**Домашняя работа №8**

**Архитектура MS SQL**

**Задания**

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

CREATE DATABASE SalesTours;

use [SalesTours];

drop table if exists Managers\_dim;

CREATE TABLE Managers\_dim (

IDManager int identity (1,1) not null,

FirstName nvarchar (40),

LastName nvarchar (40),

Position nvarchar(20),

Phone nvarchar(20),

Email nvarchar(30),

CONSTRAINT manager\_dim\_pk PRIMARY KEY (IDManager)

);

drop table if exists Clients\_dim;

CREATE TABLE Clients\_dim (

IDClient int identity (1,1) not null,

IdentificationNumber nvarchar(30) not null unique,

FirstName nvarchar (40),

LastName nvarchar (40),

BirthDate Date,

Gender nvarchar(10),

Phone nvarchar(20),

Email nvarchar(30),

CONSTRAINT client\_dim\_pk PRIMARY KEY (IDClient)

);

drop table if exists Offices\_dim;

CREATE TABLE Offices\_dim (

IDOffice int identity (1,1) not null,

City nvarchar (20),

Address nvarchar (90),

CONSTRAINT office\_dim\_pk PRIMARY KEY (IDOffice)

);

drop table if exists Tours\_dim;

CREATE TABLE Tours\_dim (

IDTour int identity (1,1) not null,

TourType nvarchar(50),

HotelName nvarchar (30),

HotelRating int,

Country nvarchar (50),

City nvarchar (30),

Address nvarchar (50),

CateringType nvarchar (30),

Transfer nvarchar (30),

MaxQuantity int,

TransportType nvarchar (30),

TransportClass nvarchar (30),

CONSTRAINT tour\_dim\_pk PRIMARY KEY (IDTour)

);

drop table if exists Sales\_fct;

CREATE TABLE Sales\_fct(

IDSale int identity (1,1) not null,

ContractNumber nvarchar(30) not null unique,

DataSales Date,

IDManager int,

IDTour int,

IDClient int,

IDOffice int,

Costs money,

SellingPrice money,

DiscountRate smallmoney,

TourCapacity int,

TourStartDate date,

TourFinishDate date,

CONSTRAINT sale\_fct\_pk PRIMARY KEY (IDSale),

CONSTRAINT sale\_fct\_manager\_fk

FOREIGN KEY (IDManager) REFERENCES [dbo].[Managers\_dim] (IDManager) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT sale\_fct\_tour\_fk

FOREIGN KEY (IDTour) REFERENCES [dbo].[Tours\_dim] (IDTour) ON DELETE CASCADE ON UPDATE CASCADE,

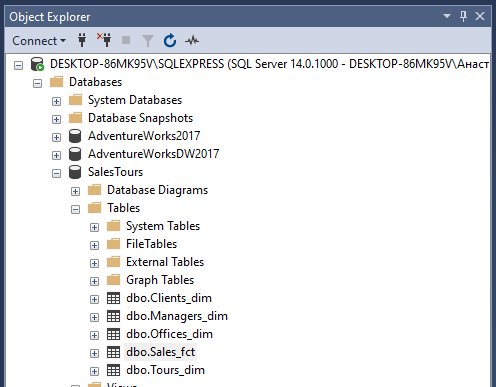
CONSTRAINT sale\_fct\_client\_fk

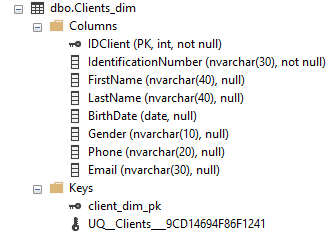
FOREIGN KEY (IDClient) REFERENCES [dbo].[Clients\_dim] (IDClient) ON DELETE CASCADE ON UPDATE CASCADE,

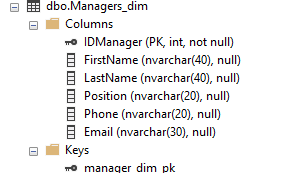
CONSTRAINT sale\_fct\_office\_fk

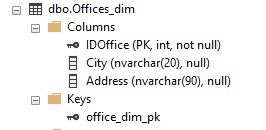
FOREIGN KEY (IDOffice) REFERENCES [dbo].[Offices\_dim] (IDOffice) ON DELETE CASCADE ON UPDATE CASCADE

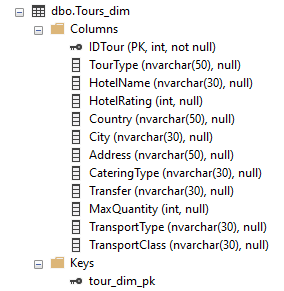
);

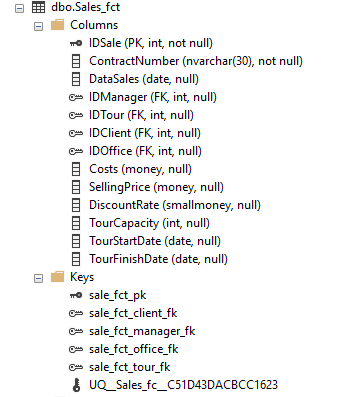












1. Напишите скрипт для получения 1 млн человек с различными именами и фамилиями.

Полезная [ссылка](https://mockaroo.com/).

2.1. Создала тестовую таблицу – «залила» в нее 1000 строк.

CREATE DATABASE TEMP;

CREATE TABLE TEST

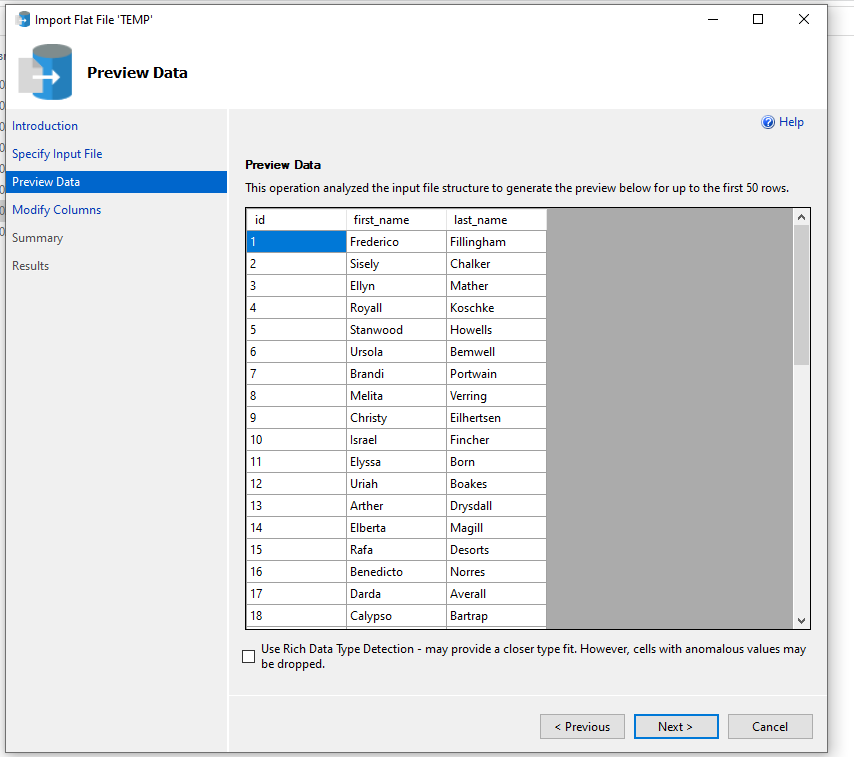
(

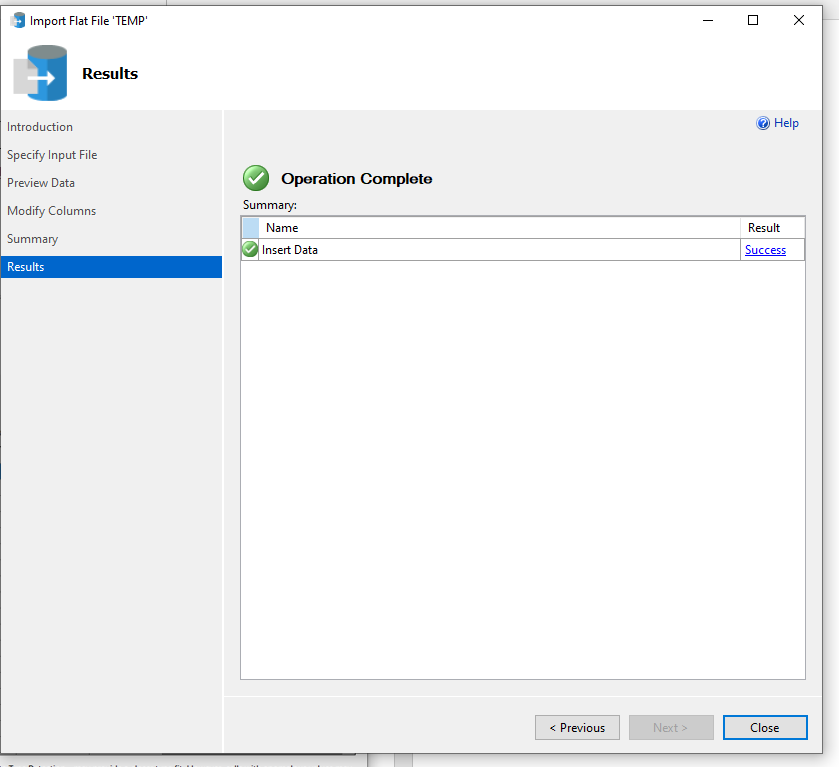
id INT not null,

first\_name NVARCHAR(100),

last\_name NVARCHAR(100),

);





2.2 скрипт для получения 1 млн человек с различными именами и фамилиями

SELECT TOP 1000000

c1.[first\_name],

c2.[last\_name]

FROM [dbo].[TEMP] c1

CROSS JOIN

[dbo].[TEMP] c2;

2.3. создала вторую тестовую таблицу и залила в нее 1000000 строк

CREATE TABLE TEST2

(

id INT identity not null,

first\_name NVARCHAR(100),

last\_name NVARCHAR(100),

);

INSERT INTO [dbo].[TEST2]([first\_name],[last\_name])

SELECT TOP 1000000

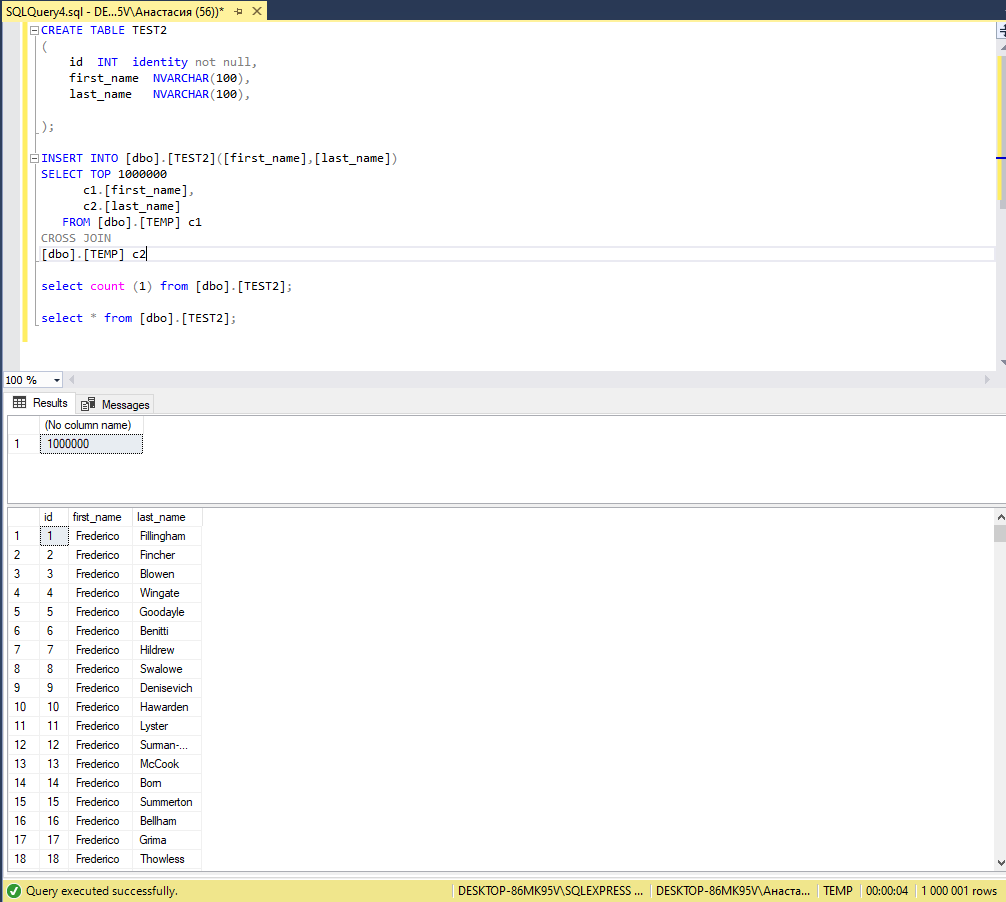
c1.[first\_name],

c2.[last\_name]

FROM [dbo].[TEMP] c1

CROSS JOIN

[dbo].[TEMP] c2



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?

Common Table Expression (CTE) — результаты запроса, которые можно использовать множество раз в других запросах. То есть, запросом мы достаем данные, и они помещаются в пространство памяти, аналогично временному представлению, которое физически не сохраняется в виде объектов. Далее мы работаем с получившейся конструкцией как с таблицей, используя такие конструкции как select, update, insert и delete.

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.

Отличия от вложенного запроса:

* вложенный запрос повторяется для каждой строки из нашей выборки, что повышает стоимость выполнения запроса.
* Пример. Теперь допустим, что нам необходимо вывести иерархический список сотрудников, т.е. мы хотим видеть, на каком уровне работает тот или иной сотрудник. Для этого пишем рекурсивный запрос:
* 