Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

**КУРСОВОЙ ПРОЕКТ**

Дисциплина: **Базы данных**

Тема: **«База данных для ведения табеля учета рабочего времени офисного персонала»**

Вариант 6

Выполнил: студент гр. КТ-41-21

Иванов И.С.

Руководитель

к.т.н., доцент Димитриев А.П.

Оценка«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

\_\_\_\_\_\_\_\_\_\_\_Димитриев А.П.

Дата: \_\_.\_\_\_\_\_\_\_\_\_2023

№\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_

Чебоксары 2023

Министерство науки И ВЫСШЕГО образования

Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

УТВЕРЖДАЮ

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Зав. кафедрой

компьютерных технологий Т.А. Лавина

**ЗАДАНИЕ**

**на курсовое проектирование**

студенту группы КТ-41-21 Иванову И.С.

1. Тема проекта (работы)

**База данных для ведения табеля учета рабочего времени офисного персонала**

2. Срок сдачи студентом законченного проекта (работы)

29.12.2023 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к проекту (работе)

3.1. СУБД – SQL Server ,

3.2. Операционная система Windows 10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

4.1. Обзор и анализ технических решений\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2. Проектирование и создание таблиц\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.3. Проектно-конструкторская часть -

4.4. Технологическая часть\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перечень графического материала (с точным указанием обязательных чертежей)

* ER-диаграмма базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Консультанты по проекту (с указанием относящихся к ним разделов проекта)

Руководитель – доцент Димитриев А.П.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Дата выдачи задания

16.09.2023 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (подпись студента)

**Оглавление**

[Введение 6](#_Toc153477791)

[1 Концепция архитектуры баз данных 6](#_Toc153477792)

[1.1 Базы данных 7](#_Toc153477793)

[1.2 Модель «сущность-связь» 7](#_Toc153477794)

[1.3 Информация в базе данных «Базы данных для ведения табеля учета рабочего времени офисного персонала» 9](#_Toc153477795)

[2 Проектирование и создание таблиц 10](#_Toc153477796)

[2.1 Структуры таблиц 10](#_Toc153477797)

[2.2 Связывание таблиц 12](#_Toc153477798)

[2.3 Заполнение таблиц 16](#_Toc153477799)

[3 Создание и запуск базовых запросов 20](#_Toc153477800)

[3.1.Общие сведения 20](#_Toc153477801)

[3.2 Запросы 20](#_Toc153477802)

[3.2.1 Запросы на заполнение таблиц (INSERT) 20](#_Toc153477803)

[3.2.2 Запросы на выборку (SELECT) 21](#_Toc153477804)

[3.2.3 Запросы на обновление данных (UPDATE) 23](#_Toc153477805)

[3.2.4 Запросы на удаление данных (DELETE) 25](#_Toc153477806)

[4 Создание и запуск усложненных запросов SQL 27](#_Toc153477807)

[4.1 Общие сведения 27](#_Toc153477808)

[4.2 Запросы 27](#_Toc153477809)

[4.2.1.Запросы изменения таблиц (Alter Table) 27](#_Toc153477811)

[4.2.2 Запрос удаления таблиц (DROP TABLE) 28](#_Toc153477812)

[5 Создание форм для заполнения базы данных 30](#_Toc153477813)

[5.1 Форма для изменения данных в таблицах 30](#_Toc153477814)

[Заключение 34](#_Toc153477815)

[Список литературы 35](#_Toc153477816)

[Приложение А 36](#_Toc153477817)

[Приложение Б 37](#_Toc153477818)

# Введение

В современном мире базы данных играют важную роль в развитии технологий. Люди ежедневно взаимодействуют со множеством окружающих предметов, и ни один из них не хранится в БД. Однако в БД могут храниться данные о некоторых из этих предметов. Сегодня сложно представить полноценную информационную систему, которая не использует базы данных. Базы данных создаются и управляются специальными программами, называемыми системами управления базами данных. Использование баз данных позволяет систематизировать информацию и эффективно организовывать работу.

Базы данных можно описать как массивы данных, которые являются моделями для информационных систем, основанных на данных [1].

Цель данного курсового проекта заключается в том, чтобы получить навыки проектирования баз данных, на примере базы данных для учета рабочего времени офисного персонала. В этой базе данных будет храниться информация о количестве отработанных часов персоналом офиса.

Задачи проекта включают исследование предметной области, разработку схемы базы данных, создание и заполнение базы данных в программном обеспечении для управления базами данных, а также разработку формы просмотра таблиц в Visual Studio.

Объектом исследования является учет рабочего времени офисного персонала, а предметом исследования разработки является база данных для этого учета.

1 Концепция архитектуры баз данных

## Базы данных

База данных является совокупностью данных, отражающих состояние объектов и их отношения в определенной предметной области. Одной из характерных черт БД является их постоянство, поскольку данные накапливаются и используются постоянно.

Существует множество систем управления базами данных (СУБД). Они могут использовать разные подходы к работе с данными и предоставлять различные функции, многие СУБД основаны на одних и тех же основных понятиях.

В современной технологии баз данных предполагается, что создание, поддержка и доступ к БД осуществляются централизованно с помощью специальной системы управления базами данных.

СУБД – это набор языковых и программных средств, предназначенных для создания, управления и совместного использования БД несколькими пользователями [3].

Визуализация базы данных – это процесс выбора, упорядочения и оформления отображаемых данных в соответствии с заданным критерием, а затем их представление на устройствах вывода или передача по каналам связи [2].

База данных – это четко структурированная и поддерживаемая в памяти компьютера совокупность данных, которая отображает актуальное состояние определенной предметной области и используется для удовлетворения информационных потребностей пользователей. Разные определения базы данных могут быть приведены в зависимости от контекста и предметной области, в которой используется термин [4].

Архитектура СУБД состоит из основных функциональных компонентов СУБД, средств взаимодействия между ними, пользователями и системным персоналом. Она описывается различными уровнями абстракции [5].

Архитектура системы баз данных сильно зависит от основной компьютерной системы, на которой она работает. Системы баз данных могут быть централизованными или клиент-серверными, где одна серверная машина выполняет работу от имени нескольких клиентских машин. Также системы баз данных могут быть адаптированы для использования параллельных компьютерных архитектур.

## 1.2 Модель «сущность-связь»

Схема «сущность-связь» (также ERD или ER-диаграмма) – это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы [6].

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. Entity – Relationship diagram, ERD, ER-диаграмма).

Понятия «ER-модель» и «ER-диаграмма» часто не различают, хотя для визуализации ER-моделей могут быть использованы и другие графические нотации, либо визуализация может вообще не применяться (например, использоваться текстовое описание).

Сущность, связь и атрибут – это основные термины ER-модели.

Сущность (entity) – это предмет, который может быть идентифицирован некоторым способом, отличающим его от других предметов. Примерами сущности являются конкретный человек, компания или событие. Подразделяется сущность на тип сущности и экземпляр сущности. Термин «тип сущности» определяет набор однотипных объектов, событий, персон, выступающих как одно целое. Экземпляр сущности соотносится с определенным объектом в наборе [7].

Атрибут: информацию о сущности или связи получают путем наблюдения или измерения и выражают множеством пар атрибут-значение. Любой атрибут имеет имя, в рамках своей сущности.

Связь – это ассоциация, устанавливаемая между сущностями. Например, отец-сын – это связь между двумя сущностями «Человек». Многие связи относятся к разряду бинарных и имеют место между парой сущностей [8].

Существуют связи трех разновидностей [9]:

– Связь «один-к-одному». Таблицы будут связаны «один-к-одному» тогда, когда одному объекту таблицы А соответствует один объект таблицы Б, и одному объекту таблицы Б соответствует один объект таблицы А. Часто, если используется этот тип связи, то имеет смысл объединить две таблицы в одну.

– Связь «один-ко-многим». Здесь один элемент первой сущности может быть связан с более чем одним элементом второй сущности, но один элемент второй сущности связан не более чем с одним элементом первой сущности. Первая сущность является в этой связи главной (основной), вторая – зависимой (подчиненной). Связь «один-ко-многим» в реляционных базах данных реализуется тогда, когда объекту А может принадлежать или соответствовать несколько объектов Б, но объекту Б может соответствовать только один объект А[2].

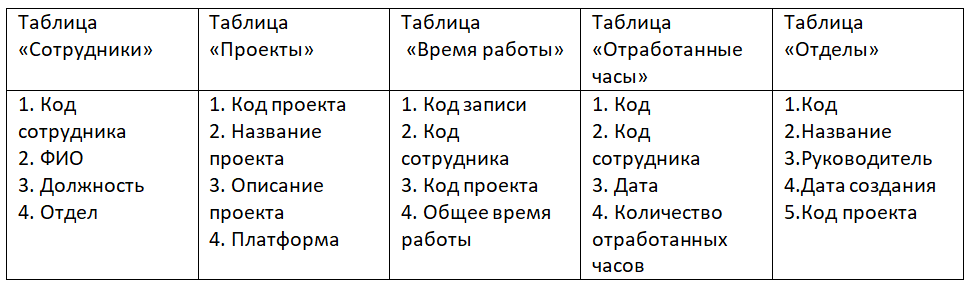
– Связь «многие-ко-многим». Здесь каждый элемент первой сущности имеет возможность быть связанным с рядом элементов второй сущности, а каждый элемент второй сущности – с рядом элементов первой. Связь «многие-ко-многим» реализуется в том случае, когда нескольким объектам из таблицы А может соответствовать несколько объектов из таблицы Б, и в то же время нескольким объектам из таблицы Б соответствует несколько объектов из таблицы А[10].

## 1.3 Информация в базе данных «Базы данных для ведения табеля учета рабочего времени офисного персонала»

Табель учета рабочего времени – это список работников организации, в котором отмечается, какое время они отработали за определенный период. Он является важным документом для составления учета рабочего времени и необходим для практической работы. Табель используется для записи информации о реально отработанном времени сотрудниками, контроля соблюдения рабочего режима и получения данных о рабочем времени в целом.

Задача: создать БД «База данных для ведения табеля учета рабочего времени офисного персонала», содержащую информацию о количестве отработанного времени сотрудниками. БД должна содержать следующую информацию (табл. 1).

Таблица 1 – Содержимое базы данных «База данных для ведения учета рабочего времени офисного персонала»



Выводы:

Рассмотрены концепции БД. Сформулированы требования к информации в БД «База данных для ведения учета рабочего времени офисного персонала».

# 2 Проектирование и создание таблиц

Главные объекты создаваемой БД – таблицы, запросы, формы [4]. Создание таблиц, определение их полей, свойств и их типов данных, а также связей между ними определяет строение разрабатываемой БД, ее схему, или логическую модель данных.

БД может иметь в себе множество таблиц, в которых хранятся данные по различным темам. Каждая таблица может состоять из большого количества полей разного типа.

Во время создания новой БД на компьютере появляется новый файл. Он используется как вместилище для всех объектов в БД, в том числе и таблиц.

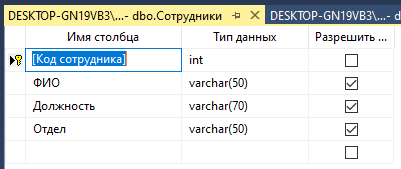
Ключевое поле – это такое поле, которое однозначно определяет каждую запись, внесенную в таблицу с данным полем. Таким образом, создадим в каждой таблице свое ключевое поле с типом данных int.

Для выполнения курсового проектирования используется программа Microsoft SQL Server Management Studio 14, выполняется соединение с сервером. Создается новая база данных, называется: База данных для ведения табеля учета рабочего времени офисного персонала.

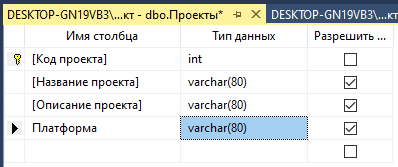
## 2.1 Структуры таблиц

Итак, выполняем переход к этапу создания таблиц.

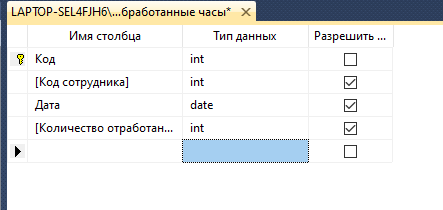
Первая таблица «Сотрудники» (рисунок 1), содержит следующие атрибуты: Код сотрудника, ФИО, Должность, Отдел. Для столбца «Код сотрудника» устанавливается тип данных: int, а также этот столбец объявляется идентификатором и назначается первичным ключом. Для столбца «Должность» устанавливается тип данных: varchar(70), для остальных столбцов устанавливается тип данных: varchar(50). После создания таблицы она сохраняется и ей присваивается название: Сотрудники.

  
Рисунок 1 – Фрагмент экранной копии, демонстрирующий процесс создания таблицы «Сотрудники*»*

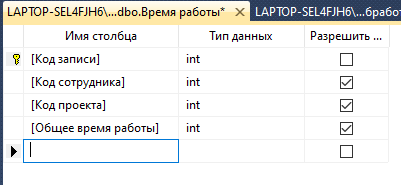
Вторая таблица «Проекты» (рисунок 2), содержит следующие атрибуты: Код проекта (ключ), Название проекта, Описание проекта, Платформа. Для столбца «Код проекта» устанавливается тип данных: int, а также этот столбец объявляется идентификатором и назначается первичным ключом. Для остальных столбцов устанавливается тип данных: varchar(80). После этого сохраняем таблицу, и ей присваивается название: Проекты.

  
Рисунок 2 – Фрагмент экранной копии, демонстрирующий процесс создания таблицы «Проекты»

Третья таблица «Отработанные часы» (рисунок 3), содержит следующие атрибуты: Код (ключ), Код сотрудника, Дата, Количество отработанных часов. Для столбца «Код» устанавливается тип данных: int, а также этот столбец объявляется идентификатором и назначается первичным ключом. Столбцу «Дата» устанавливается тип данных: date. Для столбца «Количество отработанных часов» устанавливается тип данных: int. После этого сохраняется таблица и присваивается название: Отработанные часы.

  
Рисунок 3 – Фрагмент экранной копии, демонстрирующий процесс создания таблицы «Отработанные часы»

Четвертая таблица «Время работы» (рисунок 4), содержит следующие атрибуты: Код записи, Код сотрудника, Код проекта, Общее время работы. Для столбцов с кодом записи и кодом сотрудника устанавливается тип данных: int, а также назначается первичным ключом столбец «Код записи». Для столбца «Общее время работы» также устанавливается тип данных: int. После этого сохраняется таблица и присваивается название: Время работы.

  
Рисунок 4 – Фрагмент экранной копии, отображающий процесс создания таблицы «Время работы»

Пятая таблица «Отделы» (рисунок 5) содержит следующие атрибуты: Код, Название, Руководитель, Дата создания, Код проекта. Для столбцов «Код» и «Код проекта» устанавливается тип данных: int, а также назначается первичным ключом столбец «Код». Для столбцов «Название» и «Руководитель», устанавливается тип данных: varchar(50). Для столбца «Дата создания» устанавливается тип данных: date.

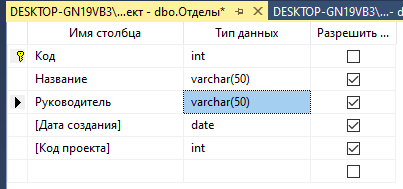
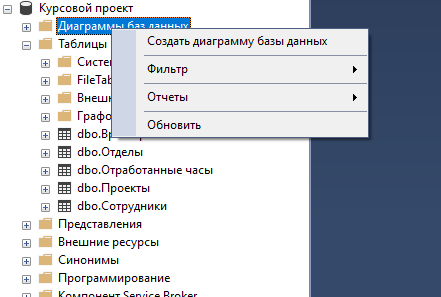


Рисунок 5 – Фрагмент экранной копии, отображающий процесс создания таблицы «Отделы»

## 2.2 Связывание таблиц

Далее необходимо перейти к связыванию таблиц. Для этого разработана ER-диаграмма (см. приложение А).

Создается диаграмма баз данных (рисунок 6).

  
Рисунок 6 – Фрагмент экрана, демонстрирующий создание диаграммы баз данных

Выбираются все таблицы (рисунок 7).

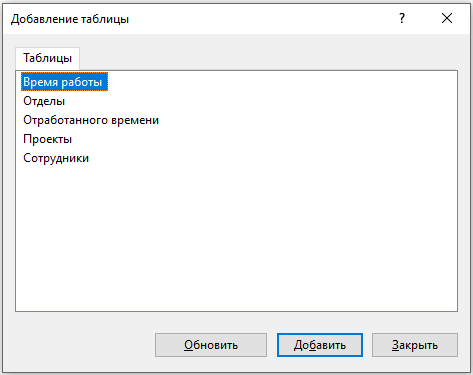


Рисунок 7 – Фрагмент экрана, демонстрирующий добавление таблиц

Соединяется таблица «Сотрудники» (Код сотрудника) с таблицей «Время работы» (Код сотрудника)(рисунок 8).

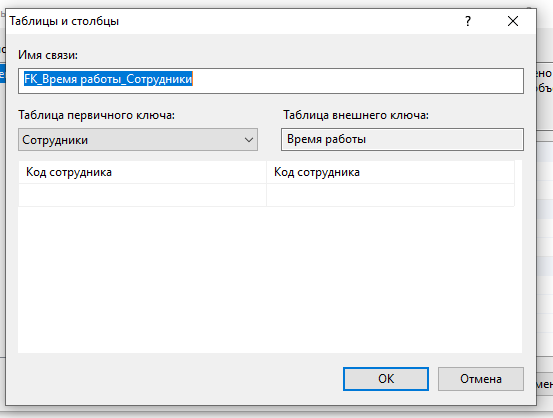


Рисунок 8 – Фрагмент экрана, демонстрирующий соединение таблиц– 1

Соединяется таблица «Проекты» (Код проекта) с таблицей «Время работы» (Код проекта)(рисунок 9).

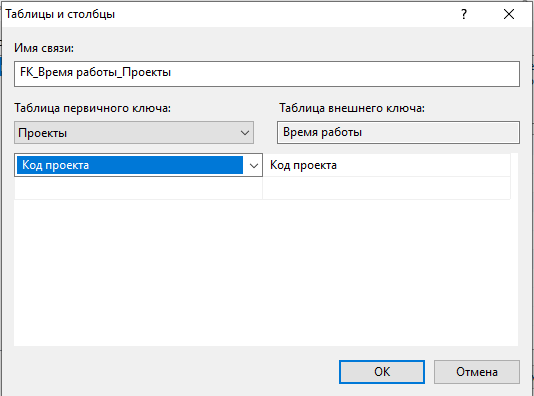


Рисунок 9 – Фрагмент экрана, демонстрирующий соединение таблиц– 2

Соединяется таблица «Сотрудники» (Код сотрудника) с таблицей «Отработанные часы» (Код сотрудника) (рисунок 10).

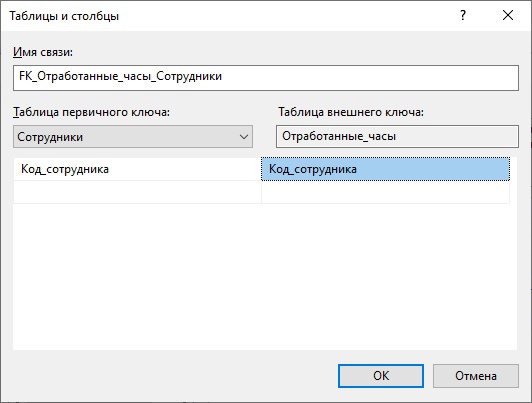


Рисунок 10 – Фрагмент экрана, демонстрирующий соединение таблиц– 3

Соединяется таблица «Отделы» (Код проекта) с таблицей «Проекты» (Код проекта)(рисунок 11).

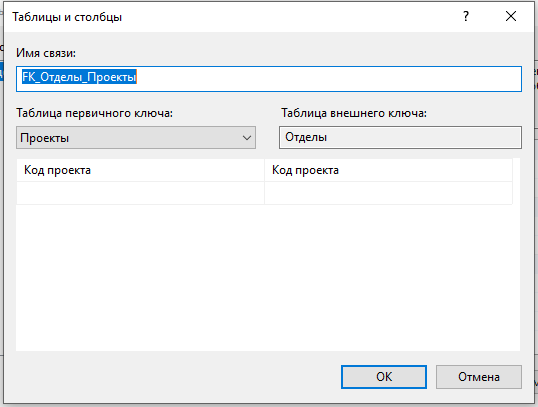


Рисунок 11 – Фрагмент экрана, демонстрирующий соединение таблиц– 4

Теперь на диаграмме (рисунок 12) можно посмотреть, как выглядит связь таблиц.

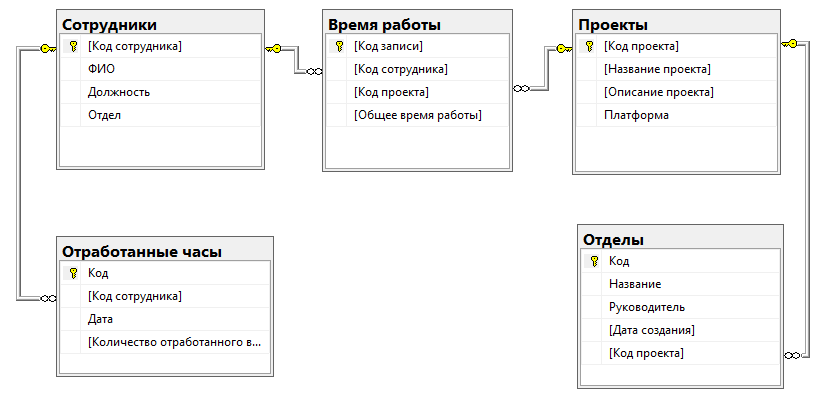
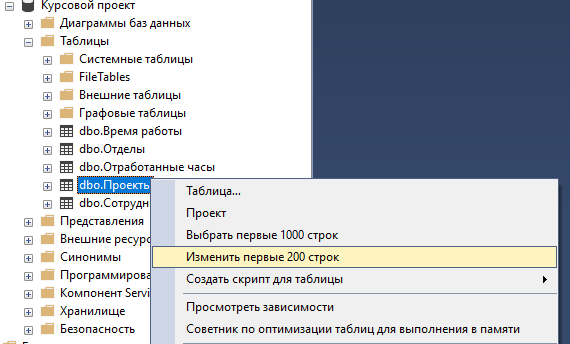


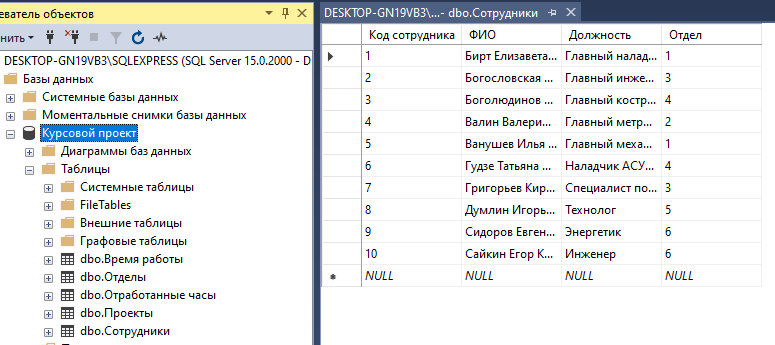
Рисунок 12 – Фрагмент экрана, демонстрирующий связь таблиц

## 2.3 Заполнение таблиц

Схема данных создана, далее заполняются таблицы. Чтобы включить заполнение, выбирается пункт: «Изменить первые 200 строк» (рисунок 11).

  
Рисунок 13 – Фрагмент экрана, демонстрирующий пункт «Изменить первые 200 строк»

Заполняются таблицы: «Сотрудники» (рисунок 14), «Проекты» (рисунок 15), «Отработанные часы» (рисунок 16), «Время работы» (рисунок 17), «Отделы» (рисунок 18).

  
Рисунок 14 – Фрагмент экрана, демонстрирующий результат заполнения таблицы «Сотрудники»

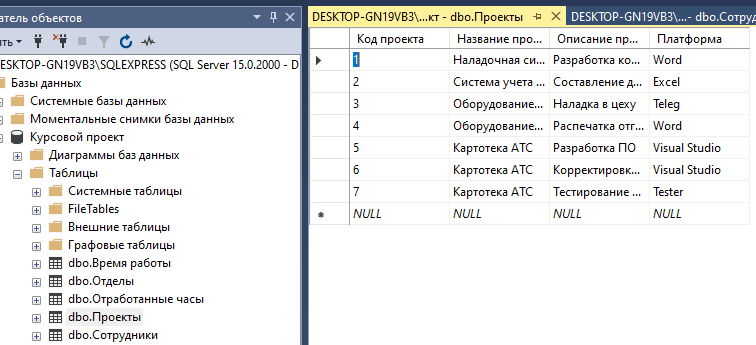


Рисунок 15 – Фрагмент экрана, демонстрирующий результат заполнения таблицы «Проекты»

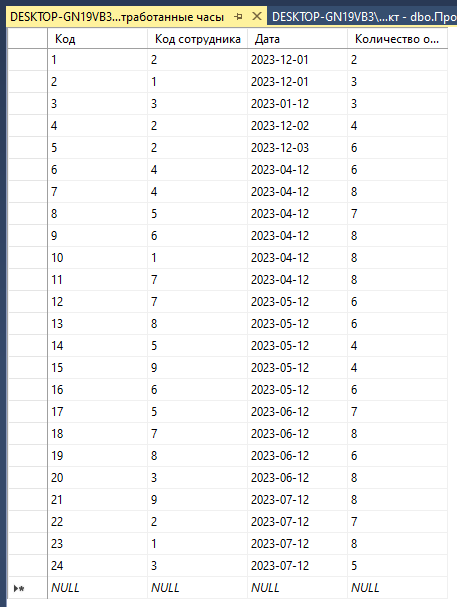


Рисунок 16 – Фрагмент экрана, демонстрирующий результат заполнения таблицы «Отработанные часы*»*

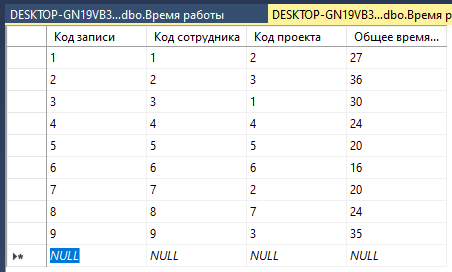


Рисунок 17 – Фрагмент экрана, демонстрирующий результат заполнения таблицы «Время работы»

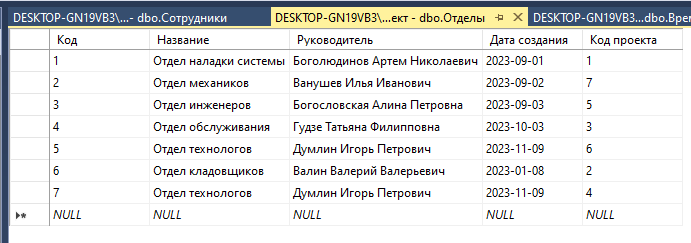


Рисунок 18 – Фрагмент экрана, демонстрирующий результат заполнения таблицы «Отделы»

Вывод: Спроектированы структуры таблиц, связи между ними. Таблицы заполнены данными.

# 3 Создание и запуск базовых запросов

## 3.1 Общие сведения

Как правило, информация в БД, которую необходимо представить в виде отчета, хранится в разных таблицах. Запрос может представлять собой обращение к данным для получения информации из базы данных или выполнения действий с данными. Запрос можно использовать для получения ответа на простой вопрос, выполнения расчетов, объединения данных из разных таблиц, а также для добавления, изменения или удаления данных в таблице [7].

Запросы получают данные из базовых таблиц и создают на их основе временную результирующую таблицу. В данном курсовом проекте предлагается создать несколько запросов: запросы на выборку, запросы с использованием диалогового окна для ввода условия, запросы с постоянно заданным условием.

## 3.2 Запросы

### 3.2.1 Запросы на заполнение таблиц (INSERT)

INSERT– оператор языка SQL, который позволяет добавить строки в таблицу, заполняя их значениями. Синтаксис: INSERT INTO<название таблицы> ([<Имя столбца>]) VALUES (<Значение>,...). Создается запрос на добавление строки в таблицу «Сотрудники» (рисунок 19). Получается результат выполнения запроса (рисунок 20).

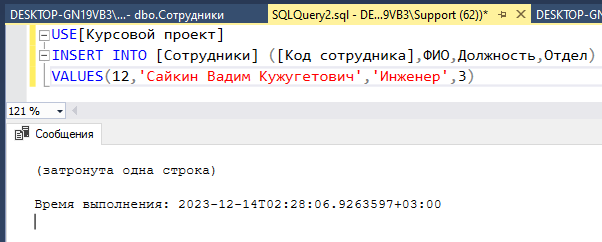


Рисунок 19 – Фрагмент экрана, содержащий листинг запросов с оператором INSERT

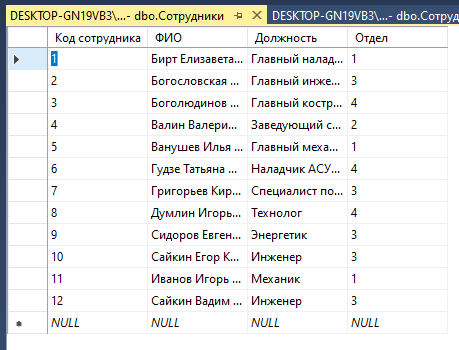


Рисунок 20 – Фрагмент экрана, содержащий результат выполнения запроса с оператором Insert

### 3.2.2 Запросы на выборку (SELECT)

SELECT – оператор языка SQL, возвращающий набор данных (выборку) из базы данных, удовлетворяющих заданному условию. Выполняется запрос на отображение таблицы «Отделы» (рисунок 21).

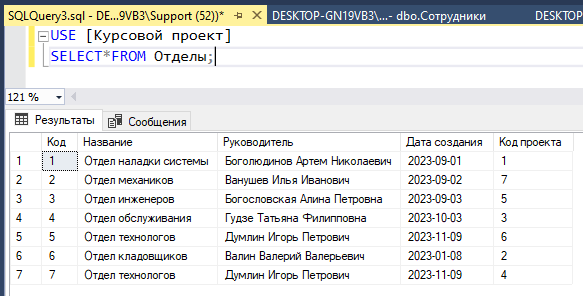


Рисунок 21 – Фрагмент экрана, содержащий листинг запроса Select и результат выполнения

Выполняется запрос для вывода отдельных столбцов из таблицы (рисунок 22).

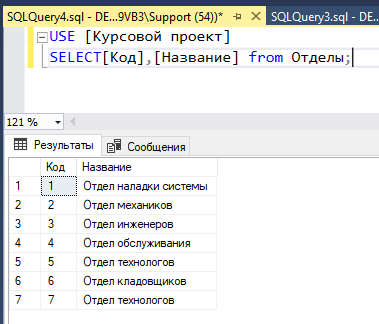


Рисунок 22 – Фрагмент экрана, содержащий листинг запроса Select для столбцов таблицы

При выборке записей из таблицы практически всегда требуется задавать определённые условия, по которым определяется, какие записи нужны, а какие – нет. И именно эти условия можно задавать с помощью конструкции WHERE.

Например, конструкция Where в SQL: Select \* From Table Where count=4;

Логические операции для конструкции WHERE: !, AND, OR, XOR.

Например, нужно отобразить в таблице отделы под номером пять и два, для этого используется конструкция Where и логическая операция or (рисунок 23).

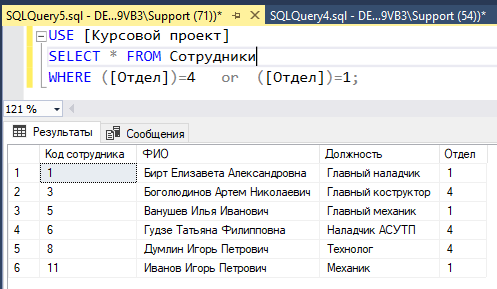


Рисунок 23 – Фрагмент экрана, содержащий листинг конструкции Where и логических операций

Также конструкция WHERE в SQL может содержать предложение LIKE, оно позволяет определить, совпадает ли указанная строка с определённым шаблоном. Из таблицы «Время работы» нужно вывести строки, у которых значение столбца «Общее время работы» оканчивается на 0, для проверки конструкции WHERE (рисунок 24).

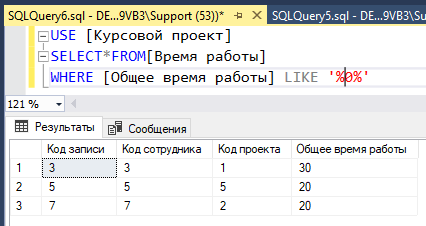


Рисунок 24 – Фрагмент экрана, содержащий листинг конструкции WHERE и LIKE

### 3.2.3 Запросы на обновление данных (UPDATE)

Оператор UPDATE используется для обновления существующих записей в таблицах.

Рассмотрим пример, отображающий, как использовать SQL-оператор UPDATE для обновления одного столбца в таблице. В этом примере применения UPDATE используется таблица «Время работы» со следующими данными (рисунок 25):

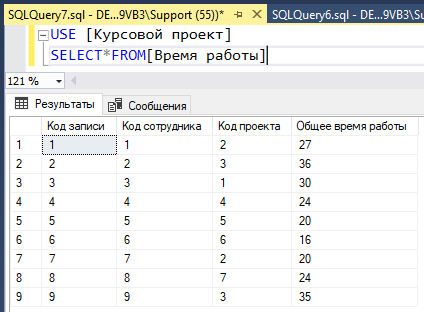


Рисунок 25 – Фрагмент экрана, отображающий содержимое таблицы «Время работы»

Теперь обновляется общее время работы для всех строк из столбца «Общее время работы», в таблице «Время работы». Вводится следующая команда: UPDATE [Время работы] (рисунок 26). Получается результат выполнения запроса (рисунок 27).

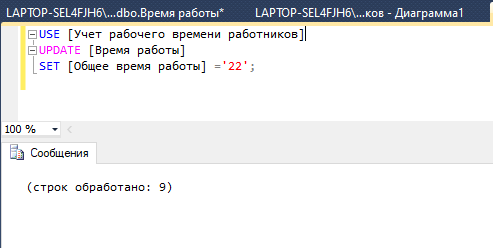


Рисунок 26 – Фрагмент экрана, содержащий листинг оператора Update

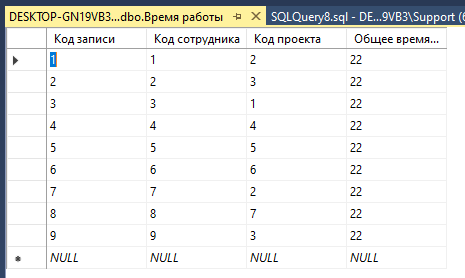


Рисунок 27 – Фрагмент экрана, содержащий результат выполнения запроса Update

###### Установка одинакового общего времени работы для всех сотрудников с помощью выполнения запроса Update может иметь свои причины и преимущества, но также может вызывать определенные проблемы и ограничения. Вот несколько комментариев по этому вопросу:

##### Унификация рабочего времени. Установка одинакового общего времени работы может быть полезной, если компания стремится к унификации рабочего графика и снижению разрывов в рабочем времени между сотрудниками. Это может способствовать более эффективной организации работы, согласованию графиков совместной работы и упрощению планирования.

##### Соблюдение правил и нормативов. В некоторых случаях установка одинакового общего времени работы может быть обусловлена требованиями или нормативами, установленными регулирующими органами или внутренними политиками компании. Например, в некоторых отраслях или регионах могут быть установлены ограничения на рабочее время, и компания может принять решение установить общее время работы, чтобы соблюдать эти правила.

### 3.2.4 Запросы на удаление данных (DELETE)

Оператор DELETE используется для удаления одной или нескольких записей из таблицы. Синтаксис оператора Delete в SQL:

Delete From Table (Удаление записей из нужной таблицы)

[Where Conditions]; (Необязательно. Условие, которое должно быть выполнено для удаления записей. Если никаких условий не предусмотрено, все записи в таблице будут удалены.)

Для примера удаляется проект «Картотека АТС» (рисунок 28). Получим результат выполнения запроса (рисунок 29).

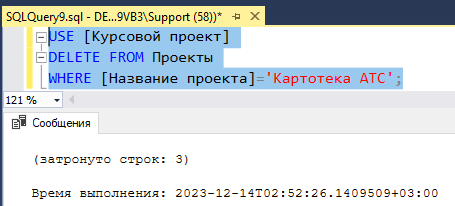


Рисунок 28 – Фрагмент экрана, содержащий листинг оператора Delete

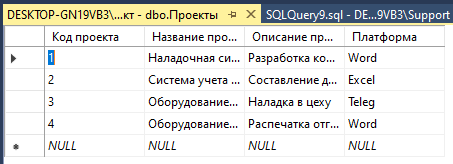


Рисунок 29 – Фрагмент экрана, содержащий результат выполнения запроса Delete

Выводы по разделу

В данном разделе были созданы запросы ввода, вывода, обновления и удаления данных таблиц. Созданные запросы были выполнены, результаты запросов представлены выше.

# 4 Создание и запуск усложненных запросов SQL

## 4.1 Общие сведения

Синтаксис оператора SELECT в предложении FROM допускает указание нескольких таблиц. Чаще всего в объемных базах данных не используются выборки из одной таблицы, т.к. выборки в таком случае, чаще всего, не являются информативными. В данном разделе будут рассмотрены усложненные запросы с операторами CREATE TABLE, DROP TABLE, ALTER TABLE.

## 4.2 Запросы

Оператор CREATE позволяет создавать и определять таблицу.

Например, создадим таблицу «Учет за неделю» (рисунок 30). Данная таблица будет наглядно показывать учет рабочего времени сотрудников, то есть количества часов, за определенную неделю.

## 

Рисунок 30 – Фрагмент экрана, содержащий листинг оператора CREATE TABLE

### 4.2.1 Запросы изменения таблиц (ALTER TABLE)

Оператор ALTER TABLE используется для добавления, изменения или удаления столбцов в таблице. SQL-оператор ALTER TABLE также используется для переименования таблицы.

Добавляется столбец «Даты» в таблицу «Учет за неделю» (рисунок 31).

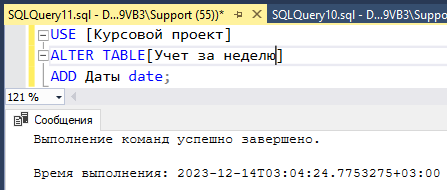


Рисунок 31 – Фрагмент экрана, содержащий листинг оператора ALTER TABLE

Изменяется столбец с помощью оператора ALTER TABLE (рисунок 32).

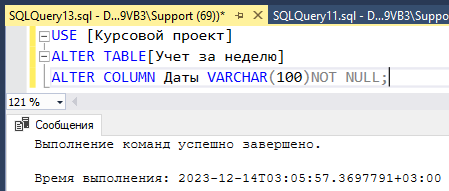


Рисунок 32 – Фрагмент экрана, содержащий листинг оператора ALTER COLUMN

### 4.2.2 Запрос удаления таблиц (DROP TABLE)

Оператор DROP TABLE позволяет удалить таблицу из базы данных SQL. Синтаксис для оператора DROP TABLE в SQL: DROP TABLE Table\_name;

Удаляется таблица «Учет за неделю» (рисунок 33).

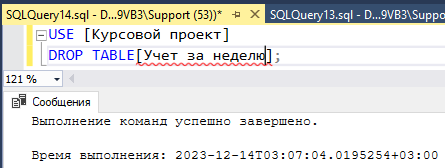


Рисунок 33 – Фрагмент экрана, содержащий листинг оператора Drop Table

Тексты запросов на SQL приводятся в Приложении Б.

Выводы по разделу

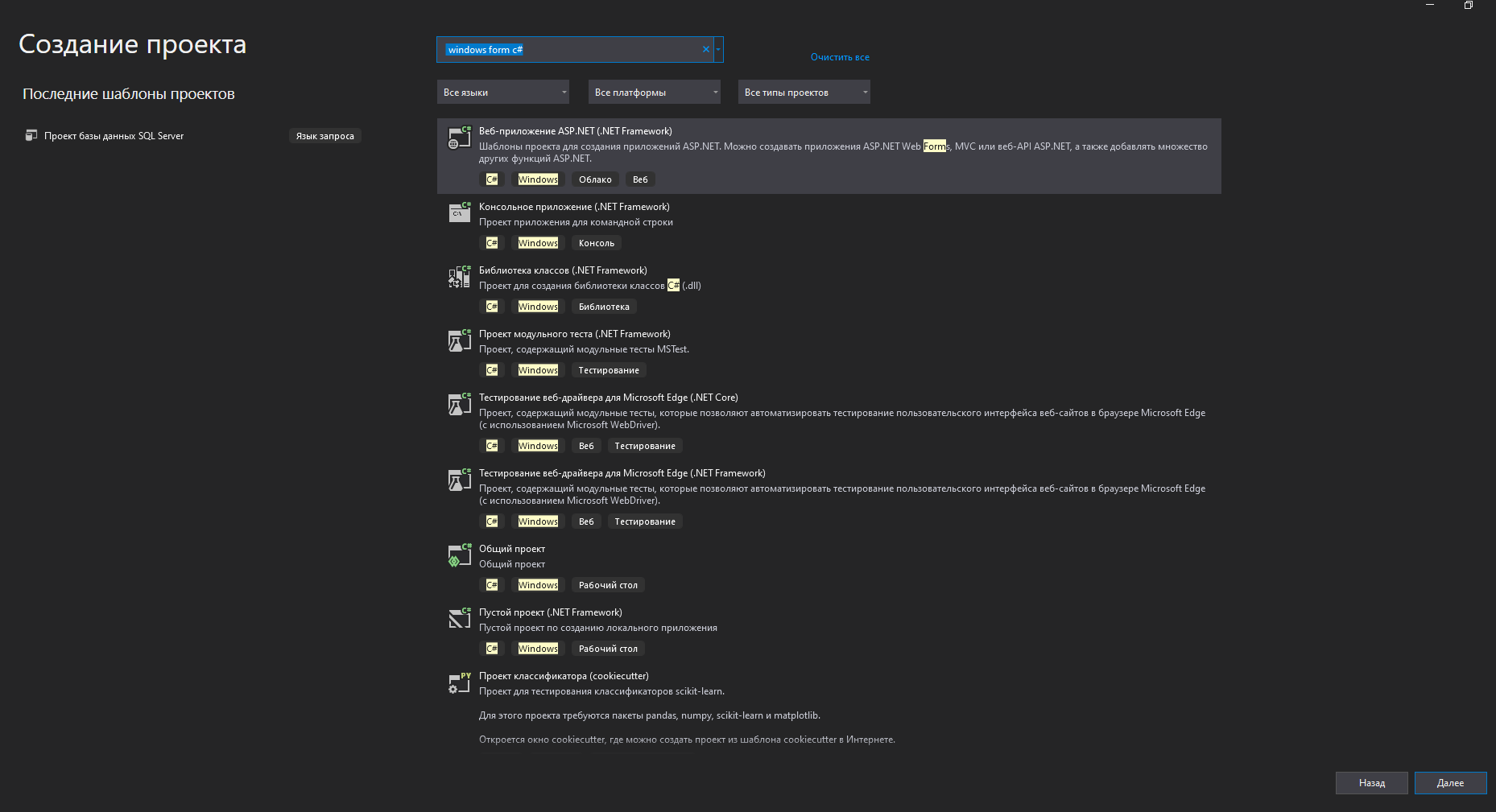
В данном разделе созданы усложненные запросы с операторами CREATE TABLE, DROP TABLE, ALTER TABLE. Созданные запросы выполнены, результаты запросов представлены выше.

# 5 Создание форм для заполнения базы данных

Форма – объект, с помощью которого можно добавлять, редактировать или отображать данные, которые хранятся в БД.

### 5.1 Форма для изменения данных в таблицах

В программе Visual Studio 2019 создается и настраивается приложение Windows Forms (рисунок 34) и (рисунок 35) соответственно.

Рисунок 34 – Окно создания проекта в Visual Studio 2019 (часть 1)

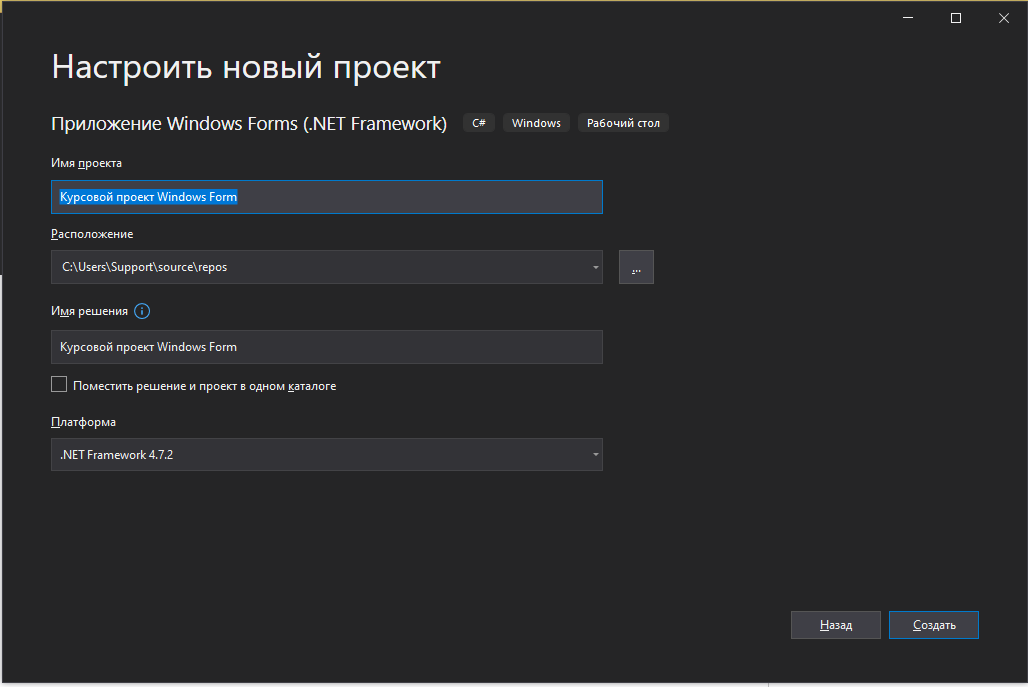
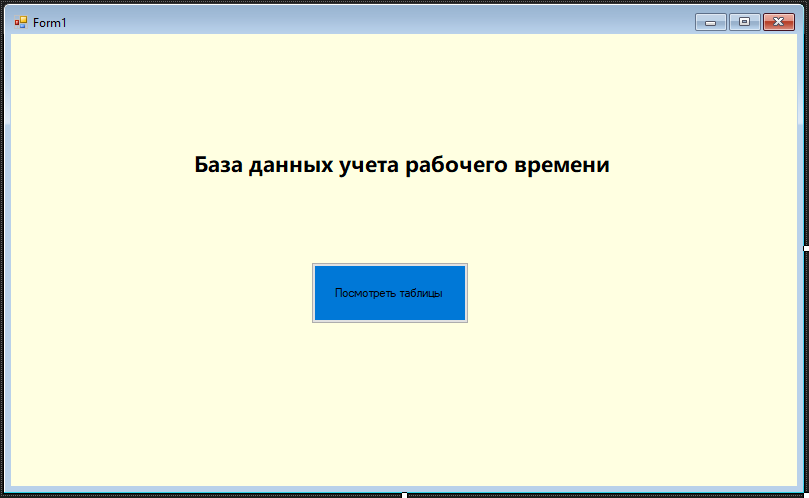


Рисунок 35 – Окно настройки проекта в Visual Studio 2019 (часть 2)

Пользовательский интерфейс проекта включает в себя основную страницу – Form1 ,а также Form2 для просмотра содержимого базы данных.

В Form1 для наглядного просмотра таблиц добавляется надпись Button1: Просмотреть таблицы (рисунок 36).

  
Рисунок 36 – Окно Form1

Добавляется новый источник данных в программе Visual Studio. Подключается созданная база данных из программы Microsoft SQL Server (SqlClient) (рисунок 37).

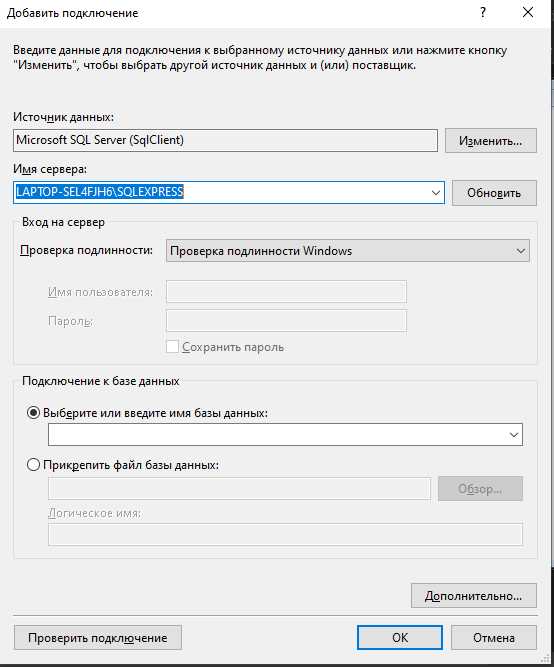


Рисунок 37 – Фрагмент экрана, показывающий подключение базы данных

В Form2 будет возможность просмотреть данные из всех созданных таблиц. Форма состоит из DataGridView.

Добавляется возможность перехода на Form2. Для этого в Form1 вставляется код для Button1 (рисунок 38).

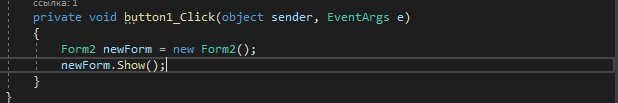


Рисунок 38 – Фрагмент экрана, содержащий текст программы обработчика события «Нажатие кнопки»

После добавления кода просматривается результат работы программы. В Form2 отображаются пять заполненных таблиц (рисунок 39). Теперь с помощью созданной программы можно посмотреть таблицы, без написания кода в Microsoft SQL.

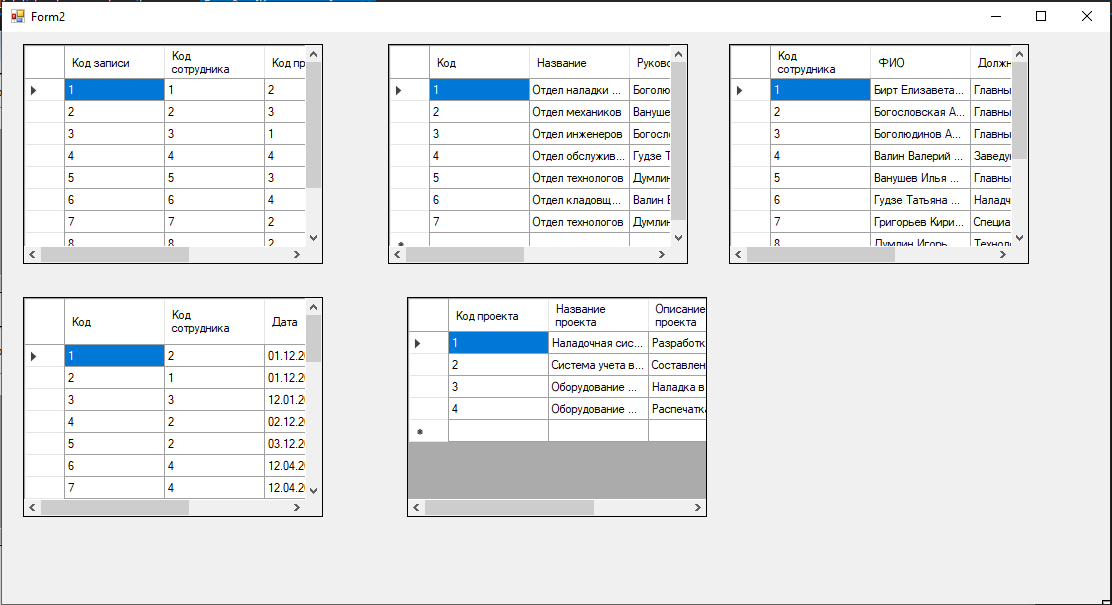


Рисунок 39 – Окно Form2

Выводы по разделу

В данном разделе созданы формы для просмотра данных таблиц. Созданные формы были выполнены, результаты выполнения форм представлены выше.

# Заключение

В рамках выполнения курсового проекта, в соответствии с полученным заданием разработан проект базы данных для ведения табеля учета рабочего времени офисного персонала.

При выполнении курсового проекта проанализировано задание на разработку базы данных, рассмотрена предметная область, в которой предстояло вести разработку, сформированы требования к информационной части системы, к организации базы данных, выполнено функциональное проектирование системы.

Также получены навыки создания таблиц, их связей, запросов, форм. Расширены теоретические знания в области проектирования баз данных.

На основании полученных результатов спроектирована база данных и интерфейс для работы с базой данных. Таблицы заполнены данными. Создано несколько запросов на выборку и других запросов: на добавление, на создание таблицы, на удаление.

Разработаны формы просмотра таблиц в Windows.

Цель курсового проекта достигнута, все поставленные задачи выполнены.

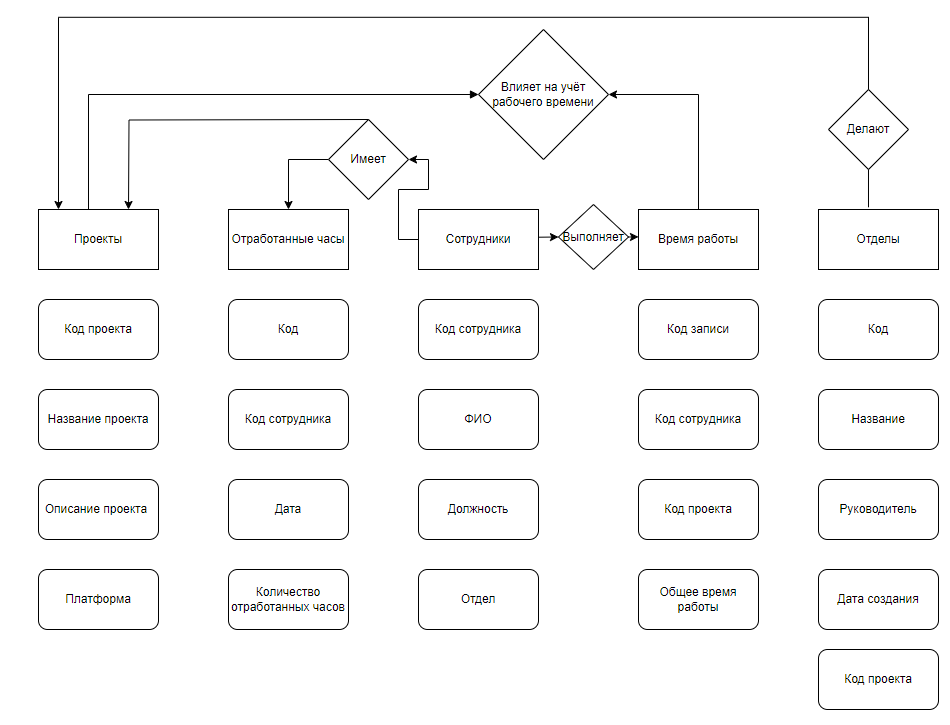
# Список литературы

1. Карпова И.П. Базы данных. Учебное пособие. Москва: Московский гос. ин-т электроники и математики, 2009. – 117 с.
2. Молинаро Э. SQL. Сборник рецептов / Э. Молинаро. – М.: Символ-плюс, **2022.**– 592 c.
3. Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К.Дж. Дейт. - М.: Символ-плюс, **2016**. – 472 c.
4. Тейлор А. SQL для чайников / А. Тейлор. - М.: Вильямс, 2014. – 416 c.
5. Бьюли, А. Изучаем SQL / А. Бьюли. - М.: Символ-плюс**, 2014**. – 309 c.
6. Эйри .Д Функции SQL. Справочник программиста / Д. Эйри. - М.: Диалектика-Вильямс, **2013**. – 768 c.
7. Кригель, А. SQL. Библия пользователя / А. Кригель. - М.: Диалектика-Вильямс, **2016**. – 752 c.
8. Оппель, Э. Дж. SQL. Полное руководство / Оппель Э. Дж. - М.: Диалектика-Вильямс, 2016. – 960 c.
9. Microsoft Corporation. Проектирование и реализация баз данных Microsoft SQL Server 2000. Учебный курс MCSE. - М.: Русская Редакция, **2017**. – 664 c.
10. Кригель A. и др. SQL. Библия пользователя, 2-е издание. Орегон, США: Диалектика-Вильямс, 2010.­­­–752 с.

# Приложение А

Диаграмма «сущность-связь»

«База данных для ведения табеля учета рабочего времени офисного персонала»



# Приложение Б

Тексты запросов

1. USE [Курсовой проект]

INSERT INTO [Сотрудники] ([Код сотрудника], ФИО, Должность, Отдел)

VALUES (12,'Сайкин Вадим Кужугетович', 'Инженер, '3');

1. USE [Курсовой проект]

SELECT \* FROMОтделы;

1. USE [Курсовой проект]

SELECT [Код],[Название] FROMОтделы;

1. USE [Курсовой проект]

SELECT \* FROM Сотрудники

WHERE ([Отдел])=4or ([Отдел])=1;

1. USE [Курсовой проект]

SELECT \* FROM [Время работы]

WHERE [Общее время работы] LIKE '%0%';

1. USE [Курсовой проект]

SELECT \* FROM [Время работы]

1. USE [Курсовой проект]

UPDATE [Время работы]

SET [Общее время работы] ='22';

1. USE [Курсовой проект]

DELETE FROM Проекты

WHERE [Название проекта] = 'Картотека АТС';

1. USE [Курсовой проект]

CREATETABLE [Учет за неделю]

([Код] intNotnull, Неделяchar(100) Notnull, [Количество часов] char(100));

1. USE [Курсовой проект]

DROP TABLE [Учет за неделю];

1. USE [Курсовой проект]

ALTER TABLE [Учет за неделю]

ADD  Даты date ;

1. USE [Курсовой проект]

ALTER TABLE [Учет за неделю]

ALTER COLUMN ДатыVARCHAR(100)NOT NULL ;

1. USE [Курсовой проект]

DROPTABLE[Учет за неделю];

Ректору ФГБОУ ВО

«Чувашский государственный

университет им. И. Н. Ульянова»

А. Ю. Александрову

студента 3 курса

группы КТ-41-21

Иванова Игоря Сергеевича

ЗАЯВЛЕНИЕ

Прошу утвердить мне тему курсового проекта/курсовой работы по дисциплине «Базы данных»:

«База данных для ведения табеля учета рабочего времени офисного персонала».

Руководитель Димитриев А.П.

Дата: \_\_.\_\_\_\_\_\_\_\_\_.20\_\_г. Подпись: \_\_\_\_\_\_\_\_\_\_