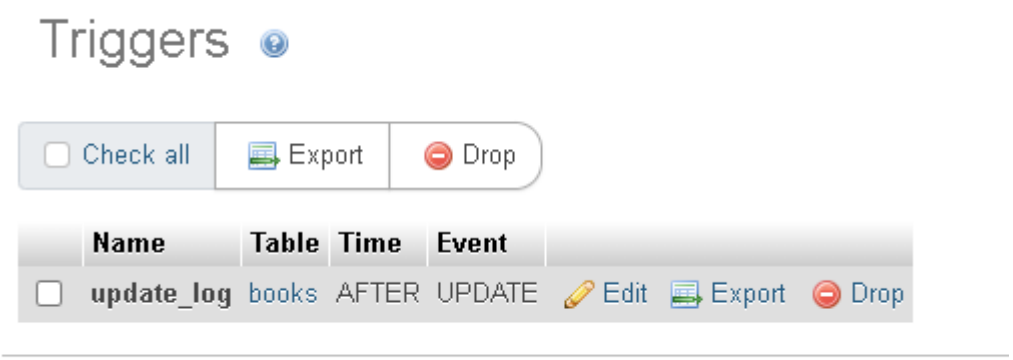
• Lab 3: Create a trigger to automatically update the last_modified timestamp of the books table whenever a record is updated.

```
CREATE TABLE book_update_log
(
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    old_title VARCHAR(200),
    new_title VARCHAR(200),
    old_price INT,
    new_price INT,
    update_time TIMESTAMP DEFAULT
CURRENT_TIMESTAMP

);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**log_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**book_id**	int(11)			Yes	*NULL*			Change Drop More
3	**old_title**	varchar(200)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
4	**new_title**	varchar(200)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
5	**old_price**	int(11)			Yes	*NULL*			Change Drop More
6	**new_price**	int(11)			Yes	*NULL*			Change Drop More
7	**update_time**	timestamp			No	current_timestamp()			Change Drop More

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

```
DELIMITER //

CREATE TRIGGER update_log
AFTER UPDATE ON books
FOR EACH ROW
BEGIN
   INSERT INTO update_log (book_id, old_title, new_title,
old_price, new_price)
   VALUES
   (
      OLD.book_id,
      OLD.title,
      NEW.title,
      OLD.price,
      NEW.price
   );
END //

DELIMITER ;
```



Triggers

Check all | Export | Drop

	Name	Time	Event	
☐	update_log	AFTER	UPDATE	Edit Export Drop

UPDATE books SET price = 200 WHERE books.book_id = 2;

```
SELECT * FROM `update_log`
```

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

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

Extra options

←T→	log_id	book_id	old_title	new_title	old_price	new_price	update_time
Edit Copy Delete	1	2	the greatest estate developer	the greatest estate developer	210	200	2025-10-26 19:44:30

Check all *With selected:* Edit Copy Delete Export

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

Query results operations

Print Copy to clipboard Export Display chart Create view

- Lab 4: Create a trigger that inserts a log entry into a log_changes table whenever a DELETE operation is performed on the books table.

```
CREATE TABLE delete_log
(
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    title VARCHAR(200),
    author VARCHAR(100),
    deleted_at TIMESTAMP DEFAULT
    CURRENT_TIMESTAMP
);
```

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**log_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**book_id**	int(11)			Yes	*NULL*			Change Drop More
3	**title**	varchar(200)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
4	**author**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
5	**deleted_at**	timestamp			No	current_timestamp()			Change Drop More

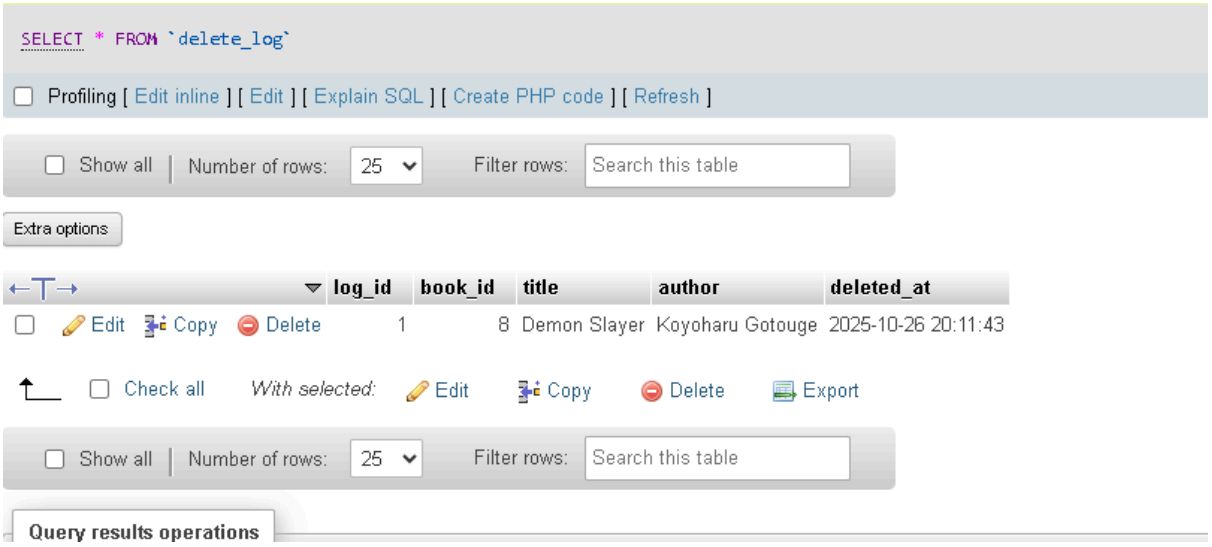
Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

```
DELIMITER //

CREATE TRIGGER book_delete_tri
AFTER DELETE ON books
FOR EACH ROW
BEGIN
    INSERT INTO delete_log (book_id, title, author)
    VALUES (OLD.book_id, OLD.title, OLD.author);
END //
```

DELIMITER ;

—----------

DELETE FROM books WHERE books.book_id = 8

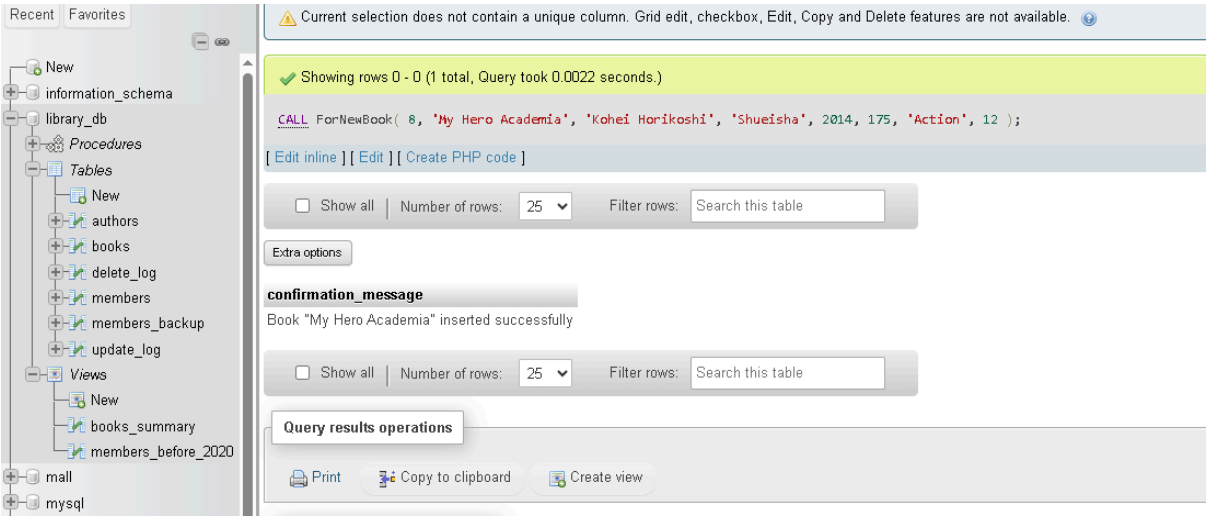


Introduction to PL/SQL

- Lab 3: Write a PL/SQL block to insert a new book into the books table and display a confirmation message.

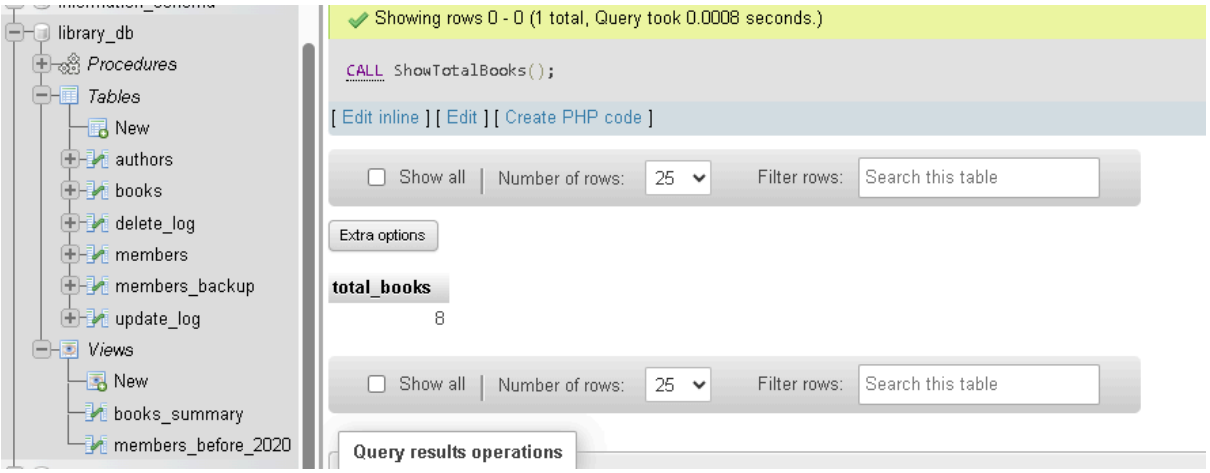
```
DELIMITER //

CREATE PROCEDURE ForNewBook
(
    IN n_book_id int,
    IN n_title VARCHAR(200),
    IN n_author VARCHAR(100),
    IN n_publisher VARCHAR(100),
    IN n_year int,
    IN n_price int,
    IN n_genre VARCHAR(100),
    IN n_author_id int
)
BEGIN
    INSERT INTO books (book_id,title, author, publisher,
    year_of_publication, price, genre, author_id)
    VALUES (n_book_id,n_title, n_author, n_publisher,
    n_year, n_price, n_genre, n_author_id);

    SELECT CONCAT('Book ''', n_title, ''' inserted
    successfully') AS confirmation_message;
END //
```

```
DELIMITER ;
—----------------------------------------------
CALL ForNewBook
(
    8
    'My Hero Academia',
    'Kohei Horikoshi',
    'Shueisha',
    2014,
    175,
    'Action',
    12
);
```



• Lab 4: Write a PL/SQL block to display the total number of books in the books table.

```
DELIMITER //

CREATE PROCEDURE ShowTotalBooks()
BEGIN
    SELECT COUNT(*) AS total_books FROM books;
END //

DELIMITER ;
```



PL/SQL Syntax

• Lab 3: Write a PL/SQL block to declare variables for book_id and price, assign values, and display the results.

```
DELIMITER //

CREATE PROCEDURE Book_by_IdAndPrice
(
    IN id INT,
    IN g_price INT
)
BEGIN
    SELECT id AS book_id, g_price AS price;
END //

DELIMITER ;
```

• Lab 4: Write a PL/SQL block using constants and perform arithmetic operations on book prices.

```
DELIMITER //

CREATE PROCEDURE Calc_B_Prices()
BEGIN
    DECLARE price1 INT DEFAULT 210;
    DECLARE price2 INT DEFAULT 180;
    DECLARE price3 INT DEFAULT 170;
    DECLARE sum_prices INT;
    DECLARE avg_price int;

    SET sum_prices = price1 + price2 + price3;
    SET avg_price = sum_prices / 3;

    SELECT
        sum_prices AS total_price,
        avg_price AS average_price;

END //

DELIMITER ;
```

```
call Calc_B_Prices();
```

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

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

Extra options

total_price	average_price
560	187

PL/SQL Control Structures

- Lab 3: Write a PL/SQL block using IF-THEN-ELSE to check if a book's price is above $200 and print a message accordingly.

```
DELIMITER //

CREATE PROCEDURE CheckBookPriceById(IN id INT)
BEGIN
    DECLARE t_price INT;

    SELECT price INTO t_price FROM books WHERE
    book_id = id;

    IF t_price > 200 THEN
        SELECT CONCAT('the book price is above 200
    (price: ', t_price, ')') AS message;
    ELSE
        SELECT CONCAT('the book price is 200 or below
    (price: ', t_price, ')') AS message;
    END IF;
END //

DELIMITER ;
```

```
call CheckBookPriceById(3);
```

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

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

Extra options

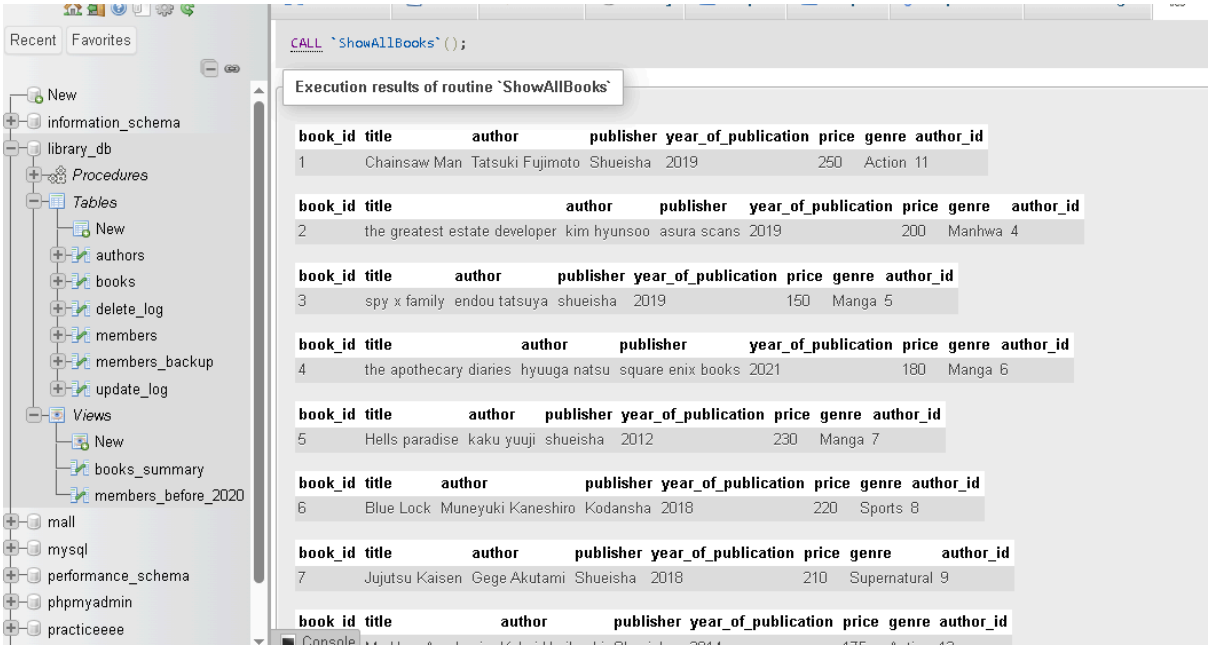
message
the book price is 200 or below (price: 150)

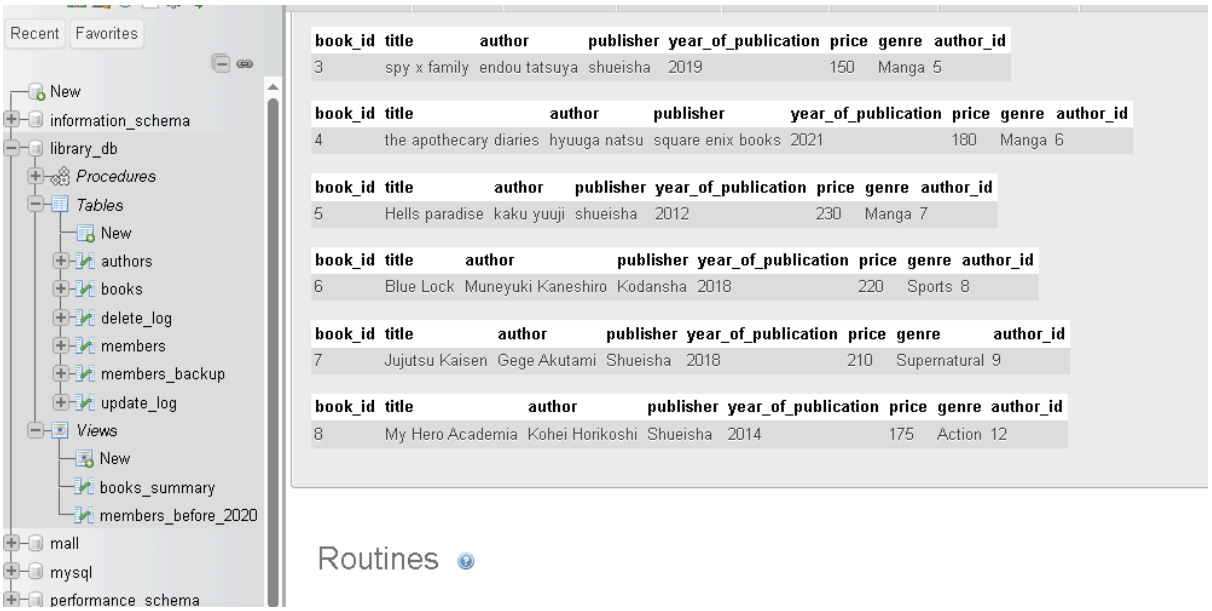
Show all | Number of rows: 25 Filter rows: Search this table

• Lab 4: Use a FOR LOOP in PL/SQL to display the details of all books one by one.

```
DELIMITER //

CREATE PROCEDURE ShowAllBooks()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE g_book_id INT;
    DECLARE g_title VARCHAR(200);
    DECLARE g_author VARCHAR(100);
    DECLARE g_publisher VARCHAR(100);
    DECLARE g_year INT;
    DECLARE g_price INT;
    DECLARE g_genre VARCHAR(100);
    DECLARE g_author_id INT;
    DECLARE cur CURSOR FOR
        SELECT book_id, title, author, publisher,
    year_of_publication, price, genre, author_id
        FROM books;
    DECLARE CONTINUE HANDLER FOR NOT
    FOUND SET done = TRUE;

    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO g_book_id, g_title, g_author,
    g_publisher, g_year, g_price, g_genre, g_author_id;
```

```
        IF done THEN
            LEAVE read_loop;
        END IF;

        SELECT
            g_book_id AS book_id,
            g_title AS title,
            g_author AS author,
            g_publisher AS publisher,
            g_year AS year_of_publication,
            g_price AS price,
            g_genre AS genre,
            g_author_id AS author_id;
    END LOOP;

    CLOSE cur;
END //

DELIMITER ;


—----------------------------------------------
CALL `ShowAllBooks`();
```


Recent
Favorites
New
information_schema
library_db
Procedures
Tables
New
authors
books
delete_log
members
members_backup
update_log
Views
New
books_summary
members_before_2020
mall
mysql
performance_schema
phpmyadmin
practiceeee
CALL `ShowAllBooks`();
Execution results of routine `ShowAllBooks`
book_id title author publisher year_of_publication price genre author_id
1 Chainsaw Man Tatsuki Fujimoto Shueisha 2019 250 Action 11
book_id title author publisher year_of_publication price genre author_id
2 the greatest estate developer kim hyunsoo asura scans 2019 200 Manhwa 4
book_id title author publisher year_of_publication price genre author_id
3 spy x family endou tatsuya shueisha 2019 150 Manga 5
book_id title author publisher year_of_publication price genre author_id
4 the apothecary diaries hyuuga natsu square enix books 2021 180 Manga 6
book_id title author publisher year_of_publication price genre author_id
5 Hells paradise kaku yuuji shueisha 2012 230 Manga 7
book_id title author publisher year_of_publication price genre author_id
6 Blue Lock Muneyuki Kaneshiro Kodansha 2018 220 Sports 8
book_id title author publisher year_of_publication price genre author_id
7 Jujutsu Kaisen Gege Akutami Shueisha 2018 210 Supernatural 9
book_id title author publisher year_of_publication price genre author_id

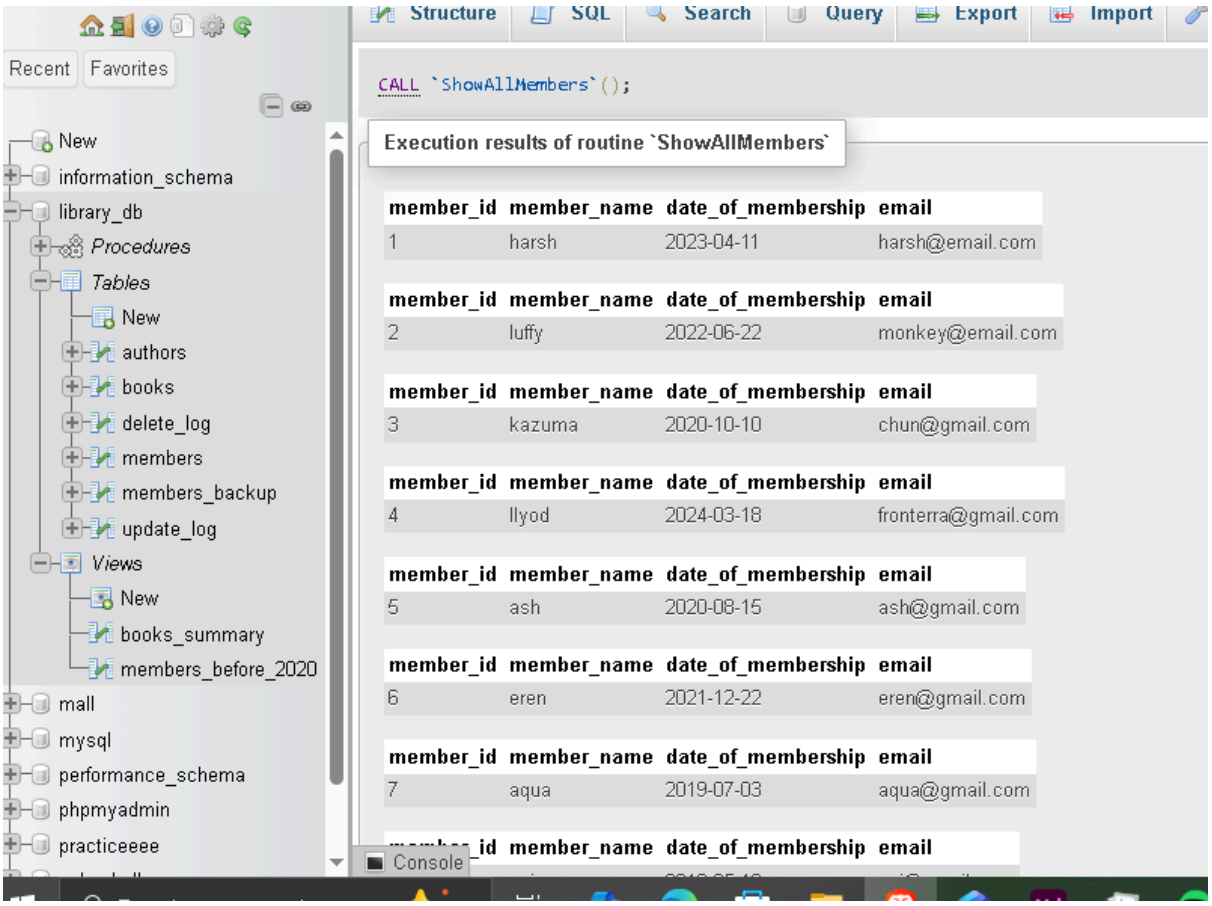

Recent
Favorites
New
information_schema
library_db
Procedures
Tables
New
authors
books
delete_log
members
members_backup
update_log
Views
New
books_summary
members_before_2020
mall
mysql
performance_schema
book_id title author publisher year_of_publication price genre author_id
3 spy x family endou tatsuya shueisha 2019 150 Manga 5
book_id title author publisher year_of_publication price genre author_id
4 the apothecary diaries hyuuga natsu square enix books 2021 180 Manga 6
book_id title author publisher year_of_publication price genre author_id
5 Hells paradise kaku yuuji shueisha 2012 230 Manga 7
book_id title author publisher year_of_publication price genre author_id
6 Blue Lock Muneyuki Kaneshiro Kodansha 2018 220 Sports 8
book_id title author publisher year_of_publication price genre author_id
7 Jujutsu Kaisen Gege Akutami Shueisha 2018 210 Supernatural 9
book_id title author publisher year_of_publication price genre author_id
8 My Hero Academia Kohei Horikoshi Shueisha 2014 175 Action 12
Routines

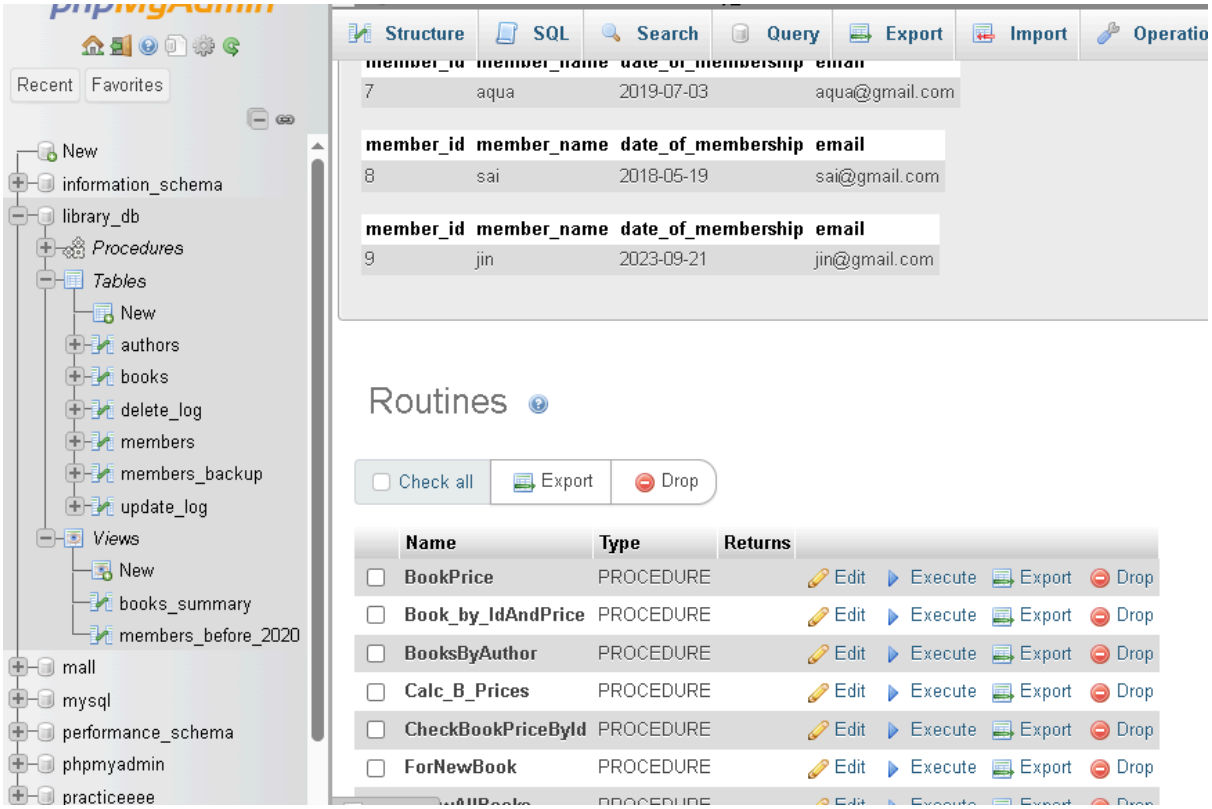
SQL Cursors

• Lab 3: Write a PL/SQL block using an explicit cursor to fetch and display all records from the members table.

```
DELIMITER //

CREATE PROCEDURE ShowAllMembers()
BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE g_member_id INT;
   DECLARE g_member_name VARCHAR(200);
   DECLARE g_date_of_membership DATE;
   DECLARE g_email VARCHAR(200);

   DECLARE cur CURSOR FOR
      SELECT member_id, member_name,
date_of_membership, email FROM members;

   DECLARE CONTINUE HANDLER FOR NOT
FOUND SET done = TRUE;

   OPEN cur;

   member_loop: LOOP
      FETCH cur INTO g_member_id, g_member_name,
g_date_of_membership, g_email;
      IF done THEN
```

```
            LEAVE member_loop;
        END IF;
        SELECT
            g_member_id AS member_id,
            g_member_name AS member_name,
            g_date_of_membership AS date_of_membership,
            g_email AS email;
    END LOOP;

    CLOSE cur;
END //

DELIMITER ;
```


Structure
SQL
Search
Query
Export
Import
Recent
Favorites
New
information_schema
library_db
Procedures
Tables
New
authors
books
delete_log
members
members_backup
update_log
Views
New
books_summary
members_before_2020
mall
mysql
performance_schema
phpmyadmin
practiceeee
CALL `ShowAllMembers`();
Execution results of routine `ShowAllMembers`
member_id member_name date_of_membership email
1 harsh 2023-04-11 harsh@email.com
member_id member_name date_of_membership email
2 luffy 2022-06-22 monkey@email.com
member_id member_name date_of_membership email
3 kazuma 2020-10-10 chun@gmail.com
member_id member_name date_of_membership email
4 llyod 2024-03-18 fronterra@gmail.com
member_id member_name date_of_membership email
5 ash 2020-08-15 ash@gmail.com
member_id member_name date_of_membership email
6 eren 2021-12-22 eren@gmail.com
member_id member_name date_of_membership email
7 aqua 2019-07-03 aqua@gmail.com
Console


Structure
SQL
Search
Query
Export
Import
Operatio
Recent
Favorites
7 aqua 2019-07-03 aqua@gmail.com
member_id member_name date_of_membership email
8 sai 2018-05-19 sai@gmail.com
member_id member_name date_of_membership email
9 jin 2023-09-21 jin@gmail.com
Routines
Check all
Export
Drop
Name Type Returns
BookPrice PROCEDURE Edit Execute Export Drop
Book_by_IdAndPrice PROCEDURE Edit Execute Export Drop
BooksByAuthor PROCEDURE Edit Execute Export Drop
Calc_B_Prices PROCEDURE Edit Execute Export Drop
CheckBookPriceById PROCEDURE Edit Execute Export Drop
ForNewBook PROCEDURE Edit Execute Export Drop

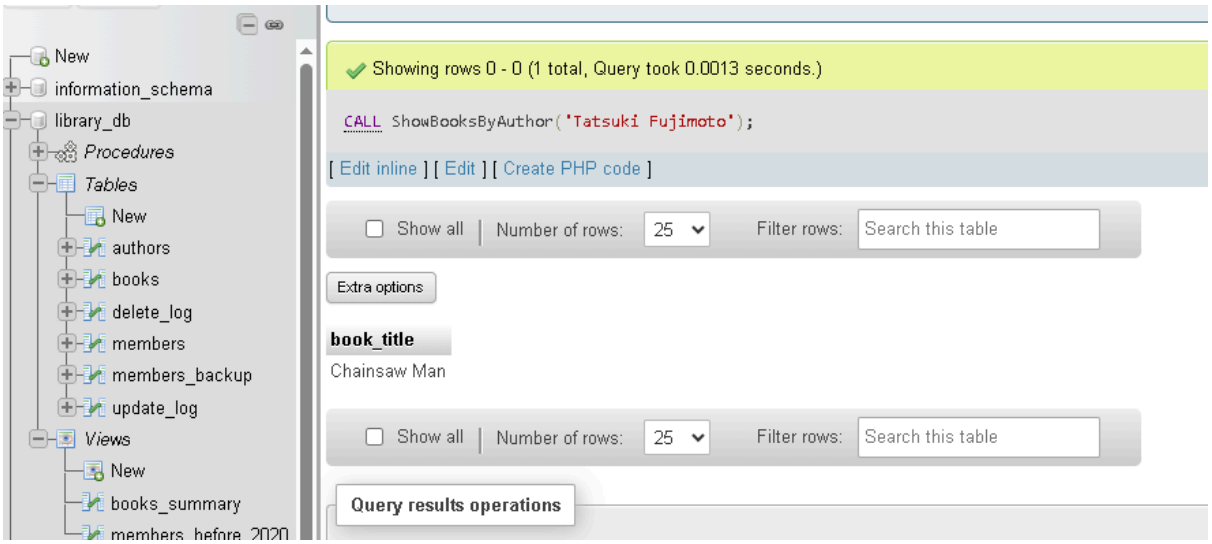
• Lab 4: Create a cursor to retrieve books by a particular author and display their titles.

```
DELIMITER //

CREATE PROCEDURE ShowBooksByAuthor(IN
p_author VARCHAR(100))
BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE g_title VARCHAR(100);
   DECLARE cur CURSOR FOR
      SELECT title FROM books WHERE author =
g_author;
   DECLARE CONTINUE HANDLER FOR NOT
FOUND SET done = TRUE;

   OPEN cur;

   books_loop: LOOP
      FETCH cur INTO g_title;
      IF done THEN
         LEAVE books_loop;
      END IF;
      SELECT g_title AS book_title;
   END LOOP;

   CLOSE cur;
END //
```

DELIMITER ;

—----------

call ShowBooksByAuthor('Tatsuki Fujimoto');



Rollback and Commit Savepoint

- Lab 3: Perform a transaction that includes inserting a new member, setting a SAVEPOINT, and rolling back to the savepoint after making updates.

```
START TRANSACTION;

INSERT INTO members (member_id, member_name,
    date_of_membership, email)
VALUES (10, 'zoro', '2019-10-27', 'zoro@email.com');

SAVEPOINT after_insert;

UPDATE members
SET member_name = 'edward'
WHERE member_id = 10;

ROLLBACK TO after_insert;

COMMIT;
```

←T→				member_id	member_name	date_of_membership	email
☐	Edit	Copy	Delete	1	harsh	2023-04-11	harsh@email.com
☐	Edit	Copy	Delete	2	luffy	2022-06-22	monkey@email.com
☐	Edit	Copy	Delete	3	kazuma	2020-10-10	chun@gmail.com
☐	Edit	Copy	Delete	4	llyod	2024-03-18	fronterra@gmail.com
☐	Edit	Copy	Delete	5	ash	2020-08-15	ash@gmail.com
☐	Edit	Copy	Delete	6	eren	2021-12-22	eren@gmail.com
☐	Edit	Copy	Delete	7	aqua	2019-07-03	aqua@gmail.com
☐	Edit	Copy	Delete	8	sai	2018-05-19	sai@gmail.com
☐	Edit	Copy	Delete	9	jin	2023-09-21	jin@gmail.com
☐	Edit	Copy	Delete	10	zoro	2019-10-27	zoro@email.com

- Lab 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.

```
START TRANSACTION;

INSERT INTO books (book_id, title, author, publisher,
    year_of_publication, price, genre, author_id)
VALUES (9, 'Vagabond', 'Takehiko Inoue', 'Kodansha',
    1998, 275, 'Historical', 13),
      (10, 'Noragami', 'Adachitoka', 'Kodansha', 2010, 210,
    'Fantasy', 14);

COMMIT;

START TRANSACTION;

SAVEPOINT before_price_update;

UPDATE books SET price = 100 WHERE book_id = 9;
UPDATE books SET price = 100 WHERE book_id = 10;

ROLLBACK TO before_price_update;

COMMIT;
```

Before

book_id	title	author	publisher	year_of_publication	price	genre	author_id
1	Chainsaw Man	Tatsuki Fujimoto	Shueisha	2019	250	Action	11
2	the greatest estate developer	kim hyunsoo	asura scans	2019	200	Manhwa	4
3	spy x family	endou tatsuya	shueisha	2019	150	Manga	5
4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga	6
5	Hells paradise	kaku yuuji	shueisha	2012	230	Manga	7
6	Blue Lock	Muneyuki Kaneshiro	Kodansha	2018	220	Sports	8
7	Jujutsu Kaisen	Gege Akutami	Shueisha	2018	210	Supernatural	9
8	My Hero Academia	Kohei Horikoshi	Shueisha	2014	175	Action	12

After

book_id	title	author	publisher	year_of_publication	price	genre	author_id
1	Chainsaw Man	Tatsuki Fujimoto	Shueisha	2019	250	Action	11
2	the greatest estate developer	kim hyunsoo	asura scans	2019	200	Manhwa	4
3	spy x family	endou tatsuya	shueisha	2019	150	Manga	5
4	the apothecary diaries	hyuuga natsu	square enix books	2021	180	Manga	6
5	Hells paradise	kaku yuuji	shueisha	2012	230	Manga	7
6	Blue Lock	Muneyuki Kaneshiro	Kodansha	2018	220	Sports	8
7	Jujutsu Kaisen	Gege Akutami	Shueisha	2018	210	Supernatural	9
8	My Hero Academia	Kohei Horikoshi	Shueisha	2014	175	Action	12
9	Vagabond	Takehiko Inoue	Kodansha	1998	275	Historical	13
10	Noragami	Adachitoka	Kodansha	2010	210	Fantasy	14

Check all *With selected:* Edit Copy Delete Export