

Бутик Р. В.

*студент фізико-математичного факультету*

**Т.А. Вакалюк**

*канд. пед. наук,*

*доцент кафедри прикладної математики та інформатики*

*Житомирського державного університету імені Івана Франка*

## **РЕАЛІЗАЦІЯ МІГРАЦІЙ БАЗ ДАНИХ ЗАСОБАМИ PHP ФРЕЙМВОРКУ PHALCON**

*Анотація: Стаття присвячена огляду міграцій баз даних. У статті розглядаються основні поняття міграцій та реалізація їх засобами Phalcon.*

*Ключові слова: PHP, web розробка, Phalcon, міграції, бази даних.*

*Аннотація: Стаття посвячена обзору міграцій баз даних. В статті рассматриваются основные понятия миграций и реализация их средствами Phalcon.*

*Ключевые слова: PHP, web разработка, Phalcon, миграции, базы данных.*

*Summary: This article reviews the database migration. This article discusses the basic concepts of migration and implementation of their means Phalcon.*

*Key words: PHP, web development, Phalcon, migrations, data bases.*

На даний момент існує безліч рішень для реалізації зручної роботи по внесенні змін у структуру бази даних. Один із аналогів являє собою міграції реалізовані в PHP фреймворку Yii2. Вони мають схожу структуру із міграціями у Phalcon та надають майже ідентичний інтерфейс для їх реалізації [5]. Крім цього, у будь-яких достатньо серйозних фреймворках /бібліотеках на мові PHP існують свої аналоги даної технології.

**Мета статті.** Вияснити що собою являють міграції та як їх реалізувати засобами фреймворку Phalcon.

**Виклад основного матеріалу.** Міграції — це зручний спосіб для нас, структуровано та організовано змінювати наші бази даних [1].

Часто при розробці досить необхідно вносити зміни вже на стадії релізу програмного забезпечення, на головному сервері. Деякі із цих змін можуть стосуватися змін в базі даних: нові стовпці, нові таблиці, видалення індексів і т. д.

Міграції доступні через Phalcon Developer Tools [3]. Нам знадобиться версія фреймворку вище 0.5.0 для використання інструментів розробника. Також необхідно використовувати PHP версії вище 5.4.

При міграції створюється набір класів, щоб описати, як наша база даних структурована в даний момент. Дані класи можна використовувати для синхронізації структури схеми у віддалених базах даних та підготовки нашої бази даних для роботи із новими змінами, які реалізує наша програма. Міграції описують дані зміни за допомогою простого PHP [2].

Phalcon Developers Tools надає скрипт для управління міграціями (генерація, запуск та відкат).

Запуск скрипта без параметрів робить простий дамп кожного об'єкта (таблиць) із бази даних в класи міграцій.

Кожна міграція має ідентифікатор версії пов'язаний з нею. Номер версії допомагає нам визначити чи дана міграція є старше або новіше поточної версії нашої бази даних. Версії також повідомляють Phalcon про робочий стан при виконанні міграції. При генерації міграції, інструкції виводяться в консоль для опису різних етапів міграції та часу виконання даних операторів. В кінці кінців, версія міграції буде створена.

За замовчуванням Phalcon Developers Tools використовує директорію `app/migrations` для дампа файлів міграції. Ми маємо можливість змінювати розміщення, встановлюючи один із параметрів перед генерацією скрипту. Кожна таблиця в базі даних має свій відповідний клас, створений в окремому файлі директорії, який посилається на її версію.

Кожен файл містить унікальний клас, який розширює `Phalcon\Mvc\Model\Migration` клас. Дані класи зазвичай мають два методи: `up()` та `down()`. `up()` виконує міграцію, а `down()` повертає до попереднього

стану зроблені зміни. Метод up() також містить магічний метод morphTable(). Магія починається тоді, коли він розпізнає зміни, котрі потребують синхронізації фактичної таблиці в базі даних.

```
1 <?php
2
3 use Phalcon\Db\Column as Column;
4 use Phalcon\Db\Index as Index;
5 use Phalcon\Db\Reference as Reference;
6
7 class ProductsMigration_100 extends \Phalcon\Mvc\Model\Migration
8 {
9     public function up()
10    {
11        $this->morphTable(
12            "products",[
13                "columns" => [
14                    new Column(
15                        "id",
16                        [
17                            "type"           => Column::TYPE_INTEGER,
18                            "autoIncrement" => true,
19                            "first"          => true
20                        ]
21                    ),
22                    new Column(
23                        "product_types_id",
24                        [
25                            "type"           => Column::TYPE_INTEGER,
26                            "after"          => "id"
27                        ]
28                    ),
29                    new Column(
30                        "name",
31                        [
32                            "type"           => Column::TYPE_VARCHAR,
33                            "after"          => "product_types_id"
34                        ]
35                    ),
36                    new Column(
37                        "price",
38                        [
39                            "type"           => Column::TYPE_DECIMAL,
40                            "after"          => "name"
41                        ]
42                    ),
43                ],
44            );
45    }
46 }
47 }
```

Рис. 1. Звичайна міграція, написана згідно синтаксису фреймворка Phalcon

На рис. 1 ми маємо приклад звичайної міграції написаної згідно синтаксису фреймворка Phalcon [4]. Клас називається “ProductsMigration\_100”. Суфікс 100 вказує на версію 1.0.0. morphTable() приймає асоціативний масив із чотирма можливими типами даних: columns, indexes, references, options. Phalcon\Db\Column використовується для визначення стовпців таблиці. Даний клас інкапсулює можливості

пов'язані із реальним стовпцем. Його конструктор приймає в якості першого параметра ім'я стовпця і масив, який описує колонки.

Міграції призначені не тільки для зміни структури таблиці. Міграція являє собою звичайний клас PHP, так що ви не обмежені даними функціями. Наприклад, після додавання стовпця можна написати код для встановлення значення цього стовпця для існуючого запису.

Як тільки згенеровані міграції завантажені на сервер, ми можемо легко запустити їх командою `phalcon run-migration`.

В залежності від того, наскільки сильно застарівшою являється база даних по міграції, Phalcon може запустити декілька версій міграції в одному процесі міграції. Якщо вказати цільову версію, Phalcon буде запускати потрібні міграції доти, поки не досягне вказаної версії.

**Висновки та перспективи подальших досліджень.** Як ми можемо спостерігати із матеріалу викладеного вище, міграції являються чудовим інструментом для реалізації змін структури бази даних, які економлять час розробника при внесенні змін, що зараз вважається актуальною проблемою у процесі розробки програмного забезпечення. Крім того, вони надають чудовий інтерфейс для їх реалізації.

#### **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ:**

1. Phalcon PHP Framework Blog, Database Migrations [Електронний ресурс]. - Режим доступу: <http://phalconphp.tumblr.com/post/22411506007/database-migrations>
2. YouTube, Tutorial PhalconPHP — Phalcon Developer Tools Database Migrations HD [Електронний ресурс]. - Режим доступу: <https://www.youtube.com/watch?v=Vx-9GiZt68I>
3. GitHub, Phalcon / phalcon-devtools [Електронний ресурс]. - Режим доступу: <https://github.com/phalcon/phalcon-devtools/blob/master/scripts/Phalcon/Mvc/Model/Migration.php>
4. Code.Htmlasks, auto\_increment in Phalcon migrations system [Електронний ресурс]. - Режим доступу:

[http://code.htmlasks.com/auto\\_increment\\_in\\_phalcon\\_migrations\\_system](http://code.htmlasks.com/auto_increment_in_phalcon_migrations_system)

5. Suyati, Yii and Phalcon — A comparative study [Электронный ресурс]. -  
Режим доступа: <http://suyati.com/yii-vs-phalcon-comparative-study/>