

RELATIONAL ARCHITECTS PRODUCT FAMILY

**CARL FEINBERG
RELATIONAL ARCHITECTS**

Relational Architects Product Family

Presented at

REXX Symposium

Annapolis, MD -- May 1992

by

Carl Feinberg

***Director of Development
Relational Architects Intl***

Some of our clients

American Hospital Association

American President Lines

Australian Telecom

Bank of Liechtenstein

Blue Cross Blue Shield

British Columbia Telephone

British Telecom

CalFarm Insurance

Ciba - Geigy

Credit Suisse (Switzerland)

Daimler Benz (Germany)

Depository Trust Company (NY)

Dow Corning

Dresdner Bank (Germany)

Ericsson (Sweden)

The Equitable Life Assurance Soc.

Federal Government Agencies

Fireman's Fund Insurance Co.

The Franklin Mint

G E Information Services

Glaxo, Inc.

The Home Depot

I B M Corporation

Iowa Public Service

Los Alamos National Labs

Los Angeles Water and Power

MCI / Telecom*USA

Mead Corporation

National Westminster Bancorp

Norwegian Telecom

NYNEX

PARS

The Pillsbury Company

State Bank of Sweden

Tandy Corporation

United States Fidelity & Guaranty

University of California

U S A A

VISA International

Product groups

DB2 Productivity Series

- RLX/REXX*
- RLX/ISPF*
- RLX/CLIST*
- RLX/Compile*
- RLX/Net*
- AcceleREXX*

Smart Jobstream Series

- Smart/CAF*
- Smart/Restart*
- Smart/QBF*
- Multi/CAF*

RLX Product Family

- Extends embedded SQL support to REXX EXECs and TSO CLISTs***
- Exactly the same embedded SQL used with COBOL and PL/1***
- Fully supports IBM's RXSQL syntax***
- Full host and indicator variable support***
- RLX parser validates SQL statements, assigns "best fit" data types***
- Full support for SQLCA and SQLDA fields***
- Extensive full screen diagnostics***
- Multi/CAF supports concurrently active DB2 plans***
- Quasi-static SQL***

33

RLX/ISPF

- Extends SQL's set processing facilities***
- Powerful composite functions***
- Load SQL query results directly into ISPF tables***
- Display and process those results on scrollable ISPF panels***
- Creates an ISPF table containing columns selected from a DB2 table***
- Encapsulated object to manipulate table***

How can we use RLX?

- Build DB2/ISPF based tools quickly and easily (DBA utilities and developer workbenches)***
- Prototype high volume applications for CICS, IMS and batch***
- Develop decision support, individual and departmental applications***
- Develop production applications for the DB2/ISPF environment***
- Testing tool for performance analysis and problem resolution (one time fixes)***
- Teaching tool - Learn SQL with immediate feedback, extensive diagnostics and context sensitive help***
- NetView automation procedures***
- Automated console operations for system administration***

Why Interpretive?

- Quick and easy development***
- 'glue' to integrate diverse components like DB2, ISPF REXX and Netview into cohesive application solutions. Combines SQL, ISPF dialog services and procedural logic into a functional unit***
- Edit and test RLX dialogs directly within PDF/Edit***
- No preprocess, compile, link edit or bind steps are required***
- Reduce application size and complexity by at least 50%***
- Reduce development, maintenance and enhancement time by at least 50%***
- Increase application functionality by 100%***

Developers can

- Apply their SQL skills immediately***
- Quickly prototype applications***
- Copy RLX SQL statements directly into their COBOL and PL/1 programs***
- Ignore data declarations and data conversion and concentrate on algorithms***
- Skip Preprocess, Compile, Link Edit and Bind steps entirely***
- Quick trial and error development approach***

Sample RLX/REXX dialog

RLXS TOWNER SYSIBM

```
----- Tables Created by SYSIBM ----- ROW 1 OF 30
Command ===>                               Scroll ===> HALF

Table Name      Owner      Type  DB Name  TS Name  DB ID  Colcount
SYSCOPY         SYSIBM    T     DSNDB06  SYSCOPY  6      14
SYSFIELDS       SYSIBM    T     DSNDB06  SYSDBAS  6      13
SYSTABLESPACE  SYSIBM    T     DSNDB06  SYSDBAS  6      23
SYSTABLES       SYSIBM    T     DSNDB06  SYSDBAS  6      31
SYSTABLEPART   SYSIBM    T     DSNDB06  SYSDBAS  6      21
SYSTABAUTH     SYSIBM    T     DSNDB06  SYSDBAS  6      21
SYSSYNONYMS    SYSIBM    T     DSNDB06  SYSDBAS  6       6
SYSRELS        SYSIBM    T     DSNDB06  SYSDBAS  6      11
SYSLINKS       SYSIBM    T     DSNDB06  SYSDBAS  6      12
SYSKEYS        SYSIBM    T     DSNDB06  SYSDBAS  6       7
SYSINDEXPART   SYSIBM    T     DSNDB06  SYSDBAS  6      16
SYSINDEXES     SYSIBM    T     DSNDB06  SYSDBAS  6      26
SYSFORIGNKEYS  SYSIBM    T     DSNDB06  SYSDBAS  6       7
SYSCOLUMNS    SYSIBM    T     DSNDB06  SYSDBAS  6      19
SYSCOLAUTH     SYSIBM    T     DSNDB06  SYSDBAS  6      10
```

Implementation of TOWNER Using RIX/REXX

```
/* RIX REXX EXEC TOWNER -- using embedded SQL and ISPF services */
arg createdby /* Obtain the creator's name as a parameter */
address RIX /* Route host commands to RIX for execution */

/* You denote REXX host variables with the standard colon prefix.*/
"rlx declare tblnames cursor for
select name, creator, type, dbname, tname, dbid, colcount
from sysibm.systables
where createdby = :createdby"

/* Address RIX recognizes all ISPF dialog service names */
"TBCREATE TBLNAMES
 NAMES(NAME, CREATOR, TYPE, DBNAME, TNAME, DBID, COLCOUNT)
 NOWRITE"

"rlx open tblnames" /* Produce SQL query result */

/* RIX FETCHES values directly thru memory into 'host' */
/* variables which RIX, ISPF and the REXX interpreter share. */
"rlx fetch tblnames into
name, :creator, :type, :dbname, :tname, :dbid, :colcount"

/* RIX updates all the host variables comprising the SQLCA */
Do while sqlcode = 0 /* While FETCHES are successful */
"TBADD TBLNAMES"
/* RIX recognizes statements it's already processed to further*/
/* improve performance. A reexecuted RIX SQL statement runs */
/* at 'static' SQL speed. */
"rlx fetch tblnames into
name, :creator, :type, :dbname, :tname, :dbid, :colcount"
END
"rlx close tblnames" /* close the cursor */

"TBTOP TBLNAMES"
Do while rc = 0 /* until user signals end or return */
"TBDISPL TBLNAMES PANEL(TBLNAMES)"
End
"TBEND TBLNAMES"
exit rc
```

Implementation of TOWNER Using RLX/ISPF

```
arg createdby      /* Obtain the creator's name as a parameter */
address RLX        /* Route host commands to RLX for execution */

/* Flow the SQL query result into an ISPF table -- with a single */
/* statement -- using the DECLARE ISPF TABLE service           */

"rlx declare tblnames ispf table for
select name, creator, type, dbname, tname, dbid, colcount
from sysibm.systables
where createdby = :createdby"

Do while rc = 0
  "rlx tbdisp1 tblnames panel(rlx)"
end

Exit rc
```

RLX Feature Summary

- NULLs and host variables fully supported***
- Automatic conversion between internal and external data formats***
- SET and ROW oriented processing***
- SQL Communications Area feedback after each RLX SQL statement***

41

Human engineered

- Interactive diagnostic facilities pinpoint errors and speed their correction***
- Profile facilities customize RLX operation***
- ISPF split screen is fully supported***
- RLX can run concurrently in both screens***

Syntax errors detected by the RLX semantic parser:

```
----- RLX SQL Parser Detected an Error ----- ROW 1 OF 4
Command ==>                                     Scroll ==> HALF
PSQ011 - Null indicator variable reference invalid within search condition

SQL statement location
Exec containing statement ==> EXAMPLE
RLX module detecting error ==> PSQLSC

SELECT CN1 , CN2 , CN3 , CN4 FROM RLXTBL WHERE CN1 = :HV1:IV1 ORDER BY CN1
                                         *
DESC

*** Press END or RETURN key to resume RLX dialog execution / termination ***
```

Data errors:

For example: Date value inconsistent with specified date format

43

```
----- Data Error Recognized ----- ROW 1 OF 6
Command ==>                               Scroll ==> HALF
PSR032 - Expected dash - between ISO/JIS date components

SQL statement location
  Exec containing statement ==> RLXSINS3
  RLX module detecting error ==> PSQFDTC

Host Variable / SQL Variable Profile
Host Variable Name      ==> DATE
Host Variable Value     ==> 12/04/1991
Host datatype origin    ==> DATE
SQL Data Type           ==> DATE                Nulls ==> Y
SQL Data Length         ==> 10                (Precision when Decimal)
SQL Data Scale          ==> 0                (0 when not Decimal)
Date or Time format name ==> ISO                (Blank when not date or time)
Date or Time format     ==> YYYY-MM-DD        (Blank when not date or time)

Statement executing when the Data Error was detected

INSERT INTO RLXREL3 (INTEGER , DATE , TIME , TIMESTAMP , FLOAT VALUES (:DV1
:IV1 , :DV2 :IV2 , :DV3 :IV3 , :DV4 :IV4 , :DV5 :IV5)

PSQ105 - No row inserted because column value was invalid for its datatype
```

RLX Administration

```
V2.3 ----- RLX Administrative Facility -----
Option ==>

1  RLX Libs      - Update RLX target libraries      Userid   - RAI4
2  RLX Libs2    - Update RLX target libraries      ISPF Ver - 3.2
3  IBM Libs     - Update IBM supplied load libraries   Op System- MVS/ESA
4  Job Parm    - Update tailored jobstream parameters CPU ID   - 2123
5  Defaults    - Update RLX/DB2 subsystem defaults CPU Model- 3090
6  Plans       - Tailor and bind RLX application plan(s)
7  Passwords  - Update RLX product passwords
8  Create     - RLX Demonstration tables
9  Load       - RLX Demonstration tables from sequential files
10 Demo       - Conduct RLX Installation Verification Procedures
11 Profiles 1 - Define RLX Session Profiles for shared usage
12 Profile2 2 - RLX Shared Profile Maintenance Facility (REXX dialog)
13 Extra Copies- Install additional RLX copies
                    (on the same or different DB2 subsystem)
14 Tools     - Portfolio of RLX tools for developers and DBAs
X  Exit      - Leave RLX Administration Menu
```

Enter END to exit

Copyright (c) Relational Architects, Inc. - 1987,1992 - All rights reserved

RLX Profile Defaults

```
----- RLX User Profile Facility 2 -----  
Command ==>  
  
RLX068 - Your RLX Session Profile was updated successfully  
CONTROL service settings  
When RLX Error ==> F (C - Cancel F - Filter R - Return)  
Error Panel ==> D (D - Display N - No Display)  
When ISPF Error ==> C (C - Cancel R - Return)  
Cursor Scope ==> L (L - Local G - Global)  
Variable Scope ==> L (L - Local G - Global)  
Statement Scope ==> L (L - Local G - Global)  
Maximum Digits ==> 9 (Before using scientific notation)  
Tracing Option ==> 0 (Integer value between 0 and 255)  
RC/LASTCC value ==> S (S - SQLCODE, N - Nonzero, Z - Zero)
```

```
----- RLX User Profile Facility 1 -----  
Command ==>  
  
RLX068 - Your RLX Session Profile was updated successfully  
  
Environmental Parameters  
DB2 Subsystem ==> DSN (DB2 subsystem with which to connect)  
Retry Count ==> 0 (Connection retries if DB2 is not active)  
Max CPU Time ==> 0 (in seconds before work is suspended)  
Max Idle Time ==> 0 (in minutes before thread is terminated)  
  
Application Plan Selection  
Max Cursors ==> 50 (Maximum number of cursors referenced)  
CSRs WITH HOLD ==> 00 (Max cursors maintained across COMMITs)  
Max Update ==> 50 (Max DELETE, INSERT, and UPDATE statements)  
Isolation Lvl ==> C (C - Cursor Stability R - Repeatable Read)  
  
Data Format Preferences  
Numeric Format ==> E (I - Integer E - Edited Decimal)  
Date format ==> I (I - ISO,U - USA,E - EUR,J - JIS,L - Local)  
Time format ==> I (I - ISO,U - USA,E - EUR,J - JIS,L - Local)
```