МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

кафедра «Вычислительная техника».

**Лабораторная работа №1**

**тема: «Использование библиотеки компилируемых шаблонов в Web-приложениях»**

по дисциплине: «Корпоративные системы»

Выполнил:

студент 4 курса, гр. ИВТВМбд-41

Захарычев Никита Алексеевич.

Проверил:

доцент кафедры ВТ,

кандидат технических наук

Мартынов Антон Иванович.

г. Ульяновск, 2018

**Задание**

1. Спроектировать БД согласно варианту задания и реализовать ее для СУБД mySQL версии 5.x.x. Под проектированием понимается разработка ER-диаграммы и нормализация БД. Для реализации необходимо подготовить скрипт на зыке SQL (диалект mySQL  5.x.x), выполняющий следующие действия:

◦ Создание структуры таблиц;

◦ Установка ссылочной целостности для таблиц типа «мастер-подчиненный»;

◦ Заполнение таблиц данными (не менее 10 строк для каждой таблицы);

2. Создать приложение на языке PHP, позволяющее устанавливать соединение с БД mySQL и обращаться к ней с запросами SQL. Приложение должно иметь средства навигации, с помощью которых можно выбирать просмотр одной из возможных таблиц (согласно варианту задания).

3. Подключить к приложению на PHP модуль библиотеки Smarty.

4. Изучить синтаксис шаблонов библиотеки Smarty и реализовать шаблон (или набор шаблонов) для отображения содержимого таблиц, полученных в результате выполнения первого этапа. Шаблон вывода должен обеспечивать следующую функциональность:

◦ Вывод названия таблицы, которая в данный момент отображается на экране;

◦ Вывод строки информатора, которая показывает количество найденных записей, например «По Вашему запросу найдено 25 записи(ей)»

◦ Вывод «шапки таблицы» с указанием имени столбцов (оформление шапки таблицы должно отличаться от оформления основного содержания таблицы)

◦ Построчный вывод содержимого таблицы согласно следующим правилам:

a) Первое поле представляет собой итератор (последовательный номер записи, а не значение идентификатора из таблицы)

b) Поля, содержащие текст выравниваются по левому краю

c) Поля, содержащие числовые значения выравниваются по центру

d) Поля даты и времени форматируются «ДД-ММ-ГГГГ» и «ЧЧ:ММ:СС» соответственно

e) Если таблица содержит данные, которые ссылаются на другую таблицу, то необходимо выводить не номер ссылки, а содержимое поля, на которое идет ссылка.

f) Поля, содержащие графическую информацию, в режиме просмотра таблиц не выводятся.

◦ Фон для четных и нечетных строк таблицы различается.

◦ При наведении курсора мыши на строку таблицы ее фон меняется на цвет отличный от фона четных и нечетных строк (подсветка текущей строки).

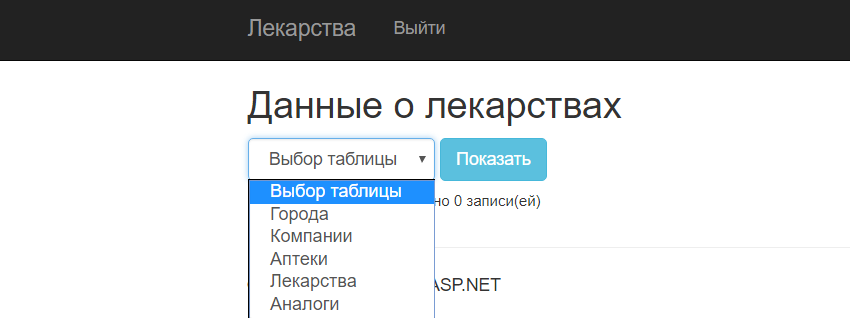
**Вариант 5**

5. Спроектируйте БД, содержащую учет лекарств в аптеке. БД должна содержать следующую информацию:

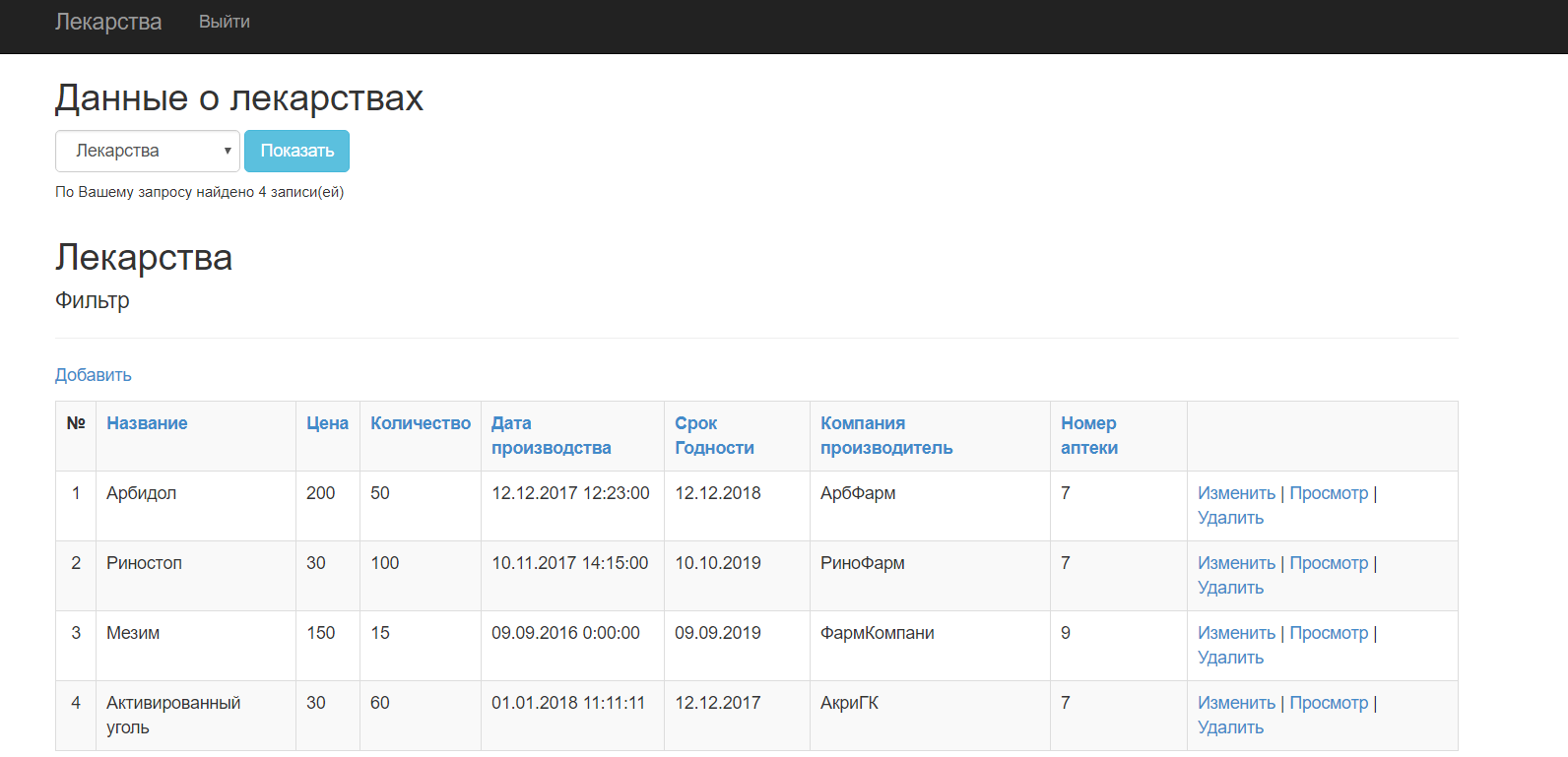
* Справочник «Города России»
* Справочник «Фармацевтические компании» с указанием названия компании и привязкой к конкретному городу.
* Информация об аптеке содержит — номер аптеки, адрес, время начала работы аптеки, время окончания работы аптеки, время обеденного перерыва, график работы по дням, схема проезда к аптеке.
* Информация о лекарстве содержит поля - наименование лекарства, стоимость одной единицы, количество единиц, дата изготовления, срок годности, компания-производитель, номер аптеки, а также список препаратов, которые могут это лекарство заменять.

**Ход работы**

В ходе работы была произведена нормализация Базы Данных. В итоге БД содержит 5 таблиц.



Основная таблица Лекарства.



**Листинг**

Шаблонизированное представление таблицы Лекарства.

@model PagedList.IPagedList<AptekaApteka.Models.Pharmacy>

@using PagedList.Mvc;

@using AptekaApteka.Models;

@using AptekaApteka.Helpers;

<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />

@{

ViewBag.Title = "Index";

}

<h2>Аптеки</h2>

@{

string modelName = ViewBag.ModelName;

int count = (Model.PageNumber \* Model.PageSize) - Model.PageSize + 1;

var addres = ViewBag.ViewPharmacy != null ? ViewBag.ViewPharmacy.Addres != null ? ViewBag.ViewPharmacy.Addres : "" : "";

var startwork = ViewBag.ViewPharmacy != null ? ViewBag.ViewPharmacy.StartWork != null ? ViewBag.ViewPharmacy.StartWork : "" : "";

var endwork = ViewBag.ViewPharmacy != null ? ViewBag.ViewPharmacy.EndWork != null ? ViewBag.ViewPharmacy.EndWork : "" : "";

var lunchtime = ViewBag.ViewPharmacy != null ? ViewBag.ViewPharmacy.LunchTime != null ? ViewBag.ViewPharmacy.LunchTime : "" : "";

var schedule = ViewBag.ViewPharmacy != null ? ViewBag.ViewPharmacy.Schedule != null ? ViewBag.ViewPharmacy.Schedule : "" : "";

var pageSize = ViewBag.ViewPharmacy != null ? ViewBag.ViewPharmacy.PageSize != 0 ? ViewBag.ViewPharmacy.PageSize : "" : "";

}

<div>

@using (Html.BeginForm("Index", "Home", FormMethod.Get))

{

<h4 style="cursor: pointer; width: 50px;" data-toggle="collapse" data-target="#demo">Фильтр</h4>

<div class="form-horizontal collapse" id="demo">

<hr />

<div class="form-group">

<label class="control-label col-md-2">Адрес</label>

<div class="col-md-10">

<input type="text" name="Addres" class="form-control" value=@(addres)>

</div>

</div>

<div class="form-group">

<label class="control-label col-md-2">Начало работы</label>

<div class="col-md-10">

<input type="number" name="StartWork" class="form-control" value=@(startwork) />

</div>

</div>

<div class="form-group">

<label class="control-label col-md-2">Окончание работы</label>

<div class="col-md-10">

<input type="number" name="EndWork" class="form-control" value=@(endwork) />

</div>

</div>

<div class="form-group">

<label class="control-label col-md-2">Обеденный перерыв</label>

<div class="col-md-10">

<input type="number" name="LunchTime" class="form-control" value=@(lunchtime) />

</div>

</div>

<div class="form-group">

<label class="control-label col-md-2">График работы</label>

<div class="col-md-10">

<input type="number" name="Schedule" class="form-control" value=@(schedule) />

</div>

</div>

<div class="form-group">

<label class="control-label col-md-2">Компания</label>

<div class="col-md-10">

@Html.DropDownList("Company",

new SelectList(ViewBag.CompanyName),

"Все",

new { @class = "form-control " })

</div>

</div>

<div class="form-group">

<label class="control-label col-md-2">Количество выдаваемых строк</label>

<div class="col-md-10">

<input type="number" name="PageSize" class="form-control" value=@(pageSize) />

</div>

</div>

<div class="form-group form-horizontal">

<input type="hidden" name="table" class="form-control" value=@(modelName) />

<input type="submit" style="margin-left: 25%" class="btn btn-info form-control-static" value="Фильтровать" />

</div>

</div>

<hr />

}

</div>

@if (Session["Admin"] != null)

{

<p>

@Html.ActionLink("Добавить", "Create", "Pharmacies", null, null)

</p>

}

<table class="table table-bordered table-hover table-responsive table-striped">

<tr>

<th class="headerTableCellColor">№</th>

<th class="headerTableCellColor">

@SortLinkHelper.SortLink(Html.ActionLink("Адрес", "Index", "Home", new

{

sortBy = "adress",

pageSize = Model.PageSize,

pharmaciesRequest = ViewBag.ModelList,

table = ViewBag.ModelName,

sortDirect = !ViewBag.SortDir

}, new { @class = "headerTableCellColor" }), ViewBag.SortBy, ViewBag.SortDir)

<!--Html.DisplayNameFor(model => model.Name)-->

</th>

<th class="headerTableCellColor">

@SortLinkHelper.SortLink(Html.ActionLink("Начало работы", "Index", "Home", new

{

sortBy = "startwork",

pageSize = Model.PageSize,

pharmaciesRequest = ViewBag.ModelList,

table = ViewBag.ModelName,

sortDirect = !ViewBag.SortDir

}, new { @class = "headerTableCellColor" }), ViewBag.SortBy, ViewBag.SortDir)

<!--Html.DisplayNameFor(model => model.SinceYear)-->

</th>

<th class="headerTableCellColor">

@SortLinkHelper.SortLink(Html.ActionLink("Окончание работы", "Index", "Home", new

{

sortBy = "endwork",

pageSize = Model.PageSize,

pharmaciesRequest = ViewBag.ModelList,

table = ViewBag.ModelName,

sortDirect = !ViewBag.SortDir

}, new { @class = "headerTableCellColor" }), ViewBag.SortBy, ViewBag.SortDir)

<!--Html.DisplayNameFor(model => model.SinceYear)-->

</th>

<th class="headerTableCellColor">

@SortLinkHelper.SortLink(Html.ActionLink("Обеденный перерыв", "Index", "Home", new

{

sortBy = "launchtime",

pageSize = Model.PageSize,

pharmaciesRequest = ViewBag.ModelList,

table = ViewBag.ModelName,

sortDirect = !ViewBag.SortDir

}, new { @class = "headerTableCellColor" }), ViewBag.SortBy, ViewBag.SortDir)

<!--Html.DisplayNameFor(model => model.SinceYear)-->

</th>

<th class="headerTableCellColor">

@SortLinkHelper.SortLink(Html.ActionLink("График работы", "Index", "Home", new

{

sortBy = "schedule",

pageSize = Model.PageSize,

pharmaciesRequest = ViewBag.ModelList,

table = ViewBag.ModelName,

sortDirect = !ViewBag.SortDir

}, new { @class = "headerTableCellColor" }), ViewBag.SortBy, ViewBag.SortDir)

<!--Html.DisplayNameFor(model => model.SinceYear)-->

</th>

<th class="headerTableCellColor">

@SortLinkHelper.SortLink(Html.ActionLink("Компания", "Index", "Home", new

{

sortBy = "companyname",

pageSize = Model.PageSize,

pharmaciesRequest = ViewBag.ModelList,

table = ViewBag.ModelName,

sortDirect = !ViewBag.SortDir

}, new { @class = "headerTableCellColor" }), ViewBag.SortBy, ViewBag.SortDir)

<!--Html.DisplayNameFor(model => model.SinceYear)-->

</th>

@if (Session["Admin"] != null)

{

<th class="headerTableCellColor"></th>

}

</tr>

@foreach (var item in Model)

{

<tr>

<td style="text-align:center">@(count++)</td>

<td>

@Html.DisplayFor(modelItem => item.Addres)

</td>

<td>

@Html.DisplayFor(modelItem => item.StartWork)

</td>

<td>

@Html.DisplayFor(modelItem => item.EndWork)

</td>

<td>

@Html.DisplayFor(modelItem => item.LunchTime)

</td>

<td>

@Html.DisplayFor(modelItem => item.Schedule)

</td>

<td>

@Html.DisplayFor(modelItem => item.Company.Name)

</td>

@if (Session["Admin"] != null)

{

<td>

@Html.ActionLink("Изменить", "Edit", "Pharmacies", new { id = item.PharmacyID }, null) |

@Html.ActionLink("Просмотр", "Details", "Pharmacies", new { id = item.PharmacyID }, null) |

@Html.ActionLink("Удалить", "Delete", "Pharmacies", new { id = item.PharmacyID }, null)

</td>

}

</tr>

}

</table>

<br />

Страница @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) из @Model.PageCount

<div style="width:300px; display:block; margin-left:auto; margin-right:auto">

@Html.PagedListPager(Model, page => Url.Action("Index", new { table = ViewBag.ModelName, page, pageSize = Model.PageSize, pharmaciesRequest = ViewBag.ModelList }))

</div>

Контроллер Главной страницы.

using AptekaApteka.DAL;

using AptekaApteka.Models;

using AptekaApteka.Models.ViewModels;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace AptekaApteka.Controllers

{

public class HomeController : Controller

{

static List<City> cities;

static List<Company> companies;

static List<Pharmacy> pharmacies;

static List<Preparation> preparations;

static List<Analog> analogs;

private AptekaContext db = new AptekaContext();

public ActionResult Index(string table,

int? page,

ViewCity viewCity,

ViewPharmacy viewPharmacy,

ViewCompany viewCompany,

ViewPreparation viewPreparation,

ViewAnalog viewAnalog,

List<ViewCity> citiesRequest,

List<ViewCompany> companiesRequest,

List<ViewPharmacy> pharmaciesRequest,

List<ViewPreparation> preparationsRequest,

List<ViewAnalog> analogsRequest,

int? pageSize,

string sortBy = "",

bool sortDirect = true

)

{

var tableNameList = new List<string>();

tableNameList.Add("Города");

tableNameList.Add("Компании");

tableNameList.Add("Аптеки");

tableNameList.Add("Лекарства");

tableNameList.Add("Аналоги");

if (("Города").Equals(table))

{

Services.pageSize = viewCity.PageSize != 0 ? viewCity.PageSize : Services.pageSizeDefault;

cities = citiesRequest != null ? cities : Services.GetCities(db, viewCity);

cities = Services.SortingCities(cities, sortDirect, sortBy);

ViewCity filter = null;

try

{

filter = Session["Filter"] as ViewCity;

}

catch

{

Session["Filter"] = null;

}

Session["Filter"] = Services.CheckViewModel(viewCity) ? viewCity : filter;

ViewBag.ViewCity = Session["Filter"];

ViewBag.Model = Services.GetCities(cities, page);

ViewBag.ModelCount = cities.Count;

ViewBag.ModelList = cities;

} else if(("Компании").Equals(table))

{

Services.pageSize = viewCompany.PageSize != 0 ? viewCompany.PageSize : Services.pageSizeDefault;

companies = companiesRequest != null ? companies : Services.GetCompanies(db, viewCompany);

companies = Services.SortingCompanies(companies, sortDirect, sortBy);

var cities = Services.GetCities(db, viewCity);

ViewBag.CityName = cities.Select(c => c.Name).ToList();

ViewBag.Model = Services.GetCompanies(companies, page);

ViewBag.ModelCount = companies.Count;

ViewBag.ModelList = companies;

ViewCompany filter = null;

try

{

filter = Session["Filter"] as ViewCompany;

}

catch

{

Session["Filter"] = null;

}

Session["Filter"] = Services.CheckViewModel(viewCompany) ? viewCompany : filter;

ViewBag.ViewCompany = Session["Filter"];

} else if(("Аптеки").Equals(table))

{

Services.pageSize = viewPharmacy.PageSize != 0 ? viewPharmacy.PageSize : Services.pageSizeDefault;

var companies = Services.GetCompanies(db, viewCompany);

pharmacies = pharmaciesRequest != null ? pharmacies : Services.GetPharmacies(db, viewPharmacy);

pharmacies = Services.SortingPharmacies(pharmacies, sortDirect, sortBy);

ViewBag.CompanyName = companies.Select(c => c.Name).ToList();

ViewBag.Model = Services.GetPharmacies(pharmacies, page);

ViewBag.ModelCount = pharmacies.Count;

ViewBag.ModelList = pharmacies;

ViewPharmacy filter = null;

try

{

filter = Session["Filter"] as ViewPharmacy;

}

catch

{

Session["Filter"] = null;

}

Session["Filter"] = Services.CheckViewModel(viewPharmacy) ? viewPharmacy : filter;

ViewBag.ViewPharmacy = Session["Filter"];

} else if (("Лекарства").Equals(table))

{

Services.pageSize = viewPreparation.PageSize != 0 ? viewPreparation.PageSize : Services.pageSizeDefault;

var pharmacies = Services.GetPharmacies(db, viewPharmacy);

preparations = preparationsRequest != null ? preparations : Services.GetPreparations(db, viewPreparation);

preparations = Services.SortingPreparations(preparations, sortDirect, sortBy);

ViewBag.PharmacyID = pharmacies.Select(c => c.PharmacyID).ToList();

ViewBag.Model = Services.GetPreparations(preparations, page);

ViewBag.ModelCount = preparations.Count;

ViewBag.ModelList = preparations;

ViewPreparation filter = null;

try

{

filter = Session["Filter"] as ViewPreparation;

}

catch

{

Session["Filter"] = null;

}

Session["Filter"] = Services.CheckViewModel(viewPreparation) ? viewPreparation : filter;

ViewBag.ViewPreparation = Session["Filter"];

} else if (("Аналоги").Equals(table))

{

Services.pageSize = viewAnalog.PageSize != 0 ? viewAnalog.PageSize : Services.pageSizeDefault;

analogs = analogsRequest != null ? analogs : Services.GetAnalogs(db, viewAnalog);

analogs = Services.SortingAnalog(analogs, sortDirect, sortBy);

var preparations = Services.GetPreparations(db, viewPreparation);

ViewBag.PreparationName = preparations.Select(c => c.Name).ToList();

ViewAnalog filter = null;

try

{

filter = Session["Filter"] as ViewAnalog;

}

catch

{

Session["Filter"] = null;

}

Session["Filter"] = Services.CheckViewModel(viewAnalog) ? viewAnalog : filter;

ViewBag.ViewPlayer = Session["Filter"];

ViewBag.Model = Services.GetAnalogs(analogs, page);

ViewBag.ModelCount = analogs.Count;

ViewBag.ModelList = analogs;

}

ViewBag.TableName = tableNameList;

ViewBag.ModelName = table;

ViewBag.Iterator = 0;

ViewBag.SortBy = sortBy;

ViewBag.SortDir = sortDirect;

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

}

}

**Вывод**

В ходе выполнения данной лабораторной работы были получены навыки в использовании библиотеки компилируемых шаблонов в Web-приложениях.