Тканко О.В.
ННК "ІПСА" НТУУ "КПІ", Київ, Україна

Обгортка для SQL запитів до реляційних баз даних

Обгортка для SQL запитів надає можливість простого доступу до реляційної бази даних, легкої її зміни, кешування запитів. Основні вимоги до базового класу обгортки для SQL запитів до реляційних баз даних:

- легка і зручна робота з класом;
- встановлення правильно налаштованого під'єднання до бази даних;
- можливість встановлення одночасно кількох під'єднань;
- економія ресурсів;
- захист від SQL ін'єкцій;
- можливість кешувати запити;
- можливість переходу на іншу базу даних.

У реляційній базі даних дані зберігаються в окремих таблицях, завдяки цьому досягається виграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин і тому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць [1]. Для того, щоб уникнути обмеження у використанні MySQL, в якості реляційної бази даних, при створенні інтерфейсу використовується механізм наслідування. При створенні об'єкта метод-конструктор викликається автоматично, а отже в конструкторі SQL wrapper'а доцільно виконати підключення, перевірити його правильність і вибрати базу даних [1,2].

Для того, щоб створити екземпляр класу обгортки (`Default_DB`), йому необхідно безпосередньо вказати місце знаходження бази даних, порт, а також зазначити логін і пароль для під'єднання до бази даних. Такий процес є громіздким і зручно мінімізувати створення екземпляру обгортки до однієї строки `$dbh = new Default_DB`. Це можна реалізувати за допомогою функції `ini_get`, що дозволяє отримувати значення змінної із файлу конфігурації. Отримані значення доцільно зберігати у статичних змінних [3].

Зручно мати можливість зміни кодування по-замовчанню і при цьому не встановлювати його вручну для кожної сесії. Таким чином, для того, щоб не виконувати зайвих запитів, потрібно перевірити, чи під'єднання вже було відкрите і після цього встановити кодування по замовчанню.

При зверненні до методу `query` базового класу обгортки параметри повинні перевірятись на наявність SQL-ін'єкції. SQL-ін'єкцію називають дії, при виконанні яких можна отримати доступ до бази даних із подальшою можливістю виконання запитів. Найчастіше такий доступ можна отримати при використанні форм відправляння інформації на сервер. Для цього, використовуючи поля вводу, відсилаються запити, які, по необережності, виконується сервером. Для того щоб захиститись від SQL-ін'єкції, всі зовнішні параметри (`$_GET`, `$_POST`, `$_COOKIE`) слід, перед тим як включити їх до SQL запиту, опрацювати за допомогою функції `mysql_real_escape_string()`, а в самому запиті помістити їх до одинарних кавичок [4,5]. Для того, щоб не турбуватись про закриття відкритого з'єднання, використовуються функція деструктору (`__destruct()`), яка викликається автоматично при видаленні об'єкта [3].

Подальшим кроком розвитку даної обгортки є автоматичне створення запитів в залежності від отриманих параметрів.

1. Методика реалізації об'єктно-реляційного відображення у середовищі.NET [Електронний ресурс]: УкрПрог, 1–3 червня 2004 р., м. Київ, Україна; Дорошенко А.Е., Романенко В.Г. – Режим доступу: http://eprints.isofts.kiev.ua/310/1/D82.pdf.
2. ООП в PHP 5 – конструкторы и деструкторы [Електронний ресурс]: PHPworld.ru – все о программировании на PHP, Москва, Россия; Леонид Лукин – 2004 – Режим доступу: http://www.phpworld.ru/php5/php5constr_destr.php.
3. Функции PHP. Информационные и опционные функции PHP [Електронний ресурс]: Андрей Транский – 2006 – Режим доступу: http://php.su/functions/?ini-get.
4. Leon Atkinson Core PHP Programming, Third Edition [Текст] /Leon Atkinson / / Prentice Hall. – August 05, 2003. – с. 62–66.
5. Як боротись з SQL-ін'єкцією за допомогою PHP [Електронний ресурс]: ART-studio. Все для веб розробника.; Артур Галма – 2010 – Режим доступу: http://art-studio.com.ua/programming/articles/php/120-securityvssqlinjection.html.

ISBN 978-966-2153-57-6, revision 0.9.2 (2011.05.12), available at http://sait.kpi.ua/books/sait2011.ebook.pdf