Project Overview

In this project, our goal was to implement Relational Algebra (RA) operators. These include the select, project, minus, union, equi-join, theta-join, and natural join operators. We based the Java implementation of these operators on the pseudo-code that was lectured on in class. We also implemented a typecheck method that will check the size of a tuple and the type of its elements to ensure that it is from the correct domain.

Key Features

- **Select Operator**
    - This project allows the user to select a specific subset of rows from a table based on a given condition in string form
- **Project Operator**
    - This project allows the user to select a specific attribute of a table based on one's given to the method as parameters and separated by spaces.
- **Minus Operator**
    - This method will allow the user to only add to the ArrayList what isn't available in the parameter table2. It will then return the new table.
- **Union Operator**
    - This method allows the user to unionize two tables together which will essentially combine each table's tuples. This implementation does allow for duplicates due to Dr. Miller allowing it for this project.
- **Equi-Join Operator**
    - This allows the user to join two tables together based on their matching column values based on the 2 attributes that the user will provide. If there happens to be any duplicate attributes the method will add a 2 to the end of the attribute name to eliminate duplicates.
- **Theta-Join Operator**
    - This will join this.table and table2 based on an inputted String condition. A nested loop is used teh compare the tuples from both tables based on this condition. Inside the loop there is a switch case operation used to decipher between the different conditional operators that could be given in the condition parameter. If there is a duplicate attribute name the method will add a 2 a to the end of the duplicate to make the name unique.
- **Natural Join Operator**
    - This method joins this.table and table2 requiring that tuples from both tables have equal common attributes.

  Typecheck
    - Will ensure that the amount of elements in an array t and the type of each element are from the correct domain. Takes in an array and returns a boolean of whether or not the inputted array's length matches the original array as well and if the domains match.

Usage/Examples

Table1

Name	Age	Job
Joel	22	Teacher
Mike	3	none
Emily	90	Store clerk

Table2

SSN	Age	State
123456789	22	NY
234567891	3	GA
345678912	90	AL

- **Select Operator**
  - If you want to select a certain row based on a predicate you can do *Table1.select("Age>30")*
    - Which in this case would output the 3rd row of table1.

Emily	90	Store clerk

- **Project Operator**
  - If you want to project a certain attribute you can do *Table1.project ("Name Age")*
    - Which would output the 1st and 2nd column of table1.

Name	Age
Joel	22
Mike	3
Emily	90

- **Minus Operator**
  - If you want to subtract the contents of one table from another you can do *Table1.minus(Table2)*
    - Which would output Table1 with only the attributes Name and Job
- **Union Operator**
  - If you want to unionize the contents of one table to another you can do *Table1.union(Table2)*
    - Which would output Table1 with its contents along with all the contents of Table2
- **Equi-Join Operator**
  - If you wanted to equi-join the contents of two tables you can do *Table1.join("Age", "Age", Table2)*
    - Which would output the combination of Table1 and Table2 but only with the records/rows in which the age's are equivalent to another record in the other table. Since both tables have the same attribute one will be renamed to Age2.

- **Theta-Join Operator**
    - If you wanted to theta-join the contents of two tables you can do Table1.join(“Age >= Age”, Table2)
        - Which would output a table consisting of all the records in which the Age attributes in Table1 are greater than or equal to the Age attributes in Table2.

- **Natural Join Operator**
    - If you wanted to Natural join the contents of two tables you can do Table1.join(Table2)
        - Which would output the combination of Table1 and Table2 with any duplicates being deleted. This would mean that the attributes of the output would be Name, Age, SSN, and State and would delete the duplicated Age attribute.