1. **Создайте вашу базу данных: таблицы, ограничения, ключи.**
2. *Создаем БД*

CREATE DATABASE MyDB\_test

1. Поочередно создаем таблицы с необходимыми типами данных и ограничениями (согл. физической схемы)

CREATE TABLE Competition (

Competition\_id INT IDENTITY(1,1) CONSTRAINT PK\_Competition\_id PRIMARY KEY,

Date\_start DATETIME NOT NULL,

Swimmer\_id INT NOT NULL,

Location\_id INT NOT NULL,

ResultTime TIME NOT NULL

)

CREATE TABLE Location (

Location\_id INT IDENTITY(1,1) CONSTRAINT PK\_Location\_id PRIMARY KEY,

Location\_name NVARCHAR(50) NOT NULL,

City\_name NVARCHAR(50) NOT NULL

)

CREATE TABLE Swimmer (

Swimmer\_id INT IDENTITY(1,1) CONSTRAINT PK\_Swimmer\_id PRIMARY KEY,

Trainer\_id INT NOT NULL,

FirstName NVARCHAR(50) NOT NULL,

LastName NVARCHAR(50) NOT NULL,

PassportNumber NVARCHAR(20) NOT NULL CONSTRAINT UQ\_PassportNumber UNIQUE,

Birthday DATE,

Gender NVARCHAR(1) NOT NULL,

Rank NVARCHAR(10),

Team\_id INT NOT NULL

)

CREATE TABLE Bridge\_Competition\_Judge (

Judge\_id INT NOT NULL,

Competition\_id INT NOT NULL

)

CREATE TABLE Judges (

Judge\_id INT IDENTITY(1,1) CONSTRAINT PK\_Judge\_id PRIMARY KEY,

FirstName NVARCHAR(50) NOT NULL,

LastName NVARCHAR(50) NOT NULL,

PassportNumber NVARCHAR(20) NOT NULL CONSTRAINT UQ\_PassportNumberJudge UNIQUE,

SwimStyle NVARCHAR(30),

Rank NVARCHAR(20),

TypesOfOficcial NVARCHAR(30)

)

CREATE TABLE Trainer (

Trainer\_id INT IDENTITY(1,1) CONSTRAINT PK\_Trainer\_id PRIMARY KEY,

FirstName NVARCHAR(50) NOT NULL,

LastName NVARCHAR(50) NOT NULL,

PassportNumber NVARCHAR(20) NOT NULL CONSTRAINT UQ\_PassportNumberTrainer UNIQUE,

Rank NVARCHAR(10)

)

CREATE TABLE Team (

Team\_id INT IDENTITY(1,1) CONSTRAINT PK\_Team\_id PRIMARY KEY,

Captain NVARCHAR(50) NOT NULL,

MembersNumber INT NOT NULL,

Country NVARCHAR(30),

TeamName NVARCHAR(30) NOT NULL,

TeamDate DATE

)

1. Добавляем ограничения по внешним ключам FK в таблицах (согл. схемы)

ALTER TABLE Competition ADD CONSTRAINT FK\_Location\_id FOREIGN KEY (Location\_id) REFERENCES Location (Location\_id)

ALTER TABLE Competition ADD CONSTRAINT FK\_Swimmer\_id FOREIGN KEY (Swimmer\_id) REFERENCES Swimmer (Swimmer\_id)

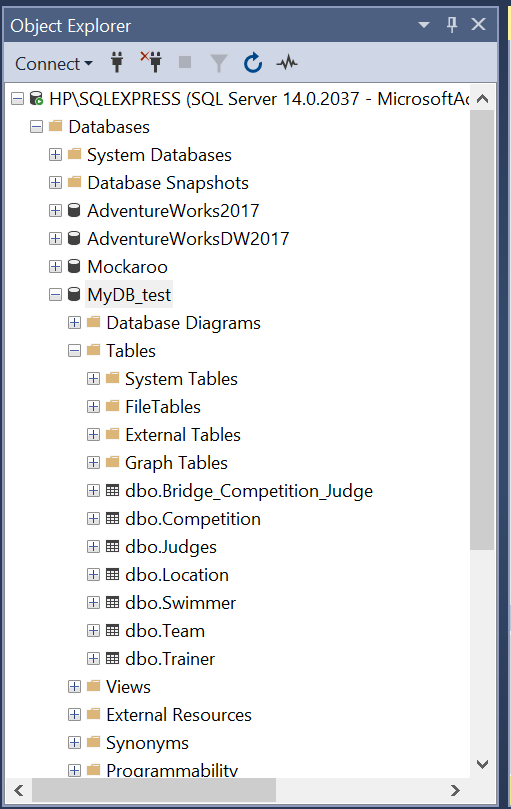
ALTER TABLE Bridge\_Competition\_Judge ADD CONSTRAINT FK\_Judge\_id FOREIGN KEY (Judge\_id) REFERENCES Judges (Judge\_id)

ALTER TABLE Bridge\_Competition\_Judge ADD CONSTRAINT FK\_Competition\_id FOREIGN KEY (Competition\_id) REFERENCES Competition (Competition\_id)

ALTER TABLE Swimmer ADD CONSTRAINT FK\_Trainer\_id FOREIGN KEY (Trainer\_id) REFERENCES Trainer (Trainer\_id)

ALTER TABLE Swimmer ADD CONSTRAINT FK\_Team\_id FOREIGN KEY (Team\_id) REFERENCES Team (Team\_id)

1. Итоговый вид дерева базы



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

1. С помощью сервиса <https://mockaroo.com/> создаем 2 таблицы .csv. В 1-й таблице имеем 1000 сгенерированных значений lastname, во 2-й таблице - 1000 сгенерированных значений firstname.
2. Создаем новую БД

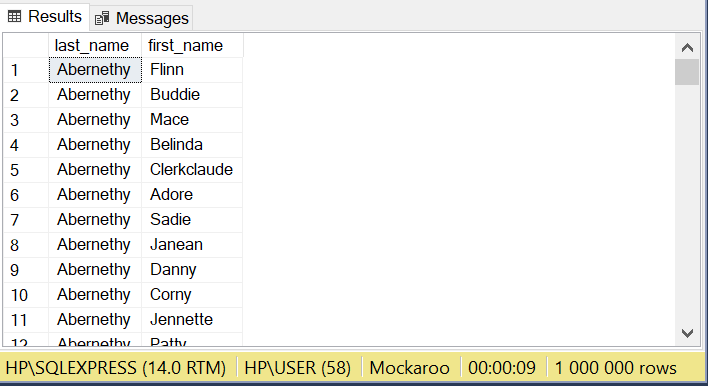
CREATE DATABASE Mockaroo

1. Загружаем в нашу созданную БД 2 файла .csv с lastname и firstname (делал через интерфейс MS SQL Studio)
2. Используем функцию CROSS JOIN (декартово произведение, которое, в нашем случае, перемножит 1000 значений lastname на 1000 значений firstname, в итоге чего получим 1000000 различных комбинаций фамилий и имени)

SELECT \*

FROM [dbo].[Last\_name] CROSS JOIN [dbo].[First\_name]

ORDER BY last\_name



**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?**

Честно, не до конца разобрался в данном вопросе. Понял, что можем использовать для замены VIEW, когда нет необходимости сохранять запрос в базе. Отличие от подзапроса в том, что запрос с WITH более читаемый, т.к. разделен на логические блоки