МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ 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: студент группы ВТ-41
Шкорба В. С.
Проверил: Картамышев С.В.

Белгород 2020

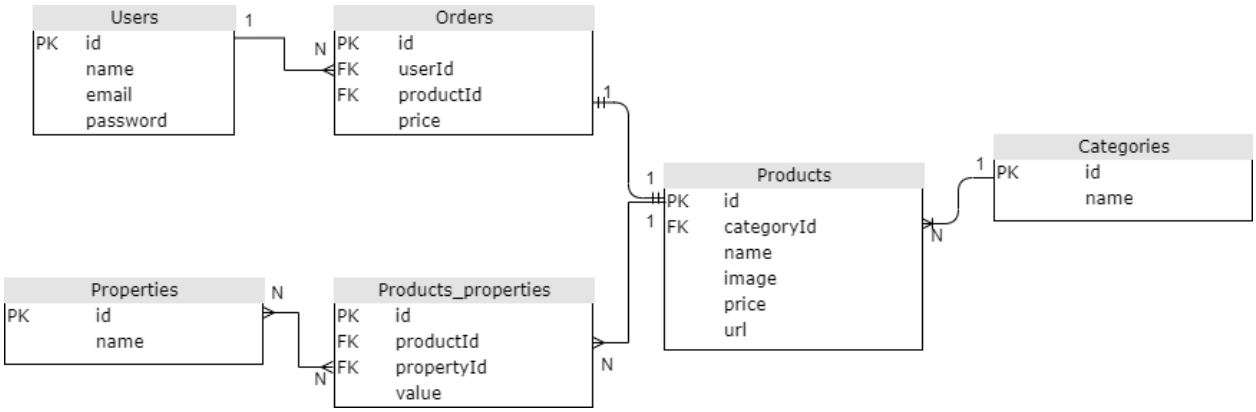
Цель: изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

Задание к лабораторной 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).

Выполнение

В ходе работы была спроектирована схема базы данных:



В качестве СУБД был выбран MySQL. Были созданы миграции для создания структуры базы данных:

```
PS D:\GitHub\web_labs\Amour> docker-compose run php yii migrate/create create_users_table
Creating amour_php_run ... done
usermod: no changes
Yii Migration Tool (based on Yii v2.0.39.3)

Create new migration '/app/migrations/m201229_125220_create_users_table.php'? (yes|no) [no]:yes
New migration created successfully.
PS D:\GitHub\web_labs\Amour>
```

```
PS D:\GitHub\web_labs\Amour> docker-compose run php yii migrate
Creating amour_php_run ... done
usermod: no changes
Yii Migration Tool (based on Yii v2.0.39.3)

Total 1 new migration to be applied:
        m201229_125220_create_users_table

Apply the above migration? (yes|no) [no]:yes
*** applying m201229_125220_create_users_table
    > create table {{%users}} ... done (time: 0.388s)
*** applied m201229_125220_create_users_table (time: 0.606s)


1 migration was applied.

Migrated up successfully.
PS D:\GitHub\web_labs\Amour>
```

Пример исходного кода миграции:

```php
<?php

use yii\db\Migration;
```

```php
/**
 * Class m201228_134726_create_products_properties
 */
class m201228_134726_create_products_properties extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%products_properties}}', [
            'id' => $this->primaryKey(),
            'productId' => $this->integer()->notNull()->comment('Товар'),
            'propertyId' => $this->integer()->notNull()->comment('Свойство'),
            'value' => $this->text()->notNull()->comment('Значение'),
            'createdAt' => $this->dateTime()->notNull()-
>comment('Дата создания'),
            'updatedAt' => $this->dateTime()->comment('Дата изменения')
        ]);

        $this-
>addForeignKey('fk_products_properties_productId', '{{%products_properties}}', 'p
roductId', '{{%products}}', 'id', 'CASCADE');
        $this-
>addForeignKey('fk_products_properties_propertyId', '{{%products_properties}}', '
propertyId', '{{%properties}}', 'id', 'CASCADE');
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this-
>dropForeignKey('fk_products_properties_productId', '{{%products_properties}}');
        $this-
>dropForeignKey('fk_products_properties_propertyId', '{{%products_properties}}');
        $this->dropTable('{{%products_properties}}');
    }
}
```

Посмотрим результат выполнения миграций с помощью Adminer:

Таблица: products

Выбрать **Показать структуру** Изменить таблицу Новая запись

поле	Тип	Комментарий
id	int *Автоматическое приращение*	
name	varchar(128)	Название
image	varchar(128) *NULL*	Изображение
price	decimal(12,2) *NULL* [**0.00**]	Цена
createdAt	datetime	Дата создания
updatedAt	datetime *NULL*	Дата изменения
url	varchar(128)	URL
categoryId	int *NULL*	Категория

Индексы

PRIMARY	*id*
INDEX	*categoryId*

Изменить индексы

Внешние ключи

Источник	Цель	При стирании	При обновлении	
categoryId	categories(*id*)	SET NULL	RESTRICT	Изменить

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

Индексы

PRIMARY	*id*

Таблица: properties

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

Индексы

PRIMARY	*id*

Таблица: products_properties

Выбрать **Показать структуру** Изменить таблицу Новая запись

поле	Тип	Комментарий
id	int *Автоматическое приращение*	
productId	int	Товар
propertyId	int	Свойство
value	text	Значение
createdAt	datetime	Дата создания
updatedAt	datetime *NULL*	Дата изменения

Индексы

PRIMARY	*id*
INDEX	*productId*
INDEX	*propertyId*

Изменить индексы

Внешние ключи

Источник	Цель	При стирании	При обновлении	
productId	products(*id*)	CASCADE	RESTRICT	Изменить
propertyId	properties(*id*)	CASCADE	RESTRICT	Изменить

Таблица: users

Выбрать **Показать структуру** Изменить таблицу Новая запись

поле	Тип	Комментарий
id	int *Автоматическое приращение*	
name	varchar(128)	Имя пользователя
email	varchar(100) *NULL*	Электронная почта
password	varchar(50)	Пароль
createdAt	datetime *NULL*	Дата создания
updatedAt	datetime *NULL*	Дата изменения

Индексы

PRIMARY	*id*

Таблица: orders

Выбрать **Показать структуру** Изменить таблицу Новая запись

поле	Тип	Комментарий
id	int *Автоматическое приращение*	
productId	int	Продукт
userId	int	Пользователь
price	float *NULL*	Сумма заказа
createdAt	datetime *NULL*	Дата создания
updatedAt	datetime *NULL*	Дата изменения

Индексы

PRIMARY	*id*
INDEX	*productId*
INDEX	*userId*

Изменить индексы

Внешние ключи

Источник	Цель	При стирании	При обновлении	
productId	products(*id*)	CASCADE	RESTRICT	Изменить
userId	users(*id*)	CASCADE	RESTRICT	Изменить

После этого с помощью утилиты Gii сгенерируем коды моделей и добавим в проект, например:

```php
<?php

namespace app\modules\v1\models;

use Yii;

/**
 * This is the model class for table "products_properties".
 *
 * @property int $id
 * @property int $productId Товар
 * @property int $propertyId Свойство
 * @property string $value Значение
 * @property string $createdAt Дата создания
 * @property string|null $updatedAt Дата изменения
 *
 * @property Product $product
 * @property Property $property
 */
class ProductProperty extends BaseModel
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'products_properties';
    }

    /**
     * {@inheritdoc}
     */
```

```php
    public function rules()
    {
        return [
            [['productId', 'propertyId', 'value'], 'required'],
            [['productId', 'propertyId'], 'integer'],
            [['value'], 'string'],
            [['createdAt', 'updatedAt'], 'safe'],
            [['productId'], 'exist', 'skipOnError' => true, 'targetClass' => Prod
uct::className(), 'targetAttribute' => ['productId' => 'id']],
            [['propertyId'], 'exist', 'skipOnError' => true, 'targetClass' => Pro
perty::className(), 'targetAttribute' => ['propertyId' => 'id']],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'productId' => 'Товар',
            'propertyId' => 'Свойство',
            'value' => 'Значение',
            'createdAt' => 'Дата создания',
            'updatedAt' => 'Дата изменения',
        ];
    }

    /**
     * Gets query for [[Product]].
     *
     * @return \yii\db\ActiveQuery
     */
    public function getProduct()
    {
        return $this->hasOne(Product::className(), ['id' => 'productId']);
    }

    /**
     * Gets query for [[Property]].
     *
     * @return \yii\db\ActiveQuery
     */
    public function getProperty()
    {
        return $this->hasOne(Property::className(), ['id' => 'propertyId']);
    }

    public function toArray(array $fields = [], array $expand = [], $recursive =
true)
    {
        return [
            'id' => $this->id,
            'value' => $this->value,
            'property' => $this->property
        ];
    }
}
```