|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **ОТЧЕТ** |
|  |
| по учебной практике  УП.02.01. Учебная практика  по профессиональному модулю ПМ.02. Разработка и администрирование баз данных |
|  |
|  |
| студентки группы 090203-9о-20/1  специальности 09.02.03 Программирование в компьютерных системах |
| Клевцовой Анны Николаевны |

|  |  |  |
| --- | --- | --- |
| Студентка |  | А.Н. Клевцова |
|  |  |  |
| Преподаватель |  | Л.В. Салахова |

Москва

2022

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение………………………....................................................................... | 3 |
| Основная часть………………………………………………………………. | 6 |
| Заключение…………………………………………..…………………......... | 9 |
| Приложение………………………………………………………………….. | 10 |

**ВВЕДЕНИЕ**

Выполнение заданий Учебной практики 02.01 ПМ.02. Разработка и администрирование баз данных подразумевает освоение ряда умений и компетенций.

Данная учебная практика формирует следующие профессиональные компетенции:

ПК 2.1. Разработка объектов баз данных.

ПК 2.2. Реализация баз данных в конкретной СУБД.

ПК 2.3. Решение вопросов администрирования базы данных.

ПК 2.4. Реализация методов и технологий защиты информации в базах данных.

По результатам освоения профессионального модуля были получены навыки:

− работы с объектами базы данных в конкретной системе управления базами данных;

− использования средств заполнения базы данных;

− использования стандартных методов защиты объектов базы данных;

Были получены знания:

− по созданию объектов баз данных в современных СУБД и по управлению доступа к этим объектам;

− по работе с современными case-средствами проектирования баз данных;

− о формировании и настраивания схем базы данных;

− о разработке прикладных программ с использованием языка SQL;

− по созданию хранимых процедур и триггеров на базах данных;

− о применении стандартных методов для защиты объектов базы данных;

Были освоены знания:

− о основных положениях теории баз данных, хранилищ данных, баз знаний;

− о основных принципах построения концептуальной, логической и физической модели данных;

− о современных инструментальных средствах разработки схемы базы данных;

− о методах описания схем баз данных в современных СУБД;

− о структурах данных СУБД, общий подход к организации представлений,

− таблиц, индексов и кластеров;

− методы организации целостности данных;

− способы контроля доступа к данным и управления привилегиями;

− основные методы и средства защиты данных в базах данных;

− модели и структуры информационных систем;

− основные типы сетевых топологий, приемы работы в компьютерных сетях;

− информационные ресурсы компьютерных сетей;

− технологии передачи и обмена данными в компьютерных сетях;

− основы разработки приложений баз данных.

Помимо этого, были получены общие компетенции:

ОК 1 Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес

ОК 2 Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество

ОК 3 Решать проблемы, оценивать риски и принимать решения в нестандартных ситуациях

ОК 4 Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития

ОК 5 Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности

ОК 6 Работать в коллективе и команде, обеспечивать её сплочение, эффективно общаться с коллегами, руководством, потребителями

ОК 7 Ставить цели, мотивировать деятельность подчинённых, организовывать и контролировать их работу с принятием на себя ответственности за результат выполнения заданий

ОК 8 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации

ОК 9 Быть готовым к смене технологий в профессиональной деятельности

**ОСНОВНАЯ ЧАСТЬ**

Данная учебная практика состояла из 16 практических работ, которые должны были выполняться на SQL Server. Некоторые задания также должны были выполняться на языке C#.

Практическая работа №1-2 позволяла приобрести навыки разработки  баз данных в среде MS SQL. Задание заключалось в проектировании базы данных, позволяющей автоматизировать работу конкурса студентов по специальностям. Помимо этого, необходимо было определить отношения и их атрибуты, привести отношения к 3 нормальной форме, задать первичные и внешние ключи, ограничения целостности, создать  логическую схему базы данных  в ERAssistant  и преобразовать её в физическую модель СУБД Microsoft Server, заполнить таблицы данными.

Выполнение работы происходило последовательно. Изначально была спроектирована логическая схема базы данных, после чего были описаны таблицы, и создана физическая модель базы данных. Были созданы таблицы с различными типами данных и составлена диаграмма базы данных. После чего, таблицы заполнялись исходя из данных, компетенций, модулей и заданий на WorldSkills. Некоторые таблички были импортированы из Microsoft Excel, с предварительным форматированием таблиц.

Практическая работа №3-4 имела цель научиться писать запросы к БД на языке SQL. Для этого была использована база данных из предыдущей практической работы, что оказалось незатруднительно.

Практическая работа №5-6  заключалась в приобретении навыков практической работы с представлениями в среде SQL SERVER.

Изначально была скачана учебная база данных Northwind. Запущен скрипт и описаны таблицы. Создана диаграмма базы данных. После чего была описана предметная область данной базы данных и созданы представления, в которых было 14 заданий.

Практическая работа №7-8 требовала умений по написанию запросов к БД на языке SQL. Необходимо было реализовать 18 запросов с различным уровнем сложности.

Практическая работа №9-10 подразумевала реализацию скриптов на создание и заполнение таблиц. Для выполнения этой практической работы была использована база данных из заданий на учебную практику прошлого года (База данных практических работ.

Таблица 1 (выполнения работ): практическая работа, студент, номер варианта, номер уровня, дата сдачи, оценка.

Таблица 2 (практических работ): название, количество часов.

Таблица 3 (студентов): ФИО, оценка за прошлый семестр), которая была составлена в виде логической схемы в ERassistant. После чего, были созданы скрипты для создания базы данных, создания ее таблиц и их заполнения. Примеры скриптов были предложены в практической работе.

Практическая работа №11-12 состояла в написании запросов к базе данных, составленной в предыдущей практической работе. Всего было 20 запросов, которые было необходимо реализовать на языке SQL.

Практическая работа №13-14 заключалась в обучении работы с базой данных MS SQL Server в С# через ADO.net. Для этого была использована база данных с практических работ №9-10 и №11-12. Изначально было создано консольное приложение в С#. После чего осуществлялось подключение к серверу с проверкой, вывод на печать параметры строки подключения. В заключение, были выполнены 4 запроса программно на С#, сюда входили запросы на выборку записей из любой из таблиц из практической работы №11-12, один из запросов из практической работы №11-12, запрос на добавление в одну из таблиц из практической работы №11-12 и запрос на обновление записей в одной из таблиц из практической работы №11-12.

Практическая работа №15-16 была предназначена для студентов, претендующих на оценки 4 и 5 и состояла из работы с данными. Для выполнения данной практической работы, было необходимо изучить документацию национального чемпионата WorldSkills 2016 года по компетенции Программные решения для бизнеса. После чего выполнить задания первой сессии WorldSkills 2016 года, 2 из которых необходимо было реализовать в консольном приложении С#.

Первые задания требовалось реализовать на SQL Server, где нужно было выполнить скрипт на создание базы данных марафона. Далее, по заданию, нужно было осуществить создание трех таблиц с помощью импорта текстовых документов. Самым интересным этапом оказалась реализация консольного режима главного экрана системы марафона. Главной задачей было написать программу на С#, реализующую таймер дней, минут и секунд до начала марафона. Дату требовалось использовать актуальную. Помимо этого, необходимо было оформить внешний вид главного экрана.

В данной практике, наиболее интересные задания оказались практическая работа №13-14 и практическая работа №15-16. При выполнении работ, особых трудностей не возникало. Также было очень полезно научиться работать с базами данных и с запросами на С#.

Выполненные работы находятся в приложении.

**ЗАКЛЮЧЕНИЕ**

В результате прохождения учебной практики были приобретены навыки работы с объектами базы данных в конкретной системе управления базами данных, использования средств заполнения базы данных, использования стандартных методов защиты объектов базы данных, также были получены знания по созданию объектов баз данных в современных СУБД и по управлению доступа к этим объектам, по работе с современными case-средствами проектирования баз данных, о формировании и настраивания схем базы данных, о разработке прикладных программ с использованием языка SQL, по созданию хранимых процедур и триггеров на базах данных, о применении стандартных методов для защиты объектов базы данных и многие другие практические умения.

Поставленные задачи тщательно изучались, после чего были реализованы на SQL Server и С#.

Помимо этого, был получен опыт в работе с коллективом, понимание сущности и социальной значимости своей будущей профессии, организована собственная деятельность, выбраны типовые методы и способы выполнения профессиональных задач, оценены их эффективность и качество. Приобретен навык принятия решения в стандартных и нестандартных ситуациях, а также осуществление поиска и использования информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития, использование информационно-коммуникационных технологий в профессиональной деятельности.

Учебная база данных позволила овладеть разработкой и администрированием баз данных. Созданием скриптов, проектированием баз данных, созданием запросов и представлений, работой в консольном режиме с базами данных.

В итоге, был получен хороший личностный и практический опыт, который в дальнейшем позволит продолжать изучение данной предметной области с бо́льшим углублением знаний и умений.

**ПРИЛОЖЕНИЯ**

Практическая работа №3-4

№1

INSERT Competencies(CompetenciesID,CompetenciesName)

VALUES (11, 'Программные решения для бизнеса'),

(12, 'Web-разработка');

№2

INSERT educational\_institutions(institutionID,InstitutionName,city)

VALUES (11, 'УКИТ им.Разумовского','Москва');

№13

SELECT Distinct InstitutionName

from dbo.educational\_institutions left outer JOIN

dbo.Participants ON dbo.educational\_institutions.institutionID = dbo.Participants.institutionID left outer JOIN

dbo.participants\_of\_the\_event ON dbo.Participants.participantsID = dbo.participants\_of\_the\_event.participantsID

WHERE competenciesID!=1 and competenciesID!=2

ORDER BY InstitutionName

№14

SELECT Distinct InstitutionName

from dbo.educational\_institutions inner JOIN

dbo.Participants ON dbo.educational\_institutions.institutionID = dbo.Participants.institutionID inner JOIN

dbo.participants\_of\_the\_event ON dbo.Participants.participantsID = dbo.participants\_of\_the\_event.participantsID

WHERE competenciesID IS NULL

№17

SELECT InstitutionName,participants.surname,name, competencies.CompetenciesName,count(scores) as count

FROM dbo.participants\_of\_the\_event INNER JOIN

dbo.Events ON dbo.participants\_of\_the\_event.eventID = dbo.Events.eventID INNER JOIN

dbo.Competence\_performance ON dbo.Events.eventID = dbo.Competence\_performance.eventID INNER JOIN

dbo.Tasks ON dbo.Competence\_performance.taskID = dbo.Tasks.tasksID INNER JOIN

dbo.Moduls ON dbo.Tasks.tasksID = dbo.Moduls.taskID INNER JOIN

dbo.Results ON dbo.Moduls.modulID = dbo.Results.modulID INNER JOIN

dbo.Participants ON dbo.participants\_of\_the\_event.participantsID = dbo.Participants.participantsID INNER JOIN

dbo.Tasks AS Tasks\_1 ON dbo.Competence\_performance.taskID = Tasks\_1.tasksID AND dbo.Moduls.taskID = Tasks\_1.tasksID INNER JOIN

dbo.Competencies ON dbo.participants\_of\_the\_event.competenciesID = dbo.Competencies.CompetenciesID AND dbo.Competence\_performance.competenciesID = dbo.Competencies.CompetenciesID INNER JOIN

dbo.educational\_institutions ON dbo.Participants.institutionID = dbo.educational\_institutions.institutionID

WHERE participants\_of\_the\_event.competenciesID=1 or participants\_of\_the\_event.competenciesID=2

GROUP BY InstitutionName,participants.surname,name, competencies.CompetenciesName

ORDER BY InstitutionName,competencies.CompetenciesName,count(scores) DESC

№18

SELECT participants.surname,competencies.CompetenciesName,scores

FROM dbo.participants\_of\_the\_event INNER JOIN

dbo.Events ON dbo.participants\_of\_the\_event.eventID = dbo.Events.eventID INNER JOIN

dbo.Competence\_performance ON dbo.Events.eventID = dbo.Competence\_performance.eventID INNER JOIN

dbo.Tasks ON dbo.Competence\_performance.taskID = dbo.Tasks.tasksID INNER JOIN

dbo.Moduls ON dbo.Tasks.tasksID = dbo.Moduls.taskID INNER JOIN

dbo.Results ON dbo.Moduls.modulID = dbo.Results.modulID INNER JOIN

dbo.Participants ON dbo.participants\_of\_the\_event.participantsID = dbo.Participants.participantsID INNER JOIN

dbo.Tasks AS Tasks\_1 ON dbo.Competence\_performance.taskID = Tasks\_1.tasksID AND dbo.Moduls.taskID = Tasks\_1.tasksID INNER JOIN

dbo.Competencies ON dbo.participants\_of\_the\_event.competenciesID = dbo.Competencies.CompetenciesID AND dbo.Competence\_performance.competenciesID = dbo.Competencies.CompetenciesID INNER JOIN

dbo.educational\_institutions ON dbo.Participants.institutionID = dbo.educational\_institutions.institutionID

ORDER BY scores DESC

№19

SELECT top 3 participants\_of\_the\_event.participantsID,surname,competencies.CompetenciesName,scores

FROM dbo.participants\_of\_the\_event INNER JOIN

dbo.Events ON dbo.participants\_of\_the\_event.eventID = dbo.Events.eventID INNER JOIN

dbo.Competence\_performance ON dbo.Events.eventID = dbo.Competence\_performance.eventID INNER JOIN

dbo.Tasks ON dbo.Competence\_performance.taskID = dbo.Tasks.tasksID INNER JOIN

dbo.Moduls ON dbo.Tasks.tasksID = dbo.Moduls.taskID INNER JOIN

dbo.Results ON dbo.Moduls.modulID = dbo.Results.modulID INNER JOIN

dbo.Participants ON dbo.participants\_of\_the\_event.participantsID = dbo.Participants.participantsID INNER JOIN

dbo.Tasks AS Tasks\_1 ON dbo.Competence\_performance.taskID = Tasks\_1.tasksID AND dbo.Moduls.taskID = Tasks\_1.tasksID INNER JOIN

dbo.Competencies ON dbo.participants\_of\_the\_event.competenciesID = dbo.Competencies.CompetenciesID AND dbo.Competence\_performance.competenciesID = dbo.Competencies.CompetenciesID INNER JOIN

dbo.educational\_institutions ON dbo.Participants.institutionID = dbo.educational\_institutions.institutionID

ORDER BY scores DESC

№20

SELECT top 3 participants\_of\_the\_event.participantsID,surname,competencies.CompetenciesName,scores

FROM dbo.participants\_of\_the\_event INNER JOIN

dbo.Events ON dbo.participants\_of\_the\_event.eventID = dbo.Events.eventID INNER JOIN

dbo.Competence\_performance ON dbo.Events.eventID = dbo.Competence\_performance.eventID INNER JOIN

dbo.Tasks ON dbo.Competence\_performance.taskID = dbo.Tasks.tasksID INNER JOIN

dbo.Moduls ON dbo.Tasks.tasksID = dbo.Moduls.taskID INNER JOIN

dbo.Results ON dbo.Moduls.modulID = dbo.Results.modulID INNER JOIN

dbo.Participants ON dbo.participants\_of\_the\_event.participantsID = dbo.Participants.participantsID INNER JOIN

dbo.Tasks AS Tasks\_1 ON dbo.Competence\_performance.taskID = Tasks\_1.tasksID AND dbo.Moduls.taskID = Tasks\_1.tasksID INNER JOIN

dbo.Competencies ON dbo.participants\_of\_the\_event.competenciesID = dbo.Competencies.CompetenciesID AND dbo.Competence\_performance.competenciesID = dbo.Competencies.CompetenciesID INNER JOIN

dbo.educational\_institutions ON dbo.Participants.institutionID = dbo.educational\_institutions.institutionID

WHERE participants\_of\_the\_event.competenciesID=1 or participants\_of\_the\_event.competenciesID=2

ORDER BY scores DESC

Практическая работа №7-8

Вариант 2

№1

UPDATE Orders SET ShipAddress='Bulvar'

WHERE OrderID='10252'

UPDATE [Order Details] SET Quantity='26'

WHERE OrderID='10252'

SELECT DISTINCT ProductName, [Order Details].\*,ShipAddress as adres, Quantity as count

FROM dbo.Products INNER JOIN [Order Details] ON [Order Details].ProductID=Products.ProductID

INNER JOIN Orders ON Orders.OrderID = [Order Details].OrderID

WHERE ProductName LIKE 'Geitost'

№2

SELECT DISTINCT ShipCountry,ProductName

FROM dbo.Products INNER JOIN [Order Details] ON [Order Details].ProductID=Products.ProductID

INNER JOIN Orders ON Orders.OrderID = [Order Details].OrderID

ORDER BY ShipCountry

№3

SELECT count(ShipCountry) as count

FROM dbo.View\_1

№9

SELECT count(ProductName) as count

FROM dbo.Products

WHERE UnitPrice < (SELECT avg(UnitPrice)

FROM dbo.Products)

№12

SELECT top 5 OrderID,ProductName,CategoryID,Quantity

FROM dbo.Products INNER JOIN [Order Details] ON [Order Details].ProductID=Products.ProductID

ORDER BY Quantity DESC

№13

SELECT DISTINCT ShipCountry,count(ProductName) as count

FROM dbo.Products INNER JOIN [Order Details] ON [Order Details].ProductID=Products.ProductID

INNER JOIN Orders ON Orders.OrderID = [Order Details].OrderID

WHERE ShipCountry='Spain'

GROUP BY ShipCountry

№14

SELECT ProductName

FROM dbo.Products P

WHERE NOT EXISTS

(

SELECT 1

FROM dbo.Products Left outer JOIN

dbo.[Order Details] ON dbo.[Order Details].ProductID = dbo.Products.ProductID Left outer JOIN

dbo.Orders ON dbo.Orders.OrderID = dbo.[Order Details].OrderID

WHERE ShipCountry='Poland' and [Order Details].ProductID = P.ProductID

)

№15

SELECT count(ProductName) as count

FROM dbo.Products P

WHERE NOT EXISTS

(

SELECT 1

FROM dbo.Products Left outer JOIN

dbo.[Order Details] ON dbo.[Order Details].ProductID = dbo.Products.ProductID Left outer JOIN

dbo.Orders ON dbo.Orders.OrderID = dbo.[Order Details].OrderID

WHERE ShipCountry='Poland' and [Order Details].ProductID = P.ProductID

)

№16

SELECT DISTINCT ShipCountry, YEAR(OrderDate) as year,sum([Order Details].OrderID) as sum

FROM dbo.Products INNER JOIN [Order Details] ON [Order Details].ProductID=Products.ProductID

INNER JOIN Orders ON Orders.OrderID = [Order Details].OrderID

GROUP BY ShipCountry,YEAR(OrderDate)

ORDER BY ShipCountry

№17

SELECT top 1 ShipCountry,YEAR(OrderDate) as year,sum(Orders.OrderID) as max

FROM dbo.Products INNER JOIN [Order Details] ON [Order Details].ProductID=Products.ProductID

INNER JOIN Orders ON Orders.OrderID = [Order Details].OrderID

GROUP BY ShipCountry,YEAR(OrderDate),Orders.OrderID

ORDER BY max DESC

Практическая работа №11-12

№1

INSERT practicalwork(practical\_workID,practical\_work,number\_of\_hours)

VALUES (11, 'Excel', 5),

(12, 'SQL', 3);

INSERT Students(studentID,surname,name,patronymic,last\_semesters\_grade)

VALUES (11, 'Романова','Полина','Алексеевна', 5),

(12, 'Осокин','Михаил','Васильевич', 5);

INSERT execution\_of\_works(Number,studentID ,practical\_workID,optionsID,levelID,date\_of\_delivery,estimation)

VALUES (21, 11,1,1, 5,'2022-10-12',5),

(22, 12,3,2, 3,'2022-11-12',5);

№2

SELECT count(\*) as count

FROM practicalwork

SELECT count(\*) as count

FROM Students

SELECT count(\*) as count

FROM execution\_of\_works

№14

SELECT \*

FROM Students

WHERE SUBSTRING(name, 1, 1) = 'А'

№15

SELECT \*,DATEPART(MONTH,date\_of\_delivery) as date

FROM execution\_of\_works

№16

SELECT surname,practical\_workID

FROM execution\_of\_works

RIGHT OUTER JOIN Students ON Students.studentID=execution\_of\_works.studentID

№17

SELECT surname, SUM(CASE WHEN estimation = 3 THEN 1 ELSE 0 END) AS Тройки

FROM practicalwork INNER JOIN execution\_of\_works ON execution\_of\_works.practical\_workID=practicalwork.practical\_workID

INNER JOIN Students ON Students.studentID=execution\_of\_works.studentID

GROUP BY surname

HAVING (SUM(CASE WHEN estimation = 3 THEN 1 ELSE 0 END) > 0)

№18

SELECT surname,CAST(AVG(estimation) AS REAL) as средняя\_оценка,count(execution\_of\_works.practical\_workID) as колво\_сданных\_работ,

max(execution\_of\_works.practical\_workID) as последняя\_работа\_по\_нумерации,min(execution\_of\_works.practical\_workID) as первая\_работа\_по\_нумерации,

sum(estimation) as сложенные\_оценки

FROM practicalwork INNER JOIN execution\_of\_works ON execution\_of\_works.practical\_workID=practicalwork.practical\_workID

INNER JOIN Students ON Students.studentID=execution\_of\_works.studentID

GROUP BY surname

№19

SELECT surname

FROM Students

WHERE EXISTS(SELECT max(estimation) FROM execution\_of\_works )

SELECT surname

FROM Students

WHERE EXISTS(SELECT estimation FROM execution\_of\_works WHERE estimation=5)

№20

SELECT DISTINCT top 1 surname, count(estimation) AS count5,max(estimation) as maxocenka

FROM practicalwork INNER JOIN execution\_of\_works ON execution\_of\_works.practical\_workID=practicalwork.practical\_workID

INNER JOIN Students ON Students.studentID=execution\_of\_works.studentID

WHERE date\_of\_delivery between '2022-05-12' and '2022-07-12' and surname like '%а%'

GROUP BY surname,date\_of\_delivery

ORDER BY count(estimation) DESC

Практическая работа №13-14

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Reflection;

namespace ConsoleConnectToServer

{

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Connect to SQL Server and demo Create, Read, Update and Delete operations.");

// Build connection string

SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();

builder.DataSource = @".\SQLExpress";

builder.InitialCatalog = "practicalworks";

builder.IntegratedSecurity = true;

// Connect to SQL

Console.Write("Connecting to SQL Server ... ");

using (SqlConnection connection = new

SqlConnection(builder.ConnectionString))

{

connection.Open();

Console.WriteLine("Server" + builder.DataSource + " База данных " +

builder.InitialCatalog);

Console.WriteLine("Done.");

}

}

catch (SqlException e)

{

Console.WriteLine(e.ToString());

}

Console.WriteLine("All done. Press any key to finish...");

Console.ReadKey(true);

}

}

}

1. Запрос на выборку записей из любой из таблиц из практической работы №11-12

Вывести 3 поля из любой таблицы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

namespace ConsoleApp10

{

class Program

{

static void Main(string[] args)

{

SqlConnection connection = new SqlConnection(@"Data Source=.\SQLExpress;Initial Catalog=practicalworks;" + "Integrated Security=True;Connect Timeout=15;Encrypt=False;" + "TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

Console.WriteLine("\nВведите таблицу:");

string table = Console.ReadLine();

Console.WriteLine("\nВведите атрибут для вывода:");

string a = Console.ReadLine();

Console.WriteLine("Введите атрибут для вывода:");

string b = Console.ReadLine();

Console.WriteLine("Введите атрибут для вывода:");

string c = Console.ReadLine();

try

{

connection.Open();

string sqlExpression = String.Format("SELECT {0}, {1},{2} FROM {3}", a,b,c,table);

SqlCommand command = new SqlCommand(sqlExpression, connection);

Console.WriteLine(command.CommandText);

command.CommandText = sqlExpression;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Console.WriteLine("\t{0}\t{1} {2}\t", reader[0], reader[1], reader[2]);

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.Read();

}

}

}

1. Один из запросов из практической работы №11-12

Создать запрос на выборку названий или фамилий, которые начинаются на первую букву вашей фамилии, а заканчиваются на последнюю букву вашей фамилии. Привести несколько вариантов решения. Сравнить результаты.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

namespace ConsoleApp10

{

class Program

{

static void Main(string[] args)

{

SqlConnection connection = new SqlConnection(@"Data Source=.\SQLExpress;Initial Catalog=practicalworks;" + "Integrated Security=True;Connect Timeout=15;Encrypt=False;" + "TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

try

{

connection.Open();

string sqlExpression = String.Format("SELECT surname FROM Students WHERE LEFT(surname,1)='К' and RIGHT(surname,1)='а'\r\n");

SqlCommand command = new SqlCommand(sqlExpression, connection);

Console.WriteLine(command.CommandText);

command.CommandText = sqlExpression;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Console.WriteLine("{0}", reader[0]);

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.Read();

}

}

}

1. запрос на добавление в одну из таблиц из практической работы №11-12

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

namespace ConsoleApp10

{

class Program

{

static void Main(string[] args)

{

SqlConnection connection = new SqlConnection(@"Data Source=.\SQLExpress;Initial Catalog=practicalworks;" + "Integrated Security=True;Connect Timeout=15;Encrypt=False;" + "TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

Console.WriteLine("\nВведите номер практической работы:");

int a = Int32.Parse(Console.ReadLine());

Console.WriteLine("Введите название практической работы:");

string b = Console.ReadLine();

Console.WriteLine("Введите кол-во часов:");

int c = Int32.Parse(Console.ReadLine());

try

{

connection.Open();

string sqlExpression = String.Format("INSERT practicalwork(practical\_workID,practical\_work,number\_of\_hours) VALUES ({0}, '{1}', {2});", a,b,c);

SqlCommand command = new SqlCommand(sqlExpression, connection);

Console.WriteLine(command.CommandText);

command.CommandText = sqlExpression;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Console.WriteLine("{0}\t{1}\t{2}\t", reader[0], reader[1], reader[2]);

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.Read();

}

}

}

1. запрос на обновление записей в одной из таблиц из практической работы №11-12

Изменить фамилию студента, используя номер по журналу

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

namespace ConsoleApp10

{

class Program

{

static void Main(string[] args)

{

SqlConnection connection = new SqlConnection(@"Data Source=.\SQLExpress;Initial Catalog=practicalworks;" + "Integrated Security=True;Connect Timeout=15;Encrypt=False;"+"TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");

try

{

connection.Open();

Console.WriteLine("\nselect studentID,surname from students:");

string sqlExpression1 = "select studentID,surname from students";

SqlCommand command1 = new SqlCommand(sqlExpression1, connection);

command1.CommandText = sqlExpression1;

SqlDataReader reader1 = command1.ExecuteReader();

while (reader1.Read())

{

Console.WriteLine("\t{0}\t{1}\t",

reader1[0], reader1[1]);

}

reader1.Close();

Console.WriteLine("\nВведите номер студента:");

int n\_j = Int32.Parse(Console.ReadLine());

Console.WriteLine("Введите фамилию:");

string last\_name = Console.ReadLine();

string sqlExpression = String.Format("UPDATE students SET surname='{0}' WHERE studentID={1}", last\_name, n\_j);

SqlCommand command = new SqlCommand(sqlExpression, connection);

Console.WriteLine(command.CommandText);

int number = command.ExecuteNonQuery();

Console.WriteLine("Обновлено объектов: {0}", number);

sqlExpression = "select studentID,surname from students";

command.CommandText = sqlExpression;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Console.WriteLine("\t{0}\t{1}\t",

reader[0], reader[1]);

}

reader.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.Read();

}

}

}

Практическая работа №15-16

<Window x:Class="yp15\_16.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:yp15\_16"

mc:Ignorable="d"

Title="MainWindow" Height="514" Width="800">

<Grid>

<Label Content=" MARAPHON SKILLS 2022" HorizontalAlignment="Center" Height="152" VerticalAlignment="Top" Width="800" FontSize="48" Background="#FF7B7878" FontWeight="Bold" Foreground="#FFF9F9F9"/>

<Label Content=" Москва, Россия&#xD;&#xA;среда 21 декабря 2022" HorizontalAlignment="Left" Height="75" Margin="247,62,0,0" VerticalAlignment="Top" Width="282" FontSize="24" FontStyle="Italic" Foreground="#FFE6DBDB"/>

<Button Content="Я хочу стать бегуном" HorizontalAlignment="Center" Height="70" Margin="0,174,0,0" VerticalAlignment="Top" Width="372" FontSize="20"/>

<Button Content="Я хочу стать спонсором бегуна" HorizontalAlignment="Center" Height="79" Margin="0,260,0,0" VerticalAlignment="Top" Width="372" FontSize="20"/>

<Button Content="Я хочу узнать больше о событии" HorizontalAlignment="Center" Height="78" Margin="0,356,0,0" VerticalAlignment="Top" Width="372" FontSize="20"/>

<Button Content="Login" HorizontalAlignment="Left" Height="37" Margin="690,397,0,0" VerticalAlignment="Top" Width="81" FontSize="16" RenderTransformOrigin="0.441,-0.197"/>

<Label x:Name="lblTime" Content=" " HorizontalAlignment="Center" Height="39" Margin="0,449,0,0" VerticalAlignment="Top" Width="800" Background="#FF6F6D6D" Foreground="#FFF9F9F9" FontSize="16"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Windows.Threading;

namespace yp15\_16

{

/// <summary>

/// </summary>

public partial class MainWindow : Window

{

DateTime endTime = new DateTime(2022, 12, 21, 0, 0, 0);

public MainWindow()

{

InitializeComponent();

DispatcherTimer t = new DispatcherTimer();

t.Interval = new TimeSpan(500);

t.Tick += new EventHandler(t\_Tick);

TimeSpan ts = endTime.Subtract(DateTime.Now);

t.Start();

}

void t\_Tick(object sender, EventArgs e)

{

TimeSpan ts = endTime.Subtract(DateTime.Now);

lblTime.Content = " " + ts.ToString("d' дней 'h' часов 'm' минут и 's' секунд для старта марафона!'");

}

}

}