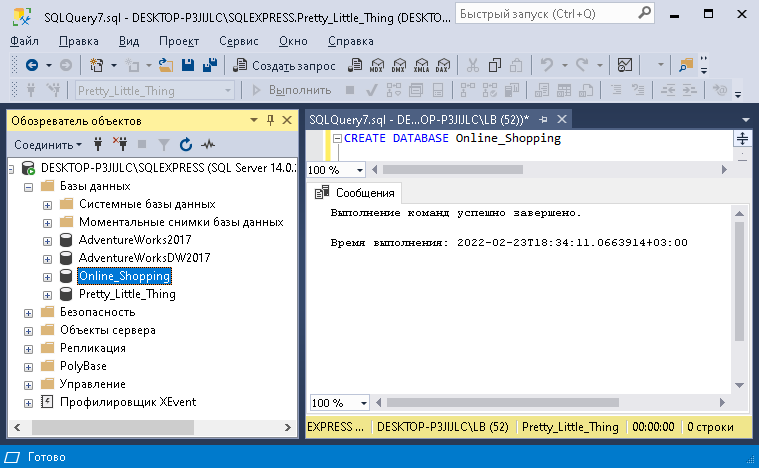
**TASK\_1**

Создайте вашу базу данных: таблицы, ограничения, ключи.

CREATE DATABASE Online\_Shopping



CREATE TABLE Customer

(CustomerID INT,

FirstName NVARCHAR (50),

LastName NVARCHAR (50),

BirthDate DATE,

Gender NVARCHAR (3),

Address NVARCHAR (50),

City NVARCHAR (50),

Country NVARCHAR (50),

PhoneNumber NVARCHAR (50),

Email NVARCHAR (30),

CONSTRAINT PK\_Customer\_CustomerID PRIMARY KEY (CustomerID)

)

CREATE TABLE Product

(ProductID INT,

ProductName NVARCHAR (50) NOT NULL,

ManufacturerName NVARCHAR (50),

ManufacturerNum INT,

ProductPrice MONEY,

Size NVARCHAR (4),

Color NVARCHAR (10),

GenderCategory NVARCHAR (10),

ClothesCategory NVARCHAR (10),

FirstSaleDate DATE,

CONSTRAINT PK\_Product\_ProductID PRIMARY KEY (ProductID),

CONSTRAINT PK\_Product\_ManufacturerNum UNIQUE (ManufacturerNum)

)

CREATE TABLE SalesPerson

(SalesPersonID INT,

FirstName NVARCHAR (50),

LastName NVARCHAR (50),

HireDate DATE,

PhoneNumber NVARCHAR (50),

CONSTRAINT PK\_SalesPerson\_SalesPersonID PRIMARY KEY (SalesPersonID)

)

CREATE TABLE Shipper

(ShipperID INT,

ShipperName NVARCHAR (50),

ShipMethod NVARCHAR (30),

CONSTRAINT PK\_Shipper\_ShipperID PRIMARY KEY (ShipperID)

)

CREATE TABLE Orders

(OrderID INT,

ProductID INT NOT NULL,

OrderQty INT,

Discount FLOAT,

OrderDade DATE,

SalesPersonID INT NOT NULL,

CustomerID INT NOT NULL,

CONSTRAINT PK\_Orders\_OrderID PRIMARY KEY (OrderID),

CONSTRAINT FK\_Orders\_ProductID FOREIGN KEY (ProductID) REFERENCES Product (ProductID),

CONSTRAINT FK\_Orders\_SalesPersonID FOREIGN KEY (SalesPersonID) REFERENCES SalesPerson (SalesPersonID),

CONSTRAINT FK\_Orders\_CustomerID FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)

)

CREATE TABLE Delivery

(DeliveryID INT,

CustomerID INT NOT NULL,

OrderID INT NOT NULL,

ShipperID INT NOT NULL,

DeliveryStatus NVARCHAR (30),

AffectedDate DATE,

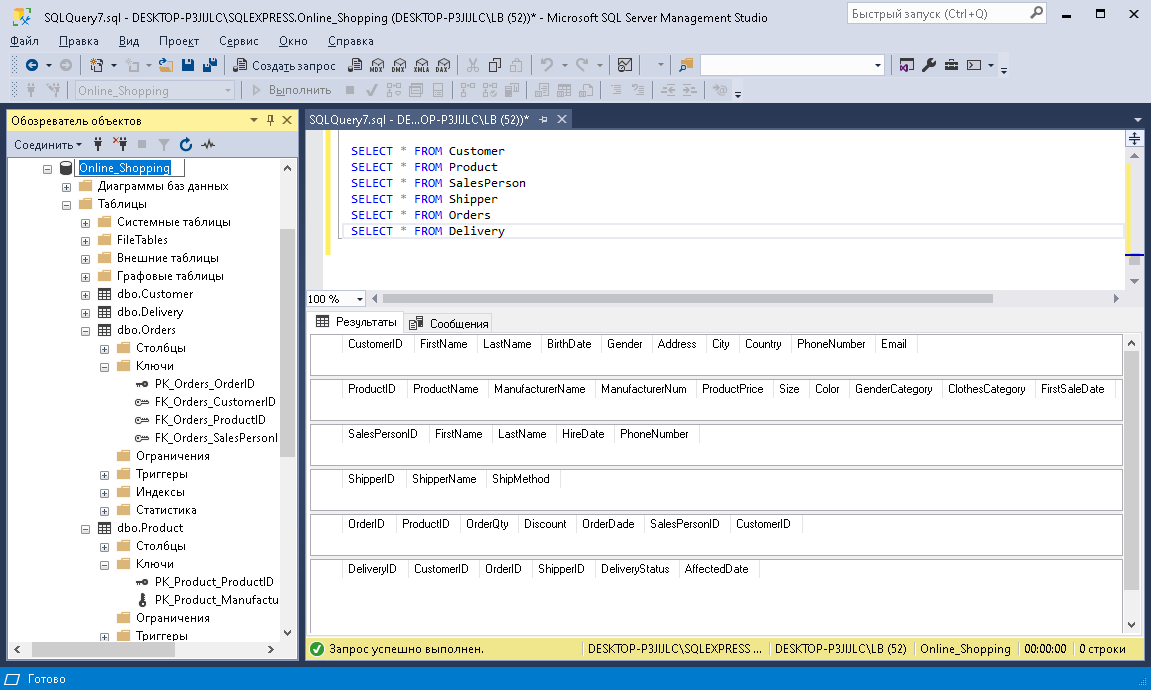
CONSTRAINT PK\_Delivery\_DeliveryID PRIMARY KEY (DeliveryID),

CONSTRAINT FK\_Delivery\_CustomerID FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID),

CONSTRAINT FK\_Delivery\_OrderID FOREIGN KEY (OrderID) REFERENCES Orders (OrderID),

CONSTRAINT FK\_Delivery\_ShipperID FOREIGN KEY (ShipperID) REFERENCES Shipper (ShipperID)

)



**Task\_2**

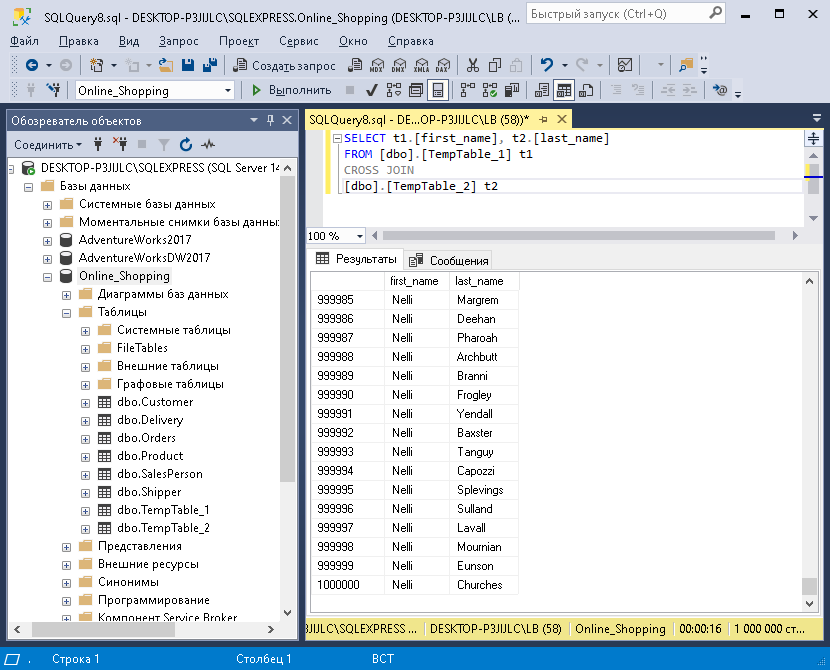
Напишите скрипт для получения 1 млн человек с различными именами и фамилиями.

SELECT t1.[first\_name], t2.[last\_name]

FROM [dbo].[TempTable\_1] t1

CROSS JOIN

[dbo].[TempTable\_2] t2



\*Для получения меньшего количества вариантов комбинаций FirstName+LastName в данном примере можно использовать оператор TOP.

SELECT TOP 999999 t1.[first\_name], t2.[last\_name]

FROM [dbo].[TempTable\_1] t1

CROSS JOIN

[dbo].[TempTable\_2] t2

**Task\_3**

Познакомьтесь с [обобщенным табличным выражением WITH](https://docs.microsoft.com/ru-ru/sql/t-sql/queries/with-common-table-expression-transact-sql?view=sql-server-ver15). Когда можем использовать данную структуру? Чем отличается от подзапроса?

Обобщенное табличное выражение (ОТВ) – это временные результирующие наборы (результаты выполнения SQL запроса), которые не сохраняются в БД в виде объектов, но к ним можно обращаться. Данную структуру можем использовать для улучшения читаемости запроса (уменьшает размер кода); в случаях, когда нужно много раз обращаться к одним и тем же таблицам/выборкам из таблиц; для создания представлений (VIEW) в части select; для написания рекурсивных запросов. От подзапроса ОТВ отличается тем, что оно может быть рекурсивным, т.е. включать ссылки на само себя. Рекурсивные ОТВ используются для запроса иерархических данных.