**Задание 1.**

CREATE DATABASE Shop;

CREATE TABLE Dim\_Shipper (

ShipperID int identity (1,1),

CONSTRAINT PK\_ShipperID PRIMARY KEY (ShipperID),

ShipperName nvarchar(50),

ShipMethod nvarchar(30)

);

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,

Size nvarchar(10),

Color nvarchar(20),

GenderCategory nvarchar(10),

ClothesCategory nvarchar(30),

FirstSaleDate date

);

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)

);

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)

);

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)

);

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),

ShipperID int NOT NULL,

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

DeliveryStatus nvarchar(30) NOT NULL,

AffectedDate date NOT NULL

);

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

**Задание 2.**

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

SELECT

T1.first\_name,

T2.last\_name

FROM dbo.Test1 T1

CROSS JOIN

dbo.Test2 T2

Изображение выглядит как текст

Автоматически созданное описание

**Задание 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?

**Common Table Expression** (CTE) или **обобщенное табличное выражение** (OTB) – это временные результирующие наборы (т.е. результаты выполнения SQL запроса), которые не сохраняются в базе данных в виде объектов, но к ним можно обращаться.

Основные способы использования:

* для улучшения читаемости запроса в случае сложных запросов (разительно уменьшают 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.

После обобщенного табличного выражения, т.е. сразу за ним должен идти одиночный запрос **SELECT, INSERT, UPDATE, MERGE или DELETE**.

**В отличие от производных таблиц, ОТВ могут ссылаться сами на себя, а на них один и тот же запрос может ссылаться несколько раз.**