

## Advance SQL Tasks:

### Task 1:

You are tasked with managing a school database that contains information about students, teachers, classes, and enrollments. Create the tables in PHPMyAdmin then insert the sample data.

#### Tables and Structure

1. **Students:** StudentID, FirstName, LastName, BirthDate, Email
2. **Teachers:** TeacherID, FirstName, LastName, Email
3. **Classes:** ClassID, ClassName, TeacherID
4. **Enrollments:** EnrollmentID, StudentID, ClassID, EnrollmentDate

#### SQL Tasks

1. **SQL INSERT INTO Statement**
  - **Task:** Add a new student to the Students table.
2. **SQL SELECT Statement**
  - **Task:** Retrieve the list of all classes along with the names of their teachers.
3. **SQL UPDATE Statement**
  - **Task:** Update the email address of the teacher with TeacherID = 2.
4. **SQL DELETE Statement**
  - **Task:** Remove the enrollment record for a student with StudentID = 3 from the Enrollments table.

### Task 2:

You are managing a library database that contains information about books, authors, and members.

The database includes the following tables:

1. `Books` (BookID, Title, AuthorID, Genre, Price, PublicationDate)
2. `Authors` (AuthorID, Name, Country)
3. `Members` (MemberID, FirstName, LastName, Email, JoinDate)
4. `BorrowedBooks` (BorrowID, BookID, MemberID, BorrowDate, ReturnDate)

Please give me the output for the following SQL statements:

Note: For some tasks fill the data to ensure that the tasks run as SQL query / phpMyAdmin

1. SQL INSERT INTO Statement

- Task: Add a new book to the `Books` table, With values ('The Great Gatsby', 1, 'Fiction', 15.99, '1925-04-10')

## 2. SQL SELECT Statement

- Task: Retrieve all the books written by the author with AuthorID = 1.

## 3. SQL UPDATE Statement

- Task: Update the price of the book with BookID = 2 to 20.99.

## 4. SQL DELETE Statement

- Task: Remove the book with BookID = 3 from the database.

## 5. SQL WHERE Clause

- Task: Find all books in the 'Science Fiction' genre.

## 6. SQL AND, OR and NOT Operators

- Task: Retrieve books that are either in the 'Fiction' genre and priced below 20, or not written by the author with AuthorID = 2.

## 7. SQL ORDER BY Keyword

- Task: Get all books ordered by their publication date in descending order.

## 8. SQL MIN() and MAX() Functions

- Task: Find the minimum and maximum price of books in the library.

## 9. SQL COUNT(), AVG() and SUM() Functions

- Task: Get the total number of books, the average price, and the total price of all books in the library.

## 10. SQL LIKE Operator

- Task: Find all books with a title that starts with 'The'.

## 11. SQL GROUP BY Statement

- Task: Get the count of books for each genre.

## 12. SQL INNER JOIN Keyword

- Task: Retrieve a list of books along with the names of their authors.

## Task3:

You are managing the member database of the library. You need to retrieve specific information about the members who have borrowed books from the library.

## Members Table

MemberID	FirstName	LastName	Email	JoinDate
1	John	Doe	john.doe@example.com	2022-12-15
2	Jane	Smith	jane.smith@library.com	2023-02-10
3	Emily	Johnson	emily.j@example.com	2023-03-05
4	Michael	Brown	michael.b@library.com	2023-04-12
5	Sarah	Davis	sarah.d@example.com	2023-06-21

## BorrowedBooks Table

BorrowID	BookID	MemberID	BorrowDate	ReturnDate
1	101	2	2023-02-11	2023-02-18
2	102	2	2023-02-20	2023-02-27
3	103	2	2023-03-05	2023-03-12
4	104	2	2023-03-15	2023-03-22
5	105	3	2023-03-06	2023-03-13
6	106	3	2023-03-20	2023-03-27
7	107	4	2023-04-13	2023-04-20
8	108	4	2023-04-25	2023-05-02
9	109	4	2023-05-10	2023-05-17
10	110	4	2023-05-20	2023-05-27
11	111	4	2023-06-05	2023-06-12
12	112	5	2023-06-22	2023-06-29

Retrieve a list of members who joined after January 1, 2023, have borrowed more than 3 books, and sort them by their last name in ascending order. Exclude members from the list who have an email ending with '@example.com'.

## Task4:

Continuing with the library management system, we'll create additional tasks focusing on different SQL functionalities.

### 1. SQL MIN() and MAX() Functions

- **Task:** Find the minimum and maximum price of books borrowed by members who joined after January 1, 2023.

### 2. SQL COUNT(), AVG() and SUM() Functions

- **Task:** Get the total number of books borrowed, the average price of borrowed books, and the total price of borrowed books by members who joined after January 1, 2023.

### 3. SQL LIKE Operator

- **Task:** Find all members whose last name starts with 'J'.
4. **SQL GROUP BY Statement**
    - **Task:** Get the count of books borrowed by each member.
  5. **SQL INNER JOIN Keyword**
    - **Task:** Retrieve a list of members along with the titles of the books they have borrowed.

### Sample Data for Books Table

BookID	Title	AuthorID	Genre	Price	PublicationDate
101	Book A	1	Fiction	10.99	2022-01-01
102	Book B	2	Non-Fiction	12.99	2021-02-01
103	Book C	3	Science Fiction	8.99	2020-03-01
104	Book D	4	Mystery	15.99	2019-04-01
105	Book E	1	Romance	9.99	2018-05-01
106	Book F	2	Fiction	7.99	2023-06-01
107	Book G	3	Thriller	6.99	2022-07-01
108	Book H	4	Fantasy	11.99	2023-08-01
109	Book I	1	Fiction	14.99	2021-09-01
110	Book J	2	Non-Fiction	13.99	2020-10-01
111	Book K	3	Science Fiction	9.49	2022-11-01
112	Book L	4	Mystery	16.99	2019-12-01