Государственное АВТОНОМНОЕ профессиональное образовательное учреждение

свердловской области

«Каменск-Уральский политехнический колледж»

Специальность 09.02.07 Информационные системы и программирование

Группа ВИСиП19-401

КУРСОВОЙ ПРОЕКТ

по МДК 09.01 ПРОЕКТИРОВАНИЕ И разработка

ВЕБ-ПРИЛОЖЕНИЙ

тема: Разработка информационной системы "Кинотеар"

Выполнил:

А.И. Карпов

Проверил:

И.М. Калмыкова

2022

Оглавление

[ВВЕДЕНИЕ 3](#_Toc122991918)

[1.АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc122991919)

[1.1Анализ предметной области 4](#_Toc122991920)

[1.2Постановка задачи 6](#_Toc122991921)

[2.ПРОЕКТНАЯ ЧАСТЬ 7](#_Toc122991922)

[2.1.Разработка информационной структуры веб-приложения 7](#_Toc122991923)

[2.2. Проектирование и реализация пользовательских форм веб-приложения 11](#_Toc122991924)

[ЗАКЛЮЧЕНИЕ 19](#_Toc122991925)

[СПИСОК ЛИТЕРАТУРЫ 20](#_Toc122991926)

ВВЕДЕНИЕ

Кинотеатр - общественное здание или его часть с оборудованием для публичной демонстрации кинофильмов.

Кинотеатр привлекает людей за счет большого экрана, когда видны мельчайшие детали фильма. Высокое качество изображения передает яркость красок каждого кадра. Благодаря хорошей акустике кинозала и правильно расположенным динамикам, звук становится объемным и воспринимается намного эффектнее. Нам нравится ходить в кинотеатр, ведь здесь можно первыми увидеть все новинки кино, причем на большом экране. Кроме того, в кинотеатр можно пойти вместе с друзьями, единомышленниками, родными и близкими людьми, тогда просмотр кино совместится с приятным совместным времяпровождением.

В прошлые времена, чтобы пойти в кино, надо было сходить в кинотеатр и купить билет не зная список фильмов. Это было затратно по времени и поэтому многие не могли себе позволить поход в кино. В дальнейшем начали появляться первые сайты, где показывался список фильмов, но всё ещё приходилось ходить в кино за покупкой билетов. Для дальнейшего упрощения появились первые web-приложения. В наше время, онлайн сервисы очень распространённые. Что бы купить билет, не нужно целенаправленно идти в кинотеатр и смотреть, есть ли места на интересующий фильм. Онлайн покупка билетов, уменьшает лишние траты времени и помогает быстро найти то, что вам нужно.

Цель данного курсового проекта – разработать веб-приложение, которое позволит сократить время для покупки билетов на киносеанс.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* + Провести анализ предметной области.
  + Спроектировать базу данных системы.
  + Реализовать базу данных.
  + Разработать запросы к базе данных.
  + Реализовать пользовательский интерфейс.
  + Провести тестирование приложения.

# 1.АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1Анализ предметной области

Кинотеатр означает большой поток людей из-за его распространённости, и возникает проблема с организацией продажи билетов. И для облегчения труда работников, и удобного оформления билета онлайн, поможет база данных.

База хранения данных сайта кинотеатра – необходимая вещь, для предоставления данных о предстоящих сеансах и занятых местах. Удобно применять между областной сети кинотеатров. Возможность в дальнейшем расширять базу, делает её особенно практичным. В работе кинотеатра особенно сложно распределять посадочные места. Удобство состоит в том, что мы можем в любой момент изменить фильм, если на него не было куплено ни одного билета. Добавление нового фильма, который вышел в прокат, тоже не составит особого труда.

Рассмотрим покупку билетов онлайн. Любой клиент, может купить билет онлайн, не выходя из дома или посмотреть предстоящие сеансы фильмов. Если по какой-либо причине, клиент не сможет прийти на фильм, он сможет вернуть билеты за 4 часа до самого сеанса.

В данной базе данных, мы облегчаем мониторинг занятых мест, и предстоящих сеансах.

Для более детального рассмотрения процесса учета мероприятий рассмотрим рисунок 1 представленный в виде диаграммы IDEF0, выполненный в программе Draw.io.

IDEF0 - Function Modeling - методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов.

Diagrams.net (ранее draw.io) – это бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript.

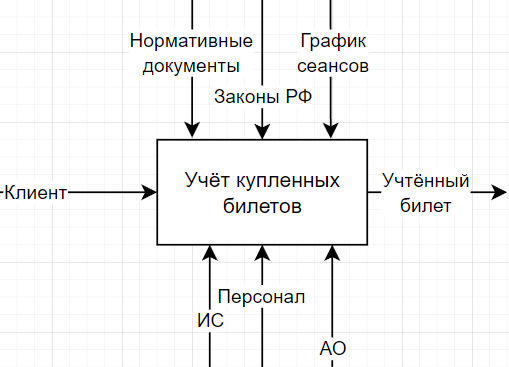


Рисунок 1 – функциональная диаграмма по учету купленных мест.

АО – аппаратное обеспечение.

ИС – информационная система.

Покупка билета происходит через любое устройство. Для того что бы купить билет нужно:

1. Зайти на сайт кинотеатра.
2. Найти интересующий фильм.
3. Посмотреть график сеансов и выбрать какой удобней.
4. Выбрать место, которое свободное и оптимальное для просмотра.
5. Оплатить билет онлайн из предложенного списка.
6. Сохранить на телефон билет, чтобы показать при в ходе в зал.

Когда выбрали интересующий фильм, и при нажатии на кнопку купить билет, происходит переадресация на выбор места.

## 1.2Постановка задачи

Основной задачей данной предметной области является автоматизация покупки билетов для похода в кинотеатр, для этого нужно разработать веб-приложение.

Данное приложение будет реализовывать следующий функционал:

- Выбор времени сеанса.

- Описание фильма.

- Покупка билета.

В данном web-приложении необходимо разработать:

* + Спроектировать базу данных системы.
  + Реализовать базу данных.
  + Разработать запросы к базе данных.
  + Реализовать пользовательский интерфейс.
  + Провести тестирование приложения.

# 2.ПРОЕКТНАЯ ЧАСТЬ

## 2.1.Разработка информационной структуры веб-приложения

Разработку базы данных было принято начать с построения ER-модели.

ER-модель - модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Все сущности имеют между собой связь один-ко-многим. ER-модель представлена на рисунке 2.

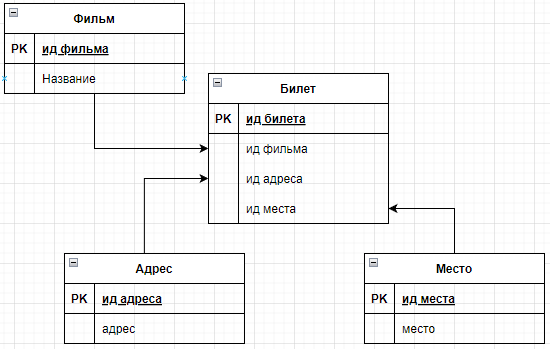


Рисунок 2. ER-модель БД.

Разработанная ER-модель состоит из 4 сущностей: билет, место, адрес, фильм.

Сущности: адрес, место, фильм созданы для подстановки в сущность заявки.

Сущности: адрес, место, фильм являются основными рабочими сущностями в БД. Каждая сущность необходима для своей задачи:

* Адрес, содержит в себе всю необходимую информацию об нахождении кинотеатра.
* Место, содержит информацию о местах рассадки в зале.
* Фильм, содержит полную информацию о фильме который будут показывать.

После проектирования ER-модели она была реализована в PhpMyAdmin. PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL.

Ниже приведены некоторые из ключевых преимуществ, которые предоставляет phpMyAdmin:

* Веб−интерфейс - Будучи веб-интерфейсом, пользовательский интерфейс phpMyAdmin доступен с помощью веб-браузера, и этот интерфейс доступен на всех платформах, где может работать веб-браузер.
* Графический интерфейс phpMyAdmin предоставляет графический интерфейс для запуска команд SQL и выполнения операций SQL и делает его довольно простым в использовании по сравнению с консольными редакторами sql.
* Интерфейс скрипта phpMyAdmin предоставляет интерфейс скрипта для запуска PHP-скриптов для подключения к базам данных и выполнения пользовательских операций.
* Многосерверный phpMyAdmin позволяет управлять несколькими серверами одновременно.
* Форматы резервного копирования phpMyAdmin позволяет создавать резервные копии баз данных в различных форматах, таких как XML, CSV, SQL, PDF, OpenDocument Text, Excel, Word, электронная таблица и т.д.
* Простой в использовании интерфейс phpMyAdmin позволяет легко создавать и запускать сложные запросы, создавать и редактировать функции, триггеры и т. д.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под

GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Преимущества MySQL:

* Легко использовать
* Предоставляет большой функционал
* Хорошие функции безопасности
* Легко масштабируется и подходит для больших баз данных
* Обеспечивает хорошую скорость и производительность
* Обеспечивает хорошее управление пользователями и множественный контроль доступа.

На рисунке 3 представлена физическая схема БД, реализованная на платформе PhpMyAdmin. На модели отражены названия атрибутов и их типы данных на сервере. Так же на модели отображены связи между сущностями для обеспечения целостности данных.

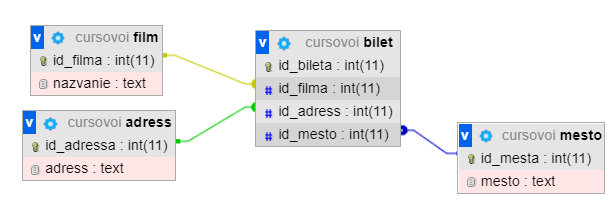


Рисунок 3 – физическая модель БД.

На рисунках 5, 6, 7, 8 – представлены заполненные данными таблицы.

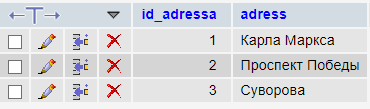


Рисунок 5 – таблица Адрес.

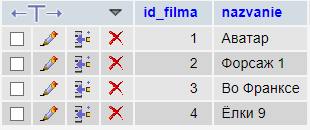


Рисунок 6 – таблица Фильмы.

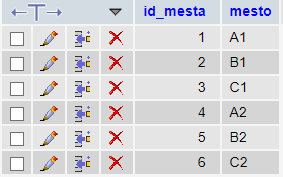


Рисунок 7 – таблица Места.

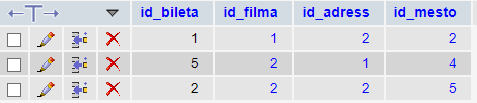


Таблица 8 – таблица Билет.

## 

## 2.2. Проектирование и реализация пользовательских форм веб-приложения

Для реализации пользовательских форм веб-приложения использовался инструмент Visual Studio Code.

Visual Studio Code (VS Code) — это редактор исходного кода. Его разработал Microsoft для всех популярных операционных систем: Windows, Linux и macOS.

Визуальный редактор кода позволяет:

* Работать с IntelliSense — автоматическим дописыванием функций при вводе первых букв.
* Выполнять отладку — искать и устранять ошибки в написанном коде.
* Удобно писать код — автоматически заполнять нужную информацию, подсвечивать элементы синтаксиса в зависимости от выбранного языка, расставлять нужные отступы.
* Контролировать версии кода, в том числе с помощью системы управления версиями Git.

Преимущества Visual Studio Code по сравнению с аналогичными программными продуктами это:

* Интеграция настроек из Sublime, Atom. После установки VS Code проверяет наличие других IDE. Если найдена Sublime или Atom VS Code предложит частично перенести настройки и быстрые клавиши в VS Code.
* Расширения. Удобная система добавления расширений. Достаточно перейти во вкладку с расширениями и написать в поиске необходимое дополнение.
* Git. Удобная работа с системой контроля версий. Работа с ней такая же, как и в Visual Studio 2019. Отображение коммитов, истории, веток и т.д.
* Поиск. Умный поиск с возможностью замены по разным критериям.
* Debug (Запуск приложений). Для веб проектов уже интегрирована отладка. Для других необходимо установить расширение. Отображение использующихся переменных при дебаге. Отображение ошибок (для некоторых языков нужно установить расширение).
* Интегрированный терминал. Можно работать через неё с Git и отлаживать приложение.
* Настройка. Большое количество настроек, как и в SublimeText и Atom. Применяются моментально, перезапускать IDE не требуется.
* Визуальное оформление. Можно кардинально изменить вид VS Code.

Также есть реализованные темы, которые применяются в один клик.

Так же был использован язык программирования PHP так как:

PHP — интерпретируемый скриптовый язык программирования общего назначения. Название представляет собой рекурсивный акроним PHP: Hypertext Preprocessor (PHP: предварительный обработчик гипертекста), но изначально оно расшифровывалось как Personal Home Page Tools (Инструменты для создания персональных веб-страниц).

Язык PHP обладает рядом неоспоримых преимуществ:

* Высокая скорость работы и, соответственно, общая производительность ресурсов
* Бюджетность, экономичность. Найти специалиста не представляется проблемой, стоимость написания программ на php не высока
* Простота освоения, простой синтаксис

Для создание скрипта был использован язык PHP.

C помощью него была реализована форма покупки билета представленная на рисунке 4.

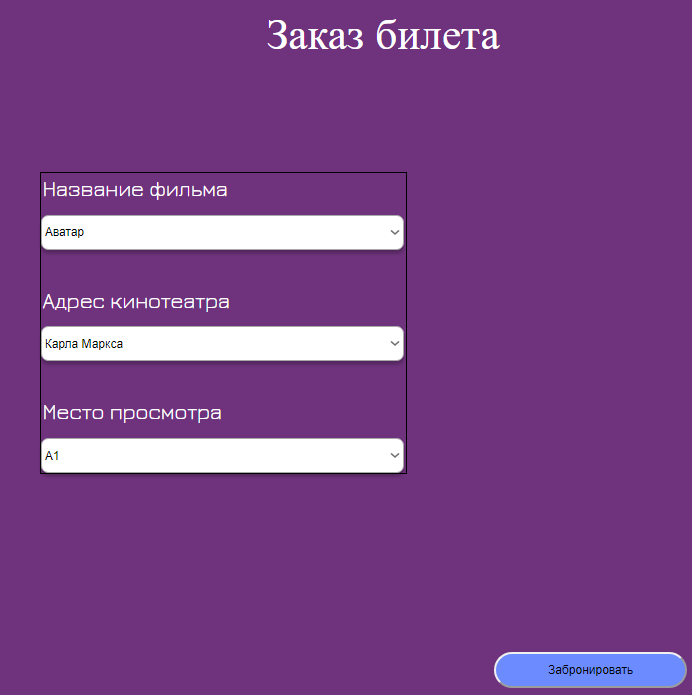


Рисунок 9 – Форма покупки билета.

Ниже представлен PHP скрипт для заполнения и отправки заявки в базу данных:

Заполнение страницы:

<body>

    <?php

    $link = mysqli\_connect('localhost', 'root', 'root', 'bronirovanie\_biletov');

Здесь происходит соединение с базой данных, где вводятся данные для авторизации в базе.

    if($link == false){

        print("Ошибка подключения");

        mysqli\_connect\_error();

    } else{

Выше происходит проверка авторизации в базе данных, если авторизоваться не смогли выдаёт нам ошибку. В другом случае, если авторизацию прошли, выполняет код который идёт ниже.

       ?> <div class="okno">

        <div class="okno\_name">Заказ билета</div>

        <div class="okno\_anketa">

            <form method="post" action="../script/script.php">

                <div class="okno\_name\_film">Название фильма</div>

                <select name="name\_film" class="okno\_name\_film\_vibor\_kino">

                    <?php

В данном скрипте мы находим таблицу film.

                        $query = "SELECT \* FROM `film`";

                        $result\_select = mysqli\_query($link, $query);

В этом цикле заполняем данными лист боксами, которые находятся на форме

                        while ($object = mysqli\_fetch\_object($result\_select)){

                            echo "<option value='$object->nazvanie'>$object->nazvanie</option>";}

                    ?>

                </select>

                <div class="okno\_name\_adress">Адрес кинотеатра</div>

                <select name="adress\_kino" class="okno\_name\_film\_vibor\_adressa">

Здесь мы находим таблицу adress.

                    <?php

                        $query = "SELECT \* FROM `adress`";

                        $result\_select = mysqli\_query($link, $query);

В этом цикле заполняем данными лист боксами, которые находятся на форме.

                        while ($object = mysqli\_fetch\_object($result\_select)){

                            echo "<option value='$object->adress'>$object->adress</option>";}

                    ?>

                </select>

                <div class="okno\_name\_mesto">Место просмотра</div>

                <select name="mesto" class="okno\_name\_film\_vibor\_mesto">

Здесь мы находим таблицу mesto.

<?php

                        $query = "SELECT \* FROM `mesto`";

                        $result\_select = mysqli\_query($link, $query);

В этом цикле заполняем данными лист боксами, которые находятся на форме

                        while ($object = mysqli\_fetch\_object($result\_select)){

                            echo "<option value='$object->mesto'>$object->mesto</option>";}

                    ?>

                </select>

            </div>

            <input class="btn\_zabron" name="btn" type="submit" value="Забронировать">

Добавляем кнопку на форму и задаём ей тип submit, что бы мы могли отправить данные с формы, в скрипт котором он будет обрабатываться и отправляться в базу данных.

            </form>

        </div>

    </div> <?php

    }

    ?>

</body>

Заполнение таблицы:

<?

$name\_film = $\_POST['name\_film'];

$adress\_kino = $\_POST['adress\_kino'];

$mesto = $\_POST['mesto'];

В данном коде мы берём данные из блока, и задаём имя переменной для дальнейшего его использования в коде.

$link = mysqli\_connect('localhost', 'root', 'root', ' bronirovanie\_biletov');

Здесь происходит соединение с базой данных, где вводятся данные для авторизации в базе.

if($link == false){

    print("Ошибка подключения!");

    mysqli\_connect\_error();

} else{

Выше происходит проверка авторизации в базе данных, если авторизоваться не смогли выдаёт нам ошибку. В другом случае, если авторизацию прошли, выполняет код который идёт ниже.

$query = "SELECT id\_filma FROM `film` WHERE nazvanie = '$name\_film'";

    $result\_select = mysqli\_query($link, $query);

Здесь мы находим таблицу в нашей базе данных, чтобы находить по названию фильма, id объекта.

    while ($object = mysqli\_fetch\_array($result\_select)){

        $key\_film = $object['id\_filma'];

    }

В этих двух строчках мы задаём переменной id, который принадлежит объекту, который нам нужен.

    $query = "SELECT id\_adressa FROM `adress` WHERE adress = '$adress\_kino'";

    $result\_select = mysqli\_query($link, $query);

Здесь мы находим таблицу в нашей базе данных, чтобы находить по названию адреса, id объекта.

    while ($object = mysqli\_fetch\_array($result\_select)){

        $key\_adressa = $object['id\_adressa'];

    }

В данном коде мы задаём переменной id найденного объекта в таблице.

    $query = "SELECT id\_mesta FROM `mesto` WHERE mesto = '$mesto'";

    $result\_select = mysqli\_query($link, $query);

Здесь мы находим таблицу в нашей базе данных, чтобы находить по названию места, id объекта.

    while ($object = mysqli\_fetch\_array($result\_select)){

        $key\_mesta = $object['id\_mesta'];

    }

В данном коде мы задаём переменной id найденного объекта в таблице.

    $sql = "INSERT INTO bilet (id\_filma , id\_adress , id\_mesto) VALUE ( '$key\_film' , '$key\_adressa' , '$key\_mesta' )";

В строчках выше мы находим таблицу bilet и объекты в таблице id\_filma, id\_adress, id\_mesto, заполняем данными из переменных key\_film, key\_adressa, key\_mesta. Учесть надо ещё тот факт, что заполнять таблицу мы можем только id объекта, названиями фильма они заполниться не могут.

    $result = mysqli\_query($link,$sql);

    if ($result == false){

        print("Ошибка при выполнении запроса фильма");

Здесь идёт проверка успешности заполнения данными, если не получилось заполнить таблицу то выводится "Ошибка при выполнении запроса фильма".

    } else

        print("Данные успешно отправлены.");

Если таблица успешно заполнена, то выводится сообщение «Данные успешно отправлены».

}

?>

Данный PHP скрипт считывает данные из формы где клиент выбирает нужные параметры, проводит подключение к базе данных bronirovanie\_biletov, если подключение не удалось то выводится ошибка о том что не удалось подключиться в базе данных.

Далее скрипт передает все данные в базу данных: Адрес кинотеатра, место посадки, и какой фильм. Если данные успешно отправлены на экран будет выведено сообщение «Данные успешно отправлены», если данные не будут добавлены на экран будет выведено сообщение «Ошибка при выполнении запроса».

# 

# ЗАКЛЮЧЕНИЕ

Цели поставленные в начале разработки были выполнены полностью,

веб-приложение имеет удобный и простой интерфейс, а также выполняет все необходимые действия.

Данное веб-приложение предназначенное для автоматизации процессов обработки данных, и хранения в БД, связанных с деятельностью компаний по продаже билетов на просмотр фильма.

БД позволяет вести занятые места в зале, по какому адресу кинотеатра, на какой фильм

В процессе работы над курсовым проектом были решены следующие задачи:

* Проведен анализ предметной области.
* Спроектирована и реализована базы данных веб-приложения.
* Спроектированы и реализованы пользовательские формы веб-приложения.

Целью данного веб-приложения является своевременное обеспечение надлежащих

людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области, при этом результатом функционирования информационных систем является информационная продукция -документы, информационные массивы, базы данных и информационные услуги, и снижение рутиной работы с бумагами.

Во время выполнения курсового проекта были использованы следующие программные продукты:

* Visual Studio Code
* PhpMyAdmin
* База данных MySQL.

Таким образом, используя разработанное веб-приложение можно забронировать билет в любое время суток, на любой интересующий вас фильм, в удобный для просмотра фильм кинотеатр.

# СПИСОК ЛИТЕРАТУРЫ

1. Гагарина Л.Г. Разработка и эксплуатация АИС. – М., Форум, 2007
2. Голицына, О.Л. и др. Базы данных; Форум; Инфра-М, 2013
3. Емельянова Н.З. Основы построения автоматизированных информационных систем. – М., Форум, 2007
4. Олифер В.Г. Компьютерные сети. Принципы, технологии. - СПб., Питер, 2009
5. Сеннов А. Access 2007/ Учебный курс. – СПб., Питер, 2007
6. [Баженова И. Ю.](http://www.knigafund.ru/authors/34341) SQL и процедурно-ориентированные языки. - Национальный Открытый Университет «ИНТУИТ», 2016 г.
7. Гвоздева В.А. Основы построения автоматизированных информационных систем. – М., Форум, 2009
8. Дунаев В..В. Базы данных Язык SQL. – СПб., БХВ - Петербург, 2006
9. Кушнир А.Н. Microsoft Office. Access 2007
10. Максимов Н.В., Попов И.И. Компьютерные сети.- М, ФОРУМ: ИНФРА-М, 2010
11. Малыхина М.П. Базы данных: основы, проектирования, использование. – СПб., БХВ-Петербург, 2006
12. Мельников В.П. Информационная безопасность. – М., Академия, 2009
13. [Лазицкас Е. А.](http://www.knigafund.ru/authors/51129), [Загумённикова И. Н.](http://www.knigafund.ru/authors/51130" \t "_blank), [Гилевский П. Г.](http://www.knigafund.ru/authors/51131" \t "_blank) Базы данных и системы управления базами данных: учебное пособие. – РИПО, 2016 г.
14. Рудикова Л.В. Базы данных. Разработка приложений. – СПб., БХВ - Петербург, 2006
15. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 - 6-е издание, 2012
16. [Шумаков П. В.](http://www.knigafund.ru/authors/45994) ADO.NET и создание приложений баз данных в среде Microsoft Visual Studio.NET: руководство разработчика с примерами на C#. - Диалог-МИФИ 2003 г.