**Создание БД**

1. **Введение**

Во время прохождения практики передо мной встала задача спроектировать и создать на локальном сервере элементарную базу данных.

Для достижения поставленной задачи мне очень помогли лекции и конспекты по таким предметам как ОПБД и ТРиЗБД, а также открытые интернет-источники.

Вооружившись тетрадью с конспектами и Google Chrome, я приступил к выполнению задачи.

1. **Проектирование БД**

Для создания базы данных необходимо её спроектировать: определиться с предметной областью, определить сущности БД, связи между сущностями (ER-диаграмма), обозначить необходимые атрибуты. Также для начала следует определить потенциальные первичные ключи, а после и внешние.

Начал я с предметной области базы данных. После небольшого количества времени, которое ушло на раздумье, я решил создать базу данных библиотеки, состоящей из четырех сущностей:

1 – Книги;

2 – Читатели;

3 – Авторы;

4 – Выдача книг;

Основные атрибуты сущностей:

Книги:

-Код книги (в дальнейшем будет выступать первичным ключом)

-Название книги

-Код автора (внешний ключ)

-Кол-во книг в наличии

Читатели:

-Читательский билет (первичный ключ)

-ФИО

-Номер телефона

Авторы:

-Код автора (первичный ключ)

-ФИО

Выдача книг:

-Код выдачи (первичный ключ)

-Код книги

-Дата выдачи

-Дата сдачи

-Читательский билет (внешний ключ)

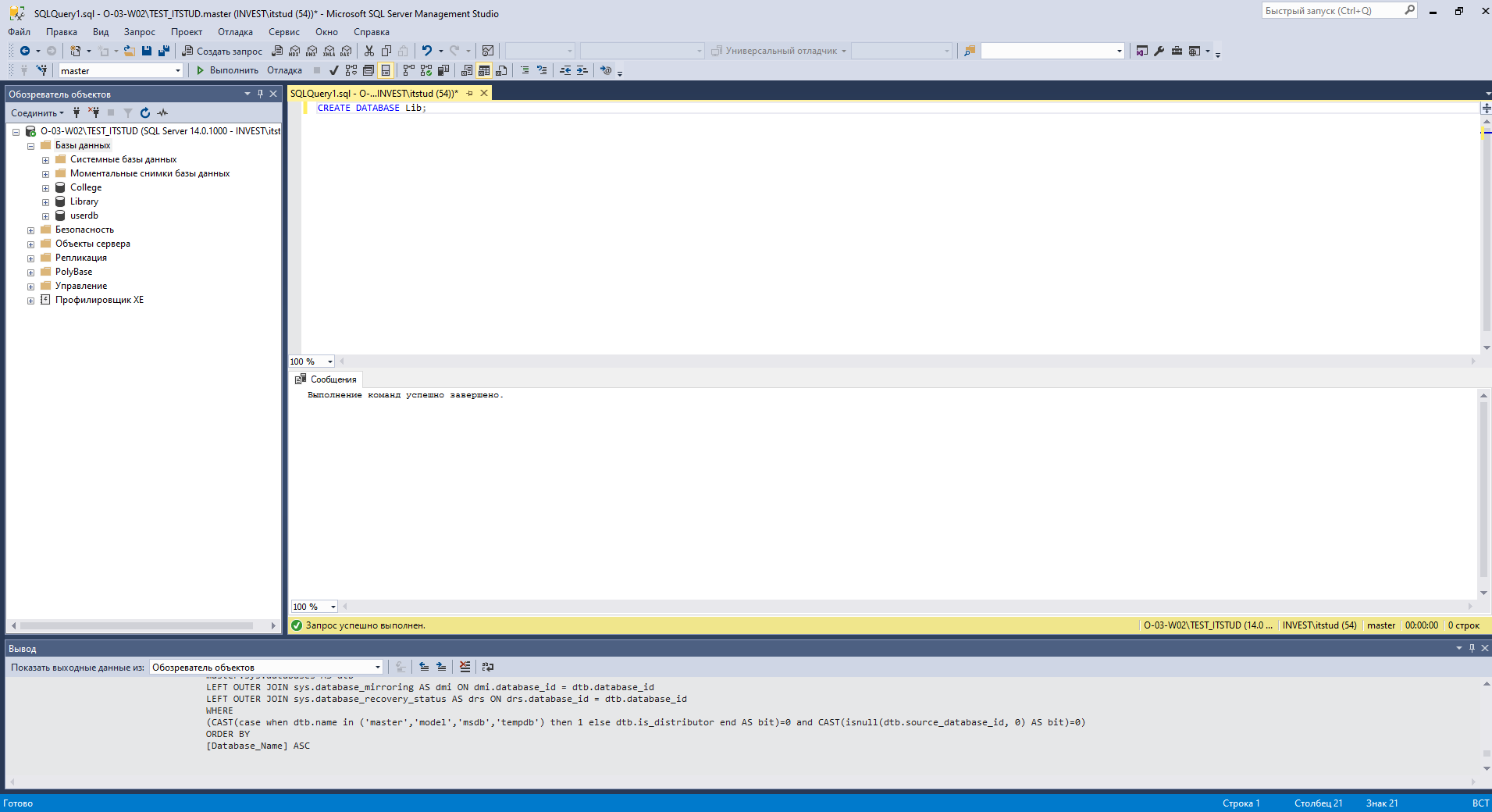
После того, как сущности, связи и атрибуты определены, я приступил к дальнейшему этапу – физическому проектированию БД. В данном отчете я совместил логическое и концептуальное проектирование и опустил такие этапы, как, например, определение типов данных. Все эти этапы я укажу в следующем разделе, когда буду создавать саму БД.

**3.Физическое проектирование БД**

В качестве СУБД для создания БД я выбрал MSSMS. Программная среда в организации, в которой я прохожу практику, совпадает с той, которую мы используем на занятиях в колледже, поэтому мне не пришлось перестраиваться под другое программное обеспечение.

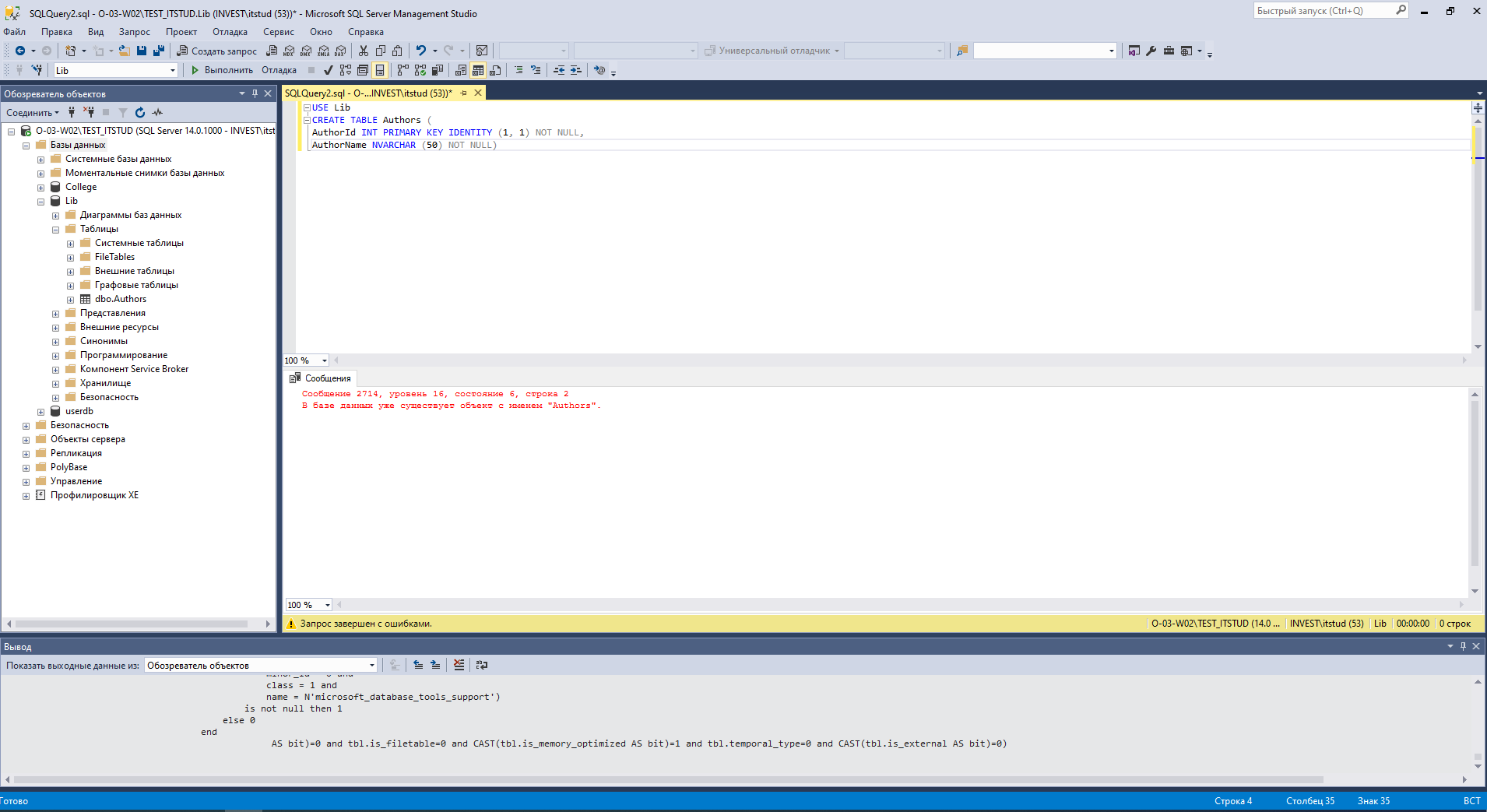
Все запросы, созданные мной в ходе выполнения данной работы, я приложу в RAR-архиве на Яндекс. Диске.

Ход выполнения работы:

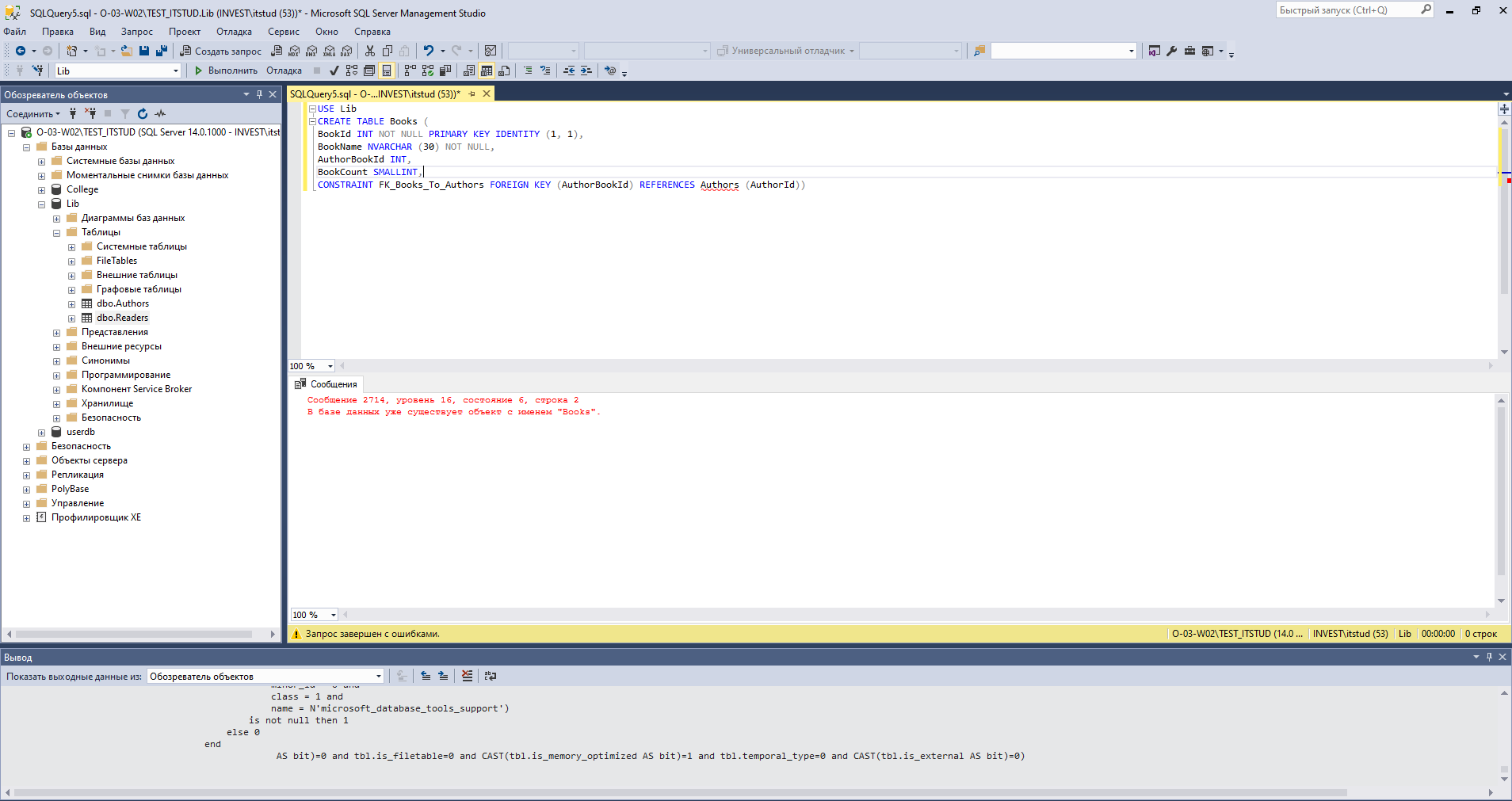
1. Для начала я создал локальную базу данных, используя SQL запрос.
2. Используя следующий запрос, я создал таблицу с авторами, где указал два атрибута, описал я их следующим образом:

AuthorsId соотвествует коду автора, является первичным ключом, имеет числовой типо данных, шаг заполнения в таблице 1 и не может быть пустым.

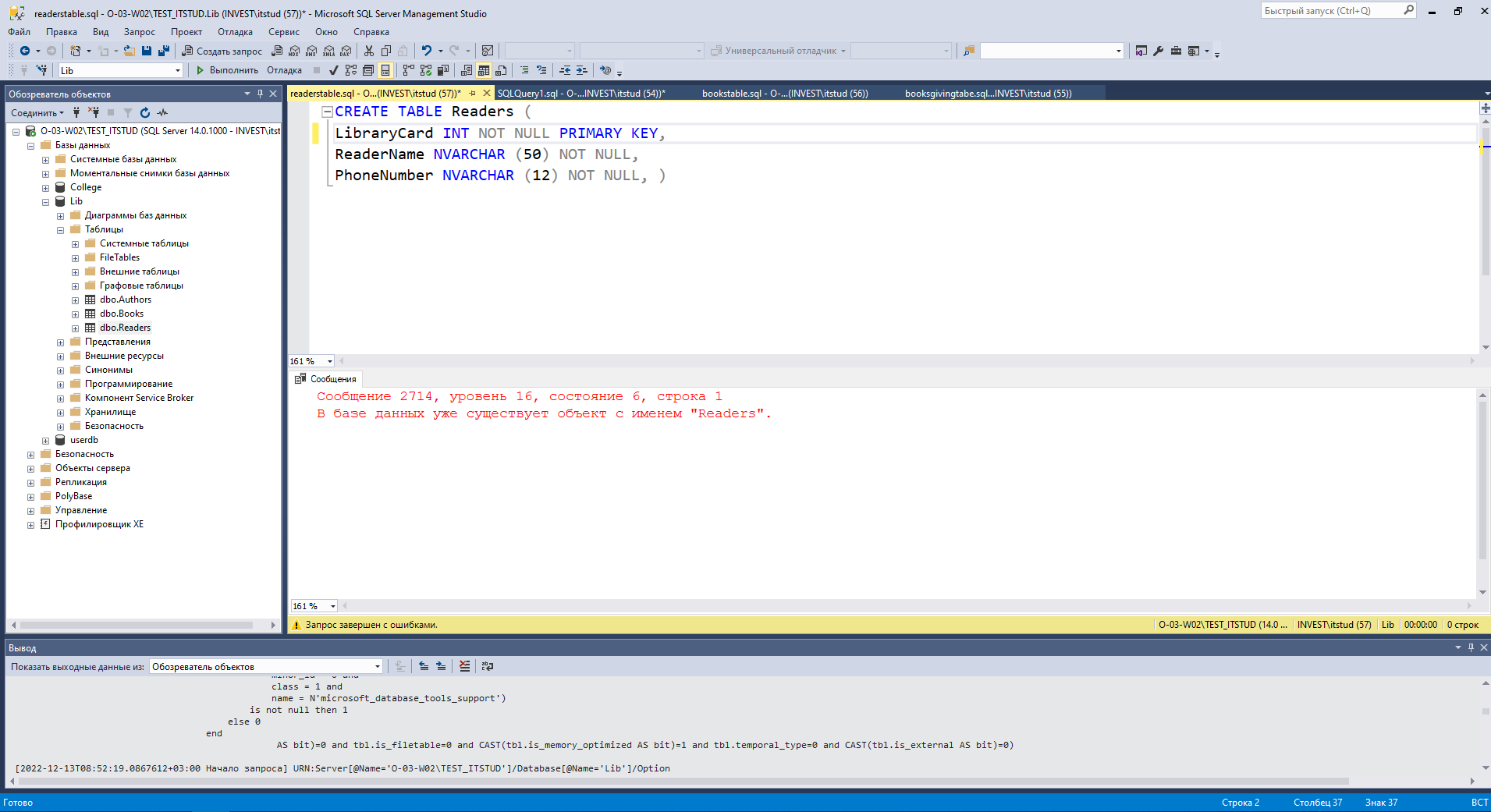
AuthorName соответсвует имени автора, имеет сторчный типо данных с маскимальной длиной – 50, не может быть пустым.

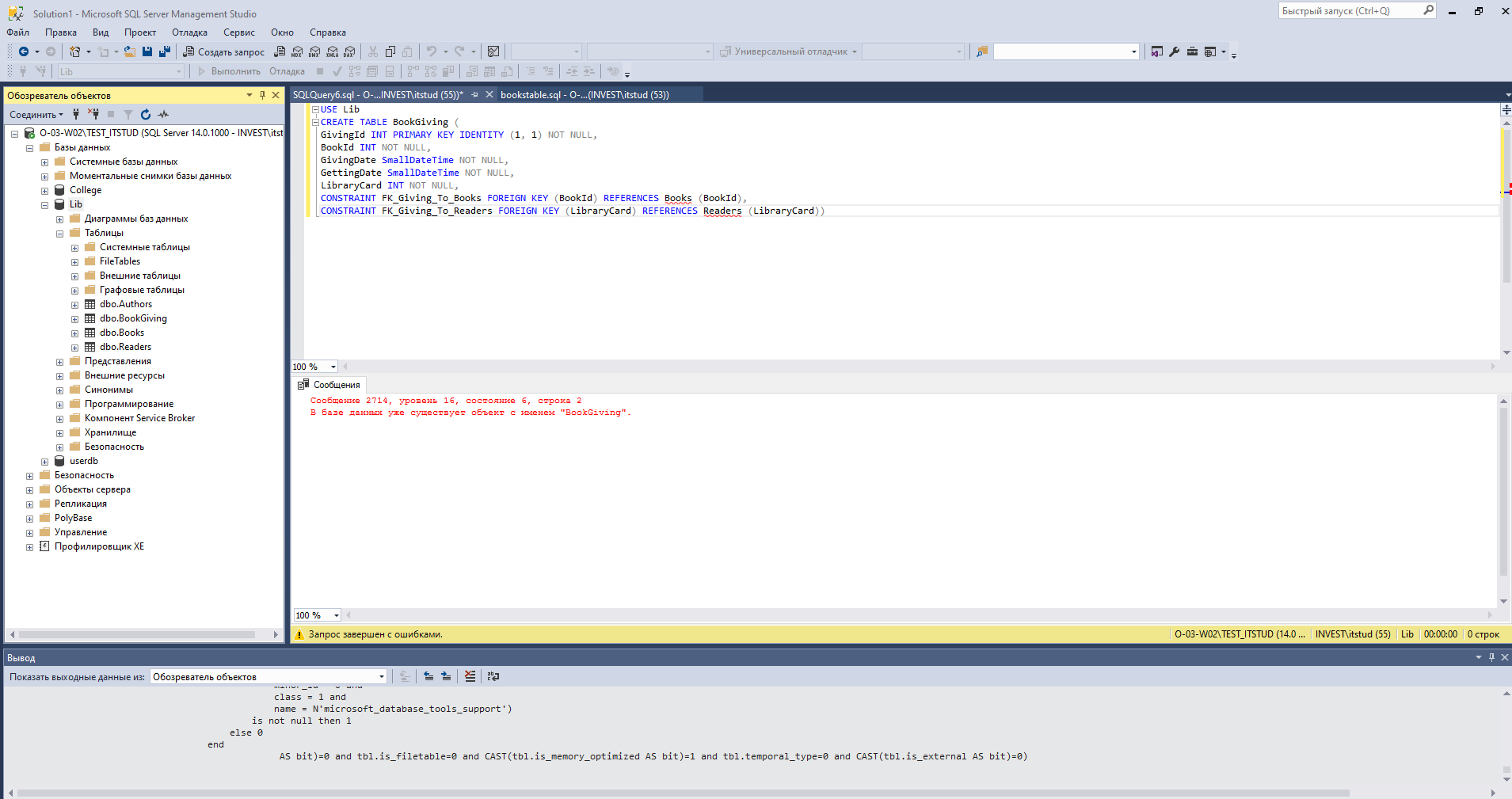
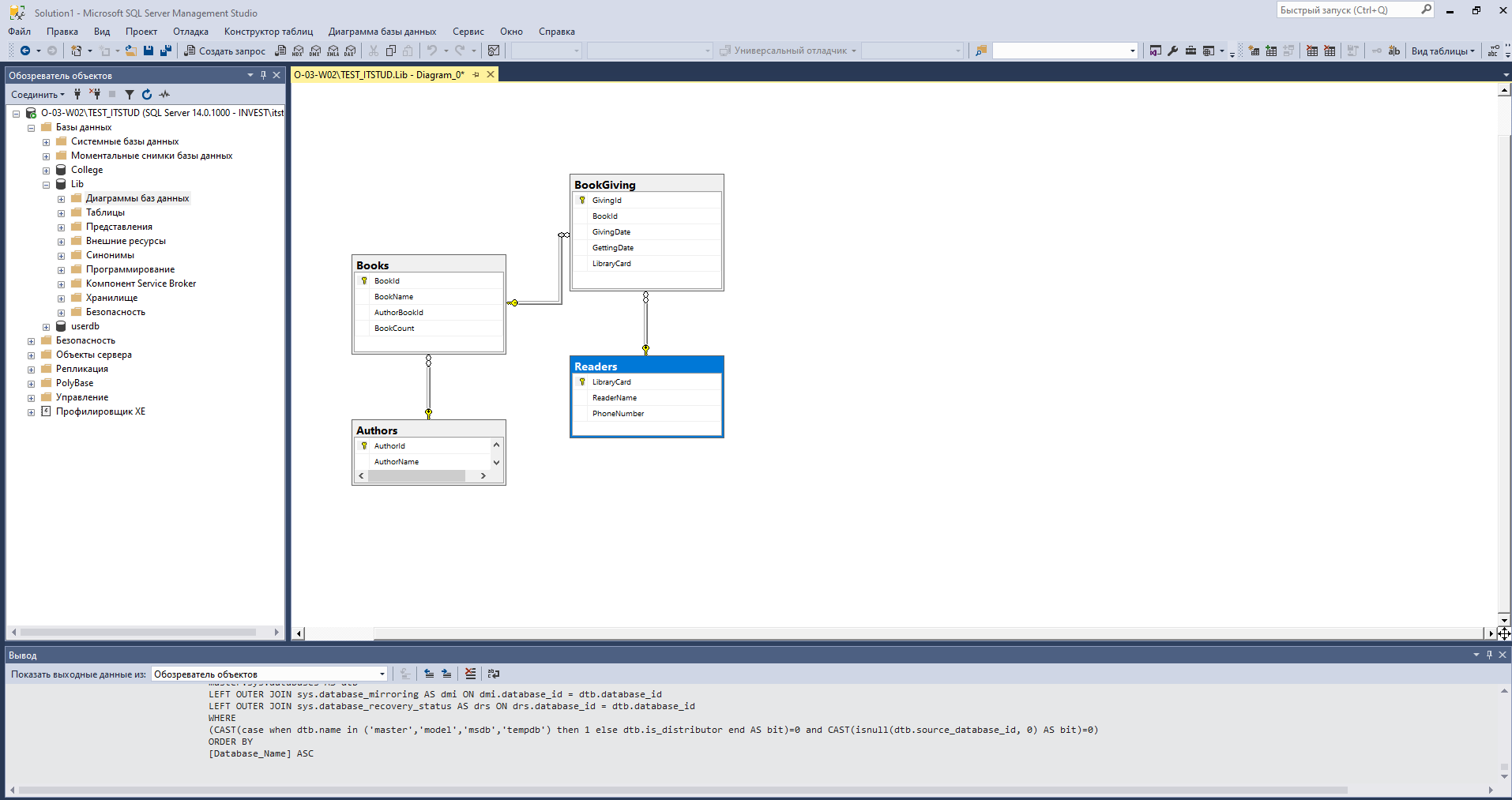


1. Дальше я создал таблицу с книгами по аналогии с предыдущим запросом, но в последней строке через ограничение я создал связь между двумя таблицами.

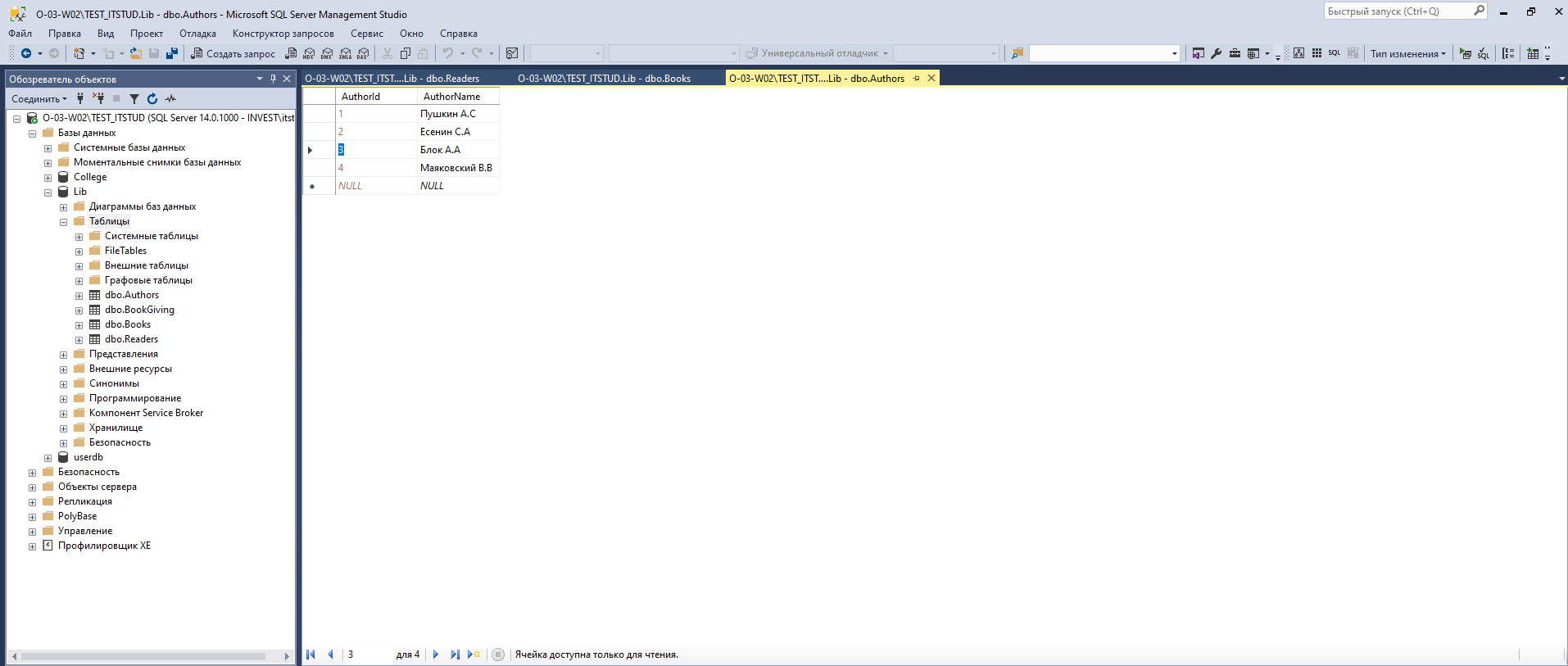


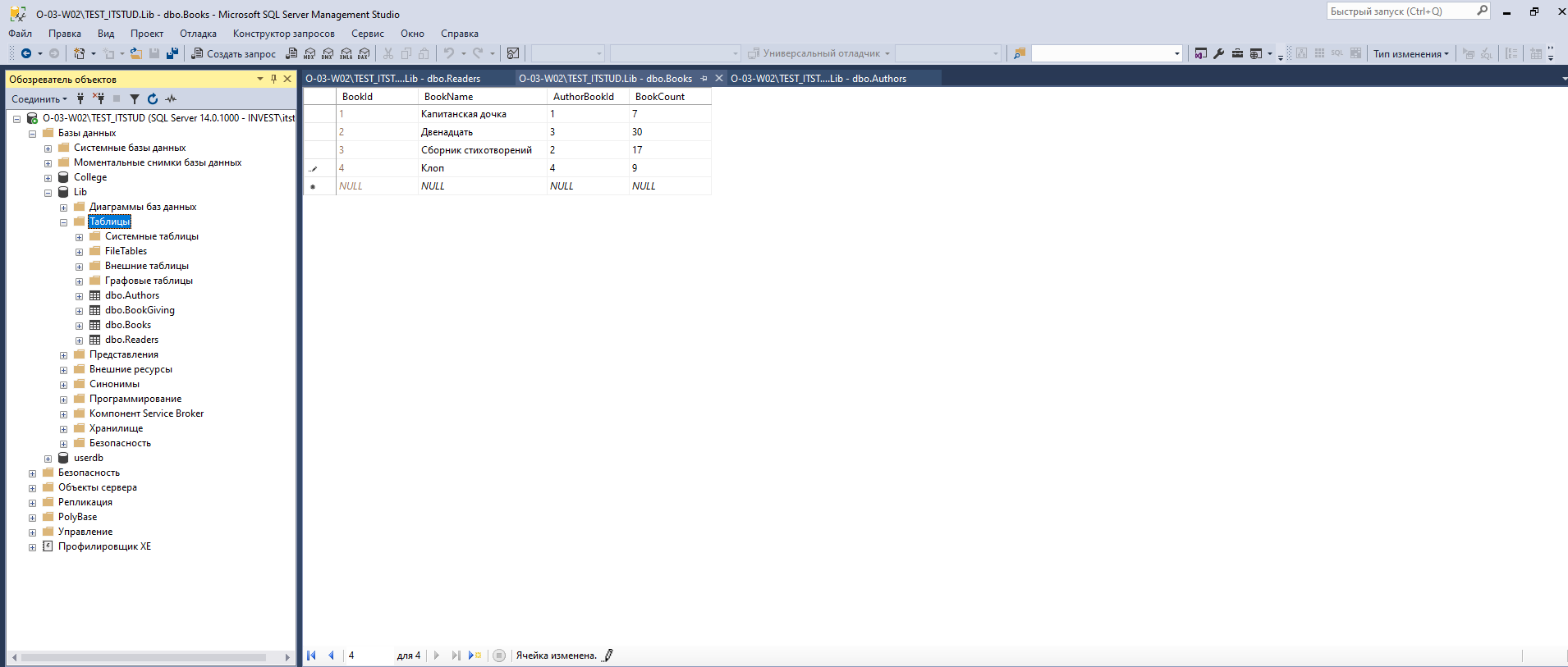
1. По аналогии с предыдущими шагами я создал таблицу читателей, не вижу смысла объяснять данный запрос.

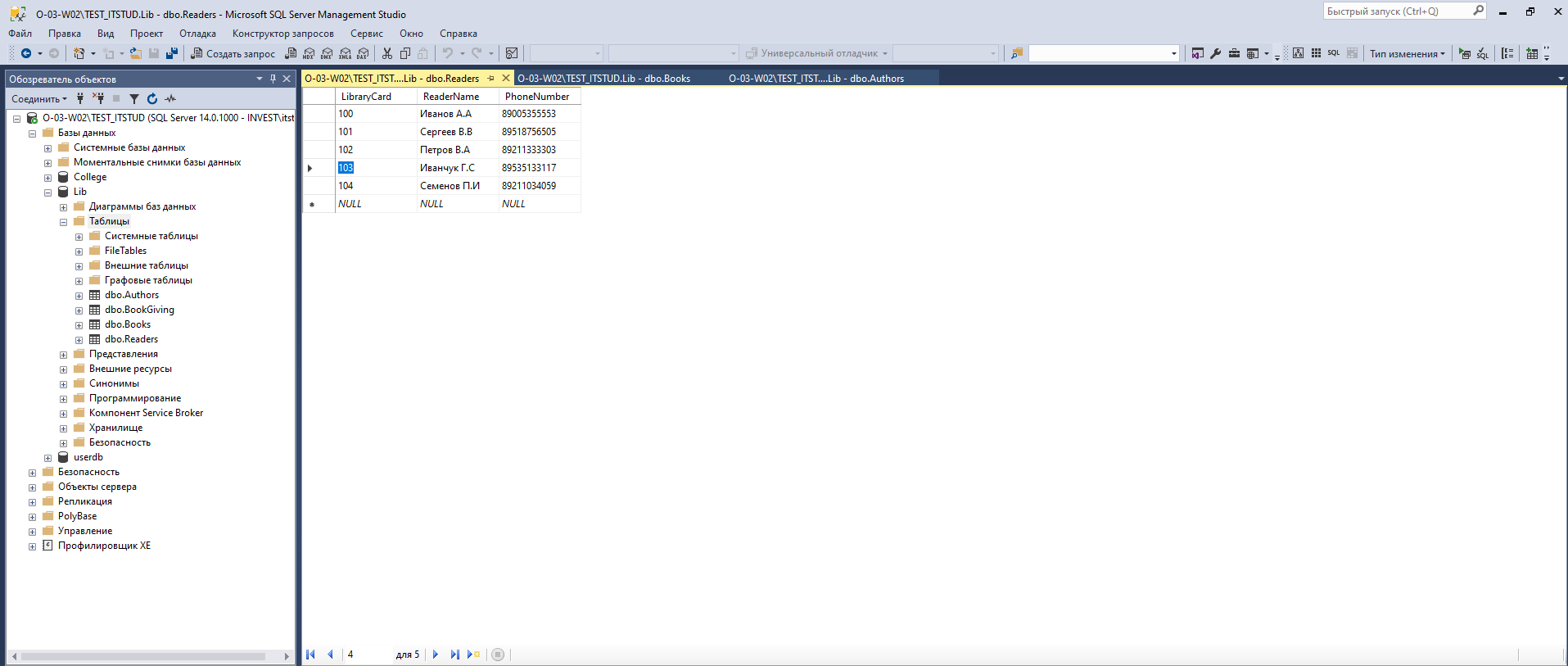


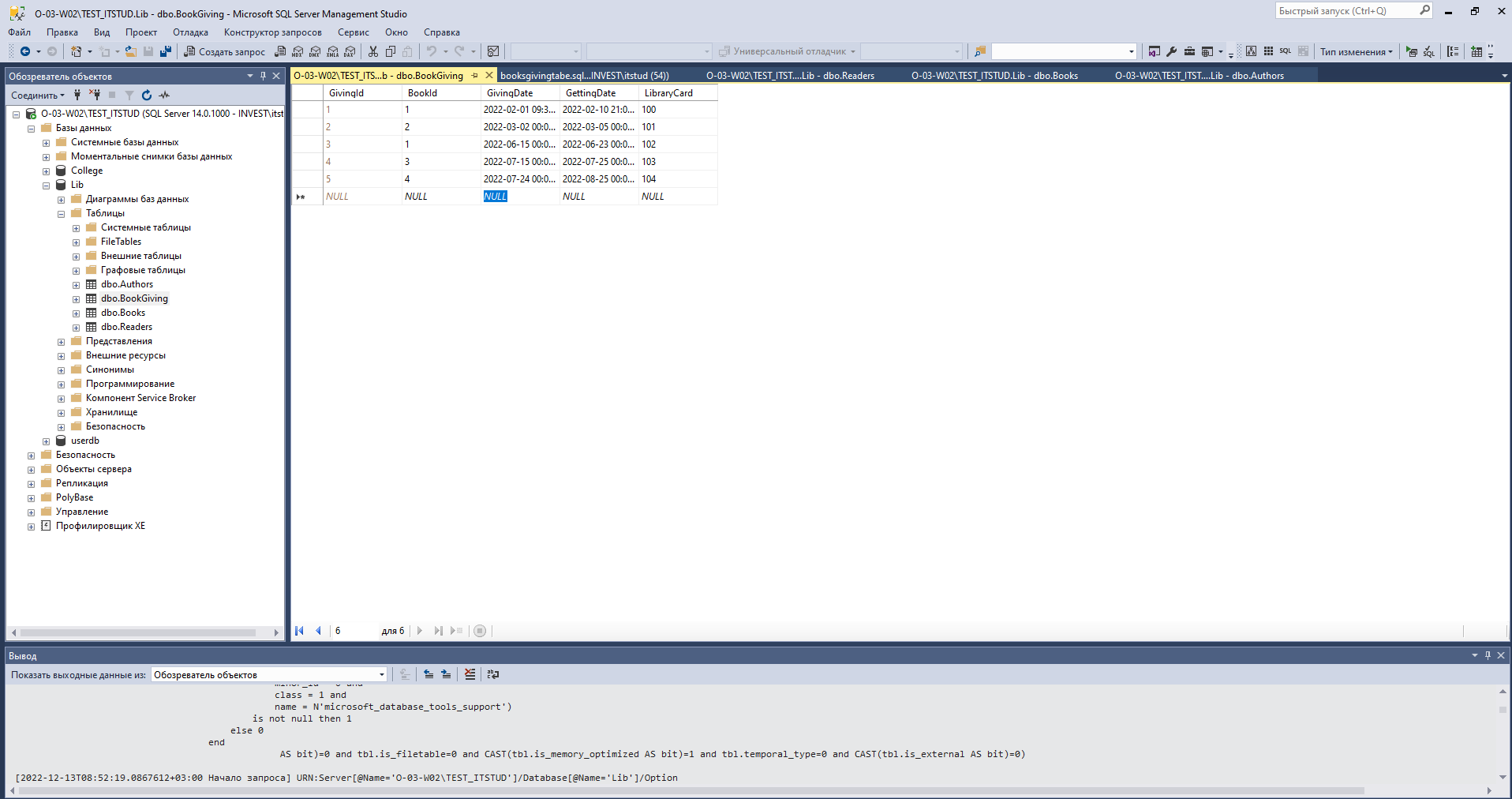
1. Создал таблицу выдачи книг последней, так как в ней было больше всего связей. Все представленные команды уже были использованы ранее, поэтому они не нуждаются в объяснении.
2. Все таблицы и связи успешно созданы. Дальше я создал диаграмму БД, чтобы в этом убедиться.

На скриншоте видно, что все связи были созданы и установлено корректно.

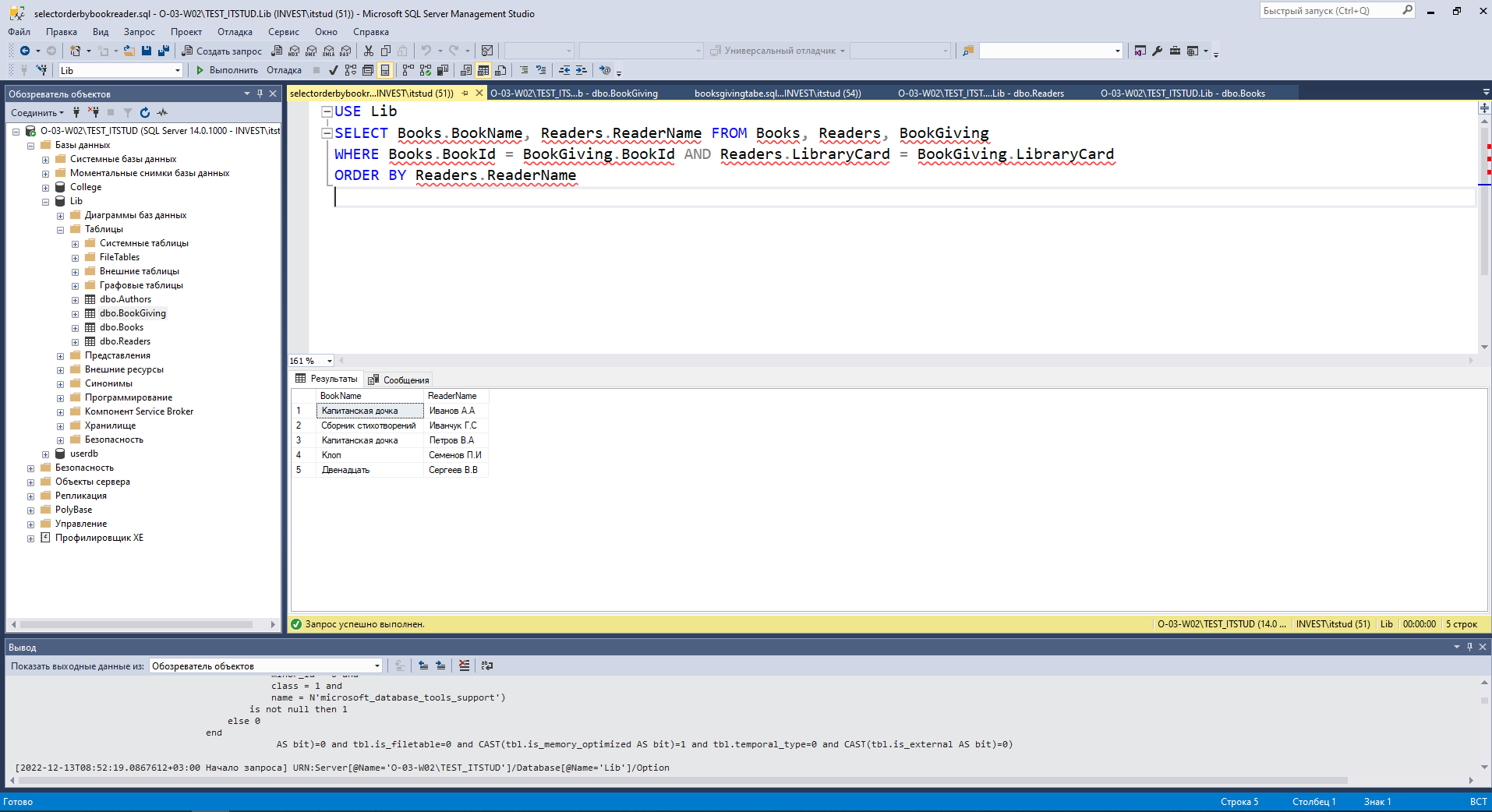
1. Дальше я заполню таблицу придуманными данными, чтобы была некая наглядность:

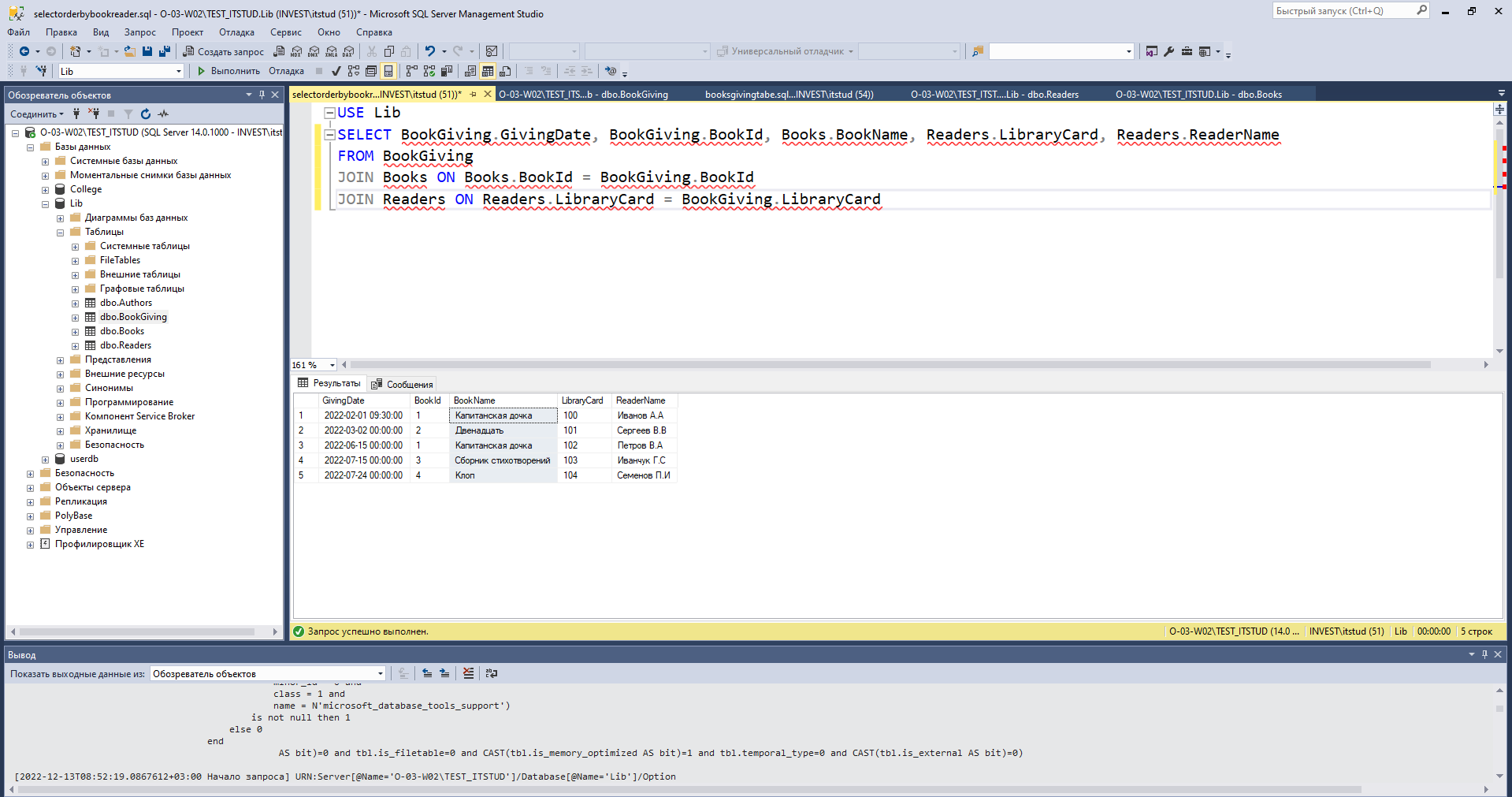


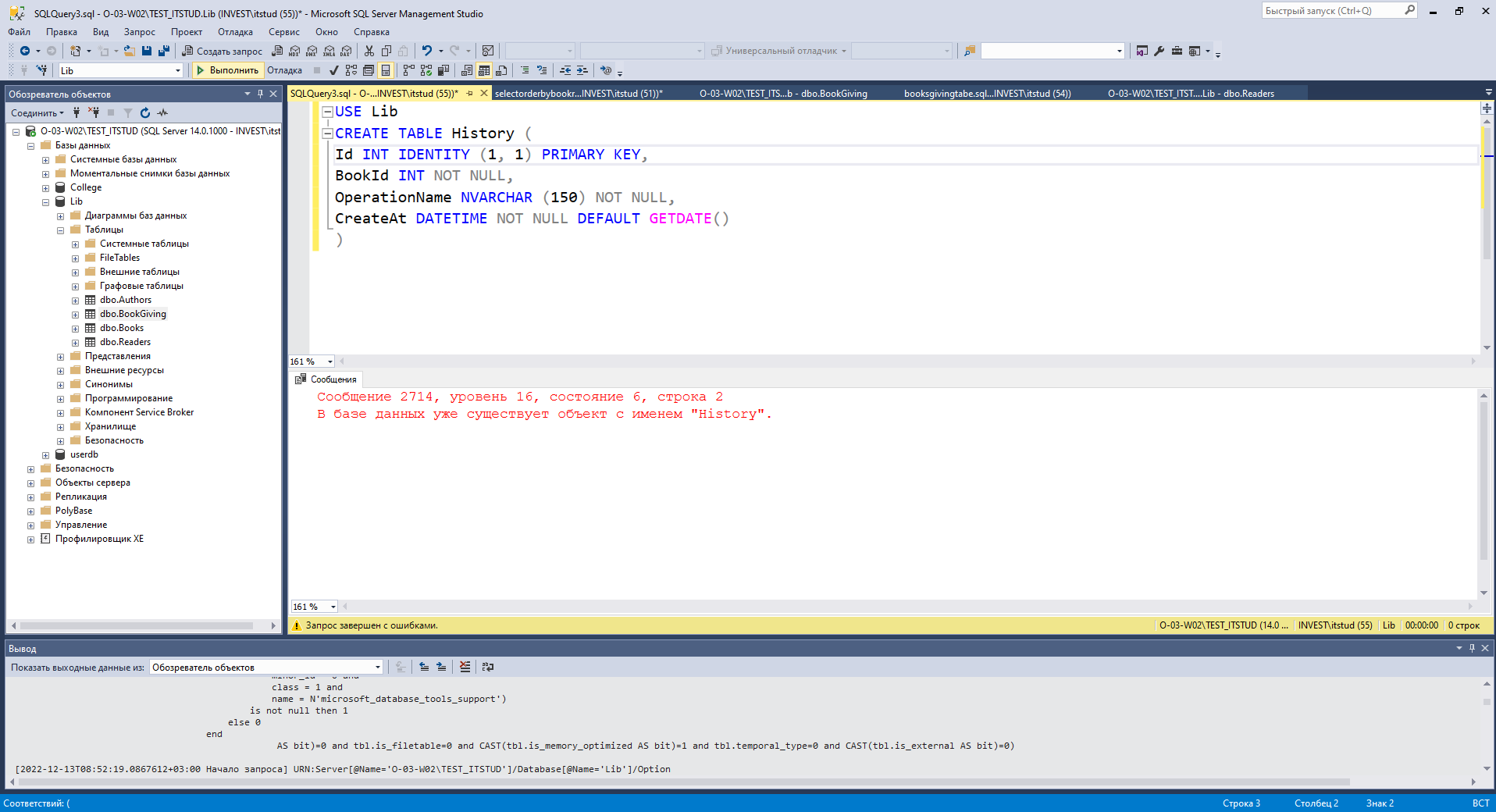
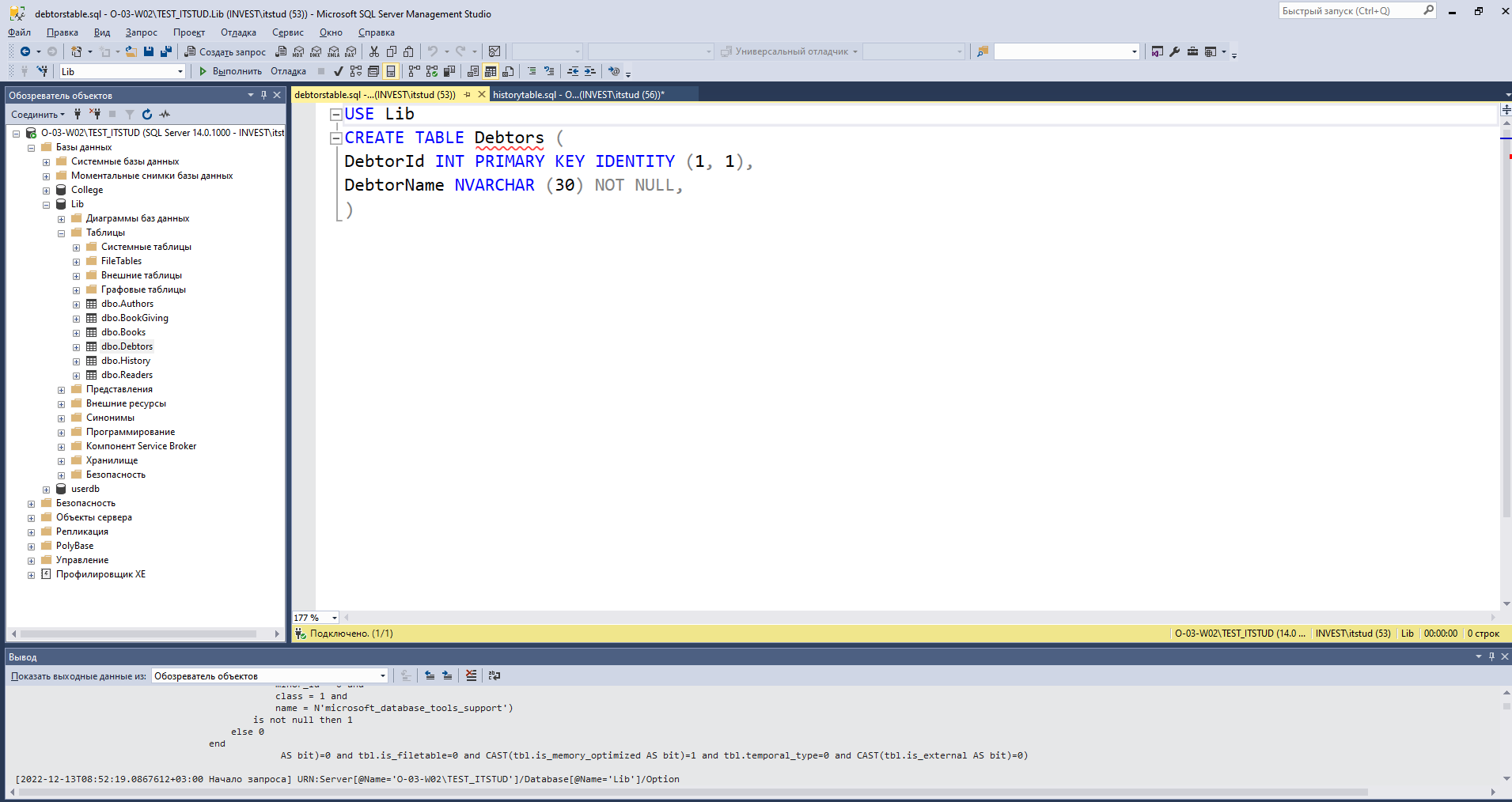


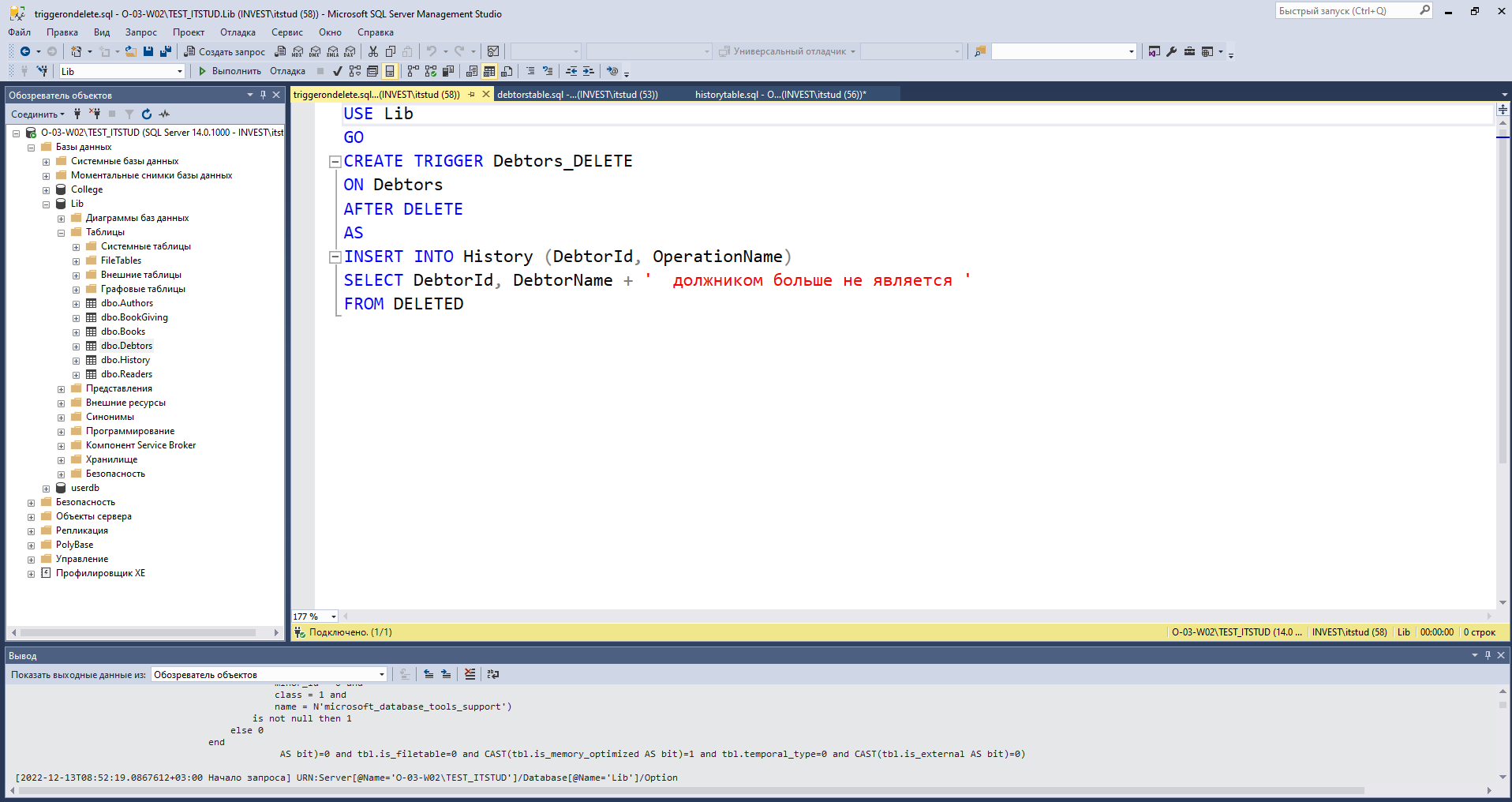


1. База данных создана и заполнена данными. Все сущности и данные в них соответствуют 6-и нормальным формам. Но без дополнительных запросов (например, выборка через SELECT), БД не имеет особого смысла и не может применяться эффективно. Поэтому следующим моим шагом стало написание запросов на выборку, INNER JOIN (OUTER JOIN в данной БД не будет нести смысла, так как она элементарна и все данные заполнены), триггера на удаление данных из БД. Эти запросы помогут наглядно показать смысл создания и ведения БД в целом.
   1. Данный запрос на выборку, включающий в себя функцию SELECT и сортировку по полю (ORDER BY), помогает узнать какую книгу (конкретное название), и кто (конкретное имя) взял.

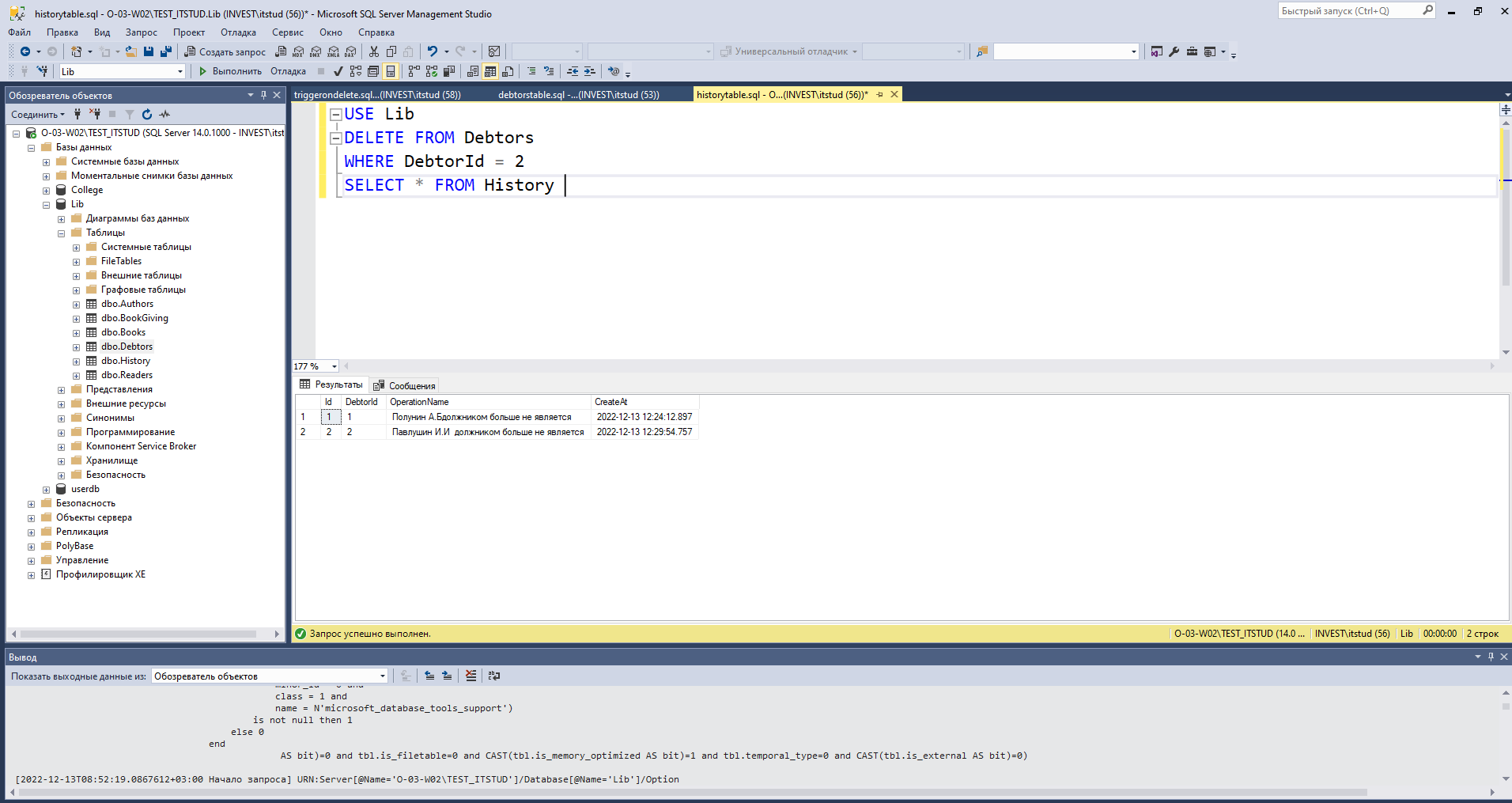


* 1. Далее представлен запрос внутреннего соединения таблиц (INNER JOIN). Когда используется данное соединение, слово INNER в синтаксисе может опускаться. Этот запрос выдает из таблиц данные о дате сдачи книг, Id книги, ее название, а также кто ее взял и Id его читательской карты.
  2. Триггер. Для реализации триггера на удаление данных из таблицы, мне потребовалось создать вспомогательную таблицу, где будет вестись учёт “отбытия” книг из библиотеки. Назвал я ее History. В данной таблице следующие атрибуты: Id операции, Id должника, наименование операции и непосредственно дата и время совершенной операции, которая берется с компьютера пользователя. А также завести новую таблицу по должникам. Её название – Debtors. Для примера я создал лишь два атрибута – Id должника и его имя. Чтобы избежать аномалии-удаления, следует не прибегать к каким-либо связям с другими таблицами, хотя они и не требуются.



Сам триггер выглядит следующим образом:

Чтобы проверить работоспособность триггера необходимо удалить из таблицы должников какую-либо строку. Производим эти манипуляции и получаем результат работы триггера:



1. **Заключение**

Поставленная задача выполнена – база данных готова. Для нее я составил дополнительные запросы, а также триггер на удаление данных. Всё корректно работает, а также нет каких-либо аномалий. В целом, пользуясь конспектами с лекций, а также дополнительными интернет-источниками, для меня не составило никаких трудностей спроектировать и сделать БД. Эту базу данных еще стоит доработать, но некоторыми аспектами я не владею. Например, обеспечение безопасности БД. Данную базу данных, а также все запросы я буду хранить на облачном хранилище Яндекс. Диск в RAR-архиве с паролем, дабы обеспечить минимальную безопасность данным.

В ходе выполнения данной работы я усвоил материал по проектированию и созданию БД, а также научился создавать резервную копию необходимой базы данных и инициализировать её на сервере.