

CS 4700/6700 Project P1

In this project students will implement a program to process relational algebra (RA) queries in Java. Combined with file editing for updating the data, this can lead to a simplified database system.

Input relation tables are given in csv format whose names have the form RN.txt (where RN is a relation's name); the first line of each such file lists the attribute names of the relation. Attributes are assumed to have either strings or integers or decimal numbers as domains for this project. Relational algebra queries are given in an input file (named as RAqueries.txt), having one query per line. Your Java program reads RAqueries.txt and then processes the queries contained in it. Your program should also write to an output file named as "RAoutput.csv" which contains one of the queries on one line and followed by the output produced for that query. The file should contain all the queries your program can process. Your Java program will know what relation names are of interest after reading RAqueries.txt.

In RAqueries.txt relational algebra operators PROJECT, SELECT, INTERSECT, and JOIN are typed as these four-letter strings: PROJ, SELE, INTE, JOIN; Natural JOIN is typed as *, UNION is typed as U, CROSS PRODUCT is typed as X, and DIFFERENCE is typed as -; subscripts are used to represent conditions for selections and to represent list of attributes for projections; the comma "," is used to represent "AND" and the word "OR" represents itself. For simplicity, in this project we do not work with the logical "OR" and "NOT" and we only work with the natural join.

csv stands for "comma separated values". An example csv line with 5 values is:

2, abc, 4, xyz, 9

An example RAqueries.txt contains the following six lines:

SELE_{Payment > 80} (Play)

PROJ_{ANO, MNO} (Play)

ACTORS * Play

(PROJ_{ANO} (ACTORS * Play))

(PROJ_{ANO} (SELE_{Payment > 80} (Play))) - ((PROJ_{ANO} (SELE_{Payment < 70} (Play)))

(PROJ_{ANO} (SELE_{Payment > 80} (Play))) U ((PROJ_{ANO} (SELE_{Payment < 70} (Play)))

Students can use any Java structures/classes as long as they are not directly intended for supporting RA operators. Students cannot use database-like systems in their Java programs. You should implement Java functions for various RA operators and then use the functions to process the queries. Your java program can assume that all the input files are in the folder where your submitted program is to be run.

Students write a report to describe the following:

1. A line indicating **How to Run the Program** (that can be copy/pasted) on Windows or Unix.
2. A list of **Implemented Operators/Queries** (using Line 1, ... Line 6 to refer to the six queries and type PROJECT, SELECT, INTERSECT, JOIN, *, UNION, Difference, CROSS PRODUCT, Composition to indicate that the listed operators were implemented). Composition (of operators) is considered as an operator.
3. **Two Screenshots** showing results produced by the implemented Java program when processing line 3 and line 4 in RAqueries.txt. If your program can not process these lines, then provide screenshots of two lines that your program can process; this will lead to lower marks. Demonstration on how to make screenshots is given in a file in the project folder on pilot.

4. **Output Produced** by the implemented Java program answering the RA queries in RAqueries.txt (these are the contents of RAoutput.csv).

Use the bolden words in each item given above as a section title in the report to facilitate reading/marking. Use the string "Section:" before the section titles. Sections not in your report will be treated as "not submitted".

Submit the following on pilot: Your report (a file) and your java program (a compressed file using zip etc).

Marking will focus on what was done, correctness, and readability.