

# CECS 323 RELAX SETUP

## OBJECTIVE:

Help you “compile” and run your Relational Algebra statements.

## INTRODUCTION:

The RA tool that we will use for our exercises needs a little setup:

1. You will find the Relax tool [here](#).
2. Near the top, there is a tab labeled “Group Editor”. Select that.
3. The raw data that we have to load into Relax can be found [here](#). Cut and paste all of that into the Relax editor.
4. Scroll to the bottom of the screen and press the “preview” button. This will load the data that we have been using for the Practice SQL into the Relax editor.
5. Scroll down a little further. Just below the “preview” button you will see the text “classModels” and to the right is some text in blue: “use Group in editor”. Select that. It will not appear as though anything is happening, but it is.
6. Go back to the top of the page, and select the “Relational Algebra” tab to get into the RA editor.

There are a few things about the Relax tool that you need to know. I have learned these by experience, and I am sure that there are other things that we will discover together as we go along. Please let me know any syntactic issues that you run into, so that I can share them with the class.

1. Relax does not support the subscripts that you have seen in RA expressions. The one thing that it does do is that the operators have ( ) around each operand, which sets off the subscript (where you need one) for the operator from the operands.
2. Put a space on either side of any Boolean operator.
3. Put a space between the end of a subscript and the ( for the operator (right or left hand).
4. Relax has a specific symbol for not equals:  $\neq$ . Be sure to use that rather than try anything else when you want to test for a not equals. I think the parser expects all operators to be a single character.
5. When you use a relational operator, be sure that both operands are enclosed in ( ). Practically speaking, this means that you really ought to build your RA statement up a bit at a time, test it at each stage, and avoid adding a lot of symbols to the statement at once. Trying for a “big bang” solution will lead to a lot of frustration.
6. Unlike many IDEs that you have used in the past, Relax does not have any highlighting that it does to show you where in the expression the matching parenthesis is. I have found that it sometimes is worth it to cut/paste my RA expression into another tool, just to help me find my matching parenthesis.
7. Oddly enough, there **is** a way to sort the tuples in an RA expression. It’s the  $\tau$  operator. The subscripts for  $\tau$  are the columns that you need to sort on, comma delimited.
8. The Relax tool is case-sensitive for all names. I have found that you can click on a relation or an attribute on the left, and it will copy that name over where your cursor is positioned in the RA editor. That’s one way to make sure you never misspell.
9. I have not found a way to do the inner join using in Relax, which means that you have to use the inner join on in each case.

## CECS 323 RELAX SETUP

10. In case you were wondering, the subscript to the join can have expressions like “customers.customerNumber = orders.customerNumber” in it so that you can be clear which “customerNumber” you mean.

11. Dates –

- a. Interestingly, RelaX does support a year() and a month() function on dates. There may be other functions as well.
- b. If you want to put a date literal into your Relational Algebra statement, you must use the date function to convert that text string into a date. For instance, to find all the orders placed on 11/24/2014, your Relational Algebra statement would be:

$\sigma \text{ date('2014-11-24')} = \text{orderDate orders}$

12. The “order by” operator (tau:  $\tau$ ) can go in ascending or descending order. You just put the string “desc” after the name of the column that you want to order by in descending order. For an order by that has multiple columns, you can specify the descending option for each column individually. Like SQL, RelaX defaults to **ascending** order.

13. RelaX **does** support the like operator the same way that SQL does. For instance, a legitimate Relational Algebra statement in RelaX (using one of the default databases) would be:

$\sigma \text{ upper(lastname) like '%JESUS\%'} \text{ Salesperson}$

- a. The upper function converts the characters in the lastname column to all upper case.
- b. The % in the text string that is the right hand operand to the like function means “zero or more characters”. In this case, JESUS can occur anywhere in the person’s last name, and they will be selected.

### What to Turn In:

- Please write and execute a simple Relational Algebra query to return all the customer names that have the word “mini” in it anywhere in the name. Assume that you do not know what the casing is. For instance, it might be “Mini”, or “MINI” or any other combination of upper and lower case letters.
- Give me a screen shot of the graphical display that RelaX makes of the Relational Algebra statement.