Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

**ОТЧЕТ по лабораторной работе №**10

по дисциплине «Базы данных»

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил:**

Кандидат тех. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

Санкт-Петербург, 2025

Оглавление

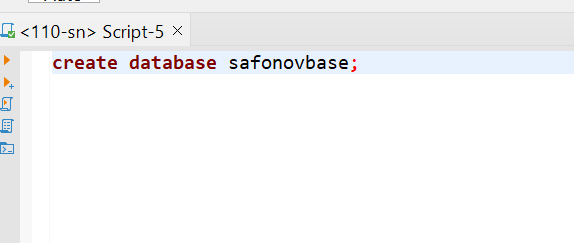
[Инфологическое проектирование баз данных. ER-диаграммы 3](#_Toc194495262)

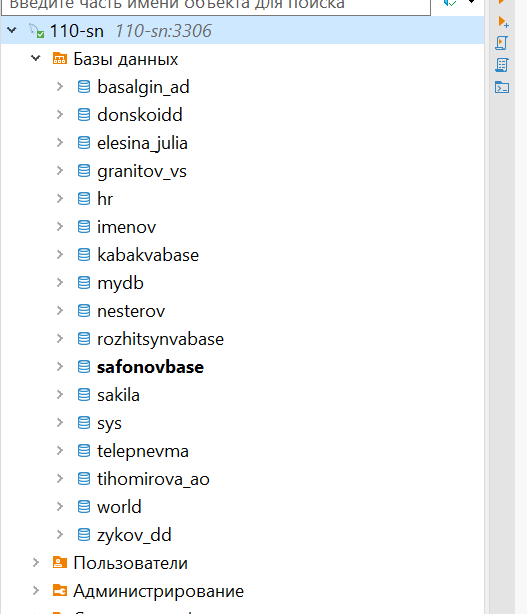
[ЛАБОРАТОРНАЯ РАБОТА 8\_1. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ В СРЕДЕ ERWIN, ЛАБОРАТОРНАЯ РАБОТА 8.2. НОТАЦИЯ IE. ЛОГИЧЕСКОЕ И ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ В СРЕДЕ ERWIN 3](#_Toc194495263)

[ЛАБОРАТОРНАЯ РАБОТА № 2 ВОСЬМОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ». 4](#_Toc194495264)

[Яндекс практикум тема "Симулятор" 7](#_Toc194495265)

Доп задание





Яндекс практикум

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

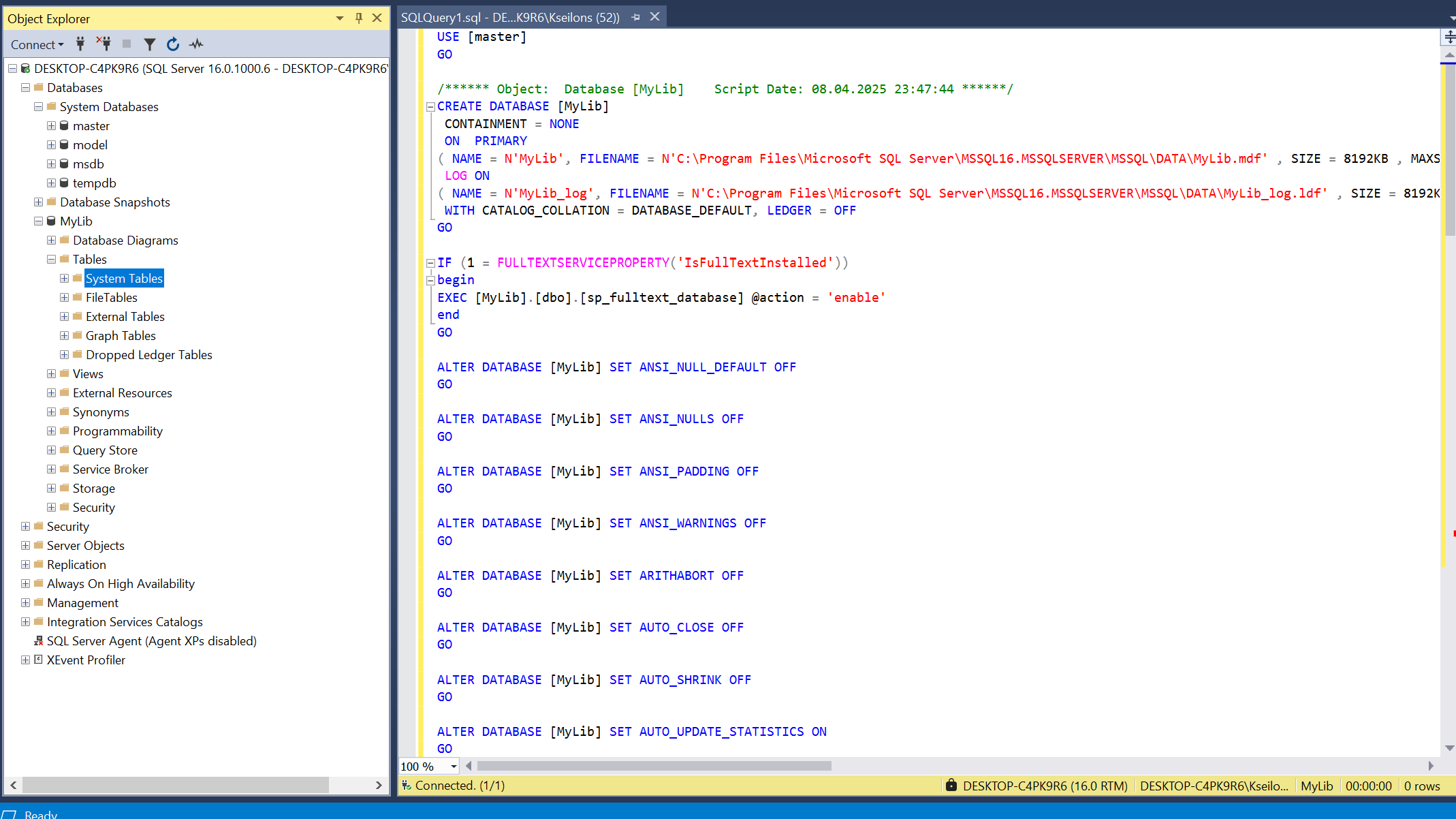
AI-generated content may be incorrect.

ЛАБОРАТОРНАЯ РАБОТА № 1 (SQL SERVER) ДЕВЯТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ».

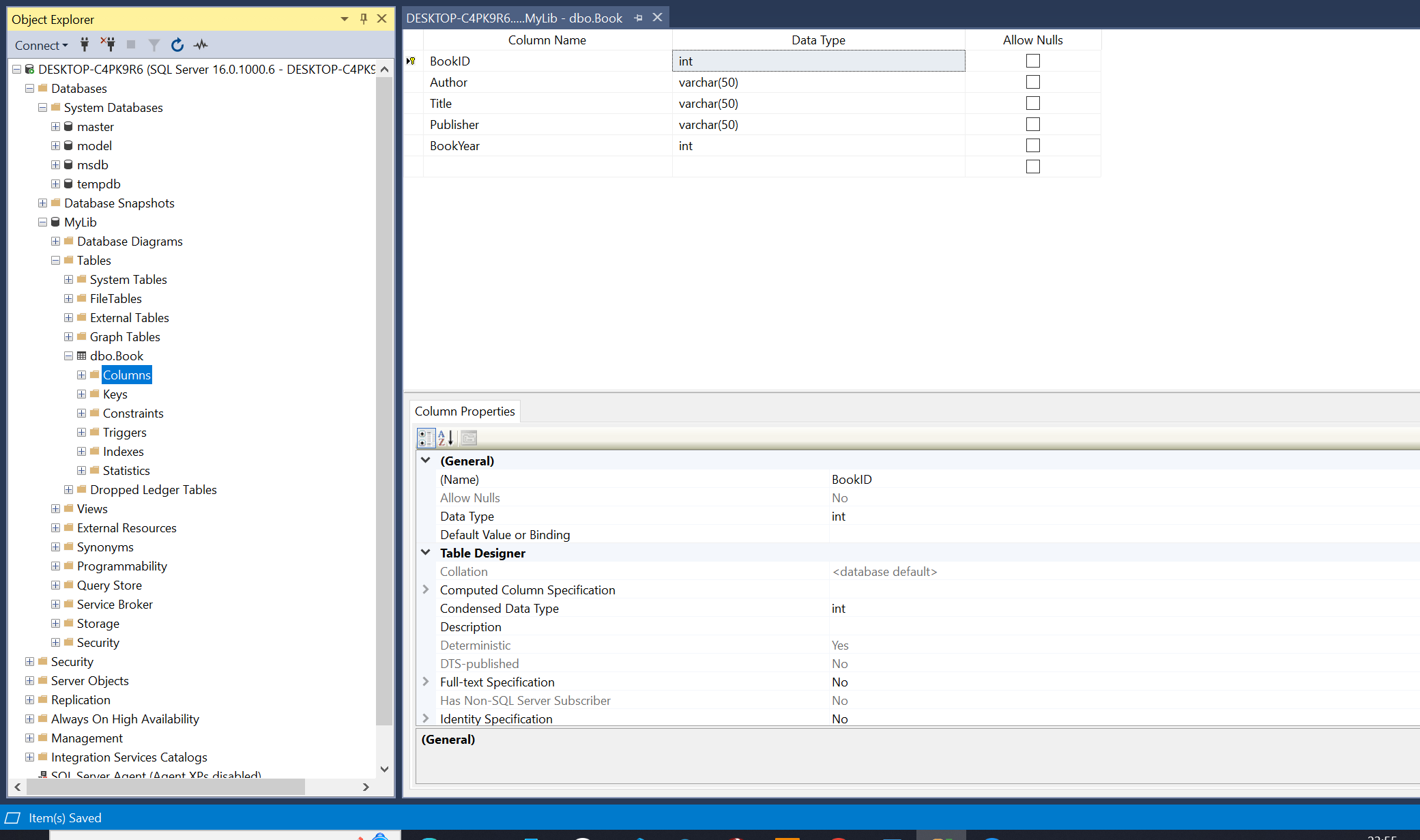
Задание. Запустите SQL Server Management Studio и подключитесь к серверу (рис. A.9.1, имя сервера зависит от настроек сделанных при установке).

Задание. С помощью Management Studio создайте новую базу данных (рис. A.9.2), назовите ее MyLib. Обратите внимание, на создаваемые файлы базы данных и журнала, название файловой группы для файла базы данных.

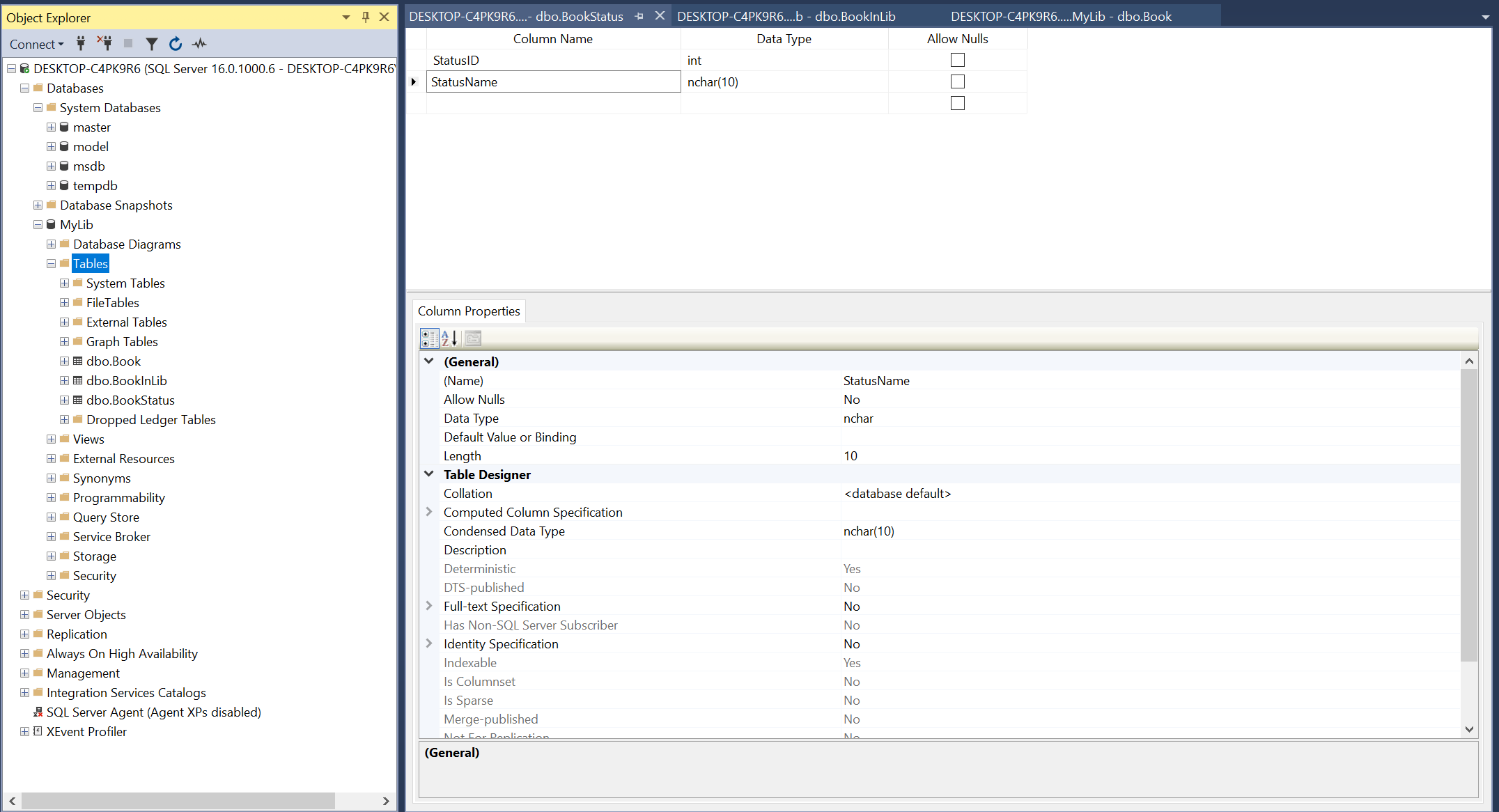
Задание. Получите по созданной базе скрипт и проанализируйте его.



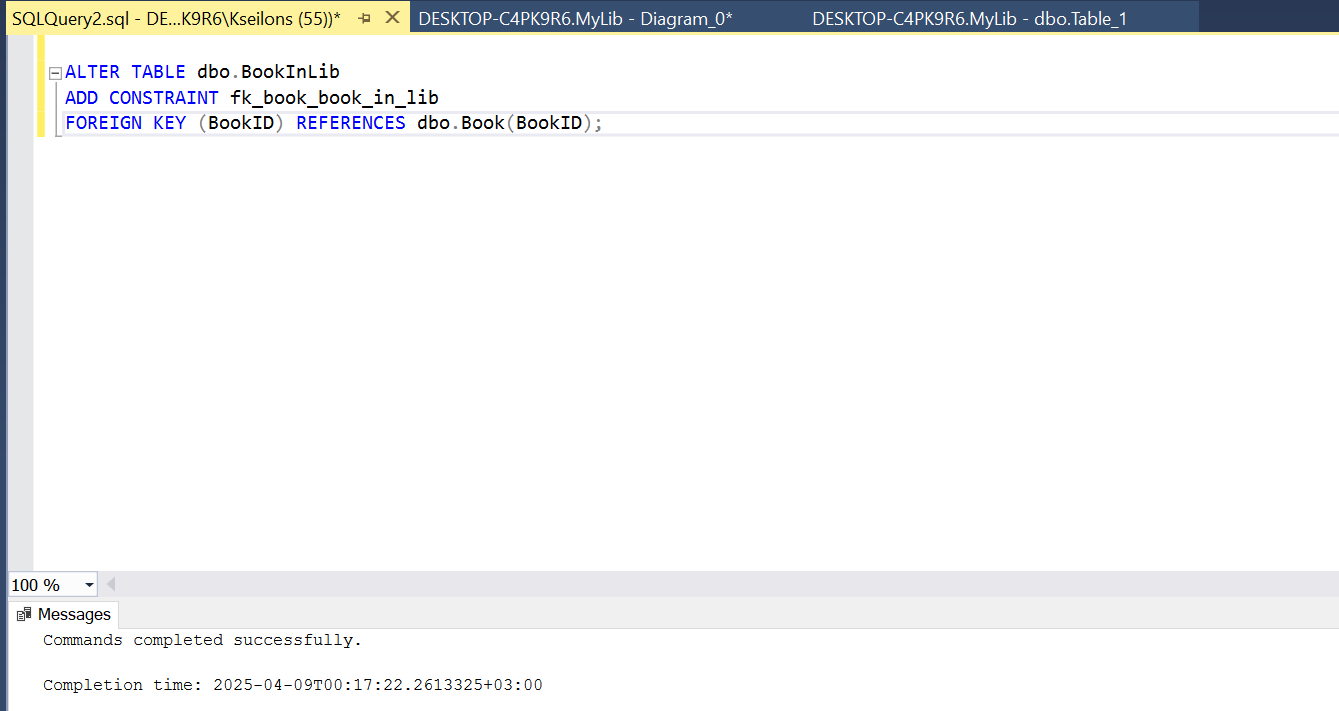
Задание. Создайте таблицу Book в графическом конструкторе (рис. A.9.4), определите ограничение первичного ключа и NOT NULL, как это ранее делалось для базы в Access. Аналогично тому, как ранее это 4 было сделано для базы данных, получите скрипт, создающий таблицу, и проанализируйте его.

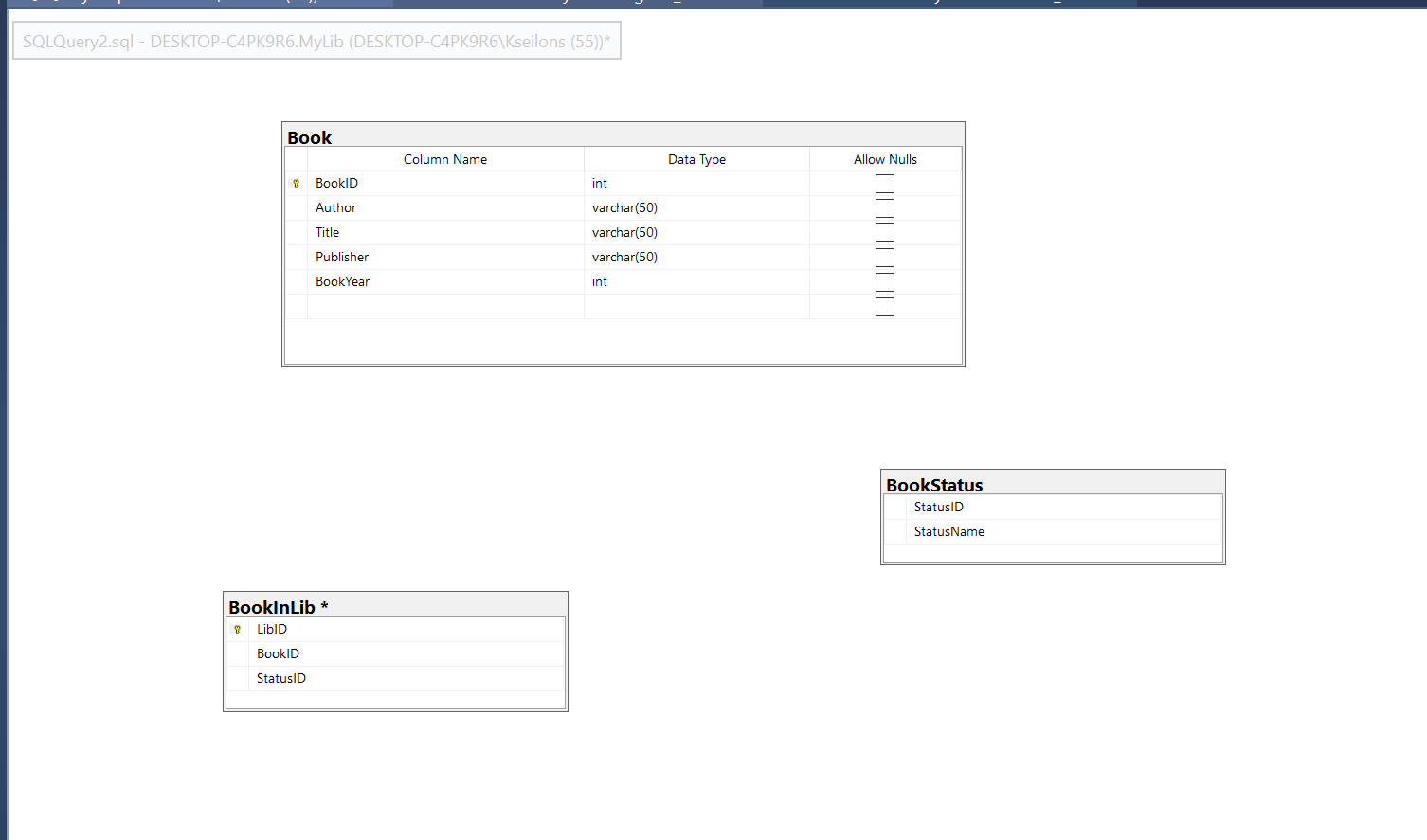


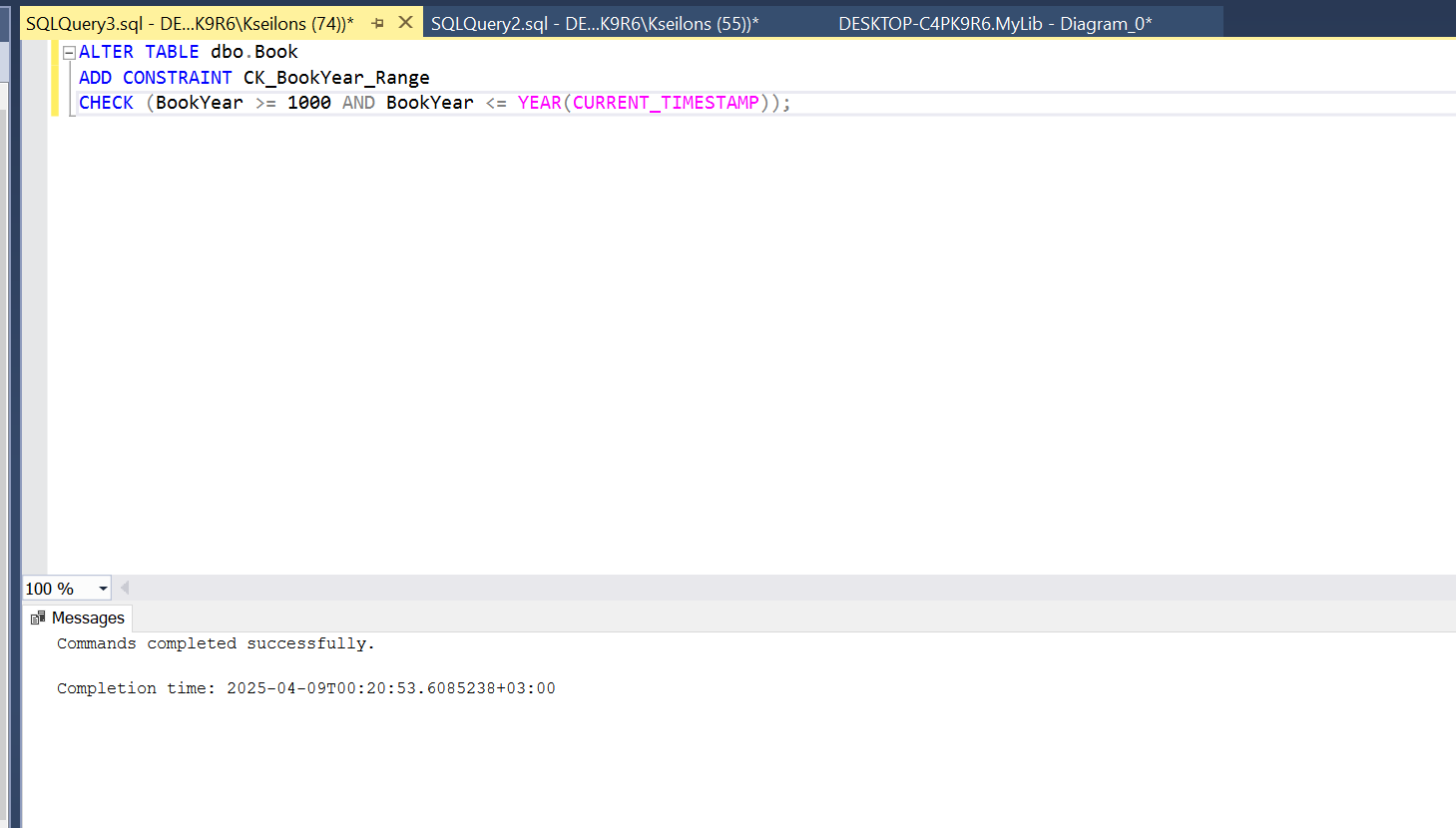
Задание. Создайте таблицы BookStatus и BookInLib с помощью скриптов CREATE TABLE. Внешние ключи не задавайте, в таблице BookStatus не забудьте определить ограничение уникальности для поля StatusName. Для указания БД, на которой выполняется скрипт, используйте выпадающий список в правой верхней части окна Management Studio или команду use :



Задание. Создайте диаграмму базы данных, добавьте на не созданные ранее таблицы. С использованием диаграммы создайте один внешний ключ. Получите скрипт для таблицы BookInLib. Второй ключ создайте с помощью скрипта (ALTER TABLE). Также добавьте ограничение на возможные значения поля BookYear в таблице Book: значение этого поля может быть в диапазоне от 1000 года до текущего включительно. Текущий год можно получить выражением Year(GetDate())



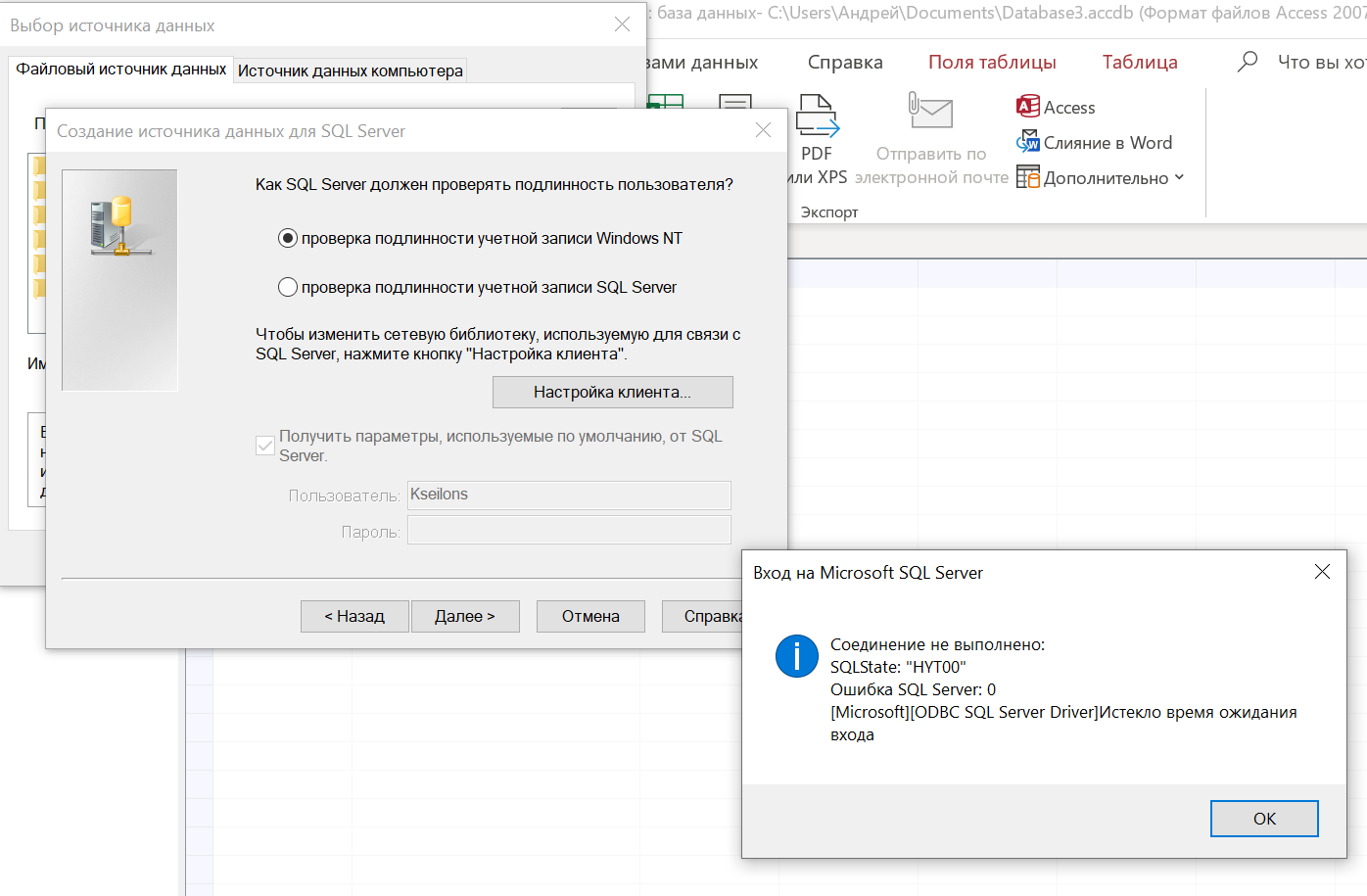


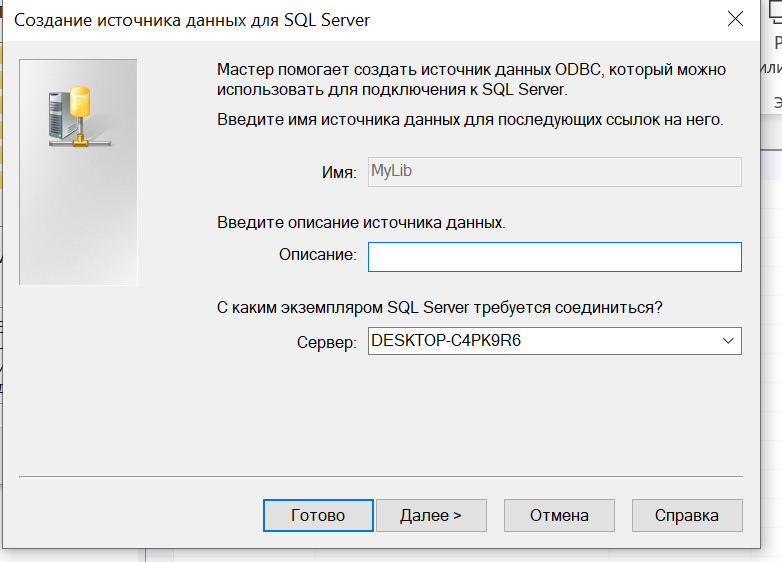


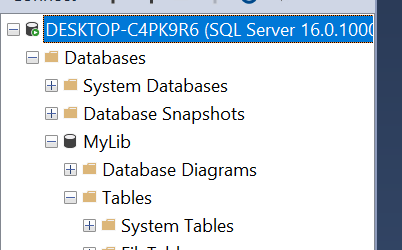
Задание. Создайте в Access новую базу данных. Укажите, что планируется использовать внешние данные. Это делается через меню Внешние данные ◊ База данных ODBC.

Для того, чтобы работать с данными, хранящимися на сервере, надо выбрать создание связанной таблицы (рис. A.9.5). После чего, потребуется создать именованный источник данных (DSN - Data Source Name), указывающий на SQL Server. Будем использовать файловый DSN, хранимый в виде файла c расширением \*.dsn. Используемый при подключении ODBC драйвер должен соответствовать SQL Server (рис. A.9.6.). ODBC драйвер – это программный компонент, отвечающий за подключение к конкретной СУБД, с учетом всех особенностей ее работы.

Укажите имя файла, в котором будет сохранен DSN, после чего опишите сервер к которому будете подключаться, укажите, что используете аутентификацию Windows, в настройке «Использовать по умолчанию» выберите вашу базу данных (обратите внимание, что автоматически подставляется системная база Master и эту настройку надо изменить) и нужные вам таблицы. Когда ссылки на внешние данные будут созданы (рис.A.9.7), проверьте, доступен ли вам ввод данных. Введите тестовый набор значений. При необходимости задайте подстановки и связи на схеме данных, создайте формы и отчет, выводящий информацию обо всех изданиях в библиотеке.





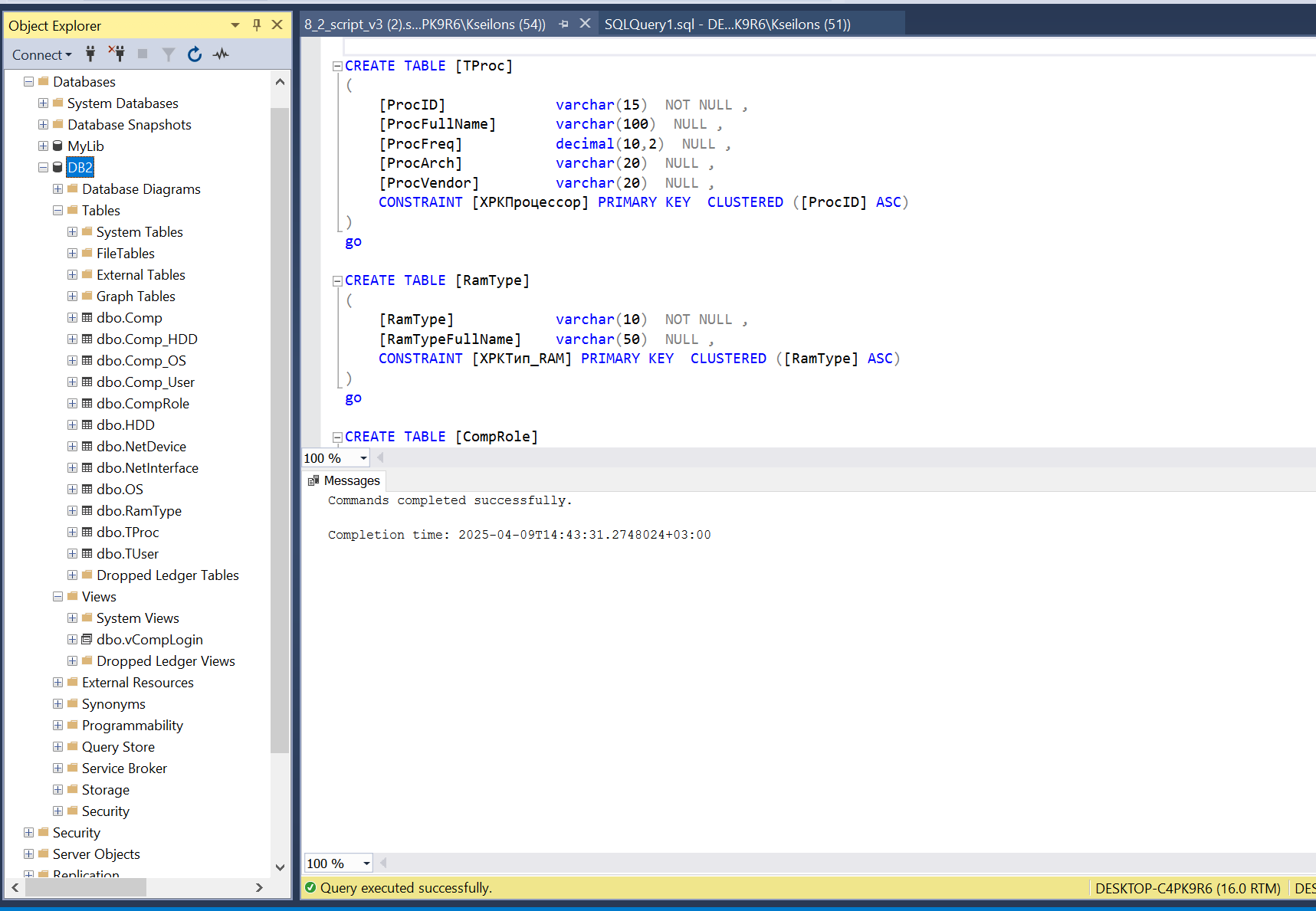
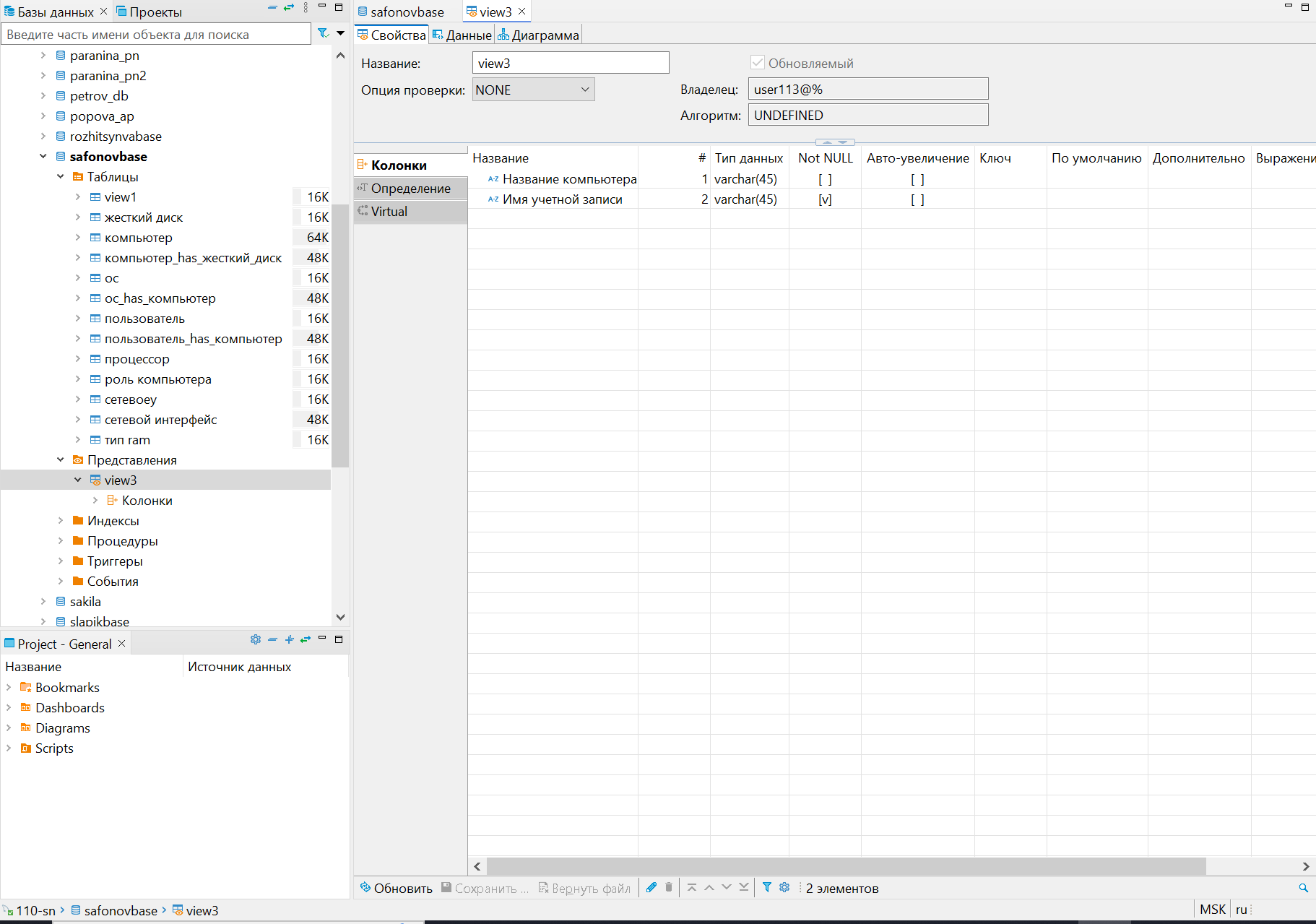


Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

АБОРАТОРНАЯ РАБОТА № 2 (SQL SERVER) ДЕВЯТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ».

В SQL Server Management Studio создайте новую базу данных. В ERwin откройте модель, созданную в ходе лабораторной работы 8.2. Определите подключение к экземпляру SQL Server (меню Action→ Database Connection), указав ранее созданную базу данных. Создайте на экземпляре SQL Server структуру базы данных, соответствующую вашей модели (меню Action-> Forward Engineer -> Schema…). В базе SQL Server проверьте наличие определенных в физической модели таблиц и представления. Если в коде представления были ошибки или используются отсутствующие таблицы / столбцы, оно не создастся, т.к. SQL Server при выполнении скрипта проводит синтаксический разбор, чего ERwin не делает. Найдите внешние ключи, опциями которых реализуются созданные в лабораторной работе 8.2 правила ссылочной целостности. Также эти правила поддерживаются триггерами, имена которых можно увидеть, раскрыв в Management Studio в окне Object Explorer (Обозреватель объектов) узел, соответствующий нужной таблице, и найдя там перечень триггер.

  
  
Workbranch  


Конспект

**1.7. Введение в язык SQ**

#### **1. Основные понятия**

SQL (Structured Query Language) – это язык для работы с реляционными базами данных. Он позволяет создавать, изменять, извлекать и управлять данными. SQL является декларативным языком, то есть пользователь описывает, **что** нужно получить, а система сама определяет, **как** это сделать.

#### **2. Группы операторов SQL**

1. **DDL (Data Definition Language)** – язык определения данных:
   * Используется для создания, изменения и удаления объектов БД (таблиц, индексов и др.).
   * Основные команды:
     + Создание таблицы (CREATE TABLE).
     + Изменение структуры таблицы (ALTER TABLE).
     + Удаление таблицы (DROP TABLE).
2. **DML (Data Manipulation Language)** – язык манипулирования данными:
   * Используется для работы с данными: добавление, изменение, удаление и выборка.
   * Основные команды:
     + Выборка данных (SELECT).
     + Добавление данных (INSERT).
     + Обновление данных (UPDATE).
     + Удаление данных (DELETE).

#### **3. Фильтрация данных**

Для выборки данных с определёнными условиями используются:

* Оператор WHERE – фильтрация по условию.
* Логические операторы (AND, OR, NOT) – комбинирование условий.
* Оператор LIKE – поиск по шаблону (например, имена на букву "А").
* Операторы IN и BETWEEN – проверка вхождения в список и диапазон значений.

#### **4. Сортировка и группировка**

* ORDER BY – сортировка результатов (по возрастанию или убыванию).
* GROUP BY – группировка данных (например, подсчёт количества людей каждого возраста).
* HAVING – фильтрация результатов после группировки.

#### **5. Агрегатные функции**

Используются для вычислений над наборами данных:

* COUNT() – подсчёт количества строк.
* SUM() – сумма значений.
* AVG() – среднее значение.
* MIN() и MAX() – минимальное и максимальное значения.

#### **6. Контрольные вопросы**

1. **DDL** нужен для управления структурой БД (создание, изменение, удаление таблиц).
2. **DML** нужен для работы с данными (добавление, изменение, удаление, выборка).
3. **Основные команды DDL**: создание, изменение и удаление таблиц.
4. **Основные команды DML**: выборка, вставка, обновление, удаление данных.
5. **Фильтрация данных**: WHERE, LIKE, IN, BETWEEN.
6. **Агрегатные функции**: COUNT, SUM, AVG, MIN, MAX.

Этот конспект кратко излагает основы SQL, включая ключевые команды и принципы работы с базами данных.

**1.8. Типы данных**

#### **1. Понятие типа данных**

Тип данных определяет:

* Множество значений, которые может содержать столбец, переменная или параметр.
* Допустимые операции для работы с этими значениями.

#### **2. Числовые типы данных**

**Целочисленные типы:**

* SMALLINT (2 байта): от -32 768 до +32 767.
* INTEGER (4 байта): от -2³¹ до +2³¹−1.
* BIGINT (8 байта): от -2⁶³ до +2⁶³−1.
* INT128 (16 байт): от -2¹²⁷ до +2¹²⁷−1.

**Числа с фиксированной точкой (точные):**

* DECIMAL(n, m) / NUMERIC(n, m)
  + n — общее количество знаков (1–38).
  + m — количество знаков после запятой (0–38).
  + Пример: DECIMAL(5, 2) для числа 123.45.
  + Размер памяти зависит от точности (2–16 байт).

**Числа с плавающей точкой (приближённые):**

* FLOAT (4 байта): от 1.175×10⁻³⁸ до 3.402×10³⁸.
* DOUBLE PRECISION (8 байт): от 2.225×10⁻³⁰⁸ до 1.179×10³⁰⁸.
* DECFLOAT (точный): 16 или 34 значащих цифры (8/16 байт).

**Операции:**

* Сложение, вычитание, умножение, деление.
* Для DECIMAL при делении важно указывать дробную часть (например, 1.00 / 3 = 0.33).

#### **3. Строковые типы данных**

**Фиксированная длина:**

* CHAR(n) / CHARACTER(n) — дополняется пробелами до длины n.
* Максимальный размер: 32 767 байт.

**Переменная длина:**

* VARCHAR(n) — хранит только фактическую длину строки.
* Максимальный размер: 32 765 байт.

**Двоичные данные:**

* BINARY(n) — фиксированная длина, дополняется нулями.
* VARBINARY(n) — переменная длина.

**Операции:**

* Конкатенация: 'Строка1' || 'Строка2'.
* Результат всегда CHAR (сохраняет пробелы).

#### **4. Логический тип данных**

* BOOLEAN: TRUE, FALSE, UNKNOWN (эквивалентно NULL).
* Примеры:
  + 1 <> 4 → TRUE.
  + NULL = NULL → NULL.
* Неявное преобразование только через CAST.

#### **5. Типы даты и времени**

**DATE:**

* Диапазон: 01.01.0001 – 31.12.9999.
* Форматы: 'dd.mm.yyyy', 'yyyy-mm-dd', '14-JUL-2025'.

**TIME:**

* Без часового пояса: TIME (например, '12:30:45').
* С часовым поясом: TIME WITH TIME ZONE.

**TIMESTAMP:**

* Дата + время: '14.07.2025 12:30'.
* С часовым поясом: TIMESTAMP WITH TIME ZONE.

**Функции:**

* NOW() — текущие дата и время.
* CURRENT\_DATE — текущая дата.

**Операции:**

* DATE + 1 → завтрашняя дата.
* DATE1 - DATE2 → интервал в днях.
* TIME1 - TIME2 → интервал в секундах.

#### **6. Тип BLOB (Binary Large Object)**

* Хранение больших данных: текст, изображения, аудио, видео.
* Подтипы:
  + 0 (BINARY) — неструктурированные данные.
  + 1 (TEXT) — текстовые данные (допускают строковые функции).
* Максимальный размер: до 2 ТБ (зависит от размера страницы БД).

#### **7. Контрольные вопросы**

1. **Тип данных** — это характеристика, определяющая допустимые значения и операции.
2. **Целочисленные типы**: SMALLINT, INTEGER, BIGINT, INT128.
3. **Фиксированная vs плавающая точка**:
   * Фиксированная (DECIMAL) — точные значения.
   * Плавающая (FLOAT) — приближённые, с погрешностью.
4. **Типы для плавающей точки**: FLOAT, DOUBLE PRECISION, DECFLOAT.
5. **Особенности умножения/деления для**DECIMAL: результат зависит от количества дробных знаков операндов.
6. CHAR**vs**VARCHAR: фиксированная vs переменная длина.
7. **Операции для строк**: конкатенация (||).
8. **Значения**BOOLEAN: TRUE, FALSE, UNKNOWN (NULL).
9. **Диапазон**DATE: 0001-01-01 – 9999-12-31.
10. **Сравнение времени в разных поясах**: используйте TIME WITH TIME ZONE.
11. **Сложение двух**DATE: недопустимо (только вычитание).
12. **Вычитание**TIME: интервал в секундах.
13. **Назначение**BLOB: хранение больших бинарных данных.
14. **Подтипы**BLOB: BINARY (0), TEXT (1).

**1.9. Домены**

#### **1. Создание домена (CREATE DOMAIN)**

**Домен** — это пользовательский тип данных с заданными ограничениями, который можно использовать для столбцов таблиц.

**Синтаксис:**

CREATE DOMAIN имя\_домена [AS] тип\_данных

[DEFAULT {литерал | NULL | контекстная\_переменная}]

[NOT NULL]

[CHECK (условие)]

[COLLATE порядок\_сортировки];

**Параметры:**

* **DEFAULT** — значение по умолчанию (используется в INSERT, если значение не указано).
* **NOT NULL** — запрет на NULL (обязательно для первичных ключей).
* **CHECK** — ограничение на допустимые значения.
* **COLLATE** — правило сортировки (например, для строк).

**Примеры:**

-- Домен для возраста (≥18)

CREATE DOMAIN D\_AGE AS INT

DEFAULT 18

CHECK (VALUE >= 18);

-- Домен для пола ('М' или 'Ж')

CREATE DOMAIN D\_GENDER AS CHAR(1)

CHECK (VALUE IN ('М', 'Ж'));

-- Домен с подзапросом (значение должно быть в таблице COUNTRY)

CREATE DOMAIN D\_COUNTRY\_CODE AS CHAR(3)

CHECK (VALUE IN (SELECT CODCOUNTRY FROM COUNTRY));

#### **2. Изменение домена (ALTER DOMAIN)**

Позволяет изменить параметры существующего домена.

**Синтаксис:**

ALTER DOMAIN имя\_домена

[TO новое\_имя]

[SET DEFAULT {литерал | NULL} | DROP DEFAULT]

[SET NOT NULL | DROP NOT NULL]

[ADD CONSTRAINT CHECK (условие) | DROP CONSTRAINT]

[TYPE новый\_тип\_данных];

**Пример:**

-- Изменение домена D099: новый CHECK и DEFAULT

ALTER DOMAIN D099

DROP DEFAULT,

SET DEFAULT CURRENT\_USER,

DROP CONSTRAINT,

ADD CONSTRAINT CHECK (

SUBSTRING(UPPER(VALUE) FROM 1 FOR 1) =

SUBSTRING(UPPER(VALUE) FROM 2 FOR 1)

);

**Важно:**

* При добавлении NOT NULL к домену, все связанные столбцы не должны содержать NULL.
* Изменение типа данных возможно, только если существующие данные могут быть преобразованы.

#### **3. Удаление домена (DROP DOMAIN)**

Удаляет домен, если на него нет ссылок в таблицах или процедурах.

**Синтаксис:**

DROP DOMAIN имя\_домена;

**Пример:**

DROP DOMAIN CODCOUNTRY;

**Ограничения:**

* Нельзя удалить домен, если он используется в таблицах.
* Сначала нужно удалить или изменить все зависимые столбцы.

#### **4. Контрольные вопросы**

1. **Когда применяется значение по умолчанию?**
   * При вставке строки (INSERT), если значение для столбца не указано.
2. **Как ограничить значения домена?**
   * Через CHECK (например, CHECK (VALUE IN ('М', 'Ж')).
3. **Что будет, если добавить**NOT NULL**к домену с существующими**NULL**?**
   * Ошибка: нельзя применить ограничение, пока в таблицах есть NULL.
4. **Пример удаления домена:**

DROP DOMAIN D\_GENDER;

**Итог:**

* Домены упрощают контроль целостности данных.
* CREATE DOMAIN задаёт правила, ALTER DOMAIN изменяет их, DROP DOMAIN удаляет.
* Ограничения (CHECK, NOT NULL) помогают избежать некорректных значений.

Red Expert