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

<table>
<thead>
<tr>
<th>American Hospital Association</th>
<th>Ericsson (Sweden)</th>
<th>National Westminster Bancorp</th>
</tr>
</thead>
<tbody>
<tr>
<td>American President Lines</td>
<td>The Equitable Life Assurance Soc.</td>
<td>Norwegian Telecom</td>
</tr>
<tr>
<td>Australian Telecom</td>
<td>Federal Government Agencies</td>
<td>NYNEX</td>
</tr>
<tr>
<td>Bank of Liechtenstein</td>
<td>Fireman's Fund Insurance Co.</td>
<td>PARS</td>
</tr>
<tr>
<td>Blue Cross Blue Shield</td>
<td>The Franklin Mint</td>
<td>The Pillsbury Company</td>
</tr>
<tr>
<td>British Columbia Telephone</td>
<td>G E Information Services</td>
<td>State Bank of Sweden</td>
</tr>
<tr>
<td>British Telecom</td>
<td>Glaxo, Inc.</td>
<td>Tandy Corporation</td>
</tr>
<tr>
<td>CalFarm Insurance</td>
<td>The Home Depot</td>
<td>United States Fidelity &amp; Guaranty</td>
</tr>
<tr>
<td>Ciba - Geigy</td>
<td>IBM Corporation</td>
<td>University of California</td>
</tr>
<tr>
<td>Credit Suisse (Switzerland)</td>
<td>Iowa Public Service</td>
<td>U S A A</td>
</tr>
<tr>
<td>Daimler Benz (Germany)</td>
<td>Los Alamos National Labs</td>
<td>VISA International</td>
</tr>
<tr>
<td>Depository Trust Company (NY)</td>
<td>Los Angeles Water and Power</td>
<td></td>
</tr>
<tr>
<td>Dow Corning</td>
<td>MCI / Telecom*USA</td>
<td></td>
</tr>
<tr>
<td>Dresdner Bank (Germany)</td>
<td>Mead Corporation</td>
<td></td>
</tr>
</tbody>
</table>
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
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

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Owner</th>
<th>Type</th>
<th>DB Name</th>
<th>TS Name</th>
<th>DB ID</th>
<th>Colcount</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSCOPY</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSCOPY</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>SYSFIELDS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>SYSTABLESPACE</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>23</td>
</tr>
<tr>
<td>SYSTABLES</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>31</td>
</tr>
<tr>
<td>SYSTABLEPART</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>SYSTABAUTH</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>SYSYNONYMS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>SYSRBSLS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>SYSLINKS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>SYSKEYS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>SYSEXTPART</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>SYSEXTPART</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>SYSFOREIGNKEYS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>SYSCOLUMNS</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>SYSCOLAUTH</td>
<td>SYSI BM</td>
<td>T</td>
<td>DSND606</td>
<td>SYSDBA5</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>
Implementation of TOWNER
Using RLX/REXX

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

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

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

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

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

/* RLX updates all the host variables comprising the SQLCA */
Do while sqlcode = 0 /* While FETCHes are successful */
  "TBADD TBLNAMES"

  /* RLX recognizes statements it's already processed to further*/
  /* improve performance. A reexecuted RLX SQL statement runs */
  /* at 'static' SQL speed. */

  "rlx fetch tblnames into
    name, :creator, :type, :dbname, :tsname, :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
Using RLX/ISPF

Implementation of TOWER
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

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

---------------------------------- 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


PSQ105 - No row inserted because column value was invalid for its datatype
# RLX Administration

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RLX Libs</td>
<td>Update RLX target libraries</td>
</tr>
<tr>
<td>2</td>
<td>RLX Libs2</td>
<td>Update RLX target libraries</td>
</tr>
<tr>
<td>3</td>
<td>IBM Libs</td>
<td>Update IBM supplied load libraries</td>
</tr>
<tr>
<td>4</td>
<td>JobParms</td>
<td>Update tailored jobstream parameters</td>
</tr>
<tr>
<td>5</td>
<td>Defaults</td>
<td>Update RLX/DB2 subsystem defaults</td>
</tr>
<tr>
<td>6</td>
<td>Plans</td>
<td>Tailor and bind RLX application plan(s)</td>
</tr>
<tr>
<td>7</td>
<td>Passwords</td>
<td>Update RLX product passwords</td>
</tr>
<tr>
<td>8</td>
<td>Create</td>
<td>RLX Demonstration tables</td>
</tr>
<tr>
<td>9</td>
<td>Load</td>
<td>RLX Demonstration tables from sequential files</td>
</tr>
<tr>
<td>10</td>
<td>Demo</td>
<td>Conduct RLX Installation Verification Procedures</td>
</tr>
<tr>
<td>11</td>
<td>Profiles 1</td>
<td>Define RLX Session Profiles for shared usage</td>
</tr>
<tr>
<td>12</td>
<td>Profile2 2</td>
<td>RLX Shared Profile Maintainence Facility (REXX dialog)</td>
</tr>
<tr>
<td>13</td>
<td>Extra Copies</td>
<td>Install additional RLX copies (on the same or different DB2 subsystem)</td>
</tr>
<tr>
<td>14</td>
<td>Tools</td>
<td>Portfolio of RLX tools for developers and DBAs</td>
</tr>
<tr>
<td>X</td>
<td>Exit</td>
<td>Leave RLX Administration Menu</td>
</tr>
</tbody>
</table>

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
- Error Panel ==> D
- When TSPF Error ==> C
- Cursor Scope ==> L
- Variable Scope ==> L
- Statement Scope ==> L
- Maximum Digits ==> 9
- Tracing Option ==> 0
- RC/LASTCC value ==> S

(C - Cancel  F - Filter  R - Return)
(D - Display  N - No Display)
(C - Cancel  R - Return)
(L - Local  G - Global)
(L - Local  G - Global)
(L - Local  G - Global)
(Before using scientific notation)
(Integer value between 0 and 255)
(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
- Retry Count ==> 0
- Max CPU Time ==> 0
- Max Idle Time ==> 0

(DB2 subsystem with which to connect)
(Connection retries if DB2 is not active)
(in seconds before work is suspended)
(in minutes before thread is terminated)

Application Plan Selection:
- Max Cursors ==> 50
- CSRs WITH HOLD ==> 00
- Max Update ==> 50
- Isolation Lv1 ==> C

(Maximum number of cursors referenced)
(Max cursors maintained across COMMITs)
(Max DELETE, INSERT, and UPDATE statements)
(C - Cursor Stability  R - Repeatable Read)

Data Format Preferences:
- Numeric Format ==> E
- Date format ==> I
- Time format ==> I

(I - Integer  E - Edited Decimal)
(I - ISO,U - USA,E - EUR,J - JIS,L - Local)
(I - ISO,U - USA,E - EUR,J - JIS,L - Local)