**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

**КУРСОВОЙ ПРОЕКТ**

По дисциплине МДК.07.01 Управление и автоматизация баз данных

**Разработка базы данных «****Строительная компания»**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_\_\_\_/Поспелов Дмитрий Сергеевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА: \_\_\_\_\_\_\_\_\_\_\_\_\_**

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ**

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_/ Егоров Данил Павлович/

(подпись) (Ф.И.О.)

Воркута

2024

Оглавление

ВВЕДЕНИЕ

Курсовая работа на тему «Строительная компания». Основными видами деятельности строительной организации являются: выполнение строительное монтажных и ремонтно-строительных работ подрядчиками и субподрядчиками.

Строительная организация осуществляет следующие виды деятельности:

* Выполнение строительно-монтажных работ;
* Производство строительных материалов и изделий;
* Выполнение работ по благоустройству територий.

Сотрудники. Реализовано управление персоналом;

Бригады. Нужна для формирования, расформирования и изменения бригад, назначения бригадиров;

Проекты. Отображение всех проектов, добавления нового проекта. За проектом закреплен прораб и бригада.

В данном курсовом проекте, мы рассмотрим процесс разработки базы данных строительная компания, а так-же посмотрим модель, отвечающую всем требованиям строительного процесса.

**Объект:** база данных «Строительная компания»

**Предмет:** автоматизация бизнес-процесса работы строительной компании

**Цель работы –** разработать базу данных для информационной системы «Строительная компания».

**Задачи:**

* Выбор инструментария;
* Проектирование базы данных;
* Разработка базы данных;

ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ

1.1 СУБД SQL Server

Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANЅІЛЅО структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Особенности функционирования SQL SERVER:

СУБД SQL server используются для создания, размещения, хранения и управления реляционными (табличными) базами данных на специальных серверах или в облаке. Они работают через настольные приложения и web- сайты. К основным преимуществам их функционирования относятся:

* высокоскоростной доступ к данным, обеспечиваемый надежной клиент-серверной архитектурой СУБД;
* простота работы и администрирования, обусловленные понятной структурой языка программирования SQL;
* безопасность хранения информации в БД – благодаря возможности шифрования данных и резервного копирования.
* Специфика работы сервера базы данных SQL server заключается в транзакционной обработке данных. Это означает, что по каждому запросу от СУБД обрабатывается и сохраняется небольшое количество информации.

Применение SQL server позволяет автоматизировать решение различных бизнес-задач, поддерживать проведение аналитики данных в режиме онлайн, отслеживать направление ресурсов СУБД, управлять транзакциями (операциями по обработке данных).

Виды SQL-server:

Для реализации функций СУБД на сегодняшний день чаще всего используются следующие SQL-серверы:

MS SQL server - многопользовательский программный продукт, разработанный компаний Microsoft, обладающий высокой производительностью и отказоустойчивостью, тесно интегрированный с ОС Windows. Этот сервер поддерживает удаленные подключения, работает с многими популярными типами данных, дает возможность создавать триггеры и хранимые данные, имеет практичные и удобные утилиты для настройки;

Oracle Database server СУБД, предназначенная для создания, консолидации и управления базами данных в облачной среде. Используя этот сервер, можно как автоматизировать обычные бизнес-операции, так и выполнять динамический многомерный анализ данных (OLAP), проводить операции с документами xml-формата и управлять разделенной и локальной информацией;

ІВМ DB2 - семейство СУБД для работы с реляционными базами данных, признанное самым производительным, имеющим высокие технические показатели и возможности масштабирования. SQL-серверы этой группы характеризуются мультиплатформенностью, способностью к мгновенному созданию резервных копий и восстановлению БД, реорганизации таблиц в онлайн-режиме, разбиению баз данных, определению пользователями новых типов данных; +

MySQL - СУБД, разработанная и поддерживаемая компанией Oracle. В основном она используется локальными или удаленными клиентами, позволяя им работать с таблицами разных типов, поддерживающих полнотекстовый поиск или выполняющих транзакции на уровне отдельных записей;

PostgreSQL – СУБД с открытым исходным кодом, работающая с объектно-реляционными базами данных. Также PostgreSQL предназначена для создания, хранения и извлечения сложных структур данных. Она поддерживает самые различные типы данных.

1.2 Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных SQL Azure. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных. Используйте SSMS для развертывания, мониторинга и обновления компонентов уровня данных, используемых приложениями, и создания запросов и скриптов.

Используйте SSMS для запроса, проектирования и управления базами данных и хранилищами данных, где бы они ни находились на локальном компьютере или в облаке.

SQL Server Management Studio (SSMS) утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

Также есть SQL Server Management Studio Express для Express версии сервера, которая является бесплатной. Однако в ней нет поддержки ряда компонентов (Analysis Services, Integration Services, Notification Services, Reporting Services) и SQL Server 2005 Mobile Edition.

Среда SQL Server Management Studio предназначена для доступа к службам Службы Analysis Services, Integration Services и Reporting Services, a также для их настройки, администрирования и управления ими. Хотя все три технологии бизнес-аналитики полагаются на среду SQL Server Management Studio, административные задачи, связанные с каждой из этих технологий, несколько отличаются.

При использовании последнего пакета обновления текущий выпуск SSMS поддерживает следующие 64-разрядные платформы:

Поддерживаемые операционные системы:

* Windows 11 (x64)
* Windows 10 (x64) версии 1607 (10.0.14393) и более поздних версий
* Windows Server 2022 (x64)
* Windows Server Core 2022 (x64)
* Windows Server 2019 (x64)
* Windows Server Core 2019 (x64)
* Windows Server 2016 (x64) 1
* Для 1 SSMS требуется платформа .NET Framework 4.7.2.

Дополнительные требования:

Права администратора необходимы для установки или обновления SSMS.

Контейнеры Windows не поддерживаются.

SSMS не поддерживается в решениях виртуализации приложений, таких как Microsoft App-V или MSIX для Windows, или сторонние технологии виртуализации приложений.

ЅЅМЅ работает только в Windows (AMD или Intel). Если вам требуется от Windows, средство, которое работает на платформах, отличных — это кроссплатформенное рассмотрите Azure Data Studio. Azure Data Studio средство для macOS, Linux и Windows.

Среда SQL Server Management Studio позволяет включать компоненты служб Reporting Services, администрировать серверы и базы данных, управлять ролями и заданиями.

В целом, SQL Server Management Studio (SSMS) — это мощный инструмент, предоставляющий полную среду для управления базами данных SQL Server. Благодаря набору инструментов и функций SSMS упрощает разработку, управление и поддержку баз данных, позволяя пользователям более эффективно работать с экземплярами SQL Server.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

2.1 Разработка диаграммы ERD

**Описание:**

ER-диаграмма графически представляет сущности (entities) предметной области, свойства (attributes) сущностей и связи (relationship) между ними

ER-диаграммы делятся на концептуальные и физические. В отличие от физических, в концептуальных ER-диаграммах не учитываются особенности конкретной базы данных. Впоследствии сущности концептуальных ER-диаграмм становятся таблицами, атрибуты — колонками, а связи реализуются путем миграции ключевых атрибутов родительских сущностей и создания внешних ключей

**Концептуальная ERD** — высокоуровневая диаграмма, которая описывает общую структуру базы данных и отношения между сущностями, не включая подробных сведений об атрибутах.

**Логическая модель** – содержит более подробную информацию, в отличии от концептуальной модели.

**Физическая ERD** - диаграмма, которая включает в себя детали реализации базы данных, такие как имена таблиц, первичные ключи, внешние ключи и типы данных.

В модели есть **три типа связей**:

● «Один-к-одному» — один экземпляр сущности связан только с одним экземпляром другой сущности. Например, пассажир рейса и его место в самолете.

● «Один-ко-многим» — один экземпляр сущности связан со множеством экземпляров другой сущности. Например, у одного пассажира может быть несколько единиц багажа, при этом каждая единица багажа может быть связана только с одним пассажиром.

● «Многие-ко-многим» — множество экземпляров одной сущности связаны со множеством экземпляров другой сущности. Например, аэропорт обслуживает несколько авиакомпаний. При этом каждая авиакомпания может обслуживаться в нескольких аэропортах.

**Основные объекты:**

**Сущность** (entity) — На ER-диаграмме сущность изображается в виде прямоугольника, внутри которого содержится имя сущности в форме существительного в единственном числе

**Связь** (relationship) — ассоциация между сущностями. Для облегчения понимания диаграммы следует добавлять названия связей.

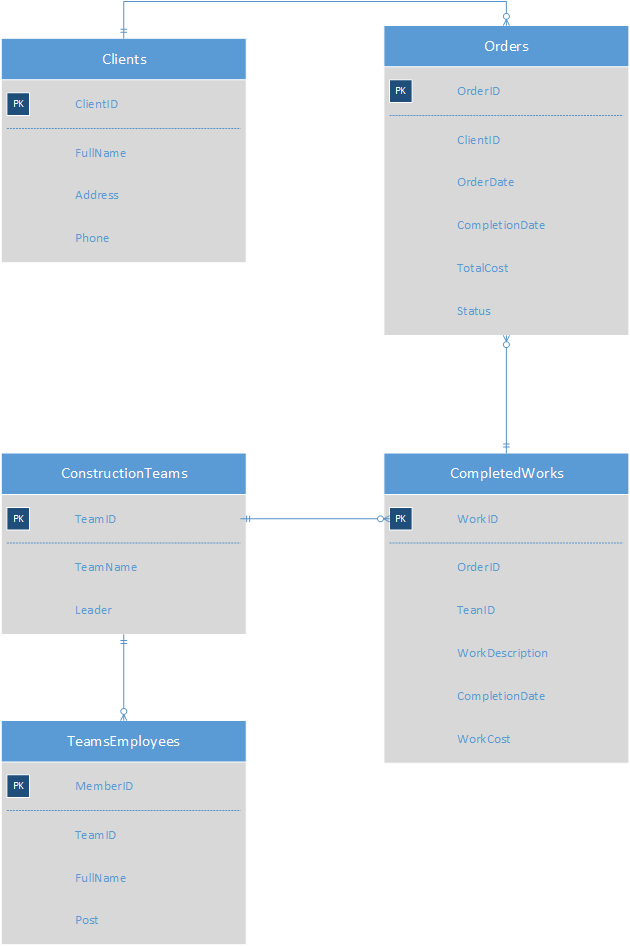
**Атрибуты** предназначены для описания сущности.

Рис 2.1 Диаграмма ERD (Entity Relationship Diagram)

2.2 Словарь данных

Словарь данных – это структура данных, которая представляет собой набор пар ключ-значение. Каждый ключ уникален в словаре данных, и ему соответствует определенное значение. Словари данных используются для хранения и организации информации, а также для быстрого доступа к данным по ключу. Они широко применяются в программировании для решения различных задач, таких как хранение конфигурационных данных, управление ресурсами и т.д. Словари данных также удобны для хранения и передачи структурированных данных между различными частями программы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clients | | | | |
| **Country** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ClientID** | INT | Y | Уникальный идентификатор клиента (PK) |
|  | **FullName** | VARCHAR (255) | N | Фамилия, имя, отчество клиента |
|  | **Address** | VARCHAR (255) | N | Место проживания клиента |
|  | **Phone** | VARCHAR (50) | N | Телефонный номер клиента |

Таблица 1. Clients

Таблица 2. Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orders | | | | |
| **Country** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **OrderID** | INT | Y | Уникальный идентификатор заказа (PK) |
| FK | **ClientID** | VARCHAR (255) | Y | Вторичный Идентификатор клиента (FK) |
|  | **OrderDate** | DATE | N | Дата создания заказа |
|  | **CompletionDate** | DATE | N | Дата завершения заказа |
|  | **TotalCost** | DECIMAL (10, 2) | N | Общая стоимость заказа |
|  | **Status** | VARCHAR (255) | N | Статус заказа клиента |

Таблица 3. ConstructionTeams

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ConstructionTeams | | | | |
| **Country** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **TeamID** | INT | Y | Уникальный идентификатор клиента (PK) |
|  | **TeamName** | VARCHAR (255) | N | Наименования строительной бригады |
|  | **Leader** | VARCHAR (255) | N | Лидер строительной бригады |

Таблица 4. Teams Employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| teams employees | | | | |
| **Country** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **MemberID** | INT | Y | Уникальный идентификатор клиента (PK) |
| FK | **TeamID** | INT | Y | Вторичный идентификатор (FK) |
|  | **FullName** | VARCHAR (255) | N | Фамилия, имя, отчество |
|  | **Post** | VARCHAR (255) | N | Почта |

Таблица 5. Completed Works

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CompletedWorks | | | | |
| **Country** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **WorkID** | INT | Y | Уникальный идентификатор клиента (PK) |
| FK | **OrderID** | INT | Y | Вторичный идентификатор |
| FK | **TeamID** | VARCHAR (255) | Y | Вторичный идентификатор |
|  | **WorkDescription** | TEXT | N | Описание выполненной работы |
|  | **CompletionDate** | DATE | N | Дата завершения |
|  | **WorkCost** | DECIMAL(10, 2) | N | Стоимость выполненной работы |

ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

3.1 Разработка базы данных

Согласно техническому заданию, была создана база данных строительной компании, с запросами

CREATE DATABASE ConstructionCompany;

USE ConstructionCompany;

-- Таблица Clients

CREATE TABLE Clients (

ClientID INT AUTO\_INCREMENT PRIMARY KEY,

FullName VARCHAR(255) NOT NULL,

Address VARCHAR(255),

Phone VARCHAR(255)

);

-- Таблица Orders

CREATE TABLE Orders (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

ClientID INT,

OrderDate DATE,

CompletionDate DATE,

TotalCost DECIMAL(10, 2),

Status VARCHAR(255),

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

-- Таблица ConstructionTeams

CREATE TABLE ConstructionTeams (

TeamID INT AUTO\_INCREMENT PRIMARY KEY,

TeamName VARCHAR(255),

Leader VARCHAR(255)

);

-- Таблица TeamEmployees

CREATE TABLE TeamEmployees (

MemberID INT AUTO\_INCREMENT PRIMARY KEY,

TeamID INT,

FullName VARCHAR(255),

Post VARCHAR(255),

FOREIGN KEY (TeamID) REFERENCES ConstructionTeams(TeamID)

);

-- Таблица CompletedWorks

CREATE TABLE CompletedWorks (

WorkID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

TeamID INT,

WorkDescription TEXT,

CompletionDate DATE,

WorkCost DECIMAL(10, 2),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (TeamID) REFERENCES ConstructionTeams(TeamID)

);

-- Заполнение таблицы Clients

INSERT INTO Clients (FullName, Address, Phone) VALUES

('Иванов Игорь', 'Москва, Ленина, 1', '+7 (999) 123-45-67'),

('Петрова Анна', 'Спб, Невский, 2', '+7 (999) 234-56-78'),

('Сидоров Алексей', 'Казань, Баумана, 3', '+7 (999) 345-67-89');

-- Заполнение таблицы Orders

INSERT INTO Orders (ClientID, OrderDate, CompletionDate, TotalCost, Status) VALUES

(1, '2023-01-10', '2023-01-20', 50000.00, 'Завершен'),

(2, '2023-02-15', '2023-02-25', 75000.00, 'В процессе'),

(1, '2023-03-01', NULL, 30000.00, 'В ожидании');

-- Заполнение таблицы ConstructionTeams

INSERT INTO ConstructionTeams (TeamName, Leader) VALUES

('Команда 1', 'Федоров Сергей'),

('Команда 2', 'Морозов Лев');

-- Заполнение таблицы TeamEmployees

INSERT INTO TeamEmployees (TeamID, FullName, Post) VALUES

(1, 'Кузнецов Андрей', 'Мастер'),

(1, 'Смирнова Ольга', 'Рабочий'),

(2, 'Григорьев Павел', 'Мастер'),

(2, 'Николаев Виктор', 'Рабочий');

-- Заполнение таблицы CompletedWorks

INSERT INTO CompletedWorks (OrderID, TeamID, WorkDescription, CompletionDate, WorkCost) VALUES

(1, 1, 'Постройка фундамента', '2023-01-18', 20000.00),

(1, 1, 'Возведение стен', '2023-01-19', 30000.00),

(2, 2, 'Строительство кровли', '2023-02-24', 40000.00);

-- Вывод всех клиентов

SELECT \* FROM Clients;

-- Вывод всех заказов с информацией о клиентах

SELECT Orders.\*, Clients.FullName, Clients.Address

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID;

-- Вывод завершенных работ с информацией о заказах и командах

SELECT CompletedWorks.\*, Orders.OrderDate, ConstructionTeams.TeamName

FROM CompletedWorks

JOIN Orders ON CompletedWorks.OrderID = Orders.OrderID

JOIN ConstructionTeams ON CompletedWorks.TeamID = ConstructionTeams.TeamID;

-- Вывод всех сотрудников конкретной команды

SELECT \* FROM TeamEmployees WHERE TeamID = 1;

-- Справка по заказам заданного клиента

SELECT Orders.\*, Clients.FullName

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID

WHERE Clients.FullName = 'Иванов Игорь';

-- Выполненные работы каждой бригады за заданный интервал времени

SELECT ConstructionTeams.TeamName, CompletedWorks.WorkDescription, CompletedWorks.CompletionDate

FROM CompletedWorks

JOIN ConstructionTeams ON CompletedWorks.TeamID = ConstructionTeams.TeamID

WHERE CompletedWorks.CompletionDate BETWEEN '2023-01-18' AND '2023-02-24';

-- стоимость всех заказов

SELECT SUM(TotalCost) AS TotalOrdersCost

FROM Orders;

-- стоимость работ, выполненных каждой бригадой

SELECT ConstructionTeams.TeamName, SUM(CompletedWorks.WorkCost) AS TotalWorkCost

FROM CompletedWorks

JOIN ConstructionTeams ON CompletedWorks.TeamID = ConstructionTeams.TeamID

GROUP BY ConstructionTeams.TeamID;

-- Список просроченных заказов

SELECT Orders.\*, Clients.FullName

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID

WHERE Orders.CompletionDate < CURDATE() AND Orders.Status != 'Завершен';

-- Список заказов, которые должны завершиться в ближайший месяц

SELECT Orders.\*, Clients.FullName

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID

WHERE Orders.CompletionDate BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 1 MONTH);

-- Отчет о платежах клиентов

SELECT Clients.FullName, SUM(Orders.TotalCost) AS TotalPayments

FROM Clients

JOIN Orders ON Clients.ClientID = Orders.ClientID

GROUP BY Clients.ClientID;

-- Отчет о задолженности клиентов

SELECT Clients.FullName, SUM(Orders.TotalCost) AS TotalDebts

FROM Clients

JOIN Orders ON Clients.ClientID = Orders.ClientID

WHERE Orders.Status != 'Завершен'

GROUP BY Clients.ClientID;

Согласно техническому заданию, были выполнены запросы.

1. Выдать список всех клиентов с персональными данными;

-- Вывод всех клиентов

SELECT \* FROM Clients;

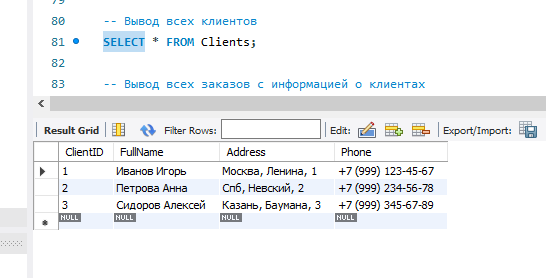


Рис 3.1 Список клиентов

1. Выдать справку по заказам заданного клиента;

-- Справка по заказам заданного клиента

SELECT Orders.\*, Clients.FullName

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID

WHERE Clients.FullName = 'Иванов Игорь';

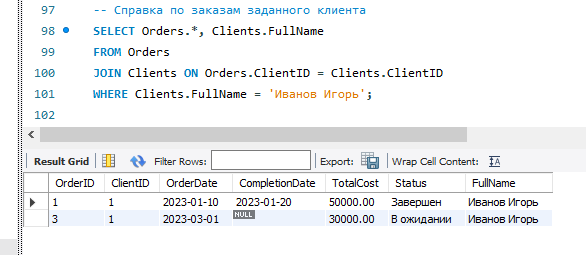


Рис 3.2 Справка по заказам клиентов

1. Сформировать список бригад и их членов, для которых указать должность;

-- Вывод всех сотрудников конкретной команды

SELECT \* FROM TeamEmployees WHERE TeamID = 1;

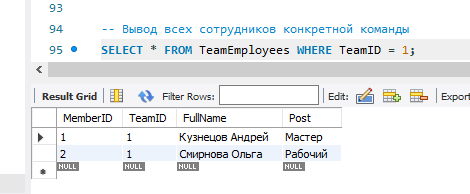


Рис 3.3 Список бригад

1. По каждой бригаде вывести выполненные работы за заданный интервал времени;

-- Выполненные работы каждой бригады за заданный интервал времени

SELECT ConstructionTeams.TeamName, CompletedWorks.WorkDescription, CompletedWorks.CompletionDate

FROM CompletedWorks

JOIN ConstructionTeams ON CompletedWorks.TeamID = ConstructionTeams.TeamID

WHERE CompletedWorks.CompletionDate BETWEEN '2023-01-18' AND '2023-02-24';

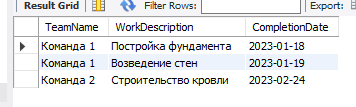


Рис 3.4 Выполнение работы

1. Рассчитать стоимость всех заказов;

-- стоимость всех заказов

SELECT SUM(TotalCost) AS TotalOrdersCost

FROM Orders;

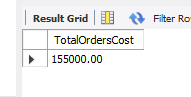


Рис 3.5 стоимость всех заказов

1. Рассчитать стоимость работ, выполненных каждой бригадой;

-- стоимость работ, выполненных каждой бригадой

SELECT ConstructionTeams.TeamName, SUM(CompletedWorks.WorkCost) AS TotalWorkCost

FROM CompletedWorks

JOIN ConstructionTeams ON CompletedWorks.TeamID = ConstructionTeams.TeamID

GROUP BY ConstructionTeams.TeamID;

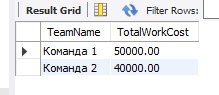


Рис 3.6 стоимость работ выполненной бригады

1. Выдать список просроченных заказов;

-- Список просроченных заказов

SELECT Orders.\*, Clients.FuиllName

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID

WHERE Orders.CompletionDate < CURDATE() AND Orders.Status != 'Завершен';

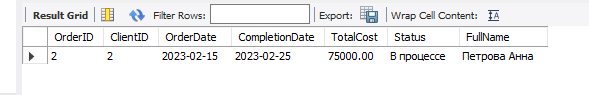


Рис 3.7 просроченные заказы

1. Выдать список заказов, которые должны завершиться в ближайший месяц;

-- Список заказов, которые должны завершиться в ближайший месяц

SELECT Orders.\*, Clients.FullName

FROM Orders

JOIN Clients ON Orders.ClientID = Clients.ClientID

WHERE Orders.CompletionDate BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 1 MONTH);

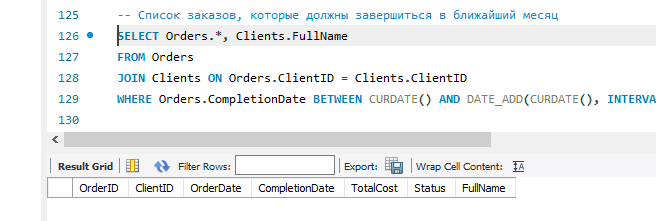


Рис 3.8 заказы в ближайший месяц

1. Формировать отчет о платежах клиентов;

-- Отчет о платежах клиентов

SELECT Clients.FullName, SUM(Orders.TotalCost) AS TotalPayments

FROM Clients

JOIN Orders ON Clients.ClientID = Orders.ClientID

GROUP BY Clients.ClientID;

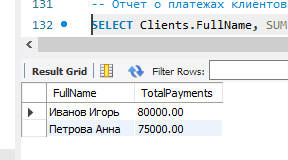


Рис 3.9 платежи клиентов

1. Сформировать отчет о задолженности клиентов.

-- Отчет о задолженности клиентов

SELECT Clients.FullName, SUM(Orders.TotalCost) AS TotalDebts

FROM Clients

JOIN Orders ON Clients.ClientID = Orders.ClientID

WHERE Orders.Status != 'Завершен'

GROUP BY Clients.ClientID;

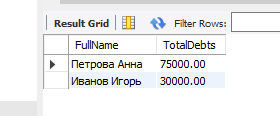


Рис 3.10 за должности клиентов

3.2 Введение в безопасность SQL Server

Защита SQL Server может рассматриваться как ряд шагов, включающих четыре области: платформу, проверку подлинности, объекты (включая данные) и приложения, которые обращаются к системе. В этой статье описывается создание и реализация эффективного плана безопасности.

Платформа SQL Server включает физические аппаратные и сетевые системы, подключающие клиентов к серверам базы данных, а также двоичные файлы, используемые для обработки запросов к базе данных.

**Безопасность платформы и сети**

Платформа SQL Server включает физические аппаратные и сетевые системы, подключающие клиентов к серверам базы данных, а также двоичные файлы, используемые для обработки запросов к базе данных. SQL Server использует файлы операционной системы для операций и хранилища данных. Рекомендации по обеспечению безопасности файлов требуют ограничения доступа к этим файлам.

3.3 Управление безопасностью уровня сервера

Завести пользователя для базы данных и дать ему какую-то роль.

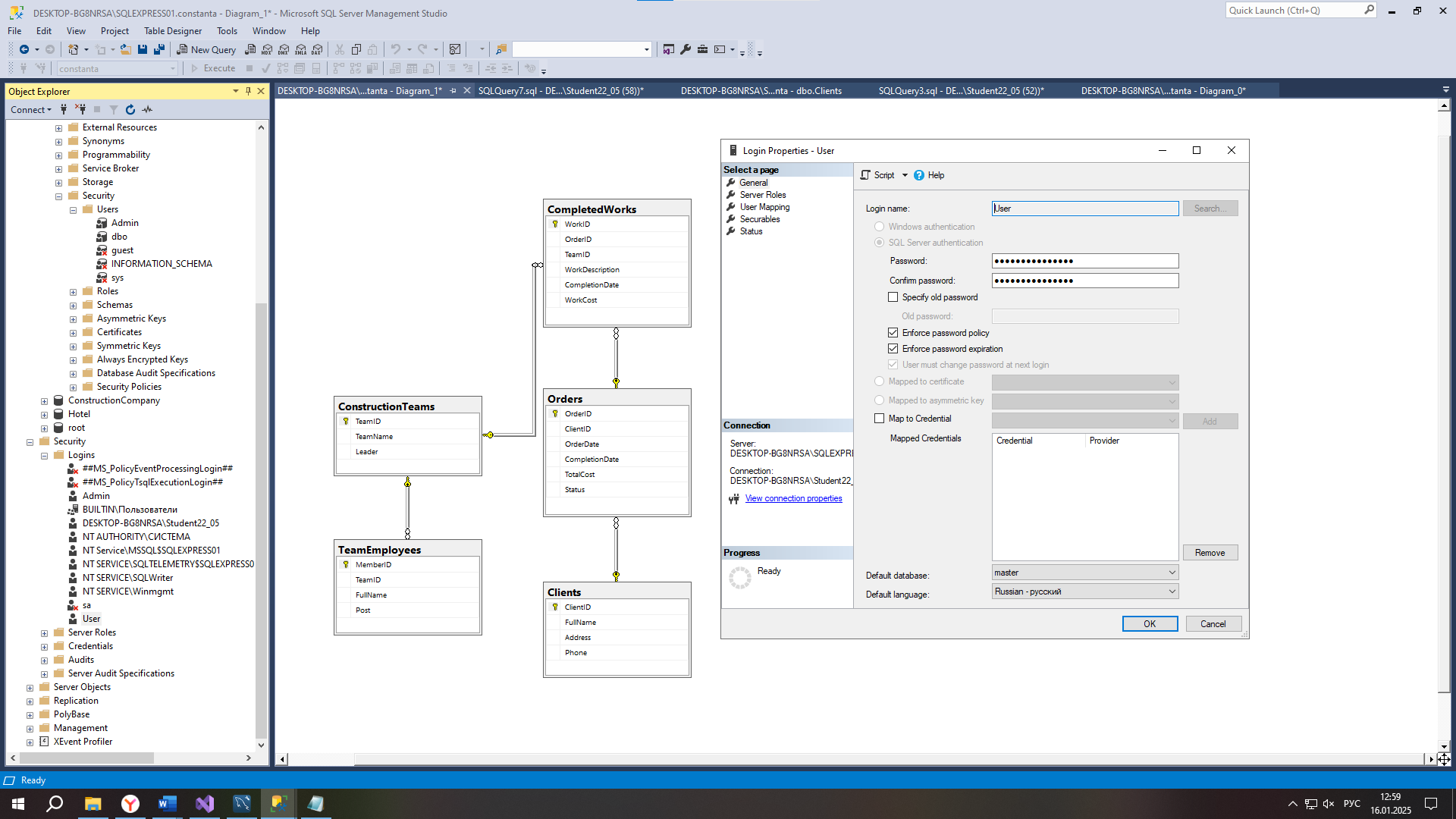


Рисунок 3.11 Добавление пользователя

Подключение пользователей к базе данных constanta

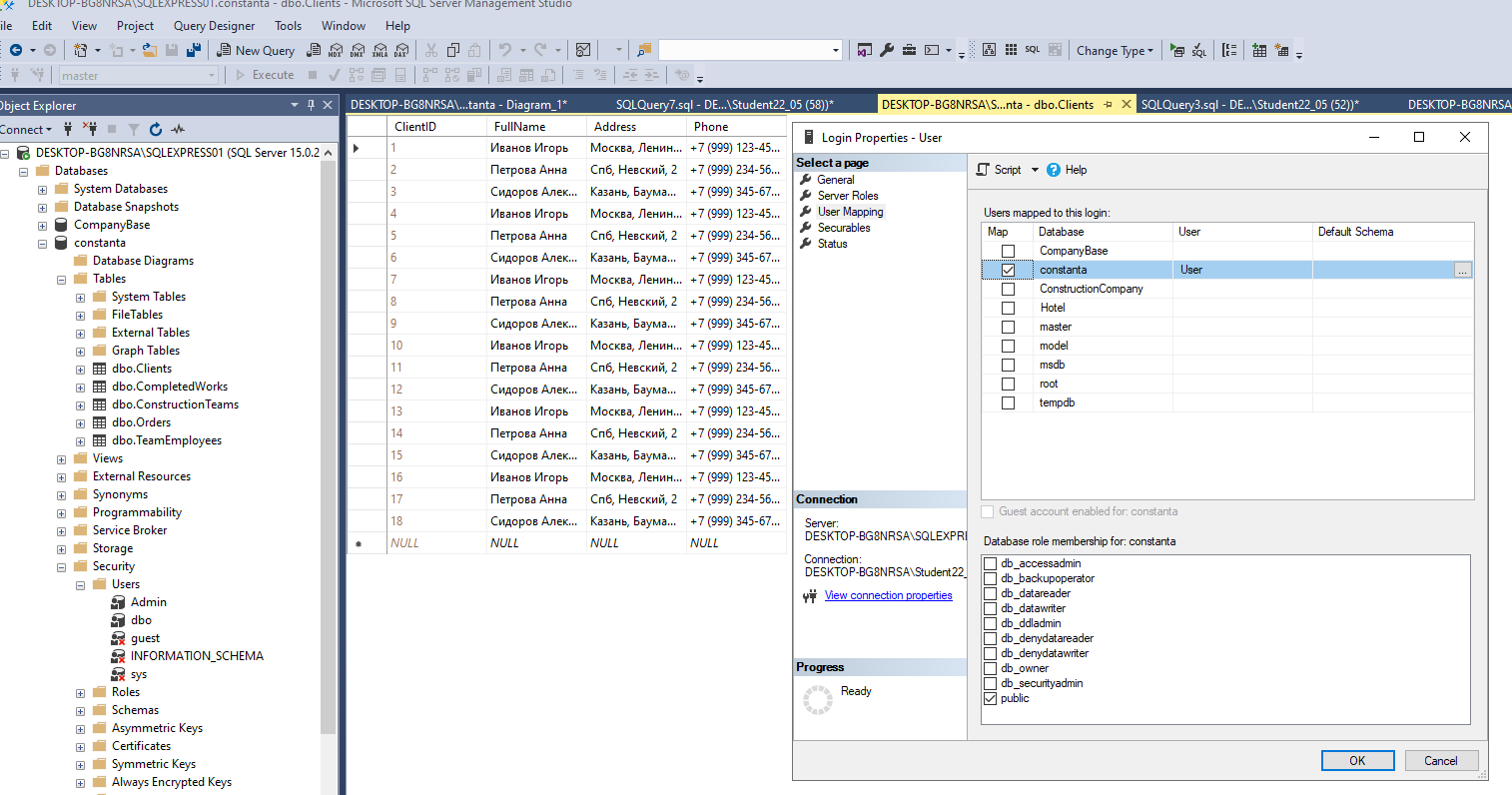


Рисунок 3.12 Разграничение прав пользователя

3.4 Управление участниками уровня базы данных

Создаю и подключаю двух пользователей к базе данных: User и Admin. На уровне сервера, чтобы потом их можно было использовать для подключения к базе данных

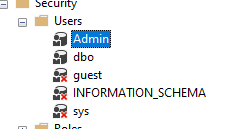


Рисунок 3.13 Созданные пользователи в БД

3.5 Управление разрешениями уровня базы данных

SQL Server применяет системные базы данных для хранения внутренних метаданных. Администратору базы данных необходимо уметь управлять этими данными, а пользователям — правильно их использовать. В базе данных назначены права доступа для ролей User и Admin к таблицам, а также права на использование функций, связанных с этими таблицами.

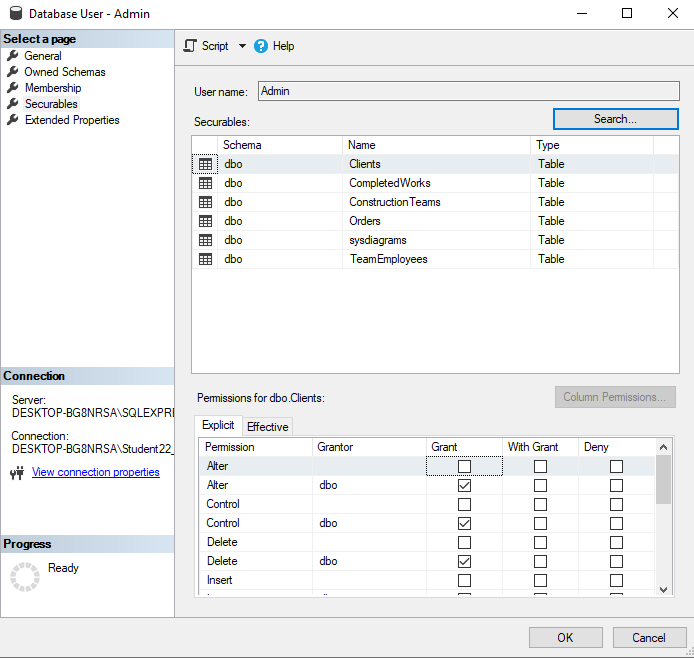
****

Рисунок 3.14 Выдача прав пользователя

3.6 Резервное копирование баз данных

Резервное копирование (BackUp) — это процедура создания копии данных с целью их защиты и восстановления в случае утраты, повреждения или удаления. Резервные копии могут храниться как на физических носителях, таких как внешние жесткие диски и USB-накопители, так и в облачных сервисах. Этот процесс имеет ключевое значение для обеспечения безопасности данных и защиты от их потери.

Рекомендуется создать резервную копию базы данных компании по строительству в файл под названием Constanta.bak.

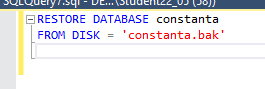


Рисунок 3.15 Код восстановления файла

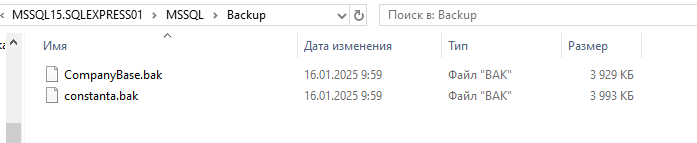


Рисунок 3.16 Файл восстановления

3.7 Шифрование данных баз данных

Шифрование базы данных – использование технологии шифрования для преобразования информации, хранящейся в базе данных (БД), в шифр текст, что делает её прочтение невозможным для лиц, не обладающих ключами шифрования.

Шифрование базы данных представляет собой метод преобразования информации, хранящейся в базе данных, в зашифрованный формат, что затрудняет доступ к ней для тех, кто не обладает соответствующими ключами шифрования. Процесс шифрования происходит как перед записью данных на диск, так и в процессе взаимодействия с программным обеспечением — например, во время ввода пароля и логина, где данные сверяются на соответствие. Это помогает защитить "неактивные" данные, но не обеспечивает должного уровня безопасности информации при ее передаче по сетям или во время использования.

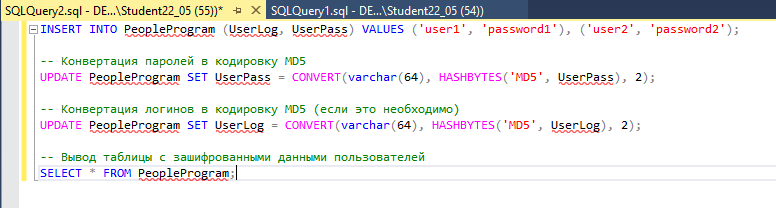


Рисунок 3.17 Запрос на шифрование данных

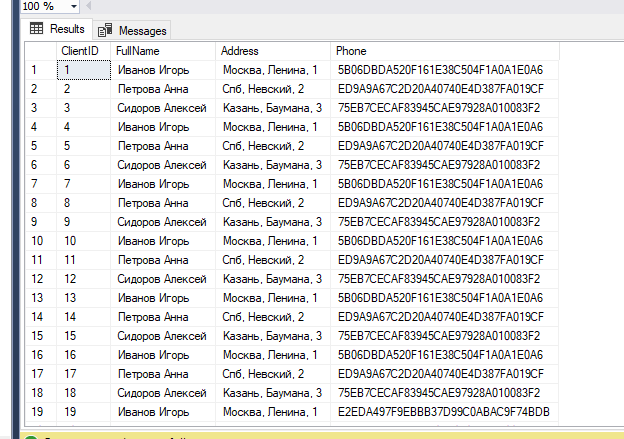


Рисунок 3.18 Зашифрованные данные

ЗАКЛЮЧЕНИЕ

Разработка базы данных «Строительная компания» имеет ключевое значение в сфере управления проектами и клиентами в современном строительном бизнесе. В условиях растущей конкуренции и высокой скорости выполнения заказов, эффективное управление информацией о клиентах, заказах, строительных бригадах и выполненных работах становится жизненно необходимым. Автоматизированная база данных позволяет оптимизировать процессы учета заказов, формирования отчетов, анализа платежей, а также улучшает взаимодействие с клиентами.

В процессе работы над проектом была разработана диаграмма Entity Relationship Diagram, а также спроектирована база данных «Строительная компания» с использованием Microsoft SQL Server Management Studio. Эта база данных ориентирована на сотрудников строительной компании и обеспечивает быстрый доступ к важной информации. С её помощью можно легко составлять отчеты, редактировать, добавлять и удалять данные, что значительно повышает эффективность работы.

База данных «Строительная компания» включает в себя информацию о клиентах и их персональных данных, сведения о заказах и строительных бригадах, а также данные о выполненных работах. В рамках проекта разработаны различные запросы, которые позволяют: выдать список всех клиентов с их персональными данными. Выдать справку по заказам конкретного клиента. Сформировать список строительных бригад с указанием членов и их должностей. По каждой бригаде вывести выполненные работы за указанный интервал времени. Рассчитать стоимость всех заказов. Определить стоимость работ, выполненных каждой бригадой. Выдать список просроченных заказов. Подготовить список заказов, которые должны быть завершены в ближайший месяц. Сформировать отчет о платежах клиентов. Подготовить отчет о задолженности клиентов.

Ожидаемый экономический эффект от внедрения управляемой и автоматизированной базы данных «Строительная компания» заключается в сокращении временных затрат на выполнение запросов, обеспечении безопасного доступа и регистрации пользователей, а также в возможности быстрого восстановления базы данных в случае сбоев. Все эти аспекты делают базу данных неотъемлемой частью успешного управления строительной компанией.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Федорова Г. Н. Сопровождение информационных систем Москва Издательский центр «Академия» 2018
2. Лагоша О. Н. Сертификация информационных систем: учебное пособие для СПО / Лань, 2021.
3. Осетрова И.С. Администрирование MS SQL Server 2014 - СПб: Университет ИТМО, 2016.
4. Костюк. А. И. Администрирование баз данных и компьютерных сетей: учебное пособие / А. И. Костюк, Д. А. Беспалов; Южный федеральный университет. - Ростов-на-Дону: Таганрог: Издательство Южного федерального университета, 2020.

Интернет-ресурсы

1. Платформа данных Майкрософт | Майкрософт – Электронный ресурс. Режим доступа: <https://www.microsoft.com/ru-ru/sql-server>
2. SQL Server Management Studio (SSMS) – Электронный ресурс. Режим доступа: <https://learn.microsoft.com/ru-ru/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>
3. Руководство по проектированию реляционных баз данных – Электронный ресурс. Режим доступа: <https://habr.com/ru/articles/193136/>
4. Что такое ER‑диаграмма – Электронный ресурс. Режим доступа: <https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/>
5. Шифрование базы данных — Википедия – Электронный ресурс. Режим доступа: <https://ru.wikipedia.org/wiki/Шифрование_базы_данных>