DBS Query Language (DBSql) for CMS

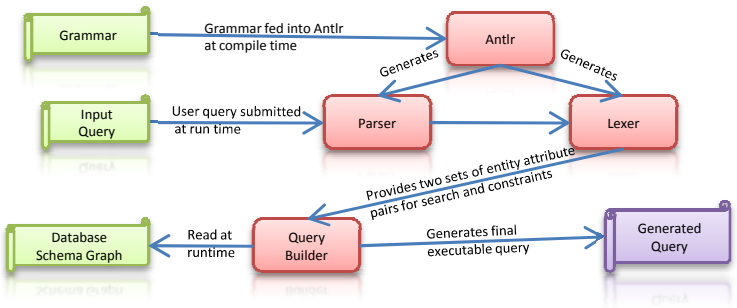
Anzar Afaq (Fermilab), Lee Lueking(Fermilab), Valentin Kuznetsov(Cornell) ,Vijay Sekhri(Fermilab)
https://twiki.cern.ch/twiki/bin/view/CMS/QL

Introduction

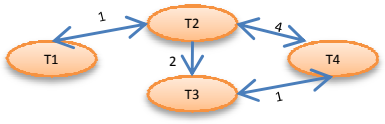
The DBS query language (DBSql) is used to perform searches on the CMS physics data catalog, DBS (http://cmsdbs.cern.ch). It employs a syntax similar to SQL (Structured Query Language) and simplifies the complexities of the underlying DBS schema. Users specify what they wish to find (the select clause) and the constraints of their search (the where clause). Multiple elements can be searched for together while imposing numerous constraints. Users do not need to know the structure of the underlying DBS schema, or the join conditions among its tables (the from clause), when writing their query.

DBSql Architecture

The user input is interpreted by a lexer and parser, then a Query Builder generates the SQL query that can be executed on the database. The ANTLR (http://www.antlr.org) parser/lexer tool uses a grammar file which defines the syntax and semantics for DBSql to generate the parsing/lexing code for compile time. The Query Builder package uses this parser and lexer to interpret the user input query. If the query does not honor the semantics of the grammar, then the parser or lexer will raise an exception. The Query Builder generates the final SQL query that is executed directly on the DBS underlying MySql or Oracle database.



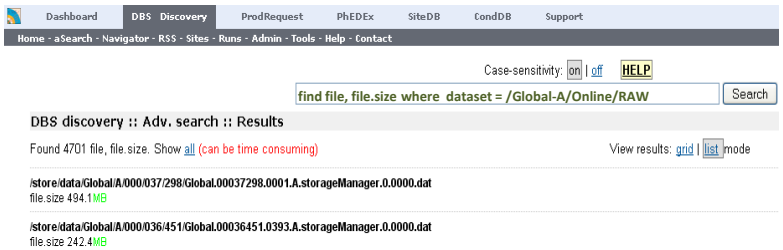
The entire database schema is represented as a weighted directed graph with nodes representing tables, and edges representing relationships between tables. The Query Builder then uses the Dijkstra's shortest path algorithm to determine a path from one table to another and resolve multi-path ambiguities. The chosen path is used to construct the final SQL query. For example in the figure below T1 through T4 represent four tables in the database and edges between the tables represents the relationships between the tables. The user can select elements from table T1 and T4 without specifying intermediate tables T2 and T3 and the join conditions. The Query Builder determines the shortest path T1->T2->T3->T4 to add the intermediate tables and join conditions in the final SQL query.



Shortest path from table T1 to T4 is
T1 -> T2 -> T3 -> T4 (Total weight 4)
Shortest path from table T4 to T1 is
T4-> T2 -> T1 (Total weight 5)

Integration of DBSql with CMS Discovery web tool

DBSql is fully integrated with the DBS discovery for searching CMS data on the web



Integration of DBSql with Command Line tool

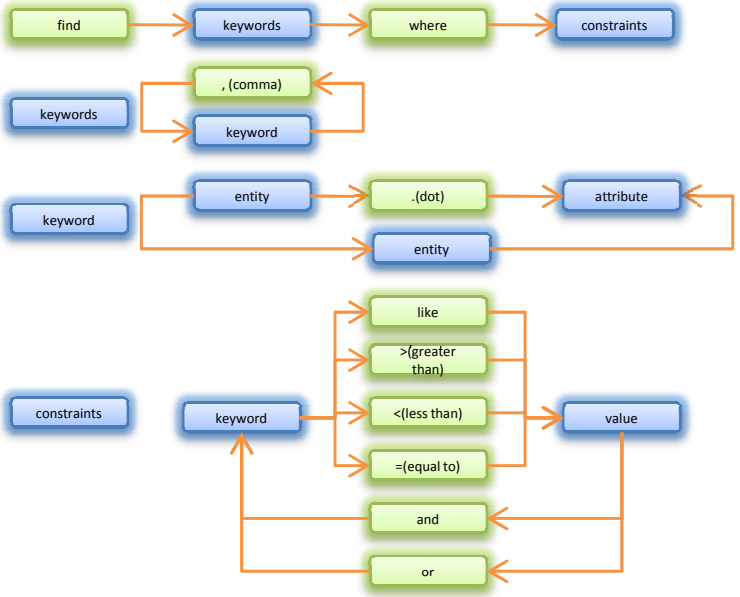
DBSql is fully integrated the DBS command line tool for searching CMS data on the command line

➢dbsql **"find file, file.size where dataset = /Global-A/Online/RAW"**
Using DBS instance at: http://cmsdbsprod.cern.ch/cms_dbs_prod_global/servlet/DBSServlet
FILES_FILESIZE FILES_LOGICALFILENAME

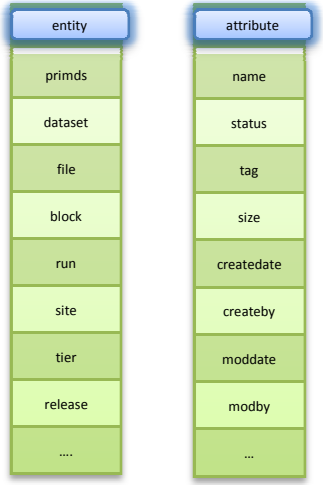
518081713	/store/data/Global/A/000/037/298/Global.00037298.0001.A.storageManager.0.0000.dat
254133550	/store/data/Global/A/000/036/451/Global.00036451.0393.A.storageManager.0.0000.dat

DBSql Grammar

The figure represents a simplified view of the grammar that defines the semantics of the DBSql. Entities are like logical tables, and each entity has a set of additional attributes (table's columns). The grammar allows any combination of attribute and entity, however not every attribute is appropriate for each entity. An incorrect combination will be detected by the Query Builder. Additional functions to count and sum items returned, e.g. number of files and file sizes respectively, make the language a powerful tool for generating summaries.



The keywords (entity or entity.attribute) used in DBSql are natural within the CMS community. Each keyword maps into a particular table or table.column in the underlying database. The mapping of these keywords to the table.column are maintained in a static file that models the graph representation of the entire schema.



Examples

How to	Example input query	Output from executing the generated query
find attributes like creation date, created by and modified by of a specific primary datasets	find primds.createdate, primds.createby, primds.modby where primds = CSA07Muon	*PRIMARYDATASET_CREATIONDATE PRIMDS_CREATEBY_DN PRIMDS_MODBY_DN* *2007-09-24 18:29:37 CEST NO_DN /DC=org/DC=doegrids/OU=People/CN=David Mason 216744*
find all dataset with name matching some pattern	find dataset where dataset like /CSA0*/CMSSW_*/*	*PATH* */CSA07Tau/CMSSW_1_6_1-PreCSA07-HLTSplit-FNALTest3-Chowder/GEN-SIM-DIGI-RAW*
find all files created exactly at 2007-04-20 11:27:21 CDT or modified in 2008 or in a specific run	find file where file.createdate = 2007-04-20 11:27:21 CDT or file.moddate > 2008 or run = 234	*FILES_LOGICALFILENAME* */store/data/CRUZET3/dataset_PD_1/RAW/v1/000/000/000/f9989f3c-2184-4966-8840-556b3cc3da30.root* */store/data/CRUZET3/dataset_PD_3/RAW/v1/000/000/000/a3d261dd-7e36-404d-9333-b8f40988df1b.root*
find the application release of a specified dataset	find release where dataset = /RelVal145TTbar/CMSSW_1_6_0_pre14-RelVal-1188635899/RECO	*APPVERSION_VERSION* *CMSSW_1_6_0_pre14*

Example input and generated query