miniHive

Milestone 4

Optimizing *miniHive*

In the fourth milestone, we optimize *miniHive*. The goal is to reduce the total amount of data stored in intermediary files within HDFS.

We have considered various options in class, and it is up to you which optimization(s) you implement. Chain folding is a good bet. You may also implement more than one idea.

In case you need to adapt the LUIGI workflow, there is a chapter from the book “Hadoop with Python” with an introduction to LUIGI in Moodle.

1 Evaluation Data

We use data from the TPC-H benchmark, describing customers and their orders. This is a famous benchmark, where synthetic data can be generated based on a scale factor (SF).

The ER-like diagram in Figure 1 describes the schema, and has been taken from the official benchmark description. We assume that all data adheres to this schema.

The parentheses following each table name contain the prefix of the column names for that table. The arrows point in the direction of the one-to-many relationships between tables. The number/formula below each table name represents the cardinality (number of rows) of the table. Some are factored by SF, the scale factor.

2 Calling *miniHive*

For evaluating a single query, *miniHive* gets

- an optional flag telling miniHive to switch optimization on,
- the optional scale factor, allowing you to derive the *approximate* sizes of tables,
- the input files, already stored in HDFS or available as local files,
- the optional execution environment (LOCAL or HDFS), set to HDFS by default,
- the SQL query over TPC-H data to evaluate.

This is how we call the optimized *miniHive* for execution on local files:

```
miniHive.py --O --SF 1 --env LOCAL "select distinct N_NAME from NATION"
```

If called in `LOCAL` mode, this outputs an approximation of the HDFS storage costs (more on this later). **Make sure your submitted version does not write any other information to standard out, as this will confuse the test scripts.**

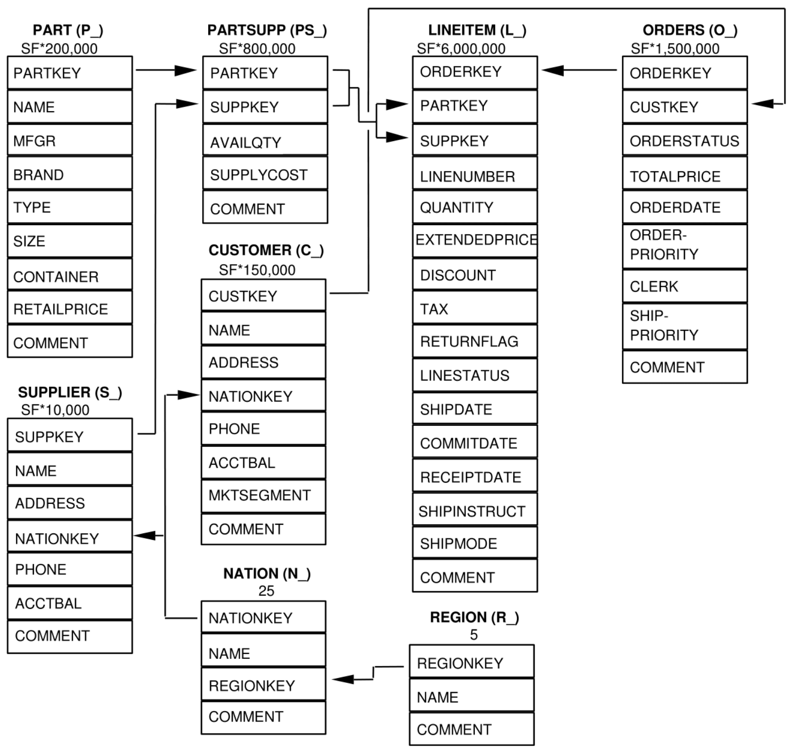

Figure 1: The TCP-H benchmark data [Source: `www.tpc.org`.]

3 Material in Moodle

- A sample dataset that was generated with scale factor 0.01 (`data_0_01.zip`).
- The file `miniHive.py`. You may alter this file.
- The file `costcouter.py`. You must not alter this file.
- A list of SQL test queries in `miniHive.q`.

4 What to Submit

Submit a flat ZIP file (without any subfolders) containing

- a README file mentioning any additional Python modules that need to be installed via `pip`, and a 1-paragraph description of the optimization(s) that you have implemented. The README must have this format:

```
# Author: <your lastname>, <your firstname>
# TODO install <comma-separated list of modules to install, or empty>

# My approach:
<1-paragraph description of your optimizations>
```

- Further, all Python files required to run *miniHive*.