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:**

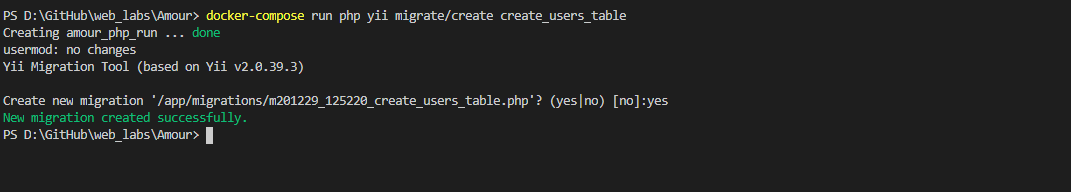
1. Выбрать подходящую СУБД.
2. Изучить методы взаимодействия web-приложения с базой данных (ORM, Active Record).
3. Разработать структуру базы данных.
4. Разработать соответствующие модели в приложении.
5. В отчёт приложить схему базы данных, а также код одной из моделей (на своё усмотрение).

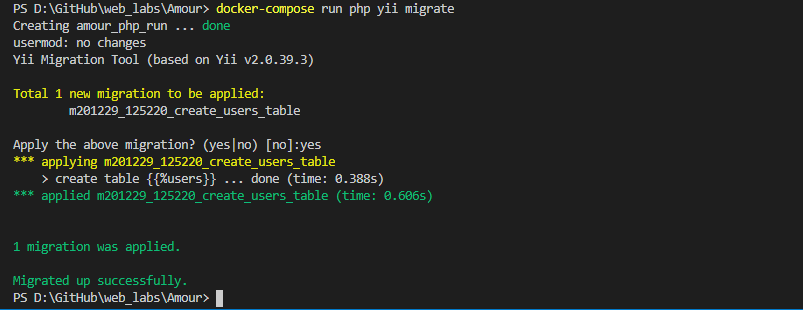
**Выполнение**

В ходе работы была спроектирована схема базы данных:



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





Пример исходного кода миграции:

<?php

use yii\db\Migration;

/\*\*

 \* 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}}', 'productId', '{{%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}}');

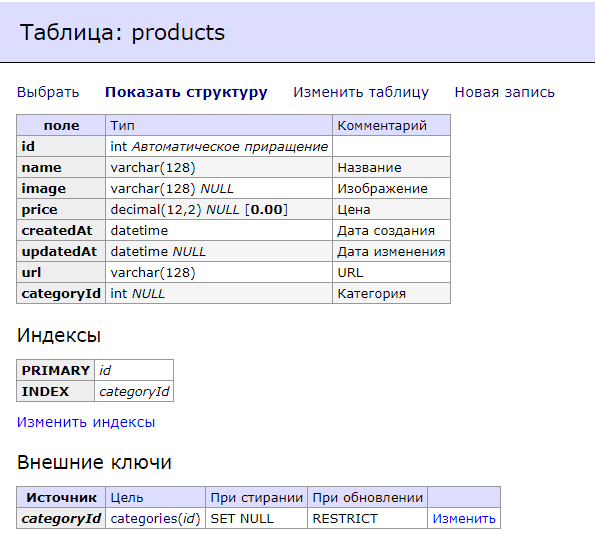
        $this->dropForeignKey('fk\_products\_properties\_propertyId', '{{%products\_properties}}');

        $this->dropTable('{{%products\_properties}}');

    }

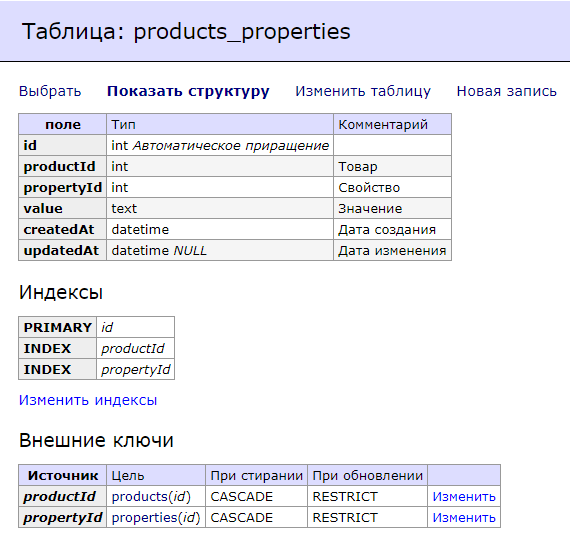
}

Посмотрим результат выполнения миграций с помощью Adminer:













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

<?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}

     \*/

    public function rules()

    {

        return [

            [['productId', 'propertyId', 'value'], 'required'],

            [['productId', 'propertyId'], 'integer'],

            [['value'], 'string'],

            [['createdAt', 'updatedAt'], 'safe'],

            [['productId'], 'exist', 'skipOnError' => true, 'targetClass' => Product::className(), 'targetAttribute' => ['productId' => 'id']],

            [['propertyId'], 'exist', 'skipOnError' => true, 'targetClass' => Property::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

        ];

    }

}