Query Processor

Provide SQL querying capabilities on CSV files.

In this assignment, you will be writing a program that takes an SQL statement as input and gives the result of the query as output. Typical queries would look like this:

Example 1

```
SELECT *
FROM student.csv
```

Example 2

```
SELECT *
FROM student.csv
WHERE student.class = “I MTech”
ORDER BY student.reg_no ASC;
```

Example 3

```
SELECT customer.name, account.balance
FROM customer.csv NATURAL JOIN account.csv
WHERE account.balance >= 100000
ORDER BY account.balance DESC;
```

Assume that the first row of the CSV file contains the attribute names.

- The `SELECT` clause should support projection of the specified attributes and projection of all the attributes (specified by `*`).
- The `FROM` clause should support cross product (cartesian join) and natural join.
- The `WHERE` clause should support equality and inequality of integers and strings. (For the sake of this assignment do not bother about nested queries and compound conditions using and, or, not, etc.)
- The `ORDER BY` clause should support sorting on an attribute in ascending or descending order. (If no ordering is specified, the default is ascending order).
- `SELECT` and `FROM` clauses are mandatory. `WHERE` and `ORDER BY` clauses are optional.
- The query result must be written to a CSV file.
- Implement materialization, do not bother about pipelining. You may assume, for the sake of simplicity, that the whole table will fit in the main memory.
- You may use Python or Java for implementation.

Implement one additional SQL feature of your choice (that is not part of the above description). Some ideas are (Please note that these are just examples and you need not limit to these):

- Enhance `WHERE` clause to include compound conditions like `AND, OR, NOT`.
- Enhance `SELECT` clause to include `AS` operator.
- Enhance `FROM` clause to include different types of joins like outer join, self join.
- Enhance `ORDER BY` clause to include sorting on multiple attributes.
- Implement `GROUP BY` clause supporting aggregation functions.
- Implement nested queries.