

Db2 Query Build

Build Complex SQL Queries Quickly

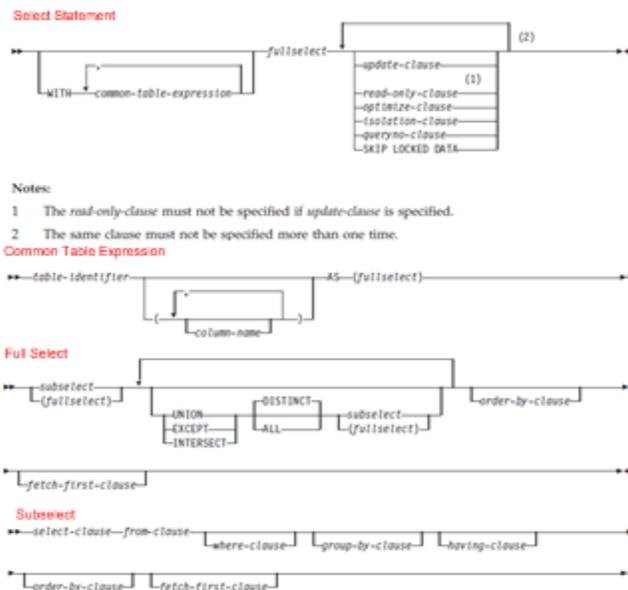
- Faster SQL
- Better SQL

Business Challenges

As Db2 SQL has evolved over the last 30+ years, it can do more and more. Taking advantage of the new features and concepts means writing more and more complicated SQL – joins – inner – left outer – right outer – full outer, unions, table expressions, recursive bill of material explosion – and many other new concepts. Making Db2 do more work in a single SQL call generally means a better performing application. Fully exploiting these SQL changes is another matter – someone has to write the new SQL.

SQL Can Be Complex

This diagram is part of the SQL query syntax. Each of the lower-case names in the diagram points to another syntax diagram.



```

TestBase Selection Criteria
Command ==>
DB2 Location. ==> DBCGLOC
Query Name. ==> CSBI.SYSTABIDX
Level. ==> 00 Show Only Query Columns. ==> Y (Y/N)
Supports ISPF Edit Line Commands: (M A B X F S LC UC)
New Component E -Expression N -Name G -Group By
Line Commands: O -Order By P -Predicate H -Having
Component Line Commands: D -Delete E -Edit SQL Q -Sub Query

S Column Name
0002 Y EXPRESSION140
N - TABLE_CREATOR
E - SUBSTR(J001.CREATOR,1,8)
0003 N J001.CREATOR VARCHAR 0128 N
P - J001.CREATOR='SYSIBM'
0004 Y EXPRESSION141
N - TABLE_NAME
E - SUBSTR(J001.NAME,1,30)
0005 N J001.TYPE CHAR 0001 N
P - J001.TYPE='T'
0007 Y EXPRESSION144
N - DBNAME
E - SUBSTR(J001.DBNAME,1,8)
0009 Y EXPRESSION145
N - TSNAME
E - SUBSTR(J001.TSNAME,1,8)
0012 Y J001.COLCOUNT SMALLINT 0002 N
0070 Y EXPRESSION142 CHAR 0008 Y
N - INDEX_CREATOR
E - SUBSTR(J002.CREATOR,1,8)
0072 Y EXPRESSION143 CHAR 0030 Y
N - INDEX_NAME
E - SUBSTR(J002.NAME,1,30)
0074 Y J002.UNIQUERULE CHAR 0001 N
0075 Y J002.COLCOUNT SMALLINT 0002 N
0076 Y J002.CLUSTERING CHAR 0001 N
    
```

This is the SQL we built:

```

TSREDD2 CSBI.Q8145000.S1 Columns 00001 00072
Command ==>
***** Top of Data *****
000001 SELECT SUBSTR(J001.CREATOR,1,8)
000002 AS TABLE_CREATOR
000003 , SUBSTR(J001.NAME,1,30)
000004 AS TABLE_NAME
000005 , SUBSTR(J001.DBNAME,1,8)
000006 AS DBNAME
000007 , SUBSTR(J001.TSNAME,1,8)
000008 AS TSNAME
000009 , J001.COLCOUNT
000010 , SUBSTR(J002.CREATOR,1,8)
000011 AS INDEX_CREATOR
000012 , SUBSTR(J002.NAME,1,30)
000013 AS INDEX_NAME
000014 , J002.UNIQUERULE
000015 , J002.COLCOUNT
000016 , J002.CLUSTERING
000017 FROM SYSIBM.SYSTABLES AS J001
000018 LEFT OUTER JOIN SYSIBM.SYSINDEXES AS J002
000019 ON J001.CREATOR = J002.TBCREATOR
000020 AND J001.NAME = J002.TBNAME
000021 WHERE J001.CREATOR='SYSIBM'
000022 AND J001.TYPE='T'
***** Bottom of Data *****
    
```

The only SQL needed is column expressions and the predicates.

SQL Results and SQL Debug

We can quickly view the SQL results for accuracy and there is an interface to [SQL Debug](#) to provide explain and Catalog drill down capabilities.

Searchable Query Repository

Queries are saved in a searchable repository. Search by creator and name, table accessed, or by description of the query.

Db2 Query Build - Menu Driven SQL

The join of two tables such as SYSIBM.SYSTABLES AND SYSIBM.SYSINDEXES using Db2 Query Build works like this: list the tables and 2-line commands entered to join them:

```

TestBase Query - SQL Build
Command ==>
Query Name. ==> CSBI.SYSTABIDX
Description ==>
Current Level==> 00 - Share Query? ==> Y - Auto Select columns? ==> N
Commands : COLS - column Select List
Line Commands:
D - Delete I - Insert Object
J - Join (no RI) JD - Join (DB2 RI) JT - Join (TB RI)
Q - Access Sub Query definition (TABLE and WITH)

ID Object (Enter Creator.Name or pattern) Left Join Join
ID ID Type Cols
jd 001 SYSIBM.SYSTABLES 000 L
jd 002 SYSIBM.SYSINDEXES 000 L
jd 003
    
```

The join of the 2 tables is based on the relationship in the Db2 catalog. Next, the SELECT list of columns and expressions and predicates can be added. For table creator SYSIBM, we want to see the tables and indexes: