

Assignment 1

Uploading Data to the Database

Database Tuning

Presentation and Demo: March 6, 2020
Report due: March 20, 2020, 23:55
Grading: 5 points

Notes

- This assignment involves reading the documentation of Java, the JDBC driver, and PostgreSQL. Finding the relevant sources of information is part of the challenge.
- It is suggested that you also have a look at the report template before you start working on the assignment.
- In the third part of this assignment, you are supposed to set up a DBMS on your own machine. The recommendation is to use MariaDB within Apache XAMPP.

Access Parameters for PostgreSQL

- Host: `biber.cosy.sbg.ac.at`
- Port: 5432
- Database: `dbtuning_ss2020`
- User/Password: you should have received them via email

The database server (`biber.cosy.sbg.ac.at`) is accessible only from within the university network. If you would like to work from home, please connect to `fanny.cosy.sbg.ac.at` via `ssh`. Java, the PostgreSQL client, and Python are installed on this machine.

Download `dblp.zip` at:

<https://dbresearch.uni-salzburg.at/downloads/teaching/2018ss/dbt/dblp.zip>

This archive contains two tab separated files (`publ.tsv` and `auth.tsv`) that store authors and their publications as found in the DBLP¹ bibliography. The imported tables have the following schemas:

- `Auth(name(49),pubID(129))`
- `Publ(pubID(129),type(13),title(700),booktitle(132),year(4),publisher(196))`

You can assume that all attribute values are strings; the maximum string length is shown in brackets.

The straightforward algorithm to load the data from the TSV file to a table issues an SQL `INSERT` query for each line in the TSV file.

¹<http://dblp.uni-trier.de/db/>

Straightforward Implementation

Implement the straightforward approach to load `auth.tsv` to the database (PostgreSQL, Java/Python).²

Efficient Approaches

The straightforward approach is slow. There are other approaches that are significantly faster. Figure out how the efficient approaches work and implement at least one of them.

Portability

Set up a different database system locally on your machine and import the data. What parts of your code do you need to change? Do you observe the same performance differences?

Report

1. Describe the two efficient approaches you implemented.
2. Give the runtime for loading `auth.tsv` with the straightforward and the efficient approaches.
3. Why are the efficient approaches faster?
4. Which tuning principle did you apply?
5. How did you ensure portability of your code?

Please indicate the average time per group member that was spent solving this assignment. The time that you indicate will have *no* impact on your grade.

Grading scheme:

Category	Max. Points
Description of your setup	0.5
Why is the straightforward approach slow?	1.5
Why is your efficient approach faster?	1.5
Report on your portability findings?	1.5
Bonus: You found the most efficient approach	0.5

Reminder: Additional questions about the involved topics/techniques will be asked during the meeting.

Important: If the grading scheme is unclear, ask the instructor!

²This might be very slow. Instead of loading all the data with this approach, you can also load part of the data and assume that runtime scales linearly. Mention this in the report.