**Задания**

1. Создайте вашу базу данных: таблицы, ограничения, ключи.

CREATE TABLE Dim\_Customer

(

CustomerID INT IDENTITY(1,1),

CONSTRAINT PK\_CustomerID PRIMARY KEY (CustomerID),

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

BirthDate DATE,

Gender NVARCHAR(10),

Address NVARCHAR(50),

City NVARCHAR(50),

Country NVARCHAR(50),

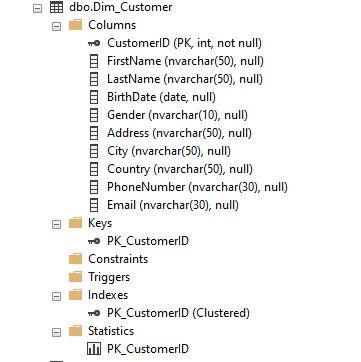
PhoneNumber NVARCHAR(30),

Email NVARCHAR(30)

);

SELECT\*

FROM Dim\_Customer



CREATE TABLE Dim\_Salesperson

(

SalespersonID INT IDENTITY(1,1),

CONSTRAINT PK\_SalePersonID PRIMARY KEY (SalespersonID),

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

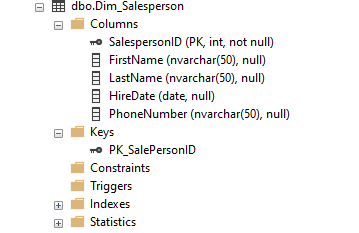
HireDate DATE,

PhoneNumber NVARCHAR(50)

);

SELECT\*

FROM Dim\_Salesperson



CREATE TABLE Dim\_Product

(

ProductID INT IDENTITY(1,1),

CONSTRAINT PK\_ProductID PRIMARY KEY (ProductID),

ProductName NVARCHAR(50),

ManufacturerName NVARCHAR(50),

ManufacturerNum INT,

CONSTRAINT UQManufactNum UNIQUE (ManufacturerNum),

ProductPrice MONEY NOT NULL,

Size NVARCHAR (10),

Color NVARCHAR (20),

GenderCategory NVARCHAR(10),

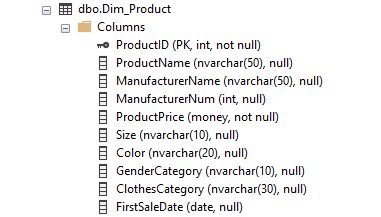
ClothesCategory NVARCHAR(30),

FirstSaleDate DATE

);

SELECT\*

FROM Dim\_Product



CREATE TABLE Dim\_Shipper

(

ShipperID INT IDENTITY(1,1),

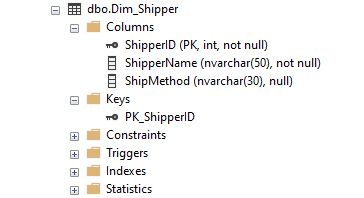
CONSTRAINT PK\_ShipperID PRIMARY KEY (ShipperID),

ShipperName NVARCHAR(50) NOT NULL,

ShipMethod NVARCHAR(30)

);

SELECT \* FROM Dim\_Shipper



CREATE TABLE Fct\_Order

(

OrderID INT IDENTITY(1,1),

CONSTRAINT PK\_OrderID PRIMARY KEY (OrderID),

ProductID INT NOT NULL,

CONSTRAINT FK\_Order\_Product FOREIGN KEY (ProductID) REFERENCES Dim\_Product(ProductID),

OrderQty INT NOT NULL,

Discount FLOAT,

OrderDate DATE,

SalespersonID INT NOT NULL,

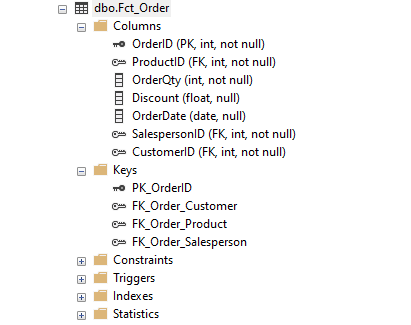
CONSTRAINT FK\_Order\_Salesperson FOREIGN KEY (SalespersonID) REFERENCES Dim\_Salesperson(SalespersonID),

CustomerID INT NOT NULL,

CONSTRAINT FK\_Order\_Customer FOREIGN KEY (CustomerID) REFERENCES Dim\_Customer(CustomerID)

);

SELECT \* FROM Fct\_Order



CREATE TABLE Fct\_Delivery

(

DeliveryID INT IDENTITY(1,1),

CONSTRAINT PK\_DeliveryID PRIMARY KEY (DeliveryID),

CustomerID INT NOT NULL,

CONSTRAINT FK\_Delivery\_Customer FOREIGN KEY (CustomerID) REFERENCES Dim\_Customer(CustomerID),

OrderID INT NOT NULL,

CONSTRAINT FK\_Delivery\_Order FOREIGN KEY (OrderID) REFERENCES Fct\_Order(OrderID),

DeliveryDate DATE,

ShipperID INT NOT NULL,

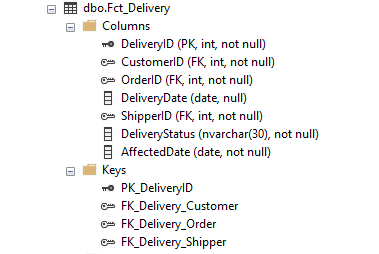
CONSTRAINT FK\_Delivery\_Shipper FOREIGN KEY (ShipperID) REFERENCES Dim\_Shipper(ShipperID),

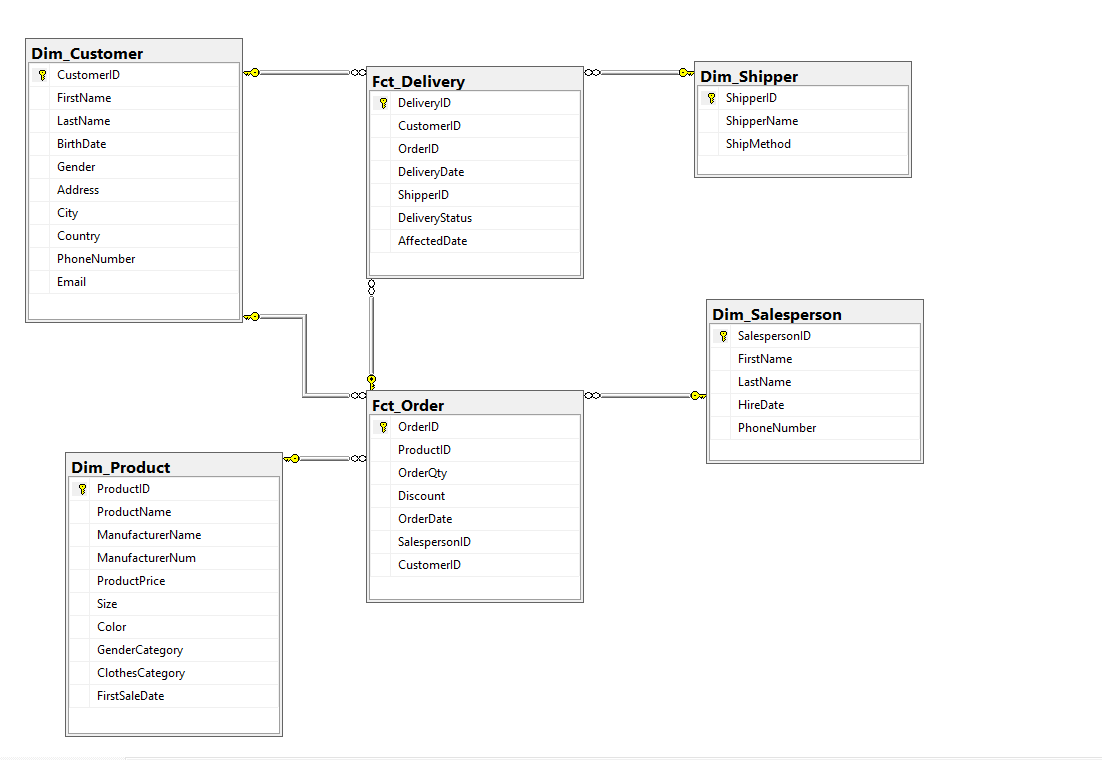
DeliveryStatus NVARCHAR(30) NOT NULL,

AffectedDate DATE NOT NULL

);

SELECT \* FROM Fct\_Delivery





1. Напишите скрипт для получения 1 млн человек с различными именами и фамилиями. Полезная [ссылка](https://mockaroo.com/).

SELECT TOP 1000000

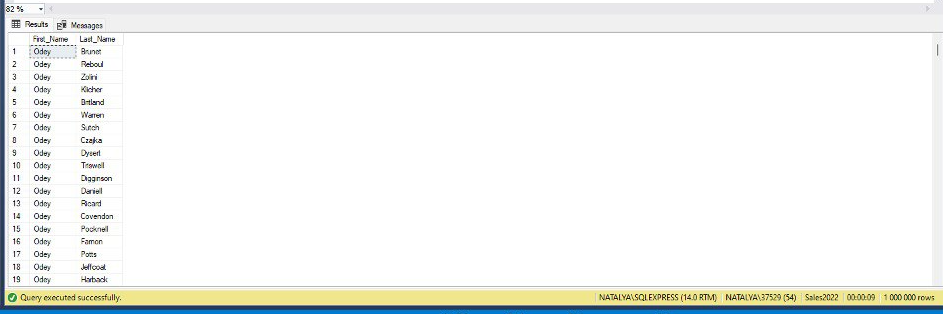
A1.First\_Name,

A2.Last\_Name

FROM dbo.RandomNames A1

CROSS JOIN

RandomNames A2



1. 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?

Обобщённые табличные выражения (временные результирующие запросы) можем использовать:

1. Как альтернатива view
2. При необходимости неоднократно ссылаться на результирующий набор.
3. Для создания рекурсивных запросов
4. Для улучшения читабельности запроса для сложных запросов

**Общее с подзапросом: не сохраняются в БД в виде объектов, время жизни ограничено продолжительностью запроса.**

**Отличие от подзапроса: могут ссылаться сами на себя, на них один и тот же запрос может ссылаться несколько раз.**