

# Лабораторная работа №14

Тема: Продвинутые SQL-запросы и адаптация веб-интерфейса для управления базой данных (OnlineBookstore)

## 1. Цель работы

- Закрепить навыки составления SQL-запросов, включая множественные соединения ( JOIN ), подзапросы, агрегатные функции с группировкой ( GROUP BY , HAVING ) и сортировку ( ORDER BY ).
- Научиться формулировать запросы для извлечения конкретных данных из связанных таблиц.
- Развить навыки адаптации существующего PHP/MySQL кода для работы с новой схемой базы данных.
- Реализовать CRUD-интерфейсы для управления несколькими сущностями реляционной базы данных через веб-приложение.
- Усовершенствовать навыки работы с HTML-формами, включая обработку нескольких внешних ключей с помощью элементов `<select>`.

## 2. Теоретическое введение

- Типы JOIN:
  - INNER JOIN : Возвращает строки, когда есть совпадение в обеих таблицах.
  - LEFT JOIN (или LEFT OUTER JOIN ): Возвращает все строки из левой таблицы и совпадающие строки из правой таблицы. Если совпадения нет, для столбцов правой таблицы возвращается NULL .

- **RIGHT JOIN** (или **RIGHT OUTER JOIN**): Возвращает все строки из правой таблицы и совпадающие строки из левой таблицы. Если совпадения нет, для столбцов левой таблицы возвращается **NULL**.
- **Подзапросы**: Запрос, вложенный внутрь другого SQL-запроса (например, в **WHERE**, **SELECT** или **FROM**). Позволяют выполнять более сложные фильтрации и вычисления.
- **Агрегатные функции**: **COUNT()**, **SUM()**, **AVG()**, **MIN()**, **MAX()**. Часто используются с **GROUP BY**.
- **GROUP BY**: Группирует строки с одинаковыми значениями в указанных столбцах.
- **HAVING**: Фильтрует группы на основе заданного условия (применяется после **GROUP BY**).
- **CRUD, PHP/MySQL, Структура веб-приложения, HTML-формы**: ... (как в предыдущей лаб. работе)

### 3. Часть 1: Продвинутые SQL-запросы

#### 3.1. Схема и данные базы данных **OnlineBookstore**

Вам предоставляется готовый SQL-скрипт для создания и наполнения базы данных **OnlineBookstore**. Вам не нужно его изменять. Эта база данных будет использоваться для написания запросов в п. 3.2 и для адаптации веб-приложения в Части 2.

SQL-скрипт для создания и наполнения БД:

```
-- Создание базы данных
CREATE DATABASE OnlineBookstore CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

-- ----- Таблица Авторы (Authors) -----
CREATE TABLE Authors (
```

```
author_id INT AUTO_INCREMENT PRIMARY KEY,  
first_name VARCHAR(50) NOT NULL,  
last_name VARCHAR(50) NOT NULL,  
country VARCHAR(50)  
);  
  
-- ----- Таблица Издательства (Publishers) -----  
CREATE TABLE Publishers (  
    publisher_id INT AUTO_INCREMENT PRIMARY KEY,  
    publisher_name VARCHAR(100) NOT NULL UNIQUE,  
    city VARCHAR(50)  
);  
  
-- ----- Таблица Категории (Categories) -----  
CREATE TABLE Categories (  
    category_id INT AUTO_INCREMENT PRIMARY KEY,  
    category_name VARCHAR(100) NOT NULL UNIQUE  
);  
  
-- ----- Таблица Книги (Books) -----  
CREATE TABLE Books (  
    book_id INT AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    author_id INT,  
    publisher_id INT,  
    category_id INT,  
    publication_year INT,  
    isbn VARCHAR(20) UNIQUE,
```

```
    price DECIMAL(10, 2) CHECK (price >= 0)
);

-- ----- Таблица Покупатели (Customers) -----
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    city VARCHAR(50)
);

-- ----- Таблица Заказы (Orders) -----
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATETIME NOT NULL,
    total_amount DECIMAL(12, 2),
    status VARCHAR(20) DEFAULT 'Pending',
);

-- ----- Таблица Элементы Заказа (Order_Items) -----
CREATE TABLE Order_Items (
    order_item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    book_id INT NOT NULL,
    quantity INT NOT NULL CHECK (quantity > 0),
    price_per_item DECIMAL(10, 2) NOT NULL
```

```
);
```

```
-- ----- Наполнение данными -----
```

```
-- Авторы
```

```
INSERT INTO Authors (author_id, first_name, last_name, country) VALUES
```

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

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

```
(3, 'Leo', 'Tolstoy', 'Russia'),
```

```
(4, 'Mark', 'Twain', 'USA'),
```

```
(5, 'Haruki', 'Murakami', 'Japan'),
```

```
(6, 'Gabriel', 'Garcia Marquez', 'Colombia'),
```

```
(7, 'Fyodor', 'Dostoevsky', 'Russia'),
```

```
(8, 'Agatha', 'Christie', 'UK'),
```

```
(9, 'J.K.', 'Rowling', 'UK'),
```

```
(10, 'Stephen', 'King', 'USA'),
```

```
(11, 'Мухтар', 'Ауэзов', 'Казахстан'),
```

```
(12, 'Олжас', 'Сулейменов', 'Казахстан'),
```

```
(13, 'Абай', 'Кунанбаев', 'Казахстан'); -- Добавим классика
```

```
-- Издательства
```

```
INSERT INTO Publishers (publisher_id, publisher_name, city) VALUES
```

```
(1, 'Penguin Books', 'London'),
```

```
(2, 'HarperCollins', 'New York'),
```

```
(3, 'Vintage Books', 'New York'),
```

```
(4, 'Scribner', 'New York'),
```

```
(5, 'Bloomsbury', 'London'),
```

```
(6, 'Doubleday', 'New York'),
```

```
(7, 'Фолиант', 'Астана'),  
(8, 'Атамұра', 'Алматы'),  
(9, 'Жазушы', 'Алматы');
```

-- Категории

```
INSERT INTO Categories (category_id, category_name) VALUES  
(1, 'Классическая литература'),  
(2, 'Современная проза'),  
(3, 'Детектив'),  
(4, 'Фэнтези'),  
(5, 'Научная фантастика'),  
(6, 'История'),  
(7, 'Поэзия'),  
(8, 'Казахстанская литература');
```

-- Книги

```
INSERT INTO Books (book_id, title, author_id, publisher_id, category_id, publication_year, isbn,  
price) VALUES  
(1, '1984', 1, 1, 5, 1949, '978-0451524935', 7500.00),  
(2, 'Pride and Prejudice', 2, 1, 1, 1813, '978-0141439518', 6000.00),  
(3, 'War and Peace', 3, 3, 1, 1869, '978-0307389907', 12000.00),  
(4, 'Anna Karenina', 3, 1, 1, 1877, '978-0143035008', 11000.00),  
(5, 'Adventures of Huckleberry Finn', 4, 2, 1, 1884, '978-0486280615', 5500.00),  
(6, 'Norwegian Wood', 5, 3, 2, 1987, '978-0375704024', 8000.00),  
(7, 'Kafka on the Shore', 5, 3, 2, 2002, '978-1400079278', 8500.00),  
(8, 'One Hundred Years of Solitude', 6, 2, 2, 1967, '978-0061120084', 9000.00),  
(9, 'Crime and Punishment', 7, 1, 1, 1866, '978-0486415871', 7000.00),  
(10, 'Murder on the Orient Express', 8, 2, 3, 1934, '978-0062073501', 6500.00),
```

```
(11, 'Harry Potter and the Sorcerer Stone', 9, 5, 4, 1997, '978-0590353427', 9500.00),
(12, 'The Shining', 10, 6, 5, 1977, '978-0385121675', 8800.00),
(13, 'It', 10, 4, 5, 1986, '978-1501142970', 10500.00),
(14, 'Путь Абая', 11, 8, 8, 1942, '978-6012940590', 15000.00),
(15, 'АЗ и Я', 12, 9, 8, 1975, '978-5604107388', 7500.00),
(16, 'Слова назидания (Қара сөздер)', 13, 7, 8, 1898, '978-6013380284', 5000.00),
(17, 'The Great Gatsby', NULL, 4, 1, 1925, '978-0743273565', 6200.00); -- Книга без автора в БД
```

-- Покупатели

```
INSERT INTO Customers (customer_id, first_name, last_name, email, city) VALUES
```

```
(1, 'Айгуль', 'Смагулова', 'a.smagulova@mail.kz', 'Атырау'),
(2, 'Бауржан', 'Ермеков', 'b.ermekov@gmail.com', 'Алматы'),
(3, 'Мадина', 'Сапарова', 'madina.s@inbox.ru', 'Астана'),
(4, 'Ержан', 'Калиев', 'e.kaliev@mail.kz', 'Актобе'),
(5, 'Дина', 'Ахметова', 'dina.a@gmail.com', 'Караганда'),
(6, 'Нуржан', 'Бекенов', 'n.bekenov@yahoo.com', 'Атырау'),
(7, 'Алия', 'Толегенова', 'aliya.t@mail.kz', 'Шымкент'),
(8, 'Серик', 'Ибраев', 'serik.ibraev@gmail.com', 'Астана'),
(9, 'Гульнара', 'Жумагалиева', 'g.zhumagalieva@inbox.ru', 'Алматы'),
(10, 'Арман', 'Нуртазин', 'arman.n@mail.kz', 'Атырау');
```

-- Заказы

```
INSERT INTO Orders (order_id, customer_id, order_date, status, total_amount) VALUES
```

```
(1, 1, '2025-04-01 10:30:00', 'Delivered', 14000.00),
(2, 2, '2025-04-05 14:00:00', 'Shipped', 8500.00),
(3, 1, '2025-04-10 09:15:00', 'Processing', 16300.00),
(4, 3, '2025-04-12 11:00:00', 'Pending', 5000.00),
(5, 5, '2025-04-15 16:45:00', 'Delivered', 7500.00),
```

```

(6, 10, '2025-04-20 12:00:00', 'Shipped', 15000.00),
(7, 2, '2025-04-22 18:00:00', 'Pending', 11000.00),
(8, 8, '2025-04-25 10:00:00', 'Processing', 9500.00);

-- Элементы Заказа
INSERT INTO Order_Items (order_item_id, order_id, book_id, quantity, price_per_item) VALUES
-- Заказ 1
(1, 1, 9, 1, 7000.00), (2, 1, 1, 1, 7000.00),
-- Заказ 2
(3, 2, 7, 1, 8500.00),
-- Заказ 3
(4, 3, 10, 1, 6500.00), (5, 3, 12, 1, 8800.00), (6, 3, 17, 1, 6200.00),
-- Заказ 4
(7, 4, 16, 1, 5000.00),
-- Заказ 5
(8, 5, 15, 1, 7500.00),
-- Заказ 6
(9, 6, 14, 1, 15000.00),
-- Заказ 7
(10, 7, 4, 1, 11000.00),
-- Заказ 8
(11, 8, 11, 1, 9500.00);

UPDATE Orders SET total_amount = (6500.00 + 8800.00 + 6200.00) WHERE order_id = 3;

```

### 3.2. Задания на SQL-запросы



Используя базу данных `OnlineBookstore`, напишите SQL-запросы для выполнения следующих задач. Внимательно читайте, какие столбцы и из каких таблиц нужно вывести.

## Базовые и JOIN запросы

1. Задача: Вывести названия (`title`) всех книг и фамилии (`last_name`) их авторов.
  - Таблицы: `Books`, `Authors`.
  - Выходные столбцы: `Books.title`, `Authors.last_name`.
2. Задача: Вывести названия (`title`) книг, изданных издательством 'Penguin Books'.
  - Таблицы: `Books`, `Publishers`.
  - Выходные столбцы: `Books.title`.
3. Задача: Вывести имена (`first_name`) и фамилии (`last_name`) покупателей из города 'Атырау'.
  - Таблицы: `Customers`.
  - Выходные столбцы: `Customers.first_name`, `Customers.last_name`.
4. Задача: Вывести названия книг (`title`) и названия их категорий (`category_name`).
  - Таблицы: `Books`, `Categories`.
  - Выходные столбцы: `Books.title`, `Categories.category_name`.
5. Задача: Вывести все книги (`Books.title`) и их авторов (`Authors.first_name`, `Authors.last_name`), включая книги, у которых автор не указан в таблице `Authors`.
  - Таблицы: `Books`, `Authors`.
  - Тип JOIN: `LEFT JOIN`.
  - Выходные столбцы: `Books.title`, `Authors.first_name`, `Authors.last_name`.
6. Задача: Вывести всех авторов (`Authors.first_name`, `Authors.last_name`) и названия (`Books.title`) их книг. Если у автора нет книг в базе, он все равно должен быть в списке, а название книги будет `NULL`.

- Таблицы: `Authors` , `Books` .
  - Тип JOIN: `LEFT JOIN` .
  - Выходные столбцы: `Authors.first_name` , `Authors.last_name` , `Books.title` .
7. Задача: Вывести название книги ( `Books.title` ), имя и фамилию автора ( `Authors.first_name` , `Authors.last_name` ) и название издательства ( `Publishers.publisher_name` ) для всех книг.
- Таблицы: `Books` , `Authors` , `Publishers` .
  - Выходные столбцы: `Books.title` , `Authors.first_name` , `Authors.last_name` , `Publishers.publisher_name` .

## Запросы с агрегацией, группировкой и сортировкой

8. Задача: Посчитать количество книг в каждой категории.
- Таблицы: `Books` , `Categories` .
  - Выходные столбцы: `Categories.category_name` , `COUNT(Books.book_id)` (назовите этот столбец `book_count` ).
9. Задача: Найти среднюю цену книг для каждого издательства.
- Таблицы: `Books` , `Publishers` .
  - Выходные столбцы: `Publishers.publisher_name` , `AVG(Books.price)` (назовите `average_price` ).
10. Задача: Вывести ID заказов ( `order_id` ) и общую стоимость ( `total_amount` ) заказов, сделанных после '2025-04-10', отсортированных по убыванию стоимости.
- Таблицы: `Orders` .
  - Выходные столбцы: `Orders.order_id` , `Orders.total_amount` .
11. Задача: Найти общую сумму продаж (сумма `total_amount` всех заказов) для каждого покупателя.
- Таблицы: `Orders` , `Customers` .

- Выходные столбцы: `Customers.first_name` , `Customers.last_name` , `SUM(Orders.total_amount)` (назовите `total_spent` ).

12. Задача: Найти авторов, у которых в базе более одной книги.

- Таблицы: `Books` , `Authors` .
- Выходные столбцы: `Authors.first_name` , `Authors.last_name` , `COUNT(Books.book_id)` (назовите `books_written` ).
- Ключевое слово: `HAVING` .

13. Задача: Показать ID заказа ( `order_id` ), название книги ( `title` ) и количество ( `quantity` ) для всех позиций во всех заказах.

- Таблицы: `Order_Items` , `Books` .
- Выходные столбцы: `Order_Items.order_id` , `Books.title` , `Order_Items.quantity` .

Запросы с подзапросами и более сложной логикой

14. Задача: Найти книги ( `title` , `price` ), цена которых выше средней цены всех книг в магазине.

- Таблицы: `Books` .
- Логика: Использовать подзапрос для вычисления средней цены `AVG(price)` .
- Выходные столбцы: `Books.title` , `Books.price` .
- Пример структуры: `SELECT title, price FROM Books WHERE price > (SELECT AVG(price) FROM Books);`

15. Задача: Вывести имена и фамилии покупателей ( `first_name` , `last_name` ), которые сделали хотя бы один заказ.

- Таблицы: `Customers` , `Orders` .
- Логика: Можно использовать `JOIN` или подзапрос с `EXISTS` или `IN` .

- Выходные столбцы: `Customers.first_name` , `Customers.last_name` .
- Пример с `EXISTS` : `SELECT first_name, last_name FROM Customers c WHERE EXISTS (SELECT 1 FROM Orders o WHERE o.customer_id = c.customer_id);`

16. Задача: Найти самую дорогую книгу в каждой категории.

- Таблицы: `Books` , `Categories` .
- Логика: Использовать `GROUP BY` по категории и `MAX(price)` , затем соединить результат с таблицей `Books` для получения названия книги, или использовать оконные функции (если поддерживается и известно) или коррелирующий подзапрос.
- Выходные столбцы: `Categories.category_name` , `Books.title` , `Books.price` (максимальная цена).

17. Задача: Вывести заказы ( `order_id` , `order_date` ), в которых есть книга '1984' ( `book_id` = 1).

- Таблицы: `Orders` , `Order_Items` .
- Логика: Использовать `JOIN` или подзапрос с `IN` / `EXISTS` .
- Выходные столбцы: `Orders.order_id` , `Orders.order_date` .

18. Задача: Вывести названия ( `title` ) книг, которые были заказаны покупателем с `email` = `'a.smagulova@mail.kz'`.

- Таблицы: `Books` , `Order_Items` , `Orders` , `Customers` .
- Логика: Потребуется соединить несколько таблиц.
- Выходные столбцы: `Books.title` .

19. Задача: Посчитать общее количество проданных экземпляров каждой книги.

- Таблицы: `Books` , `Order_Items` .
- Выходные столбцы: `Books.title` , `SUM(Order_Items.quantity)` (назовите `total_sold` ).

20. Задача: Найти покупателей ( `first_name` , `last_name` ), которые не сделали ни одного заказа.

- Таблицы: `Customers` , `Orders` .

- Логика: Использовать `LEFT JOIN` и проверку на `NULL` или подзапрос с `NOT IN / NOT EXISTS`.
- Выходные столбцы: `Customers.first_name`, `Customers.last_name`.

## 4. Часть 2: Адаптация веб-приложения для управления БД `OnlineBookstore`

### 4.1. Описание задания

В этой части вам предстоит снова адаптировать код веб-приложения (из репозитория `rawitjan/subd-kk` или аналогичного) для работы с новой базой данных `OnlineBookstore`.

Ваша задача — создать рабочие CRUD-интерфейсы как минимум для таблиц `Authors` и `Books`.

Реализация интерфейса для `Publishers` или `Categories` может быть хорошим дополнительным шагом.

### 4.2. База данных

Используйте базу данных `OnlineBookstore`, созданную и наполненную скриптом из Части 1 (п. 3.1).

Убедитесь, что эта база данных существует на вашем локальном MySQL-сервере.

### 4.3. Процесс адаптации веб-приложения

#### 1. Настройте проект:

- Создайте новую папку для проекта (например, `bookstore_manager`) в `htdocs`.
- Скопируйте структуру и файлы из шаблона (`includes/`, папки модулей, `index.php`).
- Настройте `includes/conn.php` для подключения к `$db_name = 'OnlineBookstore';`.

#### 2. Адаптация модуля для `Authors`:

- Создайте/переименуйте папку модуля в `authors`.

- Адаптируйте файлы `index.php`, `create.php`, `edit.php`, `actions.php` для работы с таблицей `Authors` и ее полями (`author_id`, `first_name`, `last_name`, `country`).

### 3. Адаптация модуля для `Books` :

- Создайте/переименуйте папку модуля в `books`.
- Адаптируйте файлы `index.php`, `create.php`, `edit.php`, `actions.php` для работы с таблицей `Books`.
- В `index.php` : При отображении списка книг используйте `JOIN` с таблицами `Authors`, `Publishers`, `Categories`, чтобы показать не только ID, но и ФИО автора, название издательства и название категории.
- В `create.php` и `edit.php` :
  - Добавьте поля для `title`, `publication_year`, `isbn`, `price`.
  - Реализуйте три выпадающих списка (`<select>`):
    - Для выбора автора (`author_id`): Заполняется данными из таблицы `Authors` (`SELECT author_id, first_name, last_name FROM Authors`).
    - Для выбора издательства (`publisher_id`): Заполняется данными из `Publishers` (`SELECT publisher_id, publisher_name FROM Publishers`).
    - Для выбора категории (`category_id`): Заполняется данными из `Categories` (`SELECT category_id, category_name FROM Categories`).
  - В `edit.php` не забудьте установить атрибут `selected` для соответствующих `<option>` на основе данных редактируемой книги.
- В `actions.php` : Обновите SQL-запросы `INSERT` и `UPDATE`, чтобы они корректно обрабатывали все поля, включая ID, полученные из трех выпадающих списков.

### 4. (Опционально) Адаптация для `Publishers` / `Categories`.

5. Обновите навигацию: В `includes/header.php` добавьте ссылки на главные страницы модулей ( `/authors/` , `/books/` и т.д.).

## 5. Требования к сдаче

- Часть 1: `.sql` файл, содержащий 20 SQL-запросов из п. 3.2 с комментариями и, по возможности, ожидаемым результатом (или его описанием).
- Часть 2: Полный архив вашего адаптированного веб-приложения для `OnlineBookstore` (все папки и файлы проекта).

## 6. Контрольные вопросы

1. Объясните разницу между `INNER JOIN` и `LEFT JOIN` . В каком из запросов Части 1 вы использовали `LEFT JOIN` и почему?
2. Приведите пример использования подзапроса из Части 1 (например, для нахождения книг дороже средней цены). Как работает этот подзапрос?
3. В чем разница между условиями в `WHERE` и `HAVING` ? Приведите пример запроса из Части 1, где использовался `HAVING` .
4. При адаптации модуля `Books` (Часть 2), какие таблицы вам пришлось дополнительно запрашивать в файлах `create.php` и `edit.php` и для чего?
5. Как вы передавали выбранные значения `author_id` , `publisher_id` , `category_id` из формы ( `create.php` / `edit.php` ) в скрипт-обработчик ( `actions.php` )?
6. Какой SQL-запрос вы использовали в `books/index.php` для отображения не только ID автора/издательства/категории, но и их названий/имен?

7. Представьте, что вам нужно добавить возможность поиска книг по названию в `books/index.php`. Как бы вы изменили SQL-запрос `SELECT` и как бы передали поисковый запрос из HTML в PHP?
8. Какие преимущества дает использование структуры проекта с разделением на папки (`includes`, `authors`, `books` и т.д.) при разработке веб-приложения?