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

Технологии Web-программирования

Лабораторная работа №4

**Разработка и проектирование базы данных**

Выполнил: студент группы ВТ-41

Клесов М.И.

Проверил:

Картамышев С.В.

Белгород 2020

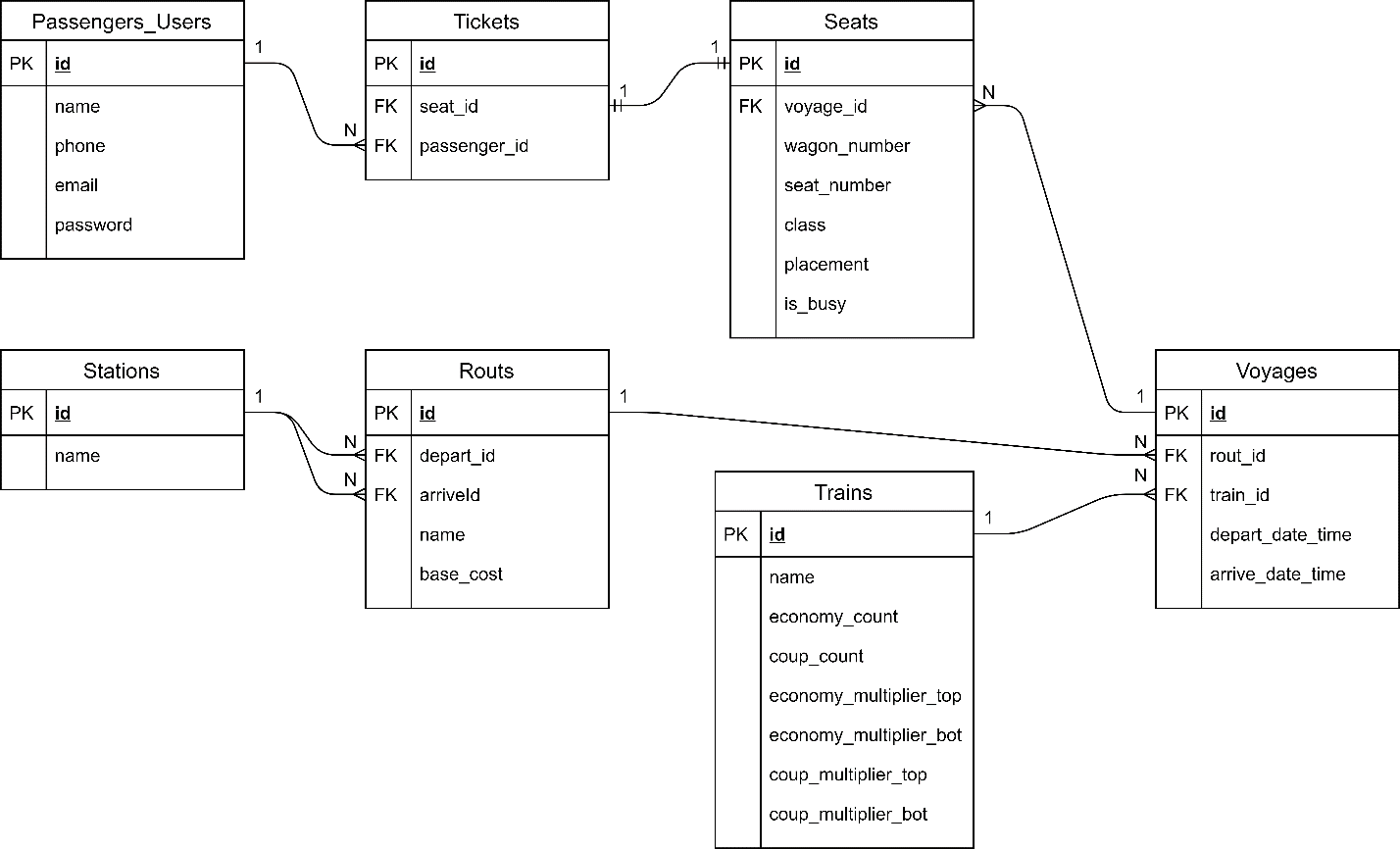
**Цель:** изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

**Задание к лабораторной работе:**

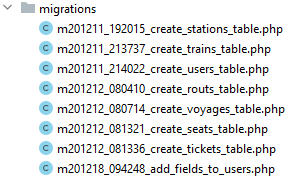
1. Развернуть базовое приложение Yii2 App Basic.
2. Настроить конфигурацию работы приложения с docker.
3. Добавить модуль для работы с API.
4. Добавить несколько контроллеров со статическими данными.
5. Продемонстрировать работу API в Postman.

**Выполнение**

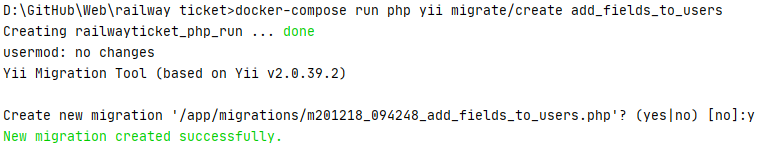
В ходе работы была спроектирована схема базы данных:



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



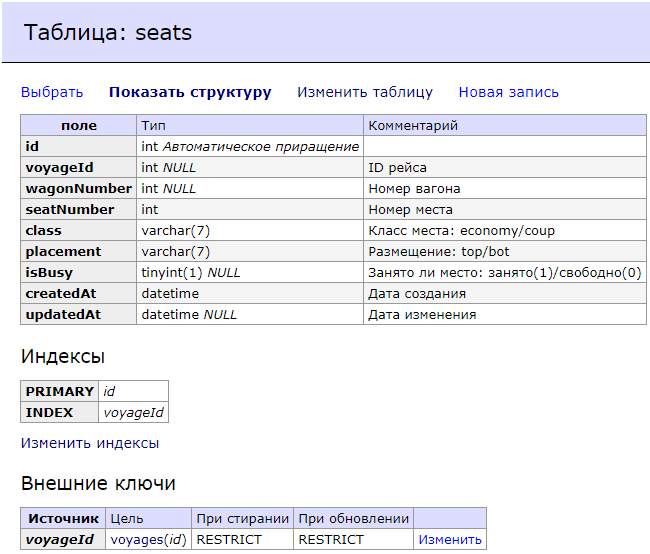
Пример создания миграции:

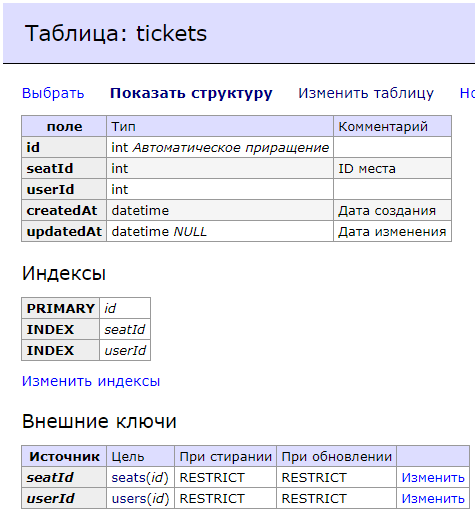
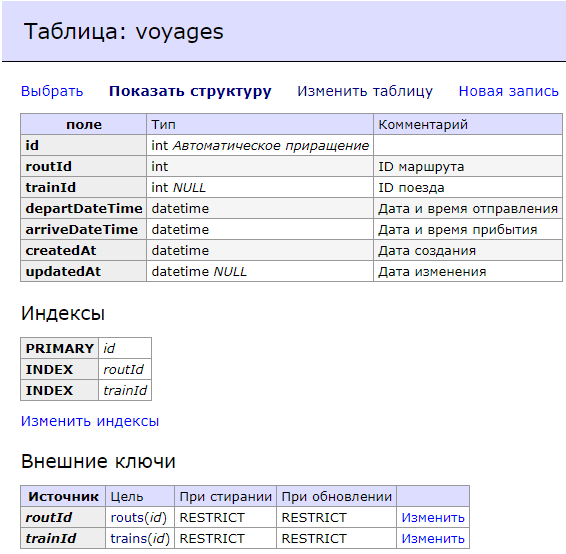


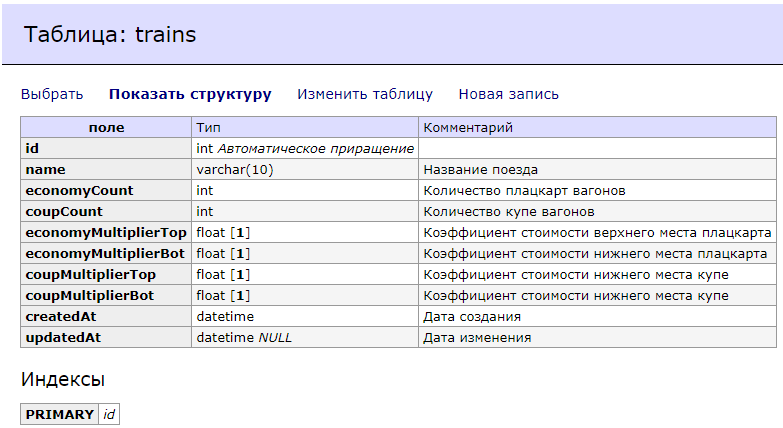
Пример исходного кода миграции:

<?php  
  
use yii\db\Migration;  
  
*/\*\*  
 \* Handles the creation of table `{{%voyages}}`.  
 \*/*class m201212\_080714\_create\_voyages\_table extends Migration  
{  
 */\*\*  
 \* Рейсы  
 \* {@inheritdoc}  
 \*/* public function safeUp()  
 {  
 $this->createTable('{{%voyages}}', [  
 'id' => $this->primaryKey(),  
 'routId' => $this->integer()->notNull()->comment('ID маршрута'),  
 'trainId' => $this->integer()->comment('ID поезда'),  
 'departDateTime' => $this->dateTime()->notNull()->comment('Дата и время отправления'),  
 'arriveDateTime' => $this->dateTime()->notNull()->comment('Дата и время прибытия'),  
 'createdAt' => $this->dateTime()->notNull()->comment('Дата создания'),  
 'updatedAt' => $this->dateTime()->comment('Дата изменения')  
 ]);  
 $this->addForeignKey('fk\_routId', '{{%voyages}}', 'routId', '{{%routs}}', 'id');  
 $this->addForeignKey('fk\_trainId', '{{%voyages}}', 'trainId', '{{%trains}}', 'id');  
 }  
  
 */\*\*  
 \* {@inheritdoc}  
 \*/* public function safeDown()  
 {  
 $this->dropForeignKey('fk\_routId', '{{%voyages}}');  
 $this->dropForeignKey('fk\_trainId', '{{%voyages}}');  
 $this->dropTable('{{%voyages}}');  
 }  
}

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 app\modules\v1\models\BaseModel;  
use Yii;  
  
*/\*\*  
 \* This is the model class for table "users".  
 \*  
 \* @property int $id  
 \* @property string|null $name Имя пользователя (пасажира)  
 \* @property string|null $phone Номер телефона пользователя (пасажира)  
 \* @property string $createdAt Дата создания  
 \* @property string|null $updatedAt Дата изменения  
 \* @property string|null $email email пользователя  
 \* @property string|null $password\_md5 hash код пароля пользователя  
 \*  
 \* @property Tickets[] $tickets Билеты пользователя  
 \*/*class User extends BaseModel  
{  
 */\*\*  
 \* {@inheritdoc}  
 \*/* public static function tableName()  
 {  
 return 'users';  
 }  
  
 */\*\*  
 \* {@inheritdoc}  
 \*/* public function rules()  
 {  
 return [  
 [['createdAt'], 'required'],  
 [['createdAt', 'updatedAt'], 'safe'],  
 [['name', 'phone', 'email'], 'string', 'max' => 128],  
 [['password\_md5'], 'string', 'max' => 32],  
 ];  
 }  
  
 */\*\*  
 \* {@inheritdoc}  
 \*/* public function attributeLabels()  
 {  
 return [  
 'id' => 'ID',  
 'name' => 'Имя пользователя (пасажира)',  
 'phone' => 'Номер телефона пользователя (пасажира)',  
 'createdAt' => 'Дата создания',  
 'updatedAt' => 'Дата изменения',  
 'email' => 'email пользователя',  
 'password\_md5' => 'hash код пароля пользователя'  
 ];  
 }  
  
 public function toArray(array $fields = [], array $expand = [], $recursive = true)  
 {  
 return [  
 'id' => $this->id,  
 'name' => $this->name,  
 'phone' => $this->phone,  
 'email' => $this->email,  
 'password\_md5' => $this->password\_md5  
 ];  
 }  
  
 */\*\*  
 \* Gets query for [[Tickets]].  
 \*  
 \* @return \yii\db\ActiveQuery  
 \*/* public function getTickets()  
 {  
 return $this->hasMany(Tickets::*className*(), ['userId' => 'id']);  
 }  
}