**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 «Кинотеатр»**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_/Коромыслов Ярослав Николаевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc187930002)

[ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ 4](#_Toc187930003)

[1.1 СУБД SQL Server 4](#_Toc187930004)

[1.2 Microsoft SQL Server Management Studio 6](#_Toc187930005)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 8](#_Toc187930006)

[2.1 Разработка диаграммы ERD 8](#_Toc187930007)

[2.2 Словарь данных 10](#_Toc187930008)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 12](#_Toc187930009)

[3.1 Разработка базы данных 12](#_Toc187930010)

[3.2 Введение в безопасность SQL Server 19](#_Toc187930011)

[3.3 Управление безопасностью уровня сервера 20](#_Toc187930012)

[3.4 Управление участниками уровня базы данных 21](#_Toc187930013)

[3.5 Управление разрешениями уровня базы данных 22](#_Toc187930014)

[3.6 Резервное копирование баз данных 23](#_Toc187930015)

[3.7 Шифрование данных баз данных 24](#_Toc187930016)

[ЗАКЛЮЧЕНИЕ 25](#_Toc187930017)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕАРТУРЫ 26](#_Toc187930018)

# 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;
* создать базу данных;
* выполнить шифрование данных;
* выполнить BackUP базы данных;
* выполнить восстановление базы данных;
* выполнить разграничение прав доступа 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 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.

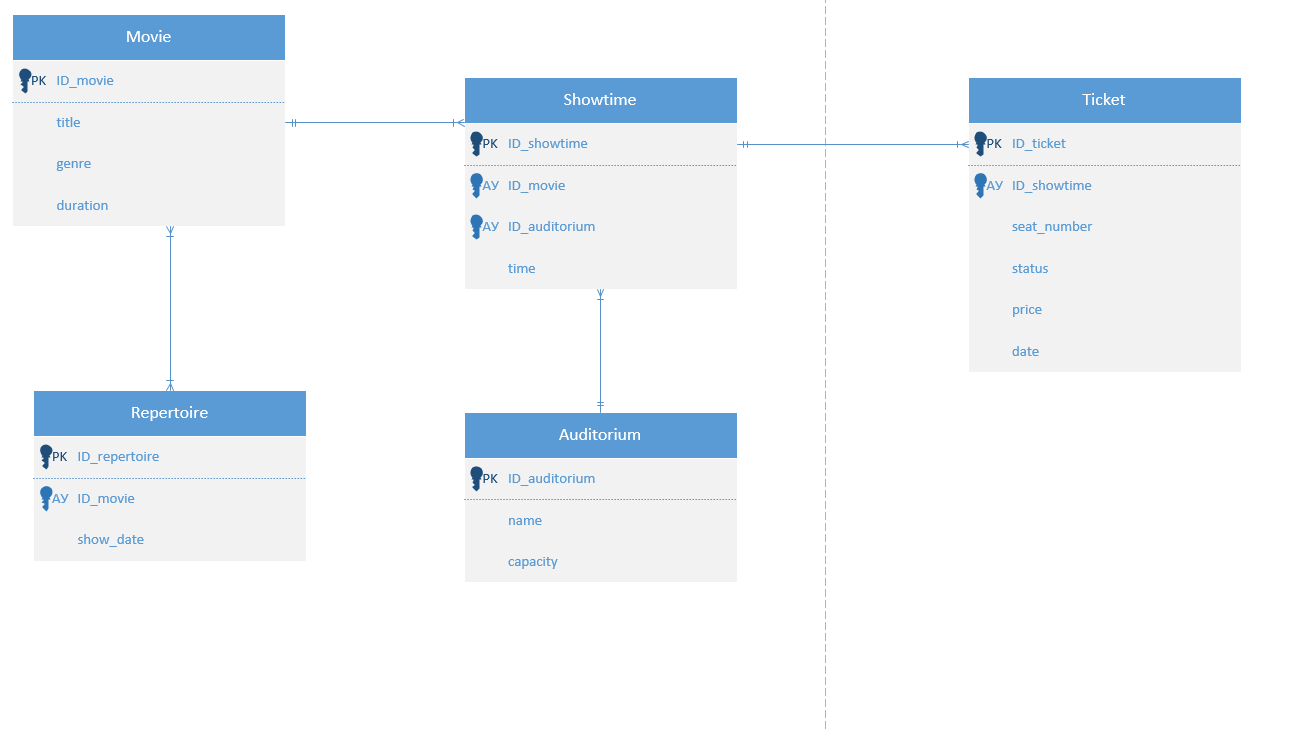


Рис 2.1 Диаграмма ERD (Entity Relationship Diagram)

# 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 | | | | |
| Movie | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID\_movie** | INTEGER | Y | Значение должно быть уникальным для каждого фильма. |
|  | **Title** | INTEGER | N | Название фильма. Содержит название, под которым фильм известен. |
|  | **Genre** | INTEGER | N | Может содержать несколько жанров, разделённых запятыми. |
|  | **Duration** | DATETIME | N | Должно быть положительным целым числом. |

*Таблица 1. Словарь данных*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Showtime | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID\_showtime** | INTEGER | Y | Уникальный идентификатор для каждого сеанса показа. |
| AУ | **ID\_movie** | INTEGER | Y | Идентификатор фильма, который показывается. Ссылается на поле id\_movie в таблице "movie". |
| АУ | **ID\_auditorium** | INTEGER | Y | Идентификатор зала, в котором проходит показ. Ссылается на поле id\_auditorium в таблице "auditorium". |
|  | **Time** | DATETIME | N | Дата и время начала сеанса показа. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Repertoire** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID\_repertoire** | INTEGER | Y | Уникальный идентификатор для каждого сеанса показа. |
| АУ | **ID\_movie** | INTEGER | Y | Идентификатор фильма, который показывается. Ссылается на поле id\_movie в таблице "movie". |
|  | **show\_date** | INTEGER | N | Номер места в зале. Формат может варьироваться в зависимости от конфигурации зала (например, "A1", "B5"). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID\_ticket** | INTEGER | Y | Уникальный идентификатор для каждого сеанса показа. |
| AУ | **ID\_ showtime** | INTEGER | Y | Идентификатор фильма, который показывается. Ссылается на поле id\_movie в таблице "movie". |
|  | **seat\_number** | INTEGER | N | Номер места в зале. Формат может варьироваться в зависимости от конфигурации зала (например, "A1", "B5"). |
|  | **Status** | INTEGER | N | Статус билета (например, "продан", "cвободен", "отменён"). |
|  | **Price** | DECIMAL(10, 2) | N | Цена билета. Формат: числовое значение с двумя знаками после запятой. |
|  | **Date** | DATETIME | N | Дата и время покупки билета. Формат |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Auditorium** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID\_auditorium** | INTEGER | Y | Уникальный идентификатор для каждого зала. |
|  | **name** | INTEGER | Y | Номер зала. |
|  | **capacity** | DECIMAL(10, 2) | N | Вместимость данного зала |

# ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

# 3.1 Разработка базы данных

CREATE DATABASE CinemaDB;

USE CinemaDB;

-- Таблица Auditorium

CREATE TABLE Auditorium (

ID\_auditorium INT PRIMARY KEY,

Name VARCHAR(255),

Capacity INT

);

-- Таблица Movie

CREATE TABLE Movie (

ID\_movie INT PRIMARY KEY,

Title VARCHAR(255),

Genre VARCHAR(100),

Duration INT

);

-- Таблица Showtime

CREATE TABLE Showtime (

ID\_showtime INT PRIMARY KEY,

ID\_movie INT,

ID\_auditorium INT,

Time DATETIME,

FOREIGN KEY (ID\_movie) REFERENCES Movie(ID\_movie),

FOREIGN KEY (ID\_auditorium) REFERENCES Auditorium(ID\_auditorium)

);

-- Таблица Ticket

CREATE TABLE Ticket (

ID\_ticket INT PRIMARY KEY,

ID\_showtime INT,

seat\_number VARCHAR(10),

status VARCHAR(20),

price DECIMAL(10,2),

date DATETIME,

FOREIGN KEY (ID\_showtime) REFERENCES Showtime(ID\_showtime)

);

-- Таблица Repertoire

CREATE TABLE Repertoire (

ID\_repertoire INT PRIMARY KEY,

ID\_movie INT,

show\_date DATETIME,

FOREIGN KEY (ID\_movie) REFERENCES Movie(ID\_movie)

);

-- Заполнение таблицы Auditorium

INSERT INTO Auditorium (ID\_auditorium, Name, Capacity) VALUES

(1, 'Main Hall', 200),

(2, 'VIP Hall', 50),

(3, 'Small Hall', 100);

-- Заполнение таблицы Movie

INSERT INTO Movie (ID\_movie, Title, Genre, Duration) VALUES

(1, 'Inception', 'Sci-Fi', 148),

(2, 'The Godfather', 'Crime', 175),

(3, 'Toy Story', 'Animation', 81);

-- Заполнение таблицы Showtime

INSERT INTO Showtime (ID\_showtime, ID\_movie, ID\_auditorium, Time) VALUES

(1, 1, 1, '2024-01-01 18:00:00'),

(2, 2, 1, '2024-01-01 20:30:00'),

(3, 3, 2, '2024-01-02 15:00:00');

-- Заполнение таблицы Ticket

INSERT INTO Ticket (ID\_ticket, ID\_showtime, seat\_number, status, price, date) VALUES

(1, 1, 'A1', 'sold', 10.00, '2024-01-01'),

(2, 1, 'A2', 'sold', 10.00, '2024-01-01'),

(3, 2, 'B1', 'available', 12.00, '2024-01-01'),

(4, 3, 'C1', 'sold', 8.00, '2024-01-02');

-- Заполнение таблицы Repertoire

INSERT INTO Repertoire (ID\_repertoire, ID\_movie, show\_date) VALUES

(1, 1, '2024-01-01'),

(2, 2, '2024-01-01'),

(3, 3, '2024-01-02');

SELECT m.Title, r.show\_date

FROM Repertoire r

JOIN Movie m ON r.ID\_movie = m.ID\_movie

WHERE r.show\_date BETWEEN '2024-01-01' AND '2024-01-02';

SELECT t.seat\_number

FROM Ticket t

WHERE t.ID\_showtime = 1 AND t.status = 'available';

SELECT t.seat\_number

FROM Ticket t

WHERE t.ID\_showtime = 1 AND t.status = 'sold';

SELECT DISTINCT m.Title

FROM Movie m

JOIN Showtime s ON m.ID\_movie = s.ID\_movie

WHERE s.Time BETWEEN '2024-01-01 00:00:00' AND '2024-01-01 23:59:59';

SELECT SUM(t.price) AS total\_sales

FROM Ticket t

WHERE t.ID\_showtime = 1 AND t.status = 'sold';

SELECT SUM(t.price) AS total\_unsold

FROM Ticket t

WHERE t.ID\_showtime = 1 AND t.status = 'available';

SELECT SUM(t.price) AS total\_sales\_period

FROM Ticket t

JOIN Showtime s ON t.ID\_showtime = s.ID\_showtime

WHERE s.Time BETWEEN '2024-01-01' AND '2024-01-02' AND t.status = 'sold';

SELECT SUM(t.price) AS total\_losses

FROM Ticket t

JOIN Showtime s ON t.ID\_showtime = s.ID\_showtime

WHERE s.Time BETWEEN '2024-01-01' AND '2024-01-02' AND t.status = 'available';

SELECT SUM(t.price) AS total\_sales

FROM Ticket t

JOIN Repertoire r ON r.ID\_movie = t.ID\_showtime

WHERE r.ID\_movie = 3 AND t.status = 'sold'; -- Замените `1` на нужный ID фильма.

SELECT s.Time,

COUNT(CASE WHEN t.status = 'sold' THEN 1 END) AS sold\_count,

COUNT(CASE WHEN t.status = 'available' THEN 1 END) AS available\_count

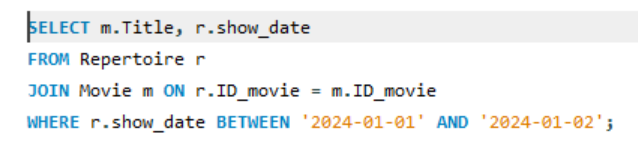
FROM Showtime s

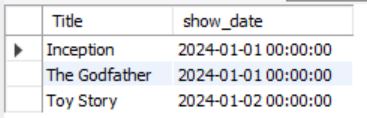
LEFT JOIN Ticket t ON s.ID\_showtime = t.ID\_showtime

WHERE s.ID\_movie = 3 -- Замените `1` на нужный ID фильма.

GROUP BY s.Time;

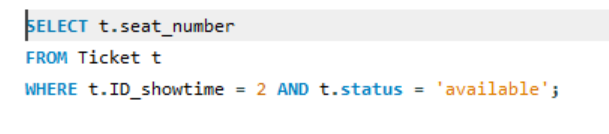
1. Выдать репертуар на заданный период времени

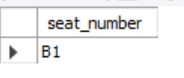




*Рис3.1 Репертуар на заданный период*

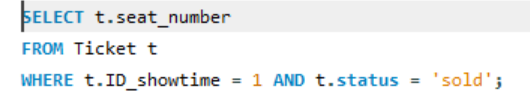
1. Сформировать список свободных мест на заданный киносеанс

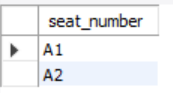




*Рис3.2 Список свободных мест*

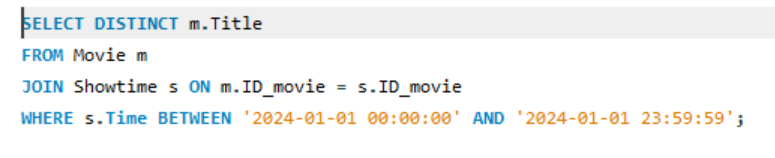
1. Сформировать список занятых мест на заданный киносеанс

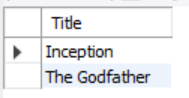




*Рис3.3 Список занятых мест*

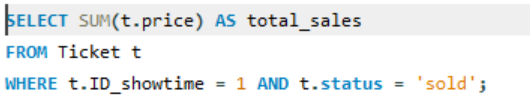
1. Выдать список кинофильмов на заданное (в течение суток) время

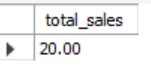




*Рис3.4 Список кинофильмов на заданное время*

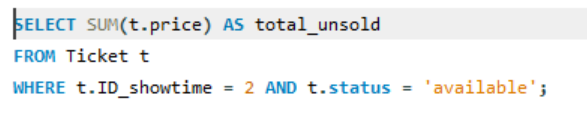
1. Рассчитать стоимость проданных билетов на заданный киносеанс

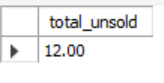




*Рис3.5Стоимость проданных билетов на заданный киносеанс*

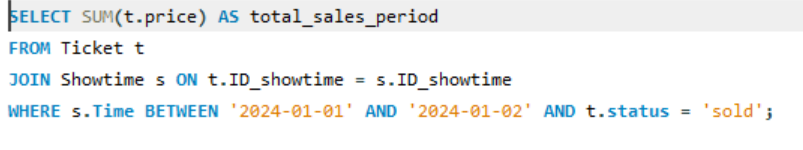
1. Рассчитать стоимость непроданных билетов на заданный киносеанс

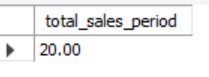




*Рис3.6 Стоимость непроданных билетов на заданный киносеанс*

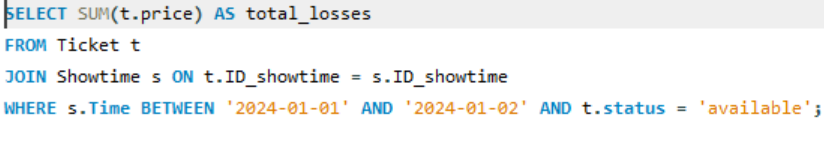
1. Рассчитать стоимость проданных билетов на заданный период времени

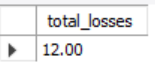




*Рис3.7 Стоимость проданных билетов на заданный период времени*

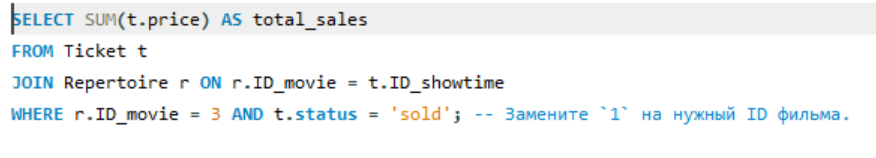
1. Рассчитать потери кинотеатра на заданный период времени

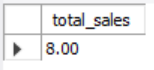




*Рис3.8 Потери кинотеатра на заданный период времени*

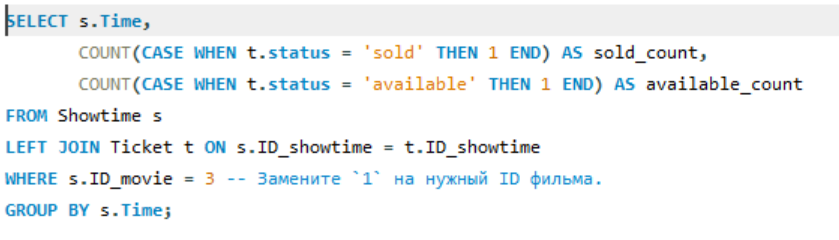
1. Для фильма из репертуара рассчитать стоимость проданных билетов

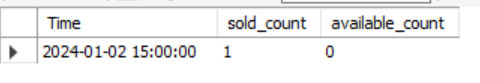




*Рис3.9 Стоимость проданных билетов из репертуара для фильма*

1. Для фильма из репертуара рассчитать количество проданных и непроданных билетов для каждого сеанса





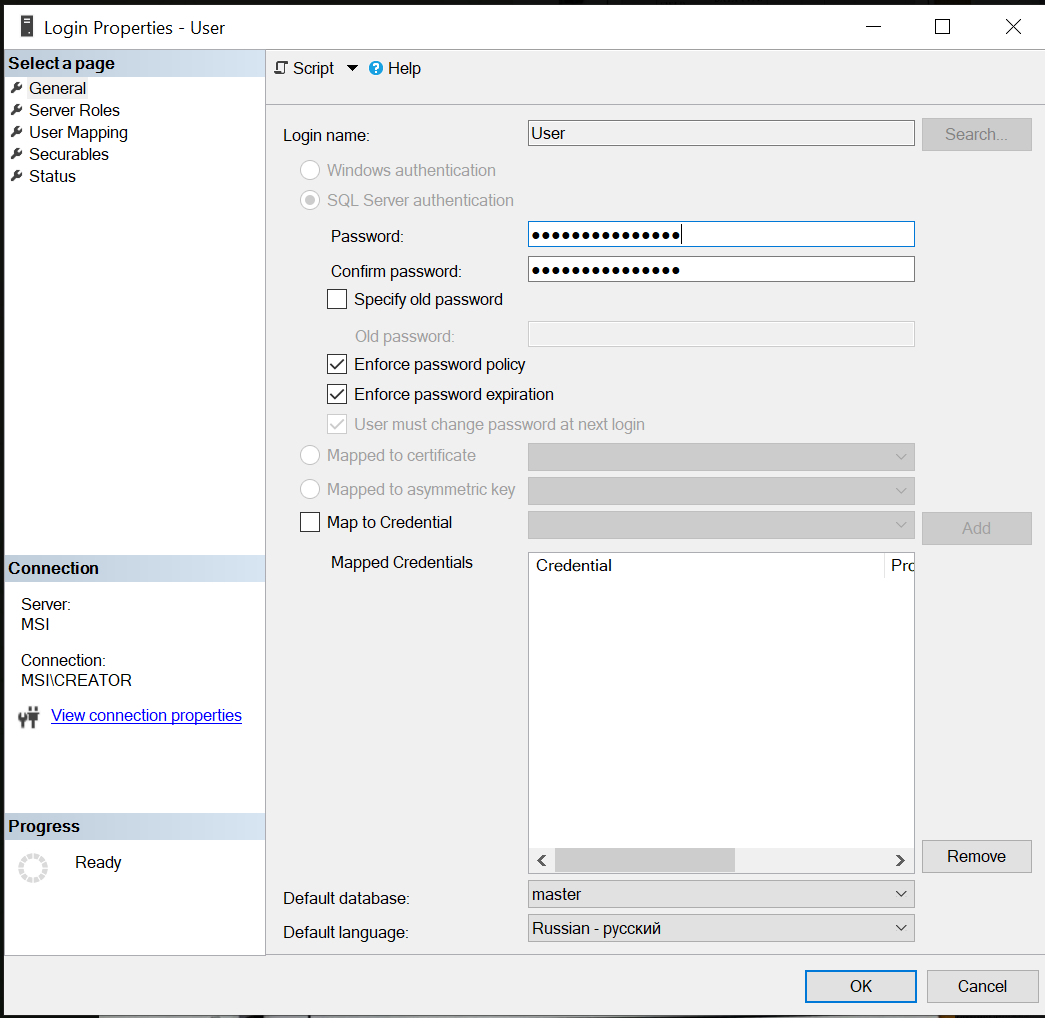
*Рис3.9 Количество проданных и непроданных билетов для каждого сеанса*

# 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.

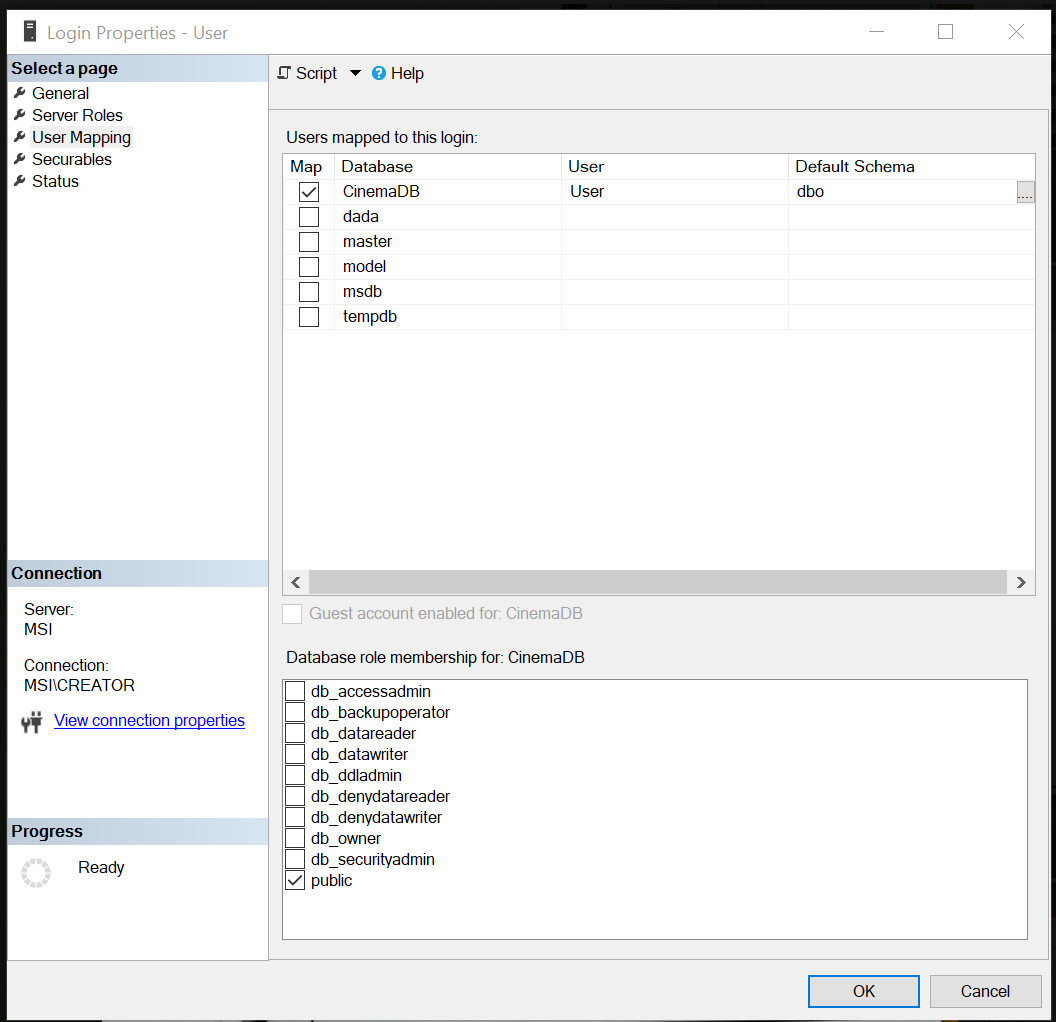
Помимо стандартных для системы управления базами данных функций, MS SQL Server содержит большой набор интегрированных служб по анализу данных. Доступ к данным, расположенным на MS SQL Server могут получить любые приложения, разработанные с помощью технологии.Net и среды разработки Visual Studio. Для 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.

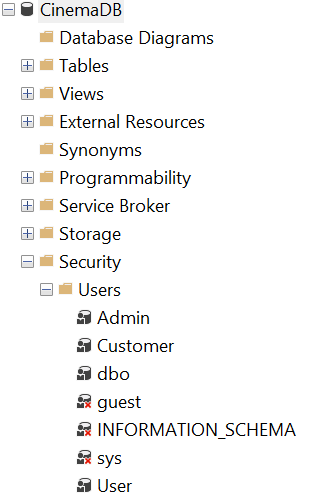
*Рис3.10 Добавление пользователя*

Подключение пользователя к базе данных



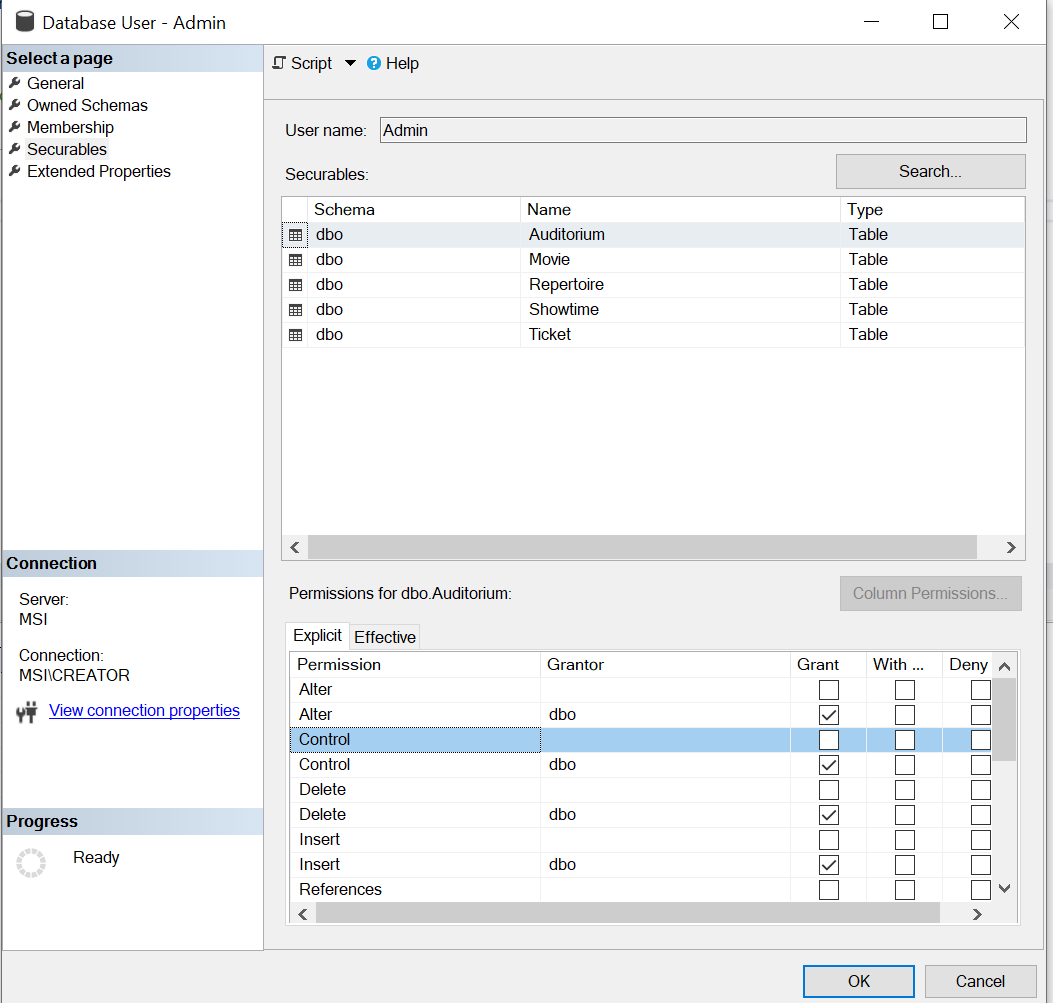
*Рис3.11 Разграничение прав доступа к базе данных*

# 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*

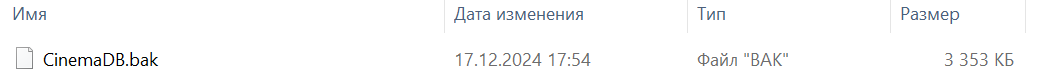
# 3.6 Резервное копирование баз данных

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

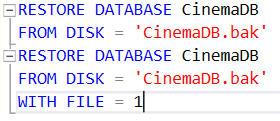


*Рис3.14 Код для создания файла восстановления*



*Рис3.15 Файл восстановления*

Запрос восстановления файла с BackUp нашей базы данных

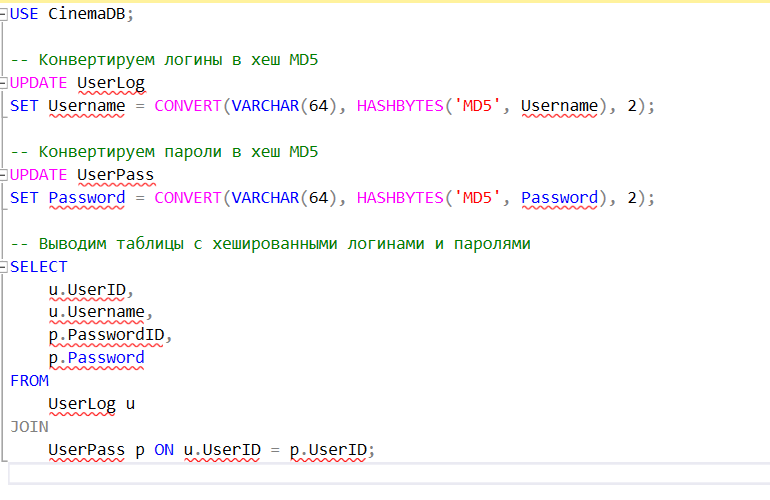


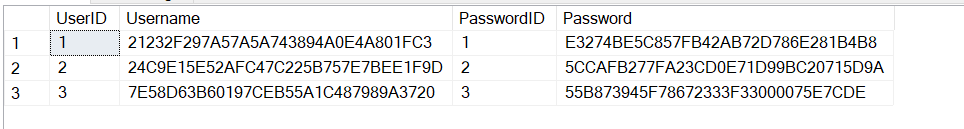
*Рис3.16 Код для восстановления базы данных*

# 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

База данных кинотеатра имеет большую актуальность в современном мире развлечений. развитием технологий и цифровизацией кинематографии, управление информацией о фильмах, сеансах, билетах, а также работа с базой клиентов и статистикой становится все более важным для эффективного функционирования кинотеатра. База данных позволяет автоматизировать процессы бронирования, продажи билетов, учета посещаемости и предпочтений зрителей, повышает качество обслуживания и управление ресурсами, что делает её неотъемлемой частью успешного кинотеатра.

При выполнение данной работы, а именно автоматизация базы данных «Кинотеатр», была разработана диаграмма Entity Relationship Diagram, была спроектирована база данных «Кинотеатр» в приложении Microsoft SQL Server Management Studio, был разработан интерфейс и функционал автоматизированной информационной системы с помощью C# WPF. Данная база данных и автоматизированная информационная система была разработана для сотрудников кинотеатра. С их помощью можно быстро создавать отчеты, редактировать, добавлять и удалять данных, пользоваться безопасным входом и регистрацией благодаря шифрованию.

Ссылка на на Github: https://github.com/pickachy777/Cinema

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕАРТУРЫ

1. ER – Diagramm Entities and connections 2023 [Электронный ресурс]. https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/
2. Введение в MySQL 2023 [Электронный ресурс]. https://metanit.com/sql/mysql/1.1.php
3. Краткое руководство. Подключение и запрос экземпляра SQL Server с помощью SQL Server Management Studio (SSMS) 2024
4. [Электронный ресурс]. https://learn.microsoft.com/ru-ru/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-ver16
5. Руководство по SQL 2022 [Электронный ресурс]. https://proselyte.net/tutorials/sql/
6. Резервное копирование и восстановление баз данных SQL Server 2025

[Электронный ресурс]. https://learn.microsoft.com/ru-ru/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases?view=sql-server-ver16

1. Соединение таблиц 2021[Электронный ресурс]. https://metanit.com/sql/mysql/5.1.php
2. Создание полной резервной копии базы данных 2025 [Электронный ресурс]. https://learn.microsoft.com/ru-ru/sql/relational-databases/backup-restore/create-a-full-database-backup-sql-server?view=sql-server-ver16
3. Создание пользователя базы данных 2025 [Электронный ресурс]. https://learn.microsoft.com/ru-ru/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver16
4. Шифрование в базах данных SQL Server 2023 [Электронный ресурс]. https://www.osp.ru/winitpro/2013/05/13035359