

Татьянкин Владислав Михайлович

студент, секретарь-администратор центра прогнозирования
ФГБОУ ВО «Югорский государственный университет»
г. Ханты-Мансийск, ХМАО – Югра

ИСПОЛЬЗОВАНИЕ И ОСОБЕННОСТИ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ (ORM)

***Аннотация:** в статье разбирается понятие «объектно-реляционное отображение» (ORM), а также раскрываются особенности такого подхода.*

***Ключевые слова:** ORM, Php, объектно-реляционное отображение, SQL.*

Для начала разберемся с понятием ORM. ORM – это многофункциональное средство для проектирования и осуществления запросов к моделям базы данных без использования прямых SQL запросов [1]. Т.е. ORM – это прослойка между базой данных и кодом, которая позволяет созданными в программе объектами манипулировать в БД.

Также понятие ORM можно раскрыть как «технология программирования использующую модели и упрощающую манипуляцию объектами для работы с БД [1]».

Смысл технологии простыми словами заключается в том, что пользователю не нужно писать прямые SQL запросы, а достаточно создать модель данных и далее посредством ORM обращаться к объектам этой модели.

Пример создания модели и обращение к ее объекту на языке PHP фреймворка laravel представлен на рисунке 1.

Laravel – на данный день самый быстроразвивающийся и популярный фреймворк получивший признание многих опытных специалистов и идеально подходящий для написания небольших и средних по сложности проектов. Встроенные возможности фреймворка – авторизация, сессия, запросы и RESTful-маршрутизация [2].

```
1. <?php
2. namespace App\Http\Controllers;
3. use App\Flight;
4. use Illuminate\Http\Request;
5. use App\Http\Controllers\Controller;
6.
7. class FlightController extends Controller
8. {
9.     public function store(Request $request)
10.    {
11.        $flight = new Flight;
12.
13.        $flight->name = $request->name;
14.
15.        $flight->save();
16.    }
17. }
```

Рис. 1. Пример создания модели и внесение в нее данных

Особенности ORM:

- схема БД явно описывается одним из используемых языков программирования;
- сниженные затраты на разработку;
- автоматическое создания SQL-запросов, избавляющие пользователя от использования языка программирования;
- автоматический перенос старых SQL-запросов на новую систему за счет низкоуровневого драйвера ORM;
- за счет встроенной логики работы с моделями ORM избавляет пользователя от большинства ошибок, появление которых возможны в процессе написания кода;
- развитые реализации ORM поддерживают отображение наследования и композиции на таблицы;
- весь генерируемый ORM код проверен и оптимизирован, поэтому не нужно задумываться о его корректности работы;

– ORM дает возможность изолировать код программы от подробностей хранения данных [3; 4].

ORM позволяет пользователю сэкономить значительное количество времени по сравнению с написанием собственных запросов, поэтому использование этой технологии в небольших проектах крайне выгодно [4].

Однако стоит понимать, что эта технология достаточно ограничена. Ее использование в конкретных целях хватит целиком и полностью. Но если пользователю понадобится делать более сложные запросы с большим количеством выборок и сложной логикой, то нативный SQL в этом случае незаменим.

Подводя итог, можно сказать, что ORM – сильно упрощающая технология работы с БД. Она не требует от пользователя проверки стабильности кода и не заставляет его заботиться о работе на концептуальном уровне. Также в большинстве фреймворков использующих эту технологию, уже встроена безопасность запросов.

Список литературы

1. Object Role Modeling [Электронный ресурс]. – Режим доступа: <http://www.orm.net> (дата обращения: 25.12.2018).

2. Laravel – экосистема, а не просто PHP-фреймворк [Электронный ресурс]. – Режим доступа: https://habr.com/post/334776/#lara_all/ (дата обращения: 25.12.2018).

3. Введение в ORM (Object Relational Mapping) [Электронный ресурс]. – Режим доступа: <http://internetka.in.ua/orm-intro/> (дата обращения: 25.12.2018).

4. Шаблон проектирования [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Шаблон_проектирования (дата обращения: 25.12.2018).