

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**  
**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ**  
**УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**  
**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

Лабораторная работа № 4  
дисциплина: Технологии web-программирования  
тема: «Разработка и проектирование базы данных»

Выполнил: ст. гр. ПВ-42  
Богров Дмитрий Алексеевич  
Проверил: Картамышев С. В.

Белгород 2020

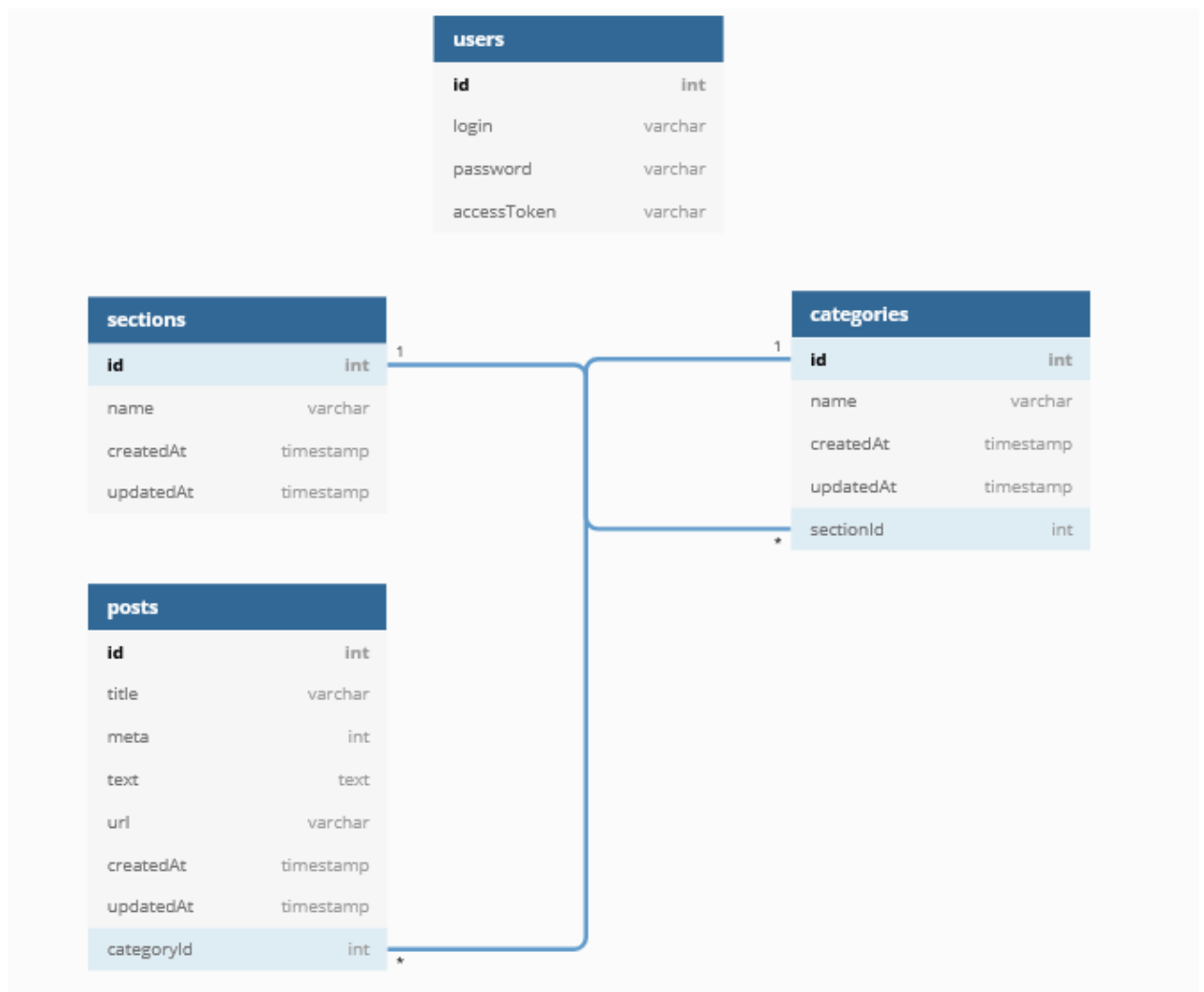
**Цель:** изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

### Задание к работе

1. Выбрать подходящую СУБД.
2. Изучить методы взаимодействия web-приложения с базой данных (ORM, Active Record).
3. Разработать структуру базы данных.
4. Разработать соответствующие модели в приложении.
5. В отчёт приложить схему базы данных, а также код одной из моделей (на своё усмотрение).

### Ход работы

Схема базы данных:



Для создания таблиц в базе данных использовались миграции.

*Create\_posts\_table.php:*

```
<?php

use yii\db\Migration;

/**
 * Handles the creation of table `{{%posts}}`.
 */
class m201120_154205_create_posts_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%posts}}', [
            'id' => $this->primaryKey(),
            'title' => $this->string(128)->notNull()->comment('Заголовок'),
            'meta' => $this->integer()->comment('Дата публикации'),
            'text' => $this->text()->comment('Текст статьи'),
            'url' => $this->string()->notNull()->comment('URL'),
            'createdAt' => $this->dateTime()->notNull()->comment('Дата
создания'),
            'updatedAt' => $this->dateTime()->comment('Дата изменения')
        ]);
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropTable('{{%posts}}');
    }
}
```

*Create\_categories\_table.php:*

```
<?php

use yii\db\Migration;

/**
 * Handles the creation of table `{{%categories}}`.
 */
class m201129_080808_create_categories_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%categories}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(64)->notNull()->comment('Наименование'),
            'createdAt' => $this->dateTime()->notNull()->comment('Дата
создания')
        ]);
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropTable('{{%categories}}');
    }
}
```

```

        создания'),
        'updatedAt' => $this->dateTime()->comment('Дата изменения')
    ]);

```

```

        $this->addColumn('{{%posts}}', 'categoryID',
            $this->integer()->comment('Категория'));
        $this->addForeignKey('fk_posts_categoryId', '{{%posts}}',
            'categoryId',
            '{{%categories}}', 'id', 'SET NULL');
    }

```

```

/**
 * {@inheritdoc}
 */
public function safeDown()
{
    $this->dropForeignKey('fk_posts_categoryId', '{{%posts}}');
    $this->dropColumn('{{%posts}}', 'categoryID');
    $this->dropTable('{{%categories}}');
}

```

Create\_sections\_table.php:

```
<?php
```

```
use yii\db\Migration;
```

```

/**
 * Handles the creation of table `{{%sections}}`.
 */
class m201129_081554_create_sections_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%sections}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(64)->notNull()->comment('Наименование'),
            'createdAt' => $this->dateTime()->notNull()->comment('Дата
        создания'),
            'updatedAt' => $this->dateTime()->comment('Дата изменения')
        ]);

        $this->addColumn('{{%categories}}', 'sectionId',
            $this->integer()->comment('Раздел'));
        $this->addForeignKey('fk_categories_sectionId', '{{%categories}}',
            'sectionId',
            '{{%sections}}', 'id', 'SET NULL');
    }

```

```

/**
 * {@inheritdoc}
 */
public function safeDown()
{
    $this->dropTable('{{%sections}}');
}

```

```
}
}
```

Create\_users\_table.php:

```
<?php
```

```
use yii\db\Migration;
```

```
/**
```

```
 * Handles the creation of table `{{%users}}`.

```

```
 */
```

```
class m201220_183408_create_users_table extends Migration
{
```

```
    /**
```

```
     * {@inheritdoc}

```

```
     */
```

```
    public function safeUp()
    {
```

```
        $this->createTable('{{%users}}', [
            'id' => $this->primaryKey(),

```

```
            'login' => $this->string(512)->notNull()->comment('Логин'),
            'password' => $this->string(512)->notNull()->comment('Пароль'),

```

```
            'accessToken' => $this->string(512)->comment('Токен
пользователя')

```

```
        ]);
    }
```

```
}
```

```
    /**
```

```
     * {@inheritdoc}

```

```
     */
```

```
    public function safeDown()
    {
```

```
        $this->dropTable('{{%users}}');
    }
```

```
}
```

В результате база данных стала выглядеть следующим образом:

Adminer 4.7.7 4.7.8

DB: bstu

SQL-запрос Импорт  
Экспорт Создать таблицу

выбрать categories  
выбрать migration  
выбрать posts  
выбрать sections  
выбрать users

Таблица: categories

Выбрать Показать структуру Изменить таблицу Новая запись

поле	Тип	Комментарий
id	int Автоматическое приращение	
name	varchar(64)	Наименование
createdAt	datetime	Дата создания
updatedAt	datetime NULL	Дата изменения
sectionId	int NULL	Раздел

Индексы

PRIMARY	id
INDEX	sectionId

Изменить индексы

Внешние ключи

Источник	Цель	При стирании	При обновлении	
sectionId	sections(id)	SET NULL	RESTRICT	Изменить

Добавить внешний ключ

Adminer 4.7.7 4.7.8

DB: bstu

[SQL-запрос](#)
[Импорт](#)
[Экспорт](#)
[Создать таблицу](#)

[выбрать categories](#)
[выбрать migration](#)
[выбрать \*\*posts\*\*](#)
[выбрать sections](#)
[выбрать users](#)

Таблица: posts

[Выбрать](#)
[Показать структуру](#)
[Изменить таблицу](#)
[Новая запись](#)

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>title</b>	varchar(128)	Заголовок
<b>meta</b>	int NULL	Дата публикации
<b>text</b>	text NULL	Текст статьи
<b>url</b>	varchar(255)	URL
<b>createdAt</b>	datetime	Дата создания
<b>updatedAt</b>	datetime NULL	Дата изменения
<b>categoryID</b>	int NULL	Категория

Индексы

<b>PRIMARY</b>	<i>id</i>
<b>INDEX</b>	<i>categoryID</i>

[Изменить индексы](#)

Внешние ключи

Источник	Цель	При стирании	При обновлении	
<b>categoryID</b>	categories( <i>id</i> )	SET NULL	RESTRICT	<a href="#">Изменить</a>

[Добавить внешний ключ](#)

Adminer 4.7.7 4.7.8

DB: bstu

[SQL-запрос](#)
[Импорт](#)
[Экспорт](#)
[Создать таблицу](#)

[выбрать categories](#)
[выбрать migration](#)
[выбрать posts](#)
[выбрать \*\*sections\*\*](#)
[выбрать users](#)

Таблица: sections

[Выбрать](#)
[Показать структуру](#)
[Изменить таблицу](#)
[Новая запись](#)

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>name</b>	varchar(64)	Наименование
<b>createdAt</b>	datetime	Дата создания
<b>updatedAt</b>	datetime NULL	Дата изменения

Индексы

<b>PRIMARY</b>	<i>id</i>
----------------	-----------

[Изменить индексы](#)

Adminer 4.7.7 4.7.8

DB: bstu

[SQL-запрос](#)
[Импорт](#)
[Экспорт](#)
[Создать таблицу](#)

[выбрать categories](#)
[выбрать migration](#)
[выбрать posts](#)
[выбрать sections](#)
[выбрать \*\*users\*\*](#)

Таблица: users

[Выбрать](#)
[Показать структуру](#)
[Изменить таблицу](#)
[Новая запись](#)

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>login</b>	varchar(512)	Логин
<b>password</b>	varchar(512)	Пароль
<b>accessToken</b>	varchar(512) NULL	Токен пользователя

Индексы

<b>PRIMARY</b>	<i>id</i>
----------------	-----------

[Изменить индексы](#)

С помощью генератора Yii/gii были сгенерированы модели для таблиц базы данных. Например, модель Категории:

```
<?php
```

```
namespace app\modules\v1\models;
```

```
use Yii;
```

```

/**
 * This is the model class for table "categories".
 *
 * @property int $id
 * @property string $name Наименование
 * @property string $createdAt Дата создания
 * @property string|null $updatedAt Дата изменения
 *
 * @property Post[] $posts
 * @property Section $section
 */
class Category extends BaseModel
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'categories';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['name'], 'required'],
            [['createdAt', 'updatedAt'], 'safe'],
            [['name'], 'string', 'max' => 64],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'name' => 'Наименование',
            'createdAt' => 'Дата создания',
            'updatedAt' => 'Дата изменения',
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function toArray(array $fields = [], array $expand = [],
$recursive = true)
    {
        return [
            'id' => $this->id,
            'name' => $this->name,
            'createdAt' => $this->createdAt,
            'section' => $this->section
        ];
    }
}

```

```
}

public function getSection()
{
    return $this->hasOne(Section::class, ['id' => 'sectionId']);
}

/**
 * @return \yii\db\ActiveQuery
 */
public function getPosts()
{
    return $this->hasMany(Post::class, ['categoryId' => 'id']);
}
}
```

**Вывод:** в результате выполнения данной работы были изучены основы взаимодействия web-приложения с базой данных, спроектирована база данных для хранения информации приложения (страницы, пользователи и т.п.).