

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Систем автоматизированного проектирования

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Базы данных»
Тема: Проектирование базы данных

Студент гр. 2308

Попов Н.А.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

Оглавление

1. Введение	3
2. Диаграмма базы данных.....	7
3. Создание базы данных.....	7
4. Создание таблиц и ограничений целостности	8
5. Заполнение таблиц данными	13
6. Разработка объектов промежуточного слоя.....	17
7. Разработка стратегии резервного копирования	25
8. Описание структуры таблиц БД.....	26
9. Заключение	30
10. Список использованных источников	30

Вариант: 27

1. Введение

Работа художественной галереи музея

Предметная область — это организация и управление процессами, связанными с произведениями искусства в рамках музея или художественной галереи. Система должна учитывать не только сами произведения, но и художников, выставки, галереи, билеты, посетителей и их взаимодействие с произведениями искусства.

Основные элементы предметной области:

1. Художники (Artists):

- Данные о художниках, их биография, дата рождения, дата смерти, национальность и другие сведения.

2. Произведения искусства (Artworks):

- Каждое произведение искусства привязано к конкретному художнику и имеет свое описание, тип (картина, скульптура, фотография и т. д.), год создания и место экспонирования.

3. Выставки (Exhibitions):

- Выставки — это мероприятия, в которых участвуют различные произведения искусства. Для каждой выставки фиксируется название, дата начала и окончания, место проведения.

4. Галереи (Galleries):

- Галереи — это физические или виртуальные помещения, где проходят выставки произведений искусства. Они могут быть расположены в разных местах (например, в разных городах или странах), и для каждой галереи фиксируются контактные данные.

5. Билеты (Tickets):

- Билеты, которые могут быть приобретены посетителями для посещения выставок в галереях. Для каждого билета указаны тип (взрослый, детский, студенческий и т.д.), цена и информация о выставке и галерее.

6. Посетители (Visitors):

- Посетители — это люди, которые приходят в галерею, чтобы ознакомиться с произведениями искусства. Важно учитывать их контактные данные, историю посещений и купленных билетов.

7. Связь произведений искусства с выставками (ArtworkExhibitions):

- Каждое произведение искусства может участвовать в нескольких выставках. Таблица связи позволяет отслеживать, на каких выставках представлено конкретное произведение.

8. Информация о покупке билетов (VisitorTickets):

- Каждому посетителю может быть привязана информация о купленных билетах, которая помогает отслеживать, кто и на какую выставку приобрел билет, а также дату покупки.

Важность предметной области:

- Музейная галерея — это культурное учреждение, которое собирает, хранит и экспонирует произведения искусства.
- Взаимодействие между художниками, произведениями искусства, выставками и посетителями является ключевым процессом в работе музея.
- Правильное управление информацией о произведениях искусства, выставках, посетителях и билетах помогает улучшить организацию работы галереи, а также предоставлять нужную информацию для посетителей, организаторов и коллекционеров.

Структура БД

1. **Таблица "Artists"**: Хранит информацию о художниках.
2. **Таблица "Artworks"**: Содержит информацию о произведениях искусства.
3. **Таблица "Exhibitions"**: Описывает выставки, их даты и местоположение.
4. **Таблица "ArtworkExhibitions"**: Обеспечивает связь между произведениями искусства и выставками.
5. **Таблица "Galleries"**: Хранит данные о галереях.
6. **Таблица "Tickets"**: Информация о типах билетов, их ценах и связи с галереями/выставками.
7. **Таблица "Visitors"**: Данные о посетителях, их контактные данные и информацию о покупках.
8. **Таблица "VisitorTickets"**: Связь между посетителями и купленными билетами.

Пример схемы базы данных:

- **Artists** (ArtistID, FirstName, LastName, BirthDate, DeathDate, Nationality)
- **Artworks** (ArtworkID, Title, YearCreated, ArtistID, ArtType, Description)
- **Exhibitions** (ExhibitionID, ExhibitionName, StartDate, EndDate, Location)
- **ArtworkExhibitions** (ArtworkID, ExhibitionID)
- **Galleries** (GalleryID, GalleryName, Location, ContactPhone)
- **Tickets** (TicketID, TicketType, Price, GalleryID, ExhibitionID)
- **Visitors** (VisitorID, FirstName, LastName, Email, PhoneNumber)
- **VisitorTickets** (VisitorID, TicketID, PurchaseDate)

Пример бизнес-процессов в системе:

1. **Художник создает произведение искусства и передает его на выставку.**

2. **Произведение искусства выставляется на выставке**, где его могут увидеть посетители.
3. **Посетители покупают билеты** для посещения выставок, и это отражается в системе.
4. **Галерея предоставляет место** для выставок, организуя пространство и поддержку для произведений искусства.
5. **Выставки меняются** в зависимости от времени года или событий, а произведения искусства могут перемещаться между галереями и выставками.

Применение в реальной жизни:

- Система базы данных для галереи поможет автоматизировать процессы управления произведениями искусства, выставками, галереями и продажей билетов.
- Она облегчает организацию работы музея, отслеживание посетителей и поддержание актуальной информации о произведениях искусства.
- Также это важный инструмент для взаимодействия с клиентами и коллекционерами, предоставляющий полную информацию о выставках и произведениях искусства.

Таким образом, система базы данных для художественной галереи музея решает важные задачи по организации и хранению информации, улучшая процессы, связанные с выставками и взаимодействием с посетителями.

2. Диаграмма базы данных

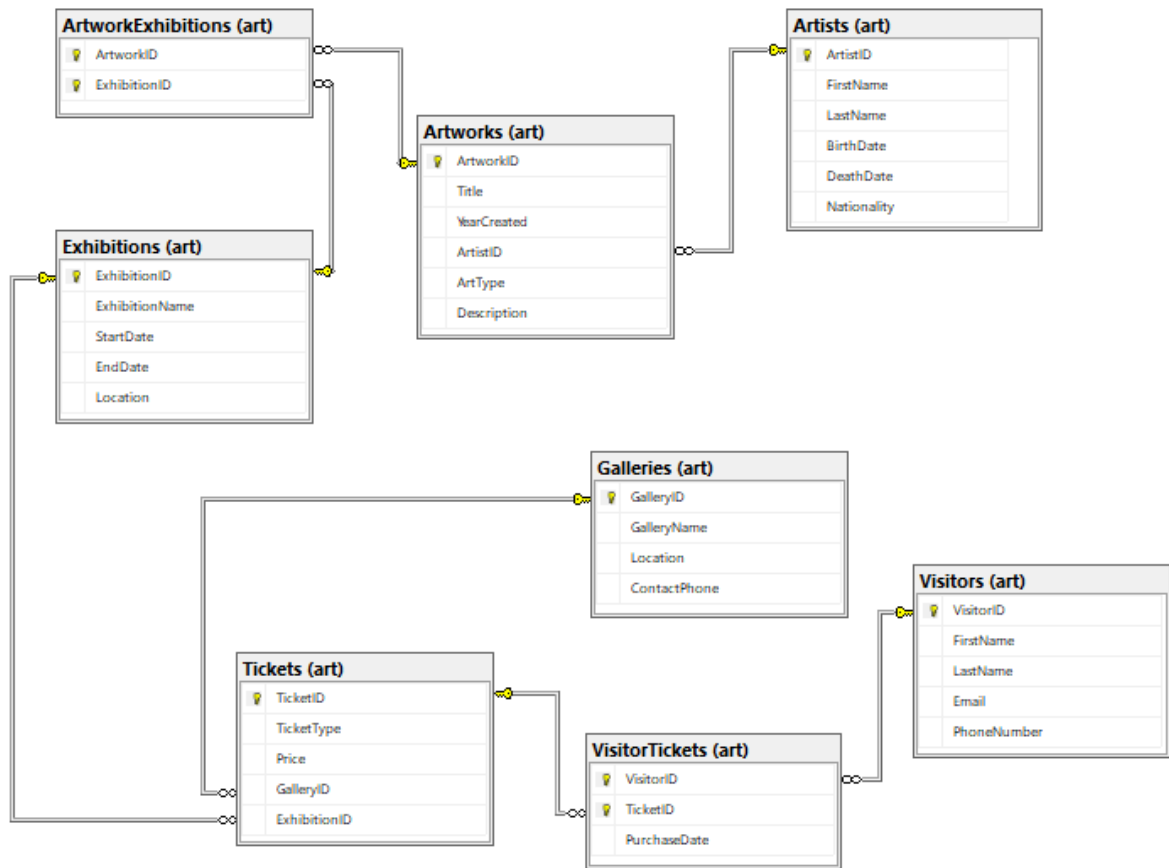


Рисунок 1 – Диаграмма базы данных

3. Создание базы данных

Для создания базы данных для выбранной предметной области воспользуемся интерфейсом SSMS, но также имеется возможность выполнить это через запрос вида:

```
CREATE DATABASE Art;  
GO
```

Кроме того, для того чтобы мы могли использовать свою схему, выполним следующую команду:

```
CREATE SCHEMA art;
```

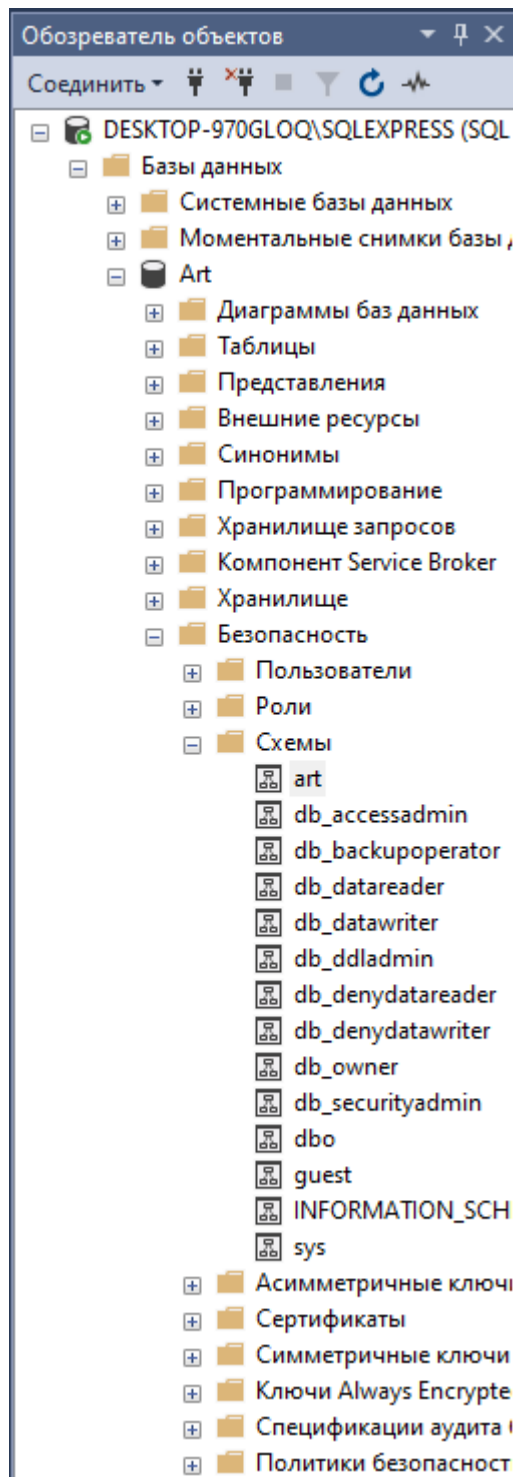


Рисунок 2 – Созданные база данных и схема

4. Создание таблиц и ограничений целостности

Создание таблицы Artist

```
CREATE TABLE art.Artists (
    ArtistID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор художника
    FirstName NVARCHAR(50) NOT NULL,        -- Имя художника
    LastName NVARCHAR(50) NOT NULL,         -- Фамилия художника
```



```

        BirthDate DATE, -- Дата рождения
        DeathDate DATE, -- Дата смерти (если применимо)
        Nationality NVARCHAR(50) -- Национальность
    );

```

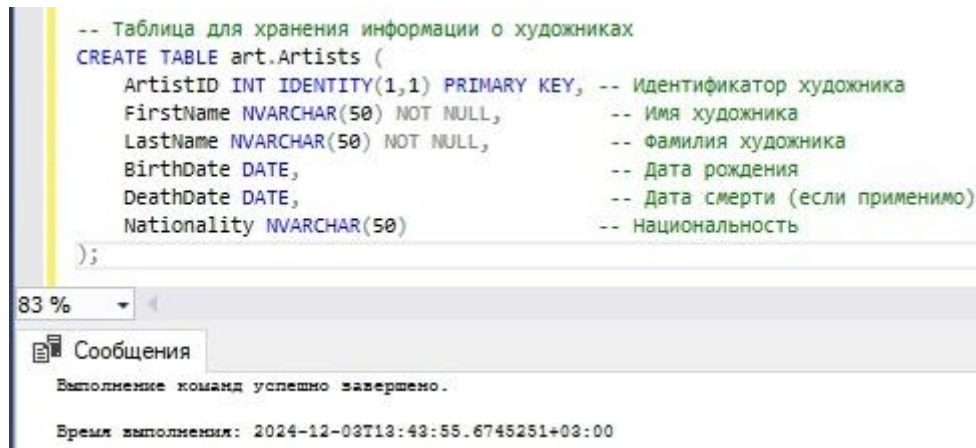


Рисунок 3 – Создание таблицы для хранения информации о художниках

Создание таблицы Artworks

```

CREATE TABLE art.Artworks (
    ArtworkID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор произведения искусства
    Title NVARCHAR(100) NOT NULL, -- Название произведения
    YearCreated INT, -- Год создания
    ArtistID INT, -- Идентификатор художника
    ArtType NVARCHAR(50), -- Тип искусства (картина, скульптура и т.д.)
    Description NVARCHAR(500), -- Описание произведения
    FOREIGN KEY (ArtistID) REFERENCES art.Artists(ArtistID) -- Внешний ключ на таблицу art.Artists
);

```

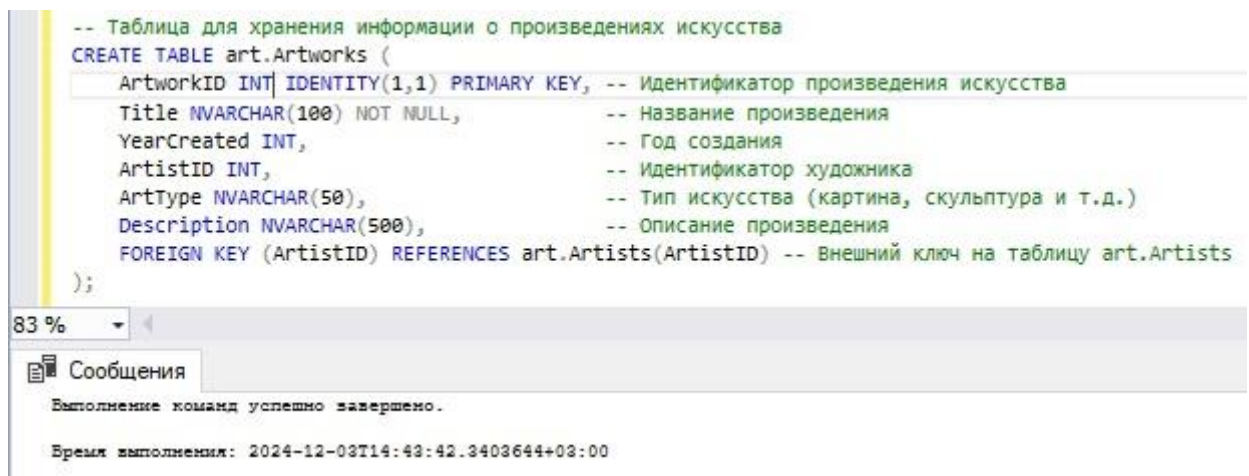


Рисунок 4 – Создание таблицы для хранения информации о произведениях искусства

Создание таблицы Exhibitions

```
CREATE TABLE art.Exhibitions (  
    ExhibitionID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор  
    -- выставки  
    ExhibitionName NVARCHAR(100) NOT NULL, -- Название выставки  
    StartDate DATE, -- Дата начала  
    EndDate DATE, -- Дата окончания  
    Location NVARCHAR(100) NOT NULL, -- Место проведения  
    CONSTRAINT CK_EndDateAfterStartDate CHECK (EndDate > StartDate) --  
    -- Ограничение: дата окончания должна быть больше даты начала  
);
```

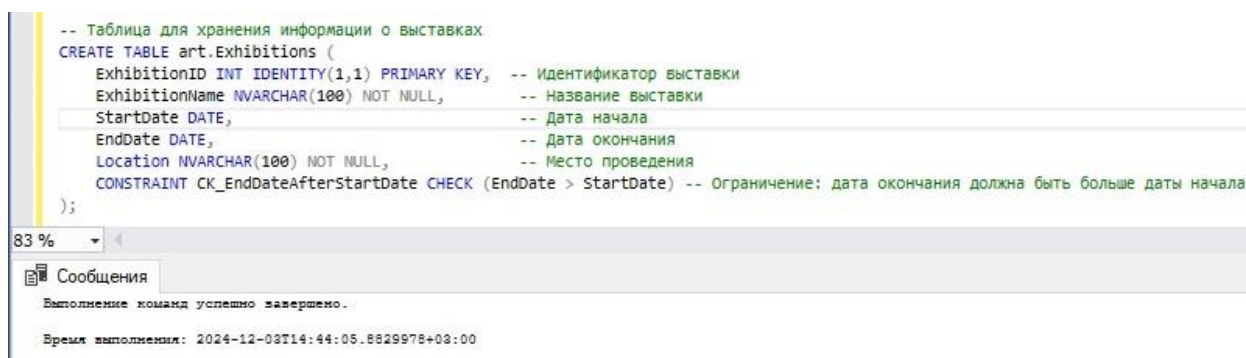


Рисунок 4 – Создание таблицы для хранения информации о выставках

Создание таблицы ArtworkExhibitions

```
CREATE TABLE art.ArtworkExhibitions (  
    ArtworkID INT, -- Идентификатор  
    -- произведения искусства  
    ExhibitionID INT, -- Идентификатор  
    -- выставки  
    PRIMARY KEY (ArtworkID, ExhibitionID), -- Составной  
    -- первичный ключ  
    FOREIGN KEY (ArtworkID) REFERENCES art.Artworks(ArtworkID) ON  
    DELETE CASCADE, -- Внешний ключ на таблицу art.Artworks  
    FOREIGN KEY (ExhibitionID) REFERENCES  
    art.Exhibitions(ExhibitionID) ON DELETE CASCADE -- Внешний ключ на  
    -- таблицу art.Exhibitions  
);
```

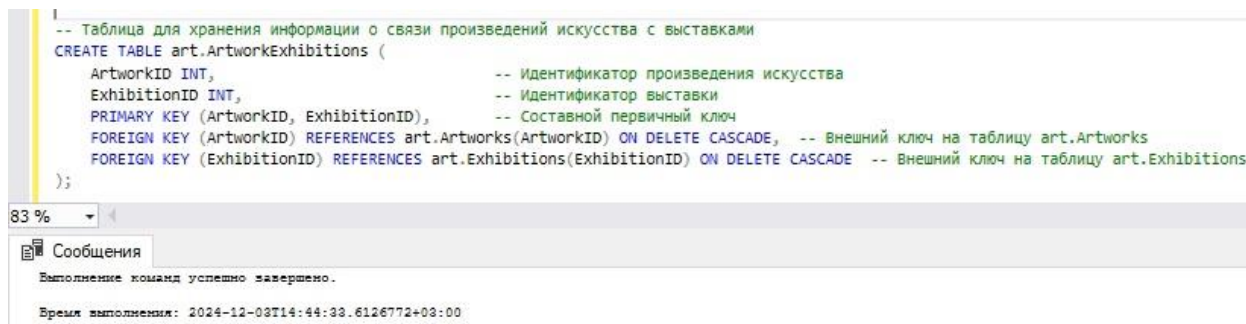


Рисунок 5 – Создание таблицы для хранения информации о связи произведений искусства с выставками

Создание таблицы Galleries

```
CREATE TABLE art.Galleries (  
    GalleryID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор галереи  
    GalleryName NVARCHAR(100) NOT NULL,      -- Название галереи  
    Location NVARCHAR(100),                  -- Местоположение  
    галереи  
    ContactPhone NVARCHAR(15) CHECK (ContactPhone NOT LIKE '%[^0-9]%' )  
)  
);
```

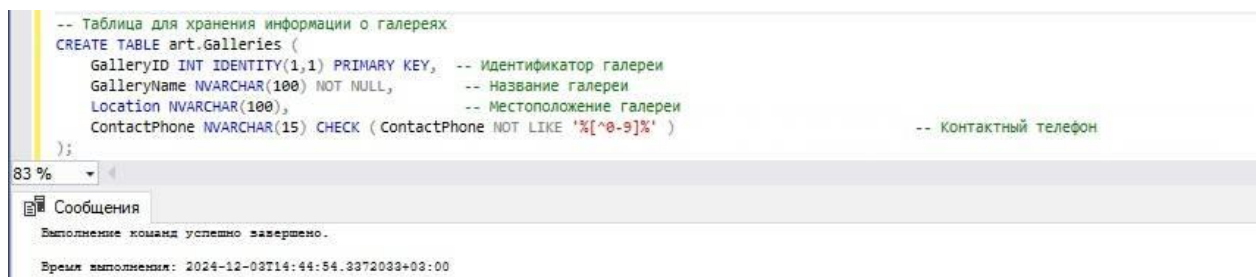


Рисунок 6 – Создание таблицы для хранения информации о галереях

Создание таблицы Tickets

```
CREATE TABLE art.Tickets (  
    TicketID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор билета  
    TicketType NVARCHAR(50) NOT NULL,      -- Тип билета  
    (например, входной, групповой)  
    Price DECIMAL(10, 2) CHECK (Price > 0) NOT NULL, -- Цена билета,  
    должна быть положительной  
    GalleryID INT,                          -- Идентификатор  
    галереи  
    ExhibitionID INT,                       -- Идентификатор  
    выставки  
    FOREIGN KEY (GalleryID) REFERENCES art.Galleries(GalleryID), --  
    Внешний ключ на таблицу art.Galleries  
    FOREIGN KEY (ExhibitionID) REFERENCES  
    art.Exhibitions(ExhibitionID) -- Внешний ключ на таблицу  
    art.Exhibitions  
)  
);
```

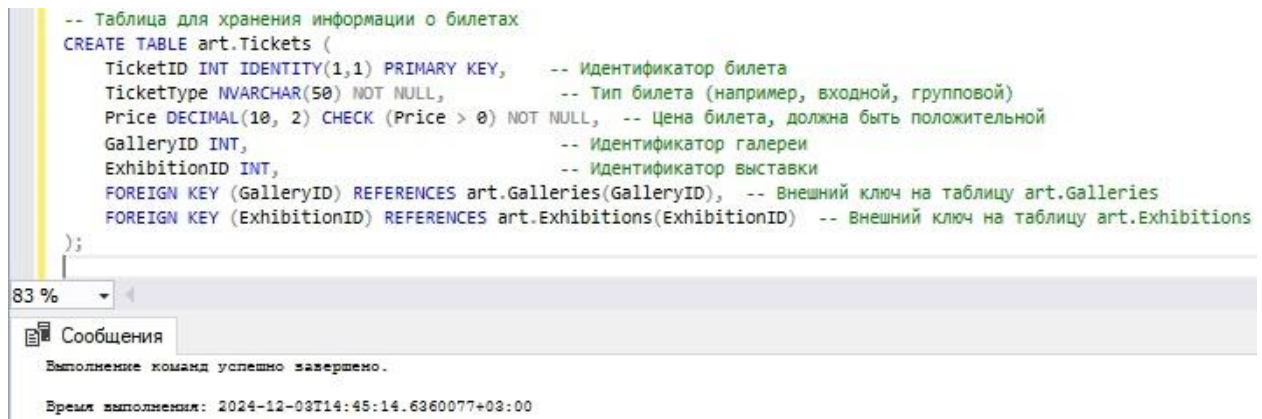


Рисунок 7 – Создание таблицы для хранения информации о билетах

Создание таблицы Visitors

```

CREATE TABLE art.Visitors (
    VisitorID INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор посетителя
    FirstName NVARCHAR(50) NOT NULL, -- Имя посетителя
    LastName NVARCHAR(50) NOT NULL, -- Фамилия посетителя
    Email NVARCHAR(100) CHECK (Email LIKE '%@yandex.ru'), -- Электронная почта должна заканчиваться на @yandex.ru
    PhoneNumber NVARCHAR(15) -- Телефон
);

```

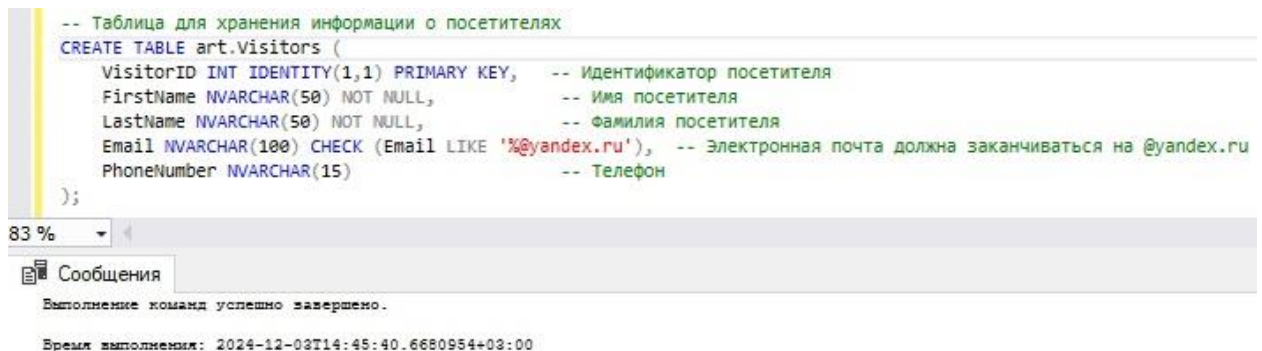


Рисунок 8 – Создание таблицы для хранения информации о посетителях

Создание таблицы VisitorTickets

```

CREATE TABLE art.VisitorTickets (
    VisitorID INT, -- Идентификатор посетителя
    TicketID INT, -- Идентификатор билета
    PurchaseDate DATETIME NOT NULL DEFAULT GETDATE(), -- Дата покупки билета
    PRIMARY KEY (VisitorID, TicketID), -- Составной первичный ключ
    FOREIGN KEY (VisitorID) REFERENCES art.Visitors(VisitorID), -- Внешний ключ на таблицу art.Visitors
);

```

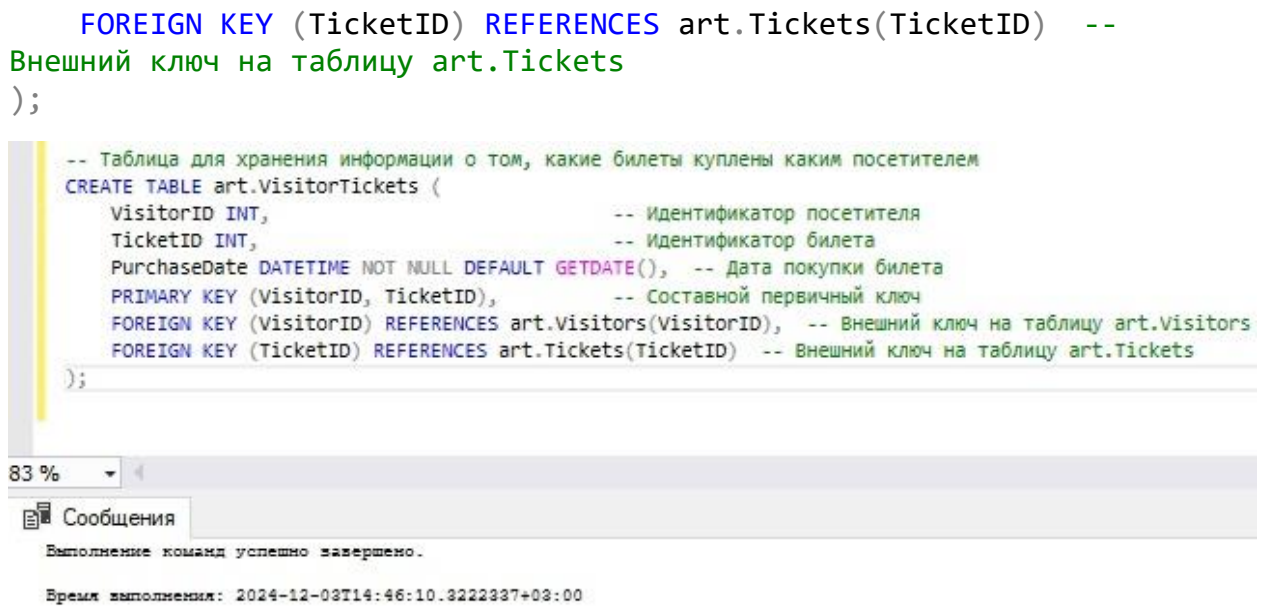


Рисунок 9 – Создание таблицы для хранения информации о том, какие билеты куплены
каким посетителем

5. Заполнение таблиц данными

Заполнение таблицы Artist

```

INSERT INTO art.Artists (FirstName, LastName, BirthDate, DeathDate,
Nationality)
VALUES
('Vincent', 'van Gogh', '1853-03-30', '1890-07-29', 'Dutch'),
('Leonardo', 'da Vinci', '1452-04-15', '1519-05-02', 'Italian'),
('Pablo', 'Picasso', '1881-10-25', '1973-04-08', 'Spanish');

```

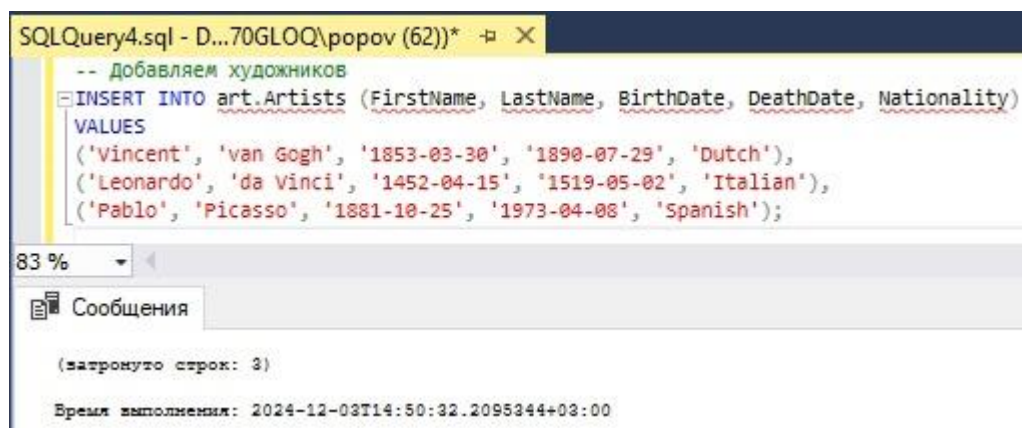


Рисунок 10 – Заполнение таблицы для хранения информации о художниках

Заполнение таблицы Artworks

```

INSERT INTO art.Artworks (Title, YearCreated, ArtistID, ArtType,
Description)

```


VALUES

```
('Starry Night', 1889, 1, 'Painting', 'Famous painting by Vincent van Gogh'),  
( 'Mona Lisa', 1503, 2, 'Painting', 'Portrait painting by Leonardo da Vinci'),  
( 'Guernica', 1937, 3, 'Painting', 'Famous painting by Pablo Picasso');
```

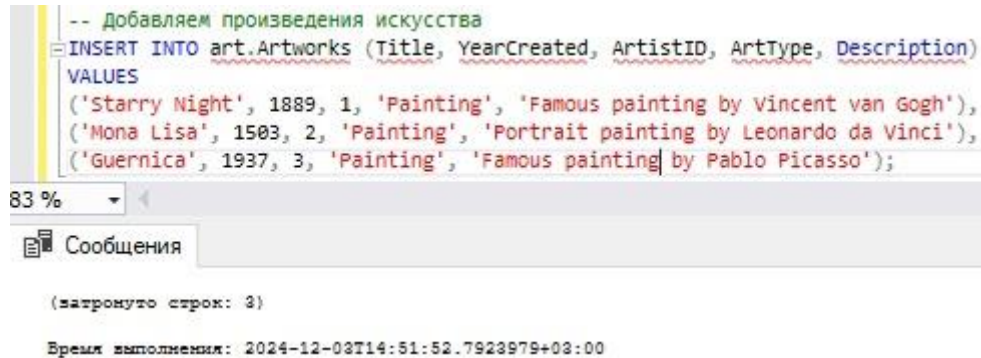


Рисунок 11 – Заполнение таблицы для хранения информации о произведениях искусства

Заполнение таблицы Exhibitions

```
INSERT INTO art.Exhibitions (ExhibitionName, StartDate, EndDate,  
Location)
```

VALUES

```
('Impressionists of the 19th Century', '2024-01-01', '2024-03-31',  
'The Louvre, Paris'),  
( 'Renaissance Art Collection', '2024-05-01', '2024-07-31', 'Uffizi Gallery, Florence'),  
( 'Modern Art Revolution', '2024-06-01', '2024-08-31', 'Museum of Modern Art, New York');
```

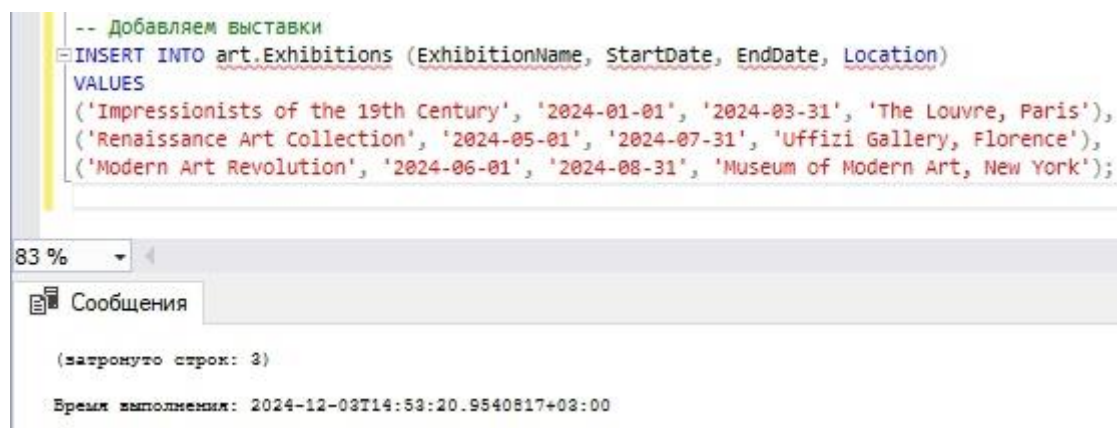


Рисунок 12 – Заполнение таблицы для хранения информации о выставках

Заполнение таблицы ArtworkExhibitions

```
INSERT INTO art.ArtworkExhibitions (ArtworkID, ExhibitionID)
```

VALUES

```
(1, 1), -- Starry Night на выставке "Impressionists of the 19th  
Century"
```

```
(2, 2), -- Mona Lisa на выставке "Renaissance Art Collection"
(3, 3); -- Guernica на выставке "Modern Art Revolution"
```

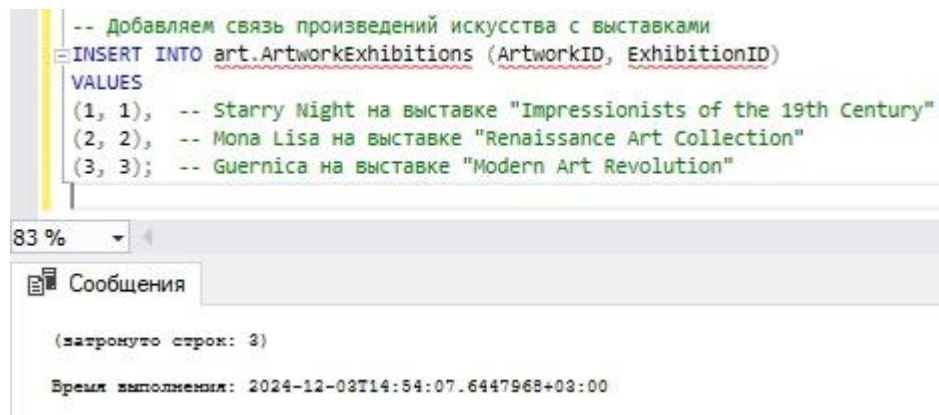


Рисунок 13 – Заполнение таблицы для хранения информации о связи произведений искусства с выставками

Заполнение таблицы Galleries

```
INSERT INTO art.Galleries (GalleryName, Location, ContactPhone)
VALUES
('The Louvre', 'Paris, France', '33123456789'),
('Uffizi Gallery', 'Florence, Italy', '390123456789'),
('Museum of Modern Art', 'New York, USA', '12125551234');
```



Рисунок 14 – Заполнение таблицы для хранения информации о галереях

Заполнение таблицы Tickets

```
INSERT INTO art.Tickets (TicketType, Price, GalleryID, ExhibitionID)
VALUES
('Adult', 20.00, 4, 1), -- Билет для взрослого для выставки
"Impressionists of the 19th Century"
('Child', 10.00, 4, 1), -- Детский билет для выставки "Impressionists
of the 19th Century"
('Student', 15.00, 5, 2), -- Студенческий билет для выставки
"Renaissance Art Collection"
```

```
('Adult', 25.00, 6, 3); -- Взрослый билет для выставки "Modern Art Revolution"
```

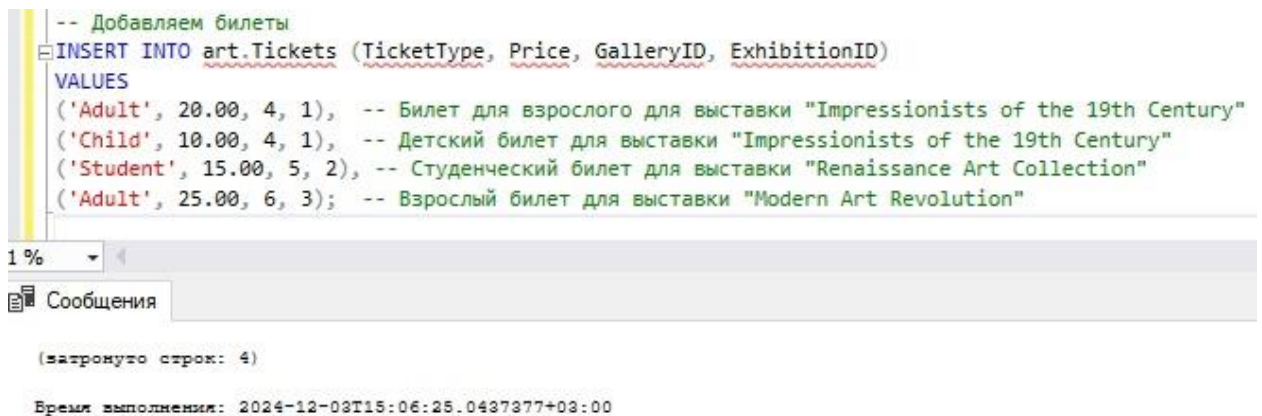


Рисунок 15 – Заполнение таблицы для хранения информации о билетах

Заполнение таблицы Visitors

```
INSERT INTO art.Visitors (FirstName, LastName, Email, PhoneNumber)
VALUES
('Alice', 'Johnson', 'alice.johnson@yandex.ru', '+1234567890'),
('Bob', 'Smith', 'bob.smith@yandex.ru', '+1987654321'),
('Charlie', 'Brown', 'charlie.brown@yandex.ru', '+1122334455');
```



Рисунок 16 – Заполнение таблицы для хранения информации о посетителях

Заполнение таблицы VisitorTickets

```
INSERT INTO art.VisitorTickets (VisitorID, TicketID, PurchaseDate)
VALUES
(1, 3, '2024-01-01 10:00:00'), -- Alice купила билет Adult для
выставки "Impressionists of the 19th Century"
(2, 4, '2024-02-01 15:00:00'), -- Bob купил билет Child для выставки
"Impressionists of the 19th Century"
(3, 5, '2024-05-15 12:30:00'), -- Charlie купил билет Student для
выставки "Renaissance Art Collection"
```



```
(1, 6, '2024-06-10 11:00:00'); -- Alice купила билет Adult для выставки "Modern Art Revolution"
```

```
-- Добавляем информацию о покупке билетов
INSERT INTO art.VisitorTickets (VisitorID, TicketID, PurchaseDate)
VALUES
(1, 3, '2024-01-01 10:00:00'), -- Alice купила билет Adult для выставки "Impressionists of the 19th Century"
(2, 4, '2024-02-01 15:00:00'), -- Bob купил билет Child для выставки "Impressionists of the 19th Century"
(3, 5, '2024-05-15 12:30:00'), -- Charlie купил билет Student для выставки "Renaissance Art Collection"
(1, 6, '2024-06-10 11:00:00'); -- Alice купила билет Adult для выставки "Modern Art Revolution"
```

91 %

Сообщения

(загружено строк: 4)

Время выполнения: 2024-12-03T15:09:03.3842163+03:00

Рисунок 17 – Заполнение таблицы для хранения информации о том, какие билеты куплены каким посетителем

6. Разработка объектов промежуточного слоя

Создание представления ArtworksWithArtists

```
CREATE VIEW art.ArtworksWithArtists AS
SELECT
    a.ArtworkID,
    a.Title,
    a.YearCreated,
    a.ArtType,
    a.Description,
    CONCAT(artist.FirstName, ' ', artist.LastName) AS ArtistName,
    ex.ExhibitionName,
    ex.StartDate,
    ex.EndDate
FROM
    art.Artworks a
JOIN
    art.Artists artist ON a.ArtistID = artist.ArtistID
JOIN
    art.ArtworkExhibitions ae ON a.ArtworkID = ae.ArtworkID
JOIN
    art.Exhibitions ex ON ae.ExhibitionID = ex.ExhibitionID;
```

```

SQLQuery5.sql - D:\70GLOQ\popov (70))*
CREATE VIEW art.ArtworksWithArtists AS
SELECT
    a.ArtworkID,
    a.Title,
    a.YearCreated,
    a.ArtType,
    a.Description,
    CONCAT(artist.FirstName, ' ', artist.LastName) AS ArtistName,
    ex.ExhibitionName,
    ex.StartDate,
    ex.EndDate
FROM
    art.Artworks a
JOIN
    art.Artists artist ON a.ArtistID = artist.ArtistID
JOIN
    art.ArtworkExhibitions ae ON a.ArtworkID = ae.ArtworkID
JOIN
    art.Exhibitions ex ON ae.ExhibitionID = ex.ExhibitionID;
91 %
Сообщения
Выполнение команд успешно завершено.
Время выполнения: 2024-12-03T15:12:51.7280280+03:00

```

Рисунок 18 – Выполнение запроса создания представления

	ArtworkID	Title	YearCreated	ArtType	Description	ArtistName	ExhibitionName	StartDate	EndDate
1	1	Stary Night	1889	Painting	Famous painting by Vincent van Gogh	Vincent van Gogh	Impressionists of the 19th Century	2024-01-01	2024-03-31
2	2	Mona Lisa	1503	Painting	Portrait painting by Leonardo da Vinci	Leonardo da Vinci	Renaissance Art Collection	2024-05-01	2024-07-31
3	3	Guernica	1937	Painting	Famous painting by Pablo Picasso	Pablo Picasso	Modern Art Revolution	2024-06-01	2024-08-31

Рисунок 19 – Проверка работоспособности созданного представления

Создание представления GalleriesWithExhibitions

```

CREATE VIEW art.GalleriesWithExhibitions AS
SELECT
    g.GalleryID,
    g.GalleryName,
    g.Location AS GalleryLocation,
    ex.ExhibitionID,
    ex.ExhibitionName,
    ex.StartDate,
    ex.EndDate,
    ex.Location AS ExhibitionLocation,
    t.TicketID,
    t.TicketType,
    t.Price

```

```
FROM
    art.Galleries g
JOIN
    art.Tickets t ON g.GalleryID = t.GalleryID
JOIN
    art.Exhibitions ex ON t.ExhibitionID = ex.ExhibitionID
```

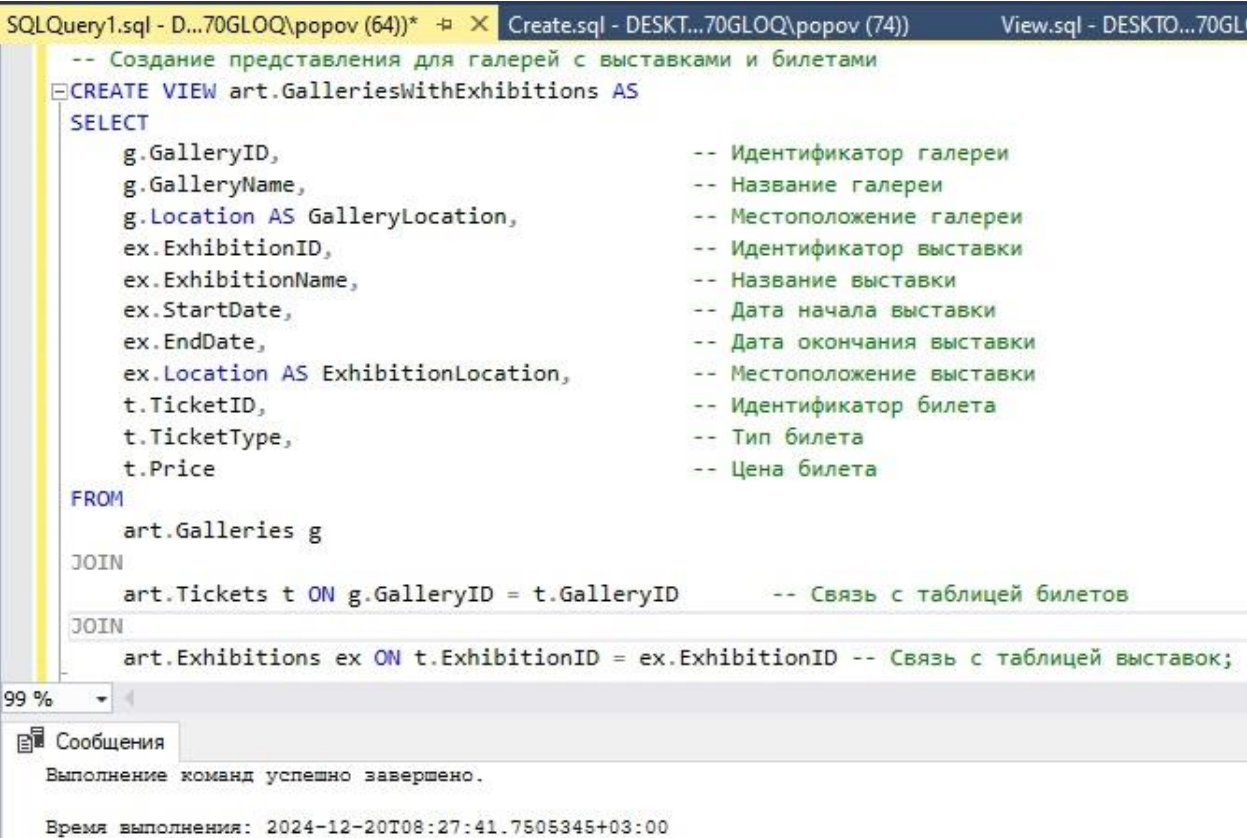


Рисунок 20 – Выполнение запроса создания представления

SELECT * FROM art.GalleriesWithExhibitions;

	GalleryID	GalleryName	GalleryLocation	ExhibitionID	ExhibitionName	StartDate	EndDate	ExhibitionLocation	TicketID	TicketType	Price
1	4	The Louvre	Paris, France	1	Impressionists of the 19th Century	2024-01-01	2024-03-31	The Louvre, Paris	3	Adult	20.00
2	4	The Louvre	Paris, France	1	Impressionists of the 19th Century	2024-01-01	2024-03-31	The Louvre, Paris	4	Child	10.00
3	5	Uffizi Gallery	Florence, Italy	2	Renaissance Art Collection	2024-05-01	2024-07-31	Uffizi Gallery, Florence	5	Student	15.00
4	6	Museum of Modern Art	New York, USA	3	Modern Art Revolution	2024-06-01	2024-08-31	Museum of Modern Art, New York	6	Adult	25.00

Рисунок 21 – Проверка работоспособности созданного представления

Создание представления VisitorsWithTickets

```
CREATE VIEW art.VisitorsWithTickets AS
SELECT
    v.VisitorID,
```

```

CONCAT(v.FirstName, ' ', v.LastName) AS VisitorName,
v.Email,
v.PhoneNumber,
t.TicketType,
t.Price,
ex.ExhibitionName,
ex.StartDate AS ExhibitionStartDate,
ex.EndDate AS ExhibitionEndDate,
g.GalleryName AS GalleryName,
g.Location AS GalleryLocation
FROM
art.Visitors v
JOIN
art.VisitorTickets vt ON v.VisitorID = vt.VisitorID
JOIN
art.Tickets t ON vt.TicketID = t.TicketID
JOIN
art.Exhibitions ex ON t.ExhibitionID = ex.ExhibitionID
JOIN
art.Galleries g ON t.GalleryID = g.GalleryID;

```

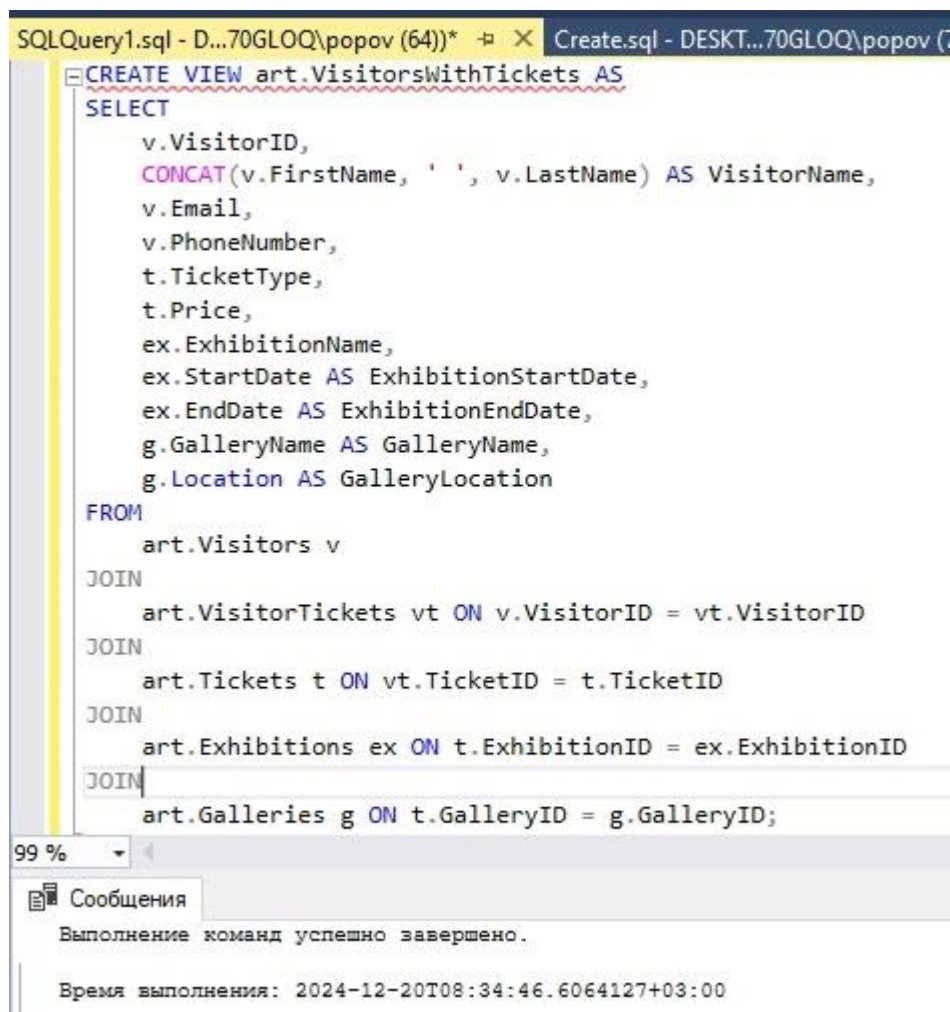


Рисунок 22 – Выполнение запроса создания представления

SELECT * FROM art.VisitorsWithTickets;											
99 %											
Результаты											
	VisitorID	VisitorName	Email	PhoneNumber	TicketType	Price	ExhibitionName	ExhibitionStartDate	ExhibitionEndDate	GalleryName	GalleryLocation
1	1	Alice Johnson	alice.johnson@yandex.ru	+1234567890	Adult	20.00	Impressionists of the 19th Century	2024-01-01	2024-03-31	The Louvre	Paris, France
2	1	Alice Johnson	alice.johnson@yandex.ru	+1234567890	Adult	25.00	Modern Art Revolution	2024-06-01	2024-08-31	Museum of Modern Art	New York, USA
3	2	Bob Smith	bob.smith@yandex.ru	+1987654321	Child	10.00	Impressionists of the 19th Century	2024-01-01	2024-03-31	The Louvre	Paris, France
4	3	Charlie Brown	charlie.brown@yandex.ru	+1122334455	Student	15.00	Renaissance Art Collection	2024-05-01	2024-07-31	Uffizi Gallery	Florence, Italy

Рисунок 23 – Проверка работоспособности созданного представления

Создание функции CountArtworksByArtist

```
CREATE FUNCTION art.CountArtworksByArtist (@ArtistID INT)
RETURNS INT
AS
BEGIN
    DECLARE @Count INT;

    SELECT @Count = COUNT(*)
    FROM art.Artworks
    WHERE ArtistID = @ArtistID;

    RETURN @Count as Result;
END;
```

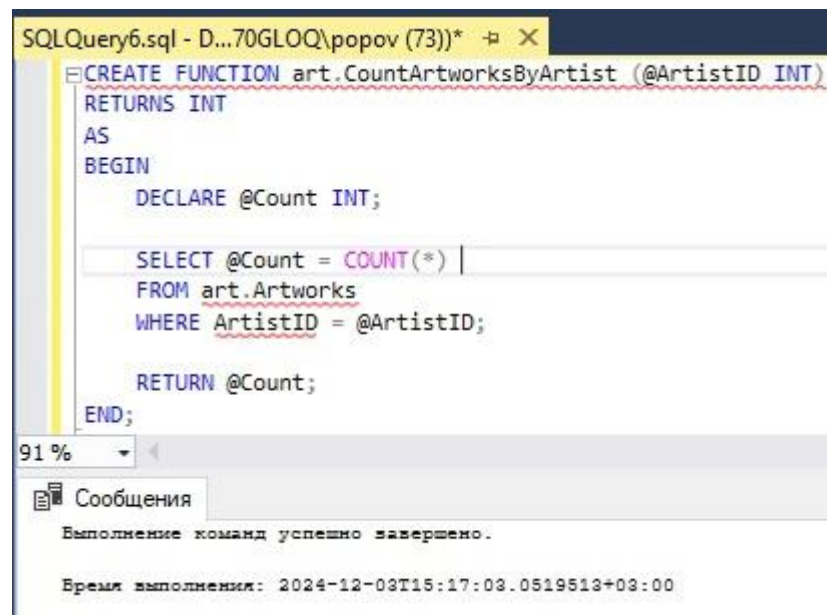


Рисунок 24 – Выполнение запроса создания функции

SELECT art.CountArtworksByArtist(1); -- Возвращает количество произведений искусства для художника с ID = 1											
91 %											
Результаты											
	(Отсутствует имя столбца)										
1	1										

Рисунок 25 – Проверка работоспособности созданной функции

Создание функции CountExhibitionsByArtist

```
CREATE FUNCTION art.CountExhibitionsByArtist (@ArtistID INT)
RETURNS INT
AS
BEGIN
    DECLARE @Count INT;

    SELECT @Count = COUNT(DISTINCT ex.ExhibitionID)
    FROM art.Artworks a
    JOIN art.ArtworkExhibitions ae ON a.ArtworkID = ae.ArtworkID
    JOIN art.Exhibitions ex ON ae.ExhibitionID = ex.ExhibitionID
    WHERE a.ArtistID = @ArtistID;

    RETURN @Count;
END;
```

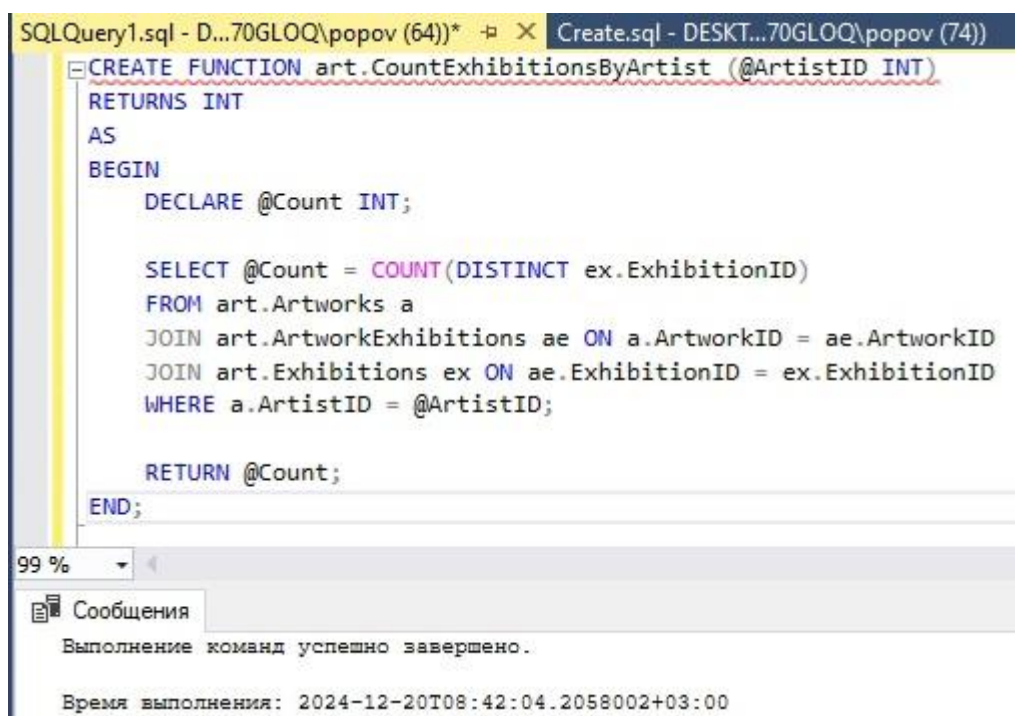


Рисунок 26 – Выполнение запроса создания функции

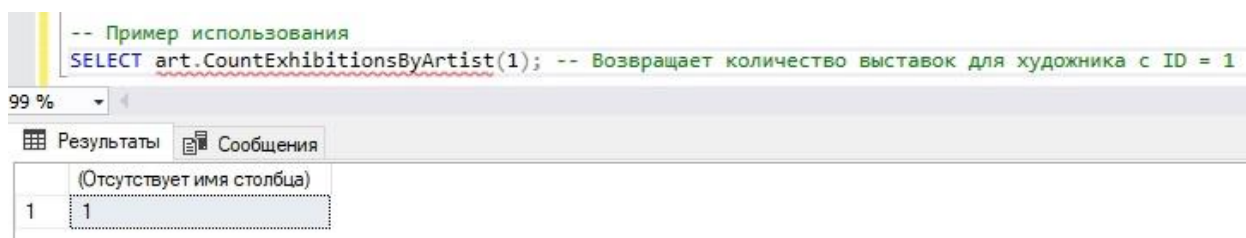


Рисунок 27 – Проверка работоспособности созданной функции

Создание функции CountVisitorsByExhibition

```

CREATE FUNCTION art.CountVisitorsByExhibition (@ExhibitionID INT)
RETURNS INT
AS
BEGIN
    DECLARE @Count INT;

    SELECT @Count = COUNT(DISTINCT vt.VisitorID)
    FROM art.VisitorTickets vt
    JOIN art.Tickets t ON vt.TicketID = t.TicketID
    WHERE t.ExhibitionID = @ExhibitionID;

    RETURN @Count;
END;

```

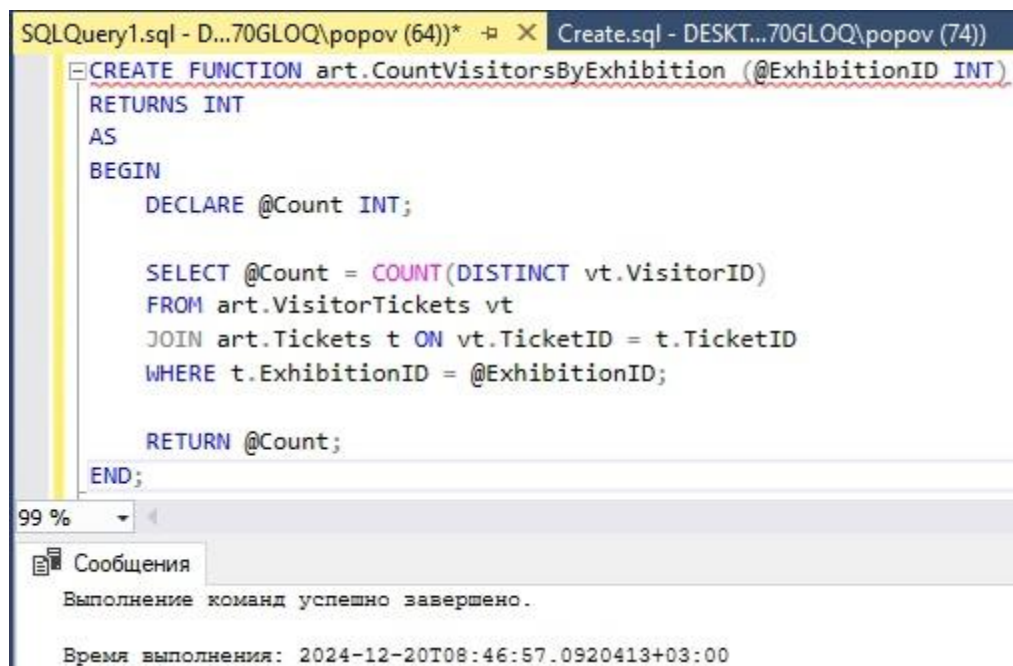


Рисунок 28 – Выполнение запроса создания функции

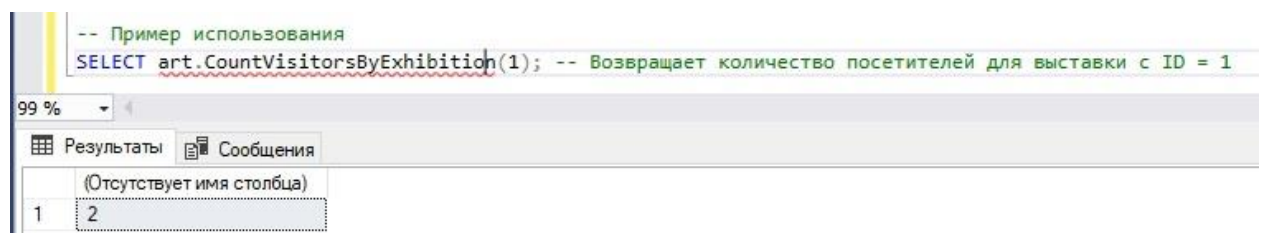


Рисунок 29 – Проверка работоспособности созданной функции

Создание процедуры AddArtwork

```

CREATE PROCEDURE art.AddArtwork
    @Title NVARCHAR(100),
    @YearCreated INT,
    @ArtistID INT,
    @ArtType NVARCHAR(50),

```

```

        @Description NVARCHAR(500)
AS
BEGIN
    -- Проверяем, существует ли уже такое произведение
    IF EXISTS (SELECT 1 FROM art.Artworks WHERE Title = @Title AND
ArtistID = @ArtistID)
    BEGIN
        RAISERROR('Произведение искусства с таким названием и
художником уже существует.', 16, 1);
    END
    ELSE
    BEGIN
        -- Добавляем новое произведение искусства
        INSERT INTO art.Artworks (Title, YearCreated, ArtistID,
ArtType, Description)
        VALUES (@Title, @YearCreated, @ArtistID, @ArtType,
@Description);
    END
END;

```

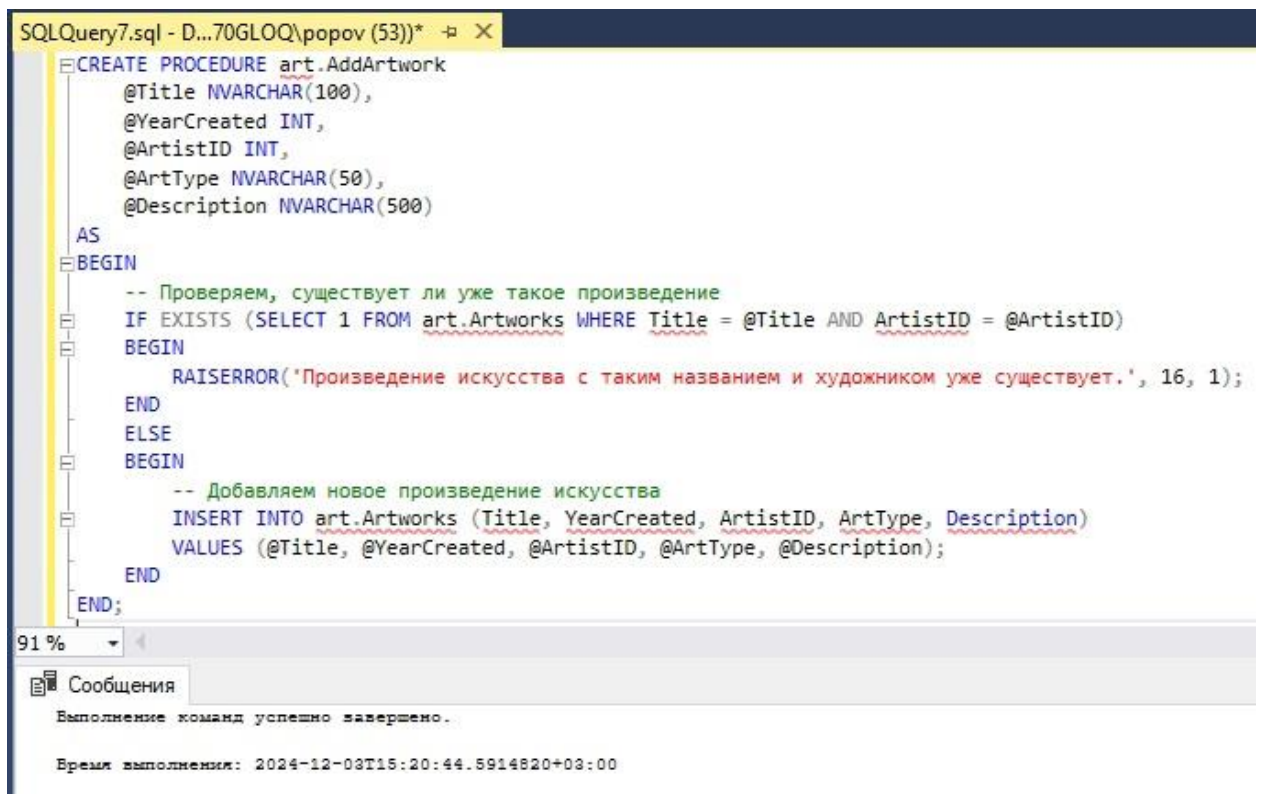
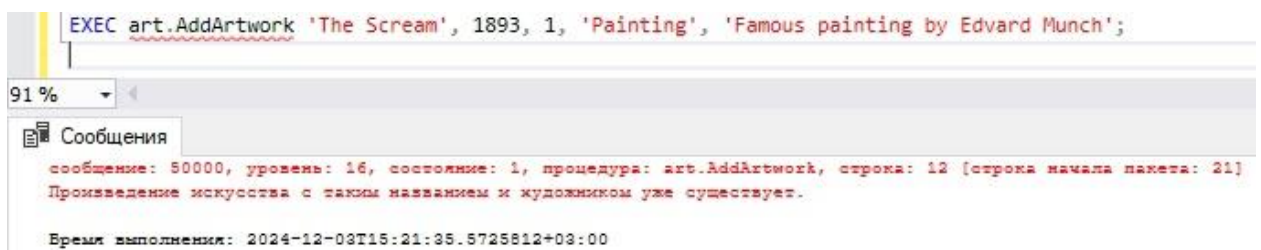


Рисунок 30 – Выполнение запроса создания процедуры



7. Разработка стратегии резервного копирования

```
BACKUP DATABASE art  
TO DISK = 'D:\Backups\art_FullBackup.bak'  
WITH FORMAT, INIT;
```



Рисунок 32 – Выполнение запроса создания резервной копии

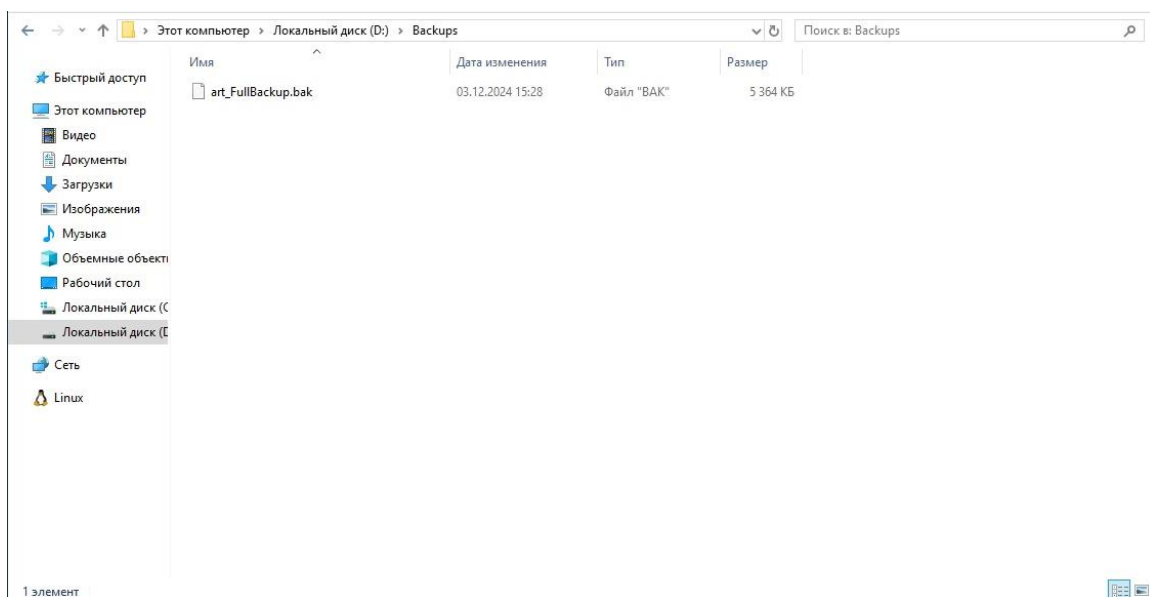


Рисунок 33 – Проверка работоспособности резервного копирования

8. Описание структуры таблиц БД

Таблица: Художники (Artists)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 1

Имя таблицы: Artists

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор художника	ArtistID	INT	П	[ArtistID] > 0, PRIMARY KEY, AUTO_INCREMENT
2	Имя художника	FirstName	NVARCHAR(50)		NOT NULL
3	Фамилия художника	LastName	NVARCHAR(50)		NOT NULL
4	Дата рождения	BirthDate	DATE		
5	Дата смерти	DeathDate	DATE		
6	Национальность	Nationality	NVARCHAR(50)		

Таблица: Произведения искусства (Artworks)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 2

Имя таблицы: Artworks

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор произведения искусства	ArtworkID	INT	П	[ArtworkID] > 0, PRIMARY KEY, AUTO_INCREMENT
2	Название произведения	Title	NVARCHAR(100)		NOT NULL
3	Год создания	YearCreated	INT		

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
4	Идентификатор художника	ArtistID	INT	В	FOREIGN KEY (ArtistID) REFERENCES art.Artists(ArtistID)
5	Тип искусства	ArtType	NVARCHAR(50)		
6	Описание произведения	Description	NVARCHAR(500)		

Таблица: Выставки (Exhibitions)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 3

Имя таблицы: Exhibitions

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор выставки	ExhibitionID	INT	П	[ExhibitionID] > 0, PRIMARY KEY, AUTO_INCREMENT
2	Название выставки	ExhibitionName	NVARCHAR(100)		NOT NULL
3	Дата начала	StartDate	DATE		
4	Дата окончания	EndDate	DATE		CONSTRAINT CK_EndDateAfterStartDate CHECK (EndDate > StartDate)
5	Место проведения	Location	NVARCHAR(100)		NOT NULL

Таблица: Связь произведений искусства с выставками (ArtworkExhibitions)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 4

Имя таблицы: ArtworkExhibitions

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор произведения искусства	ArtworkID	INT	П	FOREIGN KEY (ArtworkID) REFERENCES art.Artworks(ArtworkID) ON DELETE CASCADE

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
2	Идентификатор выставки	ExhibitionID	INT	П	FOREIGN KEY (ExhibitionID) REFERENCES art.Exhibitions(ExhibitionID) ON DELETE CASCADE
3	Совпадение ключа	PRIMARY KEY (ArtworkID, ExhibitionID)	Составной	П	

Таблица: Галереи (Galleries)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 5

Имя таблицы: Galleries

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор галереи	GalleryID	INT	П	[GalleryID] > 0, PRIMARY KEY, AUTO_INCREMENT
2	Название галереи	GalleryName	NVARCHAR(100)		NOT NULL
3	Местоположение	Location	NVARCHAR(100)		
4	Контактный телефон	ContactPhone	NVARCHAR(15)		CHECK (ContactPhone NOT LIKE '%[^0-9]%')

Таблица: Билеты (Tickets)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 6

Имя таблицы: Tickets

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор билета	TicketID	INT	П	[TicketID] > 0, PRIMARY KEY, AUTO_INCREMENT
2	Тип билета	TicketType	NVARCHAR(50)		NOT NULL
3	Цена билета	Price	DECIMAL(10, 2)		CHECK (Price > 0) NOT NULL

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
4	Идентификатор галереи	GalleryID	INT	Внешний	FOREIGN KEY (GalleryID) REFERENCES art.Galleries(GalleryID)
5	Идентификатор выставки	ExhibitionID	INT	Внешний	FOREIGN KEY (ExhibitionID) REFERENCES art.Exhibitions(ExhibitionID)

Таблица: Посетители (Visitors)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 7

Имя таблицы: Visitors

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор посетителя	VisitorID	INT	П	[VisitorID] > 0, PRIMARY KEY, AUTO_INCREMENT
2	Имя посетителя	FirstName	NVARCHAR(50)		NOT NULL
3	Фамилия посетителя	LastName	NVARCHAR(50)		NOT NULL
4	Электронная почта	Email	NVARCHAR(100)		CHECK (Email LIKE '%@yandex.ru')
5	Номер телефона	PhoneNumber	NVARCHAR(15)		

Таблица: Покупка билетов посетителями (VisitorTickets)

Дата разработки: 03.12.2024

Порядковый номер таблицы: 8

Имя таблицы: VisitorTickets

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Идентификатор посетителя	VisitorID	INT	Внешний	FOREIGN KEY (VisitorID) REFERENCES art.Visitors(VisitorID)
2	Идентификатор билета	TicketID	INT	Внешний	FOREIGN KEY (TicketID) REFERENCES art.Tickets(TicketID)

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
3	Дата покупки	PurchaseDate	DATETIME		NOT NULL, DEFAULT GETDATE()
4	Совпадение ключа	PRIMARY KEY (VisitorID, TicketID)	Составной	PRIMARY KEY	

9. Заключение

В результате выполнения задания была успешно разработана база данных для управления данными художественной галереи. Были созданы таблицы, обеспечивающие хранение информации о художниках, произведениях искусства, выставках, галереях, билетах и посетителях.

Кроме того, были реализованы объекты промежуточного слоя: функции, представления и хранимые процедуры для упрощения работы с данными и улучшения производительности. Разработана стратегия резервного копирования, включающая полные копии для защиты данных от потерь.

Эти шаги обеспечивают высокую степень удобства работы с базой данных, упрощают задачи по получению аналитической информации, а также гарантируют безопасность данных благодаря продуманной стратегии резервного копирования.

10. Список использованных источников

1.Начало работы с запросами с помощью Transact-SQL. – URL: <https://learn.microsoft.com/ru-ru/training/paths/get-started-querying-with-transact-sql/> (дата обращения 13.12.2022).

2.Основные команды SQL, которые должен знать каждый программист [Электронный ресурс] Режим доступа: <https://tproger.ru/translations/sql-recap/> (дата обращения 14.12.2022)

3. Просмотр или изменение модели восстановления базы данных (SQL Server) [Электронный ресурс] Режим доступа: <https://learn.microsoft.com/ru-ru/sql/relational-databases/backup-restore/view-or-change-the-recovery-model-of-a-database-sql-server?view=sql-server-ver16> (дата обращения 15.12.2022)