

```

operator : operator_name '(' ' ' )'
          | operator_name '(' ' ' parameter ( ' , ' parameter ) * ' ' )'
          ;

parameter : '[' parameter ( ' , ' parameter ) * ']'
          | operator
          | term
          | relation
          | string
          ;

term : function_name '(' ' ' ' ' )'
     | function_name '(' ' ' term ( ' , ' term ) * ' ' )'
     | variable
     | constant
     ;

```

Po minulej konzultácii sme došli na to, že gramatika algebry by mala vyzeráť nejak takto. Je to z dôvodu, lebo v datalogu chceme podporovať predikáty ako napríklad *subtotal*, ktoré berú ďalšie predikáty ako parametre. Taktiež chceme mať možnosť podporovať ľubovoľný operátor, ktorý si užívateľ naprogramuje. Ako základné operátory algebry zoberieme klasické *JOIN*, *ANTIJOIN*, *UNION* a *REC* a budeme musieť ešte pridať operátor *RELATION*¹, ktorý vytvorí dočasnú reláciu.

Z týchto dôvodov veríme, že jednotný *interface parametrov* pre operátory nie je možný ale zároveň nie je ani potrebný. *Loader* bude fungovať nasledovne. Pri spracovávaní konkrétneho operátora, dostane od ANTLeru takzvaný *context*², v ktorom bude konkrétny *operator_name* a konkrétne objekty parametrov. Pomocou týchto informácií sa rozhodne, ako (respektíve či vôbec) zavolá konkrétnu javovskú funkciu databázy.

¹Parmetre by mohli byť napríklad niekoľko listov konštát.

²<https://stackoverflow.com/questions/24185887/understanding-the-context-data-structure-in-antlr4>