

## Содержание

Лабораторная работа №0. «Физическая модель данных MySQL» .....	2
Лабораторная работа №1. «Выборка данных из БД» .....	3
Лабораторная работа №3. «Расширение CRUD-функционала».....	6
Лабораторная работа №4. «Разработка административного модуля».....	7
Лабораторная работа №5. «Расширение функционала» .....	9

## Лабораторная работа №0. «Физическая модель данных MySQL»

### Общие правила выполнения заданий

#### Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- Для работы с БД рекомендуется использовать MySQL Workbench, phpMyAdmin, консоль сервера MySQL и др.

#### Ход выполнения работы:

1. Выполните проектирование физической модели данных (*используйте индивидуальный вариант из файла «Варианты\_Юнивер.docx»*).
2. Создайте необходимые отношения.
3. Укажите ограничения атрибутов при необходимости.
4. Реализуйте связи между таблицами с использованием PRIMARY и FOREIGN KEYS. Укажите корректные ограничения внешнего ключа.
5. Заполните таблицы данными.

#### Требования и рекомендации по выполнению:

- Описание предметной области очень детальное. При определении необходимых отношений ориентируйтесь на запросы, которые требуется выполнить: все необходимая для их выполнения информация должна храниться в базе данных.
- Структура базы данных, набор отношений и перечень атрибутов могут быть дополнены и скорректированы проектировщиком.
- Количество записей в таблицах должно быть достаточным для определения корректности выполнения запроса.
- БД может изменяться при выполнении последующих ЛР.

# Лабораторная работа №1. «Выборка данных из БД»

## Общие правила выполнения заданий

### Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- Для работы с БД рекомендуется использовать MySQL Workbench, phpMyAdmin, консоль сервера MySQL и др.
- В качестве фреймворка используется Yii2.

### Требования и рекомендации по выполнению:

- Структура таблиц может быть изменена для выполнения заданий (дополнена новыми атрибутами)
- С каждой таблицей базы данных ассоциирован свой контроллер с необходимым набором методов.
- Результаты вывода данных должны быть отображены в табличном формате.
- При выводе данных из дочерних таблиц внешние ключи должны быть заменены на соответствующие значения из родительских таблиц (**не должны выводиться id**). **Загрузка данных может быть жадной/ленивой.**
- Требования к интерфейсу не предъявляются.

## Задания к лабораторной работе №1

### Задание №1

1. Выполните установку Yii. Можете воспользоваться установкой как из Composer, так и из архива. Установите приложение типа basic.
2. Протестируйте, что приложение Yii успешно установлено.
3. Настройте шаблон для дальнейшего выполнения заданий, добавив в него ссылку на страницу с информацией об авторе (ФИО, группу).
4. Протестируйте работоспособность ссылок.

### Задание №2

1. Настройте подключение к базе данных
2. Создайте классы – потомки класса Active Record, пропишите методы связи (hasOne/hasMany) во **ВСЕХ** классах.
3. Выведите в табличном виде данные из одной таблицы БД (*согласуйте с преподавателем*).
4. Замените внешние ключи дочерней таблицы на значения из родительской таблицы.
5. Добавьте соответствующие ссылки в меню шаблона.

## Контрольные вопросы

1. В чем особенности жадной загрузки и ленивой загрузки данных?
2. Какой метод отвечает за подключение вида?
3. Какое пространство имен необходимо указывать при создании контроллера?

## Лабораторная работа №2. «Обработка пользовательских запросов»

### Общие правила выполнения заданий

#### Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- Для работы с БД рекомендуется использовать MySQL Workbench, phpMyAdmin, консоль сервера MySQL и др.
- В качестве фреймворка используется Yii2.

#### Требования и рекомендации по выполнению:

- Результаты запросов должны быть отображены в табличном формате.
- При наличии условий в запросе, необходимые параметры запроса вводятся в форму (не прописаны статично). Вводимые данные должны быть обязательно провалидированы.
- Также данные для запроса могут быть подгружены из базы данных (возможно использование *ArrayHelper* и метода *map* - <https://www.yiiframework.com/doc/guide/2.0/ru/helper-array> ).
- При выводе данных из дочерних таблиц внешние ключи должны быть заменены на соответствующие значения из родительских таблиц (**не должны выводиться id**)
- Для выполнения запросов вам могут понадобиться методы построителя запросов (<https://www.yiiframework.com/doc/guide/2.0/ru/db-query-builder>): *where*, *orderby*, *groupby*, *select*, *with*, *find*, *findone*.
- Вы можете выполнять каждый запрос отдельно, либо настраивать соответствующие фильтры. Количество представлений не ограничено.
- Используйте соответствующие контроллеры, ассоциированные с той или иной таблицей БД.
- Требования к интерфейсу не предъявляются.

### Задания к лабораторной работе №2

#### Задание №1

1. Настройте подключение к базе данных.
2. Создайте пользовательские формы для ввода данных для пользовательских запросов.
3. Для формирования динамических запросов используйте:
  - a. Выпадающий список с возможными вариантами условий запроса.
  - b. Поле для открытого ввода данных.
  - c. Радиокнопки/чекбоксы.
4. Добавьте правила валидации. Измените стандартные сообщения об ошибках.
5. Добавьте соответствующие ссылки в меню шаблона.

#### Задание №2

1. Выполните запросы согласно индивидуальному варианту задания (*используйте индивидуальный вариант из файла «Варианты\_Юпитер.docx»*).

*В заданиях приведен шаблонный вид запроса. Все запросы являются динамическими.*

2. Результат запросов отобразите в табличном виде.

### *Контрольные вопросы*

1. В каком файле выполняется подключение к базе данных?
2. Какой метод позволяет ограничить количество возвращаемых строк данных?
3. Какой метод отвечает за открытие тега form?
4. Всегда ли необходимо указывать правила валидации?

## Лабораторная работа №3. «Расширение CRUD-функционала»

### Общие правила выполнения заданий

#### Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- Для работы с БД рекомендуется использовать MySQL Workbench, phpMyAdmin, консоль сервера MySQL и др.
- В качестве фреймворка используется Yii2.

#### Требования и рекомендации по выполнению:

- При выводе (добавлении, удалении) данных из дочерних таблиц внешние ключи должны быть заменены на соответствующие значения из родительских таблиц (**не должны выводиться id**).
- Используйте соответствующий контроллер, ассоциированный с той или иной таблицей БД.
- Требования к интерфейсу не предъявляются.

### Задания к лабораторной работе №3

#### Задание №1

1. Для таблицы, выбранной в ЛР1, добавьте интерфейс:
  - добавления новой записи;
  - редактирования существующей записи;
  - удаления записи.
2. Добавьте соответствующие ссылки в шаблон.

### Контрольные вопросы

1. Какие методы отвечают за добавление данных в базу данных?
2. Как решить проблему удаления записей из родительской таблицы при наличии подчиненных таблиц?
3. Как указать перечень записей, которые необходимо обновить?

## Лабораторная работа №4. «Разработка административного модуля»

### Общие правила выполнения заданий

#### Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- Для работы с БД рекомендуется использовать MySQL Workbench, phpMyAdmin, консоль сервера MySQL и др.
- В качестве фреймворка используется Yii2.

#### Требования и рекомендации по выполнению:

- При настройке виджетов работы с данными используйте *ArrayHelper* и метод *map* - <https://www.yiiframework.com/doc/guide/2.0/ru/helper-array> )
- При выводе данных из дочерних таблиц внешние ключи должны быть заменены на соответствующие значения из родительских таблиц (**не должны выводиться id**).
- При выводе данных настройте пагинацию.
- Для авторизации используйте ACF.
- Требования к интерфейсу не предъявляются.

### Задания к лабораторной работе №3

#### Задание №1

1. Внесите изменения в БД для хранения информации о пользователях.
2. Создайте новый модуль для администратора через генератор Gii.
3. Добавьте возможность регистрации нового пользователя.
4. Выполните настройки приложения:
  - Возможность доступа осуществляется на основании данных, представленных в таблице базы данных.
  - Доступ в административную часть должен быть разрешен только авторизованным пользователям (ACF).
5. Добавьте соответствующие ссылки в шаблон.

#### Задание №2

1. Сгенерируйте в Gii для административного модуля CRUD-интерфейс для работы с таблицами базы данных.
2. Выполните настройки приложения:
  - На главной странице модуля сделайте навигацию (любой компонент Bootstrap) для перехода к работе с таблицами.
  - На всех страницах модуля добавьте элементы навигации и возможность возврата на главную страницу.
  - Измените сгенерированный код, чтобы кнопки, подписи формы и заголовки таблицы были на русском языке.
  - Измените внешние ключи на данные из связанных таблиц.

### *Контрольные вопросы*

1. Как активировать Gii в Yii-приложении?
2. Как выполнить генерацию класса Active Record?
3. Какой метод авторизации используется в приложениях с простым контролем доступа?
4. Какие роли предусмотрены в Yii-приложении?
5. Как получить хэш пароля?



## Лабораторная работа №5. «Расширение функционала»

### Общие правила выполнения заданий

#### Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- Для работы с БД рекомендуется использовать MySQL Workbench, phpMyAdmin, консоль сервера MySQL и др.
- В качестве фреймворка используется Yii2.

#### Требования и рекомендации по выполнению:

- Выполнение заданий этой лабораторной работы является необязательным.
- Выполненные задания позволяют повысить итоговый балл.
- Возможно выполнение любого задания (или всех).
- Согласуйте с преподавателем перечень необходимых заданий для выполнения.

### Задания к лабораторной работе №5

#### Задание №1 (Настройка шаблона приложения)

Подключите произвольный bootstrap-шаблон к приложению и выполните его настройку (работоспособность ссылок, отображение меню, фоновое изображение и др.). Итоговый вид приложения должен отличаться от базовой версии шаблона Yii-приложения (отсутствие Navbar, другая система навигации и т.д.). Шаблон должен работать во всех разделах (основном и административном).

##### *Методические рекомендации*

Внутри шаблона, у вас есть доступ к двум предопределенным переменным: `$this` и `$content`. Первая представляет собой вид компонент, как и в обычных видах, тогда как последняя содержит результат рендеринга вида, который рендерится при вызове метода `render()` в контроллерах.

Когда вы рендерите вид, вызывая метод `render()` из контроллера, к результату рендеринга будет применен шаблон. По умолчанию будет использован шаблон `@app/views/layouts/main.php`.

Вы можете использовать разные шаблоны, конфигурируя `yii\base\Application::$layout` или `yii\base\Controller::$layout`. Первый переопределяет шаблон, который используется по умолчанию всеми контроллерами, а второй переопределяет шаблон в отдельном контроллере.

#### Задание №2 (Создание виджета)

Создайте и подключите собственный виджет (например, виджет с меню, виджет с перечнем запросов, виджет с результатами одного из запросов). Виджет должен работать с базой данных. Для виджета обязательно наличие представления.

##### *Методические рекомендации*

Для того, чтобы создать виджет, следует унаследовать класс `yii\base\Widget` и переопределить методы `yii\base\Widget::init()` и/или `yii\base\Widget::run()`. Как правило,

метод `init()` должен содержать код, выполняющий нормализацию свойств виджета, а метод `run()` - код, возвращающий результат рендеринга виджета. Результат рендеринга может быть выведен непосредственно с помощью конструкции `"echo"` или же возвращен в строке методом `run()`.

### Задание №3 (Настройка ЧПУ)

Выполните настройку ЧПУ и роутинг:

- избавьтесь от папки `web` в `url`-адресе;
- адреса страниц должны быть ЧПУ `/category/subcategory/title`;
- на страницах, содержащих пагинацию, должны быть также ЧПУ.

#### *Методические рекомендации*

При обработке запрошенного URL, Yii приложение первым делом разбирает URL в маршрут. Полученный маршрут используется при создании соответствующего экземпляра действия контроллера для обработки запроса. Этот процесс называется роутинг.

Обратный роутингу процесс называется Создание URL, он отвечает за создание URL из заданного маршрута и соответствующих параметров запроса. При необходимости, созданный URL всегда может быть преобразован в первоначальные маршрут и параметры запроса.

В основе роутинга и создания URL лежит использование URL manager, зарегистрированного в качестве компонента приложения `urlManager`, который поддерживает два формата URL: Обычный и Человекопонятные URL.

### Задание №4 (Создание миграций)

1 часть:

- 1) Напишите миграцию для создания одной из таблиц проекта.
- 2) Напишите миграцию для добавления нового поля в таблицу.
- 3) Напишите миграцию для удаления одного из полей таблицы.

2 часть: любая произвольная миграция.

#### *Методические рекомендации*

В связи с тем, что изменение структуры базы данных часто требует изменение исходного кода, yii поддерживает так называемую возможность миграции баз данных, которая позволяет отслеживать изменения в базах данных при помощи терминов *миграции баз данных*, которые являются системой контроля версий вместе с исходным кодом.

Yii2 предоставляет хороший инструмент для работы с миграциями через консоль.

Для работы с БД, необходимо настроить подключение в файле `common/config/main.php`

Используйте методы `createTable()`, `addColumn()`, `dropColumn()`.

### Задание №5 (Настройка RBAC)

Настройте авторизацию через RBAC, добавив новые роли в систему (помимо авторизованного пользователя), например, администратор, менеджер, пользователь, гость и т.п.

## ***Методические рекомендации***

Yii реализует общую иерархическую RBAC, следуя NIST RBAC model. Обеспечивается функциональность RBAC через компонент приложения authManager.

Использование RBAC состоит из двух частей. Первая часть — это создание RBAC данных авторизации, и вторая часть — это использование данных авторизации для проверки доступа в том месте, где это нужно.

Роль представляет собой набор разрешений (permissions) (например, создание сообщения, обновление сообщения). Роль может быть назначена на одного или многих пользователей. Чтобы проверить, имеет ли пользователь указанные разрешения, мы должны проверить, назначена ли пользователю роль, которая содержит данное разрешение.

С каждой ролью или разрешением может быть связано правило (rule). Правило представляет собой кусок кода, который будет выполняться в ходе проверки доступа для определения может ли быть применена соответствующая роль или разрешение к текущему пользователю. Например, разрешение "обновление поста" может иметь правило, которое проверяет является ли текущий пользователь автором поста. Во время проверки доступа, если пользователь не является автором поста, он/она будет считаться не имеющими разрешения "обновление поста".

И роли, и разрешения могут быть организованы в иерархию. В частности, роль может содержать другие роли или разрешения; и разрешения могут содержать другие разрешения. Yii реализует частично упорядоченную иерархию, которая включает в себя специальные деревья иерархии. Роль может содержать разрешение, но обратное не верно.

Для настройки RBAC Manager используйте yii\rbac\DbManager. Выполните миграции. Для создания данных авторизации выполните: определение ролей и разрешений; установка отношений между ролями и правами доступа; определение правил; связывание правил с ролями и разрешениями; назначение ролей пользователям.

## ***Контрольные вопросы***

1. Можно ли подключить разные шаблоны к разным страницам (видам) приложения?
2. Какие методы необходимы переопределить в собственном виджете?
3. Что позволяет настроить компонент urlManager?
4. Что представляют собой провайдеры данных?
5. Какие методы необходимо описать в классе миграции?
6. Как связаны роли, разрешения и правила в RBAC?