

Common Table Expressions

CS 2550 Spring 2025
Cody Henrichsen

1

Why

- ▶ You may have noticed in some questions on the assignments that you need to use the same data in multiple different clauses in the same query
- ▶ You can just write that same subquery in each of the clauses where you need that source of information but for some reason that doesn't really seem like the best plan of action
- ▶ This where a Common Table Expression (CTE) AKA a named query can save your day!

2

Syntax

- ▶ The syntax for a CTE is pretty easy. First, figure out the query that you want to write!
- ▶ Before the query add the keyword WITH and then a variable name to reference the query you just wrote.
 - ▶ Using good naming conventions
 - ▶ Remember you will probably want to alias it as well
- ▶ Then the keyword AS
- ▶ Parens around your subquery
- ▶ Then the query that references the CTE
 - ▶ Remember to include a valid PK/FK for the JOIN / ON

3

CTE Usage

The diagram illustrates the structure of a SQL query using a Common Table Expression (CTE). It shows a 'CTE source' block at the top, which defines the CTE. Below it, the 'CTE references' block shows how the CTE is used in the main query. Arrows indicate the flow from the source to the references.

4

Stackable!!

- Wait, what if I need multiple data sets multiple times? Do I use the WITH keyword again
- No, just append a comma after the parens for the first (or previous) named query and give a new name and query definition!

5

Stacked CTE Usage

The diagram shows a SQL query with two stacked CTEs. 'CTE #1' is defined first, and 'CTE #2' is defined second, separated by a comma. Arrows indicate the flow of data from CTE #1 to CTE #2 and then to the main query.

6
