

CS 510 - Data Engineering

11 February 2021

Answers for Data Storage Activity**Part C - Baseline - Simple INSERT**

Oregon Sample Data in 1.698s

acs2015 Data in 122.9s

Part D - The Effects of Indexes and Constraints

Without Index - 81.56s

With Index - 84.12s

I know that its supposed to improved performance but in my case it didn't. But I have used indexes before and I am aware that they are very helpful in very large datasets.

Part E - The Effects of Logging

UnLogged Table creation with acs2015 data took 13.49s

Part F - Temp Tables and Memory Tuning

acs2015 Temporary Table created in 11.81s

After changing temp_buffers to 256MB, the table took 13.03s to get created.

Part G - Batching

acs2015 data took 10.46s

Part H - Built in Facility

acs2015 data took 0.4777s

Part I - Results

Method	Code Link	Acs2015
Simple Insert	load_inserts.py	122.9s
Drop Indexes and Constrains	load_inserts.py	With Index - 84.12s
Use UnLogged Table	load_inserts.py	13.49s
Temp Table with memory tuning	load_inserts.py	11.81s, 13.03s
Batching	load_inserts_execute_batches.py	10.48s

Copy_from	load_inserts_copy_from.py	0.4777s
-----------	---------------------------	---------

Part J - Observations

This has been a very useful activity. We had a debate regarding the database performance in our project and now I think I can help my team next week we work on anything similar. Reducing the number of hits to a Database will reduce the load on the database as well. In our project we are having many hits to the database. Probably we can improve this now.