MDS
Winter Term 2024/25
miniHive, Milestone 1

The goal of the coding project is to build a mini-version of Apache Hive, called *miniHive*. The first milestone requires you to write a query compiler that translates SQL queries into relational algebra.

Translating SQL into Relational Algebra

We exclusively consider SQL statements that are conjunctive queries, of the form

$$\begin{array}{l} \text{SELECT DISTINCT } A_1, \ldots, A_n \\ \text{FROM } T_1\ t_1, \ldots,\ T_m t_m \\ \text{WHERE } C \end{array}$$

where

- $A_1, \ldots, A_n$ are attribute names,
- $T_1, \ldots, T_m$ are relation names,
- $t_1, \ldots, t_n$ are optional renamings,
- and C is a conjunction of equality conditions of the form $t_i.A = t_j.B$ or $t_i.A = c$ or $c = t_i.A$, where c is a constant, and A and B are attribute names.

The first step in *miniHive* is to translate SQL statements into relational algebra.

We make use of two third-party Python modules:

- We use `sqlparse` to parse SQL statements into a Python datastructure.
  We recommend that you make use of this library and do not parse SQL statements on your own (e.g. by using string split or similar).
  More about this module at `https://github.com/andialbrecht/sqlparse`.
- We use `radb` to handle relational algebra statements.
  This includes parsing from a string, manipulation, and serialization to a string.
  Inspect the source code of the RADB data structures at `https://github.com/junyang/radb`.

Smoke Test. Write a module `sql2ra` that takes a parsed SQL statement and performs the canonical translation into relational algebra, using the operators σ, π, $\times$, and ρ.

This is how your implementation should work when you spin up the interactive Python interpreter:

```
>>>import sqlparse
>>>import radb
>>>import sql2ra
>>>
>>> sql = "select distinct name from person where gender='female'"
>>> stmt = sqlparse.parse(sql)[0]
>>>
>>> ra = sql2ra.translate(stmt) # This invokes your code.
>>>
>>> type(ra) # Important! Do not return a raw String, but a Project object.
<class 'radb.ast.Project'>
>>>
>>> print(ra)
\project_{name} (\select_{gender = 'female'} person)
```

Test Suite. We provide a suite of unit tests at `https://github.com/miniHive/assignment/blob/master/milestone1/test_sql2ra.py`. Your solution must pass all tests.

Remarks: For Milestone 1, you are not required to make your implementation particularly efficient, instead, focus on correctness.

Do not implement a hard-coded solution, i.e., a solution that only works for the test cases provided. We will check all solutions for plagiarism using an automated tool.

Praktomat will run the suite of unit tests of `test_sql2ra.py` when you upload your solution. Your solution should also work for other, similar queries as in `test_sql2ra.py`. Make sure to upload `sql2ra.py` as a *single* file in Praktomat, in time for the deadline.

The deadline for submitting Milestone 1 is October 28, 2024, 12 noon. A successful submission that passes all public tests and passes the plagiarism check earns 5 points.