МИНОБРНАУКИ РОССИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Омский государственный университет им. Ф.М. Достоевского»
Институт среднего профессионального образования
и довузовской подготовки

Методические рекомендации по выполнению демонстрационного экзамена для специальности 09.02.07

Омск – 2025

СОДЕРЖАНИЕ

Рекомендованный хронометраж этапов выполнения задания

№	**Время** (3ч. 30мин)	Действие
1	40 мин	Разработка БД: ER – диаграмма (разработать модель, не заполняя данными), сохранить в ее в pdf, экспортировать в физическую БД, которую заполнить данными, предоставленными в задании
2	40 мин	Разработка графического интерфейса (экранные формы в MS VS)
3	50 мин	Связать графический интерфейс с БД (connect) Выполнить обработку окна авторизации в соответствии с заданием Выполнить загрузку таблиц БД в таблицы экранных форм Выполнить SQL-запросы из задания
4	20 мин	Тестирование (рассмотреть 2 или 3 тестовые случая, лучше представить в виде таблицы)
5	20 мин	Описание системы (титульный лист, описание интерфейса)
6	40 мин	Доделать, все что недоделано

АЛГОРИТМ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Модуль №1 Разработка БД

Текст задания: На основании описания брифинга и документов, представленных заказчиком, необходимо спроектировать ER-диаграмму для информационной системы. Обязательна 3 нормальная форма с обеспечением ссылочной целостности. При разработке диаграммы обратите внимание на согласованную осмысленную схему именования, создайте необходимые первичные и внешние ключи.

ER – диаграмма должна быть представлена в формате .pdf и содержать таблицы, связи между ними, атрибуты и ключи (типами данных на данном этапе можно пренебречь).

Необходимые приложения: Текст брифинга.pdf, Документы заказчика.zip

Выполнение:

1. Создаём соединение MySQL:

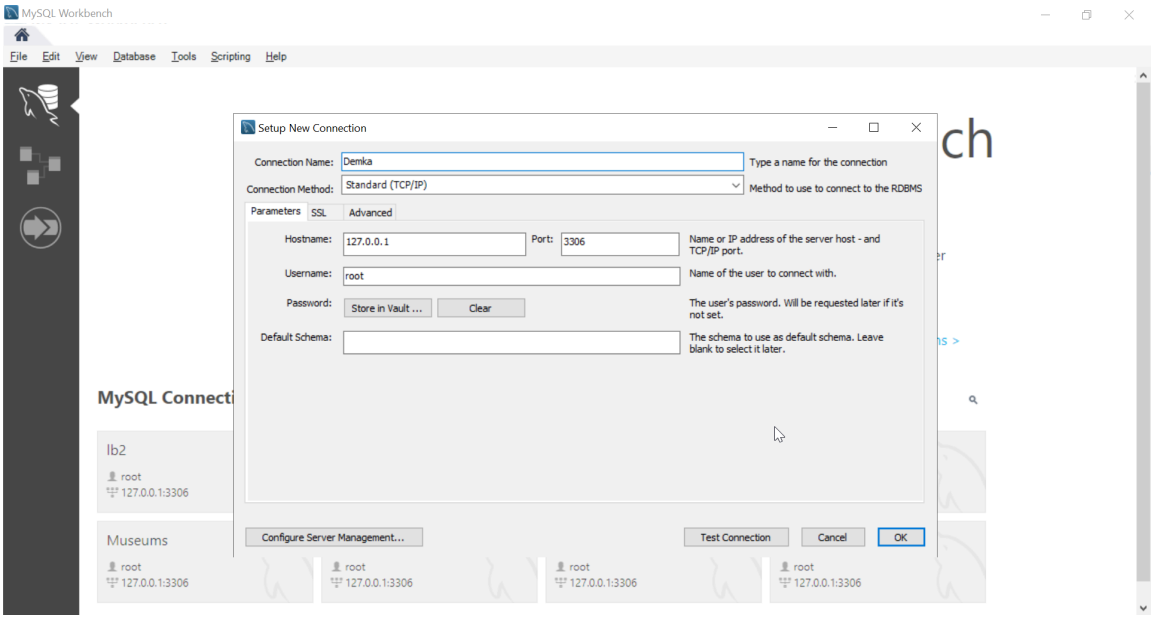

Рисунок 1 – Соединение MySQL

2. Создаём новую модель:

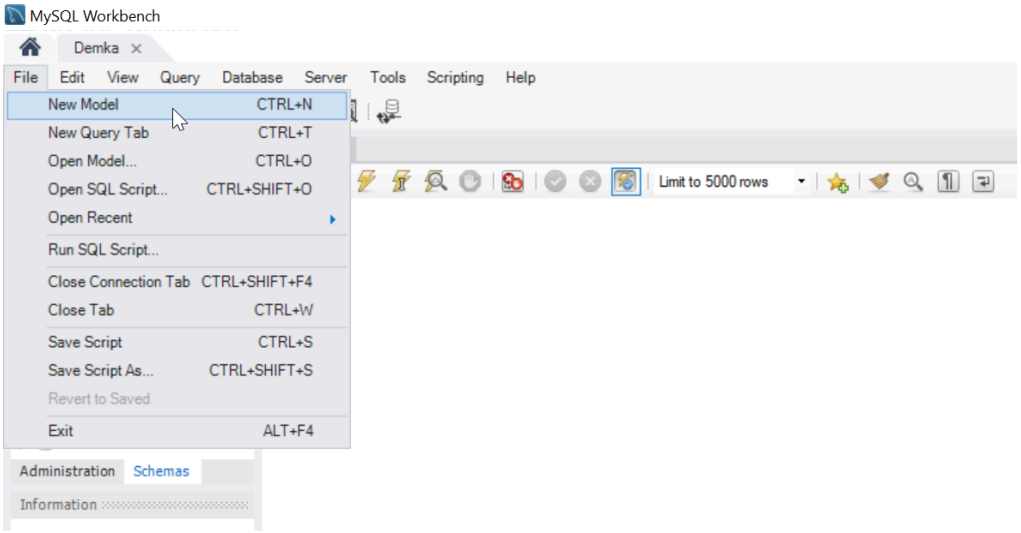

Рисунок 2 – Создание модели

3. Создаем диаграмму:

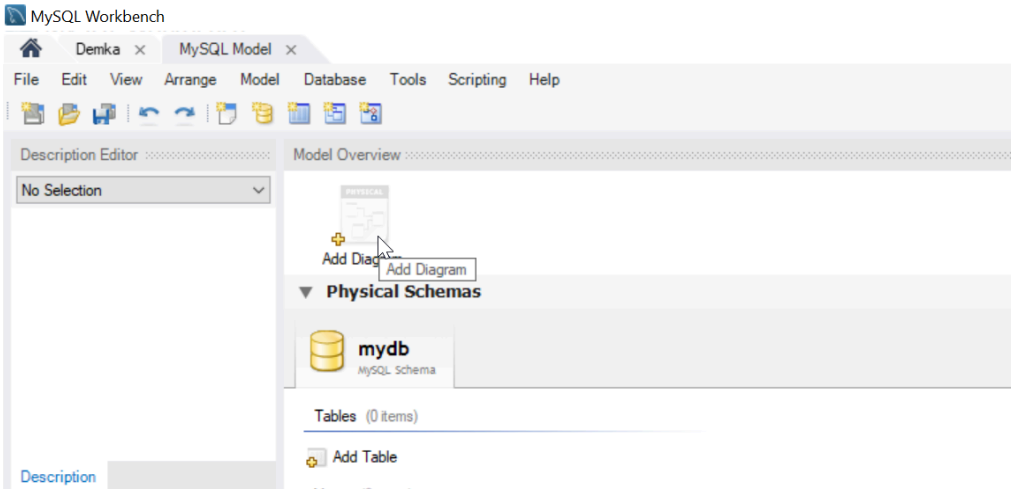

Рисунок 3 – Создание диаграммы

4. ER-моделирование (выделить сущности (таблицы), заполнить атрибуты (поля)). Таблицы Room_stock, Report, Guests, Guests_currently_living_in_the_hotel – делаются на основе Текста брифинга и Документам заказчика (по xlsx таблицам). Таблица Users обязательная и отличатся не будет. Она включает в себя ID пользователя, логин, пароль, роль (для разграничения доступа по ролям: администратор, пользователь), статус (для отслеживания статуса блокировки пользователя), попытки входа (для отслеживания количества попыток неудачного входа пользователя, 3 неудачи = блокировка), время последнего входа (для отслеживания последней

попытки входа, пользователь неактивен 30 дней = блокировка, первая попытка входа (для отслеживания первой попытки входа и перенаправления пользователя на форму смены пароля).

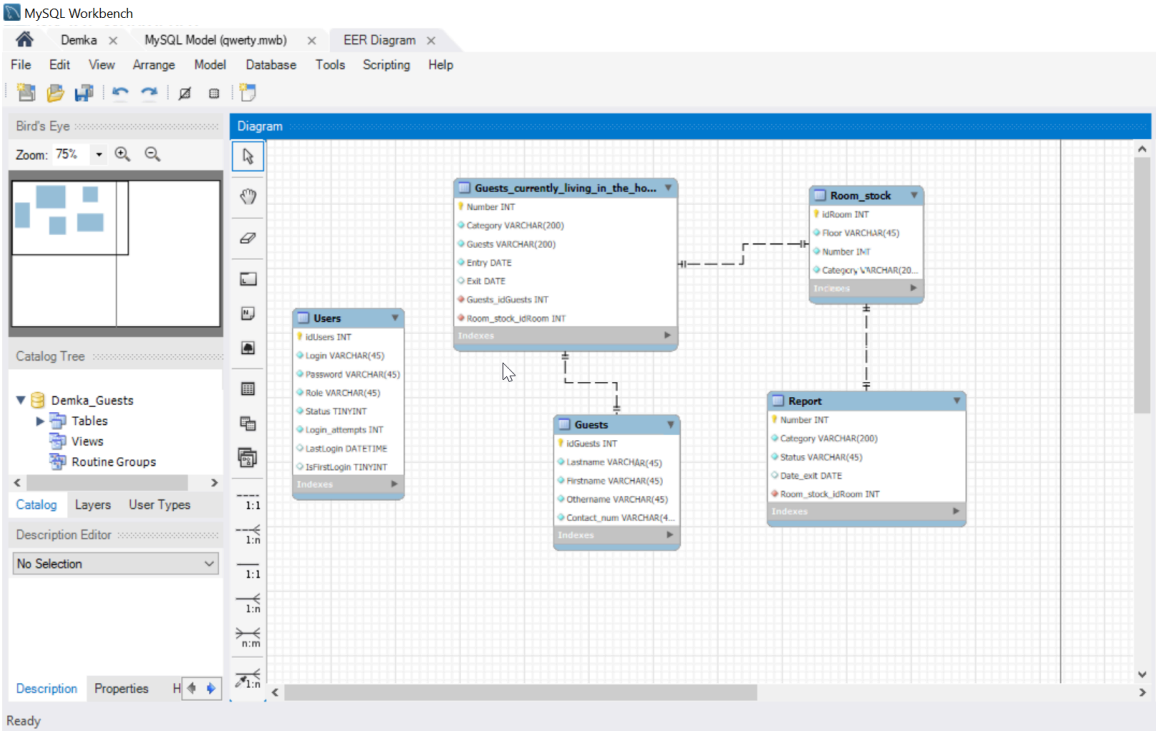

Рисунок 4 – ER-диаграмма

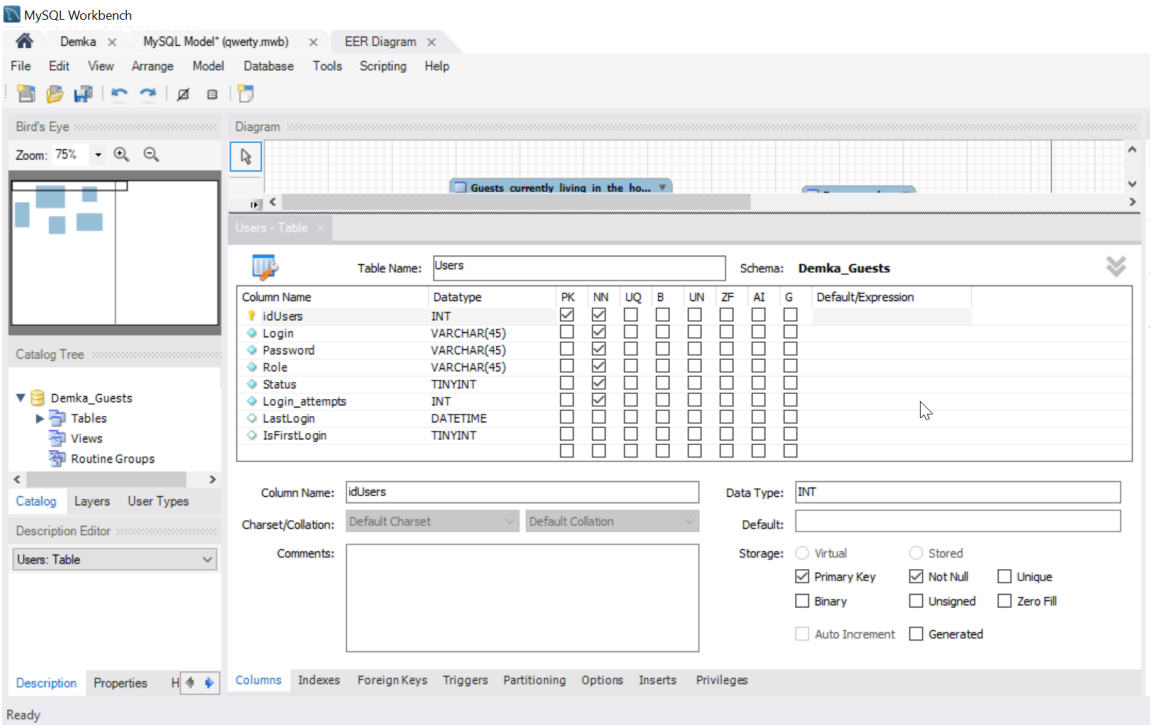

Рисунок 5 – Таблица Users

5. По требованиям задания сохраним ER-диаграмму в формате .pdf:

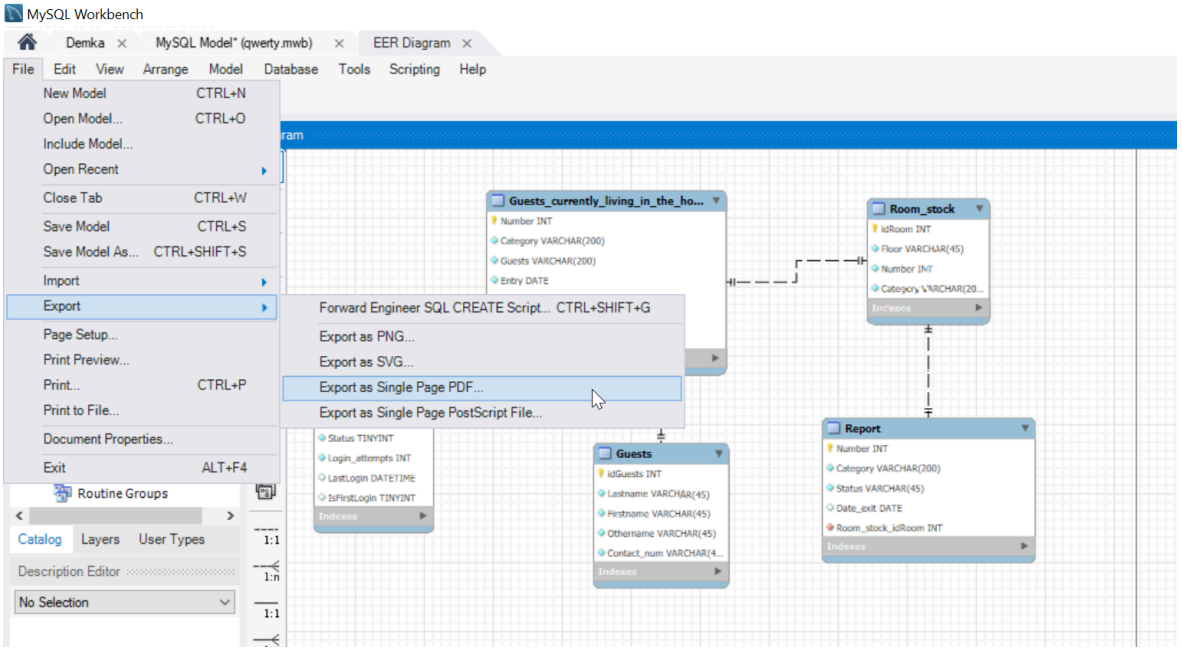

Рисунок 6 – Экспорт в pdf

6. Создаем SQL-скрипт для импортирования в физическую БД

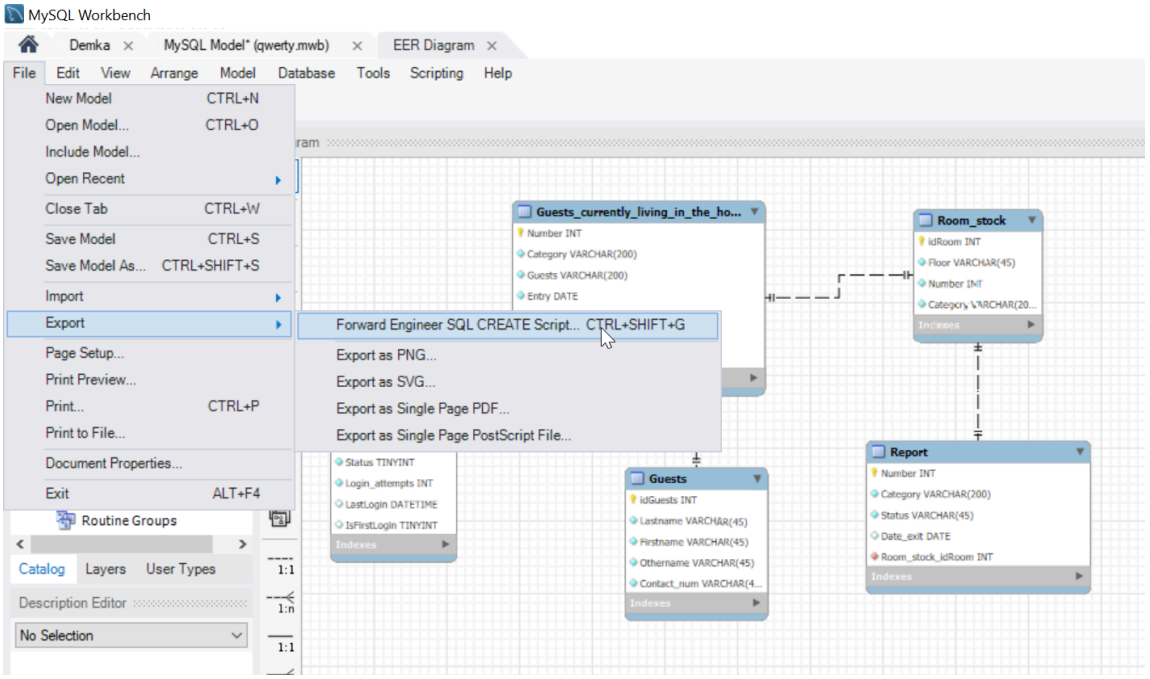

Рисунок 7 – Экспорт в физическую БД

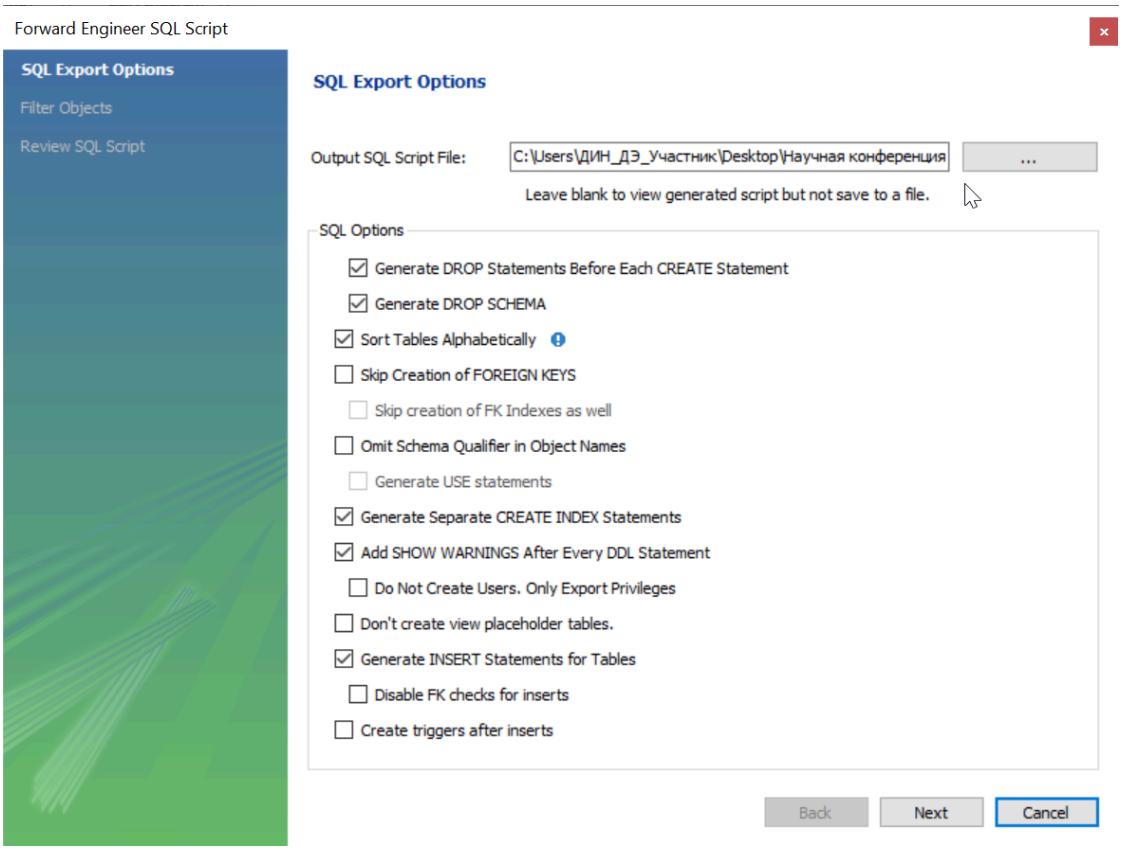

Рисунок 8 – Настройка экспорта

Листинг 1. SQL скрипт диаграммы:

```
-- MySQL Script generated by MySQL Workbench
-- Thu May 29 16:56:10 2025
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_I
N_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_S
UBSTITUTION';

-- -----------------------------------------------------
-- Schema Demka_Guests
-- -----------------------------------------------------
DROP SCHEMA IF EXISTS `Demka_Guests` ;

-- -----------------------------------------------------
-- Schema Demka_Guests
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `Demka_Guests` DEFAULT CHARACTER
SET utf8 ;
SHOW WARNINGS;
```

```
25  USE `Demka_Guests` ;
26
27  -- -----------------------------------------------------
28  -- Table `Demka_Guests`.`Guests`
29  -- -----------------------------------------------------
30  DROP TABLE IF EXISTS `Demka_Guests`.`Guests` ;
31
32  SHOW WARNINGS;
33  CREATE TABLE IF NOT EXISTS `Demka_Guests`.`Guests` (
34    `idGuests` INT NOT NULL,
35    `Lastname` VARCHAR(45) NOT NULL,
36    `Firstname` VARCHAR(45) NOT NULL,
37    `Othername` VARCHAR(45) NOT NULL,
38    `Contact_num` VARCHAR(45) NOT NULL,
39    PRIMARY KEY (`idGuests`))
40  ENGINE = InnoDB;
41
42  SHOW WARNINGS;
43
44  -- -----------------------------------------------------
45  -- Table `Demka_Guests`.`Guests_currently_living_in_the_hotel`
46  -- -----------------------------------------------------
47  DROP TABLE IF EXISTS `Demka_Guests`.`Guests_currently_living_in_the_hotel` ;
48
49  SHOW WARNINGS;
50  CREATE TABLE IF NOT EXISTS
51  `Demka_Guests`.`Guests_currently_living_in_the_hotel` (
52    `Number` INT NOT NULL,
53    `Category` VARCHAR(200) NOT NULL,
54    `Guests` VARCHAR(200) NOT NULL,
55    `Entry` DATE NOT NULL,
56    `Exit` DATE NULL,
57    `Guests_idGuests` INT NOT NULL,
58    `Room_stock_idRoom` INT NOT NULL,
59    PRIMARY KEY (`Number`),
60    CONSTRAINT `fk_Guests_currently_living_in_the_hotel_Guests`
61      FOREIGN KEY (`Guests_idGuests`)
62      REFERENCES `Demka_Guests`.`Guests` (`idGuests`)
63      ON DELETE NO ACTION
64      ON UPDATE NO ACTION,
65    CONSTRAINT `fk_Guests_currently_living_in_the_hotel_Room_stock1`
66      FOREIGN KEY (`Room_stock_idRoom`)
67      REFERENCES `Demka_Guests`.`Room_stock` (`idRoom`)
68      ON DELETE NO ACTION
69      ON UPDATE NO ACTION)
70  ENGINE = InnoDB;
71
72  SHOW WARNINGS;
73  CREATE INDEX `fk_Guests_currently_living_in_the_hotel_Guests_idx` ON
74  `Demka_Guests`.`Guests_currently_living_in_the_hotel` (`Guests_idGuests` ASC)
75  VISIBLE;
```

```
76
77  SHOW WARNINGS;
78  CREATE INDEX `fk_Guests_currently_living_in_the_hotel_Room_stock1_idx` ON
79  `Demka_Guests`.`Guests_currently_living_in_the_hotel` (`Room_stock_idRoom`
80  ASC) VISIBLE;
81
82  SHOW WARNINGS;
83
84  -- -----------------------------------------------------
85  -- Table `Demka_Guests`.`Report`
86  -- -----------------------------------------------------
87  DROP TABLE IF EXISTS `Demka_Guests`.`Report` ;
88
89  SHOW WARNINGS;
90  CREATE TABLE IF NOT EXISTS `Demka_Guests`.`Report` (
91    `Number` INT NOT NULL,
92    `Category` VARCHAR(200) NOT NULL,
93    `Status` VARCHAR(45) NOT NULL,
94    `Date_exit` DATE NULL,
95    `Room_stock_idRoom` INT NOT NULL,
96    PRIMARY KEY (`Number`),
97    CONSTRAINT `fk_Report_Room_stock1`
98      FOREIGN KEY (`Room_stock_idRoom`)
99      REFERENCES `Demka_Guests`.`Room_stock` (`idRoom`)
100     ON DELETE NO ACTION
101     ON UPDATE NO ACTION)
102 ENGINE = InnoDB;
103
104 SHOW WARNINGS;
105 CREATE INDEX `fk_Report_Room_stock1_idx` ON `Demka_Guests`.`Report`
106 (`Room_stock_idRoom` ASC) VISIBLE;
107
108 SHOW WARNINGS;
109
110 -- -----------------------------------------------------
111 -- Table `Demka_Guests`.`Room_stock`
112 -- -----------------------------------------------------
113 DROP TABLE IF EXISTS `Demka_Guests`.`Room_stock` ;
114
115 SHOW WARNINGS;
116 CREATE TABLE IF NOT EXISTS `Demka_Guests`.`Room_stock` (
117   `idRoom` INT NOT NULL,
118   `Floor` VARCHAR(45) NOT NULL,
119   `Number` INT NOT NULL,
120   `Category` VARCHAR(200) NOT NULL,
121   PRIMARY KEY (`idRoom`))
122 ENGINE = InnoDB;
123
124 SHOW WARNINGS;
125
126 -- -----------------------------------------------------
```

```
-- Table `Demka_Guests`.`Users`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Demka_Guests`.`Users` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Demka_Guests`.`Users` (
  `idUsers` INT NOT NULL,
  `Login` VARCHAR(45) NOT NULL,
  `Password` VARCHAR(45) NOT NULL,
  `Role` VARCHAR(45) NOT NULL,
  `Status` TINYINT NOT NULL,
  `Login_attempts` INT NOT NULL,
  `LastLogin` DATETIME NULL,
  `IsFirstLogin` TINYINT NULL,
  PRIMARY KEY (`idUsers`))
ENGINE = InnoDB;

SHOW WARNINGS;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Модуль №2 Соадминистрирование баз данных и серверов

Текст задания: Создайте базу данных на основании разработанной ER диаграммы, используя предпочтительную платформу, на сервере баз данных, который вам предоставлен. Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения.

После создания базы данных требуется импортировать данные из файла "Номерной фонд.xlsx".

Создайте запрос вычисляющий процент загрузки номерного фонда – это отношение количества проданных ночей к общему количеству номеров в отеле.

Необходимые приложения: Текст брифинга.pdf, Документы заказчика.zip

Выполнение:

1. Импортируем файл с моделью базы данных, для создания физической базы данных

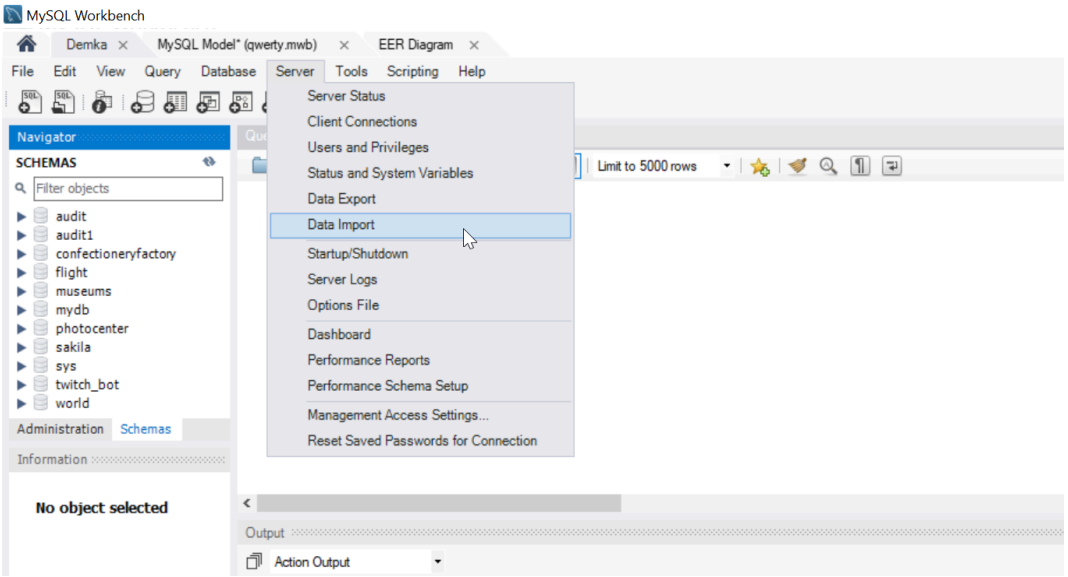

Рисунок 9 – Импорт ER-диаграммы

Далее выбираем место хранения sql-скрипта, который мы создали ранее.

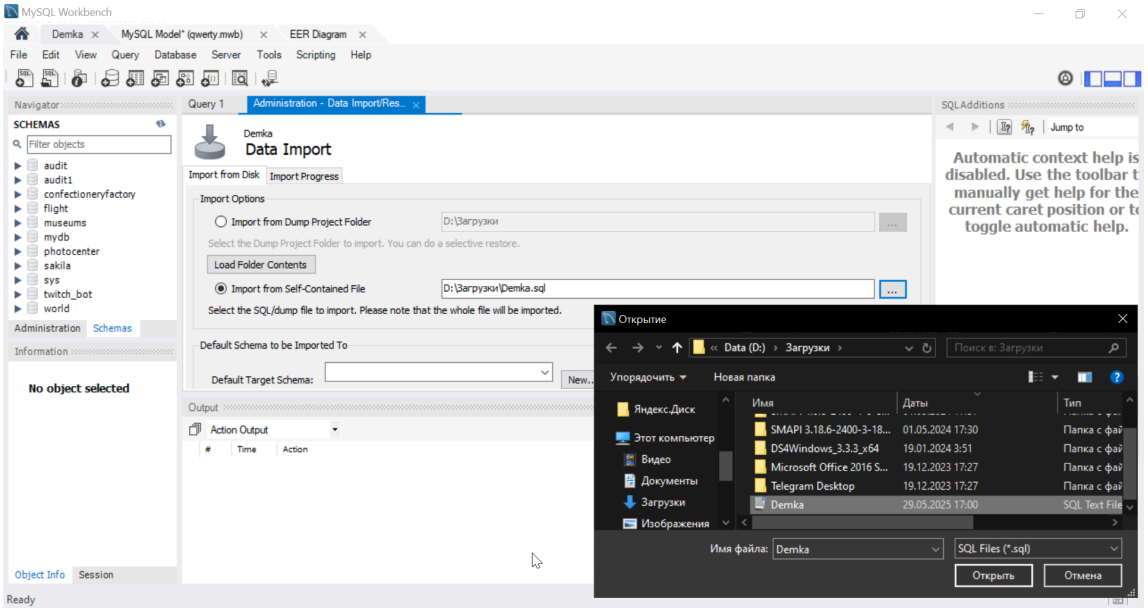

Рисунок 10 – Импорт ER-диаграммы

Переходим во вкладку Import Progress и нажимаем Start Import

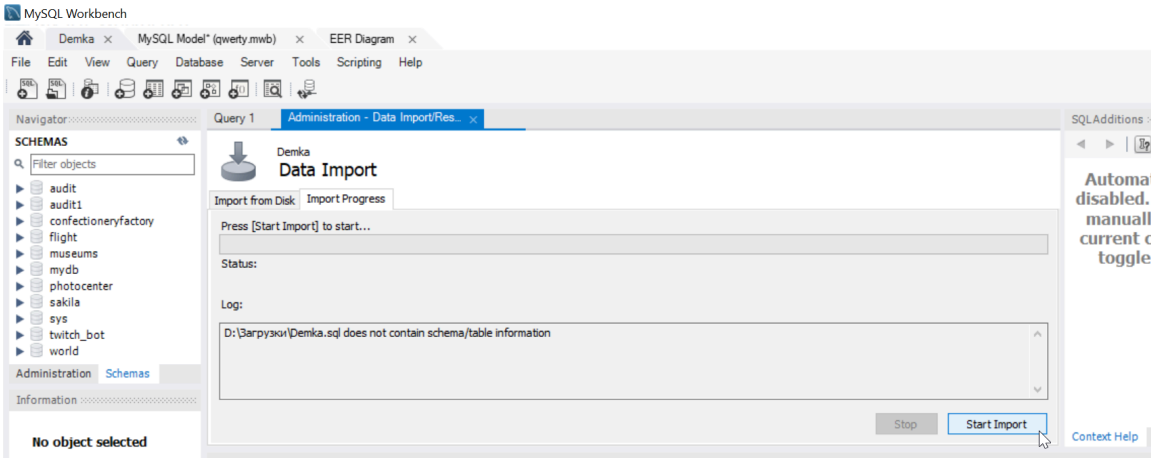

Рисунок 11 – Импорт ER-диаграммы

Обновляем список баз данных и находим нашу БД.

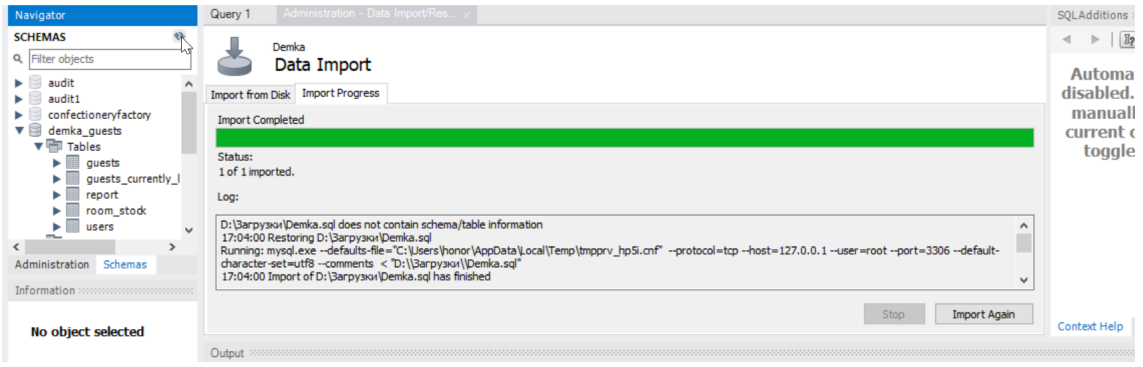

Рисунок 12 – Импорт ER-диаграммы

Далее заполняем таблицы, т.е. импортируем данные из файла "Номерной фонд.xlsx".

Необязательное задание (выполнять в последнюю очередь)

Создаем запрос вычисляющий процент загрузки номерного фонда – это отношение количества проданных ночей к общему количеству номеров в отеле.

Модуль №3 Проектирование и разработка информационных систем

Текст задания: Для выполнения задания рекомендуется создать в базе данных таблицу "Пользователи". Если такая таблица уже существует, необходимо внести некоторые изменения для реализации дальнейшего функционала приложения.

Разработайте форму для авторизации зарегистрированных пользователей с ролями "Администратор" и "Пользователь". Форма должна содержать поля текстовые поля логин, пароль и кнопку "Войти". Поля "Логин" и "Пароль" должны быть обязательными для заполнения. При неверно введенных данных, пользователь должен получить сообщение об ошибке "Вы ввели неверный логин или пароль. Пожалуйста проверьте ещё раз введенные данные".

После успешной авторизации пользователь должен получить сообщение "Вы успешно авторизовались".

При аутентификации связка «логин/пароль» должна совпадать с одной из записей в таблице "Пользователи".

При первой успешной авторизации по выданному паролю администратором должна выводится форма для смены пароля. Форма должна включать текущий пароль, новый пароль, подтверждение нового пароля. Все поля обязательные для заполнения. После заполнения формы и нажатия кнопки "Изменить пароль", система должна проверить правильность введенного текущего пароля и совпадение нового пароля с подтверждением.

В случае ошибок при заполнении формы пользователю должно выводиться сообщение об ошибке. В случае успешного изменения пароля, пользователю должно выводиться сообщение об успешной смене пароля.

Если в течении 3-х раз подряд был неверно введен логин/пароль, то учетная запись блокируется и при повторном авторизации должно появляться сообщение "Вы заблокированы. Обратитесь к администратору".

Также учетная запись должна блокироваться если пользователь не авторизовался в течении 1 месяца.

На рабочем столе пользователя с ролью "Администратор" предусмотрите функционал для добавления новых пользователей, изменения данных текущих пользователей (включая снятие блокировки). При добавлении нового пользователя следует проверять его наличие в базе данных. В случае, если пользователь с указанным логином уже существует, должно выводиться соответствующее сообщение.

Графический интерфейс необходимо разработать в соответствии с требованиями к разработке.

Разработайте проектную документацию на разработанный функционал. Включите описание функционального назначения, используемые методы с указанием параметров **(см. Модуль №3.1 Описание программы)**.

Необходимые приложения: Требования к разработке.pdf

Выполнение:

Разработка экранных форм

1. В MS VS создать проект на WinForms и .NET

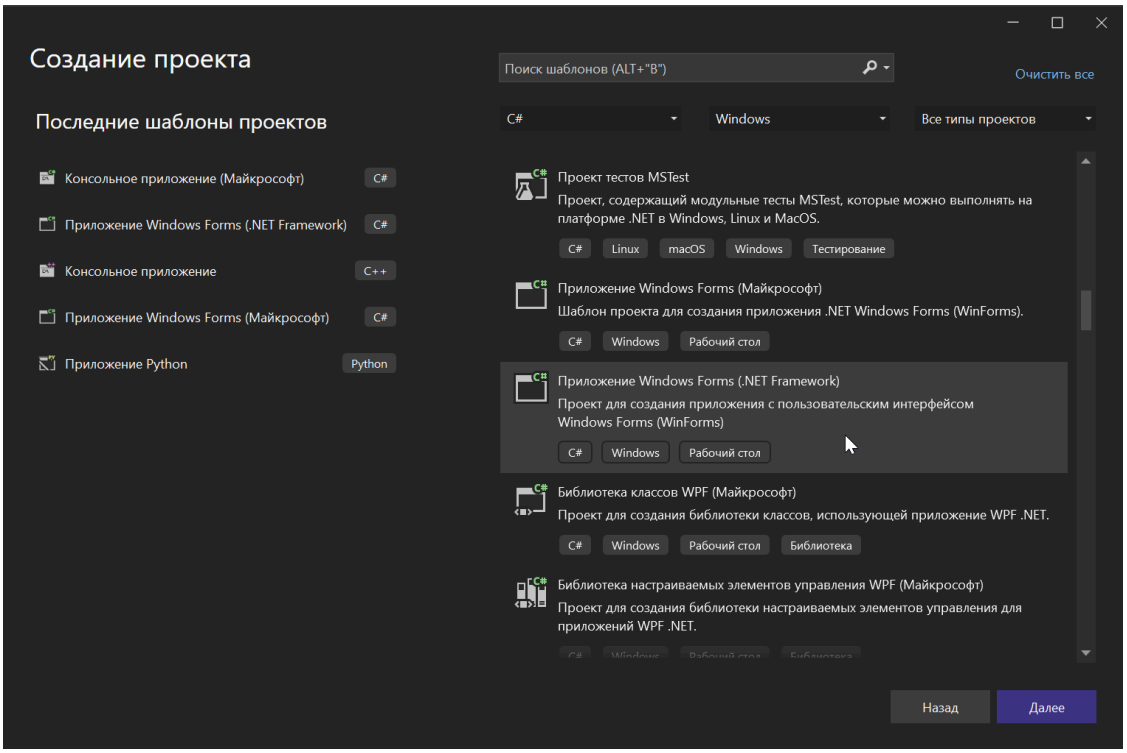

Рисунок 13 – Создание проекта

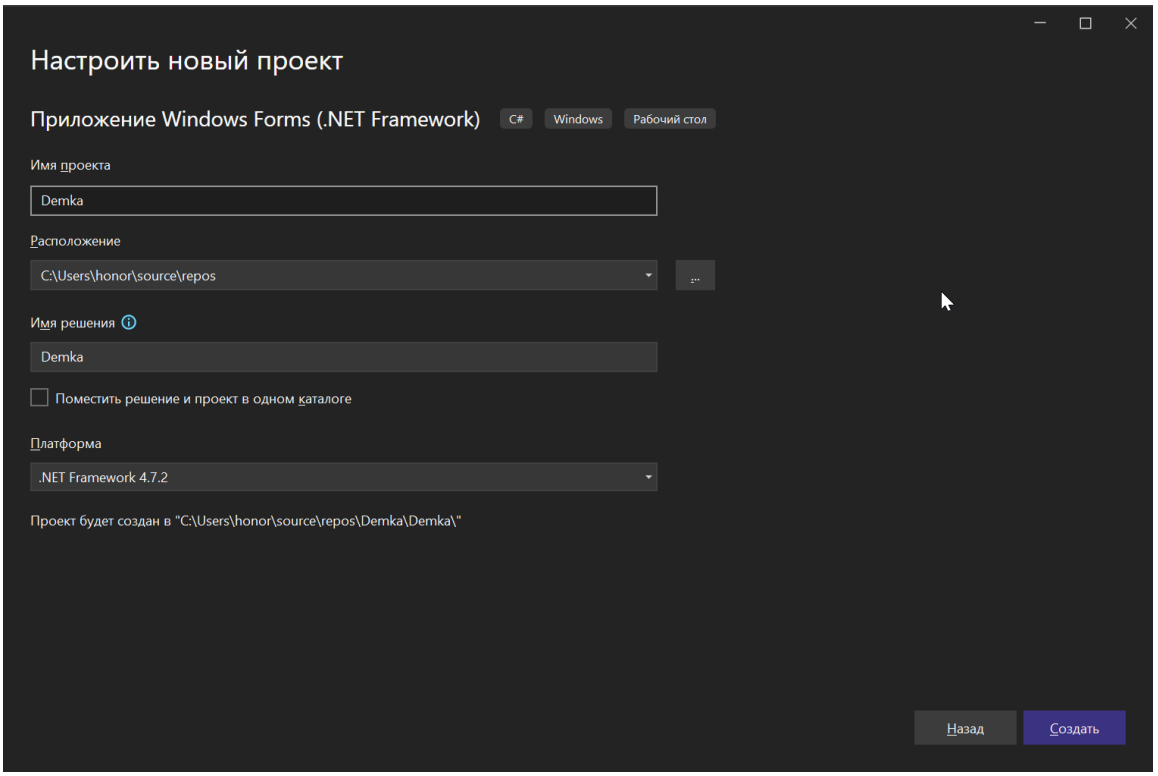

Рисунок 14 – Настройка проекта

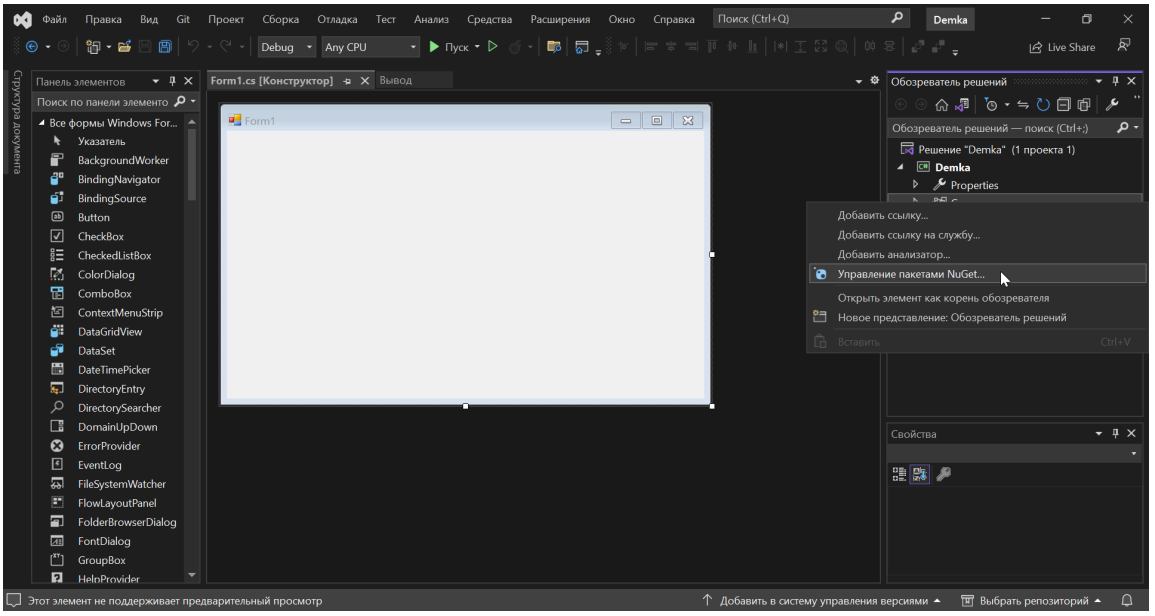

Рисунок 15 – Проверка настройки пакетов MySQL

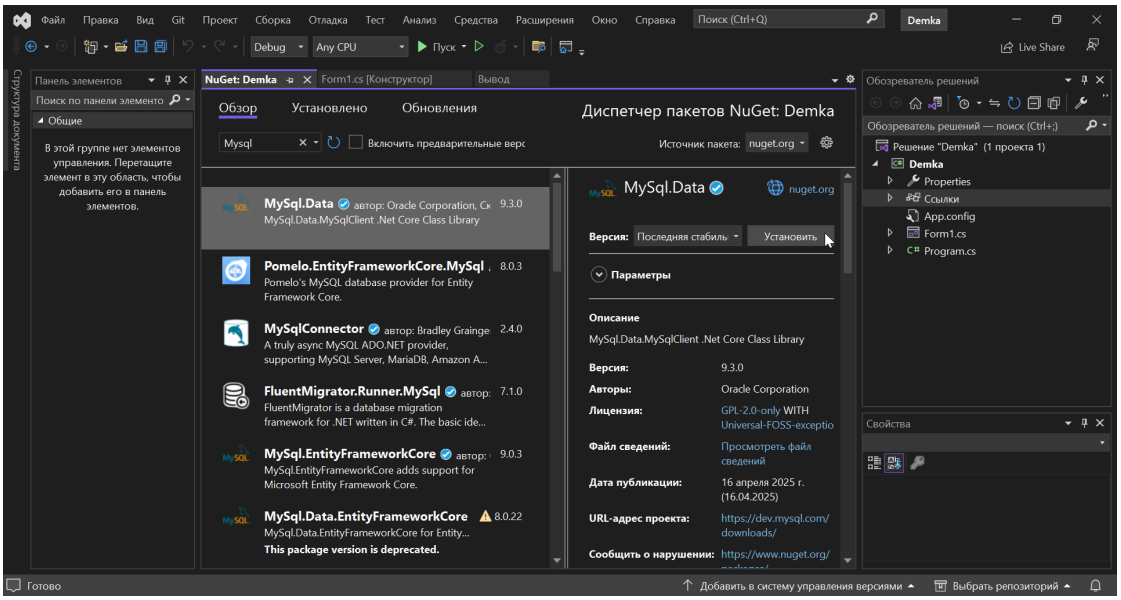

Рисунок 16 – Проверка настройки пакетов MySQL

2. Создать главное окно и меню вызова форм(menustrip)

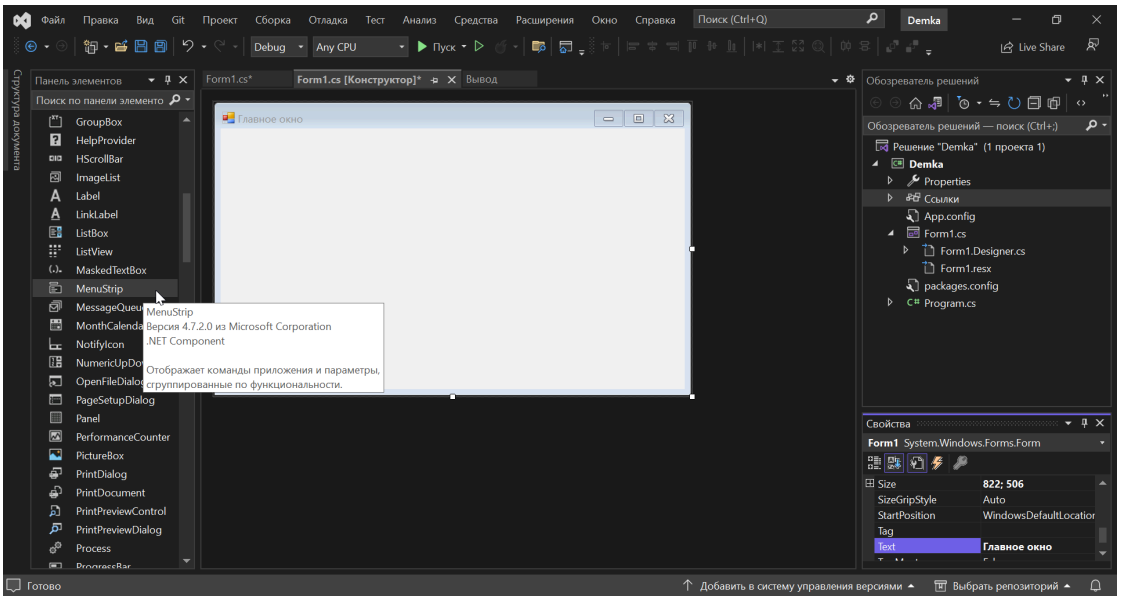

Рисунок 17 – Создание главного меню

3. В соответствии с заданной предметной областью разработать формы (с таблицами DataGrid соответствующими таблицам БД)

Разработка ИС

1. Связать графический интерфейс с БД.

Пример кода, обязательно использовать директиву using MySql.Data.MySqlClient;:

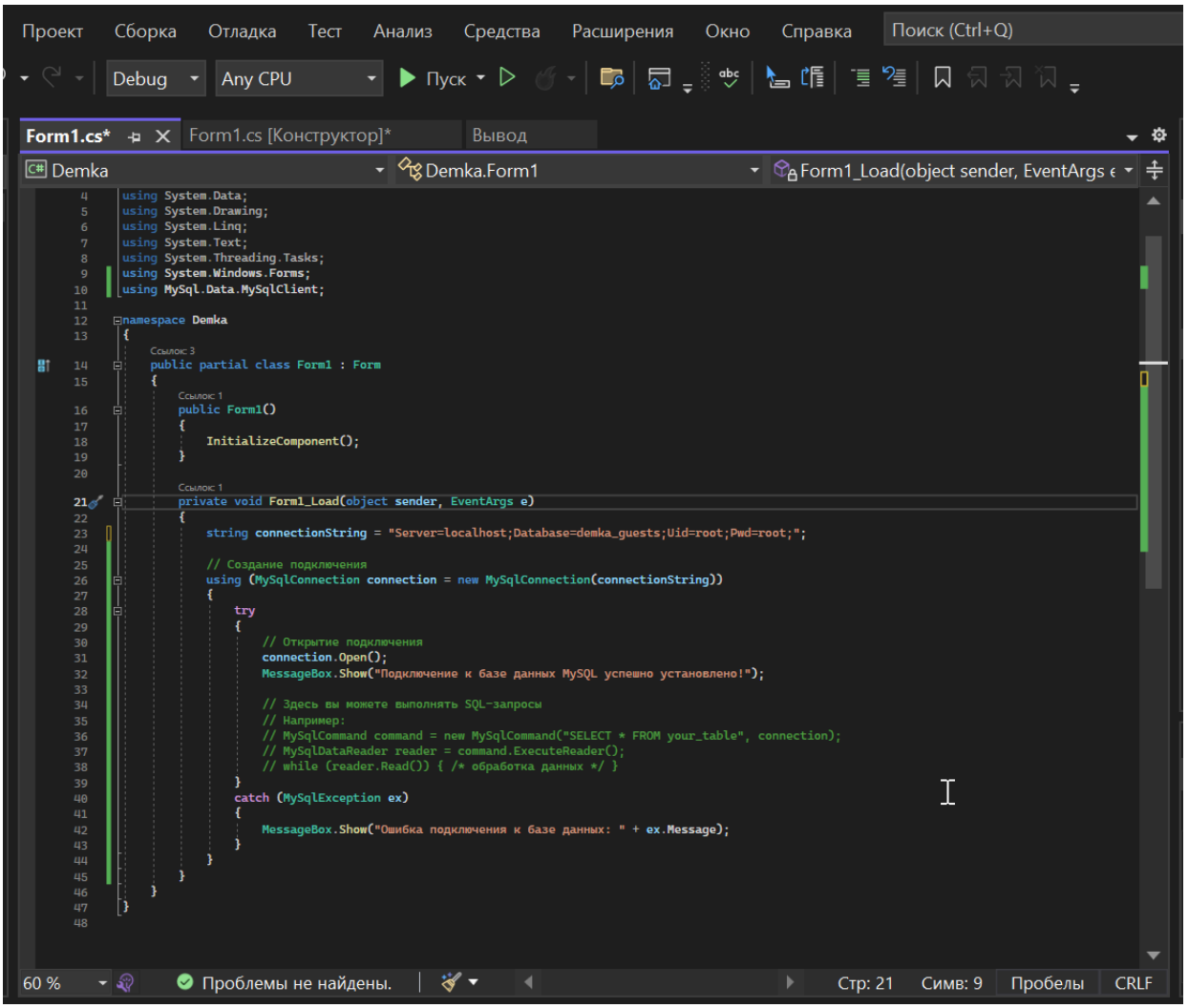

Рисунок 18 – Пример кода

Листинг 2. Подключение БД

```
1  private void Form1_Load(object sender, EventArgs e)
2        {
3            string connectionString =
4  "Server=localhost;Database=demka_guests;Uid=root;Pwd=root;";
5
6            // Создание подключения
7            using (MySqlConnection connection = new MySqlConnection(connectionString))
8            {
9               try
10              {
11                  // Открытие подключения
12                  connection.Open();
13                  MessageBox.Show("Подключение к базе данных MySQL успешно
14 установлено!");
15
16                  // Здесь вы можете выполнять SQL-запросы
17                  // Например:
18                  // MySqlCommand command = new MySqlCommand("SELECT * FROM
19 your_table", connection);
```

```
20                  // MySqlDataReader reader = command.ExecuteReader();
21
22              }
23              catch (MySqlException ex)
24              {
25                  MessageBox.Show("Ошибка подключения к базе данных: " +
26  ex.Message);
27              }
28          }
29      }
```

2. Обработка окна авторизации

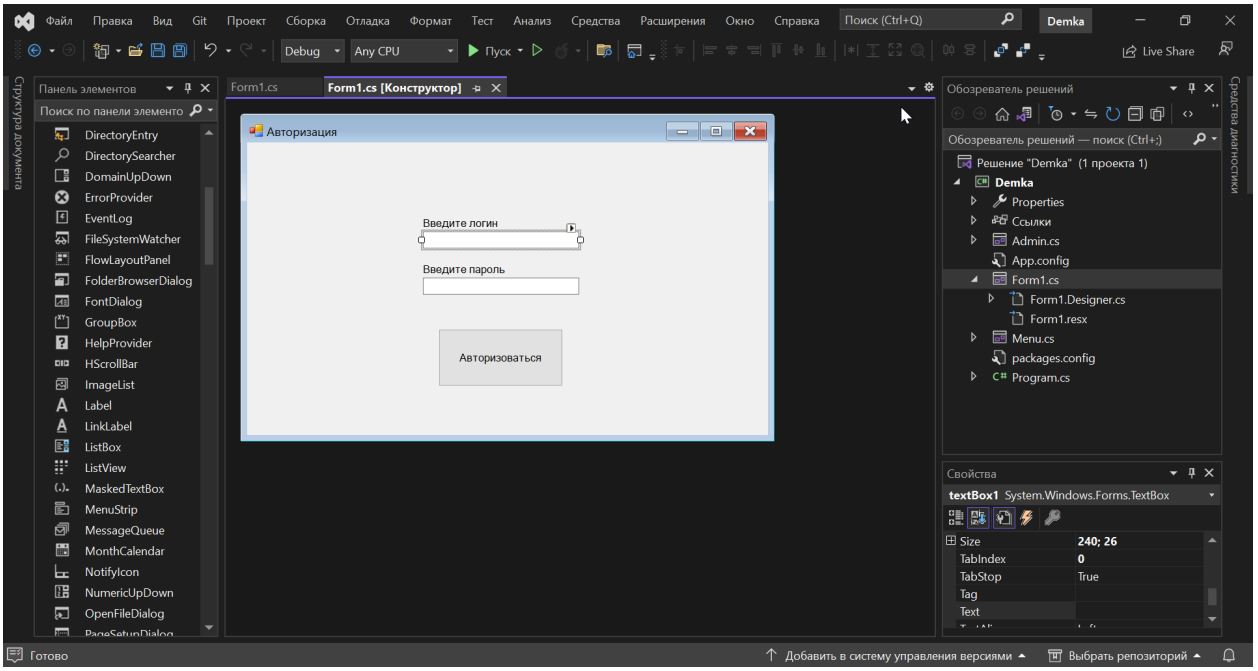

Рисунок 19 – Окно авторизации

3. Листинг 3. Пример кода для обработки окна авторизации

```
1   using System;
2   using System.Collections.Generic;
3   using System.ComponentModel;
4   using System.Data;
5   using System.Drawing;
6   using System.Linq;
7   using System.Text;
8   using System.Threading.Tasks;
9   using System.Windows.Forms;
10  using MySql.Data.MySqlClient;
11  using static System.Windows.Forms.VisualStyles.VisualStyleElement;
12
13  namespace Demka
14  {
15      public partial class Form1 : Form
```

```
    {
        // Строка подключения к базе данных MySQL
        private string connectionString =
"Server=localhost;Database=demka_guests;Uid=root;Pwd=root;";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // Инициализация при загрузке формы
        }

        // Обработчик нажатия кнопки входа
        private void button1_Click(object sender, EventArgs e)
        {
            //Обработка авторизации пользователя
        }
        // Аутентификация пользователя
        private bool AuthenticateUser(string login, string password, out bool isAdmin)
        {
          //Оформление логики авторизации пользователя и администратора
        }
        // Увеличение счетчика попыток входа
        private void IncrementLoginAttempts(string login)
        {
          //Оформление логики счетчика попыток входа пользователя
        }
        // Блокировка пользователя
        private void BlockUser(string login)
        {
            //Оформление логики удаления пользователя
        }

        // Открытие панели администратора
        private void ShowAdminPanel()
        {

        }

        // Открытие главного окна пользователя
        private void ShowMainForm()
        {

        }
    }
}
```

Модуль №3.1 Описание системы

Текст задания: Разработайте проектную документацию на разработанный функционал. Включите описание функционального назначения, используемые методы с указанием параметров.

«Общее описание системы» по ГОСТ 34

Выполнение:

В Приложении А представлен пример документа “Описание программы”. Его можно взять за основу для разработки требуемой по заданию документации.

Для задания также подойдет (или можно сделать дополнительно) руководство пользователя.

1. Создать документ в Google или Word
2. Сделать титульник с названием (см. Приложение А)
3. Сделать скрины экранных форм , вставить в док и описать

Модуль №4 Осуществление интеграции программных модулей

Текст задания: Для проверки данных от клиентов разработайте приложение, которое позволит провести валидацию на корректность данных. Результат проверки необходимо фиксировать в документе ТестКейс.docx.

Сначала заполните в документе ТестКейс.docx столбец "Действие" и "Ожидаемый результат" используя предоставленный текстовый редактор. Добавьте закладки в столбец "Результат". Необходимо провести валидацию ФИО клиента на вхождение запрещенных символов. Проверьте два любых критерия.

Для эмуляции отправки данных от клиента Вам необходимо запустить приложение TransferSimulator.exe. Методы эмулятора описаны в файле api_info.pdf. Макет формы представлен на рисунке.

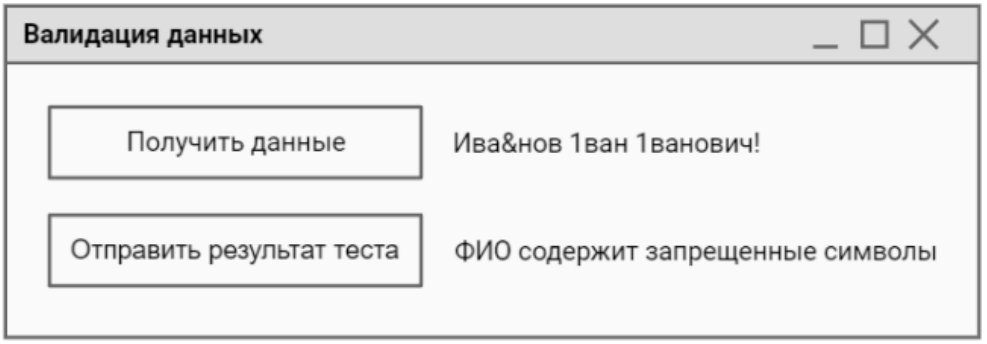

Рисунок 20 - Макет окна приложения валидации данных

При нажатии на кнопку "Получить данные" данные загружаются с эмулятора и отображаются на форме. После нажатия на кнопку "Отправить результат теста" происходит проверка данных по заполненному шаблону, и результат проверки отображается на форме и в соответствующей строке таблицы в столбце "Результат". Важно: Разрабатывать API Вам не нужно. Используйте предоставленный API из приложения.

Необходимые приложения: api_info.pdf, TransferSimulator.exe, ТестКейс.docx, Требования к разработке.pdf, Настройка ПК для эмулятора.pdf

Выполнение:

Тестирование по заданию 2024 года

1. Выбрать вариант тестирования

2. Создать таблицу см. пример на рисунке ниже (но у вас свои тест-кейсы должны быть). Описать 2-3 тест-кейса

Таблица 1. Тест-кейсы

Дата	Номер тест-кейса	Заголовок	Предусловие	Шаги проверки	Ожидаемый результат	Фактический результат	Статус

	Дата	Номер тест-кейса	Заголовок	Предусловие	Шаги проверки	Ожидаемый результат	Фактический результат	Статус
2	04.06.2023	1	Нажмите кнопку "Показать"	Проверка на показ эффективности в приложении	Нажмите кнопку "Показать"	Открытие в таблице напротив имени в строке эффективности сотрудника		
3	04.06.2023	2	Нажмите кнопку "Отменить"	Проверка на отмену действий	Нажмите кнопку "Отменить"	Должен отобразиться изначальный вариант страницы		
4	05.06.2023	3	Вызов формы	Проверка вызова формы	Нажмите кнопку "Показать\Отменить"	Должна появиться новая форма		
5		4	Добавление информации с помоощью БД	Проверка на добавление в форму через БД		В форме должна появиться информация		

Рисунок 21 – Пример заполненной таблицы

Тестирование по заданию 2025 года:

Необходимо разработать код для работы с приложением TransferSimulator.exe. Выполнить задания по тексту (см. Текст задания **Модуля №4**).

ПРИЛОЖЕНИЕ А

УТВЕРЖДЕН

….-01 13 01-ЛУ

СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

АРМ

Описание программы

…-01 13 01

Страниц 13

Литера

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

…-01 13 01

Разработал: Пушкарская Е.В.

АННОТАЦИЯ

Документ содержит принципы построения специального программного обеспечения для …….. В документе приведены следующие сведения:

- функциональное назначение программы;
- описание логической структуры программы;
- требования к используемым техническим средствам;
- вызов и загрузка программы;
- входные данные программы;
- выходные данные программы.

ПО разработано на языке C# .

В качестве среды разработки использована среда MS VS.

ПО обеспечивает пользовательский интерфейс для управления техническими средствами приемного центра, отображения их состояния, обмен данными по протоколу межмодульного обмена с программным модулем …., взаимодействие с сервером баз данных.

…-01 13 01

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Обозначение и наименование программы

Наименование программы – …..

Обозначение программы – …….

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

ПО выполняет следующие функции:

1) контроль своевременного выполнения задач по связи дежурным расчетом/аппаратной;

2) отображение состояния сеансов на визуальных формах;

3) отображение состояния технических средств на визуальных формах;

4) ввод и редактирование данных о составе средств;

5) ввод и редактирование управляемых параметров технических средств;

6) идентификации и аутентификации пользователей путем авторизации

……..

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.2.1. Основные классы

Необходимые для программы классы и их исходные файлы представлены в таблице 1.

Таблица 1

Наименование класса	Файл объявления(*.h)	Файл описания(*.cpp)
QAbout	about	about
ArhivSeans	arh_seans	arh_seans
BaseClassDlg	baseclass.h	baseclass.cpp
cBase_interf	Base_interf.h	base_interf.cpp
BaseEditDlg	baseeditdlg.cpp	baseeditdlg.cpp

3.2.2. Описание классов

Для запуска приложения используется функция main, которая принимает аргументы приложения. В качестве первого аргумента передается IP-адрес сервера, на котором находится база данных ПРЦ (далее БД). Все остальные настройки программы берутся из базы данных.

Функция main прежде всего выполняет соединение с базой данных MySQL. Если соединение с базой данных по каким-то причинам не произошло (не тот IP-адрес, либо база не установлена, либо не работает СУБД MySQL), функция main выдает сообщение об этом и завершает свою работу.

Иначе, функция далее выполняет следующие действия:

- создает объекты данных для дальнейшего управления;
- запускает главное окно приложения ArmNdrNa.

Далее приводится краткое описание основных классов программы.

Класс SqlDatabase обеспечивает соединение с базой данных PSQL и предоставляет функции для работы с БД.

Класс ManageApp на основе содержимого БД создает список объектов данных, которые являются основными элементами программы и модели данных,

…-01 13 01

включающие в себя разные наборы объектов данных.

Класс MainWindow обеспечивает работу главного окна приложения, которое включает в себя обработку команд меню, визуальное отображение состояния технических средств комплекса, вызов диалоговых форм управления техническими средствами.

Класс ObjectData имеет несколько конструкторов и может создавать объекты с типом, номером и данными, структурированными в XML-формат.

Класс ObjectManage является потомком ObjectData и является основным классом программы, так как выполняет важнейшие функции:

- содержит статический указатель на класс SqlDatabase, который используется во всех файлах программы;
- инициализирует объект, загружая его данные из БД (функция loadObjFromDB);
- при необходимости создает диалоговую форму для объекта данных, используя указатель на класс BaseClassDlg;
- обновляет данные объекта при их изменении в процессе работы, в том числе и на диалоговой форме.

Класс BaseClassDlg создает базовую диалоговую форму для управления ТС, в которой определяется основной стиль внешнего вида и поведения окна при его вызове и в процессе выполнения команд.

Классы QAK, QBFM, QDUKK, QPK, QMRPU, QUFKS, QUFSOCH, QSEV, QMRPU являются потомками BaseClassDlg. Они создают диалоговые формы управления техническими средствами ПРЦ.

Класс DlgMainWin – базовый класс для создания диалоговых форм с наполнением в зависимости от заданного делегата. В данном случае реализовано отображение графики и отображение моделей данных в виде элементов – потомков класса QWidget.

Класс BaseClassKmd обеспечивает процесс отправки и выполнения

команд, отображения результатов выполнения на соответствующих диалогах управления средствами.

Класс BaseThread – базовый класс создания потоков в программе, а класс DemonThreads осуществляет выполнение потоков из очереди потоков.

Класс Magazine – шаблонный класс очереди, который обеспечивает добавление и выборку элементов очереди.

Класс Socket – класс на основе QUdpSocket осуществляет прием и передачу данных через заданный порт.

По командам меню Данные вызываются классы: QEditAdrLIS, QEditLM, QEditOper, QEditPRA, QEditRDM, QEditRPDU, QEditWP, которые отвечают за редактирование определенных таблиц базы данных.

6. ВЫЗОВ И ЗАГРУЗКА

Запуск программы осуществляется с помощью ярлыка «АРМ ….», расположенного на рабочем столе.

…-01 13 01

После запуска приложения появится диалоговое окно регистрации пользователя. После ввода в данном окне логина и пароля выполняется процедура аутентификации пользователя и, в случае успешного ее выполнения, загружается главное окно программы…...