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



USE AdventureWorks2017

GO

--Change 1000000 to the number of your preference for your needs

SELECT TOP 1000000

M1.[first\_name],

M2.[last\_name]

FROM [dbo].[MOCK\_DATA] M1

CROSS JOIN

[dbo].[MOCK\_DATA] M2

* 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?

A Common Table Expression (CTE), also referred to as a WITH clause, is a temporary named result set that you can reference anywhere in your query. **In contrast to subqueries,** which are inserted exactly where you need them, all CTEs are defined before the main query and are then referenced in the query using the assigned name.

- CTEs Use Meaningful Names

You see the query in the order it will be executed: first the subquery and then the main query. You can determine the purpose of the subquery based on its name.

- CTEs Are Reusable Within a Query

In contrast to subqueries, you don’t have to repeat a CTE definition each time you need it in the query. You define it only once, at the beginning of your query, and then reference it when necessary.

- CTEs Divide Complex Computations into Parts

CTEs bring clarity to the computation process. When subqueries are used for computations, the query often turns into a tangled mess of subqueries. CTEs, on the other hand, show the computation process more clearly.

- CTEs Allow Recursion

Last but not least, CTEs are great at processing graphs, trees, and other hierarchical structures. This is because the WITH syntax can process recursion. A recursive query is a query that refers to itself.

Исходя из всего выше, может использовать CTE в некоторых кейсах, где есть рекурсия, или сложноый подзапрос, чтобы использовать вычисляемое значение в клаузе where и тд