

# Уніфікований інтерфейс PDO

- [Мета роботи](#)
- [Методичні вказівки](#)
- [Необхідні засоби](#)
- [Порядок виконання роботи](#)
- [Варіанти завдань](#)
- [Зміст звіту](#)
- [Контрольні питання](#)
- [Додаткові джерела](#)

## 1. Мета роботи

- Знайомство з використанням реляційних СУБД у веб-застосунках, створених за допомогою мови PHP.
- Опанування утиліти phpMyAdmin для виконання поширених операцій при роботі з базами даних.
- Вивчення засобів уніфікованого інтерфейсу PDO для роботи з СУБД MySQL.
- Розробка веб-застосунка, у якому реалізовані основні операції використання баз даних.

## 2. Методичні вказівки

Для підготовки до виконання лабораторної роботи слід пригадати синтаксис операторів мови SQL стосовно операцій вибірки (SELECT) і модифікації (INSERT, DELETE, UPDATE) даних. Потрібно ознайомитися з програмними методами виконання запитів до бази даних, а також етапами виконання **підготовленого запиту (PDO PreparedStatements)** до бази даних із застосуванням інтерфейсу PDO.

## 3. Необхідні засоби

Для виконання лабораторної роботи рекомендується використовувати таке вільне програмне забезпечення:

- редактор коду **MS Visual Studio Code** або будь-який інший, який знайомий користувачу і забезпечує відображення синтаксису (Notepad++, Sublime Text, Atom, Brackets и т.п.);
- збірка для веб-розробки **OpenServer**, до складу якої входить веб-сервер Apache, інтерпретатор мови PHP та реляційна СУБД MySQL;
- браузер Mozilla Firefox або Google Chrome;
- СКВ **Git** (для роботи з сервісом **GitHub** потрібен доступ до мережі Інтернет).

## 4. Порядок виконання роботи

1. Використовуючи утиліту phpMyAdmin, виконати імпорт готової БД за варіантом, ознайомитися з її структурою (за потреби, внести корективи).
2. За допомогою phpMyAdmin, доповнити дані. Довільний внесок до початкового вмісту бази зробить результати запитів менш передбачуваними та більш цікавими. Обов'язково зробити дамп БД.
3. Підготувати необхідні за варіантом SQL-запити до БД. Перевірити виконання запитів у phpMyAdmin, використовуючи довільні значення параметрів.

4. Створити веб-застосунок, що запитує в користувача параметри для запитів і виводить їхні результати. Використовувати при цьому підготовлені вирази (тобто звертатися до БД через методи `prepare`, `execute` та прив'язку значень параметрів, отриманих від користувача, засобами `bindValue`, `bindParam` або безпосередньо в `execute`).

Рекомендації (дотримання яких спростить подальшу роботу):

- розділити код на стартову сторінку (на якій розташовано **окремі** форми для вказівки параметрів запитів), і окремі `php`-файли для обробників **кожного** запиту на отримання даних. У майбутньому це суттєво спростить модифікацію;
- оскільки код для створення підключення до БД для всіх запитів у межах одного варіанта завдання однаковий (тому що БД використовується одна), тому є сенс розмістити його в окремому `php`-файлі, і підключати (наприклад, через `include`) його за місцем вимоги, а не дублювати в кожному обробнику;
- на стартовій сторінці, елементи управління у формах для введення параметрів користувачем, де це має сенс, варто робити не текстовими полями, а списками, що випадають (формуючи їх на основі даних із БД). Це дасть змогу користувачеві обирати тільки можливі варіанти даних для запиту, а не вводити значення самостійно - це, окрім іншого, прискорить тестування вашої роботи;
- в елементах для введення значень параметрів, у тих випадках, коли їх реалізовано як звичайні текстові поля для введення (буває, що інакше неможливо), має сенс використовувати атрибут `value` зі значенням за замовчуванням. Це також спростить налагодження застосунка та його демонстрацію.

Завдання для лабораторної роботи є індивідуальними. Можливе спільне виконання у "бригаді" варіантів **усіх** її учасників. Номер варіанта кожного студента вказано в **електронному журналі**. Виконання чужого варіанту є підставою для відхилення роботи.

Відпрацювання виставляється тільки після спільної зі студентом перевірки роботи проекту, виконаного відповідно до **всіх** завдань цього розділу та правильно обраного **варіанту**.

## 5. Варіанти завдань

**Варіант 0.** БД для зберігання інформації про [інформаційні ресурси бібліотеки](#) (Рисунок 5.0).

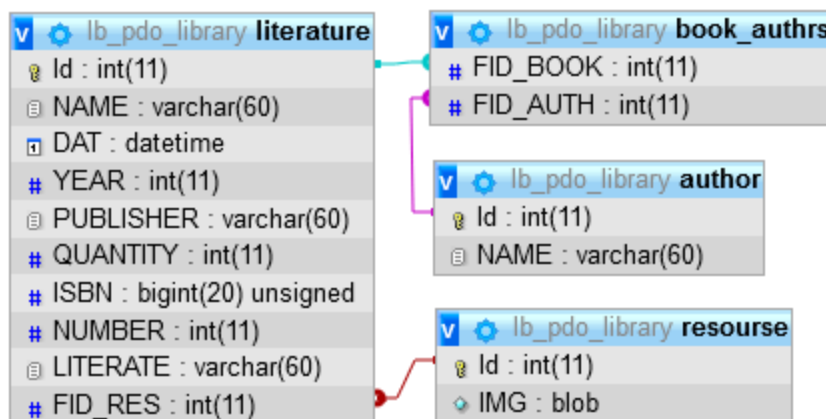


Рисунок 5.0 - Структура бази даних «Бібліотека»

Забезпечити виведення інформації о:

- книгах зазначеного видавництва;

- книжках, журналах і газетах, опублікованих за вказаний часовий період (враховувати рік видання);
- книгах зазначеного автора.

Приклади скріншотів [форми для пошуку за видавництвом](#) і відповідного [результату пошуку](#).

**Варіант 1.** БД для зберігання інформації про [розклад проведення занять](#) (Рисунок 5.1).

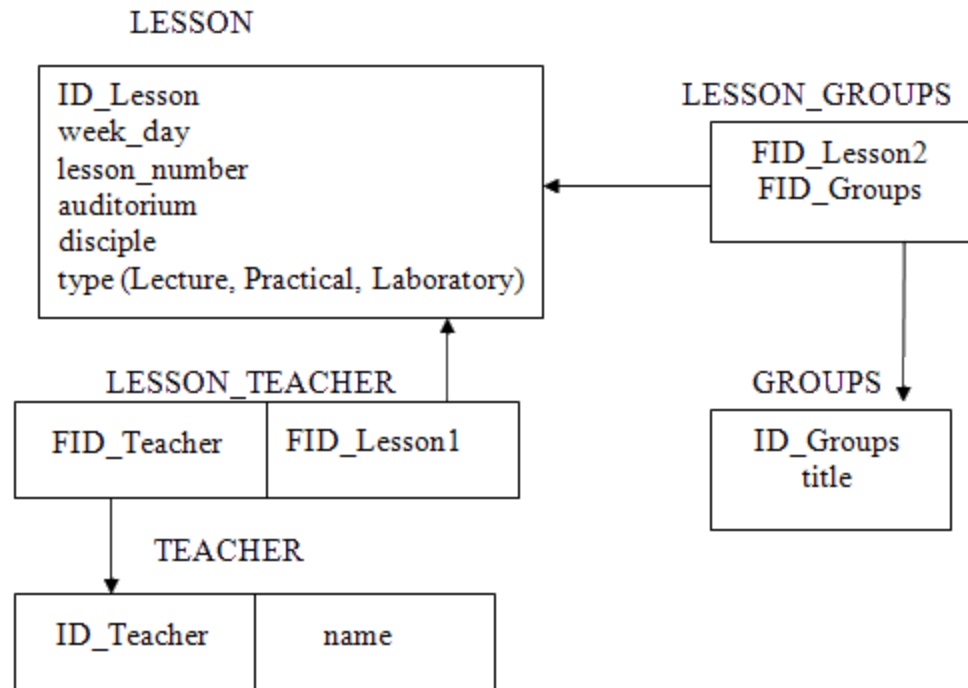


Рисунок 5.1 – Структура бази даних «Розклад занять»

Забезпечити виведення розкладу занять для:

- довільної групи зі списку;
- довільного викладача зі списку;
- аудиторії зі списку.

Приклад скріншоту [результату](#) запиту для розкладу групи КІ-12-1.

**Варіант 2.** БД для зберігання інформації про [процес роботи співробітників над проектами](#) (Рисунок 5.2).

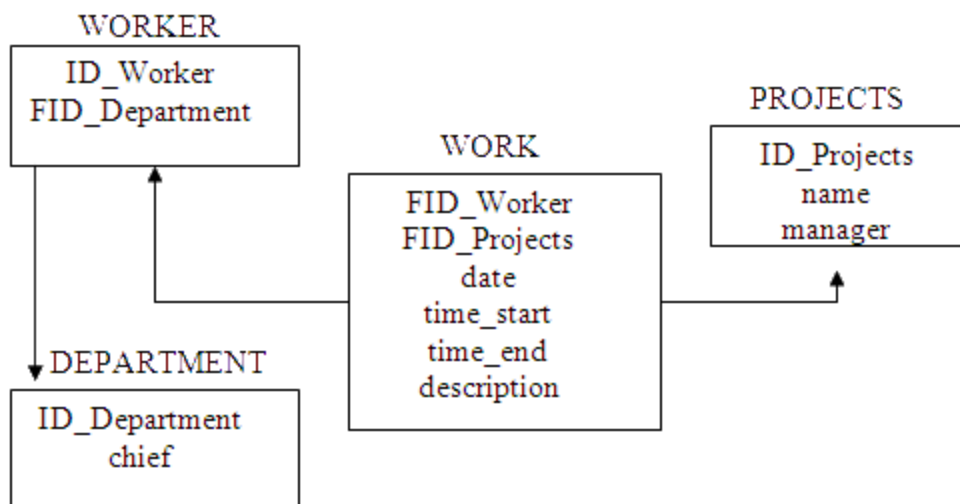


Рисунок 5.2 – Структура бази даних «Робота співробітників»

Забезпечити виведення таких даних:

- інформація про виконані завдання за обраним проєктом на зазначену дату;
- загальний час роботи над обраним проєктом;
- кількість співробітників відділу обраного керівника.

Приклад скріншота стартової сторінки застосунка з формами для вказування параметрів.

**Варіант 3.** БД для зберігання інформації про робочі зміни чергувань медсестер у поліклініці (Рисунок 5.3).

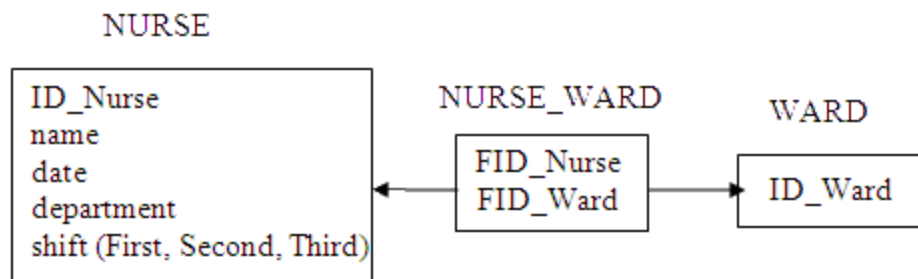


Рисунок 5.3 – Структура бази даних «Зміна чергувань» (з помилкою, не вистачає поля в таблиці ward)

Для медсестри зазначено її ім'я, відділення та робочу зміну (перша, друга або третя). Для палати (hospital ward) - назва (номер). Медсестри можуть працювати в різних палатах, в одній палаті може бути більше однієї медсестри.

Забезпечити виведення таких даних:

- перелік палат, у яких чергує обрана медсестра;
- медсестри обраного відділення;
- чергування (у будь-яких палатах) у зазначену зміну.

**Варіант 4.** БД для зберігання інформації про фільмотеку (Рисунок 1.4).

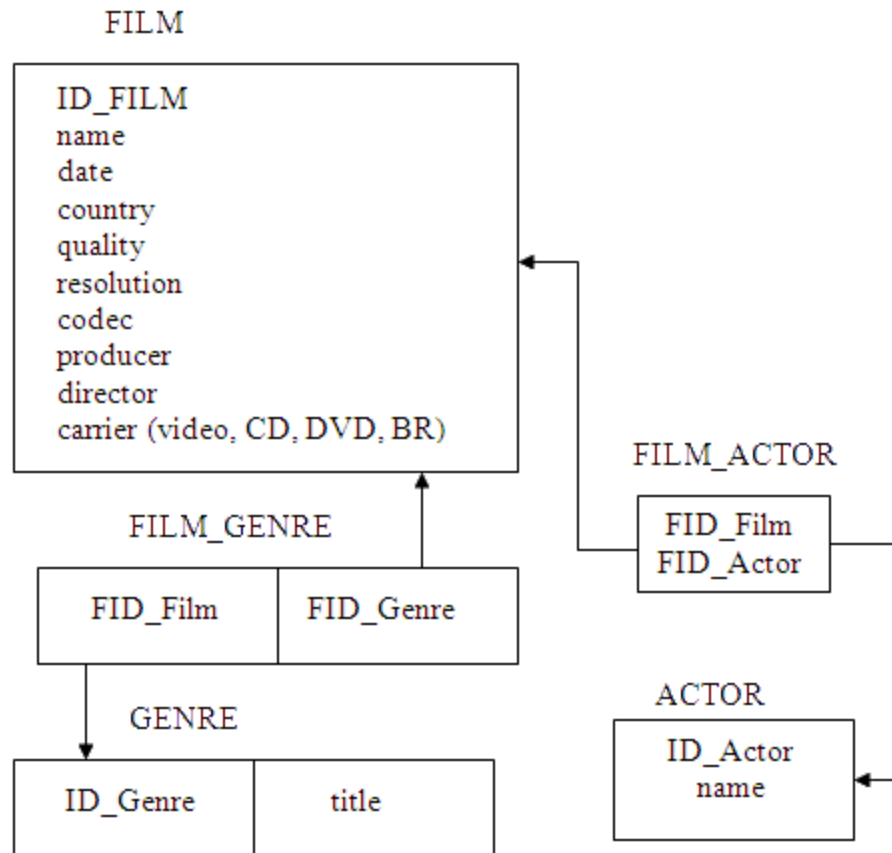


Рисунок 5.4 – Структура бази даних «Фільмотека»

Забезпечити виведення таких даних:

- список фільмів обраного жанру;
- список фільмів з обраним актором;
- список фільмів за вказаний часовий інтервал.

Приклади скріншотів [сторінки з усіма формами для пошуку](#) і [результату пошуку за одним з запитів](#).

**Варіант 5.** БД для зберігання інформації про [товари в інтернет-магазині](#) (Рисунок 5.5).

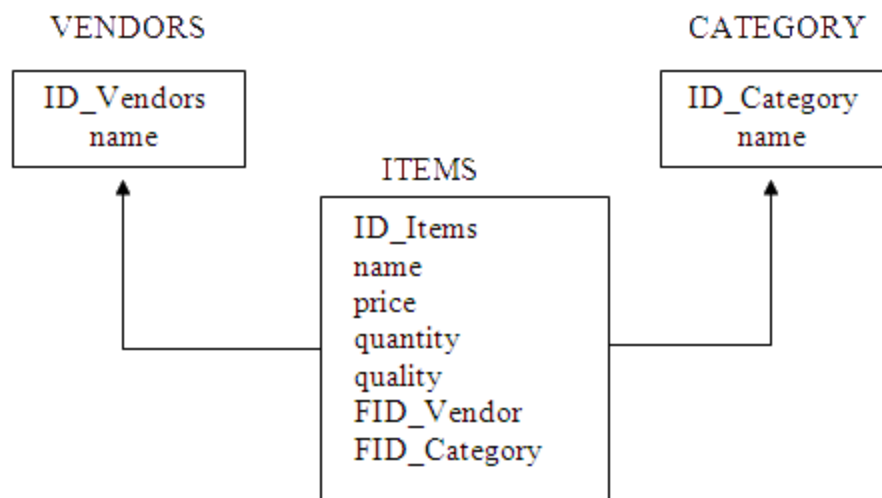


Рисунок 5.5 – Структура бази даних «Товари в магазині»

Забезпечити виведення таких даних:

- товари обраного виробника;
- товари в обраній категорії;
- товари в обраному ціновому діапазоні.

**Варіант 6.** БД для зберігання [інформації про пункт прокату машин](#) (Рисунок 5.6).

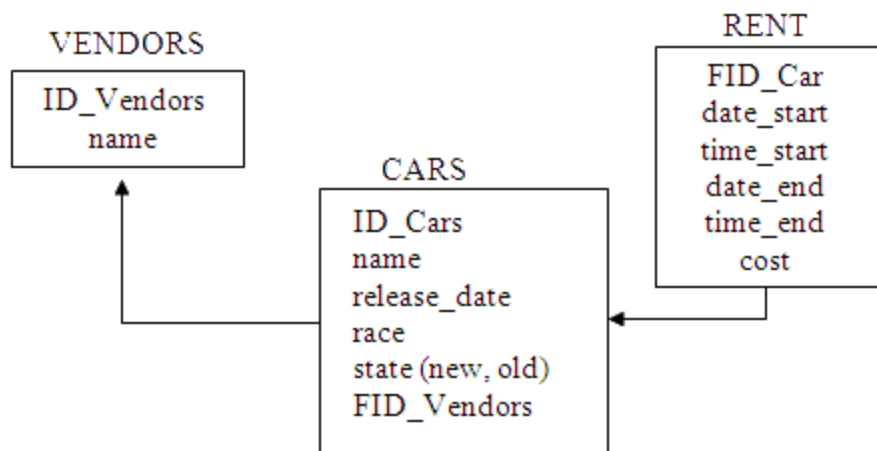


Рисунок 5.6 – Структура бази даних «Прокат машин»

Забезпечити виведення таких даних:

- отриманий дохід з прокату станом на обрану дату;
- автомобілі обраного виробника;
- вільні автомобілі на обрану дату.

**Варіант 7.** БД для зберігання інформації провайдеру про [використання мережевого трафіку](#) (Рисунок 5.7).

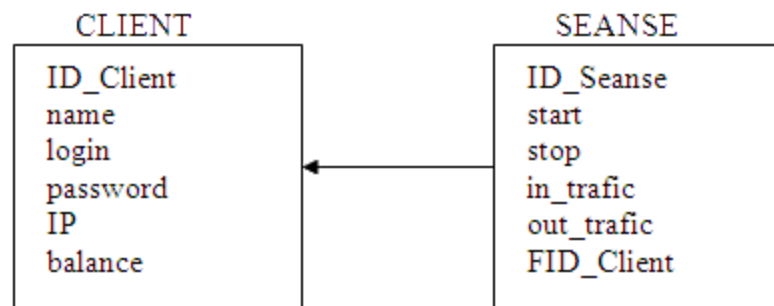


Рисунок 5.7 – Структура бази даних «Мережевий трафік»

Забезпечити виведення таких даних:

- сеанси роботи в мережі для обраного клієнта;
- сеанси роботи в мережі за вказаний проміжок часу;
- список клієнтів з від'ємним балансом рахунку.

## 6. Зміст звіту

Звіт з лабораторної роботи повинен містити: мету роботи, варіант завдання, вихідний код веб-сторінки (HTML, JavaScript и PHP), ER-діаграму структури бази даних, зображення отриманого документа, висновки по роботі. Наявність звіту не є обов'язковою. При викладанні вихідного коду на GitHub пам'ятайте про дамп БД.

## 7. Контрольні запитання

1. Які розширення PHP для роботи з СУБД MySQL вам відомі?
2. Які недоліки використання реляційних СУБД?
3. У чому полягають переваги застосування інтерфейсу доступу до баз даних PDO?
4. Яким чином здійснюється керування підключеннями до баз даних? Як створити постійне підключення?
5. Як виконати SQL-запит на зміну даних, використовуючи інтерфейс PDO?
6. Як отримати результати запиту на вибірку?
7. Як отримати кількість рядків, які були змінені під час виконання останнього запиту на виконання?
8. Як отримати ID останнього доданого рядка?
9. Як перевірити наявність помилок під час з'єднання з базою даних або виконання SQL-запиту? У чому різниця між використанням методу `errorInfo()` для об'єкта `PDOStatement` і для об'єкта `PDO`?
10. Для яких цілей призначені підготовлені запити? Як виконати зв'язування псевдозмінної та значення?
11. Які існують режими вибірки?
12. Який режим вибірки використовується за замовчуванням?
13. У чому відмінності між режимами вибірки `PDO::FETCH_NUM` и `PDO::FETCH_BOTH`?
14. Для чого служить режим вибірки `PDO::FETCH_KEY_PAIR`?
15. Як отримати рядок результуючої вибірки у вигляді об'єкта?
16. Для чого використовується метод `fetchAll`?
17. Визначення транзакції, приклад використання?

Знання відповідей на наведені запитання є необхідною, але не достатньою умовою для отримання високої оцінки за захист лабораторної роботи. Пам'ятайте про теми, розглянуті на лекціях!

### Корисні посилання

1. [Документація до мови PHP](#)
2. Посібник із сучасної PHP-розробки [PHP: The Right Way](#)
3. [Налаштування та використання PDO](#)
4. [Вільний опис процесу роботи с PDO](#)
5. [PHP Prepared Statements \(в PDO и MySQLi\)](#)
6. [Інформаційний ресурс о SQL](#)
7. Phil Sturgeon and Josh Lockhart "PHP: The Right Way"
8. Мэтт Зандстра "PHP. Объекты, шаблоны и методики программирования".
9. Энди Гутманс, Стиг Баккен, Дерик Ретанс "PHP 5. Профессиональное программирование".

[Повернутися до початку](#)