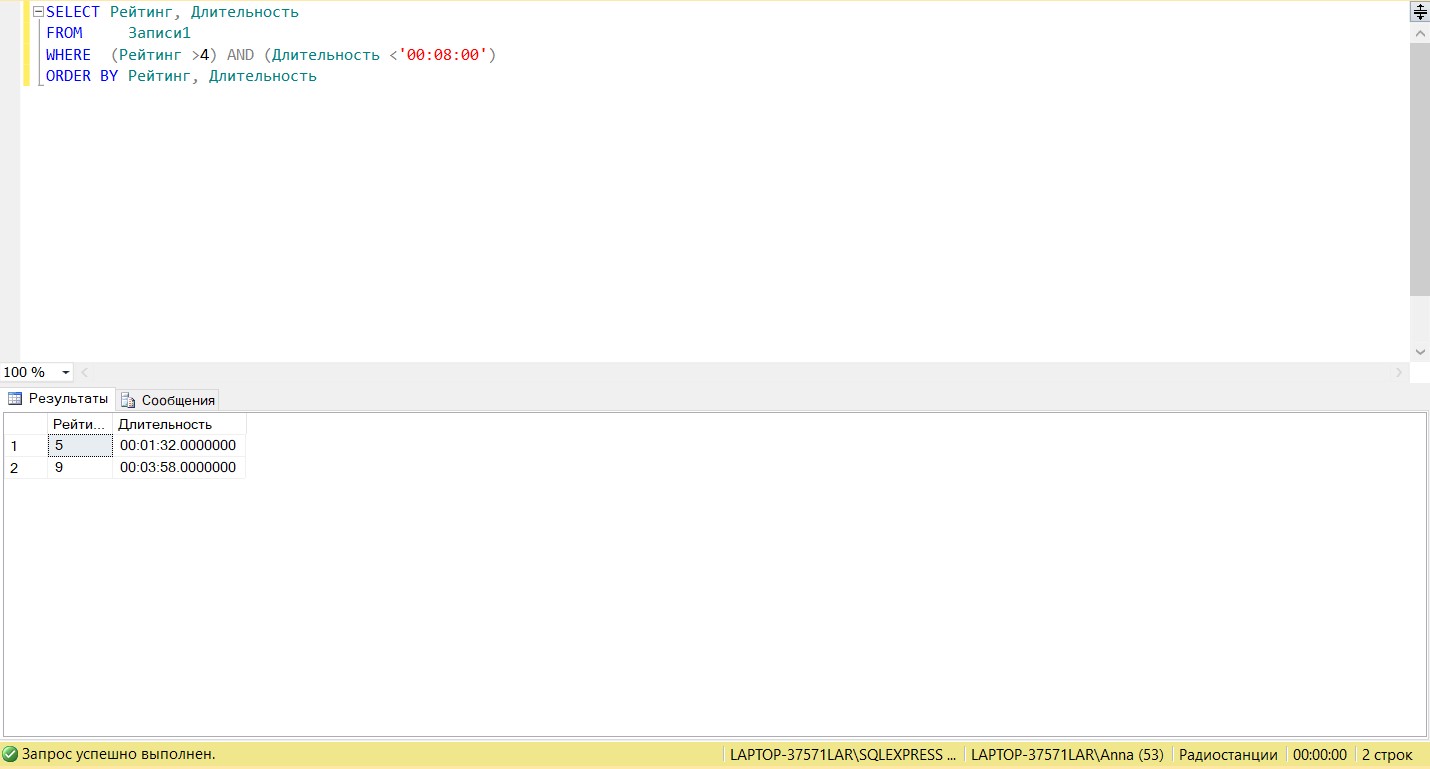
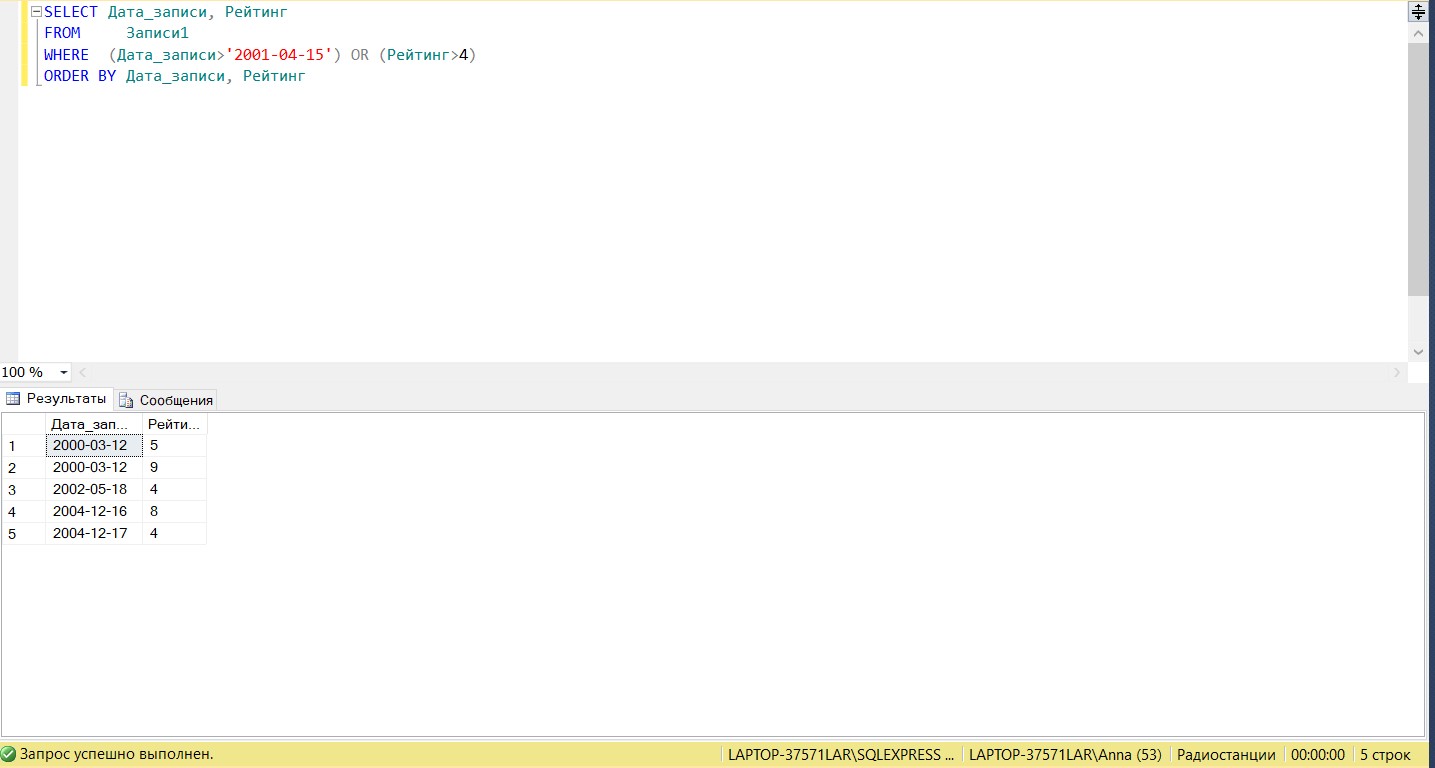
Задание 1.

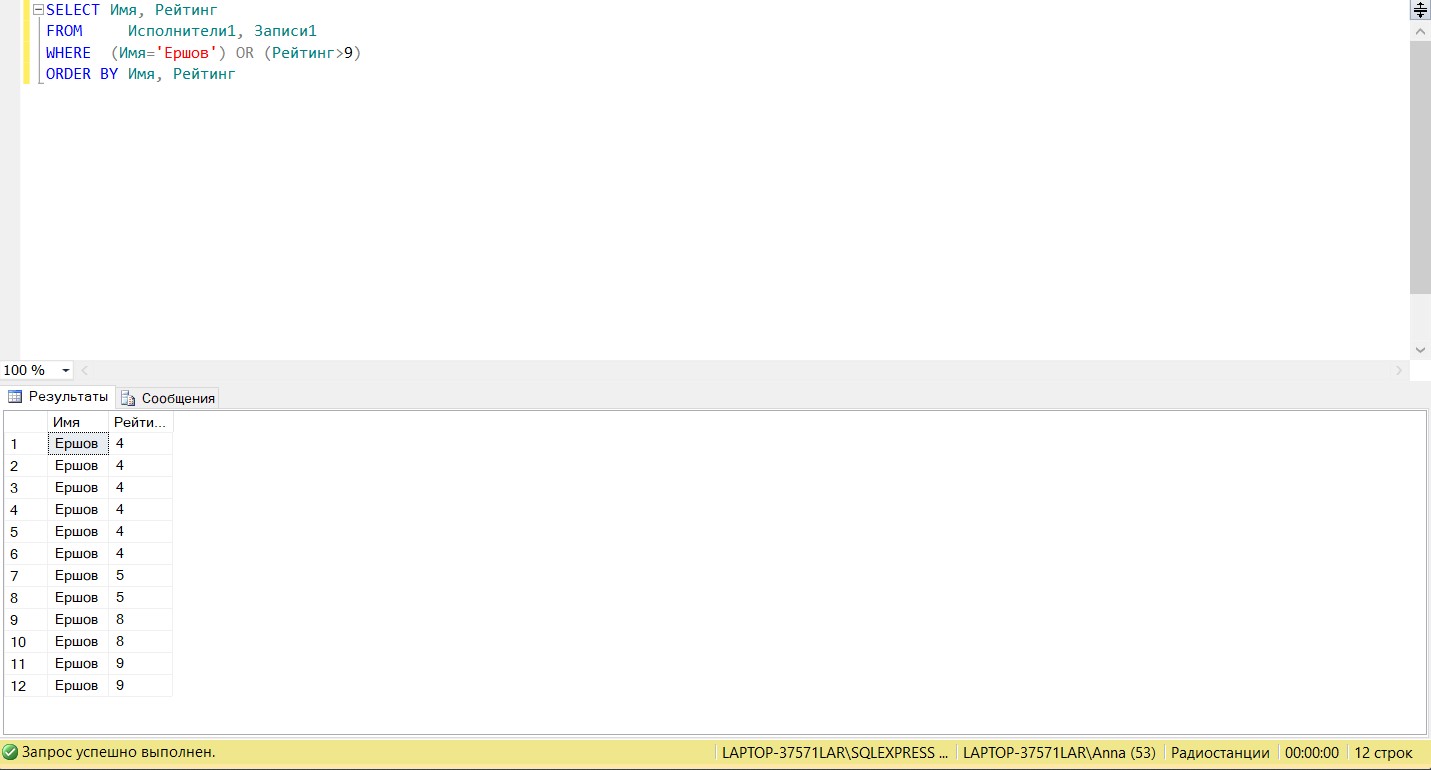
1.1a.



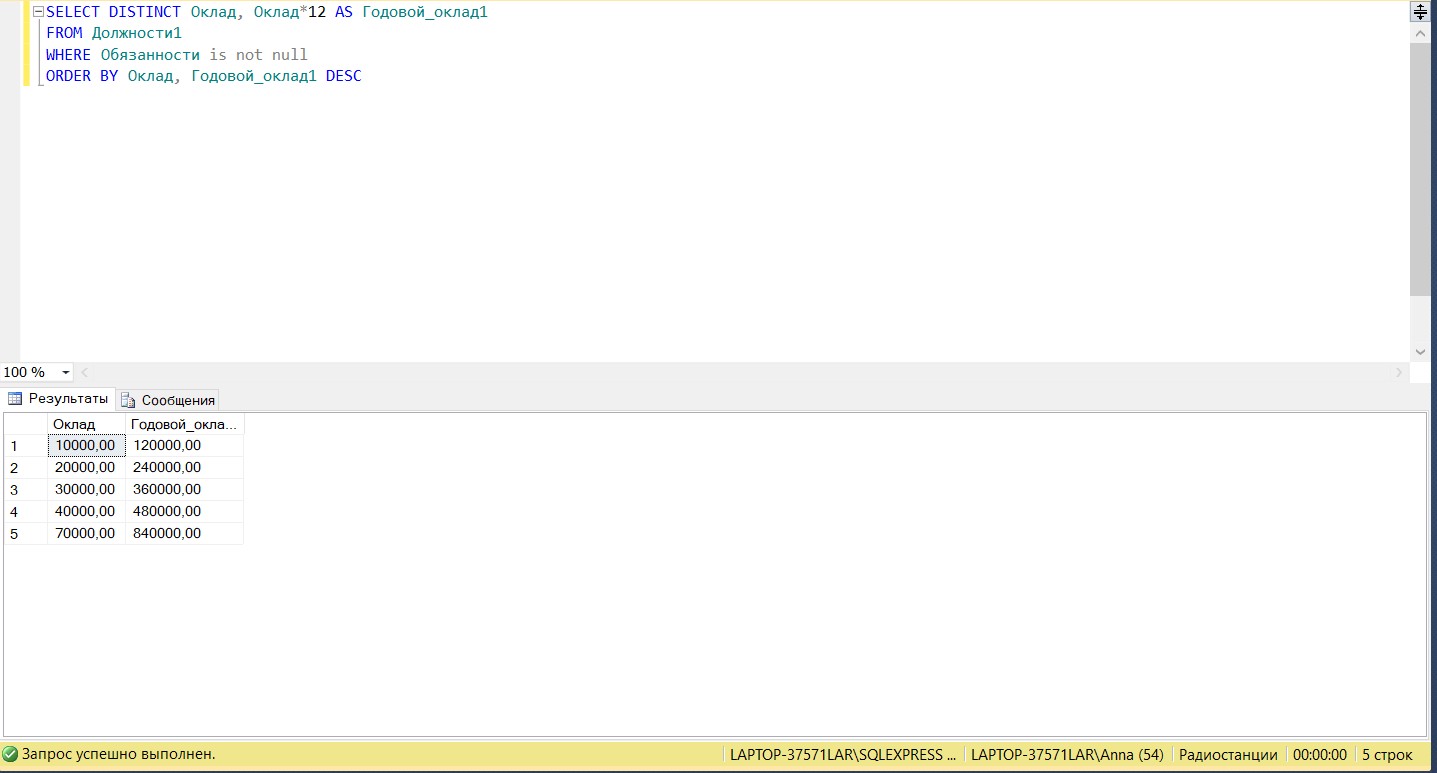
1.1b.



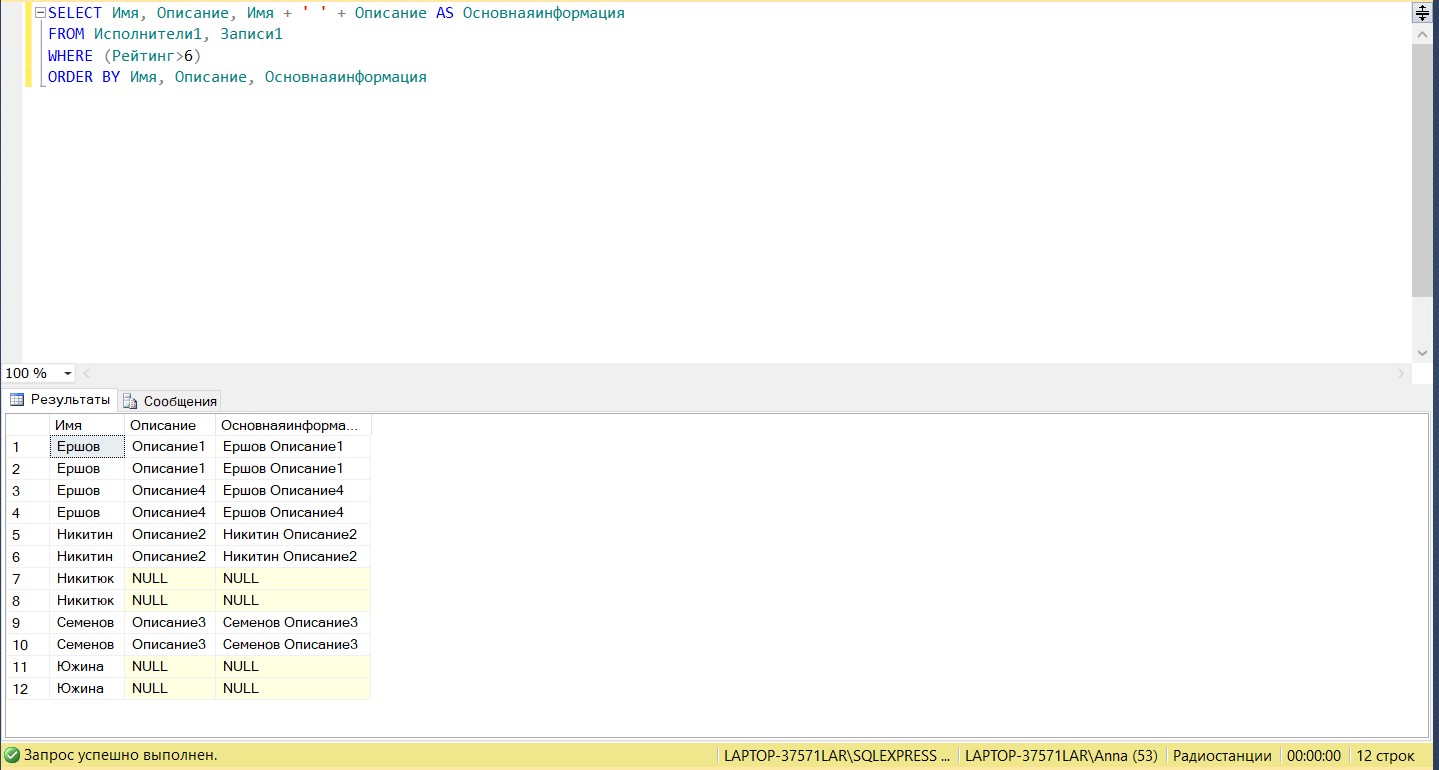
1.1c.



**1.2.(a)**



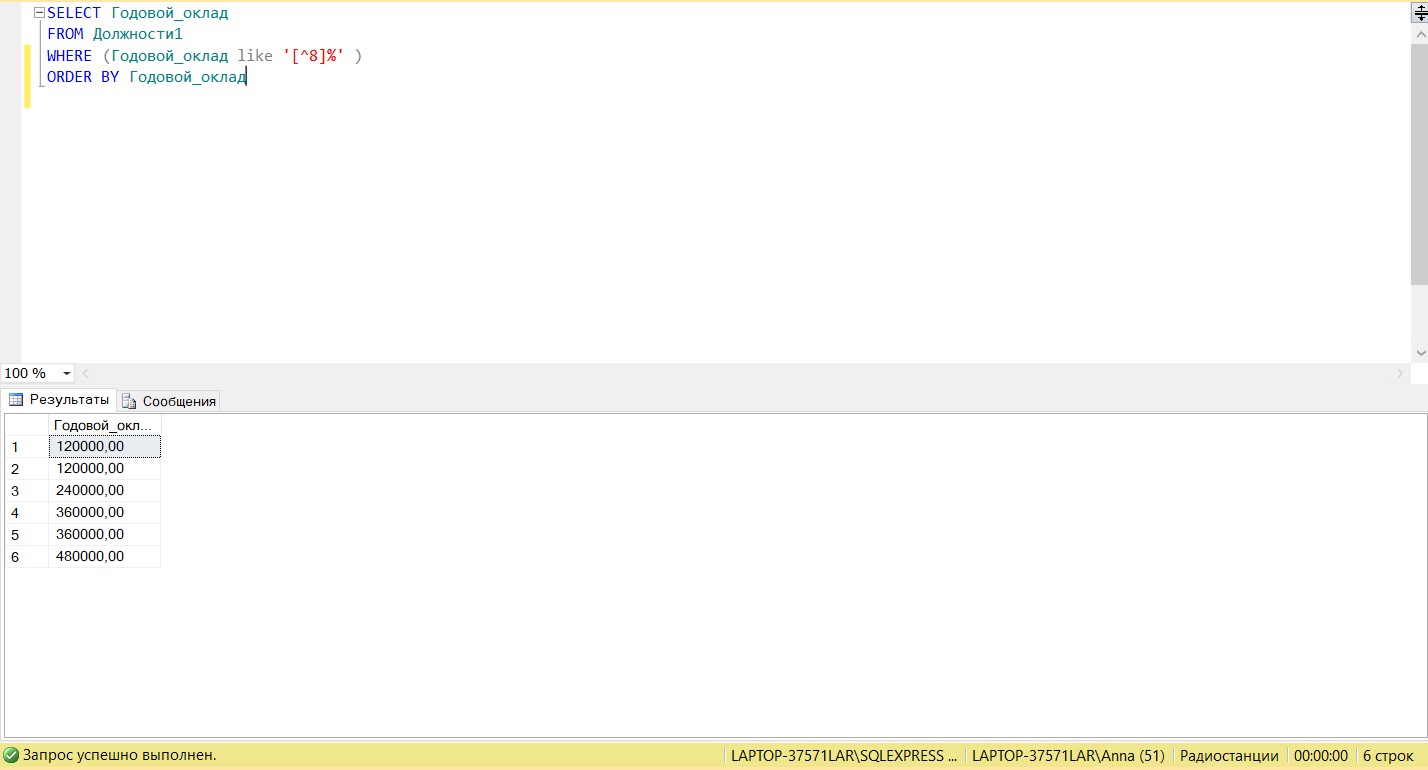
**1.2.(b)**



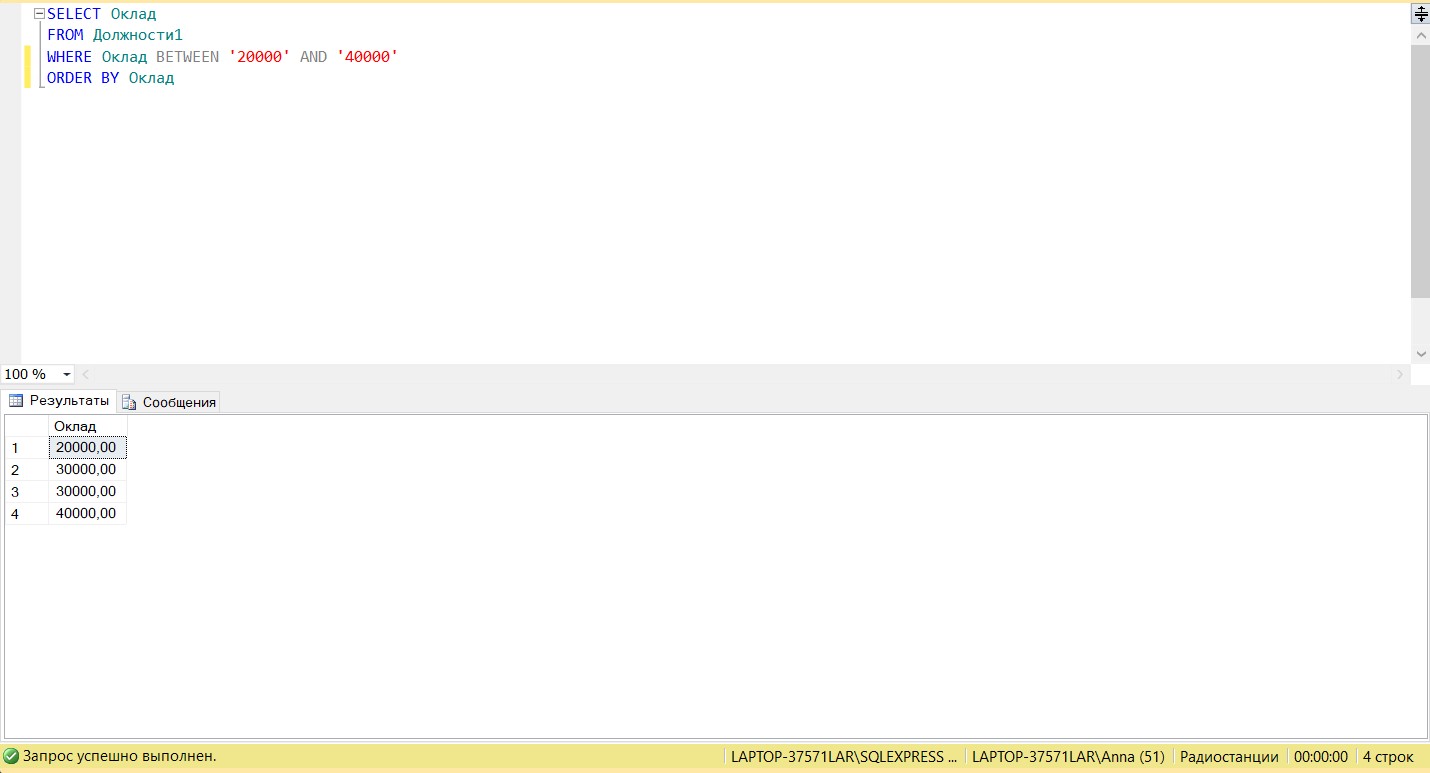
**1.3.(a)**



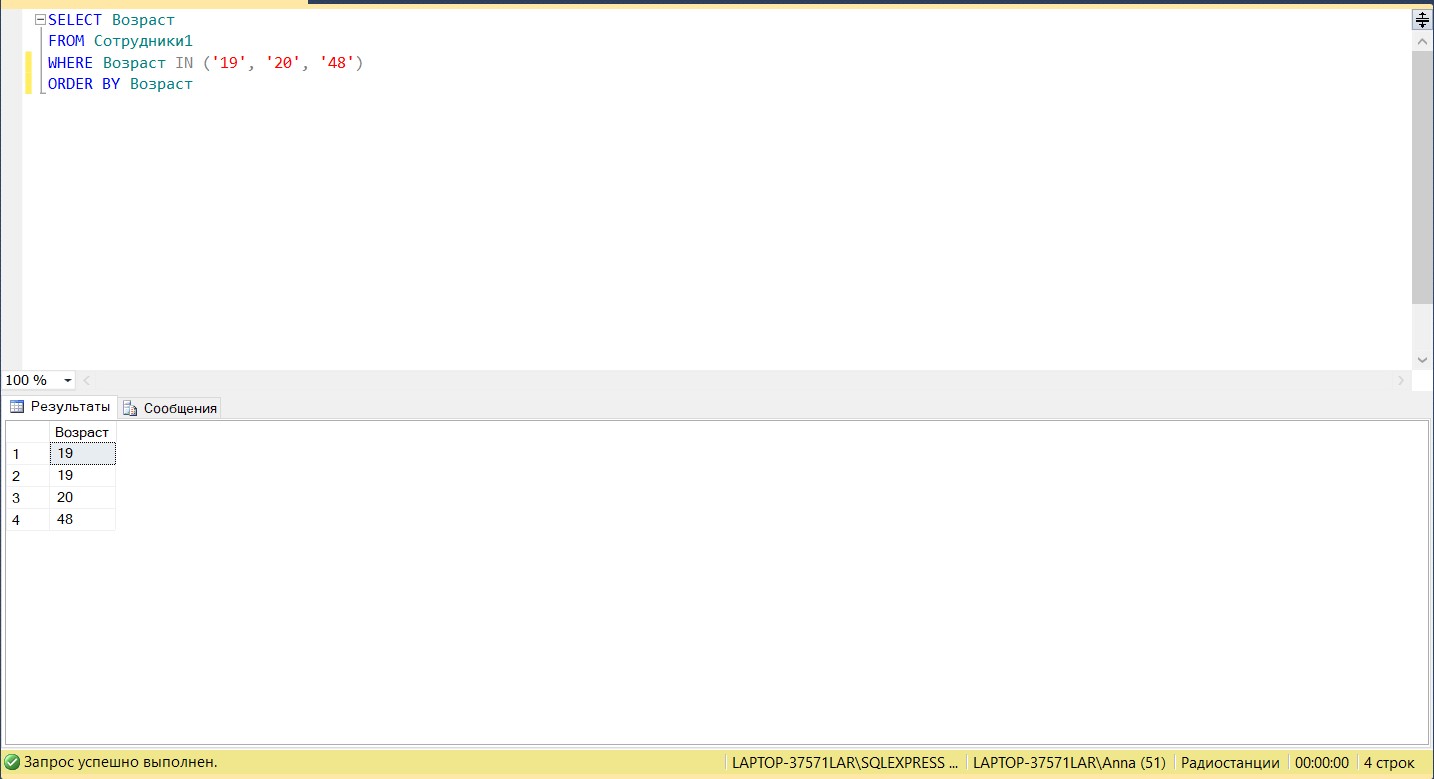
**1.3.(b)**



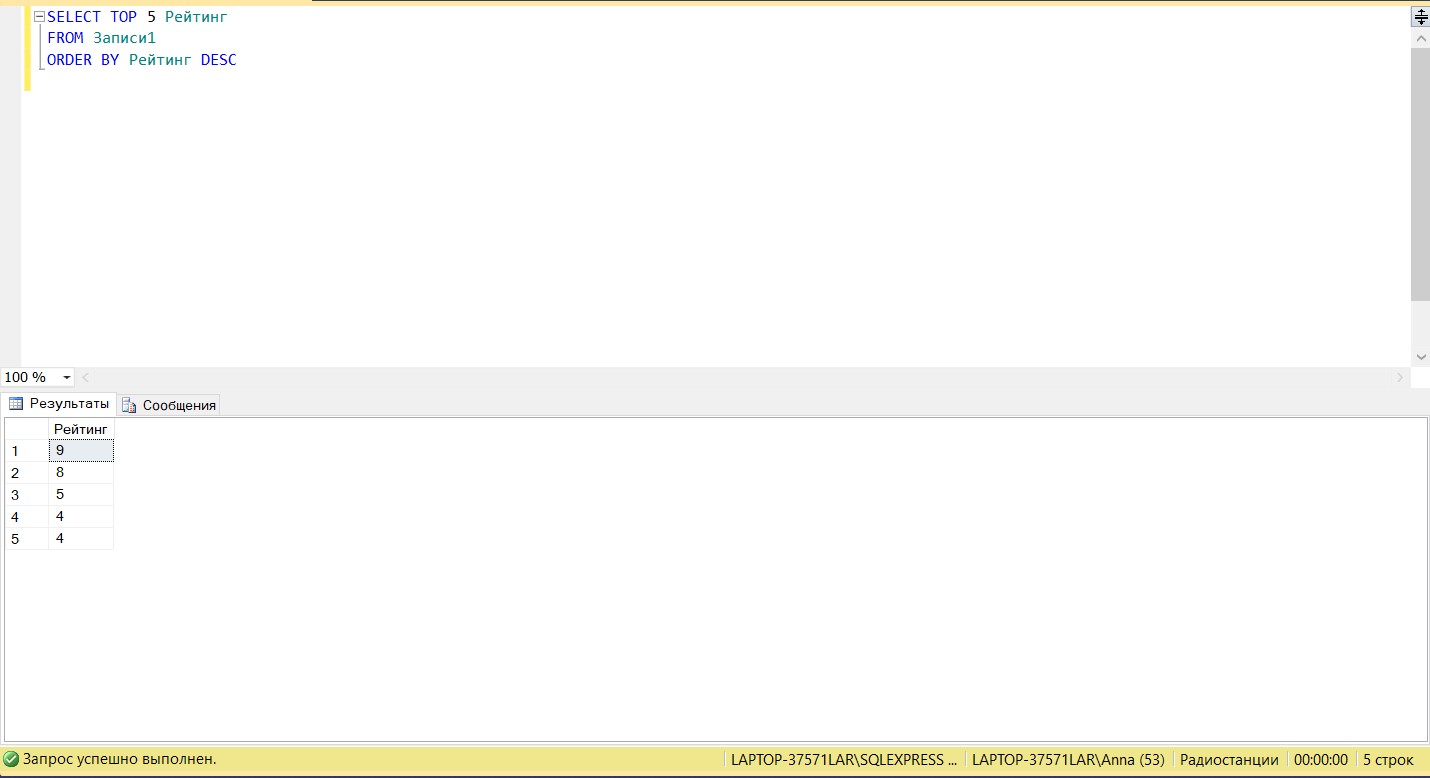
**1.4.**



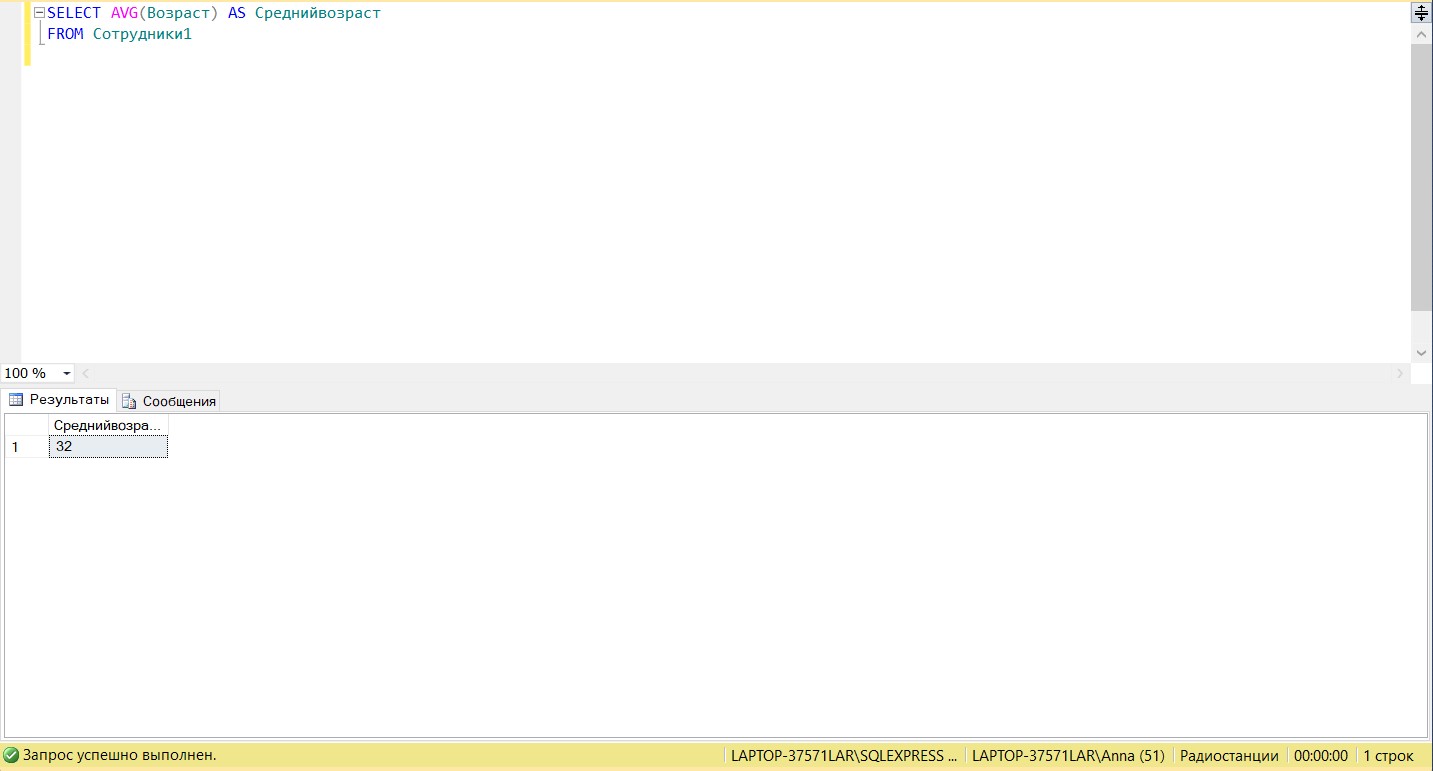
**1.5.**



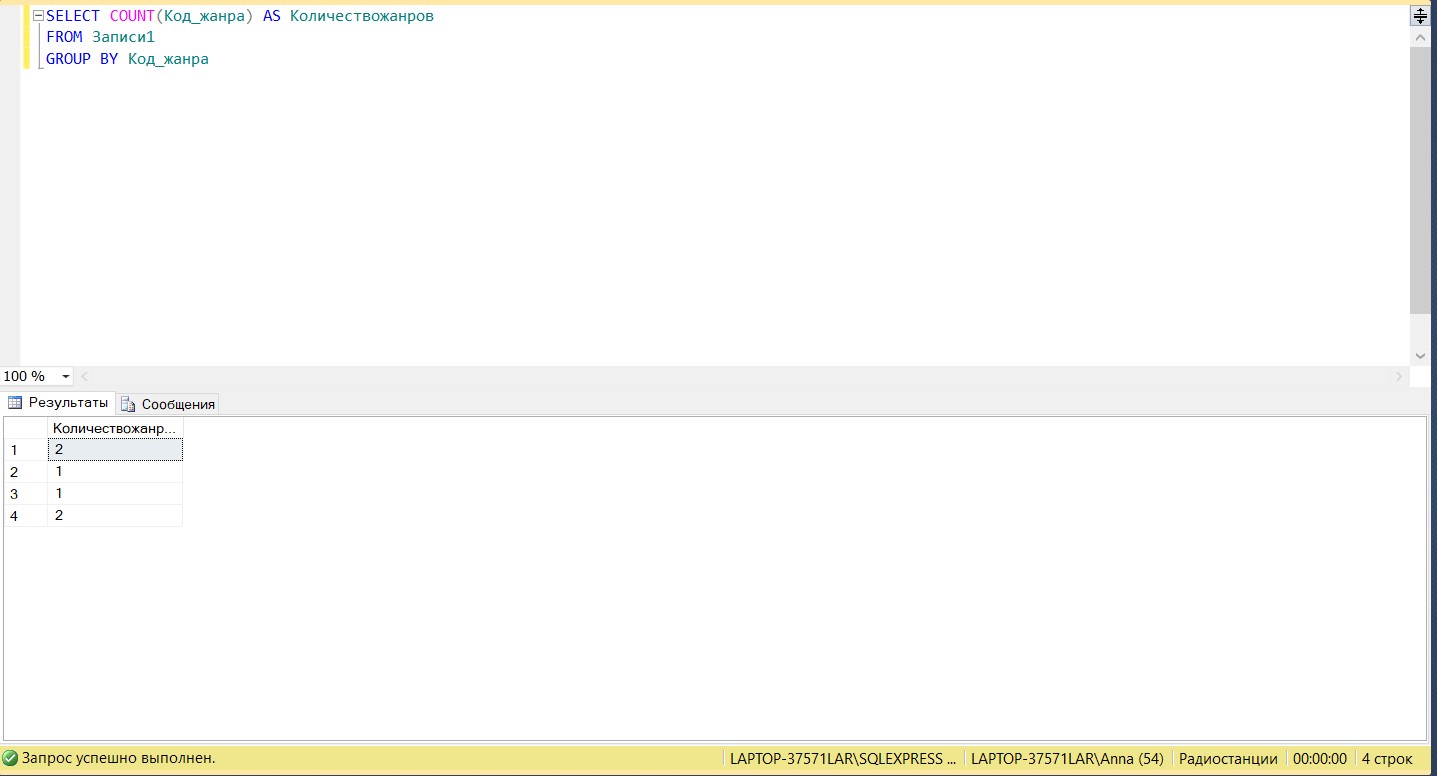
**2**



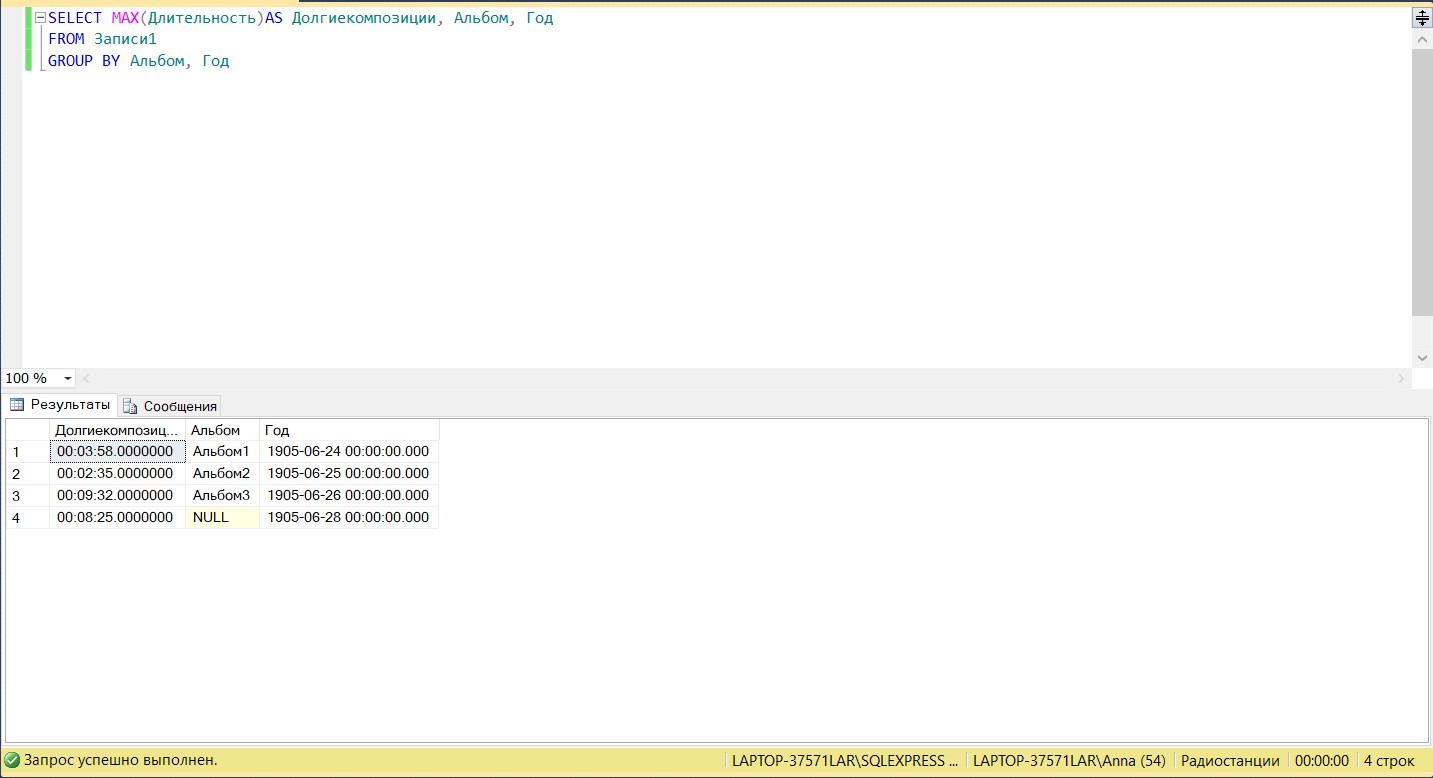
**3**



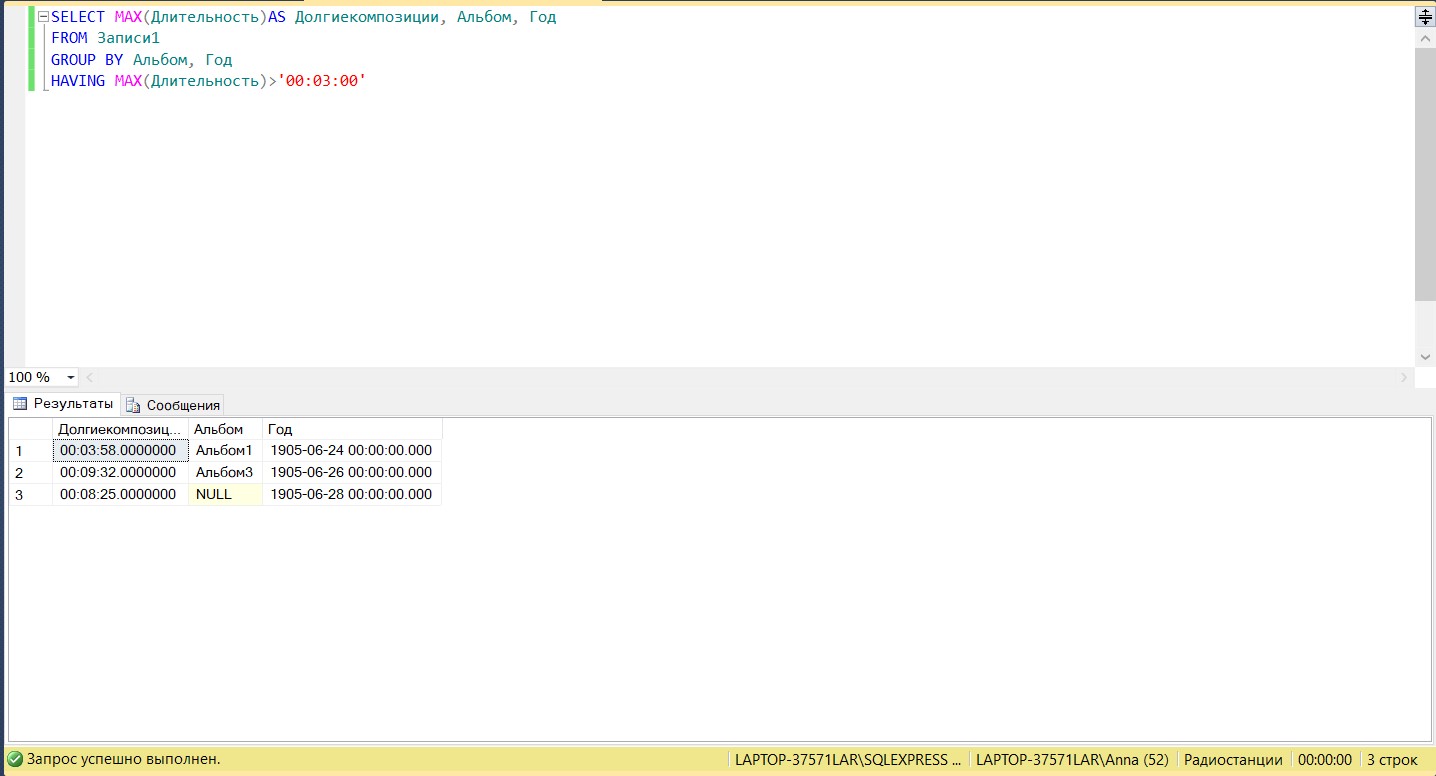
**4**



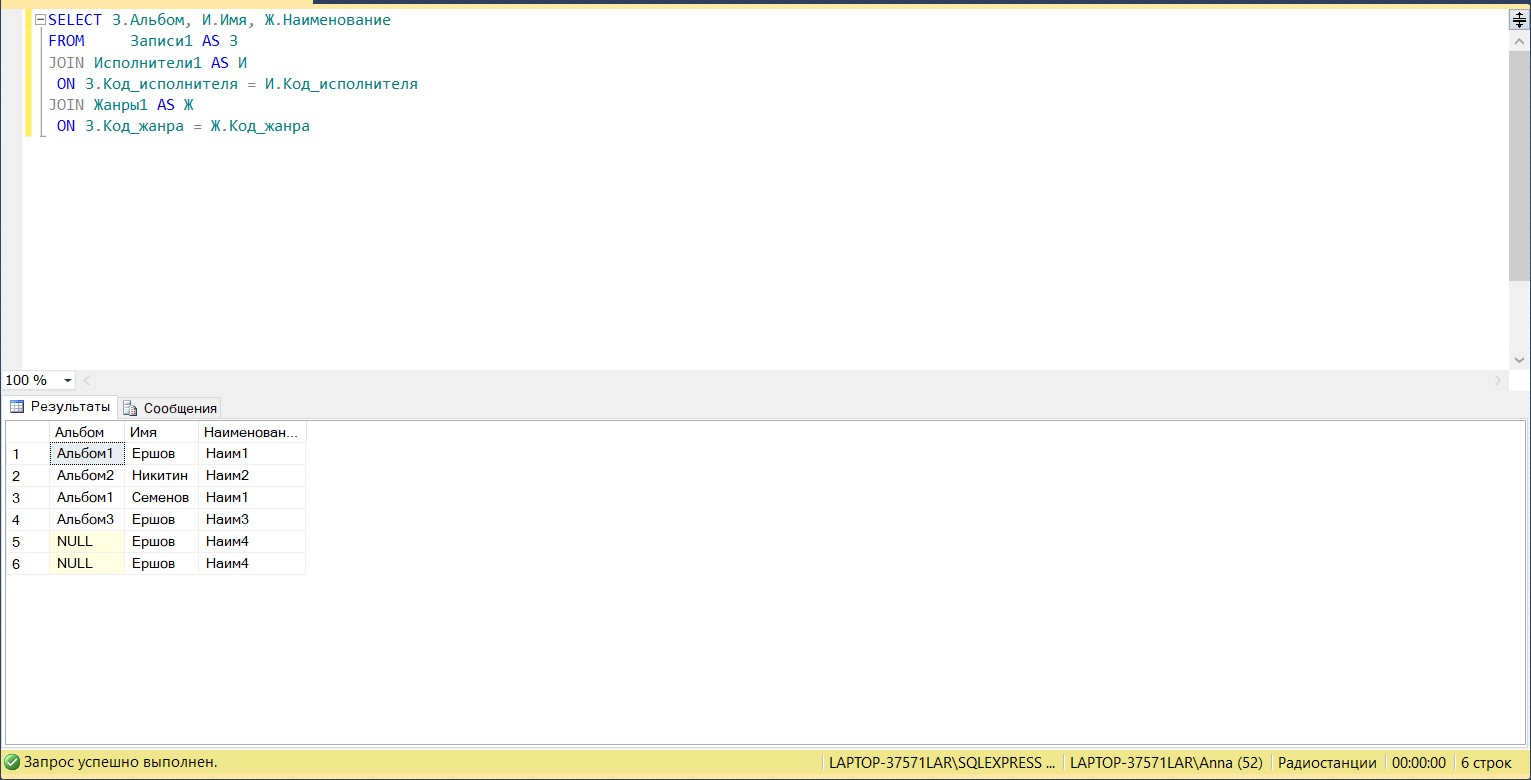
**5.**



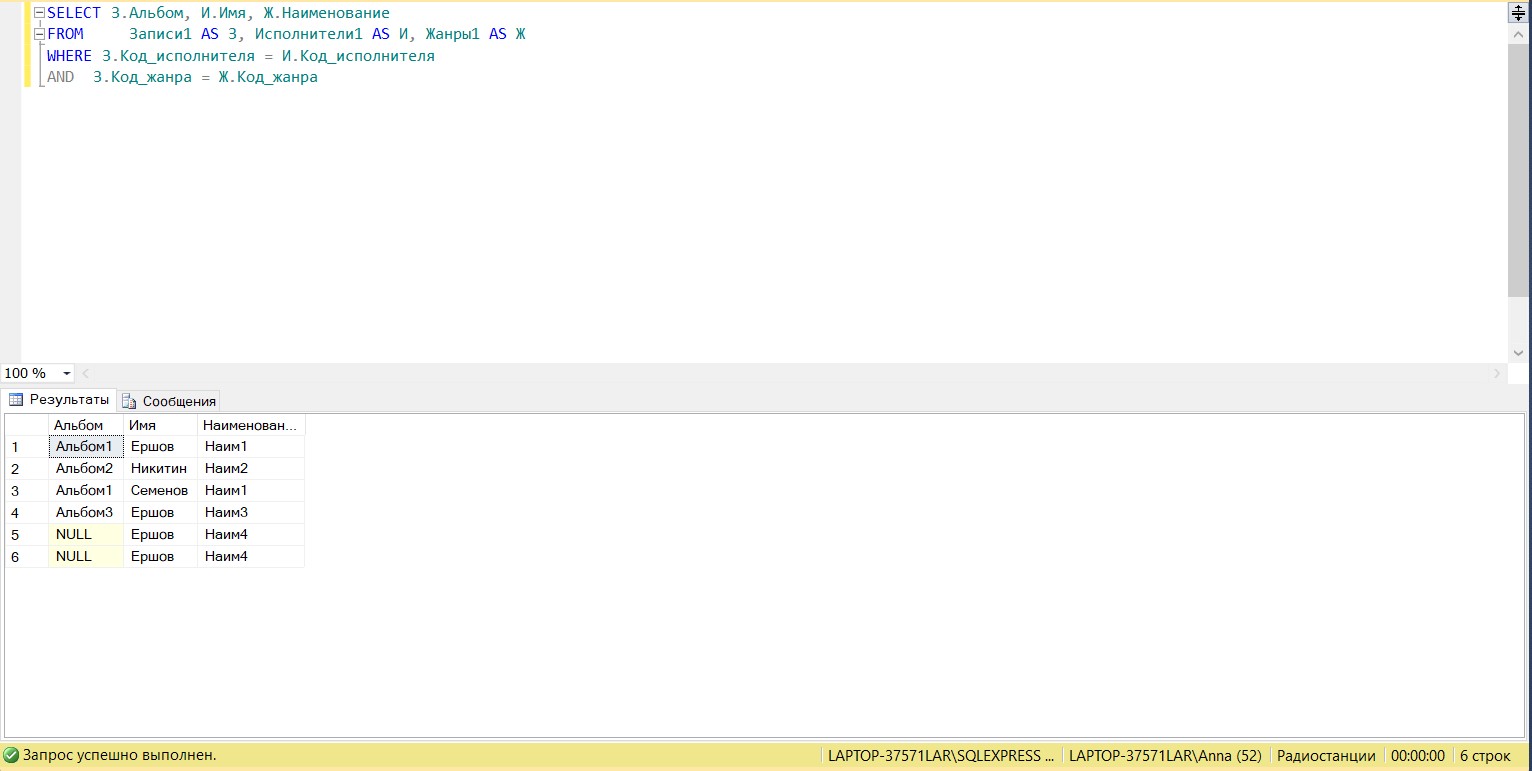
**6.**



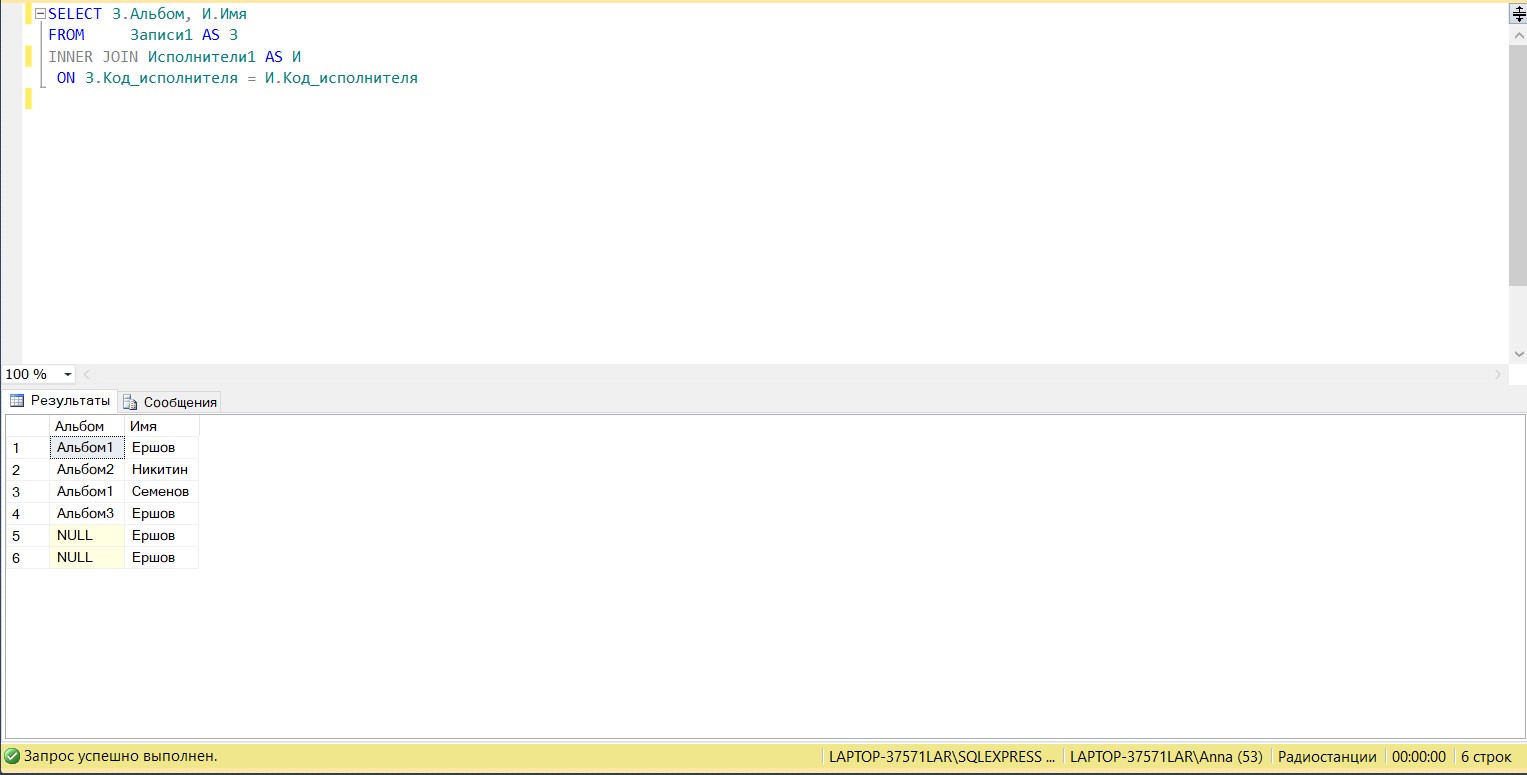
**7 (a).**



**7(b)**



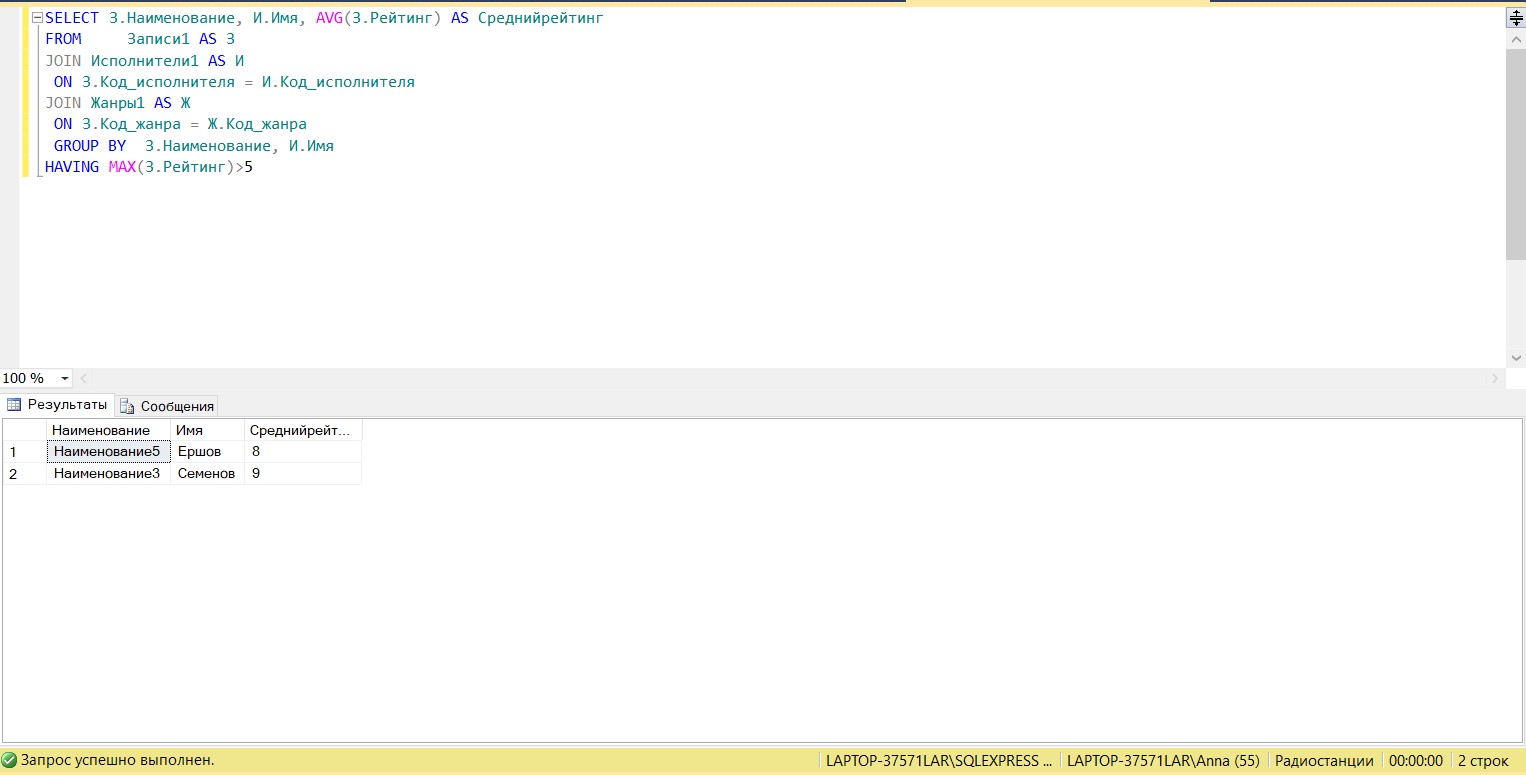
**8.**



**9.**



**10.**



**11.**

USE Радиостанции;

IF OBJECT\_ID (N'dbo.Датыпремьеры1', N'U') IS NOT NULL

DROP TABLE dbo.[Датыпремьеры1];

GO

CREATE TABLE dbo.Датыпремьеры1

(

Код\_записи int NOT NULL,

Номерисполнениявэфире int NOT NULL,

Дата\_премьеры DATETIME NOT NULL,

CONSTRAINT FK\_Датыпремьеры1\_Записи1 FOREIGN KEY(Код\_записи, Номерисполнениявэфире)

REFERENCES dbo.Записи1 (Код\_записи, Номерисполнениявэфире)

)

DELETE FROM Датыпремьеры1

INSERT INTO dbo.Датыпремьеры1

Values

(1,1,'13.03.2000'),

(2,3,'15.04.2001'),

(3,2,'13.03.2000'),

(4,4,'18.05.2002'),

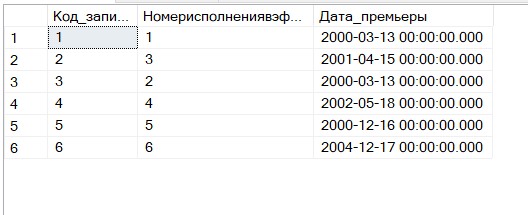
(5,5,'16.12.2000'),

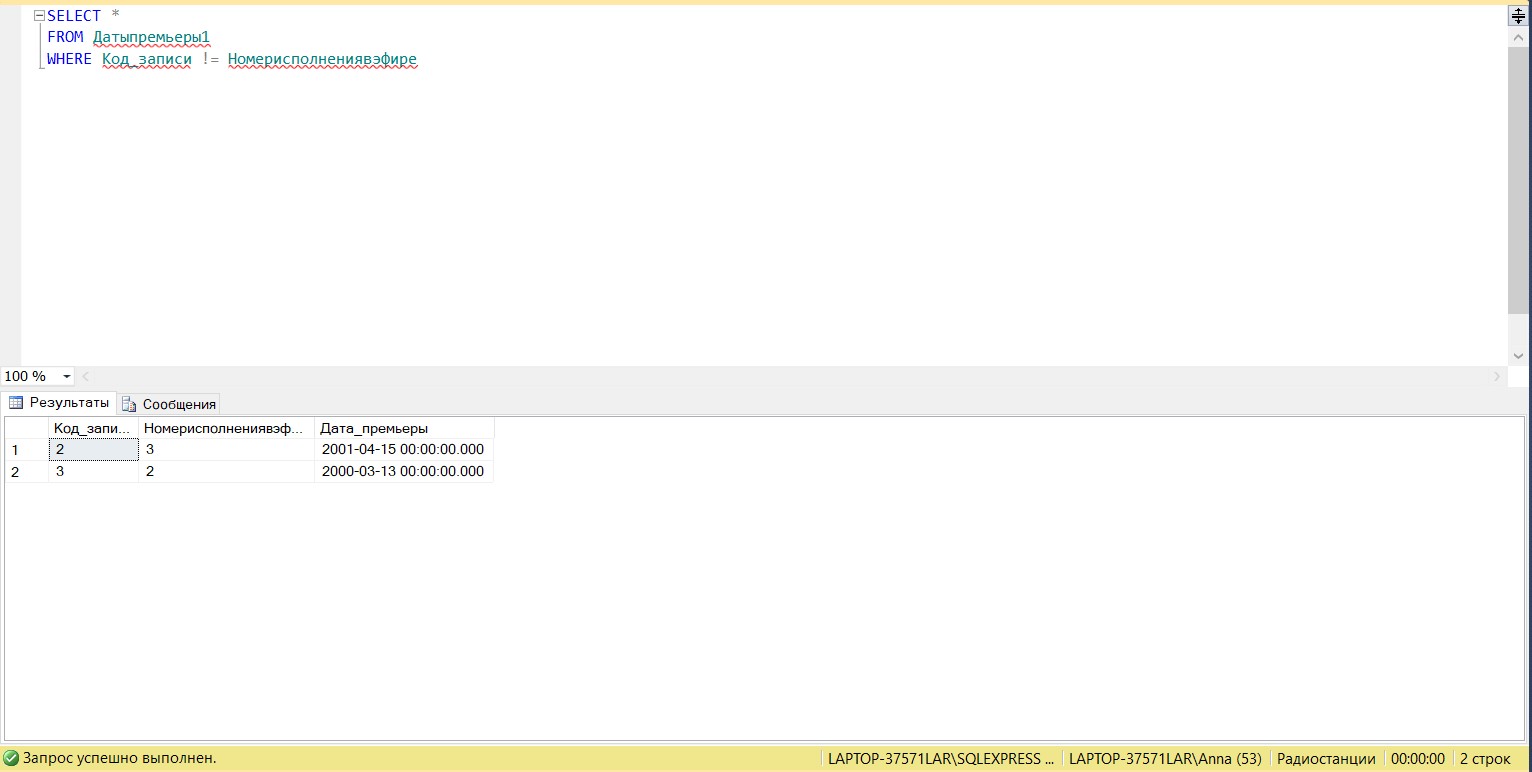
(6,6,'17.12.2004')

SELECT \*

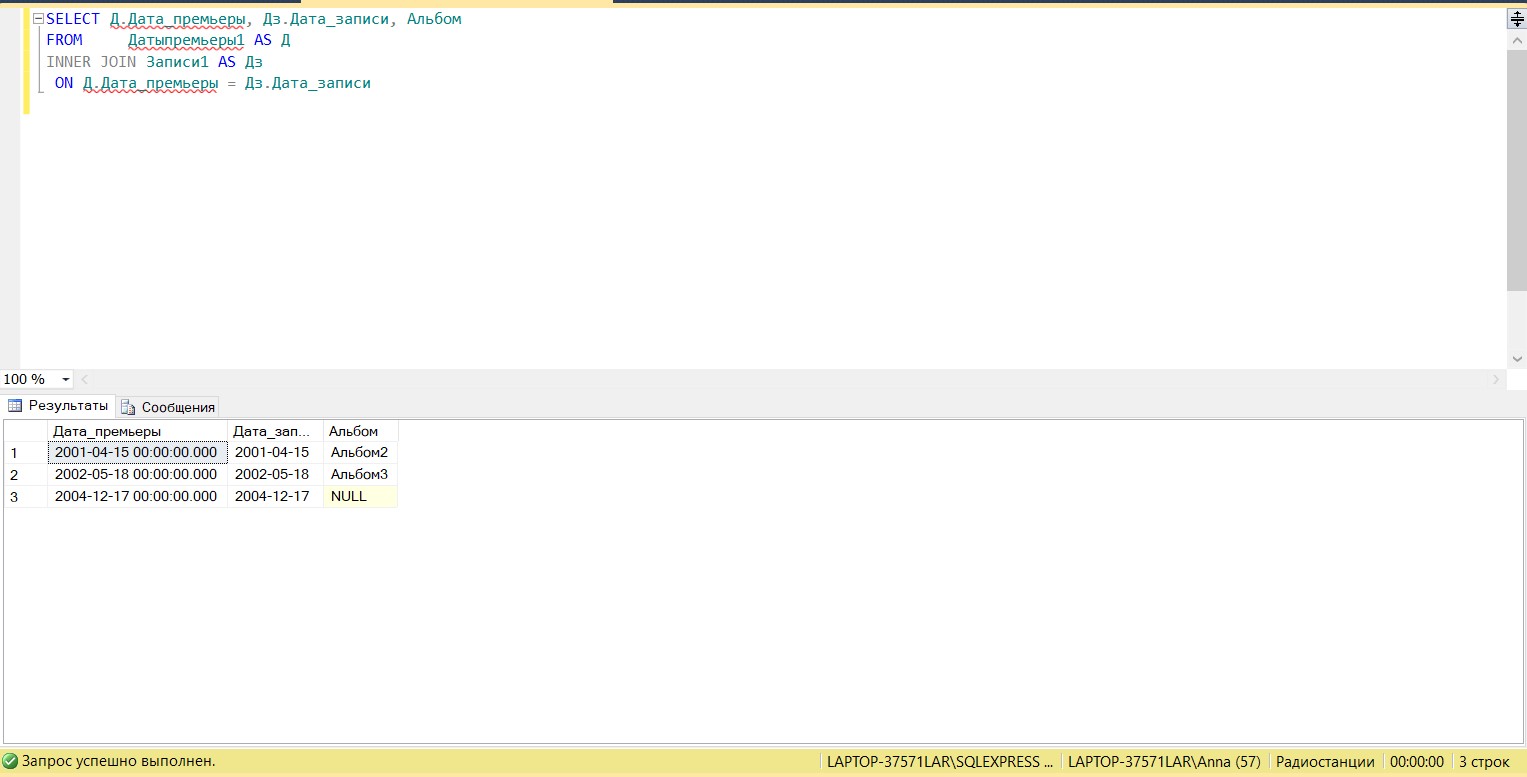
FROM dbo.Датыпремьеры1;

GO

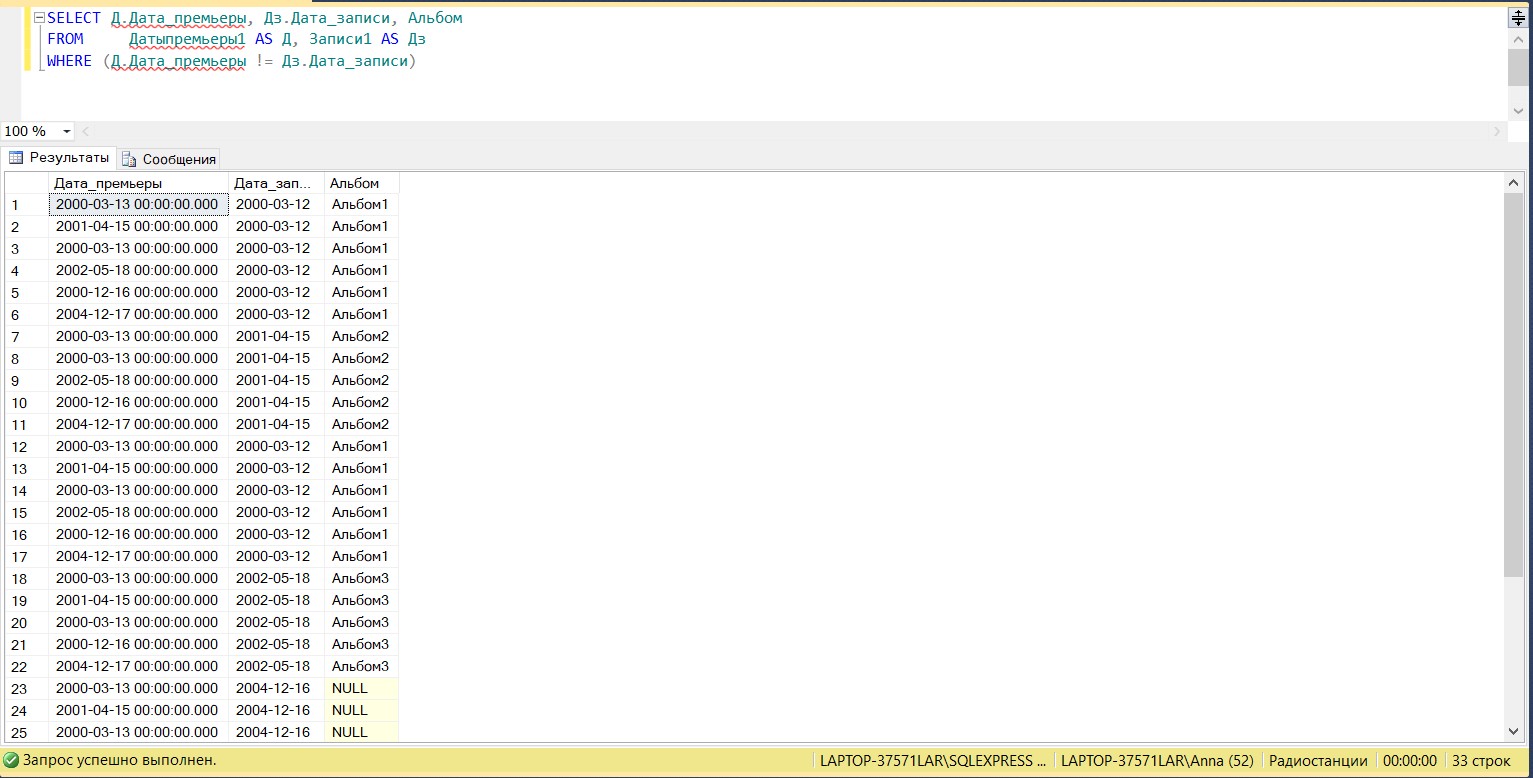




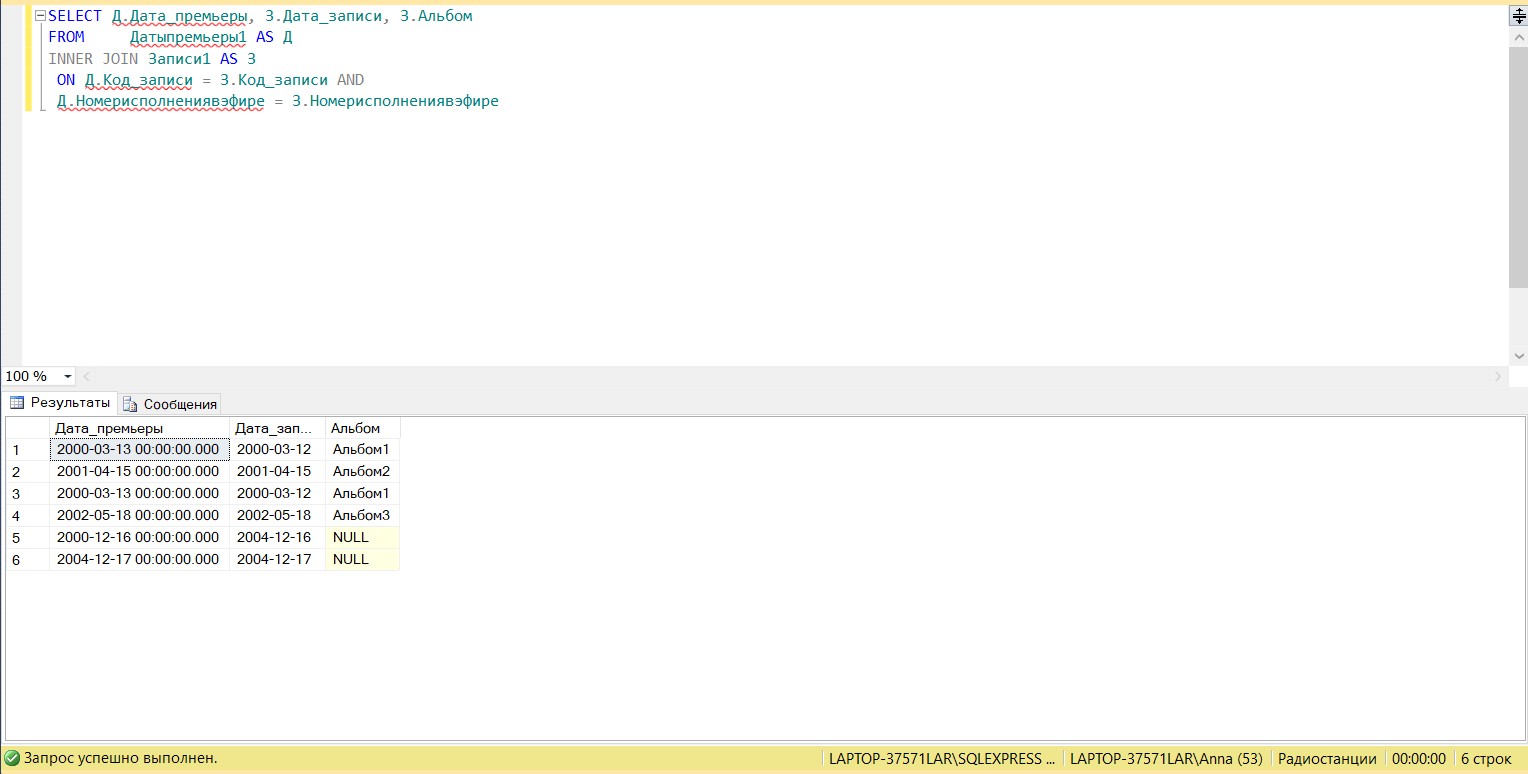
**12.**



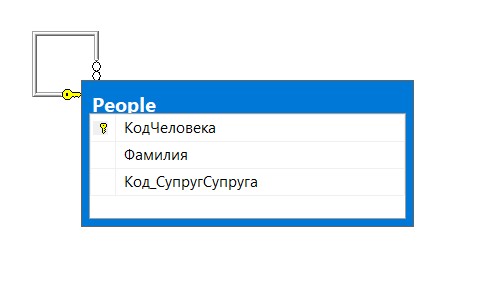
**13.**



**14**



**15.**



USE Радиостанции;

IF OBJECT\_ID (N'dbo.People', N'U') IS NOT NULL

DROP TABLE dbo.[People];

GO

CREATE TABLE People

(

КодЧеловека INTEGER IDENTITY(1, 1) PRIMARY KEY NOT NULL,

Фамилия VARCHAR (15),

Код\_СупругСупруга INTEGER FOREIGN KEY REFERENCES People (КодЧеловека) DEFAULT NULL

)

INSERT INTO dbo.People (Фамилия,Код\_СупругСупруга)

Values

('Ершов',2),

('Соловьева',1),

('Меньшиков',4),

('Меньшикова',3)

INSERT INTO dbo.People (Фамилия)

Values

('Никитюк'),

('Южина')

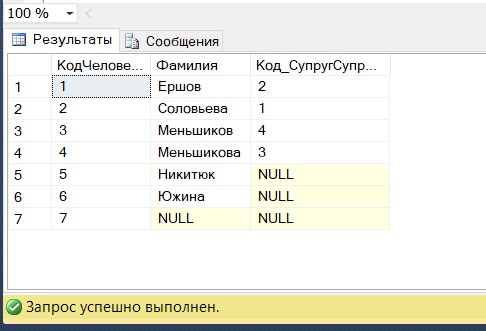
INSERT INTO dbo.People DEFAULT VALUES;

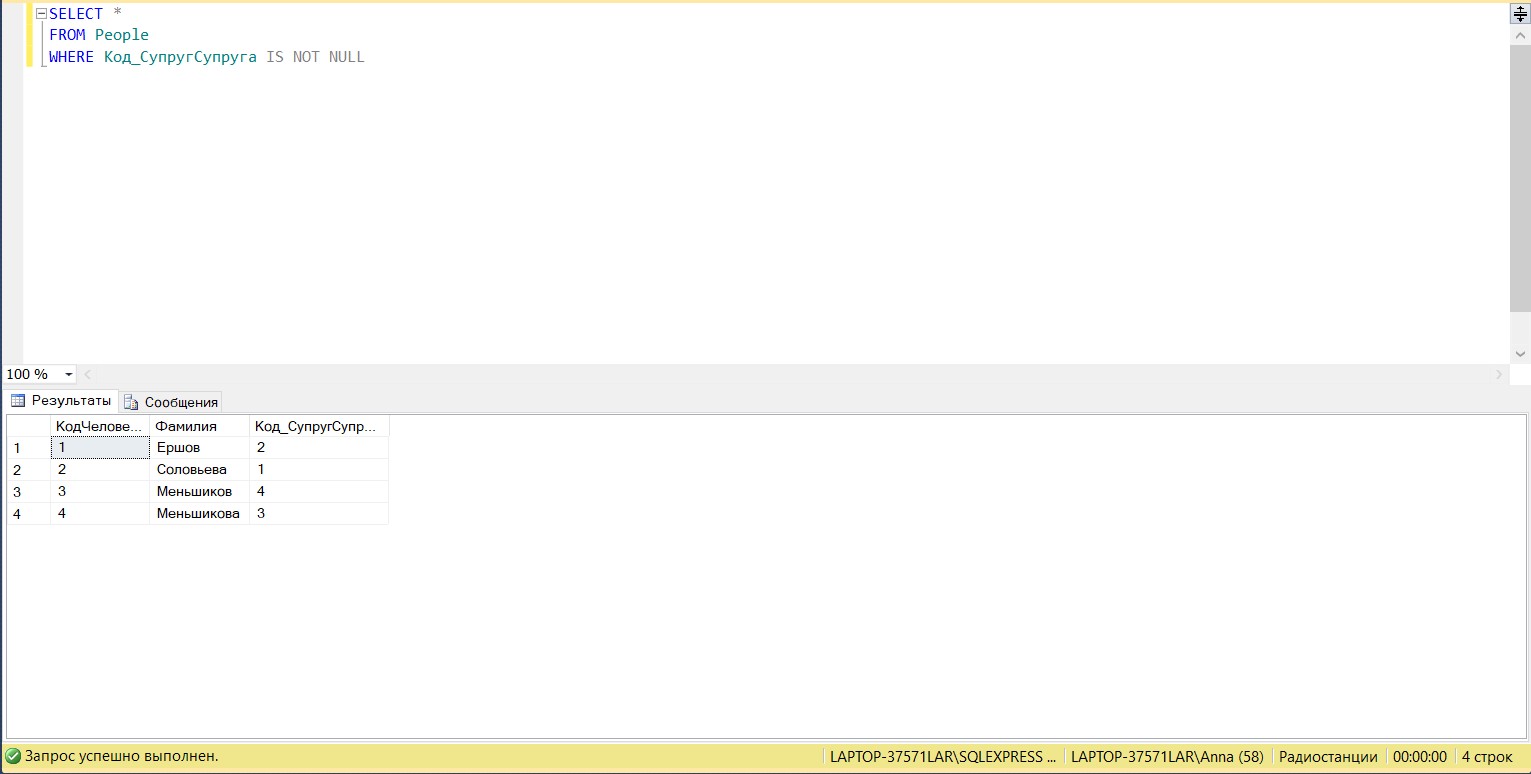
GO

SELECT \*

FROM dbo.People;

GO



**Люди, имеющие супруга:**

**16.**

USE Радиостанции;

IF OBJECT\_ID (N'dbo.Сотрудники3', N'U') IS NOT NULL

DROP TABLE dbo.[Сотрудники3];

GO

CREATE TABLE dbo.Сотрудники3

(

Код\_сотрудника int PRIMARY KEY IDENTITY(1,1) NOT NULL,

Фамилия nchar(15) NULL,

Дата\_рождения DATE NULL,

Специальность nchar(10) NULL

)

DELETE FROM Сотрудники3

SET IDENTITY\_INSERT Сотрудники3 ON

INSERT INTO dbo.Сотрудники3 (Код\_сотрудника, Фамилия, Дата\_рождения, Специальность)

Values

(1,'Карпов','01.10.1990', 'Спец1'),

(2,'Смирнов','01.10.1980', 'Спец1'),

(3,'Мишин','01.10.1995', 'Спец2'),

(4,'Серпин','01.10.1982', 'Спец2'),

(5,'Кручинин','01.10.2000', 'Спец3'),

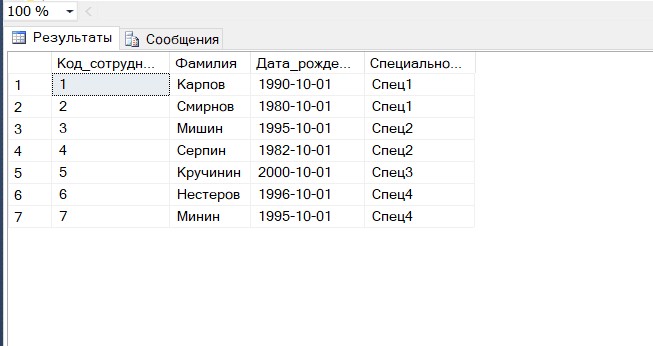
(6,'Нестеров','01.10.1996', 'Спец4'),

(7,'Минин','01.10.1995', 'Спец4')

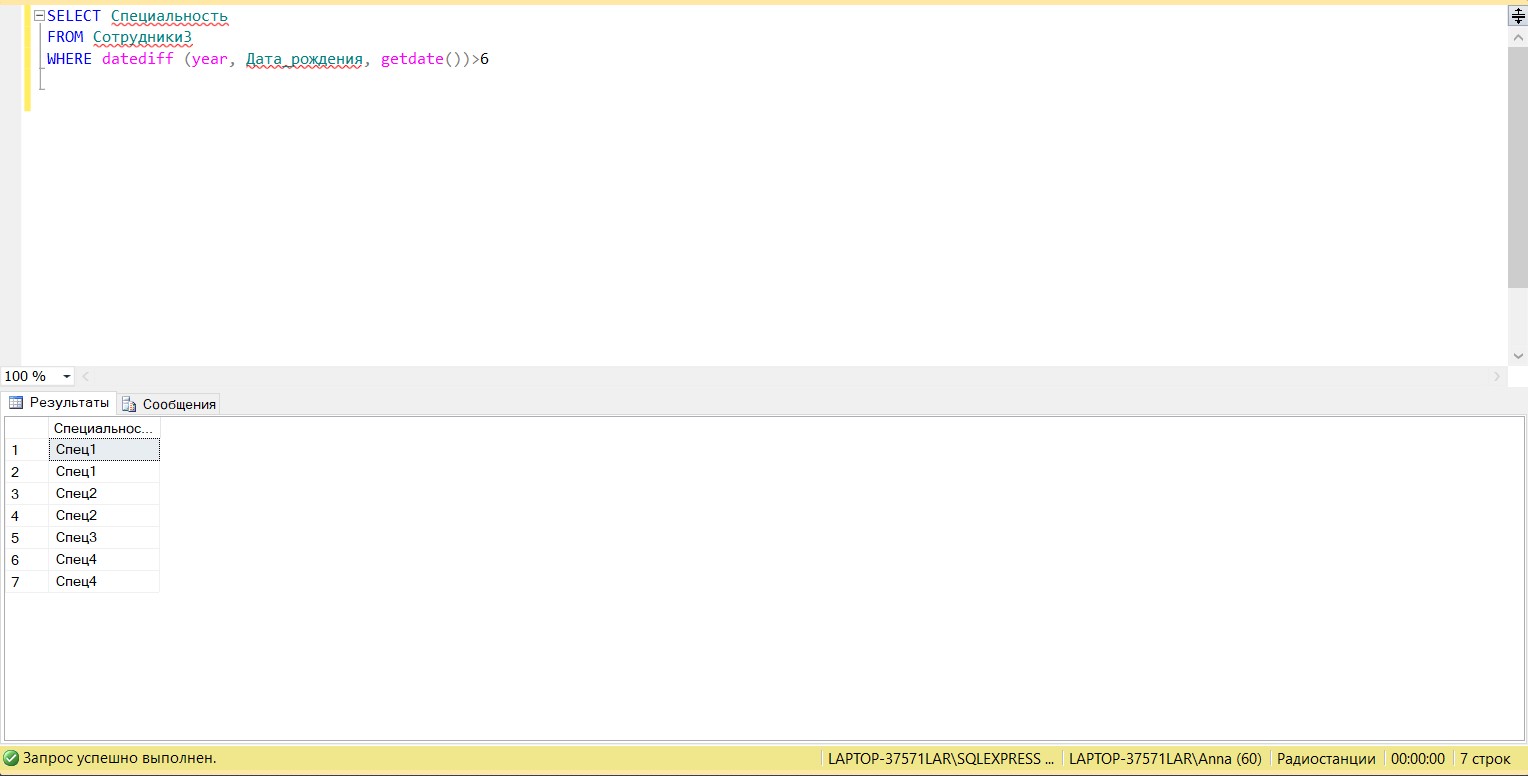
SELECT \*

FROM dbo.Сотрудники3;

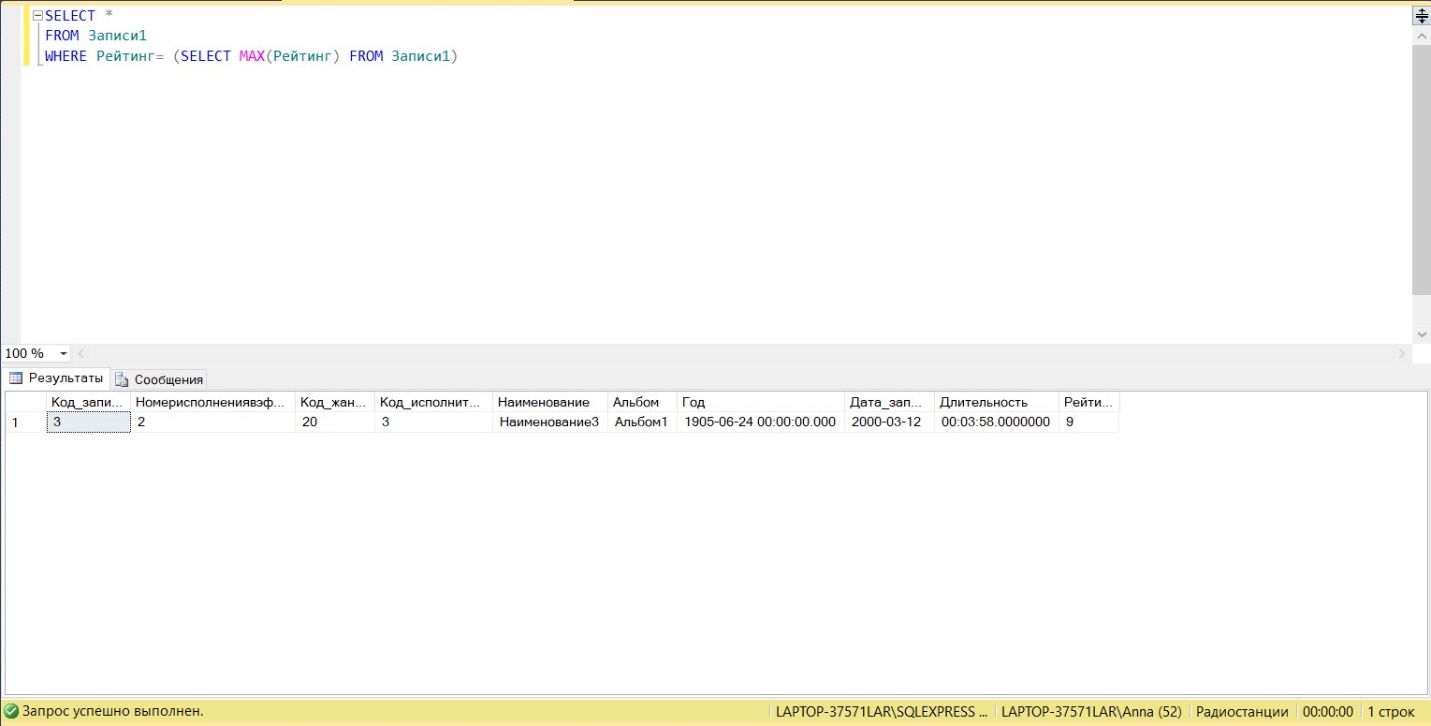
GO



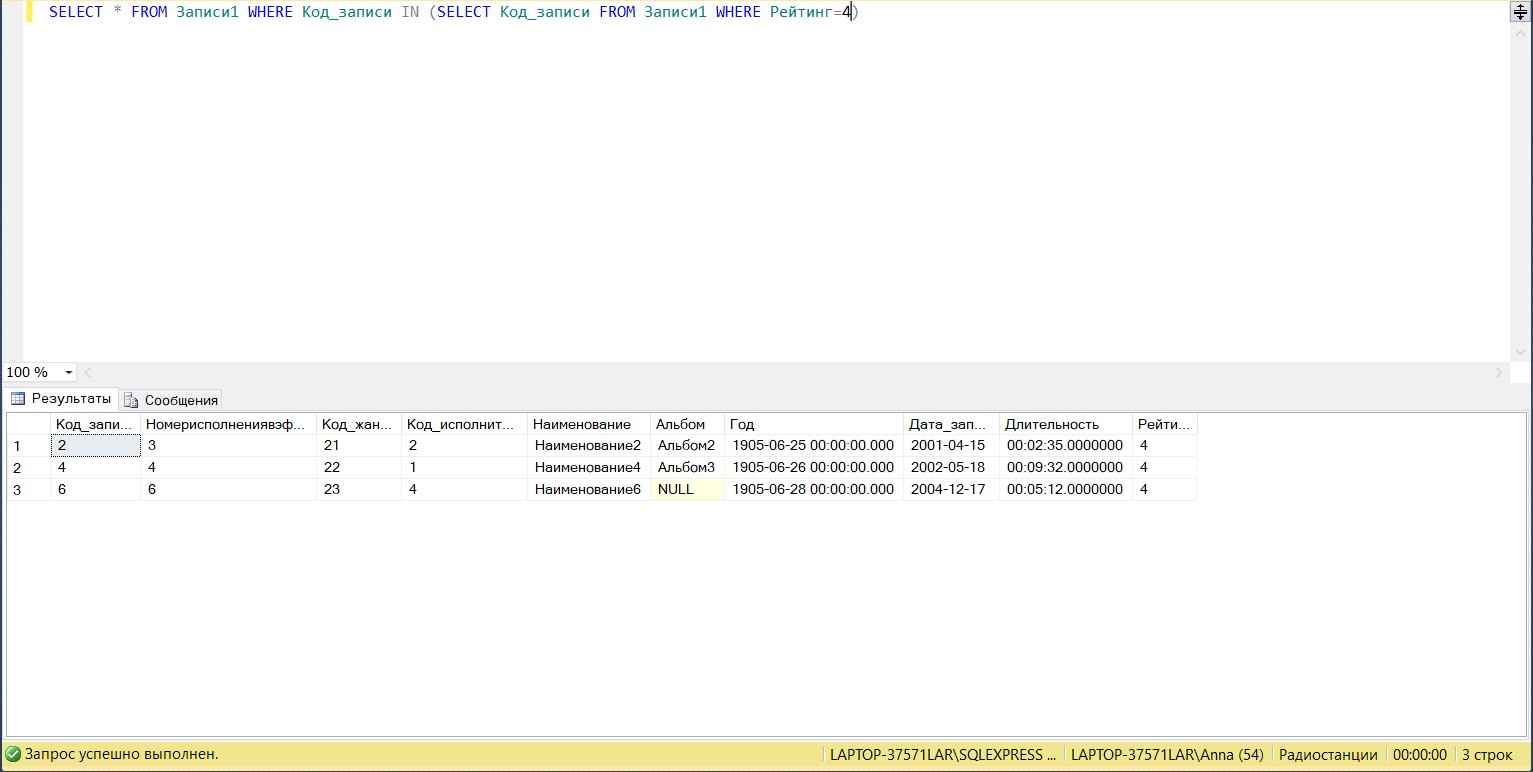
**Выборка пар:**



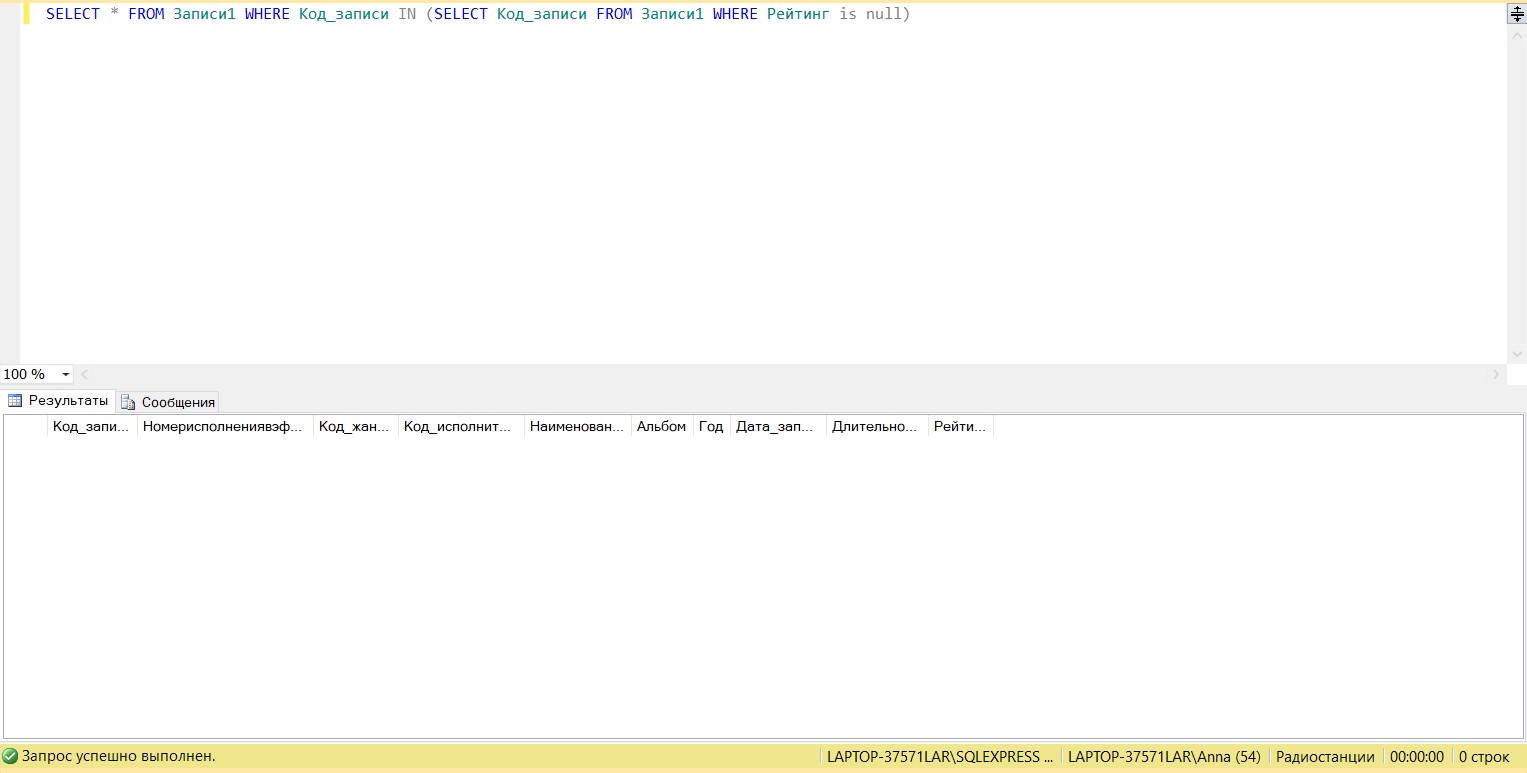
**17(a)**



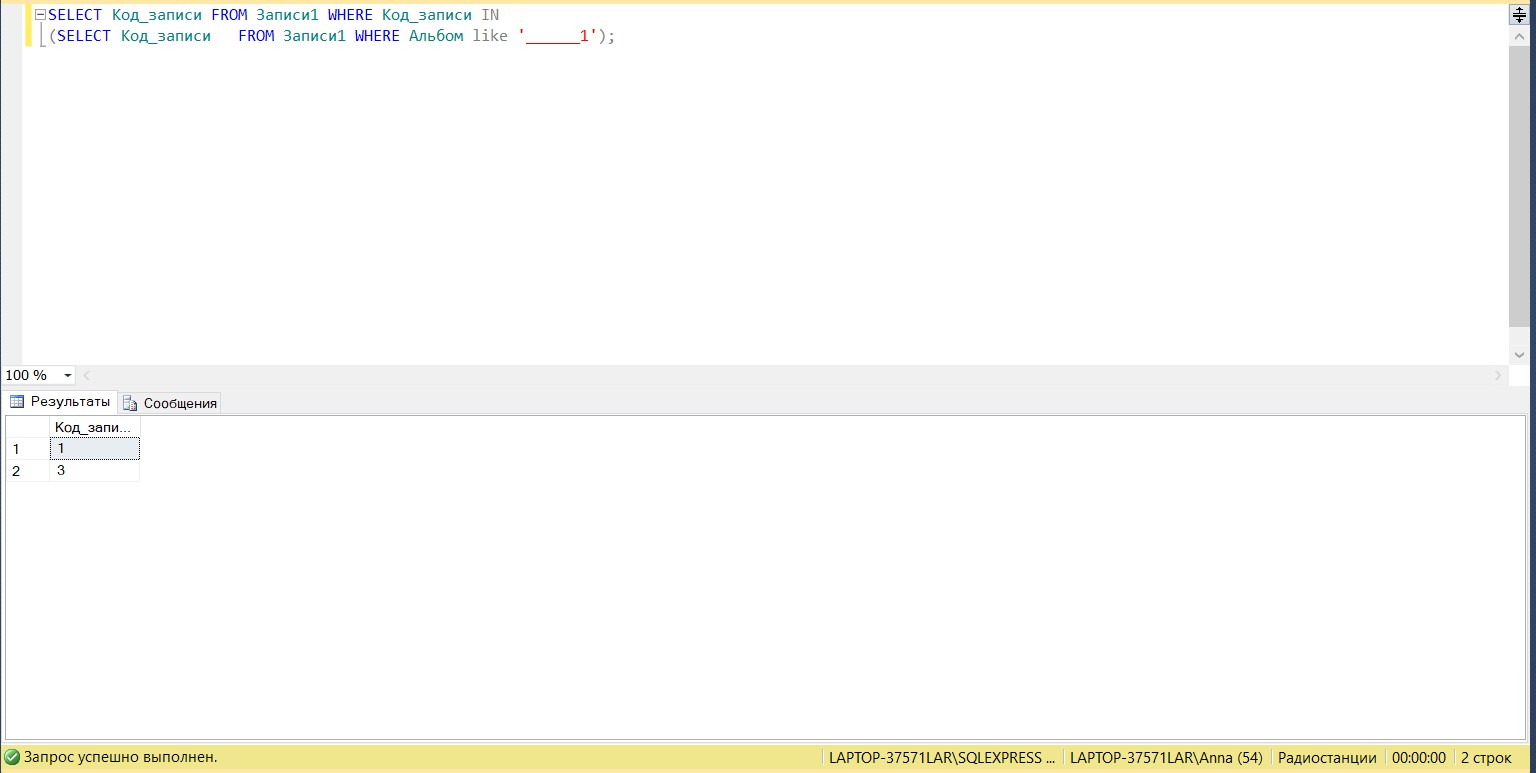
**17(b)**



**17(c)**



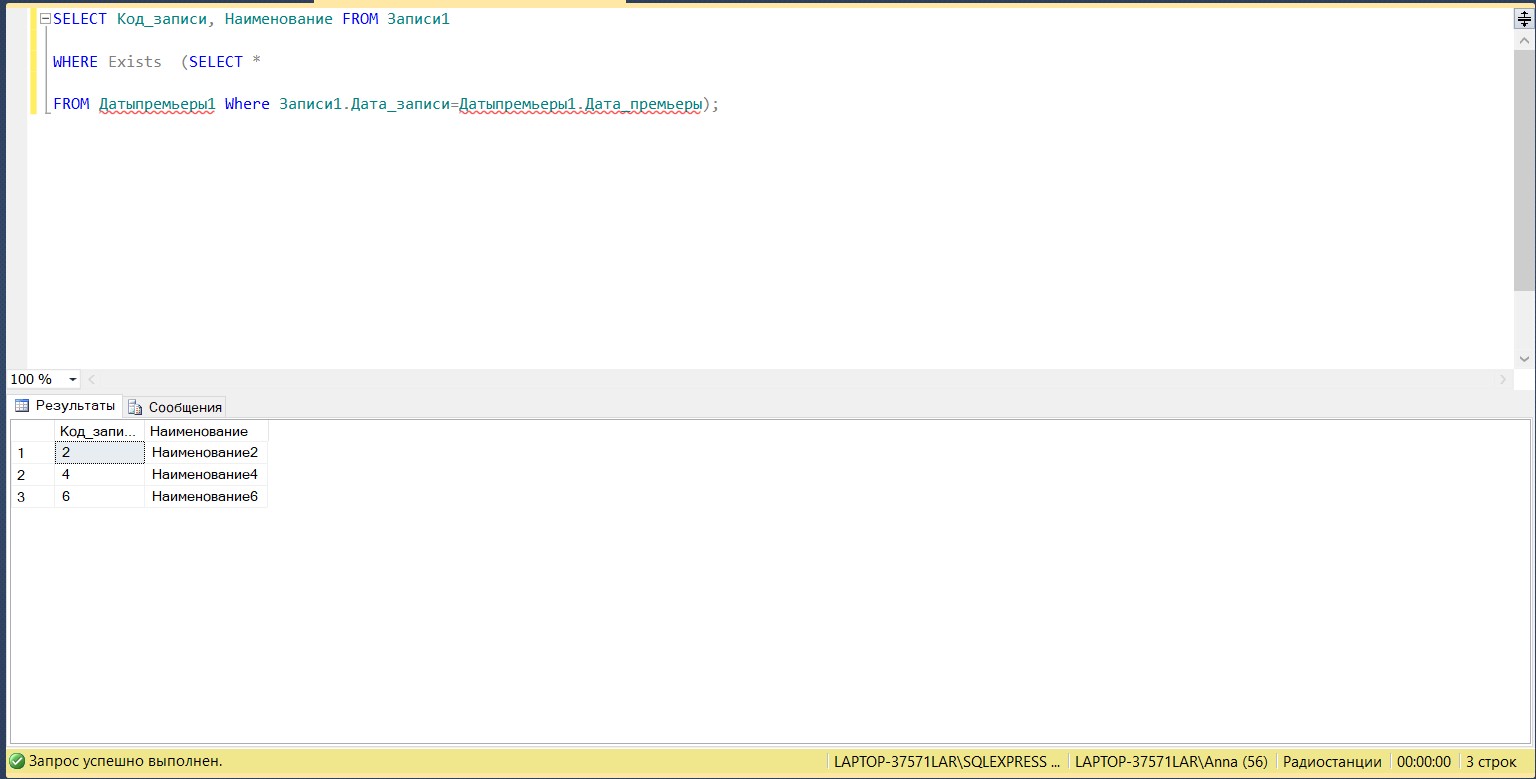
**18.**



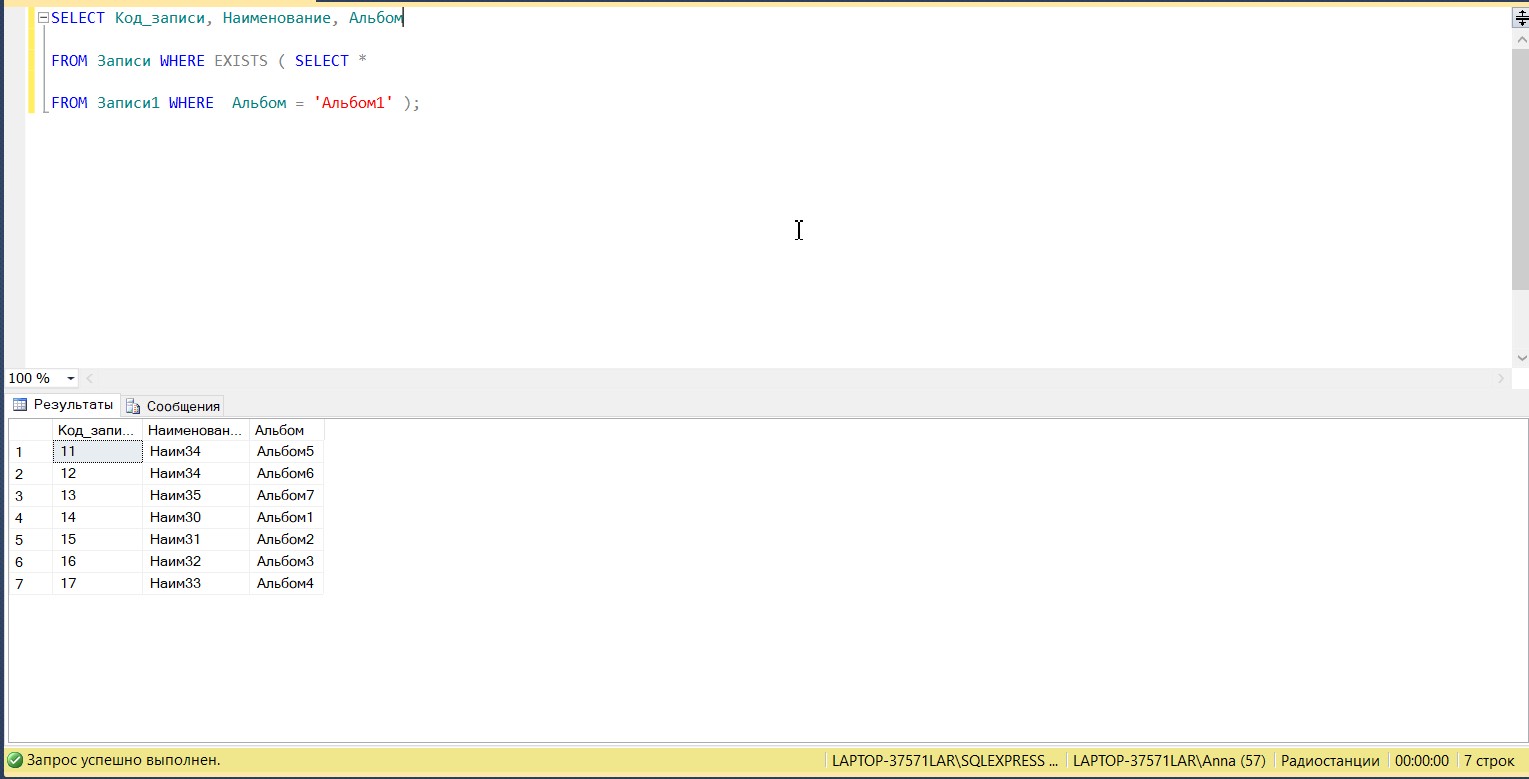
**19.**



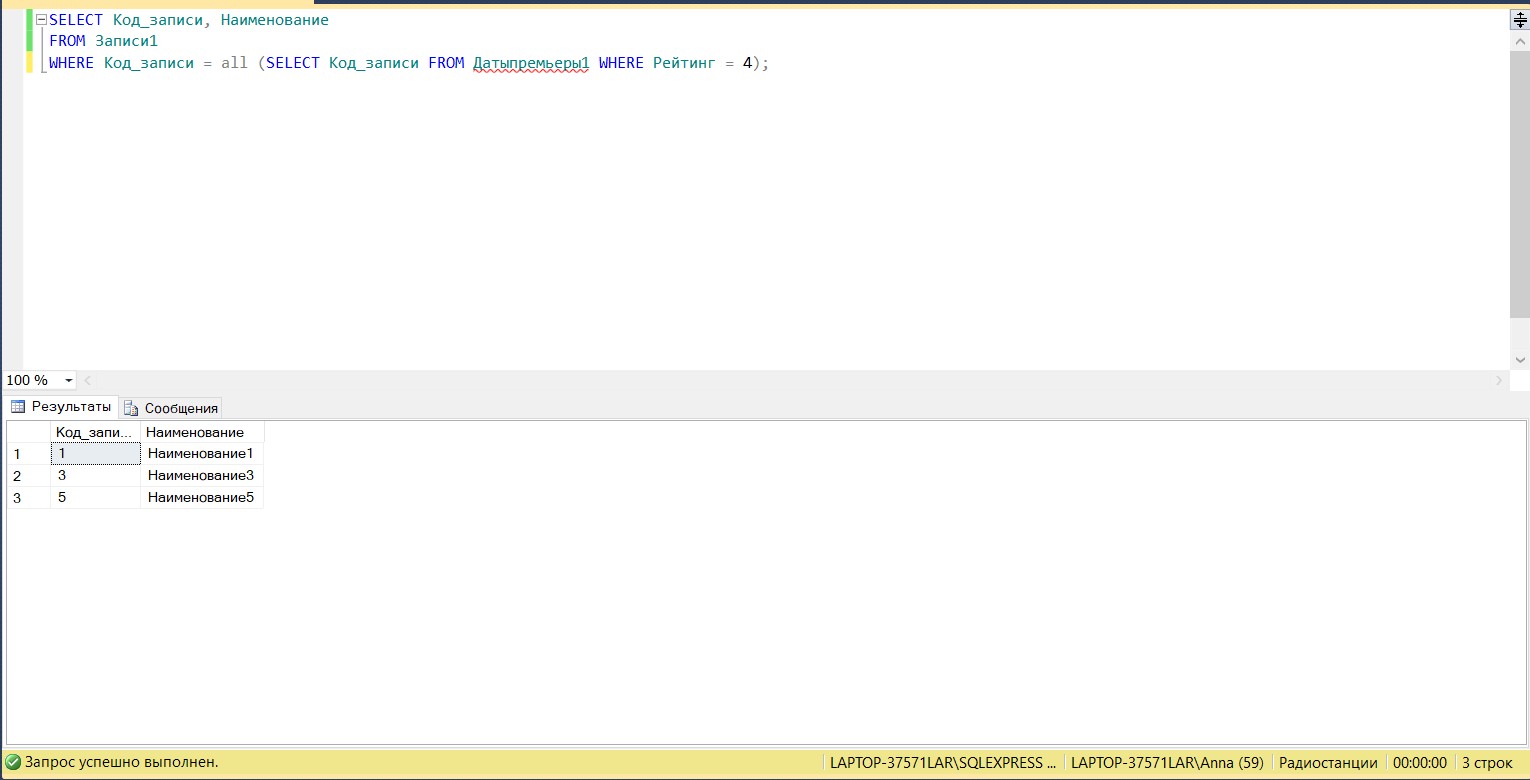
**20**



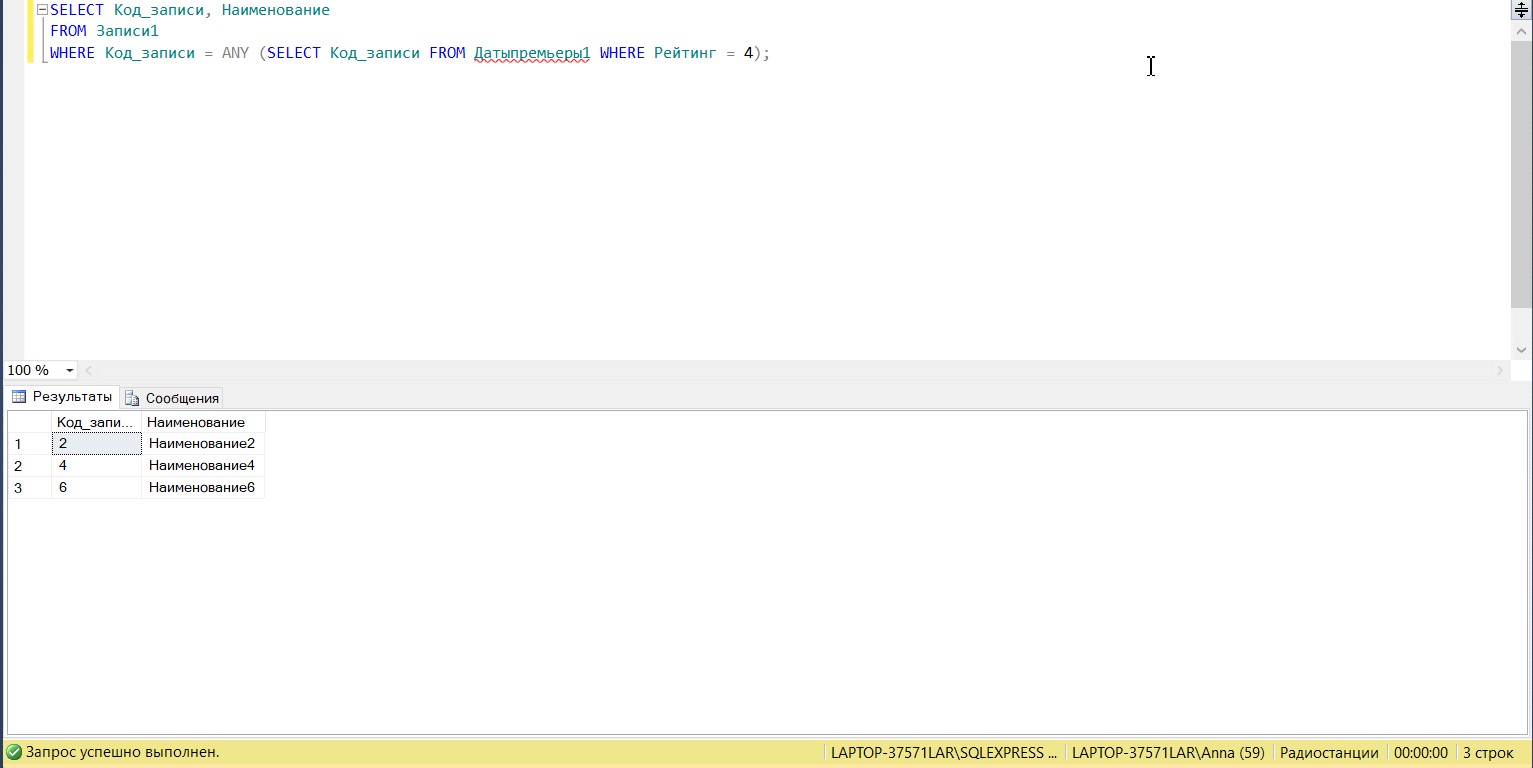
**21.**



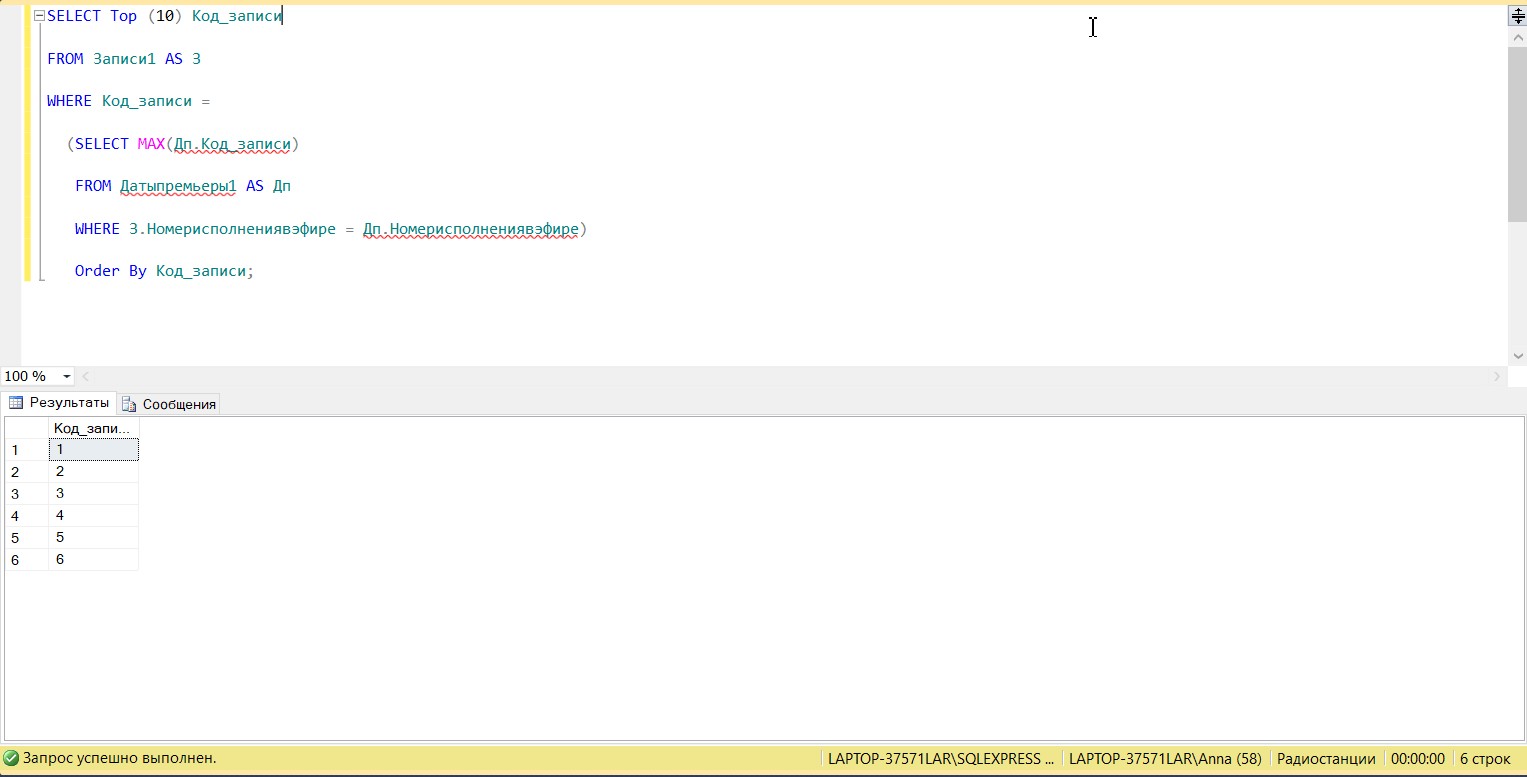
**22.**



**23.**

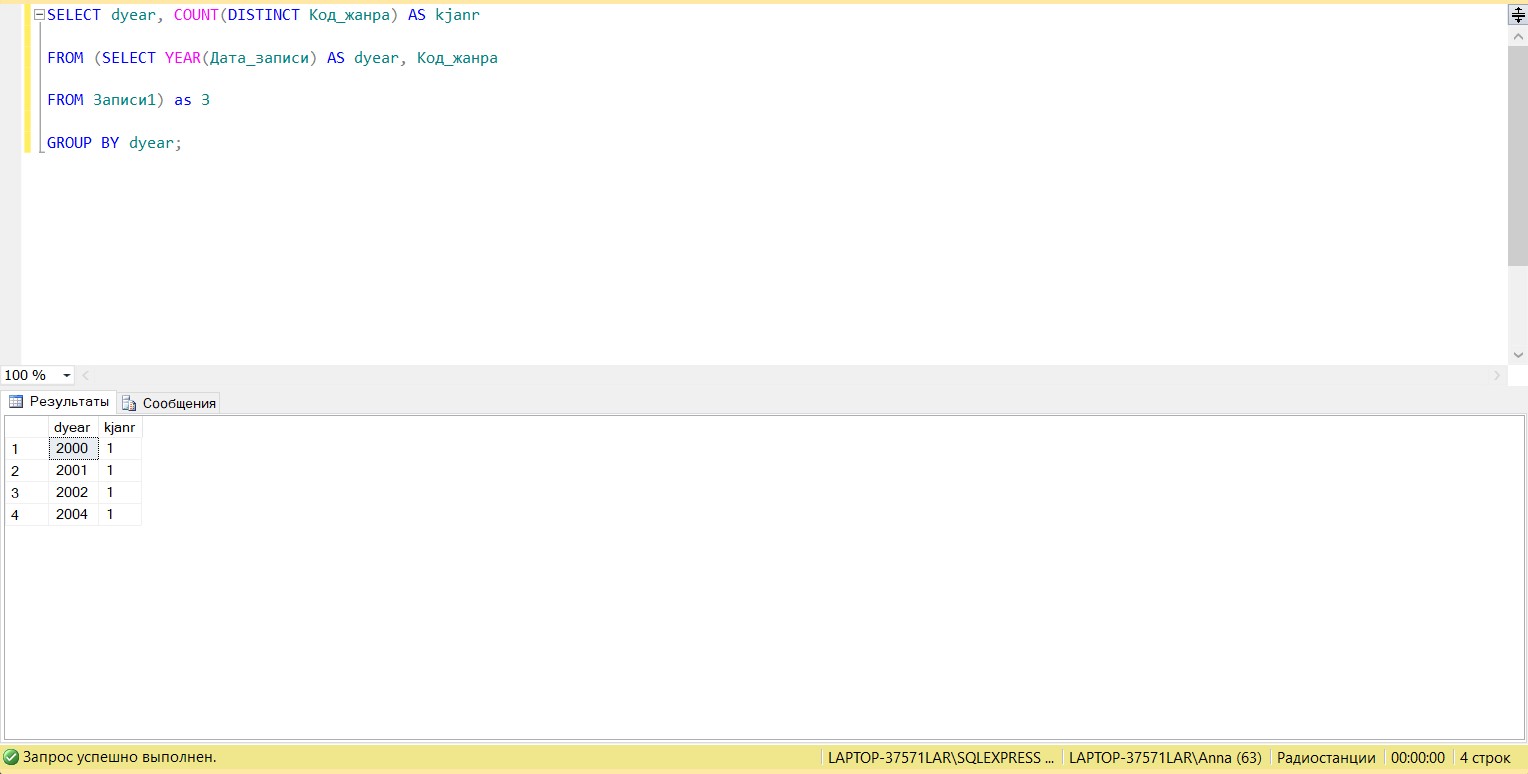


**24.**

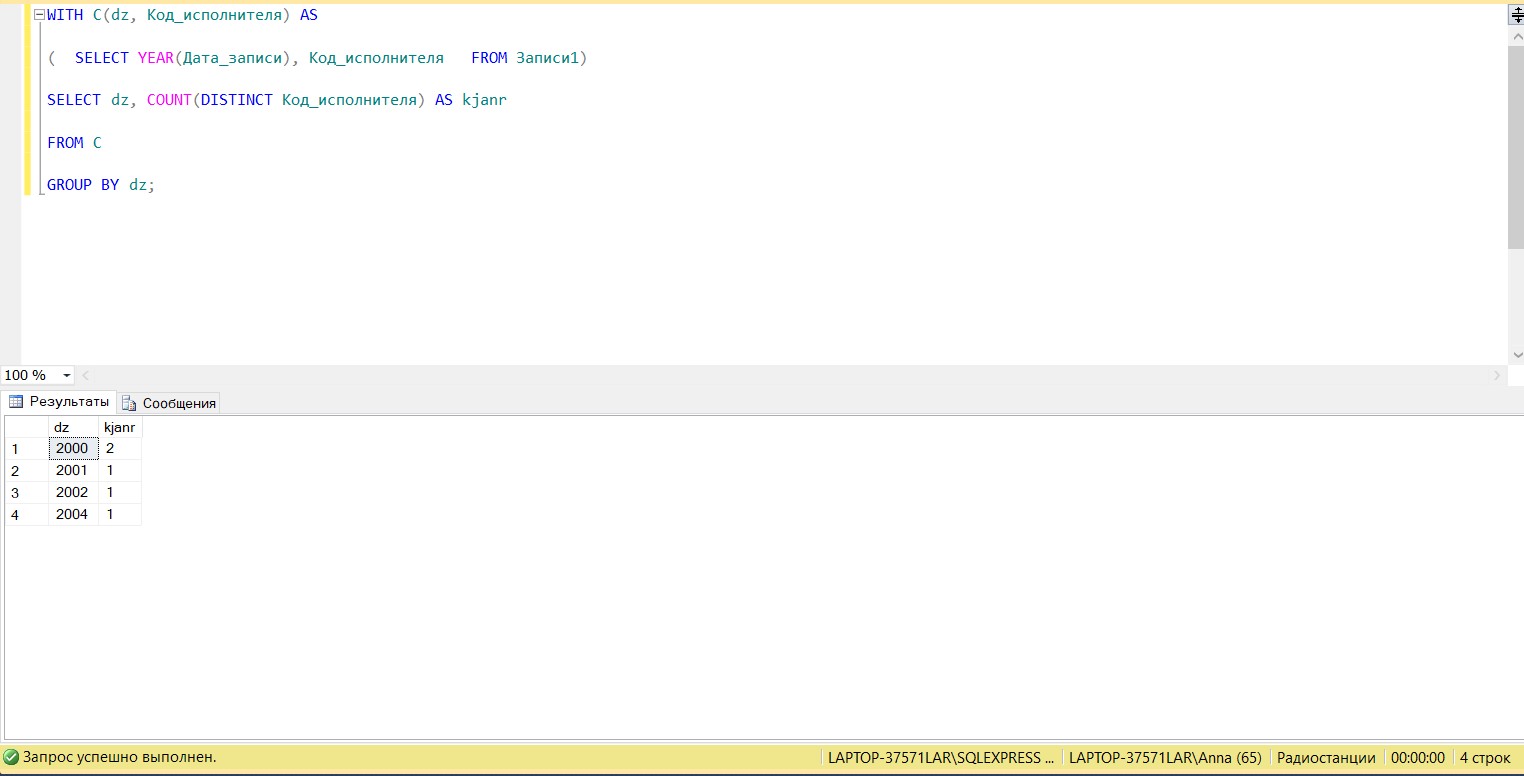


1. Выбирается строка из таблицы во внешнем запросе. Это будет текущая строка-кандидат.
2. Сохранить значения из этой строки-кандидата для использования в предложении FROM внешнего запроса.
3. Выполнить подзапрос, используя значение для текущей строки-кандидата. Использование значения из строки-кандидата внешнего запроса в подзапросе называется — внешней ссылкой.
4. Оценить предикат (Истина или Ложь) внешнего запроса на основе результатов подзапроса, выполняемого в шаге 3. Он определяет — выбирается ли строка-кандидат для вывода.
5. Повторить процедуру для следующей строки-кандидата таблицы, и так далее, пока все строки таблицы не будут проверены.

**25.**



**26.**



**27**

USE Радиостанции;

IF OBJECT\_ID (N'dbo.Сотрудники4', N'U') IS NOT NULL

DROP TABLE dbo.[Сотрудники4];

GO

CREATE TABLE dbo.Сотрудники4

(

Код\_сотрудника int PRIMARY KEY IDENTITY(1,1) NOT NULL,

Фамилия nchar(15) NULL,

Дата\_рождения DATE NULL,

Специальность nchar(10) NULL,

Код\_подчиненного int REFERENCES Сотрудники4(Код\_сотрудника) null

)

DELETE FROM Сотрудники4

SET IDENTITY\_INSERT Сотрудники4 ON

INSERT INTO dbo.Сотрудники4 (Код\_сотрудника, Фамилия, Дата\_рождения, Специальность,Код\_подчиненного)

Values

(1,'Карпов','01.10.1990', 'Спец1',2),

(2,'Смирнов','01.10.1980', 'Спец1',1),

(3,'Мишин','01.10.1995', 'Спец2',4),

(4,'Серпин','01.10.1982', 'Спец2',6),

(5,'Кручинин','01.10.2000', 'Спец3',6),

(6,'Нестеров','01.10.1996', 'Спец4',NULL)

SELECT \*

FROM dbo.Сотрудники4;

GO

WITH EmpsRN AS

(

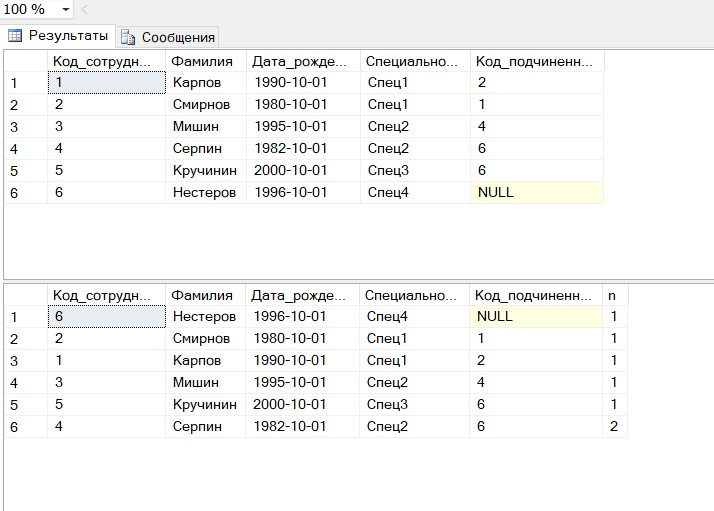
SELECT \*,

ROW\_NUMBER() OVER(PARTITION BY Код\_подчиненного ORDER BY Фамилия, Код\_сотрудника) AS n

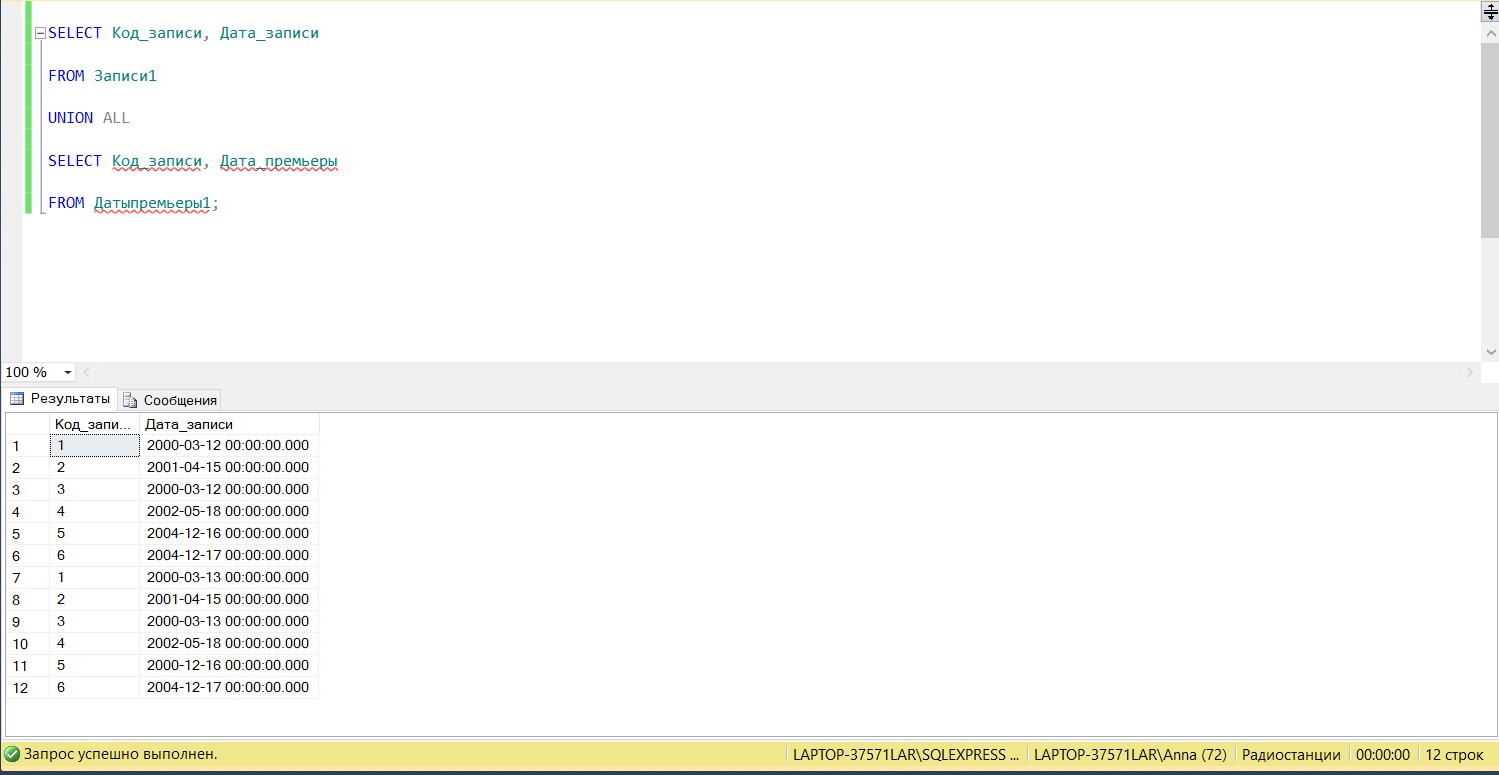
FROM dbo.Сотрудники4

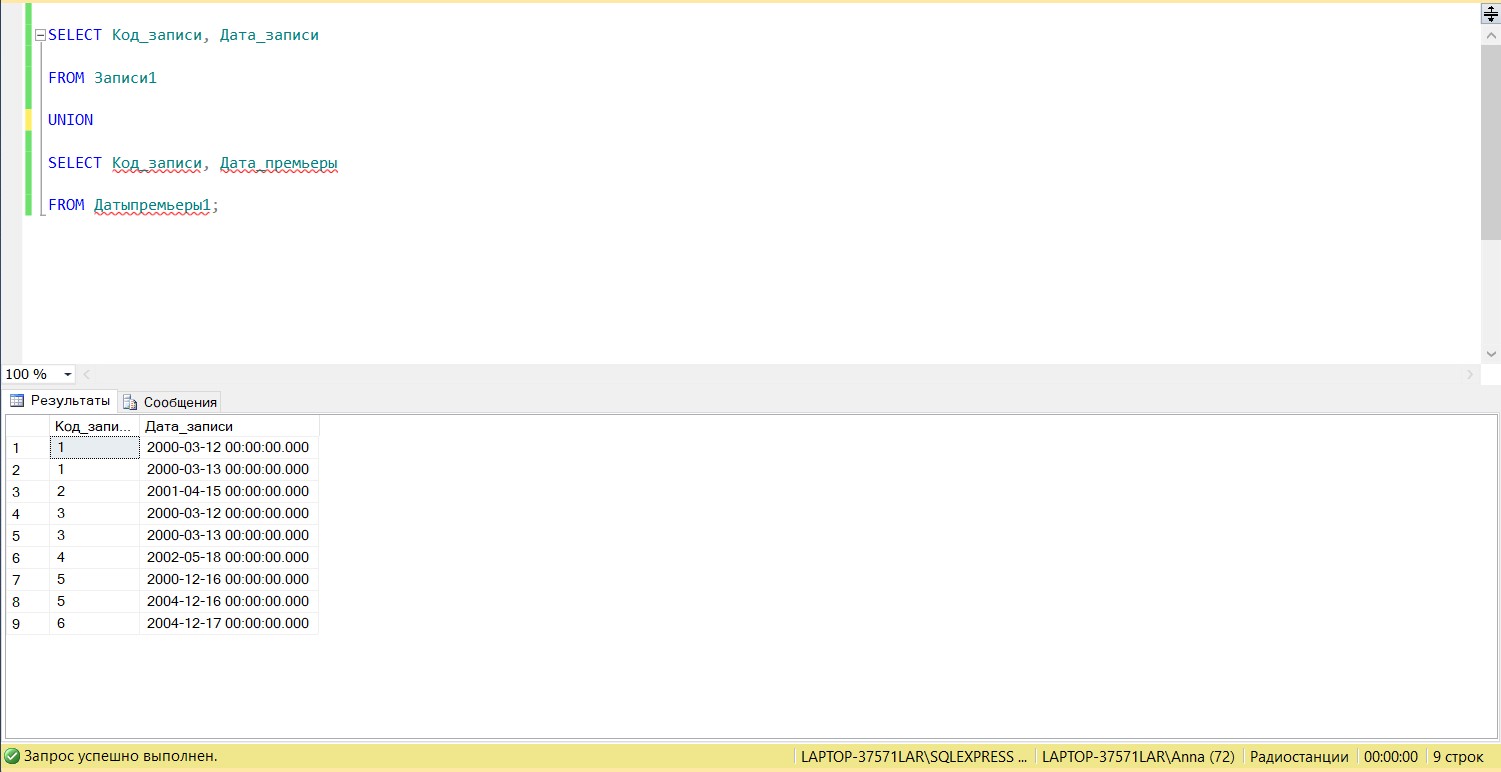
)

SELECT \* FROM EmpsRN;



**28.**





**29.**

WITH EmpsCTE AS

(

SELECT Код\_сотрудника, Код\_подчиненного, Фамилия

FROM Сотрудники4

WHERE Фамилия = 'Нестеров'

UNION ALL

SELECT C.Код\_сотрудника, C.Код\_подчиненного, C.Фамилия

FROM EmpsCTE AS P

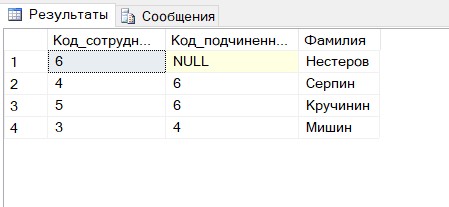
JOIN Сотрудники4 AS C

ON C.Код\_подчиненного = P.Код\_сотрудника

)

SELECT Код\_сотрудника, Код\_подчиненного, Фамилия

FROM EmpsCTE;

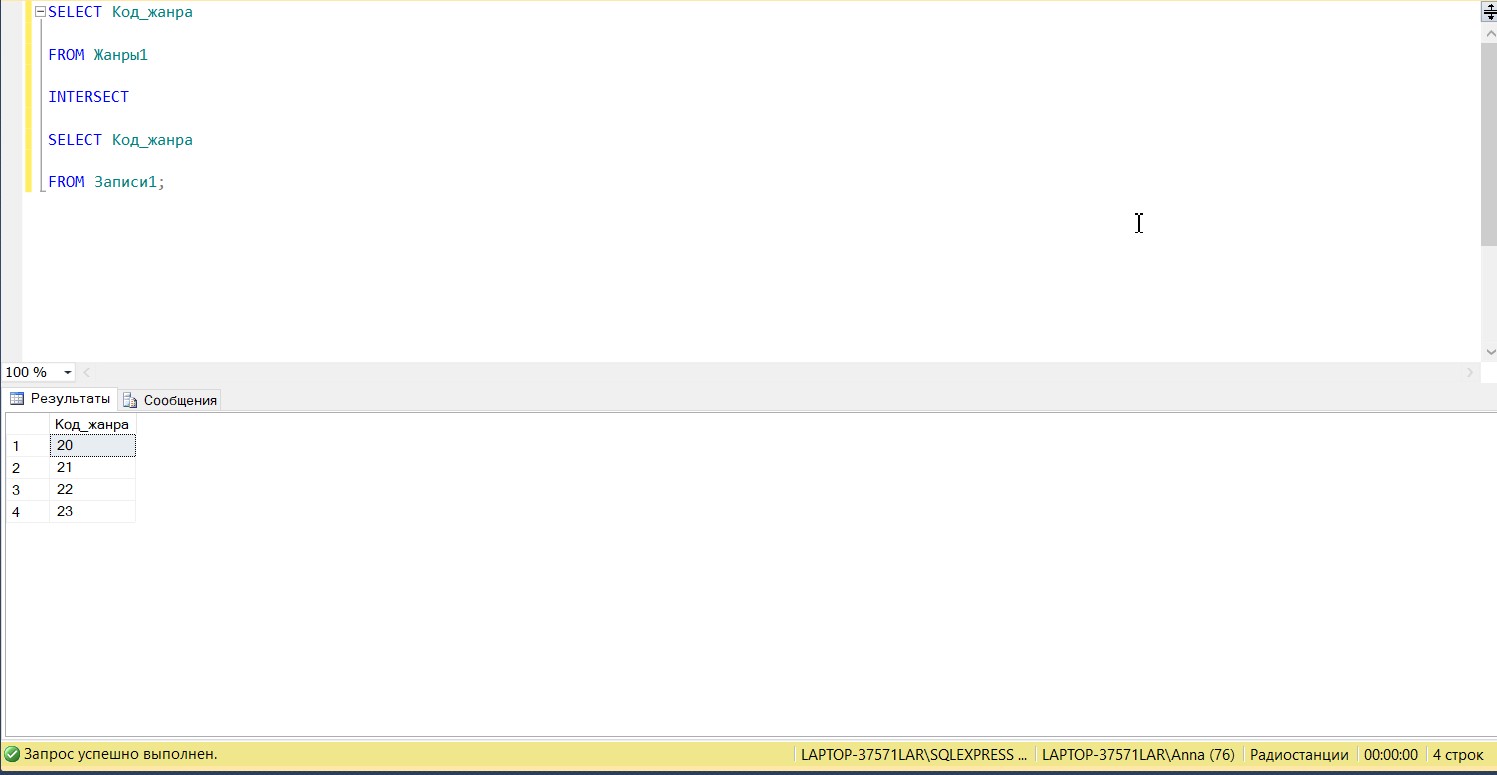


Для получения  иерархии подчинения более чем для одного уровня по всей таблице необходим запрос,  обладающий рекурсивными свойствами. Такими рекурсивными возможностями обладают рассмотренные общие табличные выражения (ОТВ). В каждом последующем запуске рекурсивного компонента ссылка на ОТВ представляет собой результирующий набор, возращенный запуском рекурсивного компонента. У рекурсивного компонента нет явного ограничения для прерывания рекурсии. Это ограничение неявное. Рекурсивный компонент инициируется многократно до тех пор, пока не вернет пустой результирующий набор или не превысит некоторое заданное ограничение.

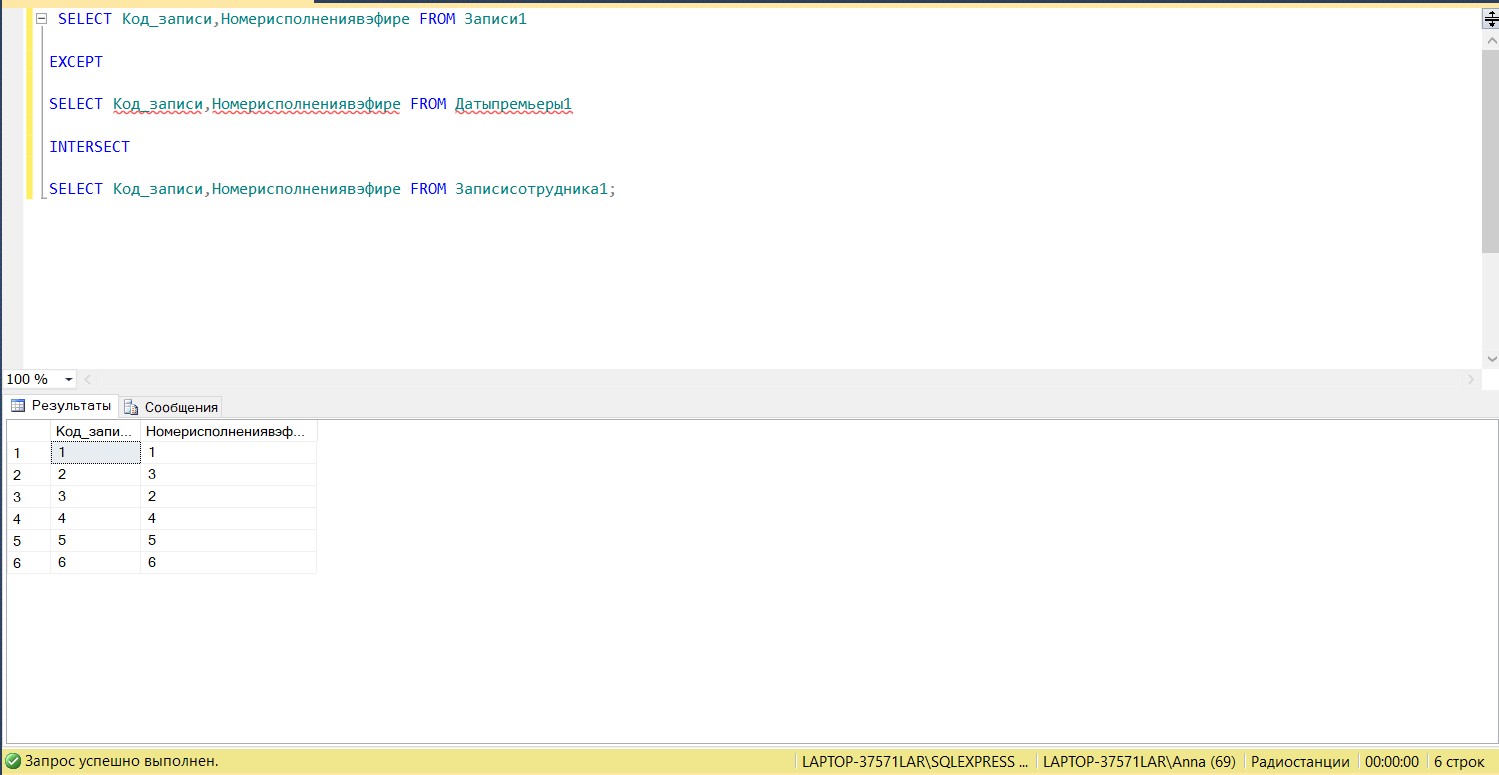
Ссылка на ОТВ во внешнем запросе представляет объединенные результирующие наборы, полученные в результате запуска фиксированного компонента и всех запусков рекурсивного компонента.

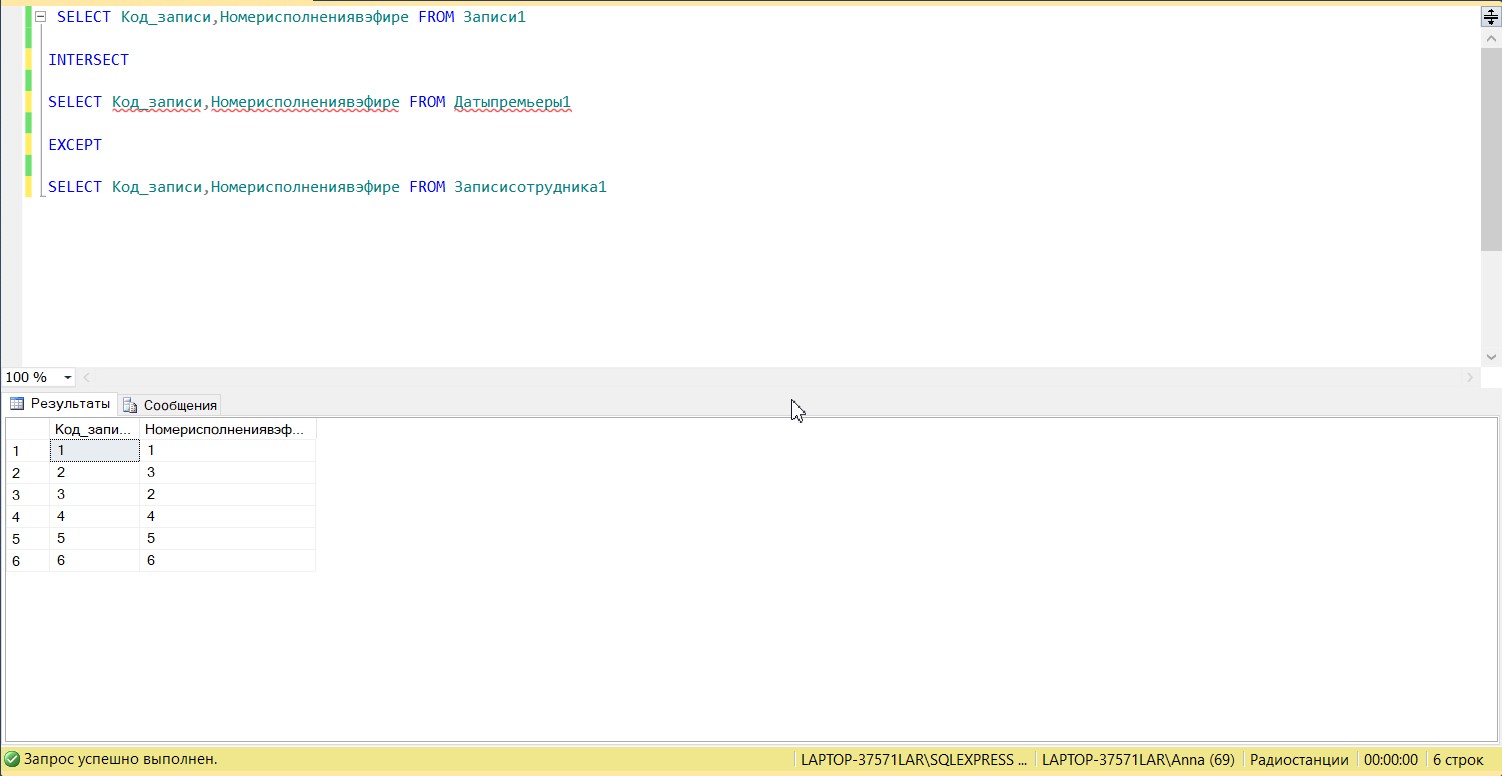
Фиксированный компонент соединяет  ОТВ, представляющее предыдущий результирующий набор, с таблицей Сотрудники для получения прямых подчинений сотрудников, возвращенных в предыдущем результирующем наборе. Т.е. рекурсивный компонент запускается многократно и при каждом запуске возвращает сотрудников на следующем уровне иерархии подчинения.

**30.**



**31.**





**32.**

**Создание представления для исполнителей с фамилией Ершов.**

CREATE VIEW ИсполнительЕршов

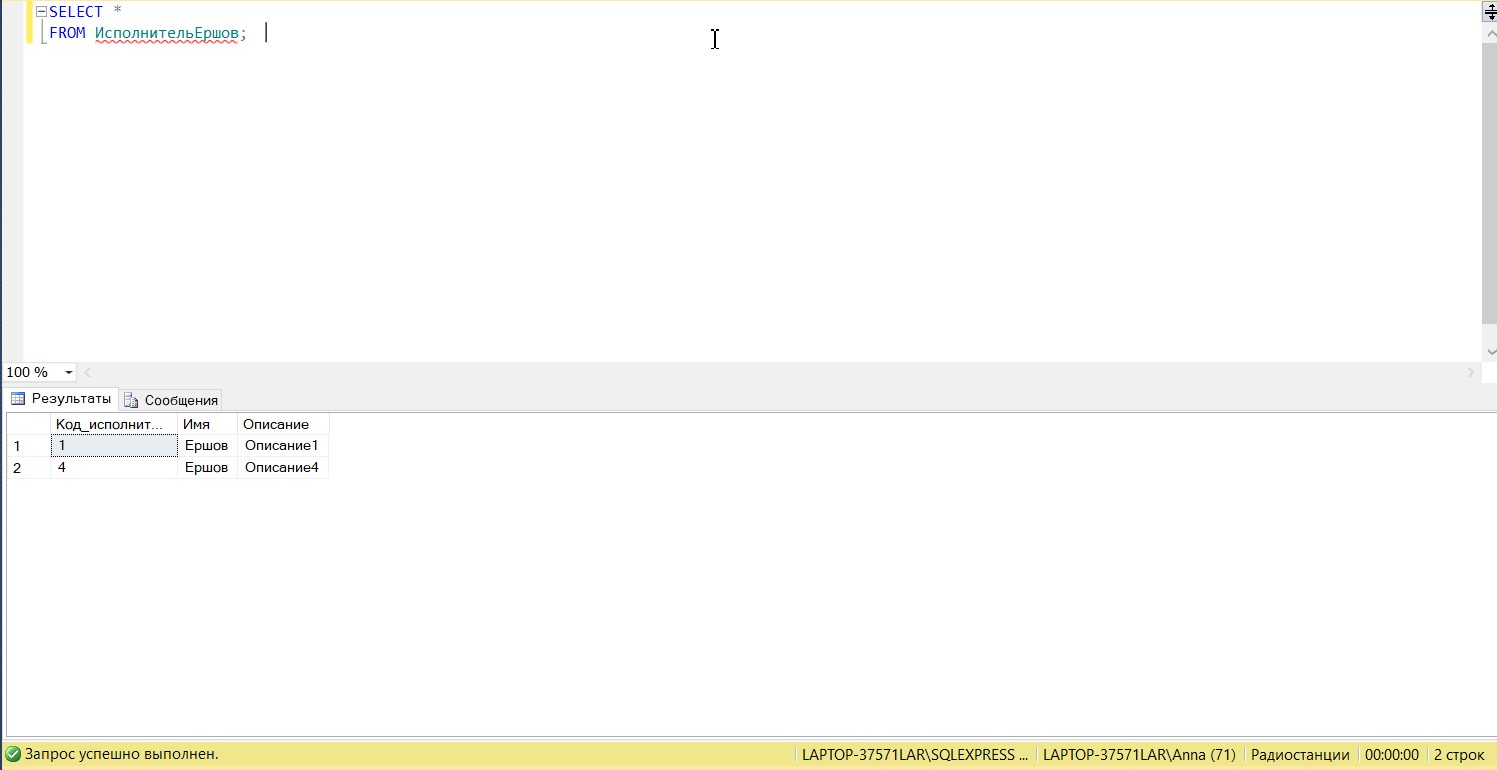
AS SELECT \*

FROM Исполнители1

WHERE Имя = 'Ершов';

SELECT \*

FROM ИсполнительЕршов;



**33.**

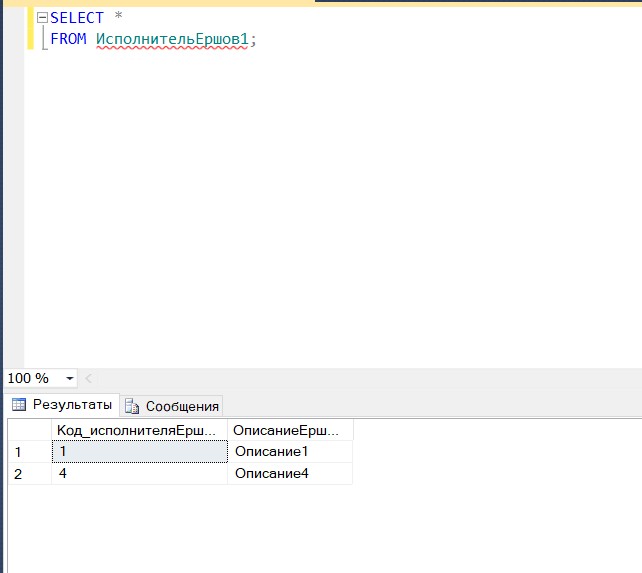
CREATE VIEW ИсполнительЕршов1 (Код\_исполнителяЕршова, ОписаниеЕршова) AS SELECT Код\_исполнителя, Описание

FROM Исполнители1

WHERE Имя = 'Ершов';

SELECT \*

FROM ИсполнительЕршов1;



**34.**

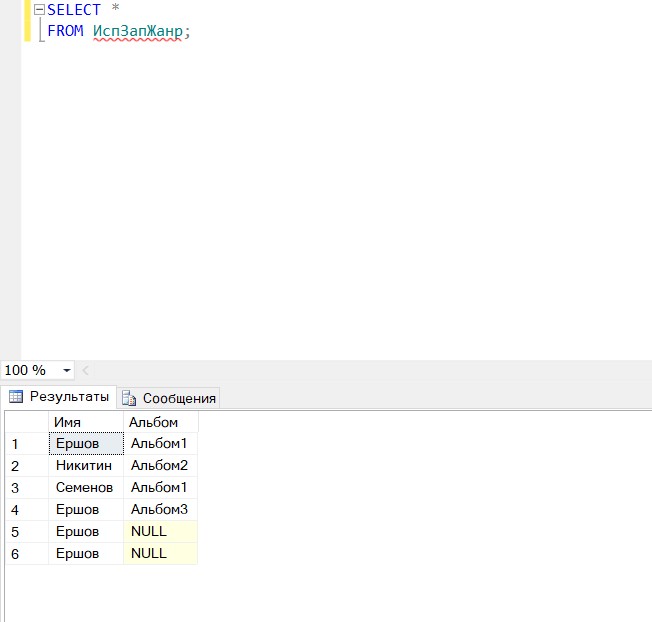
CREATE VIEW ИспЗапЖанр

AS SELECT Имя, Альбом FROM Исполнители1 AS a, Жанры1 As b, Записи1 As c

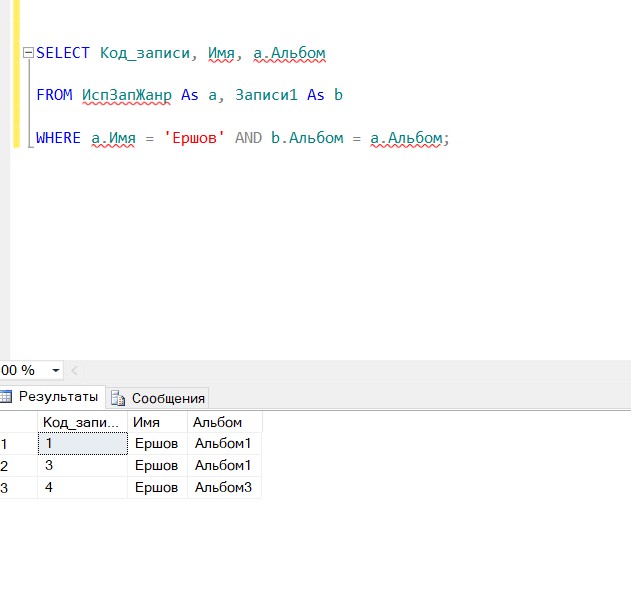
WHERE a.Код\_исполнителя = c.Код\_исполнителя AND b.Код\_жанра = c.Код\_жанра;

SELECT \*

FROM ИспЗапЖанр;



**35**



SELECT Код\_записи, Имя, a.Альбом

FROM ИспЗапЖанр As a, Записи1 As b

WHERE a.Имя = 'Ершов' AND b.Альбом = a.Альбом;

**36**

CREATE VIEW TOPReiting

AS SELECT Код\_записи, Рейтинг

FROM Записи1

WHERE Рейтинг = 4

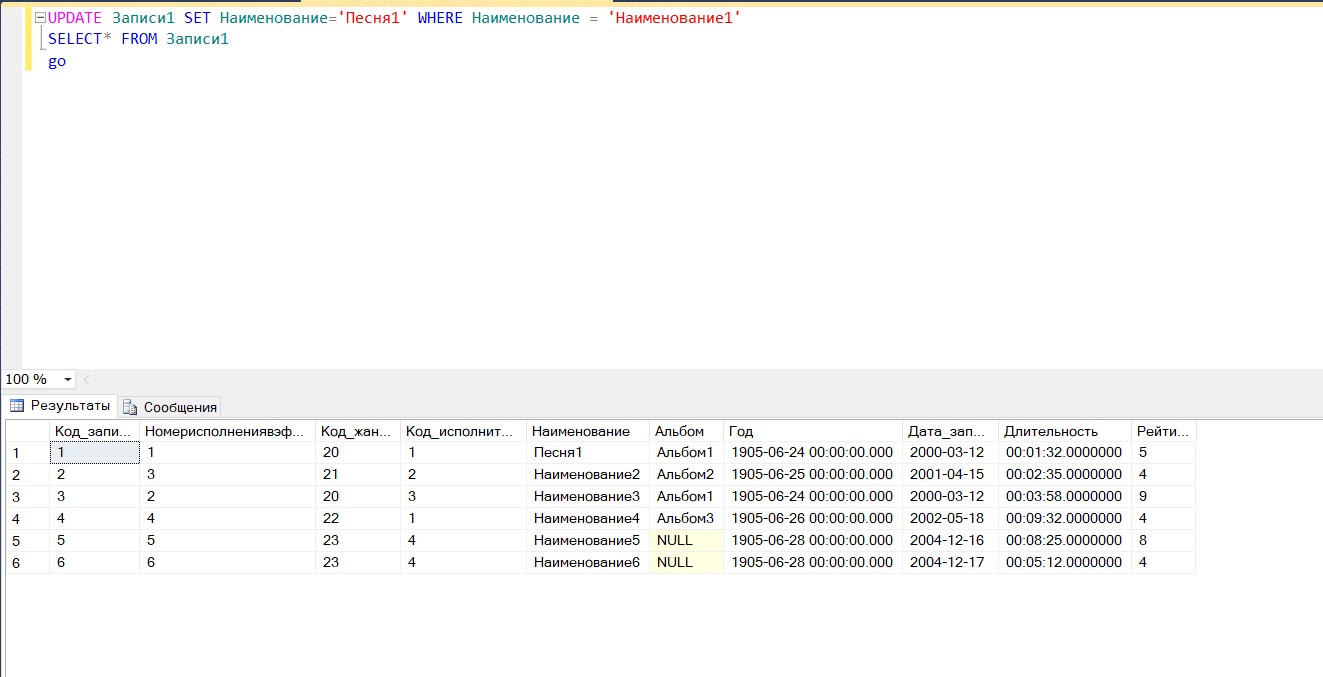
WITH CHECK OPTION;



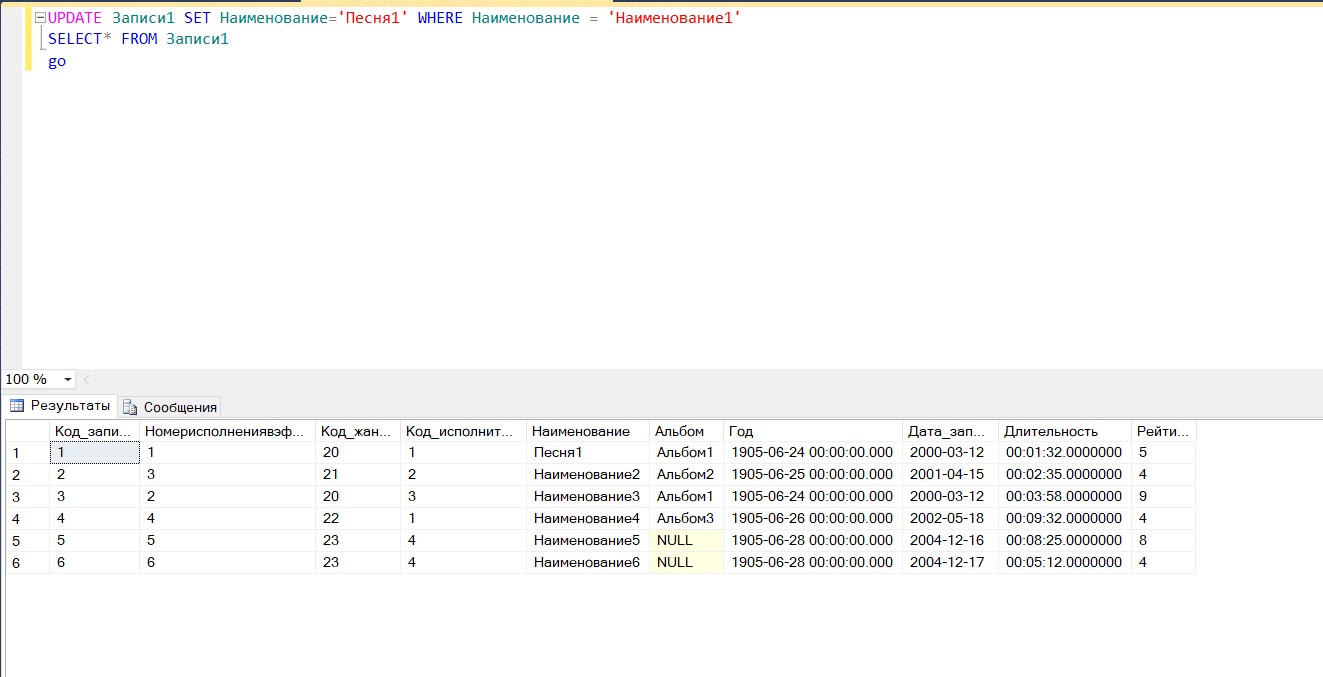
Ошибка при попытке вставки или обновления, поскольку целевое представление либо указывает WITH CHECK OPTION, либо охватывает представление, которое указывает WITH CHECK OPTION, а одна или несколько строк, получающиеся при операции, не определены в рамках ограничения CHECK OPTION.

При использовании опции WITH CHECK OPTION вставка строки в представление, не соответствующей фильтру представления, отвергается.

**37.**



**40.**



**43.**

**Удаление через подзапрос.**

DELETE T

from Датыпремьеры1 T

where Код\_записи= (

select top 1 Код\_записи

from Датыпремьеры1

where Номерисполнениявэфире = T.Номерисполнениявэфире

order by Код\_записи desc

)

**44.**

