Reentrant Layered Logic Tables (RLLT)

Colin James III, Principal Scientist, CEC Services, LLC, 1613 Morning Dr, Loveland CO 80538-4410, rltt@cec-services.com, C: 719.210.9534 , F: 970.593.1350

Keywords

automation, dispatching, Reentrant Layered Logic Tables, RLTT, process programming, scheduling, SQL, water treatment

Abstract

The water treatment industry as a manufacturing sector continues to be a candidate for the programming of process automation. The rules for dispatching such processes are mapped directly into the rows and columns of a logic table in a relational database. Layers of logic tables apply to the more complex processing instructions for water treatment. The specific area of theoretical application is the sequential control of valves for the automation of water cooling towers. For reentrant logic tables of eight rows and columns at eight layers, the possible complex processes number over 2^122. Metrics on desk top computers show that reentrant layered logic tables (RLLT) perform fast enough in real time to process eight layers of such tables in 0.08 seconds. In structured query language (SQL), the generic access form of layered logic tables is EXISTS (SELECT … FROM … WHERE SUBSTR … = [valid switch]).

Reentrant Layered Logic Tables

A logic table contains switches to instruct a process controller as to which process to invoke per logic sequence position, as in Table 1 below.

The hierarchy of layered logic tables below follow the order of six subsets derived from requirements in the order of most specific to most abstract in tables below as: 1. Procedures; 2. Processes; 3. Tasks; 4. Units; 5. Components; and 6. Requirements.

Logic Table 1 for Processes:

Row	**Column Position Number**		
Number	Process 1	Process 2	Process M
Procedure 1	Process Switches		
Procedure 2			
Procedure N			

Logic Table 2 for Tasks:

Row	**Column Position Number**		
Number	Task 1	Task 2	Task M
Process 1	Task Switches		
Process 2			
Process N			

Logic Table 3 for Units:

Row Number	**Column Position Number** Unit 1	Unit 2	Unit M
Task 1	Unit Switches		
Task 2			
Task N			

Logic Table 4 for Components:

Row Number	**Column Position Number** Comp1	Comp2	Comp M
Units 1	Component Switches		
Units 2			
Units N			

Logic Table 5 for Requiremetns:

Row Number	**Column Position Number** Reqt 1	Reqt 2	Reqt M
Comp 1	Requirement Switches		
Comp 2			
Comp N			

Generic SQL Code for Layered Logic Tables

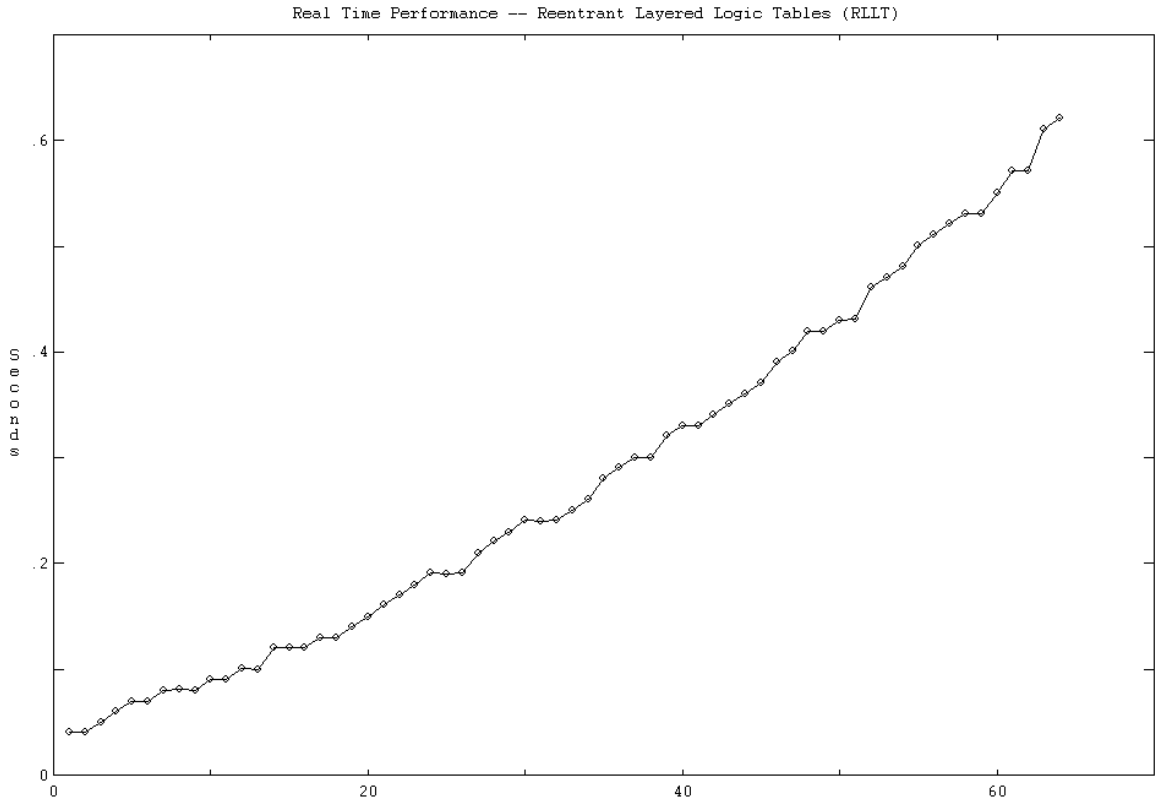
The generic SQL code for layered logic tables 1-5 is below. What is returned from the nested SQL query are the procedures and respective procedure switches to be updated.

```
SELECT   L5.procedure,   S5.switch
FROM     process AS L5, type_switch as S5
WHERE    SUBSTR( L5.switches, L4.process, 1) = S5.switch
AND EXISTS
    (SELECT  L4.process, S4.switch
    FROM     task as L4, type_switch as S4
    WHERE    SUBSTR( L4.switches, L3.task, 1) = S4.switch
    AND EXISTS
        (SELECT  L3.task, S3.switch
        FROM     unit as L3, type_switch as S3
        WHERE    SUBSTR( L3.switches, L2.unit, 1) = S3.switch
        AND EXISTS
            (SELECT  L2.unit, S2.switch
            FROM     component as L2, type_switch as S2
            WHERE    SUBSTR( L3.switches, L2.unit, 1) = S2.switch
            AND EXISTS
                (SELECT  L1.component, S1.switch
                FROM     requirement as L1, type_switch as S1
                WHERE    SUBSTR( L1.switch, user_input_reqt, 1) = S1.switch ) ) ) )
```

Performance in Real Time

For a reentrant logic table of eight switches in seven rows, the number of possible switch combinations is 8! or 40320. For eight such tables with eight columns to be layered and reentrant, the number of possible switch combinations increases to (8!)^8 or about 10^36 or 2^122 . By contrast, for a reentrant logic table of 64 switches, 63 rows, and 64 columns for up to 64 levels of layers, the number of possible switch combinations is larger by a factor of about 155 at (64!)^64 or about 10^5696 or 2^18943.

The graph below is for a logic table of four switches in three rows and 64 columns for up to 64 levels of layers. The number of possible switch combinations is therefore (4!)^64 or about 10^88 or 2^293.



The graph shows that the real time performance for a logic table of four switches in three rows and eight columns for up to eight levels of layers is 0.081 seconds. The number of possible switches is (4!)^8 or about 10^11 or 2^36.

The graph also shows that the real time performance for a logic table of four switches in three row and 64 columns for 64 levels of layers is 0.62 seconds. The number of possible switch combinations is again about 2^293. That time for the 64 levels is about eight times greater than the time for the eight levels. The number of possible switch combinations is also about eight times greater than the number for the eight levels. The graph shows this relationship as being nearly linear from eight to 64 levels.

The tests were performed on a desktop computer with 733 MHz processor, 768 MB RAM, and 80 GB hard disk. The relational database used was IBM DB2 7 with switches set to maximum size and performance.

The data above is contained in the following computer program:

```
! SQL Graph - TrueBASIC® Source Code for Graph
!
! Real time performance in 0.6 seconds for 64-layers of logic tables
!
! © Copyright 2003 by CEC Services, LLC All Rights Reserved

LIBRARY "..\TBLibs\SGLib.trc"

DIM layers(64), secs(64)

MAT READ layers, secs

DATA 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,
DATA 17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32
DATA 33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48
DATA 49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64

DATA 0.040, 0.040, 0.050, 0.060, 0.070, 0.070, 0.080, 0.081
DATA 0.080, 0.090, 0.090, 0.101, 0.100, 0.120, 0.120, 0.120
DATA 0.130, 0.130, 0.140, 0.150, 0.161, 0.170, 0.180, 0.191
DATA 0.190, 0.191, 0.210, 0.221, 0.230, 0.241, 0.240, 0.241
DATA 0.250, 0.261, 0.280, 0.291, 0.300, 0.300, 0.321, 0.330
DATA 0.330, 0.341, 0.351, 0.360, 0.371, 0.391, 0.401, 0.420
DATA 0.420, 0.430, 0.431, 0.461, 0.471, 0.481, 0.501, 0.511
DATA 0.521, 0.531, 0.531, 0.551, 0.571, 0.571, 0.611, 0.621

CALL SetText ( "Real Time Performance -- Reentrant Layered Logic Tables (RLLT)",
               "", "Seconds" )
CALL DataGraph ( layers, secs, 4, 1, "black black" )

END
```

Conclusion

RLLT is ideally suited for programming processes in the water treatment industry such as valves in a water tower. The generalized implementation is:

```
EXISTS
    (SELECT  switches available, logic switches, items returned
    FROM     type of switches available, logic table of interest
    WHERE    SUBSTR( logic switches, input point, 1) = type of switches available)
```

Acknowledgments

Thanks are due to Larry Cagg of Cagg Enterprises for helpful discussions.

References

James, C. 2003.3, “Layered Logic Tables (LLT)”, unpublished, CEC Services, LLC, Loveland CO.

James, C. 2003.2, "Software Factory", unpublished brochure, CEC Services, LLC, Loveland CO.

James, C. 2003.1, "Software Factory", unpublished poster, CEC Services, LLC, Loveland CO.

James, C. 2002.5, "The Software Development Methodology [SDM]", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.4, "Reentrant Logic Table Technology", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.3, "Static and Dynamic Driver Triggers", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.2, "Additional Information", unpublished, CEC Services, LLC, Loveland CO.

James, C. 2002.1, “Implementation Details for Multiple Billing”, CEC Services, LLC, Loveland CO.

James, C. 2001.2, "Report Accounts [RA] v 1.2 Inventory / Point of Sale", unpublished, CEC Services, LLC, Loveland CO.

James, C., 2001.1, "Report Accounts [RA] v 1.2", unpublished, CEC Services, LLC, Loveland CO.

James, C. 1999.1, “Recent Advances in Logic Tables for Reusable Database Engines”, Proceedings of the American Society of Mechanical Engineers International, Petroleum Division, 75th Anniversary Conference, Energy Sources Technology Conference & Exhibition, Houston, Texas.

James, C., 1998.5, "Multiple and Self-Modifying Logic Tables with Queries", unpublished, CEC Services, LLC, Loveland CO.

James, C. 1998.4, “A Reusable Database Engine for Accounting Arithmetic”, Proceedings of The Third Biennial World Conference on Integrated Design & Process Technology, Vol. 2, pp. 25-30, Berlin, Germany.

James, C., 1998.3, "Competency test for CEC Services, LLC", unpublished, CEC Services, LLC, Loveland CO.

James, C. 1998.2, “Theory and Application of Logic Tables in Relational Database Engines”, Doctoral Dissertation, Pacific Western University, Los Angeles.

James, C., 1998.1, "Ticket Reservations [TR] ver 1.1", unpublished, CEC Services, LLC, Loveland CO.

James, C., 1997.2, "User Documentation", unpublished, CEC Services, LLC, Loveland CO.

James, C., 1997.1, "Logic Table Design for Reports in RA", unpublished, CEC Services, LLC, Loveland CO.