|  |  |
| --- | --- |
| **лого для документов 2022** | ***Федеральное агентство по рыболовству***  ***Федеральное государственное бюджетное образовательное***  ***учреждение высшего образования***  ***«Астраханский государственный технический университет»***  **Система менеджмента качества в области образования, воспитания, науки и инноваций**  **ООО «ДКС РУС» по международному стандарту ISO 9001:2015** |

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ

**КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**БАЗЫ ДАННЫХ**

(*наименование дисциплины*)

**ОТЧЕТ**

о выполнении заданий к лабораторной работе № 15

**Язык управления данных DML.**

|  |  |
| --- | --- |
|  | Выполнила:  студент гр. ДИНРБ-31 |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кузургалиев Р.А. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2023 г. |
|  | Максимальное количеству баллов \_\_\_\_\_  ЗАЩИЩЕНО:  Получено баллов \_\_\_\_\_\_ |
|  | Преподаватель: \_\_\_\_\_\_\_\_ Мамлеева А.Р. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

**Астрахань – 2023**

**Лабораторная работа №15 «Язык управления данных DML»:**

1. Цель работы: разобраться в устройстве языка управления данных DML.

**ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №15**

Схема базы данных состоит из четырех отношений:

**Автор** (id\_authors int, fio\_authors varchar(20))

**Издательство** (id\_publishers int, name varchar(20))

**Книга** (id\_books int, name varchar(50), year\_out varchar(4))

**Сводная** (id\_out int, id\_books int, id\_authors int, id\_publishers int)

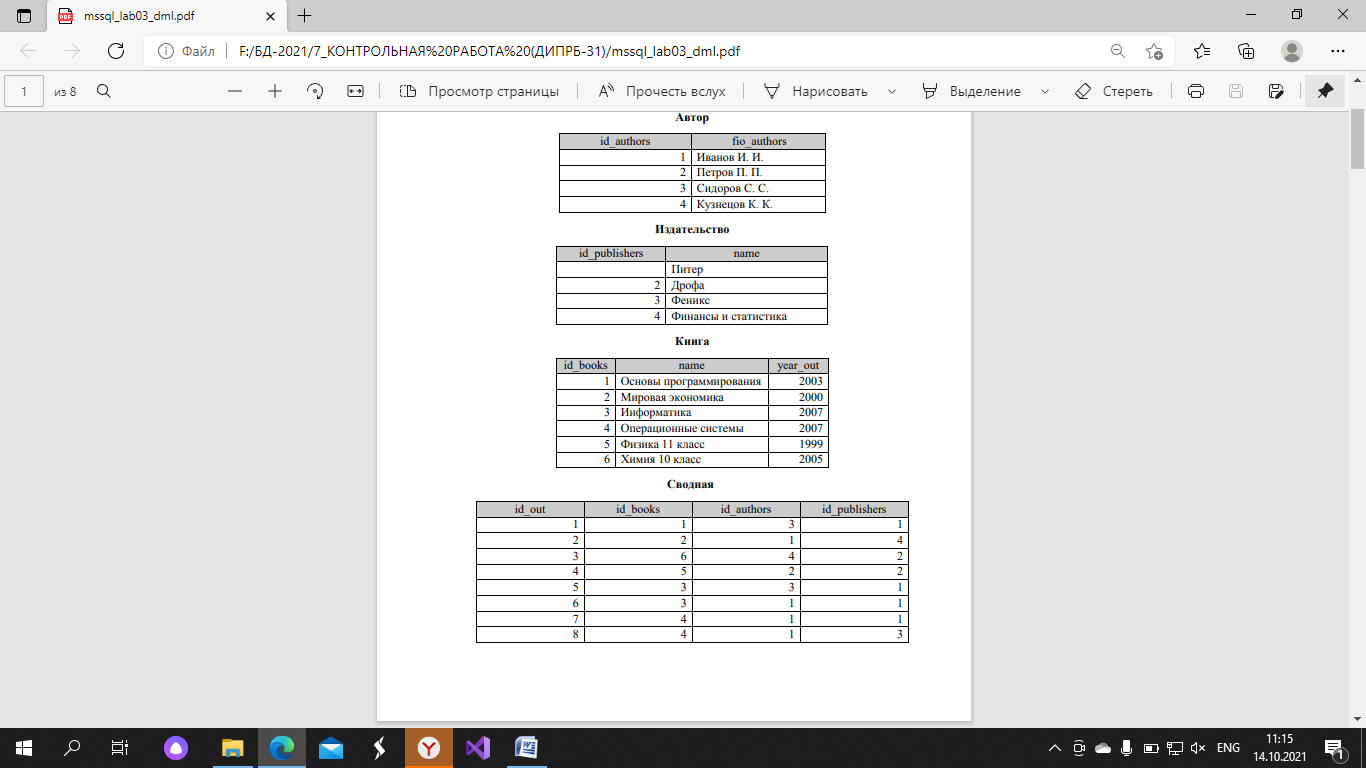


Схема базы данных представлена на рис. 1.

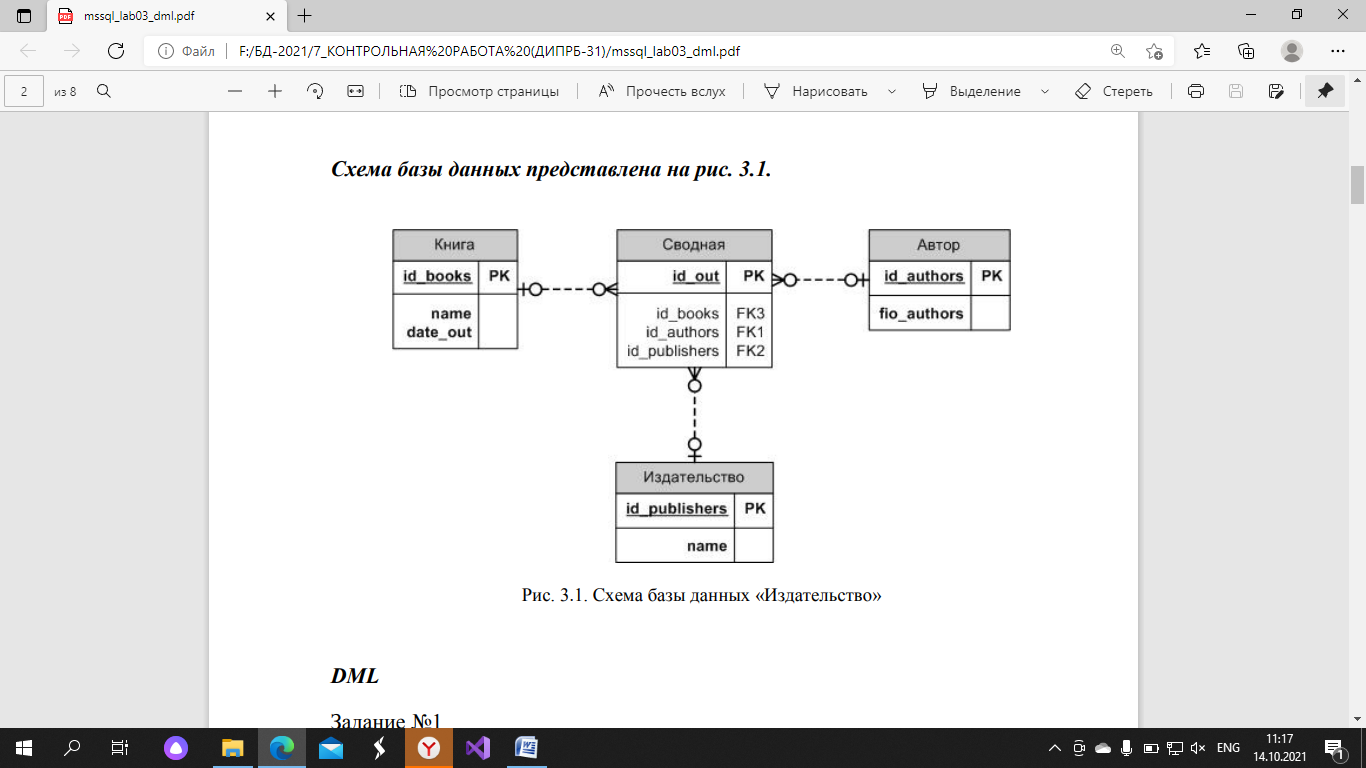


Рисунок 1 - Схема базы данных «Издательство»

***Задание №1.*** Создать и добавить в таблицы **Автор**, **Издательство**, **Книга**, **Сводная** данные, указанные в таблицах на странице 1.

Добавлять данные, используя конструкцию:

**insert into** *table\_name*(значения параметров по порядку через запятую)

CREATE TABLE [dbo].[Автор] (

[id\_authors] INT IDENTITY (1, 1) NOT NULL,

[fio\_authors] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([id\_authors] ASC),

);

CREATE TABLE [dbo].[Издательство] (

[id\_publishers] INT IDENTITY (1, 1) NOT NULL,

[name] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([id\_publishers] ASC),

);

CREATE TABLE [dbo].[Книга] (

[id\_books] INT IDENTITY (1, 1) NOT NULL,

[name] NVARCHAR (50) NULL,

[year\_out] INT,

PRIMARY KEY CLUSTERED ([id\_books] ASC),

);

CREATE TABLE [dbo].[Сводная] (

[id\_out] INT IDENTITY (1, 1) NOT NULL,

[id\_books] INT,

[id\_authors] INT,

[id\_publishers] INT,

PRIMARY KEY CLUSTERED ([id\_out] ASC),

FOREIGN KEY ([id\_authors]) REFERENCES [dbo].[Автор] ([id\_authors]),

FOREIGN KEY ([id\_publishers]) REFERENCES [dbo].[Издательство] ([id\_publishers]),

FOREIGN KEY ([id\_books]) REFERENCES [dbo].[Книга] ([id\_books]),

);

INSERT INTO [Автор] (fio\_authors) VALUES (N'Иванов');

INSERT INTO [Автор] (fio\_authors) VALUES (N'Петров');

INSERT INTO [Автор] (fio\_authors) VALUES (N'Сидоров');

INSERT INTO [Автор] (fio\_authors) VALUES (N'Кузнецов');

INSERT INTO [Издательство] ([name]) VALUES (N'Питер');

INSERT INTO [Издательство] ([name]) VALUES (N'Дрофа');

INSERT INTO [Издательство] ([name]) VALUES (N'Феникс');

INSERT INTO [Издательство] ([name]) VALUES (N'Финансы и статистика');

INSERT INTO [Книга] ([name], year\_out) VALUES (N'Основы программирования', 2003);

INSERT INTO [Книга] ([name], year\_out) VALUES (N'Мировая экономика', 2000);

INSERT INTO [Книга] ([name], year\_out) VALUES (N'Информатика', 2007);

INSERT INTO [Книга] ([name], year\_out) VALUES (N'Операционная система', 2008);

INSERT INTO [Книга] ([name], year\_out) VALUES (N'Физика 11 класс', 1999);

INSERT INTO [Книга] ([name], year\_out) VALUES (N'Химия 10 класс', 2005);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (1, 3, 1);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (2, 1, 4);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (6, 4, 2);

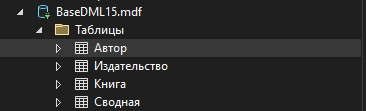
INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (5, 2, 2);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (3, 3, 1);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (3, 1, 1);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (4, 1, 1);

INSERT INTO [Сводная] ( id\_books, id\_authors, id\_publishers) VALUES (4, 1, 3);



***Задание №2***. Добавить в таблицу **Книга** запись (значение поля **name**=‟SQL Server‟, **year\_out**=2019) используя в конструкции **insert into** *table\_name* конструкцию **select**:

**insert into** *table\_name* **select** *@var* **as** *name\_column*

INSERT INTO Книга (name, year\_out) SELECT 'SQL Server', 2019;



***Задание №3.*** Добавить в таблицу **Автор** три записи, используя один оператор **insert into table\_name**. (значения поля **name**: „Андреев А.А.‟, „Киреев К.К.‟, „Максимов М.М.‟).

INSERT INTO Автор ([fio\_authors]) VALUES (N'Андреев А.А.'), (N'Киреев К.К.'), (N'Максимов М.М.');



***Задание №4.*** Добавить в таблицу **Книга** запись (значение поля **name**=‟MySQL‟, **year\_out**= значение по умолчанию).

INSERT INTO Книга (name) VALUES ('MySQL');



***Задание №5.*** Добавить в таблицу **Сводная** запись со следующими параметрами:

- **id\_books** = максимальному в таблице **Книга**;

- **id\_authors** = минимальному в таблице Автор;

- **id\_publishers** = среднему значению в таблице **Издательство** (отбросить дробную часть).

INSERT INTO Сводная (id\_books, id\_authors, id\_publishers)

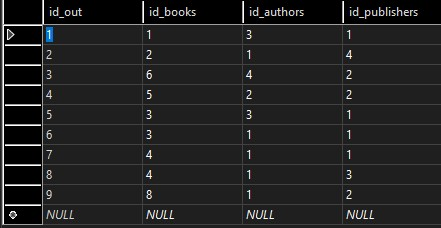
VALUES (

(SELECT MAX(id\_books) FROM Книга),

(SELECT MIN(id\_authors) FROM Автор),

(SELECT ROUND(AVG(id\_publishers),0) FROM Издательство)

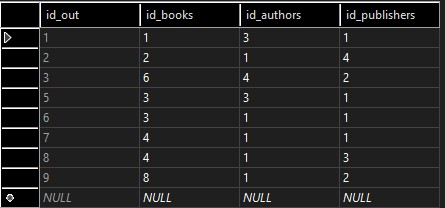
);



***Задание №6.*** Удалить из таблицы **Сводная** все записи о книгах с минимальным годом выпуска используя конструкцию:

**delete from** *table\_name* **where** *условие*

DELETE FROM Сводная WHERE id\_books IN (SELECT id\_books FROM Книга WHERE year\_out = (SELECT MIN(year\_out) FROM Книга));

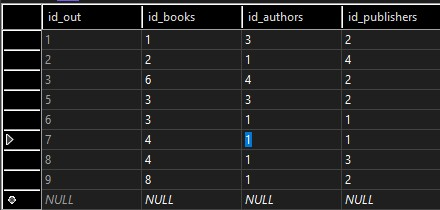


***Задание №7.*** Изменить таблицу **Сводная** так, чтобы все книги написанные автором, код у которого равен **3**, печатались в издательстве “**Дрофа**”.

Используйте конструкцию:

**update** *table\_name* **set** *name\_column=@var*

UPDATE Сводная   
SET id\_publishers = (SELECT id\_publishers FROM Издательство WHERE [name] = 'Дрофа')   
WHERE id\_authors = 3;



***Задание №8.*** Добавить к году выпуска книги в таблице **Книга** единицу, но только для тех книг, вторая буква в названии которых ‘**и**’.

UPDATE Книга

SET year\_out = CAST(CAST(year\_out AS int) + 1 AS varchar(4))

WHERE SUBSTRING([name], 2, 1) = N'и';



***Задание №9.*** Найти количество книг с максимальным годом выпуска.

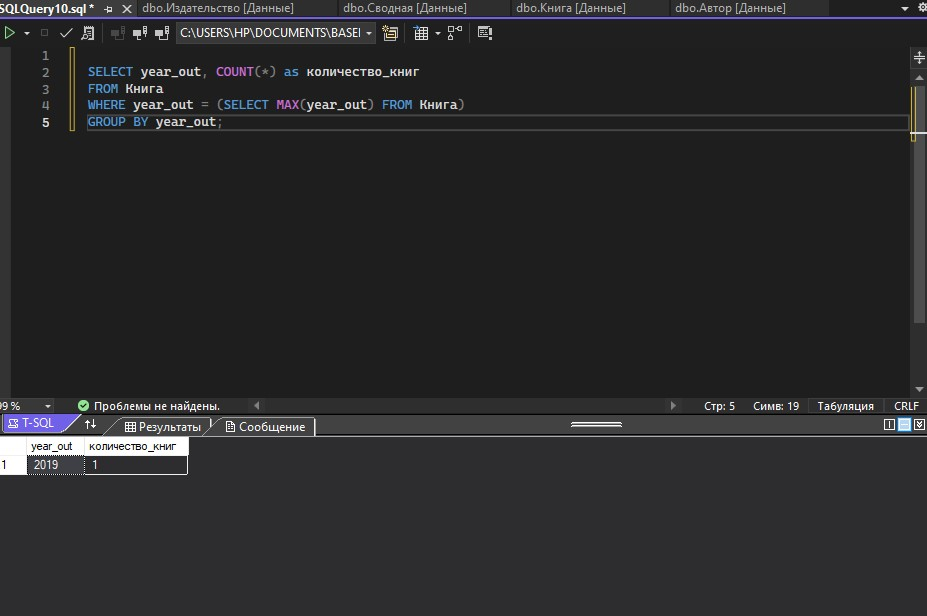
***Вывод***: *год выпуска, количество книг*

SELECT year\_out, COUNT(\*) as количество\_книг

FROM Книга

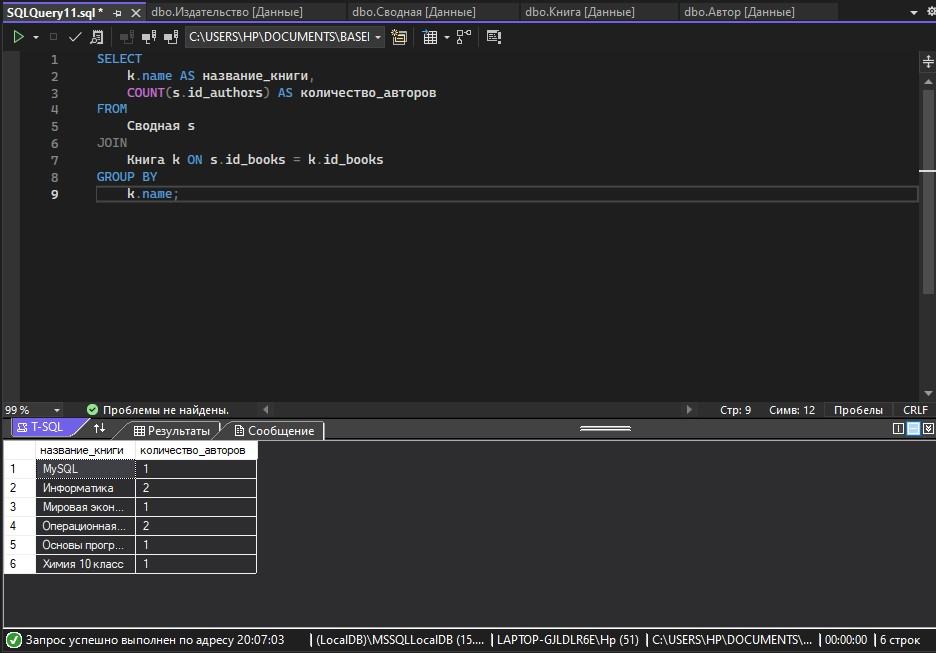
WHERE year\_out = (SELECT MAX(year\_out) FROM Книга)

GROUP BY year\_out;



***Задание №10.*** Найти количество авторов у каждой книги.

***Вывод***: *название книги, количество авторов*



SELECT

k.name AS название\_книги,

COUNT(s.id\_authors) AS количество\_авторов

FROM

Сводная s

JOIN

Книга k ON s.id\_books = k.id\_books

GROUP BY

k.name;

***Задание №11***. Найти только те книги, которые написали два автора.

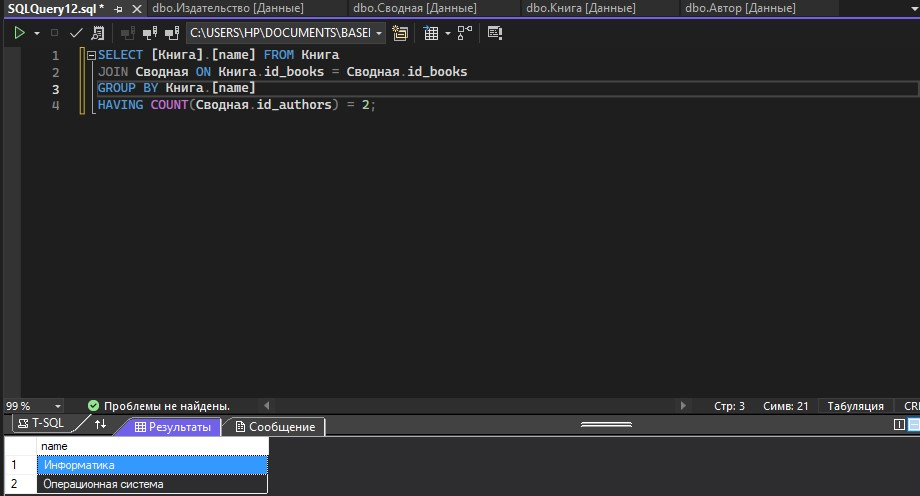
***Вывод***: *название книги*

SELECT [Книга].[name] FROM Книга

JOIN Сводная ON Книга.id\_books = Сводная.id\_books

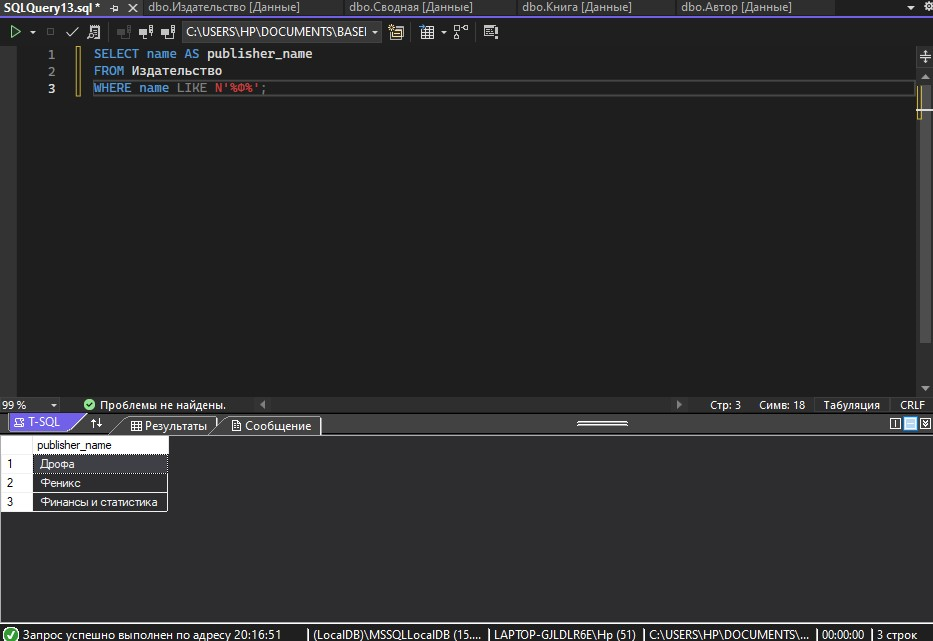
GROUP BY Книга.[name]

HAVING COUNT(Сводная.id\_authors) = 2;



***Задание №12***. Найти только те издательства, в названиях которых есть буква „**ф**‟.

***Вывод*:** *название издательства*

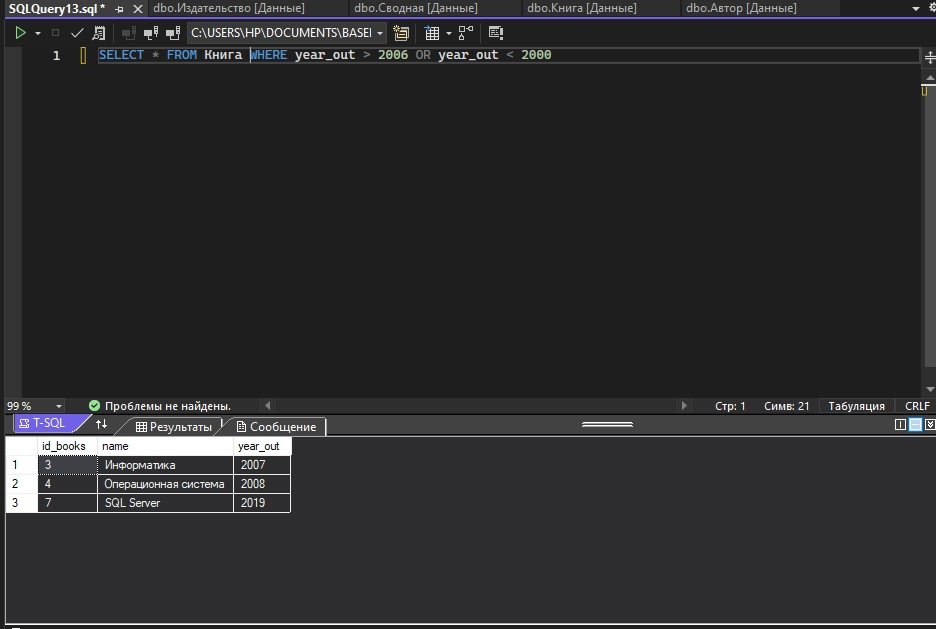


SELECT name AS publisher\_name

FROM Издательство

WHERE name LIKE N'%Ф%';

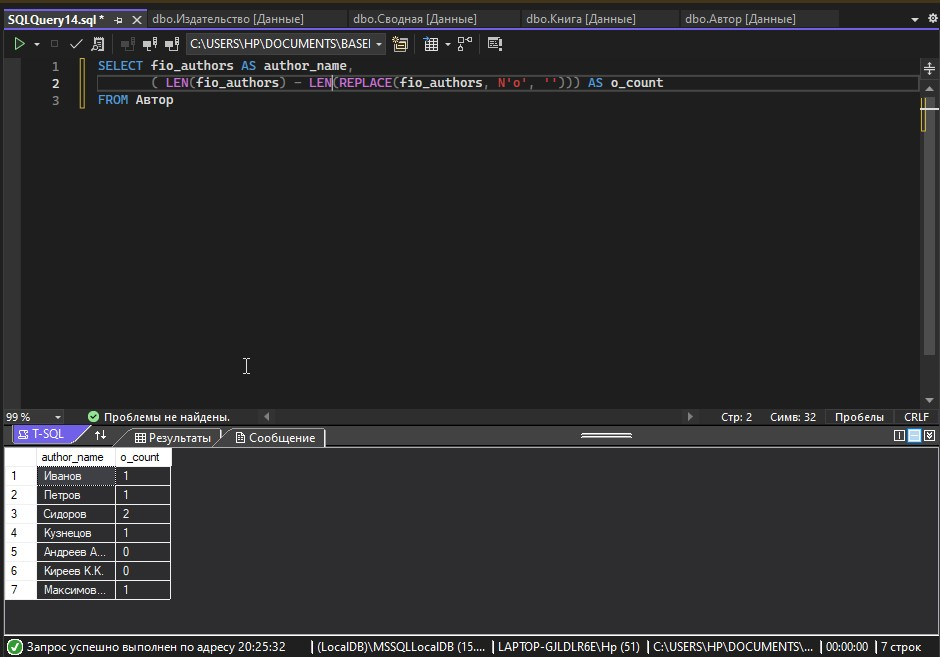
***Задание №13.*** Найти книги, выпущенные **до 2000** года и книги выпущенные **после 2006**.



SELECT \* FROM Книга WHERE year\_out > 2006 OR year\_out < 2000

***Задание №14.***  Найти количество букв “**о**” в фамилии и инициалах каждого автора.

***Вывод*:** *фамилия и инициалы, количество букв “о”*



SELECT fio\_authors AS author\_name,

( LEN(fio\_authors) - LEN(REPLACE(fio\_authors, N'о', ''))) AS o\_count

FROM Автор

***Задание №15.*** Найти тех авторов, которые написали не менее двух книг.

***Вывод*:** *фамилия и инициалы, количество книг*

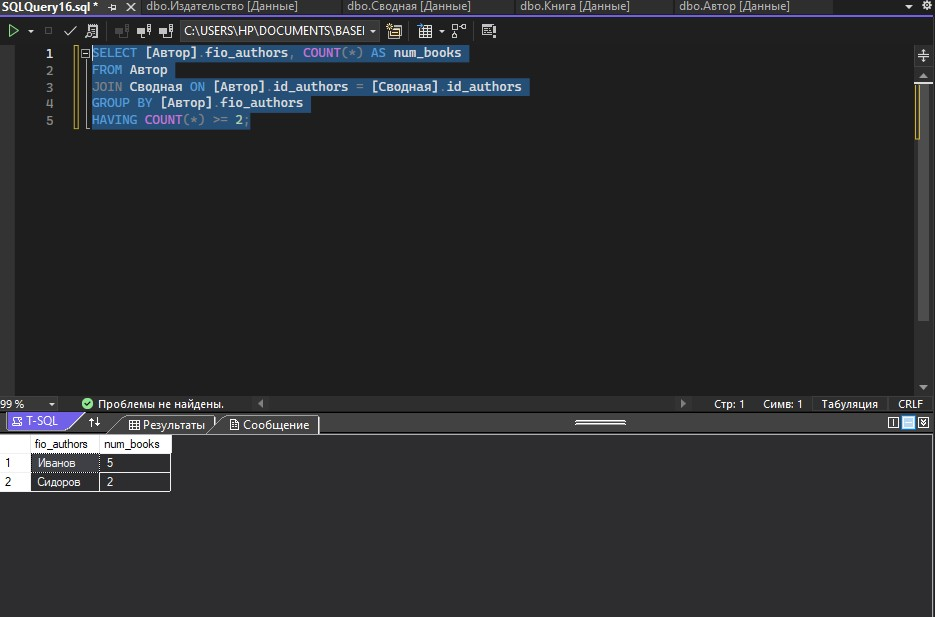
SELECT [Автор].fio\_authors, COUNT(\*) AS num\_books

FROM Автор

JOIN Сводная ON [Автор].id\_authors = [Сводная].id\_authors

GROUP BY [Автор].fio\_authors

HAVING COUNT(\*) >= 2;



***Задание №16.*** Найти сколько раз встречается каждая буква алфавита в названиях книг.

***Вывод***: *буква, количество повторений*

SELECT SUBSTRING(name, N, 1) AS letter, COUNT(\*) AS repetitions

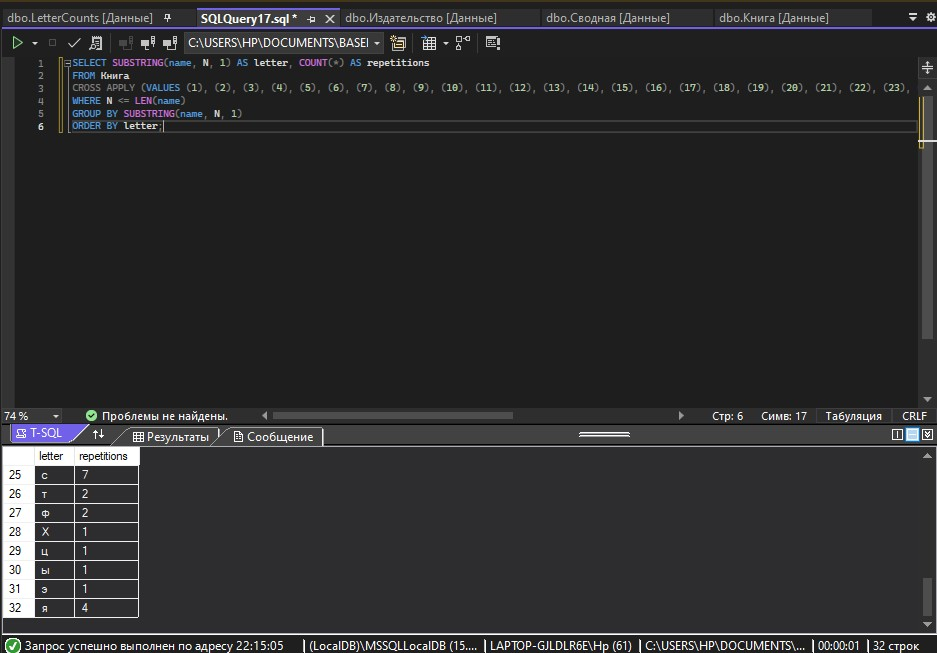
FROM Книга

CROSS APPLY (VALUES (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25), (26), (27), (28) ,(29), (30), (31), (32), (33)) AS Numbers(N)

WHERE N <= LEN(name)

GROUP BY SUBSTRING(name, N, 1)

ORDER BY letter;



1. Выводы

Вывод: в ходе выполнения лабораторной работе я изучил основы запросов DML.