Лабораторная работа 7

Представления (VIEW)

Представления или Views

Представляют собой виртуальные таблицы. Но в отличии от обычных стандартных таблиц в базе данных представления содержат запросы, которые динамически извлекают используемые данные.

Преимущества

Представления дают нам ряд преимуществ. Они упрощают комплексные SQL-операции. Они защищают данные, так как представления могут дать доступ к части таблицы, а не ко всей таблице. Представления также позволяют возвращать отформатированные значения из таблиц в нужной и удобной

Типы представлений

Кроме основных определяемых пользователем представлений, выполняющих стандартные роли, в SQL Server предусмотрены следующие типы представлений, которые соответствуют специальным назначениям в базе данных.

- Индексированные представления
- Секционированные представления
- Системные представления

Модифицируемое представление

В SQL есть такое понятие как модифицируемое представление — это означает, что при изменении данных в самом представлении, эти данные изменятся и в таблицах, которые эти данные хранят. То есть при использовании оператора UPDATE/INSERT/DELETE к представлению, данные обновятся и в таблицах.

Критерии (1)

по которые определяют, является ли представление модифицируемым:

- Оно должно выводиться в одну и только в одну базовую таблицу.
- Оно должно содержать первичный ключ этой таблицы (это технически не предписывается стандартом ANSI, но было бы неплохо придерживаться этого).
- Оно не должно иметь никаких полей, которые бы являлись агрегатными функциями.
- Оно не должно содержать DISTINCT в своем определении.
- Оно не должно использовать GROUP BY или HAVING в своем определении.

Критерии (2)

- Оно не должно использовать подзапросы (это - ANSI_ограничение которое не предписано для некоторых реализаций)
- Оно может быть использовано в другом представлении, но это представление должно также быть модифицируемыми.
- Оно не должно использовать константы, строки, или выражения значений (например: comm * 100) среди выбранных полей вывода.
- Для INSERT, оно должно содержать любые пол основной таблицы которые имеют ограничение NOT NULL, если другое ограничение по умолчанию, не определено.

Например, пусть у нас есть три связанных таблицы:

```
CREATE TABLE Products
(
    Id INT IDENTITY PRIMARY KEY,
    ProductName NVARCHAR(30) NOT NULL,
    Manufacturer NVARCHAR(20) NOT NULL,
    ProductCount INT DEFAULT 0,
    Price MONEY NOT NULL
);
```

```
CREATE TABLE Customers
(
    Id INT IDENTITY PRIMARY KEY,
    FirstName NVARCHAR(30) NOT NULL
);
```

```
CREATE TABLE Orders
(
    Id INT IDENTITY PRIMARY KEY,
    ProductId INT NOT NULL REFERENCES Products(Id) ON DELETE CASCADE,
    CustomerId INT NOT NULL REFERENCES Customers(Id) ON DELETE CASCADE,
    CreatedAt DATE NOT NULL,
    ProductCount INT DEFAULT 1,
    Price MONEY NOT NULL
);
```

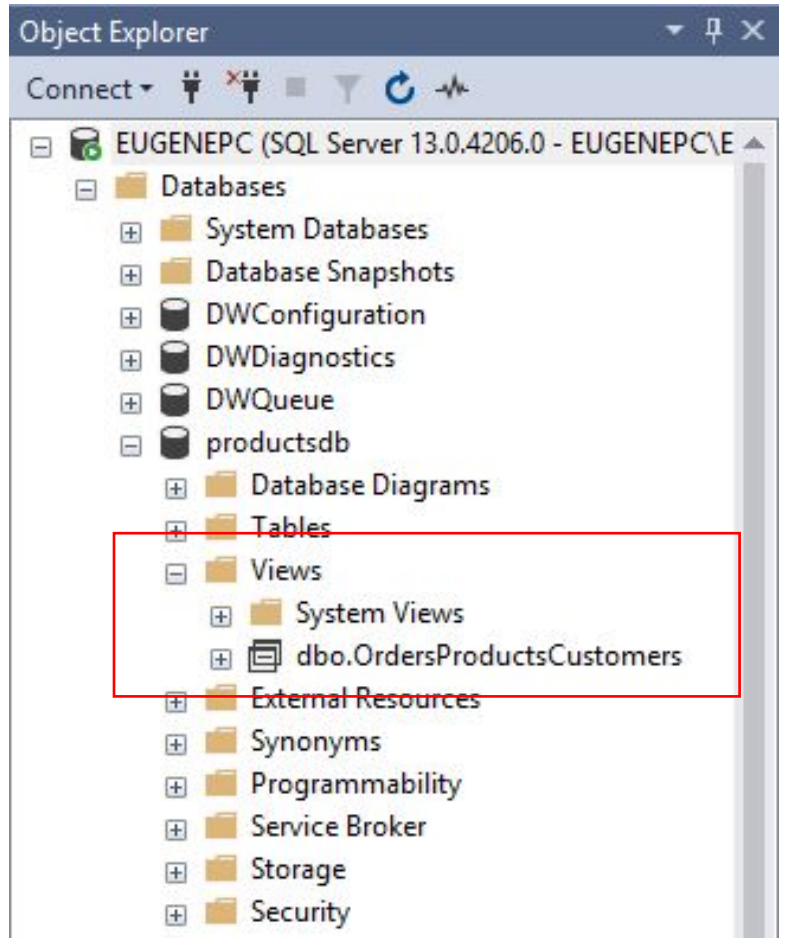
Для **создания представления** используется команда **CREATE VIEW**, которая имеет следующую форму:

CREATE VIEW название_представления [(столбец_1, столбец_2, ...)]

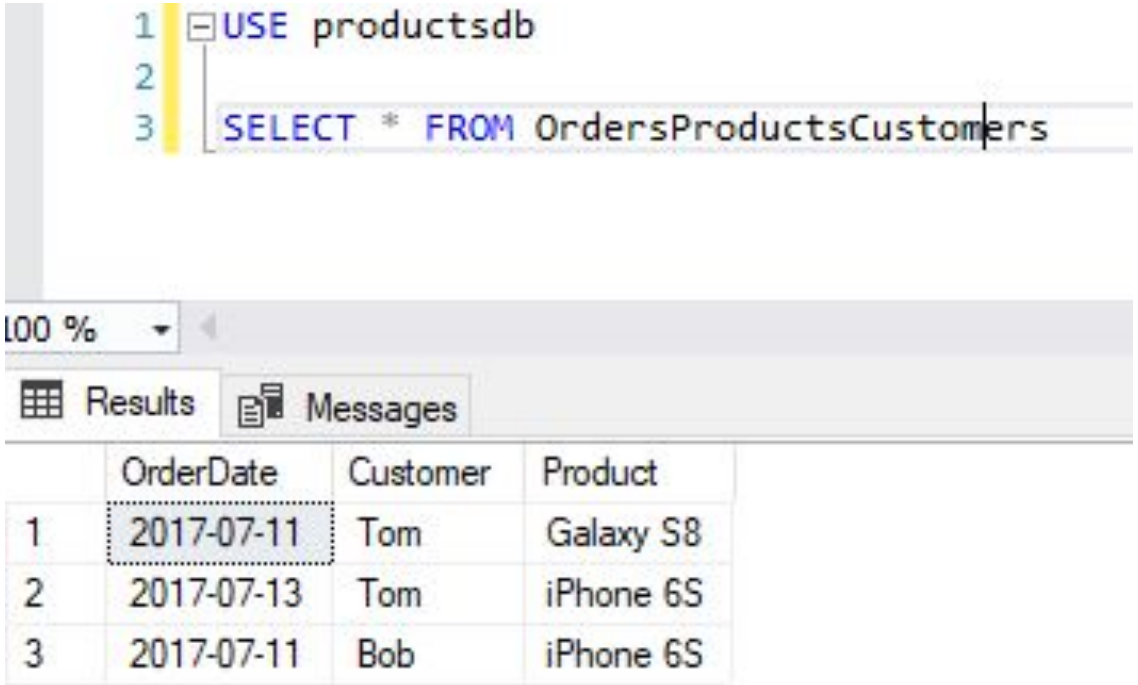
AS выражение_SELECT

Теперь добавим в базу данных, в которой содержатся данные таблицы, следующее представление:

```
CREATE VIEW OrdersProductsCustomers AS
SELECT Orders.CreatedAt AS OrderDate,
        Customers.FirstName AS Customer,
        Products.ProductName As Product
FROM Orders INNER JOIN Products ON Orders.ProductId = Products.Id
INNER JOIN Customers ON Orders.CustomerId = Customers.Id
```



SELECT * FROM OrdersProductsCustomers



	OrderDate	Customer	Product
1	2017-07-11	Tom	Galaxy S8
2	2017-07-13	Tom	iPhone 6S
3	2017-07-11	Bob	iPhone 6S

Изменение представления

Для изменения представления используется команда **ALTER VIEW**. Эта команда имеет практически тот же самый синтаксис, то и CREATE VIEW:

```
ALTER VIEW название_представления [(столбец_1, столбец_2, ....)]
AS выражение_SELECT
```

Например, изменим выше созданное представление OrdersProductsCustomers:

```
ALTER VIEW OrdersProductsCustomers
AS SELECT Orders.CreatedAt AS OrderDate,
        Customers.FirstName AS Customer,
        Products.ProductName AS Product,
        Products.Manufacturer AS Manufacturer
FROM Orders INNER JOIN Products ON Orders.ProductId = Products.Id
INNER JOIN Customers ON Orders.CustomerId = Customers.Id
```

Удаление представления

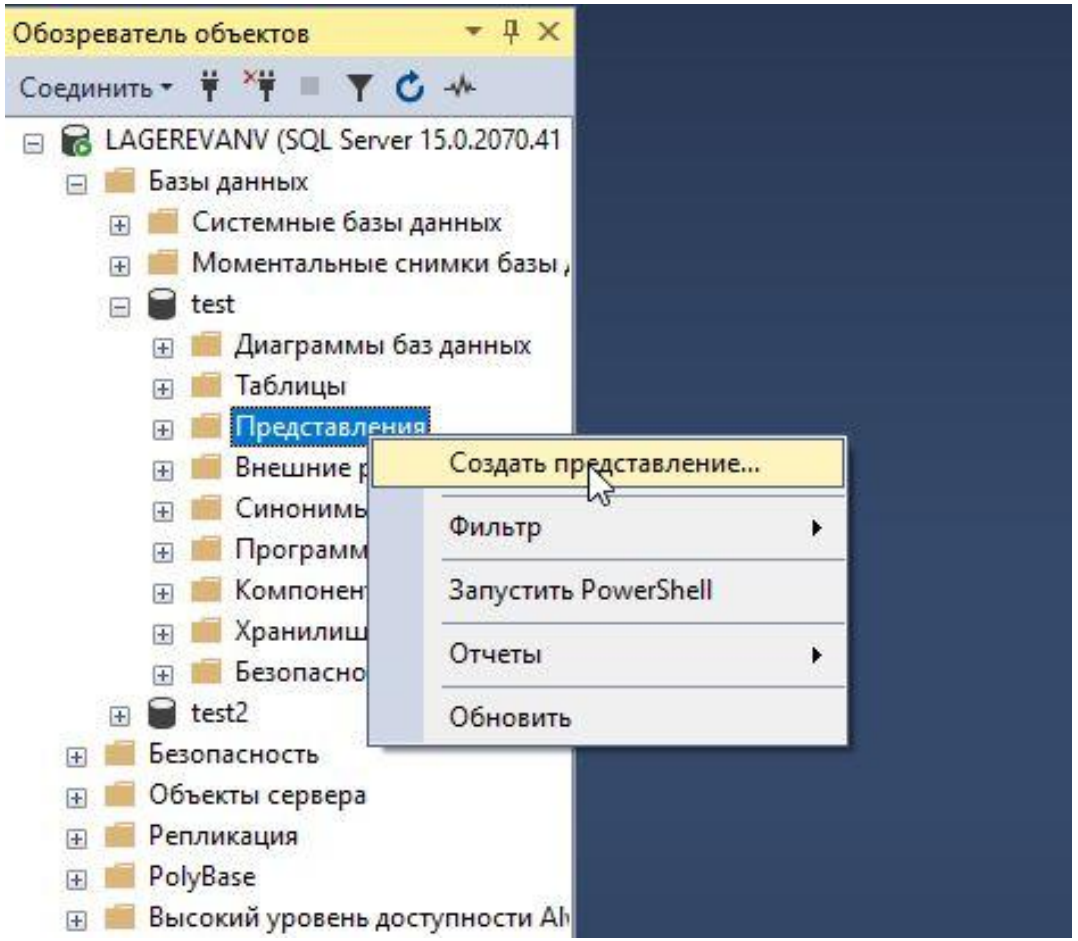
Для удаления представления вызывается команда **DROP VIEW**:

```
DROP VIEW OrdersProductsCustomers
```

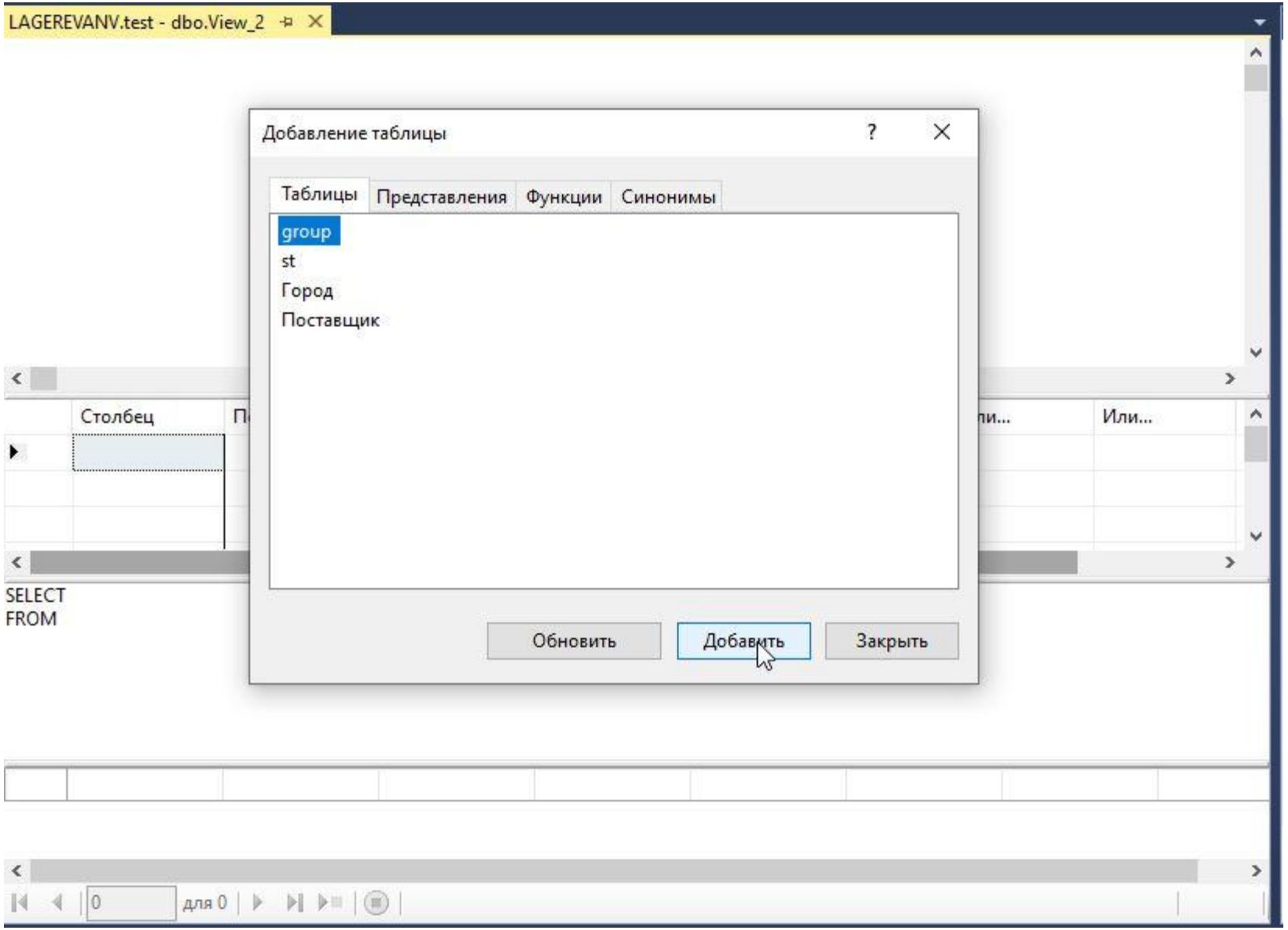
Также стоит отметить, что при удалении таблиц также следует удалить и представления, которые используют эти таблицы.

Представления в Management Studio

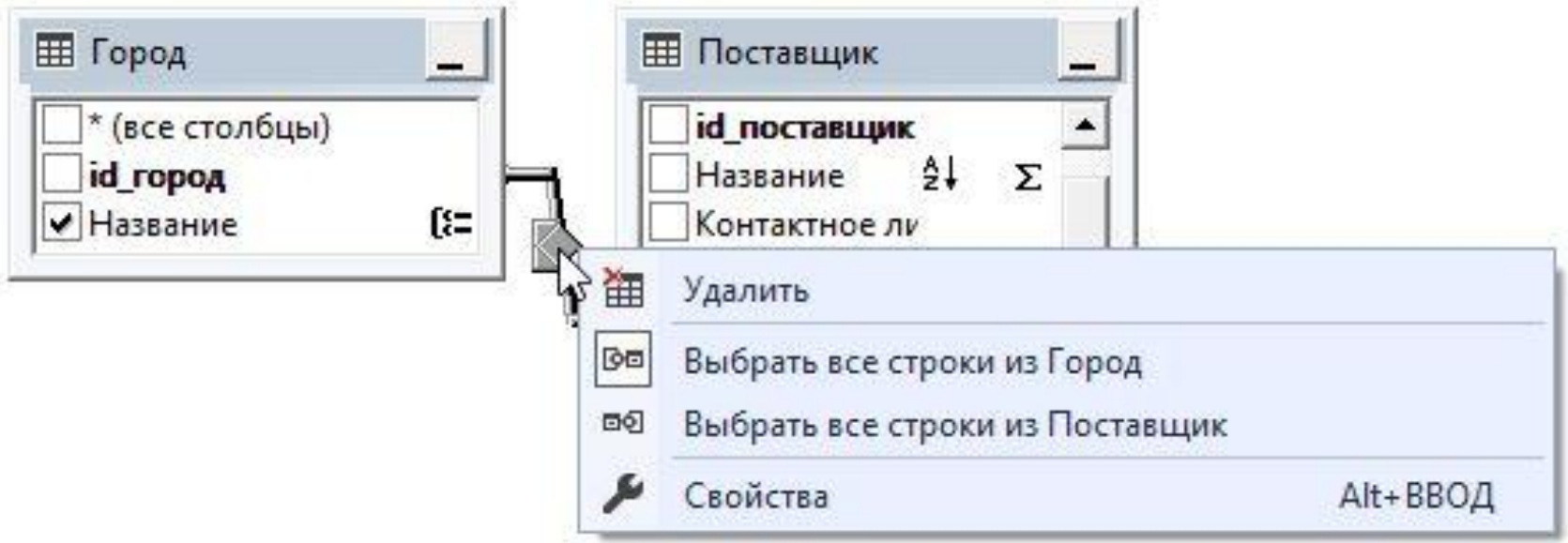
1)



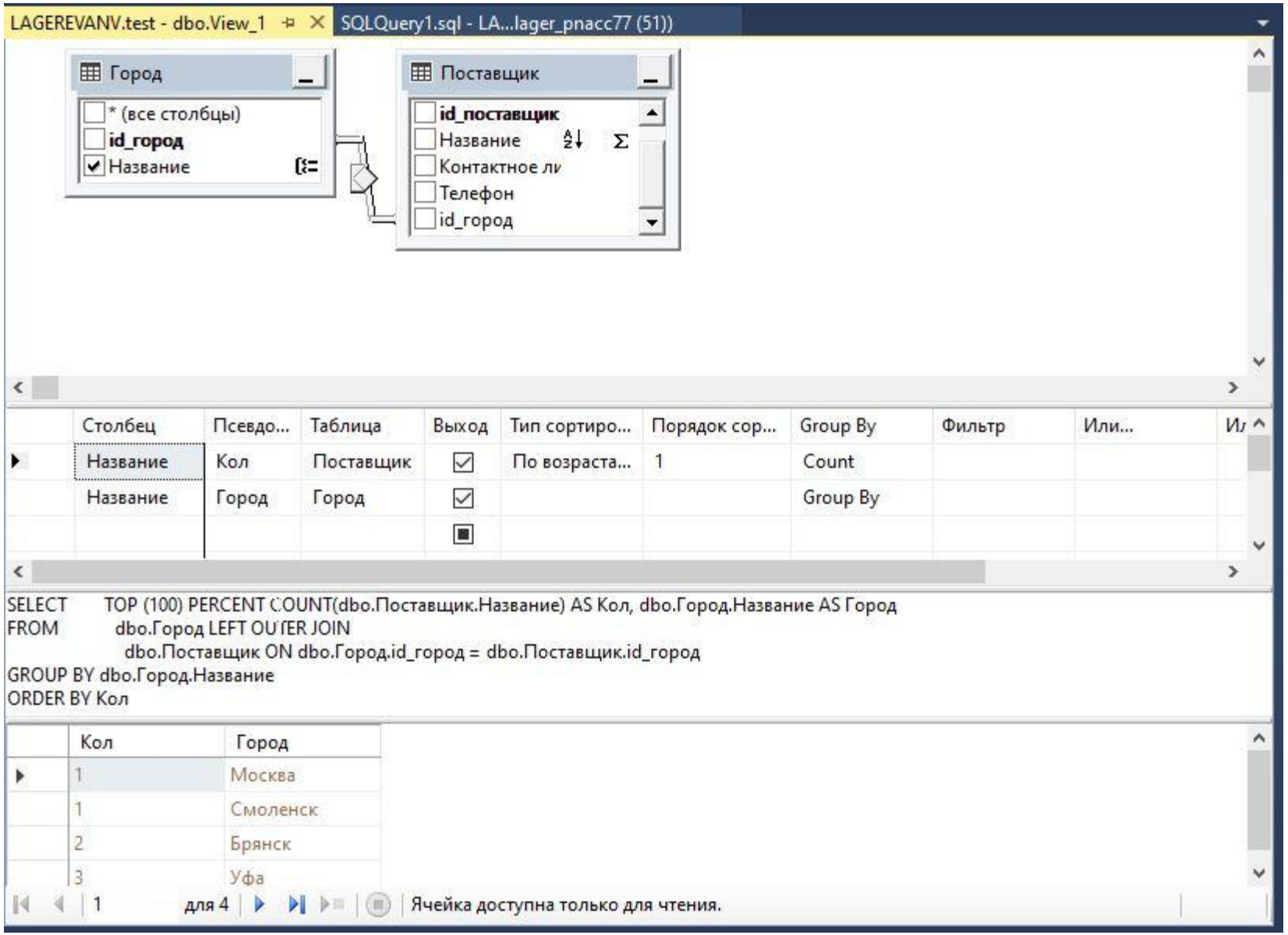
2)


LAGEREVANV.test - dbo.View_2
Добавление таблицы
Таблицы
Представления
Функции
Синонимы
group
st
Город
Поставщик
Обновить
Добавить
Закрыть
Столбец
Или...
SELECT
FROM
0
для 0

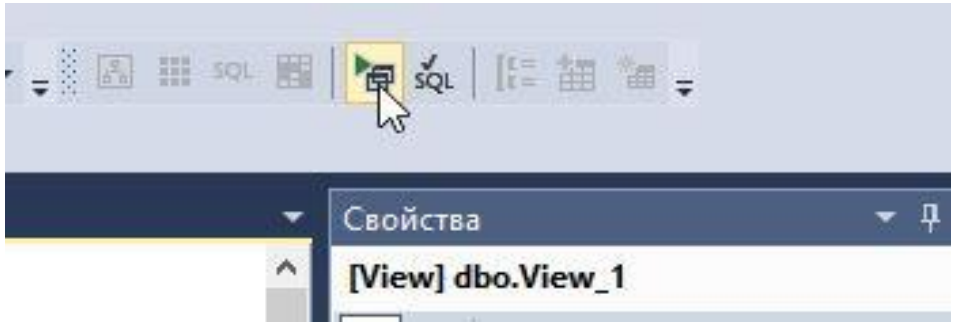
3)


Город
* (все столбцы)
id_город
Название
Поставщик
id_поставщик
Название
Контактное ли
Удалить
Выбрать все строки из Город
Выбрать все строки из Поставщик
Свойства
Alt+ВВОД

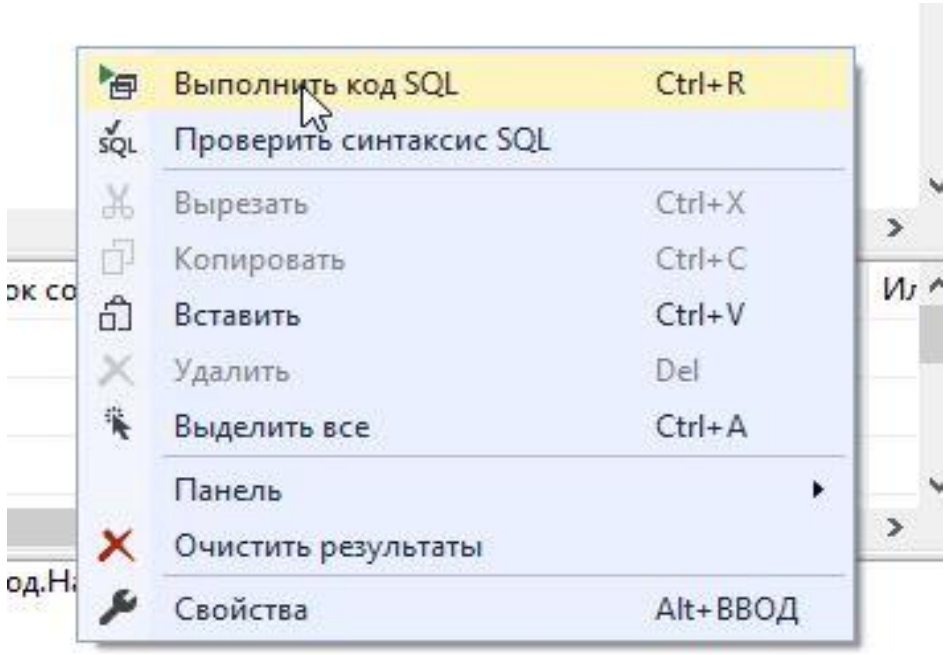
4)


LAGEREVANV.test - dbo.View_1
SQLQuery1.sql - LA...lager_pnacc77 (51))
Город
* (все столбцы)
id_город
Название
Поставщик
id_поставщик
Название
Контактное ли
Телефон
id_город
Столбец | Псевдо... | Таблица | Выход | Тип сортиро... | Порядок сор... | Group By | Фильтр | Или...
Название | Кол | Поставщик | Название | Город | Город | По возраста... | 1 | Count | Group By
SELECT TOP (100) PERCENT COUNT(dbo.Поставщик.Название) AS Кол, dbo.Город.Название AS Город
FROM dbo.Город LEFT OUTER JOIN
dbo.Поставщик ON dbo.Город.id_город = dbo.Поставщик.id_город
GROUP BY dbo.Город.Название
ORDER BY Кол
Кол | Город
1 | Москва
1 | Смоленск
2 | Брянск
3 | Уфа
1 для 4
Ячейка доступна только для чтения.

5)


Свойства
[View] dbo.View_1

или


Выполнить код SQL Ctrl+R
Проверить синтаксис SQL
Вырезать Ctrl+X
Копировать Ctrl+C
Вставить Ctrl+V
Удалить Del
Выделить все Ctrl+A
Панель
Очистить результаты
Свойства Alt+ВВОД

Задания к защите лабораторной работы 7:

1. Создать представление для своей БД.

Требования:

✔Не менее 2-х таблиц

✔Для столбцов псевдонимы на русском языке

✔Использовать группировку и/или несколько фильтров

✔Сортировка

Задания к защите лабораторной работы 7:

2. Определить какое из этих представлений - модифицируемое и объя

```
CREATE VIEW Dateorders (odate, ocount)
    AS SELECT odate, COUNT (*)
       FROM Orders
       GROUP BY odate;
```

```
CREATE VIEW Someorders
   AS SELECT snum, onum, cnum
      FROM Orders
      WHERE odate IN (10/03/1990,10/05/1990);
```

```
CREATE VIEW Londoncust
   AS SELECT *
      FROM Customers
      WHERE city = 'London';
```

```
CREATE VIEW SJsales (name, number, percentage)
    AS SELECT sname, snum, comm  100
       FROM Salespeople
       WHERE city = 'SanJose';
```

```
CREATE VIEW Salesonthird
    AS SELECT *
       FROM Salespeople
       WHERE snum IN
          (SELECT snum
             FROM Orders
             WHERE odate = 10/03/1990);
```