Common Table Expressions - Chapter 1

The following has been republished from my Common Table Expressions book.

Chapter 1. Memory Tables

There was a 1996 Michael Keaton movie called *Multiplicity* where he duplicates himself to get more work done. It first started out that he just needed help to finish up his work and be able to attend his child's ballgame. The theme of the movie shows us that duplicating yourself has costs that you need to think about. Fortunately, if the same data in your SQL Server needs to be in two places at once you don't need to copy the data to a new table. For many years there have been options to virtualize the same result set without duplicating the data. One way is to use the Common Table Expression (CTE). This book will cover this topic extensively and compare CTEs to the existing ways of virtualizing your data.

READER NOTE: *Most chapters in this book will have follow along tutorial steps. These steps are listed in a* ***READER NOTE*** *at the beginning of the chapter. The script for this book is available at http://SteveStedman.com.*

Virtual Tables

We often have the need to look in the mirror when getting ourselves ready for the day. We check our appearance so that we know how the world will see us, and we can make adjustments if needed. While you are alone standing in front of the mirror it appears that there are two people in the room. In fact, there is just you and a virtual image of you reflected in the mirror.

Like the reflection in a mirror, we can reflect data from one or more tables into a new result set for use in our SQL Server. This data is not a copy but merely a reflection coming from the old data. We

can even add to this data with an expression or our own custom labels. Many techniques have long existed to virtualize data from objects in SQL server.

Views

At first, when we are talking about virtual tables, people think we are talking about views. That is often true since a view is basically a query from a table saved as a new name. If we create a view that queries from an employee table and call it vEmployee, then it is easy to query this view. Simply type a SELECT statement (just like a regular query) from the view.

```
--This will query the Employee table
SELECT * FROM Employee
```

```
--This will query the vEmployee view
SELECT * FROM vEmployee
```

Views have been around a long time and are very well known. If we want to create a new view we will need permission in the database to create new objects. Once those objects are in place many other people can use them. Views are particularly handy when you want to abstract some business logic that is commonly replicated in queries. Save that in the view then it is available for easy reuse.

Temporary Named Result Sets

A table that we want to use and never save will basically be loaded and run in memory for as long as we need it. If we have a view that only we need (and nobody else) then maybe we don’t need a new view. We can use some form of temporary named result set. The most commonly known temporary result set is a temp table.

Temp tables will not put new objects into the database for other people to see or evaluate. One drawback to a temp table is we still need permission to create tables in order to create temp tables. There are also some performance considerations with temp tables depending on the server you are working with.

Temp tables have their use, and I have often run into people asking me the question of why I recommend CTE versus table tables, and we will get into that later. I want to be sure that you understand CTE's are not replacement for temp tables, instead there are another tool that can be used along side temp tables.

There are some temporary objects (including Common Table Expressions) that we can use that don't require any Create permissions to run. You will see how this works in Chapter 2.

Common Table Expressions

Common Table Expressions were introduced in SQL 2005 and are the newest form of an in-memory result set. If so many temporary named result sets and virtual tables have been around for so long, why do we need Common Table Expressions? Think of it this way: have you ever seen a software make-over program where your picture can be taken and you can instantly see what you would look like in dozens of different outfits? You could say this is an improvement to a changing room and a mirror since you can try out many ideas in far less time.

Common Table Expressions have many of the advantages of other memory based tables but also have options and power not seen in the other virtual type tables. This book will help you see how all these types work and what cool tricks you can do with the Common Table Expression.

Points to Ponder – **Memory Tables**

1. For years there have been options to virtualize the same result set without duplicating the data.
2. A view is often called a virtual table and is basically a query from a table saved as a new name.
3. SQL has many temporary named result set options which are basically memory based tables.
4. Common Table Expressions are another form of in-memory result sets.
5. Common Table Expressions are not a replacement for these items, but instead a tool to use along side temp tables and views.

Review Quiz - Chapter One

1. A view is often called:
2. A temporary named result set.
3. A virtual table.
    1. What version of SQL Server introduced the Common Table Expression?
4. SQL Server 2000
5. SQL Server 2005
6. SQL Server 2019
7. SQL Server 2017

Answer Key

1. A view queries just like a table and is called a virtual table. This makes (b) the correct answer.
2. The CTE was introduced in SQL Server 2005. This makes (b) is the correct answer.