Лабораторная работа № 12. Создание БД заданной предметной области

Задание:

1. Выполнить нормализацию даталогической модели предметной области вашего варианта индивидального задания до 3НФ.

Какой результат я имею на данный момент

Table Movie {

id int [pk, increment]

description varchar(255)

rating int

age\_limit int

number\_of\_views int

duration int

release\_year int

country\_id int

website\_id int

publisher\_producer\_id int

director\_id int

genre\_id int

}

ref: Publisher\_Producer.id <> Movie.publisher\_producer\_id

ref : Director.id <> Movie.director\_id

ref : Website.id - Movie.website\_id

ref : Genre.id <> Movie.genre\_id

table Genre {

id int [pk, increment]

name varchar(255)

}

table Publisher\_Producer {

id int [pk, increment]

publisher\_id int

address\_id int

name varchar(255)

}

ref : Publisher\_Producer.publisher\_id > Publisher.id

ref : Address.id - Publisher\_Producer.address\_id

table Publisher {

id int [pk, increment]

name varchar(255)

website\_id int

}

ref : Publisher.website\_id - Website.id

table Director {

id int [pk, increment]

country\_id int UNIQUE

name varchar(255)

}

table Country {

id int [pk, increment]

name varchar(255)

}

ref : Director.country\_id > Country.id

ref : Movie.country\_id > Country.id

table Collection {

id int [pk, increment]

name varchar(255)

description text

}

table MovieCollection {

collection\_id int

movie\_id int

}

ref : Collection.id < MovieCollection.collection\_id

ref : Movie.id < MovieCollection.movie\_id

table Website {

id int [pk, increment]

name varchar(255)

url text

}

table Address {

id int [pk, increment]

country\_id int

street varchar(255)

house int

placement int

}

ref : Country.id < Address.country\_id

table Rentaler {

id int [pk, increment]

name varchar(255)

website\_id int

}

ref : Website.id - Rentaler.website\_id

table Rentaler\_Concrete {

id int [pk, increment]

rentaler\_id int

address\_id int

phone\_number varchar(255)

}

ref : Rentaler\_Concrete.rentaler\_id > Rentaler.id

ref : Rentaler\_Concrete.address\_id > Address.id

table User {

id int [pk, increment]

age int

name varchar(255)

}

table UserFavoriteMovies {

movie\_id int

user\_id int

}

ref : User.id > UserFavoriteMovies.user\_id

ref : Movie.id > UserFavoriteMovies.movie\_id

table UserFavoriteGenre {

genre\_id int

user\_id int

}

ref : User.id > UserFavoriteGenre.user\_id

ref : Genre.id > UserFavoriteGenre.genre\_id

table Carrier { // носитель

id int [pk, increment]

size int

name varchar(255)

type\_id int

}

ref : Carrier.type\_id > Type.id

table Type {

id int [pk, increment]

name varchar(255)

}

table Recording\_Format {

id int [pk, increment]

name varchar(255)

}

table RentedCarriers {

id int [pk, increment]

start\_rent time

end\_rent time

carrier\_id int

rentaler\_concrete\_id int

}

ref : Carrier.id - RentedCarriers.carrier\_id

ref : Rentaler\_Concrete.id < RentedCarriers.rentaler\_concrete\_id

table MovieCarrier {

movie\_id int

carrier\_id int

recording\_format\_id int

}

// один фильм может быть как копия на разных носителях

// как и один носитель может хранить много фильмов

ref : Carrier.id < MovieCarrier.carrier\_id

ref : Movie.id < MovieCarrier.movie\_id

ref : Recording\_Format.id < MovieCarrier.recording\_format\_id

проверим на НФ-ы

1НФ)

Она требует

1. Атомарность данных – все значения в столбцах должны быть неделимыми (в моем результате она выполняется — любое значение в моих столбцах является неделимым)

2. Уникальность строк – каждая строка должна иметь уникальный идентификатор (первичный ключ). (да — во всех таблицах он есть (у кого-то это ID, у кого-то он составной))

3. Отсутствие повторяющихся групп – не должно быть столбцов с одинаковыми данными. (таких у меня не наблюдается)

2НФ)

Она требует

1. Должна выполняться 1НФ (как показано выше — она выполняется).

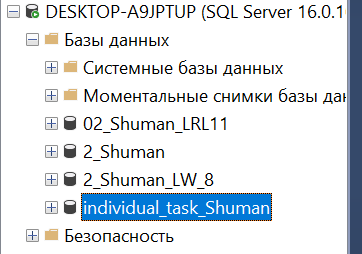
2. Отсутствие частичной зависимости – все неключевые атрибуты должны зависеть от всего первичного ключа, а не от его части (актуально для составных ключей).

2. Построить структуру БД для предметной области вашего варианта индивидального задания в выбранной СУБД (создать таблицы, ключи, ограничения в соответствии с разработанной в ЛР-6 даталогической моделью БД). Для этого использовать любой из доступных вариантов:

1. создать структуру БД с использованием экспорта из используемого в ЛР-3 CASE-средства для разработки моделей БД.

2. создать структуру БД с использованием интерфейсных элементов клиентского приложения для работы с выбранной СУБД (например, MS SQL Management Studio. 3. создать структуру БД с использованием операторов DDL и клиентского приложения для работы с выбранной СУБД.

Создал новую БД



Вставляем и запускаем сгенерированные запросы

-- Задание 2

-- сгенерированно для DBML на сайте https://dbdiagram.io/d

CREATE TABLE [Movie] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [description] varchar(255),

  [rating] int,

  [age\_limit] int,

  [number\_of\_views] int,

  [duration] int,

  [release\_year] int,

  [country\_id] int,

  [website\_id] int,

  [publisher\_id] int,

  [director\_id] int,

  [genre\_id] int

)

GO

CREATE TABLE [Genre] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255)

)

GO

CREATE TABLE [Publisher] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255),

  [website\_id] int

)

GO

CREATE TABLE [Director] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [country\_id] int,

  [name] varchar(255),

  [description] text

)

GO

CREATE TABLE [Country] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255)

)

GO

CREATE TABLE [Collection] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255),

  [description] text

)

GO

CREATE TABLE [MovieCollection] (

  [collection\_id] int,

  [movie\_id] int

)

GO

CREATE TABLE [Website] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255),

  [url] text

)

GO

CREATE TABLE [LocalityType] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(50),

  [code] varchar(20)

)

GO

CREATE TABLE [Locality] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [country\_id] int,

  [locality\_type\_id] int,

  [name] varchar(255)

)

GO

CREATE TABLE [Address] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [locality\_id] int,

  [street] varchar(255),

  [house] int

)

GO

CREATE TABLE [Rentaler] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255),

  [website\_id] int

)

GO

CREATE TABLE [Rentaler\_Concrete] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [rentaler\_id] int,

  [address\_id] int,

  [phone\_number] varchar(255)

)

GO

CREATE TABLE [User] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [age] int,

  [name] varchar(255)

)

GO

CREATE TABLE [UserFavoriteMovies] (

  [user\_id] int,

  [movie\_id] int

)

GO

CREATE TABLE [UserFavoriteGenre] (

  [genre\_id] int,

  [user\_id] int

)

GO

CREATE TABLE [Carrier] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [size] int,

  [name] varchar(255),

  [type\_id] int

)

GO

CREATE TABLE [Type] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255)

)

GO

CREATE TABLE [RentedCarriers] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [start\_rent] time,

  [end\_rent] time,

  [carrier\_id] int,

  [rentaler\_concrete\_id] int

)

GO

CREATE TABLE [Recording\_Format] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [name] varchar(255)

)

GO

CREATE TABLE [MovieCarrier] (

  [movie\_id] int,

  [carrier\_id] int,

  [recording\_format\_id] int

)

GO

CREATE TABLE [WatchHistory] (

  [id] int PRIMARY KEY IDENTITY(1, 1),

  [movie\_id] int,

  [user\_id] int,

  [watched\_at] datetime DEFAULT (GETDATE())

)

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица сущности Фильм.

Хранит информацию о фильмах:

- количество просмотров

- описание

- рейтинг

- страна производства

- и другие метаданные

',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Movie';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица сущности Жанр, хранит название Жанра',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Genre';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Сущность Издатель (Дисней, Пиксар и т.п.).

',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Publisher';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица сущности Режиссер:

хранит в себе имя, страну происхождения

и описание режиссера

',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Director';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица сущности Страна: хранит название страны',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Country';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица сущности Коллекция: хранит название и описание коллекции',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Collection';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица, описывающая содержимое коллекций',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'MovieCollection';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Таблица, описывающая Веб-сайт',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Website';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Хранит виды населенных мест (город, деревня, поселок и т.п.)',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'LocalityType';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Связывает Страну и населенный пункт (с указанием типа)',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Locality';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Хранит адрес объекта',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Address';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Компания, предоставляющая носители с фильмами в аренду.

Может иметь несколько точек по миру

',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Rentaler';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Конкретная точка аренды с адресом и контактами',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Rentaler\_Concrete';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Пользователь системы: имя и возраст',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'User';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Связь пользователя и понравившихся фильмов',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'UserFavoriteMovies';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Связь пользователя и предпочитаемых жанров',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'UserFavoriteGenre';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Физический носитель фильма (диск, кассета и т.д.)',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Carrier';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Типы носителей (DVD, Blu-ray, VHS и т.д.)',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Type';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'История аренды носителей',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'RentedCarriers';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Форматы записи (PAL, NTSC и т.д.)',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'Recording\_Format';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'Связь между фильмом и его носителями',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'MovieCarrier';

GO

EXEC sp\_addextendedproperty

@name = N'Table\_Description',

@value = 'хранит информацию о просмотре фильма

',

@level0type = N'Schema', @level0name = 'dbo',

@level1type = N'Table',  @level1name = 'WatchHistory';

GO

ALTER TABLE [Movie] ADD FOREIGN KEY ([country\_id]) REFERENCES [Country] ([id])

GO

ALTER TABLE [Movie] ADD FOREIGN KEY ([website\_id]) REFERENCES [Website] ([id])

GO

ALTER TABLE [Movie] ADD FOREIGN KEY ([publisher\_id]) REFERENCES [Publisher] ([id])

GO

ALTER TABLE [Movie] ADD FOREIGN KEY ([director\_id]) REFERENCES [Director] ([id])

GO

ALTER TABLE [Movie] ADD FOREIGN KEY ([genre\_id]) REFERENCES [Genre] ([id])

GO

ALTER TABLE [Publisher] ADD FOREIGN KEY ([website\_id]) REFERENCES [Website] ([id])

GO

ALTER TABLE [Director] ADD FOREIGN KEY ([country\_id]) REFERENCES [Country] ([id])

GO

ALTER TABLE [MovieCollection] ADD FOREIGN KEY ([collection\_id]) REFERENCES [Collection] ([id])

GO

ALTER TABLE [MovieCollection] ADD FOREIGN KEY ([movie\_id]) REFERENCES [Movie] ([id])

GO

ALTER TABLE [Locality] ADD FOREIGN KEY ([country\_id]) REFERENCES [Country] ([id])

GO

ALTER TABLE [Locality] ADD FOREIGN KEY ([locality\_type\_id]) REFERENCES [LocalityType] ([id])

GO

ALTER TABLE [Address] ADD FOREIGN KEY ([locality\_id]) REFERENCES [Locality] ([id])

GO

ALTER TABLE [Rentaler] ADD FOREIGN KEY ([website\_id]) REFERENCES [Website] ([id])

GO

ALTER TABLE [Rentaler\_Concrete] ADD FOREIGN KEY ([rentaler\_id]) REFERENCES [Rentaler] ([id])

GO

ALTER TABLE [Rentaler\_Concrete] ADD FOREIGN KEY ([address\_id]) REFERENCES [Address] ([id])

GO

ALTER TABLE [UserFavoriteMovies] ADD FOREIGN KEY ([user\_id]) REFERENCES [User] ([id])

GO

ALTER TABLE [UserFavoriteMovies] ADD FOREIGN KEY ([movie\_id]) REFERENCES [Movie] ([id])

GO

ALTER TABLE [UserFavoriteGenre] ADD FOREIGN KEY ([genre\_id]) REFERENCES [Genre] ([id])

GO

ALTER TABLE [UserFavoriteGenre] ADD FOREIGN KEY ([user\_id]) REFERENCES [User] ([id])

GO

ALTER TABLE [Carrier] ADD FOREIGN KEY ([type\_id]) REFERENCES [Type] ([id])

GO

ALTER TABLE [RentedCarriers] ADD FOREIGN KEY ([carrier\_id]) REFERENCES [Carrier] ([id])

GO

ALTER TABLE [RentedCarriers] ADD FOREIGN KEY ([rentaler\_concrete\_id]) REFERENCES [Rentaler\_Concrete] ([id])

GO

ALTER TABLE [MovieCarrier] ADD FOREIGN KEY ([movie\_id]) REFERENCES [Movie] ([id])

GO

ALTER TABLE [MovieCarrier] ADD FOREIGN KEY ([carrier\_id]) REFERENCES [Carrier] ([id])

GO

ALTER TABLE [MovieCarrier] ADD FOREIGN KEY ([recording\_format\_id]) REFERENCES [Recording\_Format] ([id])

GO

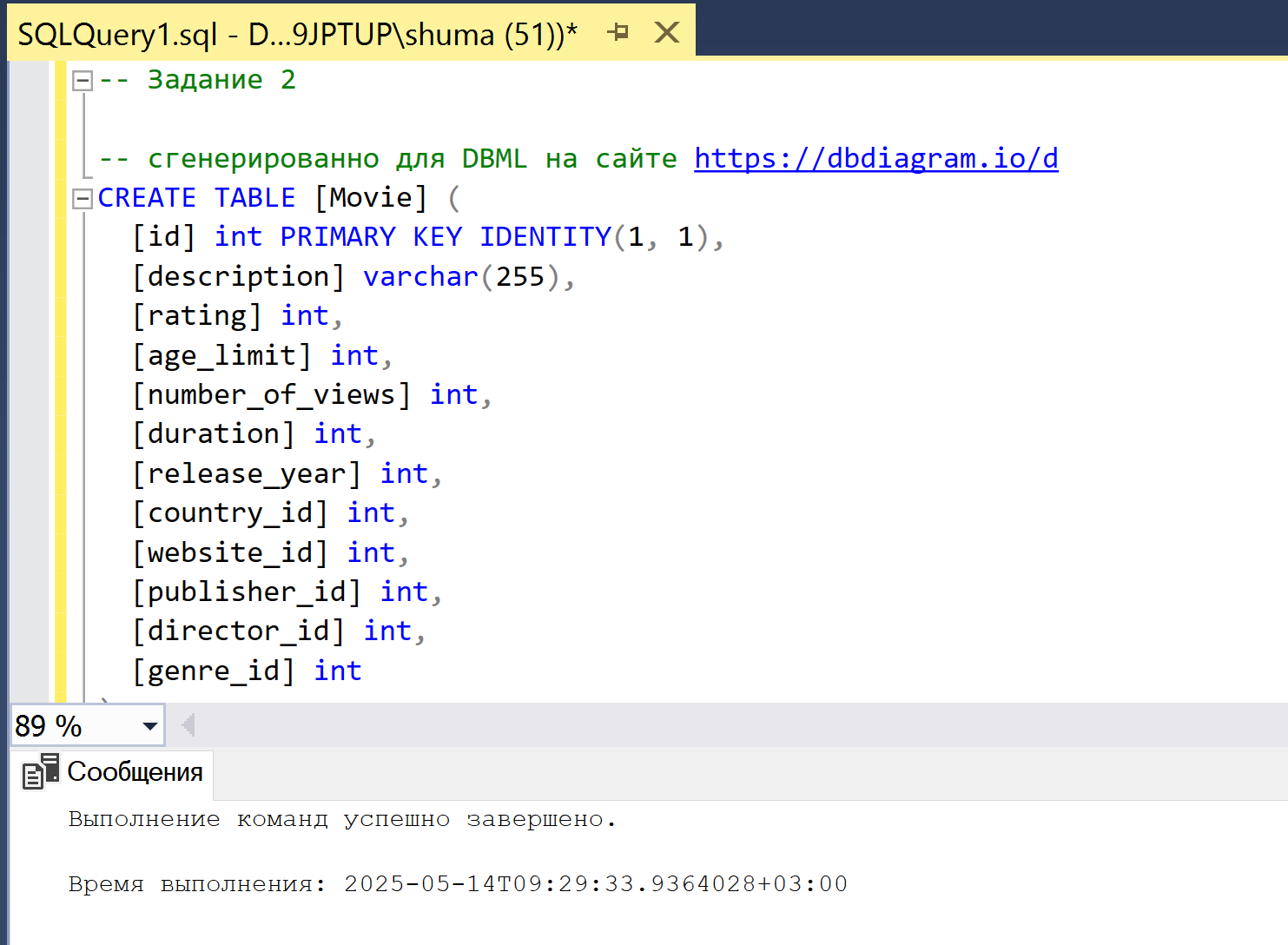
ALTER TABLE [WatchHistory] ADD FOREIGN KEY ([movie\_id]) REFERENCES [Movie] ([id])

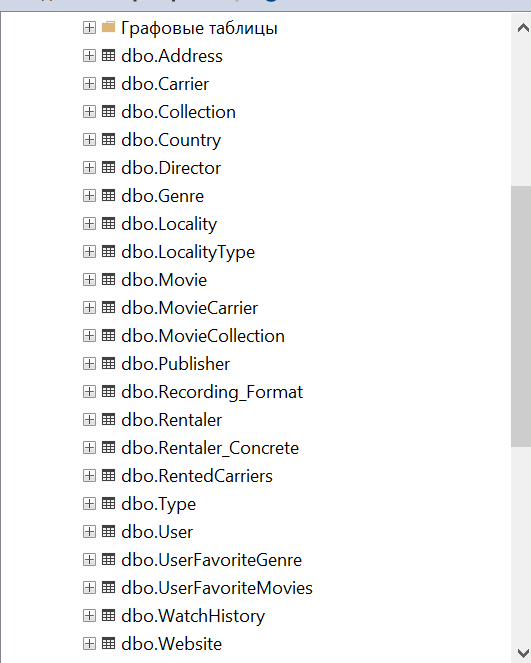
GO

ALTER TABLE [WatchHistory] ADD FOREIGN KEY ([user\_id]) REFERENCES [User] ([id])

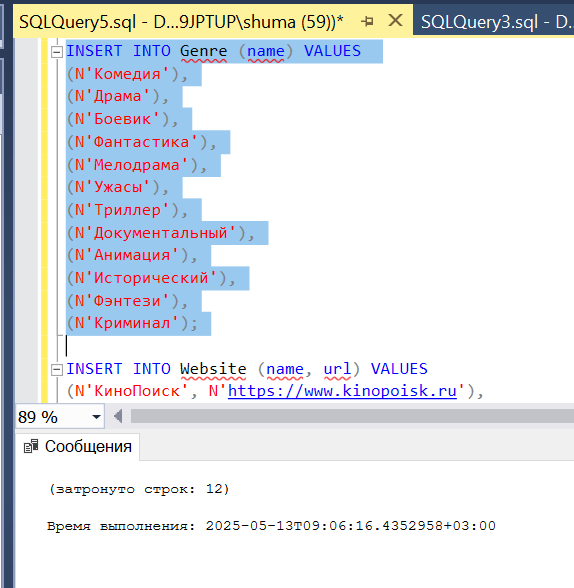
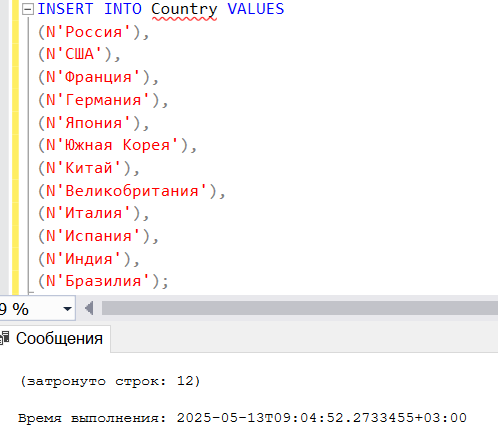
GO

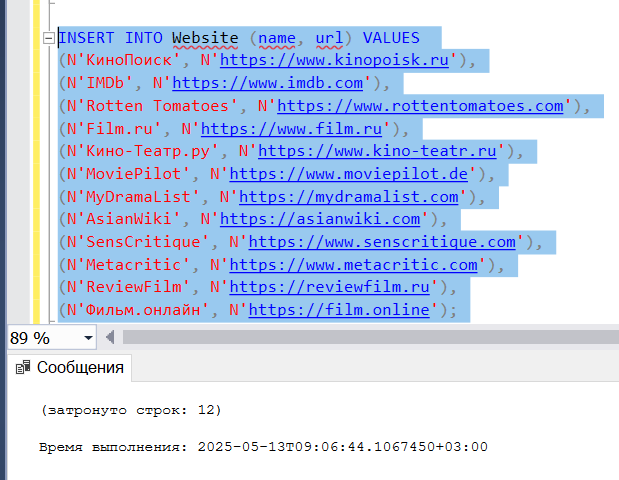
Замечание: инструмент(сайт), с помощью которого я генерировал запрос на построение таблиц и связей, вместо функции GETDATE() использовал функцию now(), которая выдавала ошибку

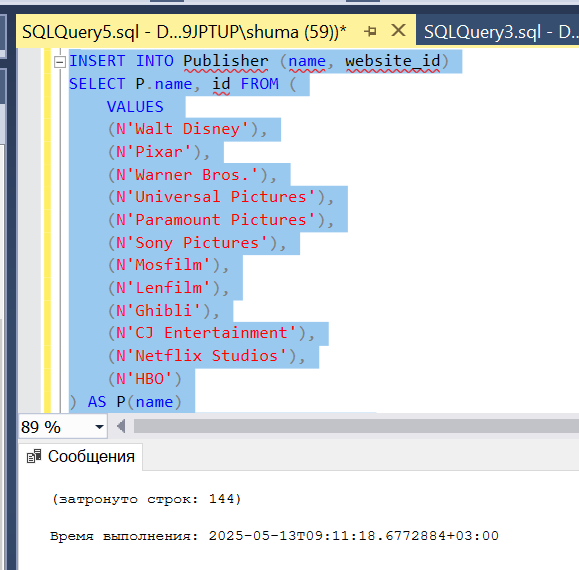




3. Внести в созданные таблицы БД информацию, соответствующую предметной области вашего варианта индивидального задания (не менее 10- 15 записей в каждой таблице, используя операторы DML. Тексты запросов сохранить в файле.







4. Сформировать не менее 10 запросов на выборку данных, соответствующих специфике предметной области, потребностям предполагаемых пользователей и описанию выходных документов (ЛР-4). Тексты запросов сохранить в файле. 5. Отчет о выполнении ЛР-12 производится демонстрацией преподавателю структуры созданной БД, содержания ее таблиц и выполнением сформированных в п.3 и п.4 запросов, а также оформлением отчета в MS Word и представления его на учебный портал.

Идеи запросов  
1) любимый жанр месяца

2) сколько фильмов посмотрел на английском

3)