**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**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»

Выполнила: ст. группы ВТ-41  
Горохов В.О.  
Проверил: Картамышев С.В.

Белгород 2020

**Цель работы:** 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).

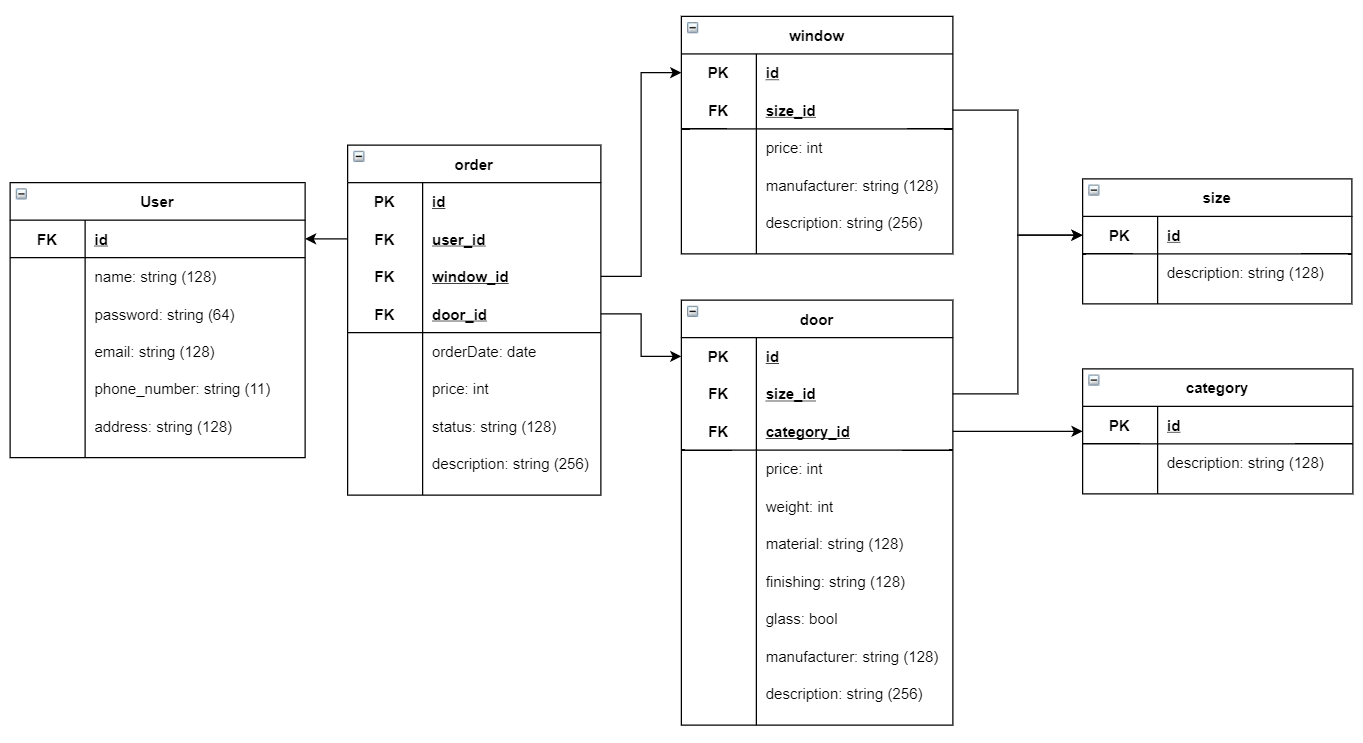
3. Разработать структуру базы данных.

4. Разработать соответствующие модели в приложении.

5. В отчёт приложить схему базы данных, а так же код одной из моделей (на своё усмотрение).

**Выполнение работы**

Для сайта “интернет-магазин дверей и окон” была разработана следующая структура базы данных:



В файле конфигурации db.php и файле docker-compose были прописаны следующие настройки базы данных:

<?php

return [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=mysql;dbname=bstu',

'username' => 'bstu',

'password' => 'bstu321!',

'charset' => 'utf8',

];

Были созданы миграции с помощью команды docker-compose run php yii migrate/create create\_....\_table

Миграция size:

<?php  
use yii\db\Migration;  
  
/\*\*  
 \* Handles the creation of table `{{%size}}`.  
 \*/  
class m201215\_180235\_create\_size\_table extends Migration {  
 public function safeUp () {  
 $this->createTable('{{%size}}', [  
 'id' => $this->primaryKey(),  
 'description' => $this->string(128)->comment('Описание'),  
 ]);  
 }  
  
 public function safeDown () {  
 $this->dropTable('{{%size}}');  
 }  
}

Миграция category:

<?php  
use yii\db\Migration;

class m201215\_181113\_create\_category\_table extends Migration {  
 public function safeUp () {  
 $this->createTable('{{%category}}', [  
 'id' => $this->primaryKey(),  
 'description' => $this->string(128)->comment('Описание'),  
 ]);  
 }  
  
 public function safeDown () {  
 $this->dropTable('{{%category}}');  
 }  
}

Миграция window:

<?php  
use yii\db\Migration;  
  
class m201215\_181321\_create\_window\_table extends Migration {  
 public function safeUp () {  
 $this->createTable('{{%window}}', [  
 'id' => $this->primaryKey(),  
 'size\_id' => $this->integer()->notNull()->comment('Размер'),  
 'price' => $this->integer()->comment('Цена'),  
 'manufacturer' => $this->string(128)->comment('Производитель'),  
 'description' => $this->string(256)->comment('Описание'),  
 ]);  
 $this->addForeignKey('fk-window-size\_id', '{{%window}}', 'size\_id', '{{%size}}', 'id', 'NO ACTION', 'NO ACTION');  
 }  
  
 public function safeDown () {  
 $this->dropForeignKey('fk-window-size\_id', '{{%window}}');  
 $this->dropTable('{{%window}}');  
 }  
}

Миграция door:

<?php  
use yii\db\Migration;

class m201215\_181557\_create\_door\_table extends Migration {  
 public function safeUp () {  
 $this->createTable('{{%door}}', [  
 'id' => $this->primaryKey(),  
 'size\_id' => $this->integer()->notNull()->comment('Размер'),  
 'category\_id' => $this->integer()->notNull()->comment('Категория'),  
 'price' => $this->integer()->comment('Цена'),  
 'weight' => $this->integer()->comment('Вес'),  
 'material' => $this->string(128)->comment('Материал'),  
 'finishing' => $this->string(128)->comment('Отделка'),  
 'glass' => $this->boolean()->comment('Стекло'),  
 'manufacturer' => $this->string(128)->comment('Производитель'),  
 'description' => $this->string(256)->comment('Описание'),  
 ]);  
 $this->addForeignKey('fk-door-size\_id', '{{%door}}', 'size\_id', '{{%size}}', 'id', 'NO ACTION', 'NO ACTION');  
 $this->addForeignKey('fk-door-category\_id', '{{%door}}', 'category\_id', '{{%category}}', 'id', 'NO ACTION', 'NO ACTION');  
 }

public function safeDown () {  
 $this->dropForeignKey('fk-door-size\_id', '{{%door}}');  
 $this->dropForeignKey('fk-door-category\_id', '{{%door}}');  
 $this->dropTable('{{%door}}');  
 }  
}

Миграция user:

<?php  
use yii\db\Migration;

class m201215\_183312\_create\_user\_table extends Migration {  
 public function safeUp () {  
 $this->createTable('{{%user}}', [  
 'id' => $this->primaryKey(),  
 'name' => $this->string(128)->notNull()->comment('Имя'),  
 'password' => $this->string(64)->notNull()->comment('Пароль'),  
 'email' => $this->string(128)->comment('Электронная почта'),  
 'telephone' => $this->string(11)->comment('Номер телефона'),  
 'address' => $this->string(128)->comment('Адрес'),  
  
 ]);  
 }  
 public function safeDown () {  
 $this->dropTable('{{%user}}');  
 }  
}

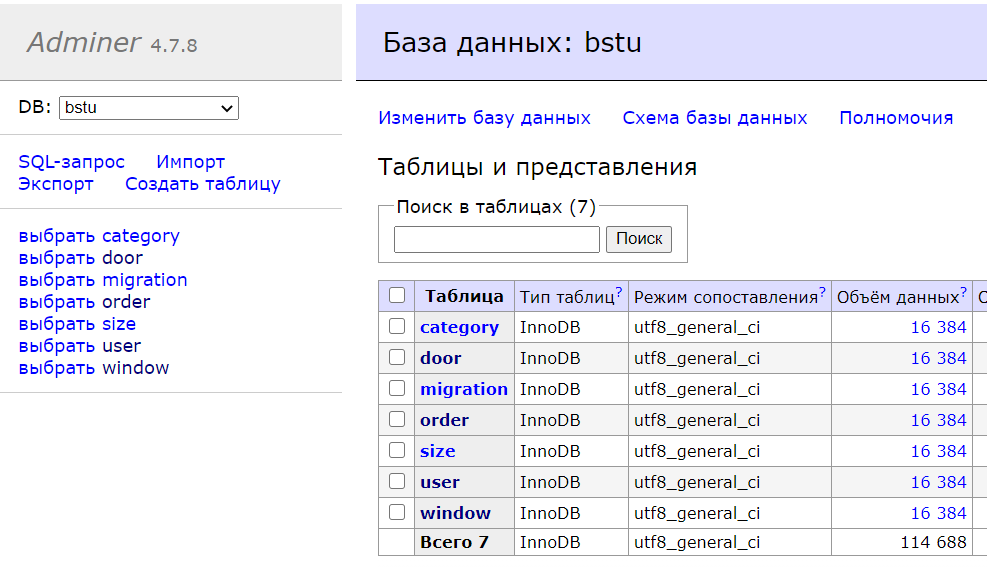
Миграция order:

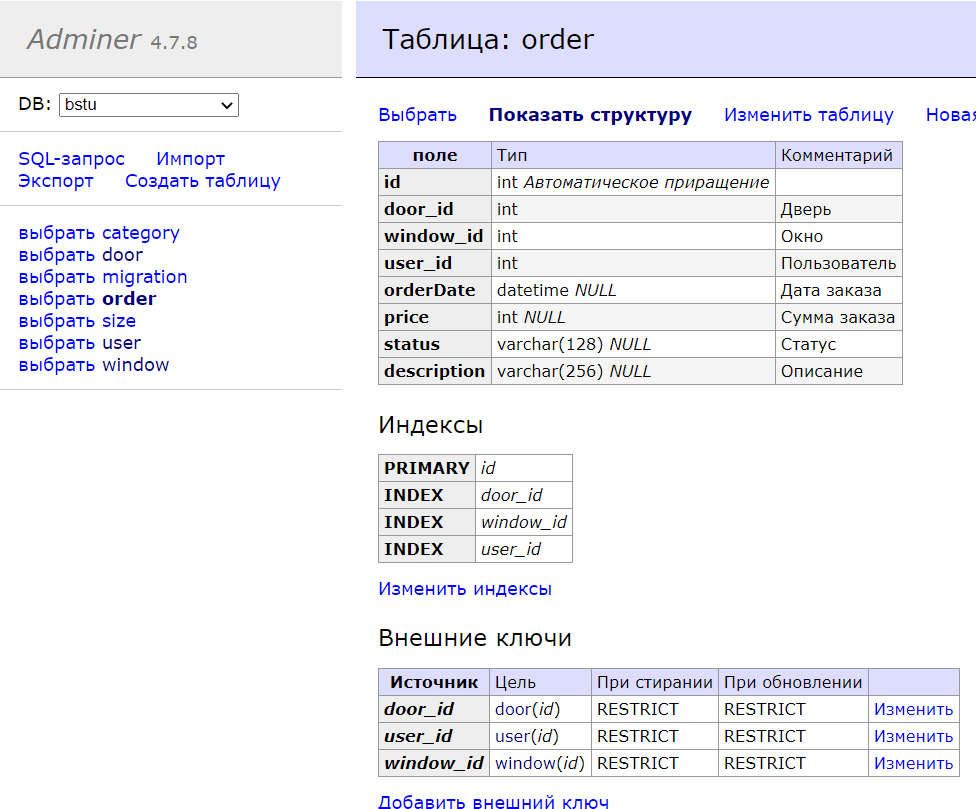
<?php  
use yii\db\Migration;

class m201215\_183800\_create\_order\_table extends Migration {  
 public function safeUp () {  
 $this->createTable('{{%order}}', [  
 'id' => $this->primaryKey(),  
 'door\_id' => $this->integer()->notNull()->comment('Дверь'),  
 'window\_id' => $this->integer()->notNull()->comment('Окно'),  
 'user\_id' => $this->integer()->notNull()->comment('Пользователь'),  
 'orderDate' => $this->dateTime()->comment('Дата заказа'),  
 'price' => $this->integer()->comment('Сумма заказа'),  
 'status' => $this->string(128)->comment('Статус'),  
 'description' => $this->string(256)->comment('Описание'),  
 ]);  
  
 $this->addForeignKey('fk-order-door\_id', '{{%order}}', 'door\_id', '{{%door}}', 'id', 'NO ACTION', 'NO ACTION');  
 $this->addForeignKey('fk-order-window\_id', '{{%order}}', 'window\_id', '{{%window}}', 'id', 'NO ACTION', 'NO ACTION');  
 $this->addForeignKey('fk-order-user\_id', '{{%order}}', 'user\_id', '{{%user}}', 'id', 'NO ACTION', 'NO ACTION');  
 }  
  
 public function safeDown () {  
 $this->dropForeignKey('fk-order-door\_id', '{{%order}}');  
 $this->dropForeignKey('fk-order-window\_id', '{{%order}}');  
 $this->dropForeignKey('fk-order-user\_id', '{{%order}}');  
 $this->dropTable('{{%order}}');  
 }  
}

Созданные миграции были занесены в БД командой: docker-compose run php yii migrate/up

Для проверки таблиц в БД, был использованн adminer





Далее с помощью генератора моделей yii code generator были сгенерированы соответствующие модели.

Для примера приведем код модели Product.php:

<?php  
  
namespace app\models;  
  
use Yii;  
  
/\*\*  
 \* This is the model class for table "order".  
 \*  
 \* @property int $id  
 \* @property int $door\_id Дверь  
 \* @property int $window\_id Окно  
 \* @property int $user\_id Пользователь  
 \* @property string|null $orderDate Дата заказа  
 \* @property int|null $price Сумма заказа  
 \* @property string|null $status Статус  
 \* @property string|null $description Описание  
 \*  
 \* @property Door $door  
 \* @property User $user  
 \* @property Window $window  
 \*/  
class Order extends \yii\db\ActiveRecord  
{  
 /\*\*  
 \* {@inheritdoc}  
 \*/  
 public static function tableName()  
 {  
 return 'order';  
 }  
  
 /\*\*  
 \* {@inheritdoc}  
 \*/  
 public function rules()  
 {  
 return [  
 [['door\_id', 'window\_id', 'user\_id'], 'required'],  
 [['door\_id', 'window\_id', 'user\_id', 'price'], 'integer'],  
 [['orderDate'], 'safe'],  
 [['status'], 'string', 'max' => 128],  
 [['description'], 'string', 'max' => 256],  
 [['door\_id'], 'exist', 'skipOnError' => true, 'targetClass' => Door::className(), 'targetAttribute' => ['door\_id' => 'id']],  
 [['user\_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::className(), 'targetAttribute' => ['user\_id' => 'id']],  
 [['window\_id'], 'exist', 'skipOnError' => true, 'targetClass' => Window::className(), 'targetAttribute' => ['window\_id' => 'id']],  
 ];  
 }  
  
 /\*\*  
 \* {@inheritdoc}  
 \*/  
 public function attributeLabels()  
 {  
 return [  
 'id' => 'ID',  
 'door\_id' => 'Дверь',  
 'window\_id' => 'Окно',  
 'user\_id' => 'Пользователь',  
 'orderDate' => 'Дата заказа',  
 'price' => 'Сумма заказа',  
 'status' => 'Статус',  
 'description' => 'Описание',  
 ];  
 }  
  
 /\*\*  
 \* Gets query for [[Door]].  
 \*  
 \* @return \yii\db\ActiveQuery  
 \*/  
 public function getDoor()  
 {  
 return $this->hasOne(Door::className(), ['id' => 'door\_id']);  
 }  
  
 /\*\*  
 \* Gets query for [[User]].  
 \*  
 \* @return \yii\db\ActiveQuery  
 \*/  
 public function getUser()  
 {  
 return $this->hasOne(User::className(), ['id' => 'user\_id']);  
 }  
  
 /\*\*  
 \* Gets query for [[Window]].  
 \*  
 \* @return \yii\db\ActiveQuery  
 \*/  
 public function getWindow()  
 {  
 return $this->hasOne(Window::className(), ['id' => 'window\_id']);  
 }  
}

**Вывод:** в результате выполнения лабораторной работы изучили основы взаимодействия web-приложения с базой данных, спроектировали базу данных для хранения информации приложения и сгенерировали соответствующие модели.