Содержание

[**Введение** 4](#_Toc162661723)

[**1.** **Проектирование базы данных** 5](#_Toc162661724)

[**1.1.** **Анализ и описание предметной области** 5](#_Toc162661725)

[**1.2.** **Построение инфологической (концептуальной модели) предметной области** 6](#_Toc162661726)

[1.3. Построение даталогической модели БД 7](#_Toc162661727)

[**2.** **Создание базы данных в Microsoft SQL Server и управление учетными записями и ролями** 8](#_Toc162661728)

[2.1. Создание базы данных, резервное копирование и восстановление базы данных 8](#_Toc162661729)

[2.2. Создание пользователей и ролей для доступа к серверу через утилиту Microsoft SQL Server Management Studio 9](#_Toc162661730)

[**3.** **Разработка таблиц и ограничений** 11](#_Toc162661731)

[**4** **Создание запросов** 16](#_Toc162661732)

[**4.1 Создание запросов на выборку** 16](#_Toc162661733)

[**4.2**  **Создание многотабличных запросов. Запросы на соединение** 19](#_Toc162661734)

[**4.3** **Создание запросов на группировку и сортировку данных. Запросы на изменение. Использование встроенных функций.** 24](#_Toc162661735)

[**5** **Создание представлений** 28](#_Toc162661736)

[**Заключение** 30](#_Toc162661737)

[**Список источников** 31](#_Toc162661738)

# **Введение**

В современной проектной деятельности базы данных играют важную роль в достижении стратегических целей компаний. Для эффективного управления проектами необходимо иметь доступ к достоверным и актуальным данным. В данной работе будет рассмотрена роль баз данных, их методы использования и инструменты для улучшения управления проектами. Будут рассмотрены основные аспекты баз данных и их значение в контексте проектной деятельности

Цель данной работы заключается в разработке базы данных «Проектная организация».

Для достижения этой цели необходимо решить следующие задачи:

1. Анализ предметной области проектной организации
2. Проектирование инфологической и даталогической модели базы данных.
3. Создание базы данных в среде Microsoft SQL Server.
4. Создание учетных записей, ролей и пользователей.
5. Разработка таблиц, ограничений и представлений.
6. Создание запросов для получения необходимой информации.

Эти задачи позволят покрыть основные аспекты проектирования и разработки базы данных для проектной организации

Объектом исследования является организация деятельности отделов и управление рабочим процессом.

Предметом исследования является управление персоналом и организция проектныйх работ.

1. **Проектирование базы данных**
   1. **Анализ и описание предметной области**

Проектная организация - это организация, специализирующаяся на выполнении проектов в определенной области или отрасли. Она может заниматься разработкой, управлением и выполнением проектов от их начала до завершения.

Предметная область проектной организации включает в себя несколько ключевых аспектов:

* Сотрудники организации: Включают в себя конструкторов, инженеров, техников, лаборантов и прочий обслуживающий персонал. Каждый работник имеет свои персональные данные, включая фамилию, имя, год рождения, контактную информацию и должность в организации.
* Отделы: Каждый отдела имеет свои характеристики, такие как название,этаж, начальник.
* Организации: Включает информацию о названии организации и прочую контатную информацию включая адрес и контактный телефон .
* Договра: Описывает всю необходимую информацию о договоре: его номер, дату заключение,организацию-заказчика и стоимость проектной работы
* Проектные работы: Содержит информацию о дате начала и конца проведения работы, номер договра по которому ведется каждая и отдел отвечающий за ее исполнение.

Основными сущностями данной предметной области являются отделы, сотрудники, организации, договра и проектные работы. Проведенный анализ позволил определить ключевые атрибуты каждой сущности и выявить взаимосвязи между ними, что позволяет нам создать связи и отношения сущностями для дальнейшего эффективного взаимодействия с базой данных и ее данным.

Полученные результаты послужат основой для дальнейшего проектирования базы данных, которая будет эффективно поддерживать управление проектной организацией, что позволит сотрудникам и самой организации в целом функцонировать производительней.

* 1. **Построение инфологической (концептуальной модели) предметной области**

На основе анализа предметной области мы можем построить инфологическую модель, которая представляет собой абстрактное описание сущностей и связей между ними. Концептуальная модель представляет собой высокоуровневое представление о базе данных, независимое от конкретных технических реализаций.

Инфологическая (концептуальная модель) предметной области проектная организация представлена в соответствии с рисунком 1.

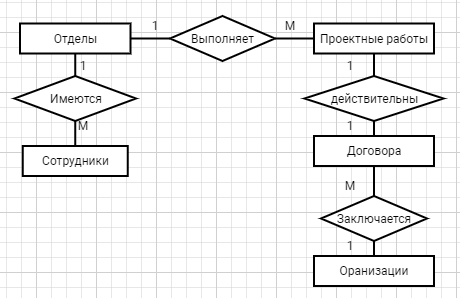


Рисунок 1. Инфологическая (Концептуальная модель) предметной области

Ключевыми сущностями концептуальной модели являются "Сотрудники", "Отделы", "Проектные работы", "Договра" и "Организации».

Между этими сущностями существуют различные отношения например выполнение различных проектных работ отделами или заключение договоров с организациями

Связи этих сущенстей отображают всю логическую модель взаимодейсвтия сущностей.

* 1. Построение даталогической модели БД

После создания инфологической модели предметной области переходим к построению даталогической модели базы данных (БД).

Даталогическая модель БД представлена в соответсвии с рисунком 2

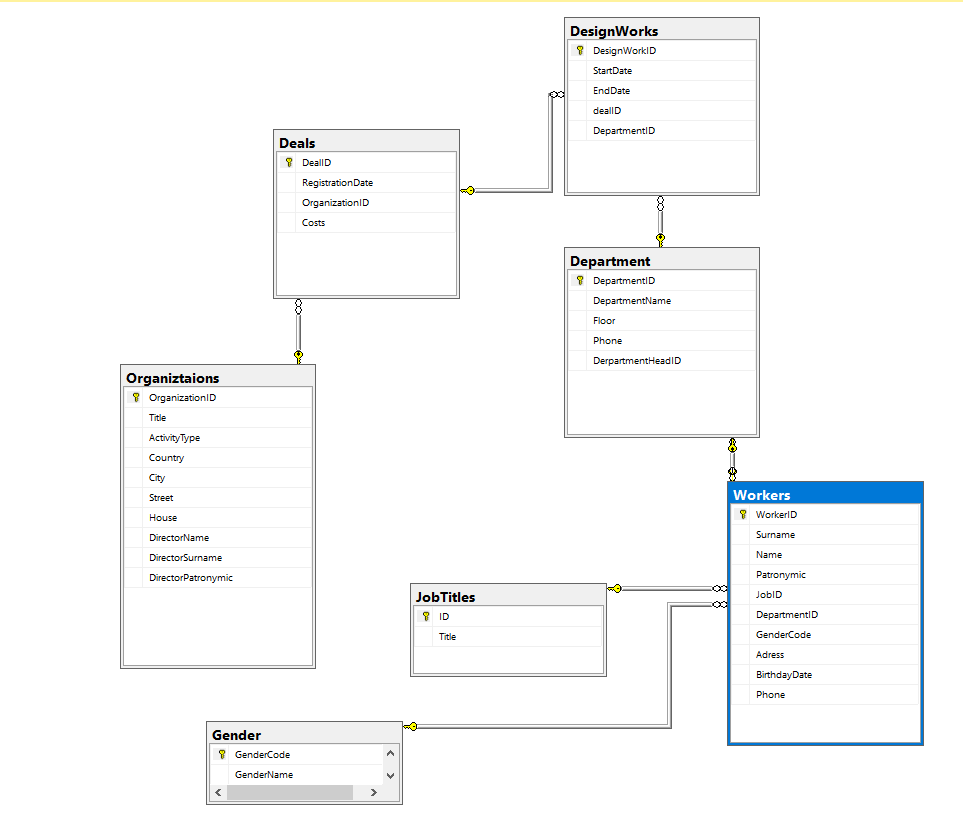


Рисунок 2. Даталогическая модель базы данных "Проектная организация"

Даталогическая модель представляет собой конкретную структуру данных, которая будет реализована в выбранной системе управления базами данных (СУБД), в данном случае - Microsoft SQL Server.

1. **Создание базы данных в Microsoft SQL Server и управление учетными записями и ролями**
   1. Создание базы данных, резервное копирование и восстановление базы данных

Мы создаем базу данных «Проектная органмизация» и резервное копирование с помощью команды TSQL.

Команда на создание базы данных и резервное копирование представлены в соответствии с рисунками 3 - 4.



Рисунок 3. Создание базы данных

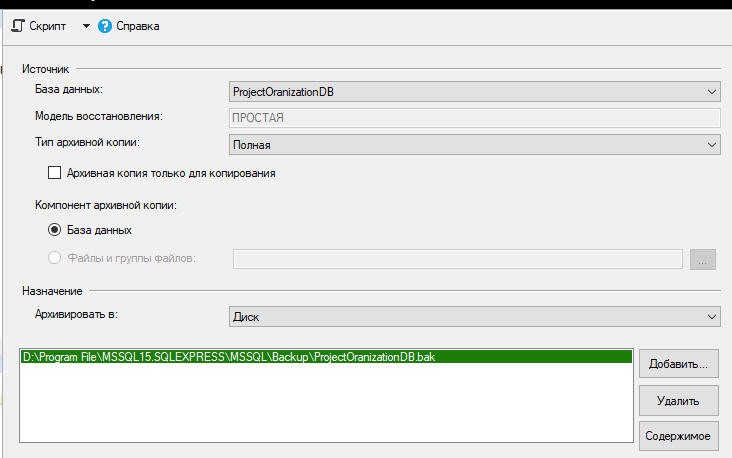


Рисунок 4. Создание резервной копии базы данных

Выполнение данных этапов проекта позволяет осуществить ключевые процессы, необходимые для эффективного управления базой данных. Грамотное выполнение всех шагов по созданию, резервному копированию и восстановлению базы данных обеспечит стабильность работы системы и защиту данных. Важно следовать определенной процедуре и не забывать об актуализации регулярных резервных копий для предотвращения потерь информации.

* 1. Создание пользователей и ролей для доступа к серверу через утилиту Microsoft SQL Server Management Studio

Создадим 2 должностных лица, которые смогут работать с БД.

Создание учетных записей через SQL-запрос составлены в соответствии с рисунками 5-6.



Рисунок 5. Создание учетной записи для секретаря директора



Рисунок 6. Создание учетной записи WorkersManager

Для каждого должностного лица определим набор привилегий, которыми он может пользоваться.

Основные задача секретаря - поддержка рабочей документации: создание и обновление документов, подготовка презентаций, организация документооборота. Поэтому для учетной записи «DirectorAsisstent» создадим пользователя что может просматривать данные из таблиц в базе данных.

Должностное лицо использующее учетную запись «WorkersManager» должно управлять данными о сотрудниках, из-за этого для этого имени для входа создадим пользователя, что может редактировать информацию о работниках организации.

Создание ролей и имен входа для учетных записей «DirectorAsisstent» и «WorkersManager», а также присваивание ролей и полномочий представлено в соответствие с рисунками 7-15



Рисунок 7.Создание имени для входа "DirectorAsisstent" для учетной записи "DirectorAsisstent"



Рисунок 8.Создание имени для входа для WorkersManager



Рисунок 9. Создание роли workersManager



Рисунок 10. Создание роли Asisstent



Рисунок 11. Присваивание полномочий db\_datareader роли Asisstent



Рисунок 12. Присваивание полномочий db\_datareader роли workerManager



Рисунок 13. Присваивание полномочий db\_writer роли workerManager



Рисунок 14. Присваивание полномочий роли Asisstent пользователю DirectorAsisstent



Рисунок 15. Присваивание полномочий workerManager пользователю Manager

Ограничение доступа Пользователю Manager представлено в соответствие с рисунком 16:



Рисунок 16. Ограничение доступа к базе данных и разрешение изменения и прочтения только таблицы "Сотрудники"

Таким образом учетные записи базы данных помогают повысить безопасность, а также способствуют удобству в эксплуатации базы.

**3. Разработка таблиц и ограничений**

Создание таблиц, их ограничений и связи будут создаваться в базе данных «ProjectOranizationDB» через утилиту Microsoft SQL Server Management Studio (далее MS SMS) и transact-sql.

1. Создание таблиц с помощью MS SMS:

Создание таблиц «Отделы», «Сотрудники», «Организации», «Договора», «Проектные работы» представлены в соответствие с рисунками 17-21:

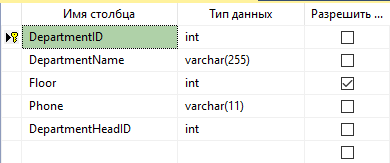


Рисунок 17. Создание таблицы "Отделы"

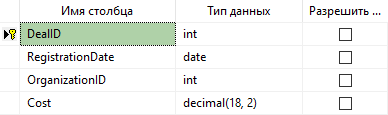


Рисунок 18. создание таблицы "Договора"

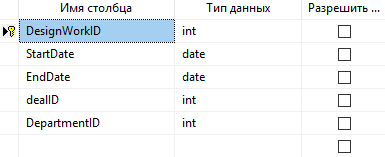


Рисунок 19. Создание таблицы "Проектные работы"

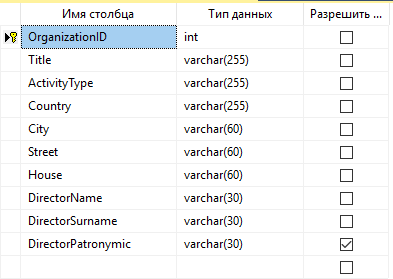


Рисунок 20. Создание таблицы "Организации"

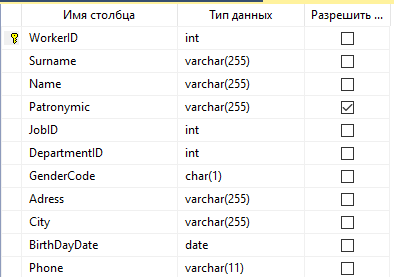


Рисунок 21Создание таблицы "Сотрудники"

1. Создание таблиц с помощью transact-sql:

Создание таблицы «Пол»:

use ProjectOranizationDB

CREATE TABLE [dbo].[Gender] (

GenderCode CHAR(1) PRIMARY KEY

CHECK (GenderCode IN ('м', 'ж')),

GenderName VARCHAR(50) NOT NULL

);

Создание таблицы «Должности»

use ProjectOranizationDB

CREATE TABLE JobTitles (

ID INT IDENTITY(1,1) PRIMARY KEY,

JobTitle VARCHAR(100) NOT NULL

);

Создание диаграммы базы данных, которая отражает структуру таблиц и связей между ними с помощью SQL Server Management Studio представлено в соответствии с рисунком 22.

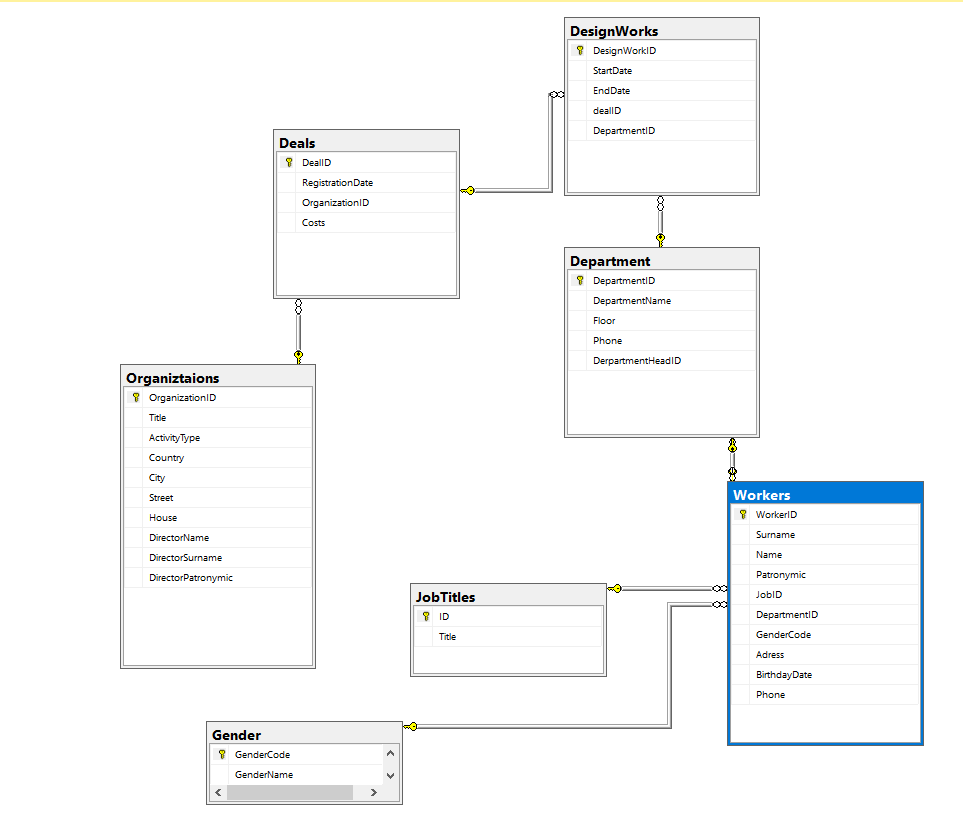


Рисунок 22. Создание диаграммы базы данных

Заполнение таблиц «Пол» и «Должности» будет выполнятся с помощью языка transact-sql, а таблицы «Организации», «Договора», «Отделы», «Работники» и «Проектные работы» будут осуществляться вручную

Заполнение таблиц «Пол»:

INSERT INTO Gender (GenderCode, GenderName)

VALUES

('м', 'Мужской'),

('ж', 'Женский');

Заполнение таблиц «Должности»:

INSERT INTO JobTitles (JobTitle) VALUES ('конструктор');

INSERT INTO JobTitles (JobTitle) VALUES ('инженер');

INSERT INTO JobTitles (JobTitle) VALUES ('техник');

INSERT INTO JobTitles (JobTitle) VALUES ('лаборант'),

INSERT INTO JobTitles (JobTitle) VALUES ('начальник отдела');

Заполнение таблиц «Организации», «Договора», «Отделы», «Работники» и «Проектные работы» представлено в соответствие с рисунками 23- 27:

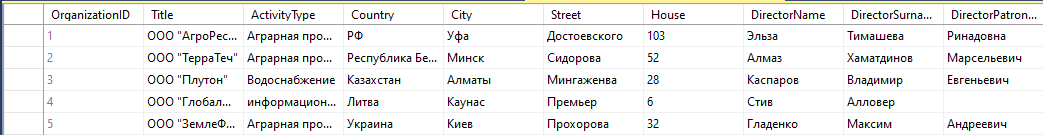


Рисунок 23. Заполнение таблицы "Организации"

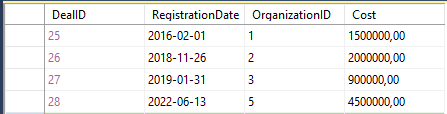


Рисунок 24. Заполнение таблицы "Договора"

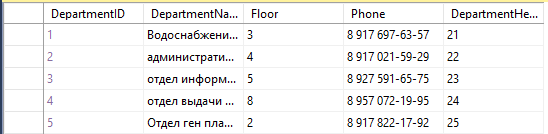


Рисунок 25. Заполнение таблицы "Отделы"

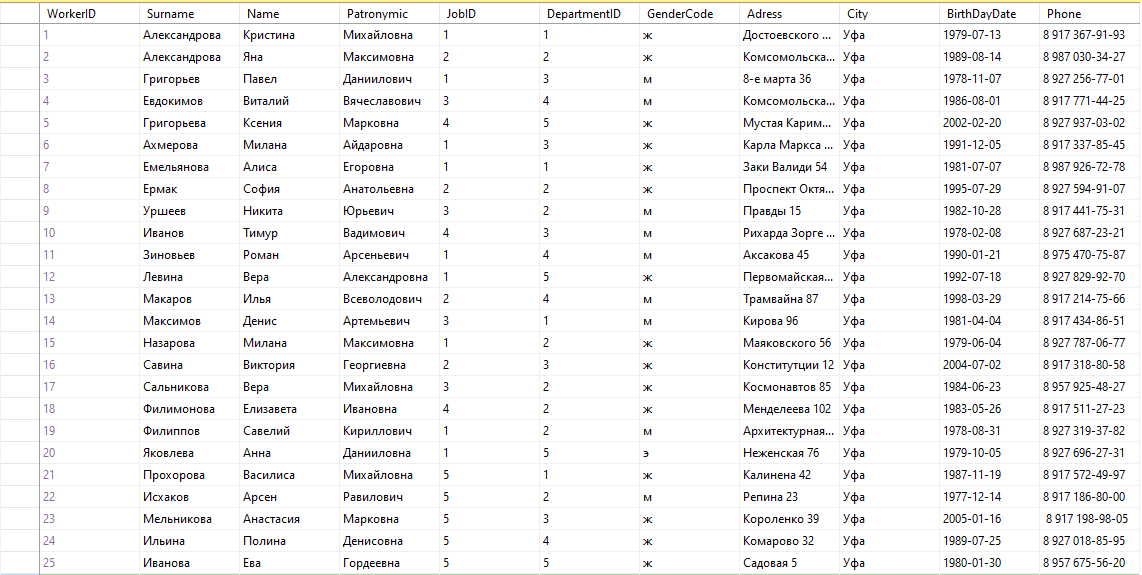


Рисунок 26. Заполнение таблицы "Сотрудники"

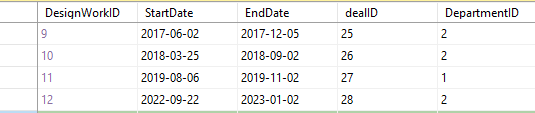


Рисунок 27. Заполнение таблицы "Проектные работы"

Таким образом мы заполнили все таблицы базы данных, создали все связи между таблицами.

**4 Создание запросов**

**4.1 Создание запросов на выборку**

Запрос с использованием операторов сравнения

SELECT Workers.Surname as Фамилия ,Workers.Name as Имя

FROM Workers

WHERE Workers.GenderCode like 'м';

Результат данного запроса представлен в соответствии с рисунком 28:



Рисунок 28. Результат запроса с использованием оператора сравнения.

Запрос с использованием логических операторов AND, OR и NOT:

SELECT Workers.Surname as Фамилия, Workers.Name as Имя

FROM Workers

WHERE Workers.GenderCode like 'м' OR Workers.DepartmentID = 2;

Результат данного запроса представлен в соответствии с рисунком 29:

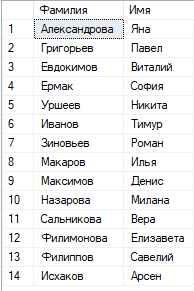


Рисунок 29. Запрос с использованием логических операторов.

Запрос на использование комбинации логических операторов:

SELECT Workers.Surname as Фамилия ,Workers.Name as Имя

FROM Workers

WHERE Workers.GenderCode like 'ж' AND Workers.DepartmentID = 1 OR Workers.DepartmentID = 2;

Результат данного запроса представлен в соответствии с рисунком 30:



Рисунок 30. Результат запроса на использование комбинаций логических операторов

Запрос на использование выражений над столбцами:

SELECT Cost as [Цена проектной работы], (Cost/7.7) as [ Сумма полученная после оплаты налогов]

FROM Deal;

Результат данного запроса представлен в соответствии с рисунком 31:

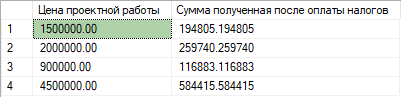


Рисунок 31. Результат запроса на использование выраэений над столбцами.

Запрос с проверкой на принадлежность множеству:

SELECT Surname as [Фамилия], Name as [Имя]

FROM Workers

WHERE Surname IN ('Александрова')

Результат данного запроса представлен в соответствии с рисунком 32:



Рисунок 32. Результат запроса на проверку принадлежности множеству

Запрос с проверкой на принадлежность диапазону значений:

SELECT Surname as [Фамилия], Name as [Имя]

FROM Workers

WHERE DepartmentID BETWEEN 1 AND 3;

Результат данного запроса представлен в соответствии с рисунком 33:



Рисунок 33. Результат запроса с проверкой на принадлежность диапазону значений

Запрос с проверкой на соответствие шаблону:

SELECT Surname as [Фамилия], Name as [Имя]

FROM Workers

WHERE Surname like 'Е%';

Результат данного запроса представлен в соответствии с рисунком 34:

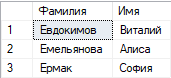


Рисунок 34. Результат запроса с проверкой на соответствие шаблону

Запрос с проверкой на неопределенное значение:

SELECT Title as [Название организации ], Directorsurname as [Фамилия директора], DirectorName as [Имя директора]

FROM Organization

WHERE DirectorPatronymic is NULL

Результат данного запроса представлен в соответствии с рисунком 35:



Рисунок 35. Результат запроса на неопределенное значение

Таким образом мы разработали простейшие запросы, которые помогут в эксплуатации нашей базы данных.

**4.2 Создание многотабличных запросов. Запросы на соединение**

Запрос с использованием соединения двух таблиц по равенству:

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

INNER JOIN JobTitles ON Workers.JobID = JobTitles.ID;

Результат данного запроса представлен в соответствии с рисунком 36:

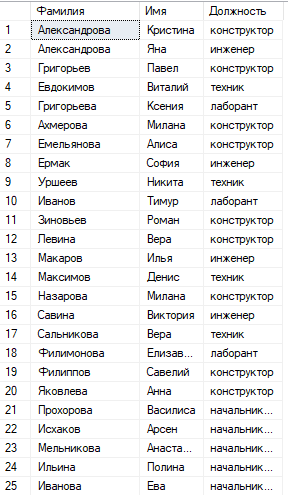


Рисунок 36.Результат запроса с использованием соединения двух таблиц по равенству

Запрос с использованием соединения двух таблиц по равенству и условием отбора:

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

INNER JOIN JobTitles ON Workers.JobID = JobTitles.ID

WHERE JobTitle = 'техник';

Результат данного запроса представлен в соответствии с рисунком 37:

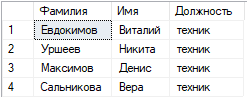


Рисунок 37. Результат запроса с ипспользованием соединения двух таблиц по равенству и условием отбора

Запрос с использованием соединения по трем таблицам:

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

INNER JOIN JobTitles ON Workers.JobID = JobTitles.ID

INNER JOIN Department ON Workers.WorkerID = Department.DepartmentHeadID

WHERE JobTitle like 'начальник отдела'

Результат данного запроса представлен в соответствии с рисунком 38:

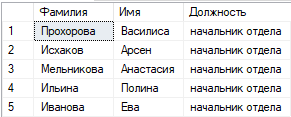


Рисунок 38. Результат запроса с использованием соединения по трем таблицам

Запрос на основе копий ранее созданных запросов на соединение по равенству на запросы с использованием внешнего полного соединения таблиц (JOIN):

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

FULL JOIN JobTitles ON Workers.JobID = JobTitles.ID;

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

FULL JOIN JobTitles ON Workers.JobID = JobTitles.ID

WHERE JobTitle = 'техник';

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

FULL JOIN JobTitles ON Workers.JobID = JobTitles.ID

FULL JOIN Department ON Workers.WorkerID = Department.DepartmentHeadID

WHERE JobTitle like 'начальник отдела'

Их результат представлен в соответствии с рисунками 39- 41:

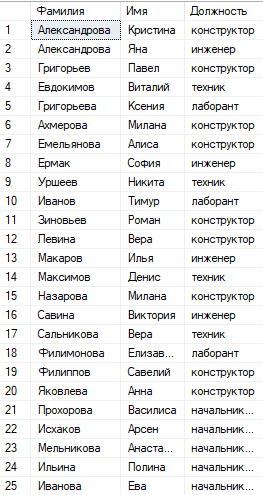


Рисунок 39. Результат запроса с использованием внешнего полного соединения таблиц (JOIN).

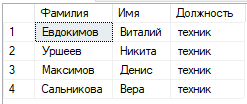


Рисунок 40. Результат запроса с использованием внешнего полного соединения таблиц (JOIN).

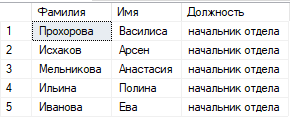


Рисунок 41. Результат запроса с использованием внешнего полного соединения таблиц (JOIN).

Запрос с использованием левого внешнего соединения:

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

LEFT JOIN JobTitles ON Workers.JobID = JobTitles.ID;

Результат данного запроса представлен в соответствии с рисунком 42:



Рисунок 42. Результат запроса с использованием левого внешнего соединения.

Запрос на использование правого внешнего соединения:

SELECT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

RIGHT JOIN JobTitles ON Workers.JobID = JobTitles.ID;

Результат данного запроса представлен в соответствии с рисунком 43:

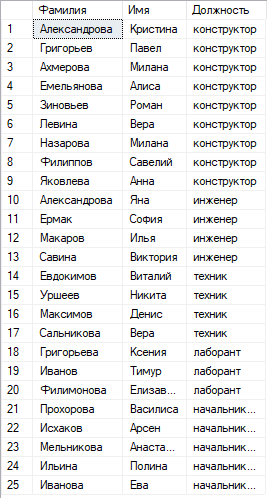


Рисунок 43. Результат запроса с использованием правого внешнего соединения.

Запрос с использованием симметричного соединения и удаление избыточности:

SELECT DISTINCT Surname as [Фамилия], Name as [Имя], JobTitle[Должность]

FROM Workers

FULL JOIN JobTitles ON Workers.JobID = JobTitles.ID;

Результат данного запроса представлен в соответствии с рисунком 44:



Рисунок 44. Результат запроса с использованием симметричного соединения и удаления избыточности.

**4.3 Создание запросов на группировку и сортировку данных. Запросы на изменение. Использование встроенных функций.**

Запрос с использованием функции COUNT:

SELECT JobTitle as [Должность], COUNT(JobID) as [Количество сотрудников этой должности]

FROM WOrkers

JOIN JobTitles ON Workers.JobID = JobTitles.ID

GROUP BY JobTitle

Результат данного запроса представлен в соответствии с рисунком 45:

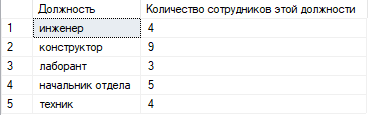


Рисунок 45. Результат запроса с использованием функции COUNT.

Запрос с использованием функции SUM:

SELECT SUM (DEAL.Cost) as [Суммарная стоимость всех сделок]

FROM Deal;

Результат данного запроса представлен в соответствии с рисунком 46:



Рисунок 46. Результат запроса с использованием функции SUM.

Запрос с использованием временных функций:

SELECT RegistrationDate as [Дата заключения сделаки], CONVERT (DATE, CURRENT\_TIMESTAMP) as [Текущая дата]

FROM Deal;

Результат данного запроса представлен в соответствии с рисунком 47:

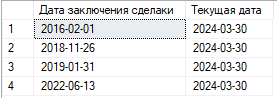


Рисунок 47. Результат запроса с использованием временных функций.

Запрос с использованием группировки по одному столбцу:

SELECT JobTitle as [Должность], COUNT(JobID) as [Количество сотрудников этой должности]

FROM WOrkers

JOIN JobTitles ON Workers.JobID = JobTitles.ID

GROUP BY JobTitle

Результат данного запроса представлен в соответствии с рисунком 48:

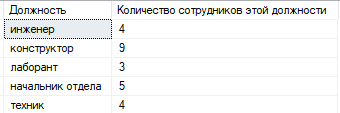


Рисунок 48.Результат запроса с использованием группировки по одному столбцу.

Запрос с использованием условия отбора групп HAVING:

SELECT JobTitle as [Должность], COUNT(JobID) as [Количество сотрудников этой должности]

FROM WOrkers

JOIN JobTitles ON Workers.JobID = JobTitles.ID

GROUP BY JobTitle

HAVING COUNT(JobID)> 4;

Результат данного запроса представлен в соответствии с рисунком 49:



Рисунок 49. Результат запроса с использованием условия отбора групп HAVING.

Запрос с использованием сортировки по столбцу:

SELECT JobTitle as [Должность], COUNT(JobID) as [Количество сотрудников этой должности]

FROM WOrkers

JOIN JobTitles ON Workers.JobID = JobTitles.ID

GROUP BY JobTitle

ORDER BY COUNT (JobID);

Результат данного запроса представлен в соответствии с рисунком 50:

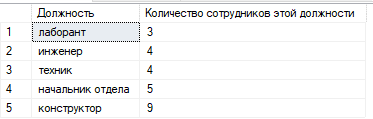


Рисунок 50. Результат с использованием сортировки по столбцу.

Запрос на добавление новых данных в таблицу:

INSERT INTO JobTitles (JobTitle)

VALUES ('Уборщик');

Результат данного запроса представлен в соответствии с рисунком 51:

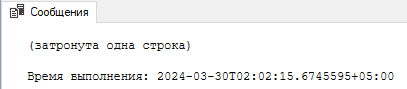


Рисунок 51. Результат запроса на добавление новых данных в таблицу.

Запрос на обновление существующих данных в таблице:

UPDATE Organization

SET DirectorName = 'Джон'

WHERE DirectorPatronymic is NULL

Результат данного запроса представлен в соответствии с рисунком 52:

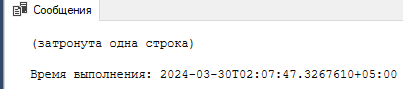


Рисунок 52. Результат запроса на обновление существующих данных в таблице.

Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE:

UPDATE Workers

SET Workers.Name ='Даша'

WHERE Workers.Surname in (SELECT Workers.Surname FROM Workers WHERE Workers.Surname = 'Иванова')

Результат данного запроса представлен в соответствии с рисунком 53:

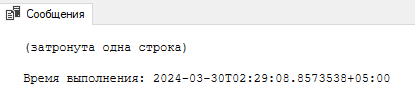


Рисунок 53. Результат запроса на обновление существующих данных по результатам подзапроса во фразе WHERE.

Запрос на удаление существующих данных:

DELETE FROM JobTitles

WHERE JobTitle like 'Уборщик'

Результат данного запроса представлен в соответствии с рисунком 54:

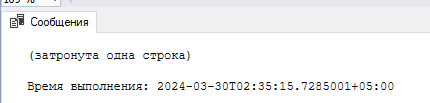


Рисунок 54. Разработка запроса на удаление существующих данных

Таким образом запросы на группировку и сортировку данных, многотабличные запросы, запросы на выборку помогают вывести нужные данные менеджеру базы данных, а также повысить эффективность и ускорить его работу.

**5 Создание представлений**

Создадим 2 представления по ранее созданным запросам (на выбор) с помощью графических средств утилиты Enterprise Manager.

Создание первого представления «FirstView» и результат его работы представлен в соответствии с рисунком 55:

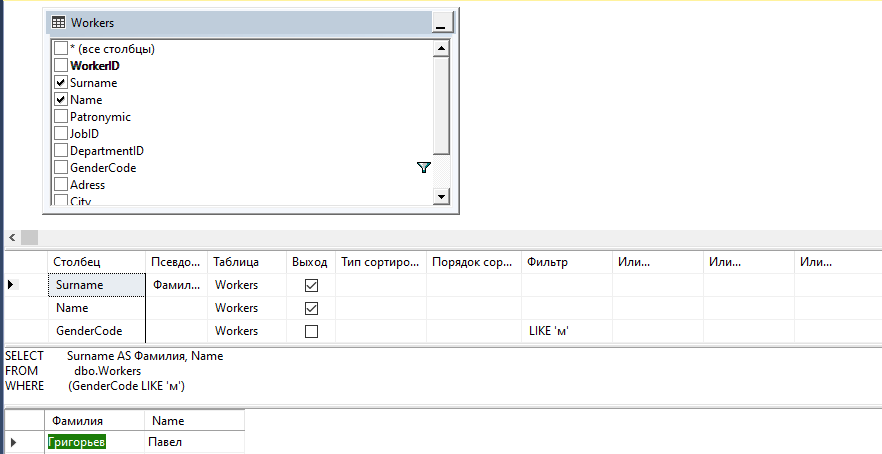


Рисунок 55. Создание первого представления "FirstView".

Создание второго представления «SecondView» и результат его работы представлен в соответствии с рисунком 56:

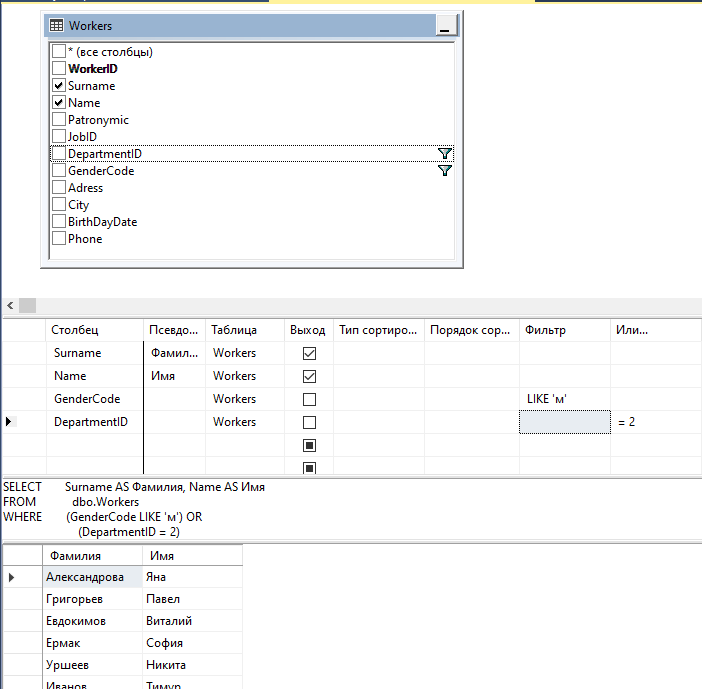


Рисунок 56. Создание второго представления "SecondView".

Создадим 2 представления по ранее по ранее созданным запросам (на выбор) с помощью мастера Create View Wizard.

Создание третьего представления «ThirdView»:

CREATE VIEW ThirdView

as

SELECT Surname as [Фамилия], Name as [Имя]

FROM Workers

WHERE Surname like 'Е%';

Результат его работы представлен в соответствие с рисунком 57:

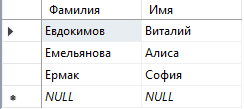


Рисунок 57. Результат работы представления "ThirdView"

Создание четвертого представления «FourthView»:

CREATE VIEW FourthView

as

SELECT Cost as [Цена проектной работы], (Cost/7.7) as [Сумма полученная после оплаты налогов]

FROM Deal;

Результат его работы представлен в соответствие с рисунком 58:

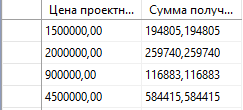


Рисунок 58. Результат работы представления "FourthView"

Таким образом представления в базе данных нужны менеджеру для того чтобы удобным ему образом вывести данные, а также сохранить структуру именованного запроса.

# **Заключение**

В рамках данной работы мы провели исследование и анализ существующих систем управления для проектной организации. Определены требования к базе данных, которые позволят эффективно управлять проектной организации

В результате проведенного анализа мы спроектировали инфологическую и даталогическую модели базы данных, которые будут использоваться для организации и хранения данных.

На основе разработанных моделей мы создали саму базу данных, которая будет служить основой для работы с информацией о сотрудниках и проектных работах. Создали учетные записи сотрудников, назначив им соответствующие роли и права доступа. Таким образом, мы обеспечили безопасность и контроль доступа к данным.

Для того чтобы база данных была полезной, мы разработали и заполнили таблицы актуальными данными. Это включает информацию об организациях, договорах и других сущностях, которые относятся к работе проектной организации. Таким образом, база данных содержит полную и актуальную информацию, которую можно использовать для различных целей.

Для получения необходимой информации из базы данных мы разработали различные запросы и представления. Это позволяет нам эффективно извлекать данные и получать нужные отчеты, статистику и аналитику. Запросы и представления помогают нам решать разнообразные задачи и удовлетворять потребности пользователей.

Таким образом поставленная цель достигнута, задачи выполнены.

# **Список источников**

1. Дейт К. Дж. Введение в системы баз данных. - М.: Вильямс, 2005. - 1024 с.
2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. - М.: Вильямс, 2007. - 1408 с.
3. Херсонский В. И. Базы данных: учебник для вузов. - М.: Форум: Инфра-М, 2006. - 576 с.
4. Рамзаева Л. Г. Базы данных: учебник для вузов. - М.: Юрайт, 2015. - 365 с.
5. Силбершатц А., Корт Х., Садри С. Системы баз данных. - М.: Вильямс, 2006. - 1136 с.
6. Мак-Коннелл Дж. С. Анализ требований и создание спецификаций. - СПб.: Питер, 2002. - 608 с.
7. Бейза Й., Дейт К. Дж. Введение в системы баз данных. - М.: Вильямс, 2005. - 1024 с.
8. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. - М.: Вильямс, 2002. - 720 с.
9. Microsoft SQL Server 2012. Руководство для начинающих. - М.: ДМК Пресс, 2013. - 864 с.
10. Бен-Ган И., Мачкасова Д., Саркаа П. Microsoft SQL Server 2012. Руководство для начинающих. - М.: ДМК Пресс, 2013. - 864 с.
11. Бейза Й., Дейт К. Дж. Введение в системы баз данных. - М.: Вильямс, 2005. - 1024 с.
12. SQL Server Central. [Электронный ресурс]. URL: <http://www.sqlservercentral.com/> (дата обращения: 01.05.2022).
13. Microsoft SQL Server Documentation. [Электронный ресурс]. URL: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15> (дата обращения: 01.05.2022).
14. Stack Overflow. [Электронный ресурс]. URL: <https://stackoverflow.com/> (дата обращения: 01.05.2022).
15. W3Schools. [Электронный ресурс]. URL: <https://www.w3schools.com/sql/> (дата обращения: 01.05.2022).