

JOE CELKO'S

THINKING IN SETS

Auxiliary, Temporal, and Virtual Tables in SQL

Joe Celko



ELSEVIER

AMSTERDAM • BOSTON HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier

MORGAN KAUFMANN PUBLISHERS

CONTENTS

Preface	xvii
SQL Is Declarative, Not Procedural	1
1.1 Different Programming Models	2
1.2 Different Data Models	4
1.2.1 Columns Are Not Fields	5
1.2.2 Rows Are Not Records	7
1.2.3 Tables Are Not Files	11
1.2.4 Relational Keys Are Not Record Locators	13
1.2.5 Kinds of Keys	15
1.2.6 Desirable Properties of Relational Keys	17
1.2.7 Unique But Not Invariant	18
1.3 Tables as Entities	19
1.4 Tables as Relationships	20
1.5 Statements Are Not Procedures	20
1.6 Molecular, Atomic, and Subatomic Data Elements	21
1.6.1 Table Splitting	21
1.6.2 Column Splitting	22
1.6.3 Temporal Splitting	24
1.6.4 Faking Non-INF Data	24
1.6.5 Molecular Data Elements	25
1.6.6 Isomer Data Elements	26
1.6.7 Validating a Molecule	27
2 Hardware, Data Volume, and Maintaining Databases	29
2.1 Parallelism	30
2.2 Cheap Main Storage	31
2.3 Solid-State Disk	32
2.4 Cheaper Secondary and Tertiary Storage	32
2.5 The Data Changed	33
2.6 The Mindset Has Not Changed	33

CONTENTS

3	Data Access and Records	37
3.1	Sequential Access	38
3.1.1	Tape-Searching Algorithms	38
3.2	Indexes	39
3.2.1	Single-Table Indexes	40
3.2.2	Multiple-Table Indexes	40
3.2.3	Type of Indexes	41
3.3	Hashing	41
3.3.1	Digit Selection	42
3.3.2	Division Hashing	42
3.3.3	Multiplication Hashing	42
3.3.4	Folding	42
3.3.5	Table Lookups	43
3.3.6	Collisions	43
3.4	Bit Vector Indexes	44
3.5	Parallel Access	44
3.6	Row and Column Storage	44
3.6.1	Row-Based Storage	44
3.6.2	Column-Based Storage	45
3.7	JOIN Algorithms	46
3.7.1	Nested-Loop Join Algorithm	47
3.7.2	Sort-Merge Join Method	47
3.7.3	Hash Join Method	48
3.7.4	Shin's Algorithm	48
4	Lookup Tables	51
4.1	Data Element Names	52
4.2	Multiparameter Lookup Tables	55
4.3	Constants Table	56
4.4	OTLT or MUCK Table Problems	59
4.5	Definition of a Proper Table	62
5	Auxiliary Tables	65
5.1	Sequence Table	65
5.1.1	Creating a Sequence Table	67

CONTENTS

5.1.2	Sequence Constructor	68
5.1.3	Replacing an Iterative Loop	69
5.2	Permutations	72
5.2.1	Permutations via Recursion	72
5.2.2	Permutations via CROSS JOIN	73
5.3	Functions	75
5.3.1	Functions without a Simple Formula	76
5.4	Encryption via Tables	78
5.5	Random Numbers	79
5.6	Interpolation	83
Views		87
6.1	Mullins VIEW Usage Rules	88
6.1.1	Efficient Access and Computations	88
6.1.2	Column Renaming	89
6.1.3	Proliferation Avoidance	90
6.1.4	The VIEW Synchronization Rule	90
6.2	Updatable and Read-Only VIEWS	91
6.3	Types of VIEWS	93
6.3.1	Single-Table Projection and Restriction	93
6.3.2	Calculated Columns	93
6.3.3	Translated Columns	94
6.3.4	Grouped VIEWS	95
6.3.5	UNIONed VIEWS	96
6.3.6	JOINS in VIEWS	98
6.3.7	Nested VIEWS	98
6.4	Modeling Classes with Tables	100
6.4.1	Class Hierarchies in SQL	100
6.4.2	Subclasses via ASSERTIONS and TRIGGERS	103
6.5	How VIEWS Are Handled in the Database System	103
6.5.1	VIEW Column List	104
6.5.2	VIEW Materialization	104
6.6	In-Line Text Expansion	105
6.7	WITH CHECK OPTION Clause	106
6.7.1	WITH CHECK OPTION as CHECK() Clause	110
6.8	Dropping VIEWS	112

CONTENTS

6.9	Outdated Uses for VIEWS	113
6.9.1	Domain Support	113
6.9.2	Table Expression VIEWS	114
6.9.3	VIEWS for Table Level CHECKQ Constraints	114
6.9.4	One VIEW per Base Table	115
Virtual Tables		117
7.1	Derived Tables	118
7.1.1	Column Naming Rules	118
7.1.2	Scoping Rules	119
7.1.3	Exposed Table Names	121
7.1.4	LATERALO Clause	122
7.2	Common Table Expressions	124
7.2.1	Nonrecursive CTEs	124
7.2.2	Recursive CTEs	126
7.3	Temporary Tables	128
7.3.1	ANSI/ISO Standards	128
7.3.2	Vendors Models	128
7.4	The Information Schema	129
7.4.1	The INFORMATION_SCHEMA Declarations	130
7.4.2	A Quick List of VIEWS and Their Purposes	130
7.4.3	DOMAIN Declarations	132
7.4.4	Definition Schema	132
7.4.5	INFORMATION_SCHEMA Assertions	135
Complicated Functions via Tables		137
8.1	Functions without a Simple Formula	137
8.1.1	Encryption via Tables	138
8.2	Check Digits via Tables	139
8.2.1	Check Digits Defined	139
8.2.2	Error Detection versus Error Correction	140
8.3	Classes of Algorithms	141
8.3.1	Weighted-Sum Algorithms	141
8.3.2	Power-Sum Check Digits	144
8.3.3	Luhn Algorithm	145
8.3.4	Dihedral Five Check Digit	146

8.4	Declarations, Not Functions, Not Procedures	148
8.5	Data Mining for Auxiliary Tables	152
Temporal Tables		155
9.1	The Nature of Time	155
9.1.1	Durations, Not Chronons	156
9.1.2	Granularity	158
9.2	The ISO Half-Open Interval Model	159
9.2.1	Use of NULL for "Eternity"	161
9.2.2	Single Timestamp Tables	162
9.2.3	Overlapping Intervals	164
9.3	State Transition Tables	174
9.4	Consolidating Intervals	178
9.4.1	Cursors and Triggers	180
9.4.2	OLAP Function Solution	181
9.4.3	CTE Solution	181
9.5	Calendar Tables	182
9.5.1	Day of Week via Tables	183
9.5.2	Holiday Lists	184
9.5.3	Report Periods	186
9.5.4	Self-Updating Views	186
9.6	History Tables	188
9.6.1	Audit Trails	189
0	Scrubbing Data with Non-1 NF Tables	191
10.1	Repeated Groups	192
10.1.1	Sorting within a Repeated Group	195
10.2	Designing Scrubbing Tables	198
10.3	Scrubbing Constraints	201
10.4	Calendar Scrubs	202
10.4.1	Special Dates	203
10.5	String Scrubbing	204
10.6	Sharing SQL Data	205
10.6.1	A Look at Data Evolution	206
10.6.2	Databases	207

CONTENTS

10.7	Extract, Transform, and Load Products	208
10.7.1	Loading Data Warehouses	209
10.7.2	Doing It All in SQL	211
10.7.3	Extract, Load, and then Transform	212
11	Thinking in SQL	215
11.1	Warm-up Exercises	216
11.1.1	The Whole and Not the Parts	216
11.1.2	Characteristic Functions	217
11.1.3	Locking into a Solution Early	219
11.2	Heuristics	220
11.2.1	Put the Specification into a Clear Statement	220
11.2.2	Add the Words "Set of All..." in Front of the Nouns	220
11.2.3	Remove Active Verbs from the Problem Statement	221
11.2.4	You Can Still Use Stubs	221
11.2.5	Do Not Worry about Displaying the Data	222
11.2.6	Your First Attempts Need Special Handling	223
11.2.7	Do Not Be Afraid to Throw Away Your First Attempts at DDL	223
11.2.8	Save Your First Attempts at DML	224
11.2.9	Do Not Think with Boxes and Arrows	225
11.2.10	Draw Circles and Set Diagrams	225
11.2.11	Learn Your Dialect	226
11.2.12	Imagine that Your WHERE Clause Is "Super Amoeba"	227
11.2.13	Use the Newsgroups, Blogs, and Internet	228
11.3	Do Not Use BIT or BOOLEAN Flags in SQL	228
11.3.1	Flags Are at the Wrong Level	229
11.3.2	Flags Confuse Proper Attributes	230
12	Group Characteristics	235
12.1	Grouping Is Not Equality	237
12.2	Using Groups without Looking Inside	238
12.2.1	Semiset-Oriented Approach	239
12.2.2	Grouped Solutions	240

	12.2.3	Aggregated Solutions	241
12.3		Grouping; over Time	242
	12.3.1	Piece-by-Piece Solution	243
	12.3.2	Data as a Whole Solution	244
12.4		Other Tricks with HAVING Clauses	245
12.5		Grouping/s, Rollups, and Cubes	247
	12.5.1	GROUPING SET Clause	247
	12.5.2	The ROLLUP Clause	249
	12.5.3	The CUBE Clause	249
	12.5.4	A Footnote about Super Grouping	250
12.6		The WINDOW Clause	250
	12.6.1	The PARTITION BY Clause	251
	12.6.2	The ORDER BY Clause	252
	12.6.3	The RANGE Clause	253
	12.6.4	Programming Tricks	253
13		ling Sf>ecificati()ns into Code	255
13.1		Signs of IBad SQL	255
	13.1.1	Is the Code Formatted Like Another Language?	256
	13.1.2	Assuming Sequential Access	256
	13.1.3	Cursors	256
	13.1.4	Poor Cohesion	257
	13.1.5	Table-Valued Functions	257
	13.1.6	Multiple Names for the Same Data Element	258
	13.1.7	Formatting in the Database	258
	13.1.8	Keeping Dates in Strings	259
	13.1.9	BIT Flags, BOOLEAN, and Other Computed Columns	259
	13.1.10	Attribute Splitting Across Columns	259
	13.1.11	Attribute Splitting Across Rows	259
	13.1.12	Attribute Splitting Across Tables	260
13.2		Methods of Attack	260
	13.2.1	Cursor-Based Solution	261
	13.2.2	Semiset-Oriented Approach	262
	13.2.3	Pure Set-Oriented AnDroach	264

13.2.4	Advantages of Set-Oriented Code	264
13.3	Translating Vague Specifications	265
13.3.1	Go Back to the DDL	266
13.3.2	Changing Specifications	269
14	Using Procedure and Function Calls	273
14.1	Clearing out Spaces in a String	273
14.1.1	Procedural Solution #1	274
14.1.2	Functional Solution #1	275
14.1.3	Functional Solution #2	278
14.2	The PRD() Aggregate Function	280
14.3	Long Parameter Lists in Procedures and Functions	282
14.3.1	The IN() Predicate Parameter Lists	283
15	Numbering Rows	287
15.1	Procedural Solutions	287
15.1.1	Reordering on a Numbering Column	289
15.2	OLAP Functions	291
15.2.1	Simple Row Numbering	291
15.2.2	RANK()andDENSE_RANK()	292
15.3	Sections	293
16	Keeping Computed Data	297
16.1	Procedural Solution	297
16.2	Relational Solution	299
16.3	Other Kinds of Computed Data	299
17	Triggers for Constraints	301
17.1	Triggers for Computations	301
17.2	Complex Constraints via CHECK() and CASE Constraints	302
17.3	Complex Constraints via VIEWS	305
17.3.1	Set-Oriented Solutions	306
17.4	Operations on VIEWS as Constraints	308
17.4.1	The Basic Three Operations	308
17.4.2	WITH CHECK OPTION Clause	308

17.4.3	WITH CHECK OPTION as CHECK() clause	313
17.4.4	How VIEWS Behave	314
17.4.5	UNIONed VIEWS	316
17.4.6	Simple INSTEAD OF Triggers	317
17.4.7	Warnings about INSTEAD OF Triggers	321
18	Procedural and Data Driven Solutions	323
18.1	Removing Letters in a String	323
18.1.1	The Procedural Solution	324
18.1.2	Pure SQL Solution	325
18.1.3	Impure SQL Solution	326
18.2	Two Approaches to Sudoku	326
18.2.1	Procedural Approach	327
18.2.2	Data-Driven Approach	327
18.2.3	Handling the Given Digits	329
18.3	Data Constraint Approach	331
18.4	Bin Packing Problems	335
18.4.1	The Procedural Approach	336
18.4.2	The SQL Approach	336
18.5	Inventory Costs over Time	339
18.5.1	Inventory UPDATE Statements	343
18.5.2	Bin Packing Returns	344
	Index	349