**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 «Отдел кадров»**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_/Зайцев Владислав Сергеевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc188347986)

[ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ 4](#_Toc188347987)

[1.1 СУБД SQL Server 4](#_Toc188347988)

[1.2 Microsoft SQL Server Management Studio 6](#_Toc188347989)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 8](#_Toc188347990)

[2.1 Разработка диаграммы ERD 8](#_Toc188347991)

[2.2 Словарь данных 10](#_Toc188347992)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 13](#_Toc188347993)

[3.1 Разработка базы данных 13](#_Toc188347994)

[3.2 Введение в безопасность SQL Server 17](#_Toc188347995)

[3.3 Управление безопасностью уровня сервера 19](#_Toc188347996)

[3.4 Управление участниками уровня базы данных 20](#_Toc188347997)

[3.5 Управление разрешениями уровня базы данных 21](#_Toc188347998)

[3.6 Резервное копирование баз данных 22](#_Toc188347999)

[3.7 Шифрование данных баз данных 23](#_Toc188348000)

[ЗАКЛЮЧЕНИЕ 25](#_Toc188348001)

[СПИСОК ЛИТЕРАТУРЫ 27](#_Toc188348002)

# ВВЕДЕНИЕ

Кинотеатры играют важную роль в культурной жизни общества, предоставляя людям возможность наслаждаться фильмами и отвлекаться от повседневной рутины. Разработка базы данных для кинотеатра является ключевым шагом в повышении эффективности управления данными и ресурсами, а также в улучшении обслуживания посетителей.

Данная задача требует глубокого понимания потребностей и особенностей кинотеатрального бизнеса, а также навыков построения функциональной и эффективной базы данных. В данном курсовом проекте мы рассмотрим процесс разработки базы данных кинотеатра, а также посмотрим модели, отвечающие всем требованиям кинотеатральной индустрии.

**Объект:** база данных «Кинотеатр».

Предмет: автоматизация бизнес - процесса «Кинотеатра»

**Цель работы –** разработать базу данных для ИС «Кинотеатр».

**Задачи:**

* выбор инструментария;
* проектирование базы данных;
* выполнить шифрование данных;
* выполнить BackUP базы данных;
* выполнить восстановление базы данных;
* выполнить разграничение прав доступа базы данных;
* создать базу 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 ANSI/ISO по 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

Диаграммы «сущность-связь» (ER-диаграммы) представляют собой стандартный инструмент для описания данных и их взаимосвязей в рамках определенной предметной области. Они включают в себя сущности и связи, которые отражают ключевые бизнес-правила. В отличие от функциональных диаграмм, которые акцентируют внимание на процессах, ER-диаграммы сосредоточены на структурировании данных.

Основные элементы ER-модели включают сущности, атрибуты и связи.

Сущность представляет собой класс объектов, которые могут быть реальными или абстрактными (например, люди, события или предметы), и информация о которых важна для анализа данных. Структура данных описывает правила и ограничения, которые определяют связи между различными элементами данных.

Каждая сущность должна обладать следующими характеристиками:

* **Уникальное имя** — для четкой идентификации сущности.
* **Один или несколько атрибутов**, которые принадлежат этой сущности или могут быть унаследованы через связи с другими сущностями.
* **Один или несколько атрибутов**, которые позволяют однозначно идентифицировать каждый экземпляр данной сущности.

**Атрибут** — это любая характеристика сущности, которая имеет значение в контексте рассматриваемой области и используется для квалификации, идентификации, классификации, количественной оценки или описания состояния сущности. Таким образом, атрибут представляет собой определенный тип характеристик или свойств, связанных с множеством реальных или абстрактных объектов. Экземпляр атрибута — это конкретная характеристика определенного экземпляра сущности. Атрибуты могут быть разделены на ключевые, то есть те, которые входят в состав уникального идентификатора ключа, и описательные - прочие.

Первичный ключ — это атрибут или комбинация атрибутов, а также связей, которые служат для уникальной идентификации каждого экземпляра сущности. Он представляет собой набор признаков, позволяющих отличить один объект от другого.

Описательные атрибуты могут быть обязательными или необязательными.

Обязательные атрибуты всегда имеют определенное значение для каждой сущности, тогда как необязательные могут не иметь значения. Обязательные описательные атрибуты помечаются символом «\*», а необязательные — символом «о».

Связь — это отношение между одной сущностью и другой, или же между экземплярами одной и той же сущности. Каждая связь может иметь одну из двух модальностей. Если хотя бы один экземпляр одной сущности связан с хотя бы одним экземпляром другой сущности, то такая связь считается обязательной. Необязательная связь, в свою очередь, представляет собой условное отношение между сущностями.

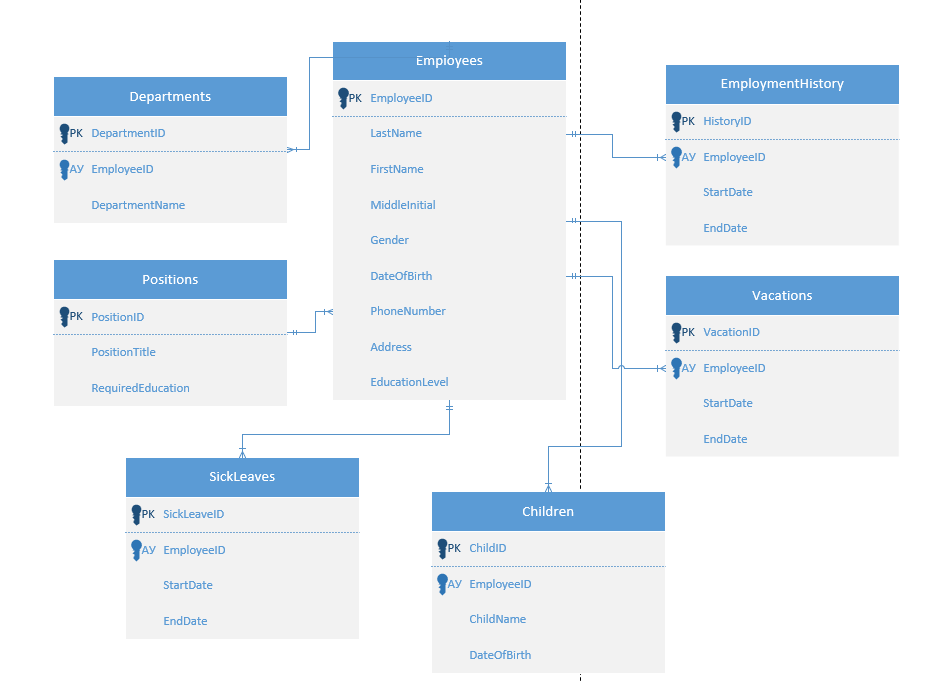


Рис 2.1 Диаграмма ERD (Entity Relationship Diagram)

# 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Словарь данных | | | | |
| Employee | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | EmployeeID | INT | Yes | Уникальный идентификатор сотрудника. |
|  | LastName | VARCHAR(50) | Yes | Фамилия сотрудника. |
|  | FirstName | VARCHAR(50) | Yes | Имя сотрудника. |
|  | MiddleInitial | CHAR(1) | No | Отчество сотрудника (первая буква). |
|  | Gender | CHAR(1) | Yes | Пол сотрудника ('M' - мужской, 'F' - женский). |
|  | DateOfBirth | DATE | Yes | Дата рождения сотрудника. |
|  | PhoneNumber | VARCHAR(15) | No | Номер телефона сотрудника. |
|  | Address | VARCHAR(100) | No | Адрес проживания сотрудника. |
|  | EducationLevel | VARCHAR(50) | No | Уровень образования сотрудника. |
|  | PositionID | INT | Yes | Идентификатор должности (внешний ключ). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Positions | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| **PK** | PositionID | INT | Yes | Уникальный идентификатор должности. |
|  | PositionTitle | VARCHAR(50) | Yes | Название должности. |
|  | RequiredEducation | VARCHAR(50) | No | Требуемый уровень образования для данной должности. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Department | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| **PK** | DepartmentID | INT | Yes | Уникальный идентификатор отдела. |
|  | DepartmentName | VARCHAR(50) | Yes | Название отдела. |
| FK | EmployeID | INT | Yes | Идентификатор менеджера отдела (внешний ключ). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EmploymentHistory | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| **PK** | HistoryID | INT | Yes | Уникальный идентификатор записи истории трудовой деятельности. |
| FK | EmployeeID | INT | Yes | Идентификатор сотрудника (внешний ключ). |
|  | PreviousCompanyName | VARCHAR(100) | No | Название предыдущей компании, в которой работал сотрудник. |
|  | StartDate | DATE | Yes | Дата начала работы в предыдущей компании. |
|  | EndDate | DATE | No | - Дата окончания работы в предыдущей компании. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SickLeave | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| **PK** | SickLeaveID | INT | Yes | - Уникальный идентификатор записи о больничном. |
| FK | EmployeeID | INT | Yes | - Идентификатор сотрудника (внешний ключ). |
|  | StartDate | DATE | Yes | - Дата начала больничного листа. |
|  | EndDate | DATE | Yes | - Дата окончания больничного листа. |
| Vacations | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| **PK** | VacationID | INT | Yes | - Уникальный идентификатор записи об отпуске. |
| FK | EmployeeID | INT | Yes | - Идентификатор сотрудника (внешний ключ). |
|  | StartDate | DATE | Yes | - Дата начала отпуска. |
|  | EndDate | DATE | Yes | - Дата окончания отпуска. |
| **Children** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| **PK** | ChildID | INT | Уникальный идентификатор ребенка. | PRIMARY KEY, IDENTITY(1,1) |
| FK | EmployeeID | INT | Идентификатор сотрудника, родителя ребенка. | FOREIGN KEY, ссылается на Employees(EmployeeID) |
|  | ChildName | VARCHAR(50) | Имя ребенка. | Не может быть NULL |
|  | DateOfBirth | DATE | Дата рождения ребенка. | Не может быть NULL |

# ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

# 3.1 Разработка базы данных

USE EmployeeManagementSystem;

GO

-- Таблица Должности

CREATE TABLE Positions (

PositionID INT PRIMARY KEY IDENTITY(1,1),

PositionTitle VARCHAR(50),

RequiredEducation VARCHAR(50)

);

-- Таблица Сотрудники

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

LastName VARCHAR(50),

FirstName VARCHAR(50),

MiddleInitial CHAR(1),

Gender CHAR(1),

DateOfBirth DATE,

PhoneNumber VARCHAR(15),

Address VARCHAR(100),

EducationLevel VARCHAR(50),

PositionID INT,

FOREIGN KEY (PositionID) REFERENCES Positions(PositionID)

);

--Таблица Отделы

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY IDENTITY(1,1),

DepartmentName VARCHAR(50),

ManagerID INT,

FOREIGN KEY (ManagerID) REFERENCES Employees(EmployeeID)

);

-- Таблица История трудовой деятельности

CREATE TABLE EmploymentHistory (

HistoryID INT PRIMARY KEY IDENTITY(1,1),

EmployeeID INT,

PreviousCompanyName VARCHAR(100),

StartDate DATE,

EndDate DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

-- Таблица Больничные

CREATE TABLE SickLeaves (

SickLeaveID INT PRIMARY KEY IDENTITY(1,1),

EmployeeID INT,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

-- Таблица Отпуска

CREATE TABLE Vacations (

VacationID INT PRIMARY KEY IDENTITY(1,1),

EmployeeID INT,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

-- Вставка данных в таблицу Должности

INSERT INTO Positions (PositionTitle, RequiredEducation) VALUES

('Software Engineer', 'Bachelor''s Degree'),

('HR Manager', 'Master''s Degree'),

('Data Analyst', 'Bachelor''s Degree'),

('Project Manager', 'Bachelor''s Degree');

-- Вставка данных в таблицу Сотрудники

INSERT INTO Employees (LastName, FirstName, MiddleInitial, Gender, DateOfBirth, PhoneNumber, Address, EducationLevel, PositionID) VALUES

('Smith', 'John', 'A', 'M', '1985-06-15', '123-456-7890', '123 Elm St, Cityville', 'Bachelor''s Degree', 1),

('Doe', 'Jane', 'B', 'F', '1990-09-20', '098-765-4321', '456 Oak St, Townsville', 'Master''s Degree', 2),

('Johnson', 'Emily', 'C', 'F', '1988-12-30', '234-567-8901', '789 Pine St, Villageburg', 'Bachelor''s Degree', 3),

('Brown', 'Michael', 'D', 'M', '1982-03-25', '345-678-9012', '321 Maple St, Cityville', 'Bachelor''s Degree', 4);

-- Вставка данных в таблицу Отделы  
INSERT INTO Departments (DepartmentName, ManagerID) VALUES

('IT Department', 1),

('HR Department', 2);

-- Вставка данных в таблицу История трудовой деятельности

INSERT INTO EmploymentHistory (EmployeeID, PreviousCompanyName, StartDate, EndDate) VALUES

(1, 'Tech Solutions Inc.', '2010-01-01', '2015-12-31'),

(2, 'People First LLC.', '2016-01-01', '2020-05-31'),

(3, 'Data Insights Corp.', '2015-06-01', '2019-11-30'),

(4, 'Project Management Co.', '2014-07-01', NULL);

-- Вставка данных в таблицу Больничные

INSERT INTO SickLeaves (EmployeeID, StartDate, EndDate) VALUES

(1, '2023-01-10', '2023-01-15'),

(2, '2023-02-20', '2023-02-25');

-- Вставка данных в таблицу Отпуска

INSERT INTO Vacations (EmployeeID, StartDate, EndDate) VALUES

(3, '2023-06-01', '2023-06-10'),

(4, '2023-07-15', '2023-07-20');

1. Выдать личную карточку сотрудника





*Рис3.1 Личная карточка сотрудника*

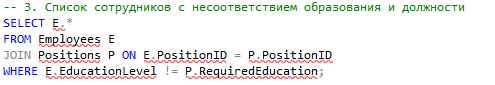
1. Сформировать список сотрудников без высшего образования





*Рис3.2 Список свободных мест*

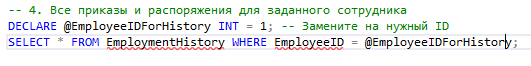
1. Сформировать список сотрудников с несоответствием образования и должности





*Рис3.3 Список сотрудников с несоответствием образования и должности*

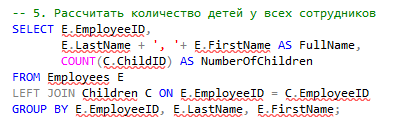
1. Выдать список приказов и распоряжений для заданного сотрудника

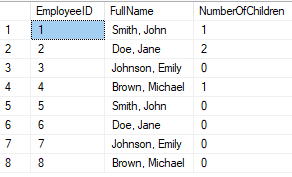




*Рис3.4 Список приказов и распоряжений сотрудника*

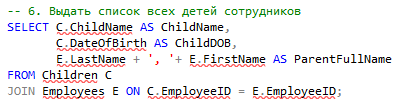
1. Рассчитать количество детей у всех сотрудников

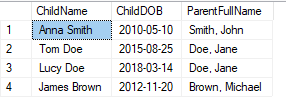




*Рис3.5 Количество детей сотрудников*

1. Выдать список всех детей сотрудников

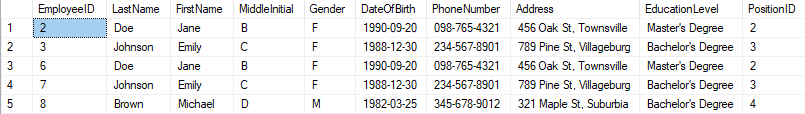




*Рис3.6 Список всех детей сотрудников*

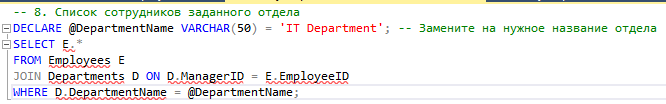
1. Вывести список сотрудников с проживанием вне города





*Рис3.7 Список сотрудников с проживанием вне города*

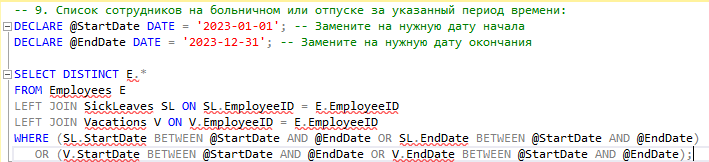
1. Вывести список сотрудников заданного отдела

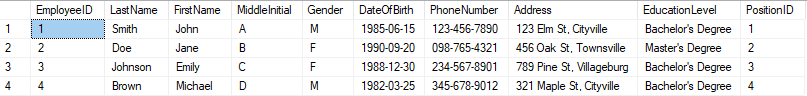




*Рис3.8 Список сотрудников заданного отдела*

1. Вывести список сотрудников на больничном или отпуске за указанный период времени





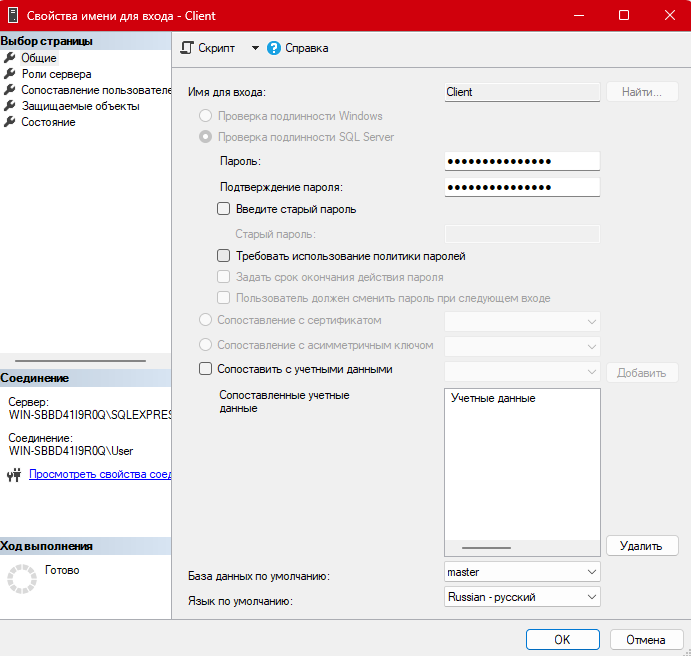
*Рис3.9 Список сотрудников на больничном или отпуске за указанный период времени*

# 3.2 Введение в безопасность SQL Server

Среда MS SQL Server предоставляет множество различных функций для создания безопасных приложений баз данных. В каждой версии MS SQL Server есть свои средства безопасности, как и в каждой версии Windows, при этом возможности более поздних версий шире, чем возможности более ранних. Важно понимать, что сами по себе средства безопасности не могут гарантировать защиту приложения базы данных. Каждое приложение базы данных имеет уникальные требования, среду выполнения, модель развертывания, физическое расположение и количество пользователей. Некоторым приложениям, работающим локально, необходима минимальная защита, тогда как другим локальным приложениям или приложениям, развернутым через Интернет, могут требоваться строгие меры безопасности вместе с постоянным мониторингом и контролем.

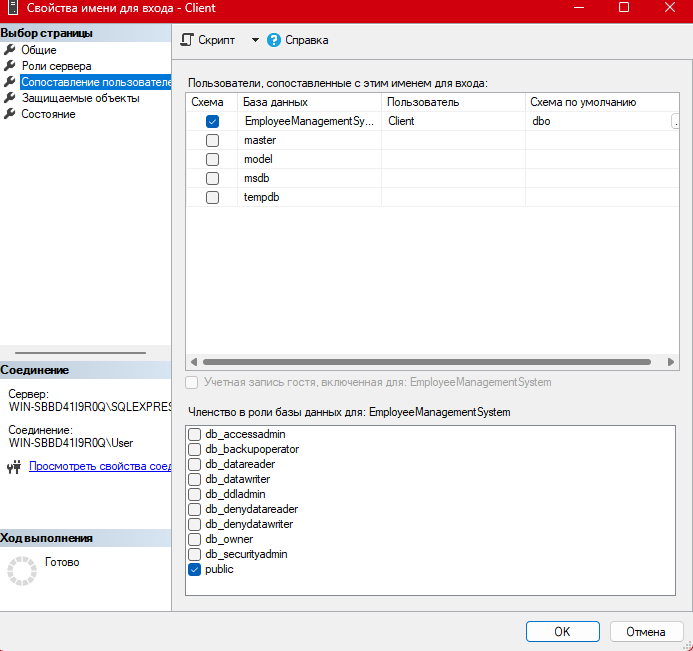
Помимо стандартных для системы управления базами данных функций, MS SQL Server содержит большой набор интегрированных служб по анализу данных. Доступ к данным, расположенным на MS SQL Server могут получить любые приложения, разработанные с помощью технологии.Net и среды разработки Visual Studio. Для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server используется инструментарий утилиты SQL Server Management Studio. В ней существует поддержка ряда компонент и средств по созданию и управлению базами данных, средств аналитической обработки данных (Analysis Services), средств отчётности (Reporting Services), а также множество средств, упрощающих разработку приложений.

# 3.3 Управление безопасностью уровня сервера

Создание пользователя и присвоение ему роли для базы данных. 

*Рис3.10 Добавление пользователя*

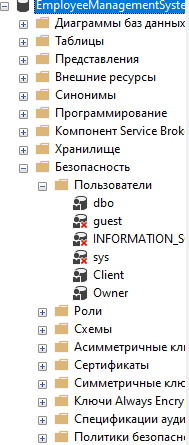
Подключение пользователя к базе данных



*Рис3.11 Разграничение прав доступа к базе данных*

# 3.4 Управление участниками уровня базы данных

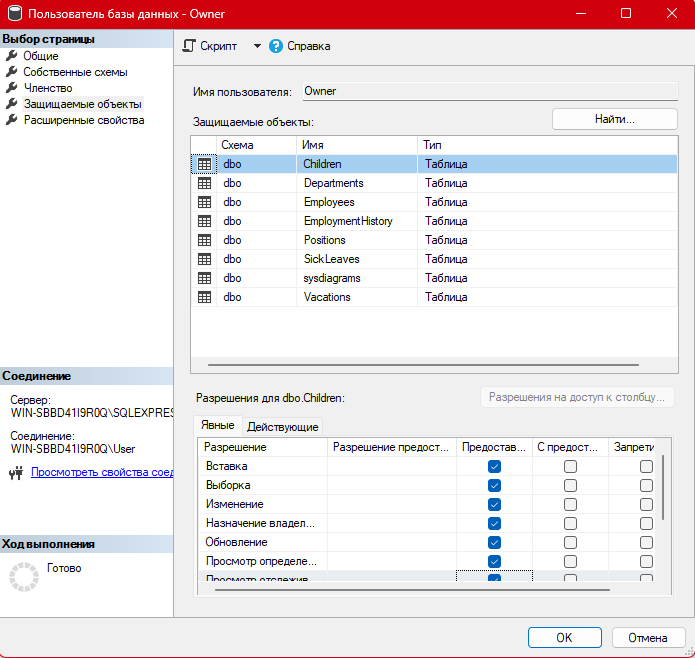
Создание двух пользователей User и Admin на уровне сервера, для дальнейшего подключения их к базе данных.



*Рис3.12 Созданные пользователи в базе данных CinemaDB*

# 3.5 Управление разрешениями уровня базы данных

SQL Server использует системную базу данных для поддержания внутренних метаданных. Администратор базы данных сервера должен уметь управлять ими, а пользователь правильно ими пользоваться. В базе данных выдали права доступа User и Admin к таблицам и права пользования функциями в данных таблицах.



*Рис3.13 Выдача прав пользователю*

# 3.6 Резервное копирование баз данных

BackUp (резервное копирование) – это процесс создания копии данных с целью их сохранения и возможного восстановления в случае их потери, повреждения или удаления. Резервное копирование может быть выполнено как на физических носителях, таких как внешние жесткие диски, USB – накопители, так и в различных облачных хранилищах. Резервное копирование играет важную роль в обеспечении безопасности данных и защите от потери информации.

Cоздание BackUp для базы данных CinemaDB



*Рис3.14 Код для создания файла восстановления*



*Рис3.15 Файл восстановления*

Запрос восстановления файла с BackUp нашей базы данных

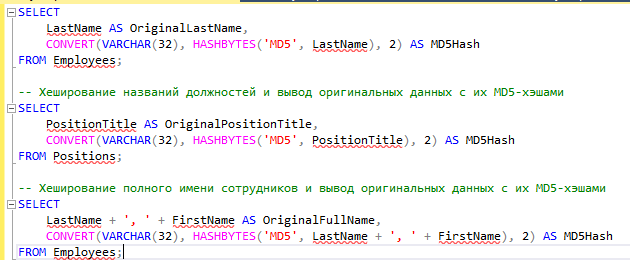


*Рис3.16 Код для восстановления базы данных*

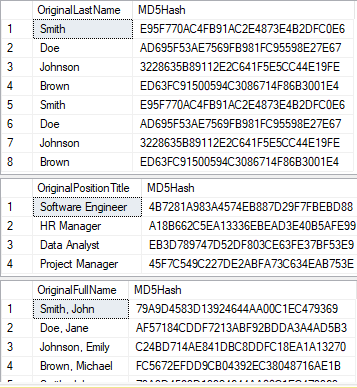
# 3.7 Шифрование данных баз данных

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

Данные шифруются перед записью на диск, а так же шифруются в самой программе во время ввода пароля и логина и сравнивается потом друг с другом в коде, что решает проблему защиты «неактивных» данных, но не обеспечивает сохранность информации при передаче по каналам связи или во время использования.



*Рис3.17 Запрос на шифрование данных*



*Рис3.18 Зашифрованные данные*

# ЗАКЛЮЧЕНИЕ

Разработка базы данных «Отдел кадров» представляет собой важный шаг в автоматизации процессов управления кадровой информацией в организации. Созданная система позволит эффективно хранить, обрабатывать и защищать данные о сотрудниках, их должностях, отпусках, больничных и других аспектах кадрового учета.

Функциональные характеристики системы включают ввод, хранение и поиск информации, а также шифрование данных и возможность их восстановления. Эти требования обеспечивают надежность работы системы и защиту от несанкционированного доступа. Кроме того, предусмотрены механизмы для резервного копирования базы данных, что является важным аспектом для обеспечения сохранности информации.

В заключение можно отметить, что разработка базы данных «Отдел кадров» значительно повысит эффективность работы отдела кадров, сократит время на обработку информации и снизит вероятность ошибок при формировании отчетов. Ожидается, что внедрение данной системы приведет к улучшению качества управления кадровыми данными и оптимизации работы всей организации.

# СПИСОК ЛИТЕРАТУРЫ

1. Андреев, С. В. (2020). Разработка и внедрение системы управления кадрами на основе информационных технологий. Москва: Издательство "Инфра-М".
2. Богданов, И. П. (2022). Автоматизация процессов управления персоналом с использованием СУБД. Санкт-Петербург: Питер.
3. Воронова, Н. В. (2021). Практическое руководство по созданию баз данных для кадрового учета. Новосибирск: Сибирское университетское издательство.
4. Герасимов, В. А. (2019). Информационные системы в управлении персоналом: современное состояние и перспективы. Москва: Наука.
5. Дмитриева, Е. В. (2023). Базы данных для HR-менеджеров: создание и использование информационных систем в отделах кадров. Казань: Издательство Казанского университета.
6. Куликов, А. Ю. (2021). Управление персоналом с использованием ERP-систем. Екатеринбург: Уральское издательство.
7. Петров, М. И. (2020). Применение реляционных баз данных в управлении кадрами. Москва: Издательский дом "Бизнес-Гармония".
8. Смирнова, О. Д. (2024). Современные подходы к автоматизации кадрового учета с использованием баз данных. Санкт-Петербург: Издательство "Лань".
9. Федорова, Т. В. (2022). Информационные технологии в кадровом учете: от теории к практике. Ростов-на-Дону: Феникс.
10. Шевченко, А. К. (2021). Современные информационные системы в сфере HR: анализ, проектирование и внедрение. Москва: ГУ-ВШЭ.