

miniHive

Milestone 1

The goal of this coding project is to build a mini-version of Apache Hive, called *miniHive*. The first milestone is to write a query compiler that translates simple SQL queries into relational algebra. Later, we will add selection pushing to optimize the query, and then we will compile the relational algebra query into a physical query plan of MapReduce jobs. But you are not completely on your own with this...

Translating SQL into Relational Algebra

We consider simple SQL statements of the form

```
SELECT DISTINCT  $A_1, \dots, A_n$ 
FROM  $T_1 t_1, \dots, T_m t_m$ 
WHERE  $C$ 
```

where

- A_1, \dots, A_n are attribute names,
- T_1, \dots, T_m are relation names,
- t_1, \dots, t_m are optional renamings,
- and C is a conjunction of atomic equality conditions of the form $t_i.A = t_j.B$ or $t_i.A = c$, 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 existing Python modules:

- We use `sqlparse` to parse SQL statements.
More on this module at <https://github.com/andialbrecht/sqlparse>.
- We use `radb` to handle relational algebra statements.
More on this module at <https://github.com/junyang/radb>.
This module requires Python 3.6!

1 Material in Moodle

- The file `test_sql2ra.py` contains unit tests. Make sure your implementation passes these tests.

2 What to Submit

Write and submit 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! Do not return a raw String, but a Project object.
<class 'radb.ast.Project'>
>>>
>>> print(ra)
\project_{name} (\select_{gender = 'female'} person)
```

Remarks: For Milestone 1, you are not asked to find any particular optimizations to make your implementation more efficient. All you are required to provide is a correct and clean implementation.
