

# CS 166 Lab 7 Assignment: Indexing

The purpose of the following assignment is to explore how indexes can be used to improve the performance of SQL queries, the different types of indexes supported by Postgres, and the issues involved around indexes and the optimizer.

Having read chapter 8 in the textbook will help you understand many aspects of this lab.

For this assignment you will need to download and unzip lab7.zip from Canvas (eLearn).

## Tasks

1. Unzip the lab 7 zip folder and use scp to copy the folder to cs166 server. You should not be ssh into cs166 server for this command. Be sure to replace the text in red with the appropriate information.

```
scp -r /path/to/lab/ <netid>@cs166.cs.ucr.edu:~/
```

2. Login into cs166 server, and navigate to lab 7 folder. Run the start and create scripts to set up your environment. Do **not** run cs166\_db\_start, the startPostgreSQL.sh script already does that for you.

```
cd cs166_lab7_updated
```

```
source startPostgreSQL.sh
```

```
source createPostgreDB.sh
```

3. Write your SQL queries for Exercise 1 below in the queries.sql file. Execute your queries.sql script using the following command and take a screenshot of the output.

```
cs166_psql $USER'_lab7_DB' < queries.sql
```

4. Follow Exercise 2 below and write the index statements in create\_indexes.sql script.

5. The measure.sh script creates the necessary tables and runs queries.sql script and create\_indexes.sql script for you. Use this to test your queries and indices.

Execute the measure.sh script using the following command, and take a screenshot of the output. You only need to include the screenshot from one run.

```
source measure.sh
```

6. Create a pdf file. Experiment with B+ tree indexes built on different columns. Run the measure.sh script **at least 5 times**. In your pdf, record the execution times as well as the average execution times before and after index creation. You may use this template if you would like, or create something similar: [Lab 7 Template](#)

7. Make sure to stop the server when you are done with the lab.

```
source stopPostgreSQL.sh
```

## Exercise 1 (SQL)

Write the following SQL queries in queries.sql **both with and without** using indexes:

1. Count how many parts in NYC have more than 70 parts on\_hand
2. Count how many total parts on\_hand, in both NYC and SFO, are Red
3. List all the suppliers that have more total on\_hand parts in NYC than they do in SFO.
4. List all suppliers that supply parts in NYC that aren't supplied by anyone in SFO.
5. Update all of the NYC on\_hand values to on\_hand - 10.
6. Delete all parts from NYC which have less than 30 parts on\_hand.

Some notes about the data model and the questions:

- part\_number is the primary key for each part table. But it is not unique across both tables.
- If a part has the same number in NYC and SFO it is the same part, regardless of color, etc.
- If I say, e.g. "Red parts", I mean color\_name = "Red" not color = 0.
- Different suppliers may supply the same part in NYC and SFO.

## Exercise 2 (Indexes)

If you want to measure execution time of individual query you need to execute "\timing" command in interactive psql terminal. I have provided you with a measure.sh script, which measures this time for you. Execution time can vary depending on the multiple factors, hence be sure that you report **average** time across several executions.

All your queries should have been written in queries.sql. File create\_indexes.sql should contain the relevant drop index and create index statements for all **the indexes you decide are best for the queries** you have written. DON'T create all possible indexes unnecessarily.

Notice that the last two queries are changing the data. To make the experiment reasonable, make sure that your queries run on the same data, with and without indexes.

You should experiment with B-tree indexes, built on different columns. Once you have experimented and are satisfied with the indexes you chose, follow the instructions mentioned in step 6 to record your results.

Use the following syntax for create index statement:

```
CREATE INDEX index_name
ON table_name
[USING BTREE]
(index_col_name)
```

At the top of your create\_indexes.sql, use the following syntax for drop index statement:

```
DROP INDEX index_name;
```

You can find the Postgres documentation on indexes at

<http://www.postgresql.org/docs/8.1/static/indexes.html>

## Turn-in:

Submit the following 3 files on eLearn (do NOT zip the files):

- 1) Exercise 1 queries in **queries.sql** script
- 2) Exercise 2 indexes in **create\_indexes.sql** script
- 3) **lab7.pdf** file with the following
  - a) 2 taken screenshots
  - b) Experiment results including the averages

\*You may use this template if you would like, or create something similar: [Lab 7 Template](#)