HW3 - Implement a Very Simple Database

Submit Assignment

Due Nov 13 by 11:59pm **Points** 10 **Submitting** a file upload **File Types** zip
Available Oct 30 at 12am - Nov 18 at 11:59pm 20 days

Overview

This homework provides some insight into the implementation of a database engine by implementing a very small, simple database engine on top of CSVFiles. There are two parts to the homework.

1. Building a simple database catalog.
2. Building a query engine using information defined in the catalog.

Part 1: CSVCatalog

You can find an implementation template in Python at **CSVCatalog_Template.** (https://github.com/donald-f-ferguson/W4111-f18/blob/master/Homework_Templates/HW3/CSVCatalog_Template.py) There is an empty **catalog_unit_test.py** (https://github.com/donald-f-ferguson/W4111-f18/blob/master/Homework_Templates/HW3/catalog_unit_tests.py) file, which I will be filling out with tests.

CSVCatalog implements a simple database catalog. The catalog defines three types:

1. A *TableDefinition* represents metadata information about a CSVDataTable. The database engine will maintain the data in a CSV file. The catalog contains information about:
   1. The path/file name for the data.
   2. Column names, types and whether or not NULL is an allowed value. The column names defined via the catalog API are a subset of column headers in the underlying CSV file.
   3. Columns that comprise the primary key.
   4. A set of one or more index definitions. An index definition has a name, type of index (PRIMARY, UNIQUE, INDEX) and columns that comprise the index value.
2. ColumnDefinition: A class defining a column.
3. IndexDefinition: A class defining an index.

The catalog supports:

- Defining a new table.
- Dropping an existing table definition.
- Loading a previous defined table definition.
- Adding and removing columns from a table definition.
- Adding and removing indexes from a table definition.

You will save the catalog information in a set of tables you define and create. Your definition must include constraints that you think make sense. You must put your tables in a schema named CSVCatalog.

HW 3 Part 2 will build on the previous CSVDataTable. Part 2 uses the catalog from part 1 to process and optimize queries.

Part 2: CSVDataTable Engine

We will discuss this part of the homework on Friday, after an initial discussion of query processing and optimization. The HW definition will provide an updated CSVDataTable implementation.

Submission Format

Your submission is a zip file containing the following files/directories.

- README.md
- /src
  - CSVCatalog.py (your implementation)
  - CSVDataTable.py (your implementation building on base impl.)
- /test
  - unit_test_catalog.py: The test file provided and any additional tests your write.
  - unit_test_catalog.txt: A file containing the execution of your tests.
  - unit_test_csv_table.py: A test file provided and any additional tests you write.
  - unit_test_csv_table.txt: A file containing the execution of your tests.
- /data:
  - Minimally, People.csv. Teams.csv and Batting.csv from the Lahman 2017 data.
  - Additional CSV file you decide to include.
- /sql
  - create.sql: SQL statements to create your tables and constraints for the catalog.