

```

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*<sup>1</sup>, ktorý vytvorí dočasnú reláciu.

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

---

<sup>1</sup>Parametre by mohli byť napríklad niekoľko listov konštát.

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