Euroforth 2022

FQL Result Set Analysis

Introducing Forth Local Functions

Euroforth 2022

FQL Result Set Analysis

Before...

```
: TAB-SETDATA { | pinnum poutnum pdescr piter[ GtkTreeIter ] -- } \ Set table data
...
  SQL| SELECT innum,outnum,notes                          \ Get table entries
       FROM tables
       WHERE tabnum = | CURRTAB FQL-N+ |
       ORDER BY innum
  |SQL> IF
    PROWS 0 ?DO
      FQL-NEXTROW IF
        0 PCOL ZDIGITS -> pinnum                         \ Get input number
        1 PCOL ZDIGITS -> poutnum                        \ Get output number
        2 PCOL         -> pdescr                         \ Get description
        TABTREESTORE piter[ NULL gtk_tree_store_append   \ Add a row
        TABTREESTORE piter[
        0 pinnum 1 poutnum 2 pdescr -1 gtk_tree_store_set  \ Put data into the row
      THEN
    LOOP
  THEN
...
```

Euroforth 2022

FQL Result Set Analysis

The goal...

```
: TAB-SETDATA { | piter[ GtkTreeIter ] -- } \ Set table data
...
  SQL| SELECT innum,outnum,notes                          \ Get table entries
       FROM tables
       WHERE tabnum = | CURRTAB FQL-N+ |
       ORDER BY innum
  |SQDO
    TABTREESTORE piter[ NULL gtk_tree_store_append       \ Add a row
    TABTREESTORE piter[
    0 r-innum 1 r-outnum 2 r-notes -1 gtk_tree_store_set  \ Put data into the row
  SQLOOP
...
```

Euroforth 2022

FQL Result Set Analysis

Rolling up the flow control

```
  IF
    PROWS 0 ?DO
      FQL-NEXTROW IF
...
      THEN
    LOOP
  THEN
```

```
: |SQDO ( zaddr--- ) \ Runs an SQL query, starts a DO..LOOP and fetches row
  NOINTERP ;            \ Cannot interpret
NDCS:                   \ Compiling
  IN-SQL OFF            \ Finished compiling SQL
  POSTPONE (|SQL>)      \ Run the query
  0 CLIT,               \ The initial value for the DO..LOOP
  s_?do,  3             \ Compile the DO
  POSTPONE FQL-NEXTROW  \ Get the next row
  s_?br>,  2            \ Compile the IF
;

: SQLOOP ( --- ) \ Complete a query analysis loop
  NOINTERP  ;           \ Cannot interpret
NDCS:
  2 ?PAIRS  s_res_br>,  \ Compile the THEN
  3 ?PAIRS  s_loop,     \ Compile the LOOP
;
```

Euroforth 2022

FQL Result Set Analysis

How to identify the column names?

Possibility 1:

Identify at execution time

Possibility 2:

At compilation time, create set of locals

Euroforth 2022

FQL Result Set Analysis

Parsing the SQL query

Easy

```
SQL| SELECT ircallm,ircalle,irspeed,irtimeswitch,timeswitches.tsresult
    FROM ironer
    ...
```

Hard

```
SQL| SELECT IFNULL(operator.name,CONCAT( | P" Unknown operator" ZSP Z+ FQL-Z$+ | ,
       work.opnum) )  AS 'Operator',
         DATE_FORMAT(start , | Z" %e/%m/%Y %H:%i" FQL-Z$+ | )     AS 'Clock in',
       DATE_FORMAT(end   , | Z" %e/%m/%Y %H:%i" FQL-Z$+ | )     AS 'Clock out',
       TIME_FORMAT(MAKETIME(worked/60,worked MOD 60,0),'%H:%i') AS 'Worked'
    FROM work
    ...
```

Euroforth 2022

FQL Result Set Analysis

New scanning words needed

: SQSKIP (c-addr,u---c-addr',u')
\ Skip over leading occurrences of any non-printable character, or comma

: SQSCAN (c-addr,u---c-addr',u')
\ Scan to the first occurrence of any non-printable character, or comma

: SQWORD (caddr1,u1---caddr2,u2,caddr3,u3)
\ Identify the first word in an SQL string

Euroforth 2022

FQL Result Set Analysis

How to identify the column names?

First idea:

- **Create local value for each column**
- **Populate set of values for each row read**

Euroforth 2022

FQL Result Set Analysis

Better idea:
Local functions!

```
: SQL-COLCOMP, ( xt--- ) \ Compiling action of a child of SQL-MAKECOL
  >BODY @ CLIT,       \ The column number
  POSTPONE PCOLCONV   \ Fetches column data, converting to number if appropriate
;

: SQL-MAKECOL ( paddr,pu--- ) \ Make a column function
  Z" r-" PAD2 2 MOVE              \ Set prefix
  >R                              \ Save length
  PAD2 2+ R@ MOVE                 \ Copy name
  >TEMP-DICT                      \ Switch to local dictionary
  PAD2 R> 2+ ($CREATE)            \ Create the column word
  SQL-COLNUM ,                    \ Set the column number
  ['] SQL-COLCOMP, SET-COMPILER   \ When a child is being compiled
  >REAL-DICT                      \ Back to normal dictionary
;
```

Euroforth 2022

FQL Result Set Analysis

Start and cleanup

```
: ?START-LOCALS ( --- ) \ Set dictionary pointers to local, if not already done so
  TDPstart @ 0= IF    \ Not yet initialised
    START-LOCALS      \ No need to restore, it is done by ;
  THEN
;
```

```
: MICROSS-CLEANUP-LOCALS ( --- ) \ Clean up locals, if used - called by ; etc.
  TDPstart @ IF       \ Only compile if we had local values or functions
    FORGET-LOCALS     \ Lose local definitions
  THEN
; ' MICROSS-CLEANUP-LOCALS ' CLEANUP-LOCALS PATCHXT 2DROP DROP

: MICROSS-HASLVs? ( ---f ) \ True if current definition has local vals or funcs
  TDPstart @     \ Local values or functions exist
; ' MICROSS-HASLVs? ' HASLVs? PATCHXT 2DROP DROP
```

Euroforth 2022

FQL Result Set Analysis Demonstration

```
: SQTEST1 { | ptest -- } \ List details for step programs
  123 -> ptest
  SQL| SELECT program,name,miny,maxy,run
       FROM stepprogram
  |SQDO
    cr r-program . r-name z$. SPACE r-miny . r-maxy . r-run . ptest .
  SQLOOP
;

sqtest1
1 Test 1 100 120 0 123
2 Test 2 200 220 1 123  ok


: SQTEST2 { | ptest2 -- } \ Analyse operator privileges
  SQL| SELECT COUNT(*) AS numoperators, privil
       FROM operator
       GROUP BY privil
  |SQDO
    cr r-privil . r-numoperators .
    456 -> ptest2
  SQLOOP
  cr ptest2 .
;

sqtest2
0 6
1 204
2 1
3 14
4 2
456  ok
```

Euroforth 2022

FQL Result Set Analysis

New and unique feature of Forth!

Dynamically create a function that has scope only within the current definition