

Exercise 7

Released: May 21, 2019 · Discussion: May 27, 2019

1 Star Schema Benchmark

The Star Schema Benchmark is a collection of queries on a database schema for quantitative comparison of database management systems in OLAP workloads. In this assignment you shall have a look at the Star Schema Benchmark [2] as it will be used through out this exercise.

1. What are the benefits of a star schema? Which applications make use of it?
2. Describe schema and queries of the Star Schema Benchmark. What pattern do you recognize in the queries?

2 Star Joins

1. What is a Star Join?
2. What techniques do you know to calculate star joins efficiently? Discuss their advantages and disadvantages and in which situations they should be used.
3. What is a Semi Join? How can they be used to optimize Star Joins?
4. Apply all techniques to query 2.3 of the Star Schema Benchmark.
5. An advanced technique to efficiently calculate star joins is the so called „Invisible Join“, proposed by Abadi et. al. [1]. Explain its functioning and apply it to the example query from the previous assignment.

3 Pipeline Implementation

Pipelining is a way to reduce I/O operations by passing results directly to the next operator instead of writing the whole result to disk. In this exercise you have to implement at least two operators in the template `04_volcano`.

Implement the functions `open()`, `next()` and `close()` according to the volcano iterator model for:

- a *Selection*
- a *Nested Loops Join*

Download the file `04_volcano.zip` from the course website¹ and extract it. If you extract it into the folder of the previous project(s) you may have to merge the `CMakeLists.txt` files. You have to complete the files `include/volcano/selection.h` and `include/volcano/nested_loops_join.h`.

We have implemented two plans (in `04_volcano.cpp`) for a simple query for testing purposes.

- What do you see if you execute them?
- Which plan is faster? Explain why.

Build instructions:

1. You may have to modify the path to the workloads (e.g. to `workloads/lineitems.tbl`)
2. Extract the archive and navigate into the extracted folder.
3. Run `cmake` to create a makefile for your system: `cmake .`
4. Run `make` to create an executable binary file: `make`
5. Execute the created binary file (e.g. `./04_volcano` on linux)

References

- [1] Daniel J. Abadi, Samuel R. Madden, and Nabil Hachem. Column-stores vs. row-stores: How different are they really? In *SIGMOD*, pages 967–980. ACM, 2008.
- [2] Pat O’Neil, Elizabeth J. O’Neil, and Xuedong Chen. The star schema benchmark (ssb), 2009. Revision 3, <http://www.cs.umb.edu/~poneil/StarSchemaB.PDF>.

¹http://dbis.cs.tu-dortmund.de/cms/en/teaching/ss19/arch-dbms/exercises/04_volcano.zip