Eric Dee

Module 3 Query Optimization

10/1/2022

***1.A Finding missing index information (A.1-3)***

Text

Description automatically generated with medium confidence

***Solved missing index (Step 1.A.3 completion):***

Text

Description automatically generated

***1.4. Miscellaneous index creation***

Graphical user interface, text, application, email

Description automatically generated

***2. A.***

Graphical user interface, text, application

Description automatically generated

***2.B***

Index is a reserved keyword. I chose to use the single responsibility principle, and build around the QueryForIndexes method. I also wanted to experiment with Type definition. I would probably just define the table in the procedure using “declare as table” were I to write this again.

Graphical user interface, text

Description automatically generated

***3.***

I like sticking to the single responsibility principle here for two reasons. The first reason is that the cost to keep my queries portable remains relatively low in this instance, as the initial procedure is a simple query, and the second reason is that I found it separates the batch display nicely. For something like dbo operations, I don’t see this causing an issue for anyone but those who live on the bleeding edge.

Graphical user interface, text, application

Description automatically generated

The selection query costs 40% of the entire sequence, and the usp\_Index query is higher at 57% due to the local table being created, but that only takes up 18% of the entire function, and the temporary table I cannot see getting large enough to worry about. SQL also seems to be able to determine that these inserts can be run in parallel, as there is an index scan taking place on both of them within the same sequence.