Relational Algebra Format

The syntax for relational algebra is described below:

1. **Unary Operations**
   - **Selection**: either `S(` `cond` `,` `expr` `)` or σ(`cond` `,` `expr`)
   - **Projection**: either `P([` `attr` `],` `expr` `)` or π(`attr` `,` `expr`)
   - **Renaming**: either `R(` `s` `(` `attr` `),` `expr` `)` or ρ(`s` `(` `attr` `),` `expr`)
     - `cond` is the *selection condition*.
     - `s` is the new relation name.
     - `attr` are the *attributes*.
     - `expr` is a relational algebra expression.
2. **Binary Set Operations**
   - **Union**: either `expr` `|` `expr` or `expr` $\cup$ `expr`
   - **Intersection**: either `expr` `&` `expr` or `expr` $\cap$ `expr`
   - **Set Difference**: `expr` `-` `expr`
   - **Cartesian Product**: either `expr` `*` `expr` or `expr` $\times$ `expr`
     - `expr` is a relational algebra expression.
3. **Selection Conditions**
   - **Conjunction**: either `cond` `/\` `cond` or `cond` $\wedge$ `cond`
   - **Disjunction**: either `cond` `\/` `cond` or `cond` $\vee$ `cond`
   - **Negation**: either `~` `cond` or $\neg$ `cond`
     - `cond` is the *selection condition*.
4. **Relational Operations**
   - **Equal**: `val` `==` `val`
   - **Non-equal**: either `val` `!=` `val` or `val` `<>` `val`
   - **Less than**: `val` `<` `val`
   - **Less than equal**: either `val` `<=` `val` or `val` $\leq$ `val`
   - **Greater than**: `val` `>` `val`
   - **Less than equal**: either `val` `>=` `val` or `val` $\geq$ `val`
     - `val` is a *value* or *attribute*.
5. **Attributes**
   - **Single attribute**: lowercase or numeric but cannot start with number
     e.g., `attr` and `a1`
   - **Multiple attributes**: comma-separated *single attribute*
     e.g., `attr, a1`