**Лабораторна робота 20.**

**Робота із SQL Server через драйвер PDO\_SQLSRV.**

**Мета роботи:** Навчитись підключатись до БД, виконувати запити на вибірку, модифікацію даних, збережені процедури та обробляти результат, що вони повертають, використовуючи драйвер PDO\_SQLSRV

**Задачі:**

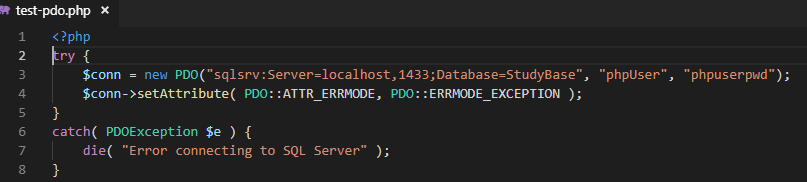
* Познайомитись із поняттям PDO, його перевагами та недоліками
* Виконати підключення до БД, вибірку та модифікацію даних
* Виконати збережену процедуру із параметрами та обробити множинне повернення даних
* Познайомитись із командами керування транзакціями
* Виконання індивідуального завдання згідно варіанту

1. PHP Data Objects (PDO) — розширення для PHP, що надає розробнику простий і універсальний інтерфейс для доступу до різних баз даних. PDO пропонує єдині методи для роботи з різними базами даних, хоча текст запитів може трохи відрізнятися. Оскільки багато СУБД реалізують свій діалект SQL, який в тій чи іншій мірі підтримує стандарти ANSI і ISO, то при використанні простих запитів можна добитися сумісності між різними мовами. На практиці це означає, що можна досить легко перейти на іншу СУБД, при цьому не змінюючи або частково змінюючи код програми.

Тобто, з одного боку маємо універсальний інтерфейс класів для роботи із переважною більшістю найпопулярніших СУБД, але при цьому деяку обмеженість функціоналу через складність використання специфічних конструкцій того чи іншого діалекту SQL.

Розширення Microsoft SQL Server для PHP надає об'єктно орієнтований інтерфейс PDO за допомогою драйвера PDO\_SQLSRV. PDO підтримують об'єкти двох класів - PDO і PDOStatement. Клас PDO містить методи, що дозволяють застосунку PHP підключатися до інстанції SQL Server. Клас PDOStatement представляє інструкцію і результати її виконання.

1. У папці з проектом створимо «test-pdo.php». У даному файлі виконаємо підключення до БД та вибірку із БД спеціальності із кодом = 2, використовуючи PDO\_SQLSRV (рис.1)



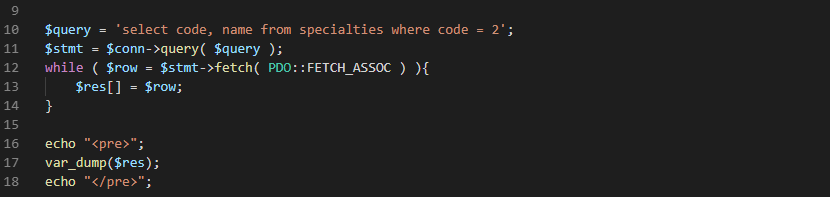


Рис.1. Реалізація підключення до БД та вибірки даних із використанням PDO\_SQLSRV

Результат перевіримо, набравши в адресному рядку «http://sql-server/test-pdo.php» (рис.2)

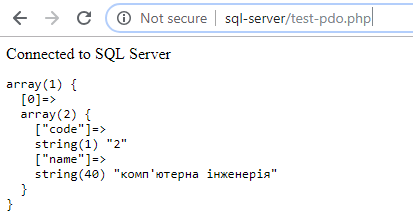


Рис.2. Результат вибірки спеціальності із кодом 2

1. Далі виконаємо послідовне додавання, зміну та видалення даних із виведенням результату. Почнемо із додавання, яке реалізуємо через метод query() для отримання коду рядку, який було згенеровано БД автоматично (рис.3) (тут і надалі етап підключення до БД опускається).

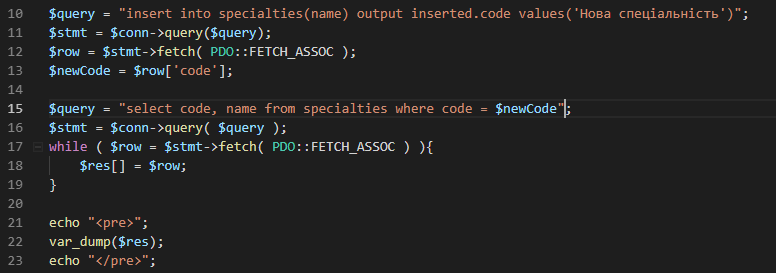


Рис.3. Додавання рядку до таблички спеціальностей

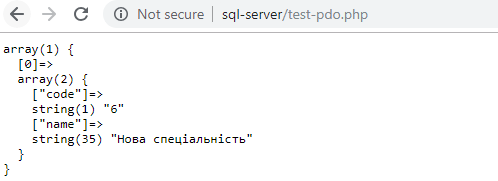


Рис.4. Виведення доданої спеціальності на екран

Зміну зміни тільки що доданого рядку виконаємо, використавши методи prepare() та execute() (рис.5-6)

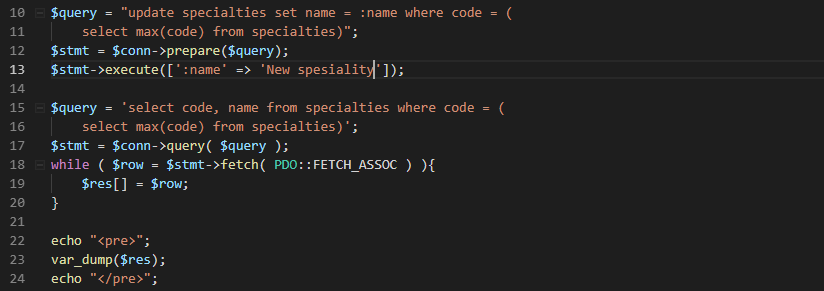


Рис.5. Використання методу exec()

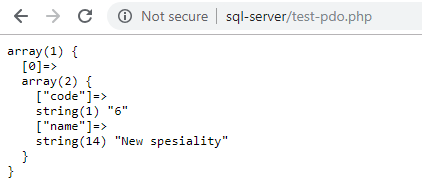


Рис.6. Виведення зміненої спеціальності на екран

Прості запити також можна виконувати із використанням методу exec(). Використаємо даний метод для видалення створеного рядку (рис.7-8)

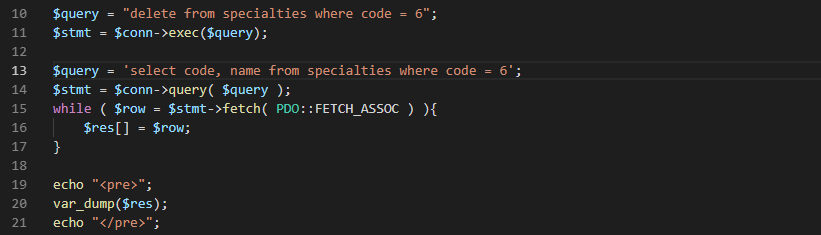


Рис.7. Видалення рядку із кодом 6.

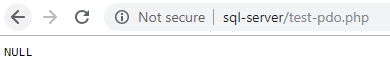


Рис.8. Спроба читання видаленого рядку.

1. Тепер використовуючи PDO\_SQLSRV виконаємо якусь збережену процедуру із параметрами, що повертає декілька результуючих наборів. Для прикладу створимо процедуру, що за кодом факультету повертає його назву у вигляді параметру, а також його кафедри та студентські групи у вигляді наборів рядків (рис.9)

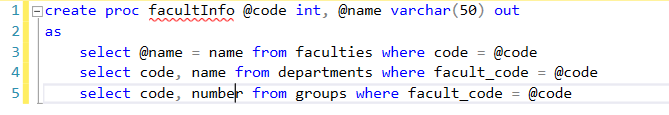


Рис.9. Створення процедури

Далі виконаємо створену процедуру, спочатку в SSMS (рис.10), а потім у клієнтському застосунку із використанням PDO\_SQLSRV (рис.11) та перевіримо роботу процедури у браузері.

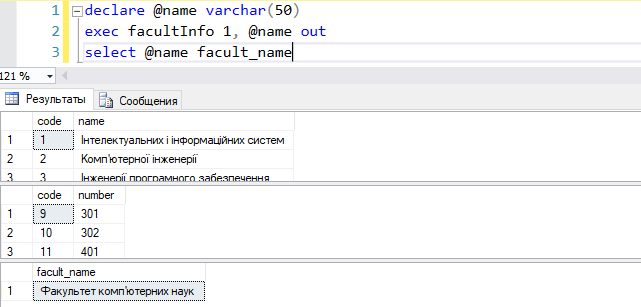


Рис.10. Виконання створеної процедури в SSMS

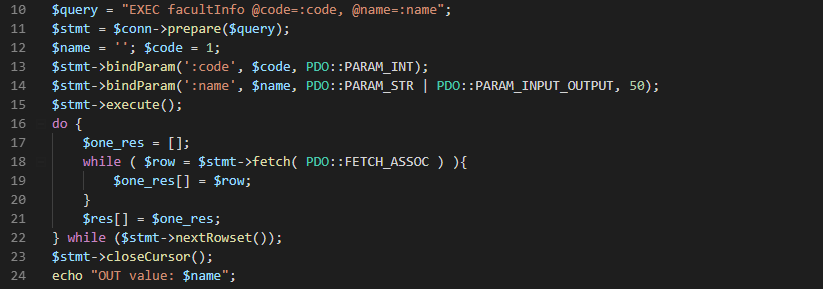
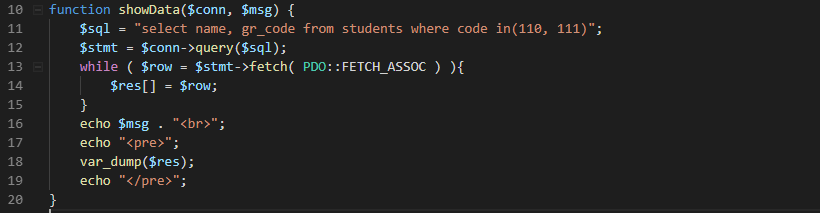


Рис.11. Код виконання створеної процедури в PHP

1. Розглянемо роботу із транзакціями у PDO. Реалізуємо приклад, аналогічний тому, що виконувався у попередній роботі – виконаємо 2 команди зміни даних у межах однієї транзакції та зафіксуємо транзакцію, якщо обидві команди виконались успішно, або відкотимо транзакцію, якщо хоча б одна не виконалась (рис.12-13)



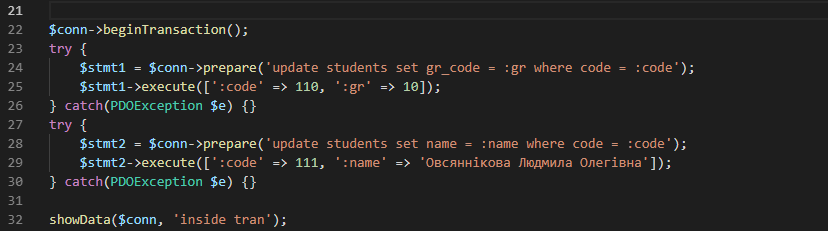




Рис.12. Робота із транзакціями у PDO.

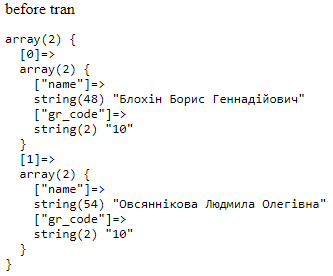
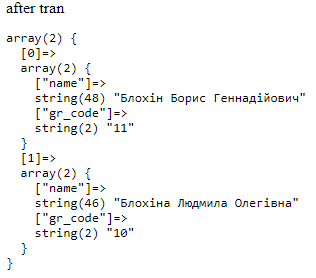
 

Рис.13. Приклад виконання транзакції із наступною фіксацією.

Бачимо, що в результаті фіксації транзакції застосовуються обидві команди зміни даних. Допустимо помилку у імені таблиці у першій команді та виконаємо скрипт повторно (рис.14-15)

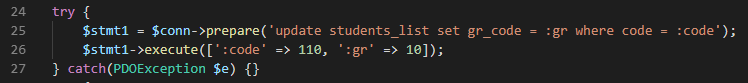


Рис.14. Помилка у назві таблиці для перевірки відкату транзакції

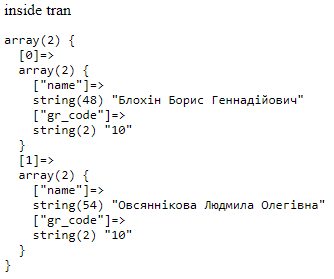
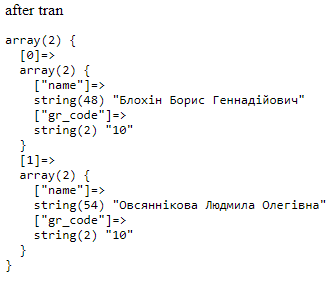
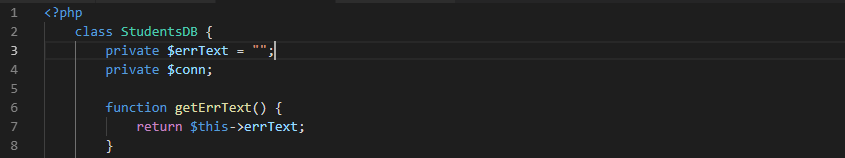
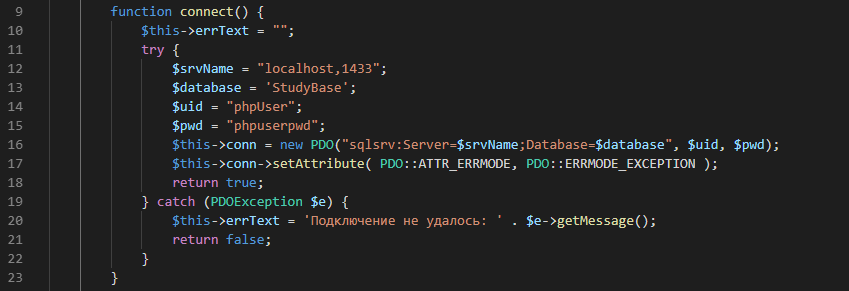
 

Рис.15. Приклад виконання транзакції із наступним відкатом.

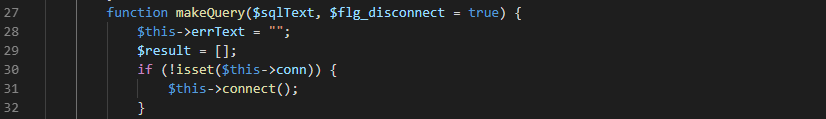
Транзакція відкочується повністю, отже ні перша ні друга команди не виконується.

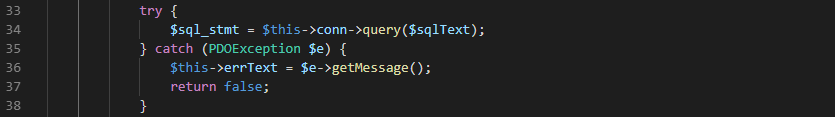
1. Далі для закріплення матеріалу переробимо клас StudentsDB, що реалізовано у файлі «data/db.php» та реалізовує функціонал підключення-відключення до БД, вибірки даних, виконання команд модифікації даних та збереженої процедури (рис.16).

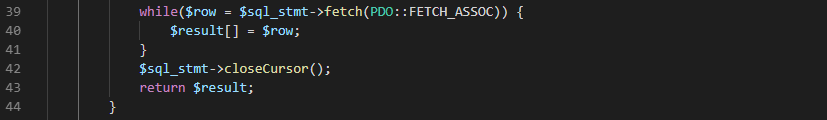


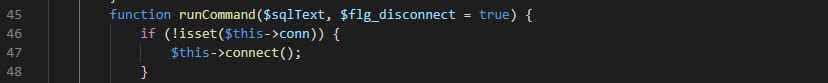


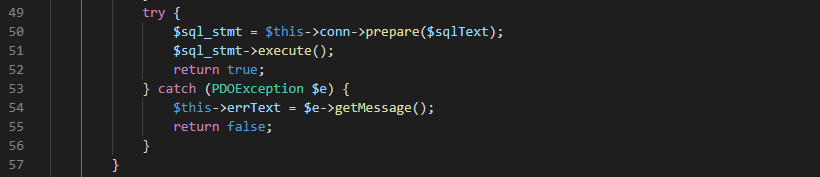


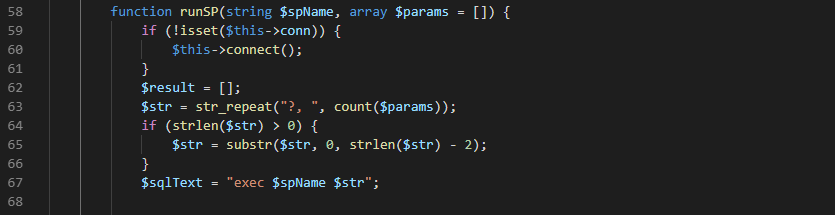


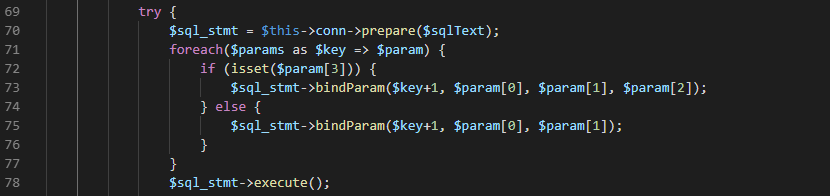


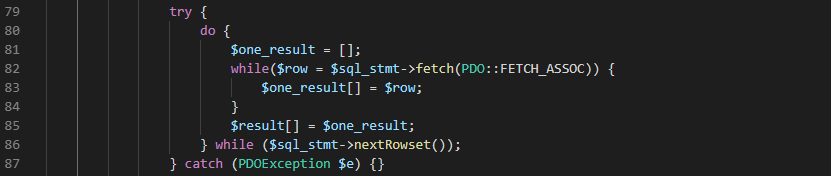


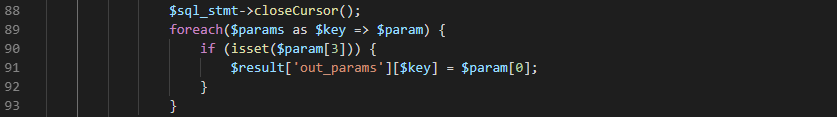












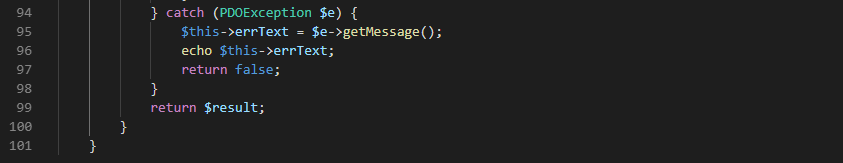
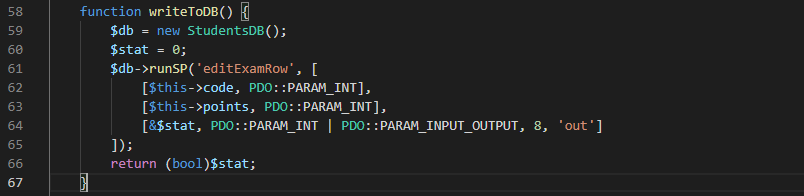


Рис.16. Реалізація класу StudentsDB із використанням PDO\_SQLSRV

Також трохи змінимо процедури, що виконують збережені процедури у класах ExamRow та Exam (рис.17)



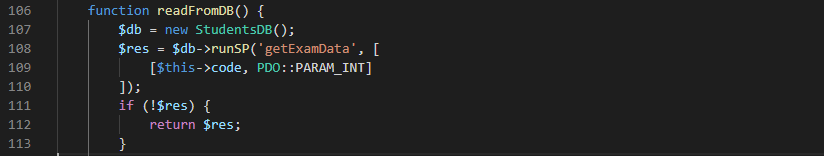


Рис.17. Зміна методів writeToDB та readFromDB у класах ExamRow та Exam

Перевіримо роботу наших змін. Запустимо у браузері нашу сторінку «http://sql-server/» та виконаємо додавання, зміну та видалення даних.

1. Кінець роботи.

**Завдання для індивідуального виконання.**

Для власного застосунку, що реалізовано у попередніх роботах, який забезпечує роботу із однією таблицею вашої БД, реалізувати взаємодію із SQL Server через PDO\_SQLSRV.