Scaling Database Systems
Winter Term 2025/26
Chair of Scalable Database Systems
Prof. Dr.-Ing. S. Scherzinger

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

Milestone 1: Query Compiler

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.

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 it 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)
>>>
>>> type(ra) # Important! Return a Project object, not a raw string.
<class 'radb.ast.Project'>
>>>
>>> print(ra)
\project_{name} (\select_{gender = 'female'} person)
```

Remarks: For Milestone 1, focus on *correctness* rather than efficiency.

Praktomat will execute the unit tests in `test_sql2ra.py` upon submission. Solutions that rely on hard-coding or other non-general methods will not be accepted by us. Secret tests in Praktomat will detect such cases.

Submit your solution as a *single file* named `sql2ra.py` via Praktomat: https://praktomat.sdbs.fim.uni-passau.de (accessible only within the Uni Passau network or via the Uni Passau VPN).

Deadline: November 25, 2025, 12:00 (noon, daylight). A valid submission must pass all public tests, the plagiarism check, and must not be hard-coded.
