

1. Record Nr.	UNINA9910784543103321
Autore	Celko Joe
Titolo	Joe Celko's SQL programming style [[electronic resource]] / Joe Celko
Pubbl/distr/stampa	Amsterdam ; ; Boston, : Elsevier, 2005
ISBN	1-281-00810-9 9786611008109 1-4237-2245-0 0-08-047883-2
Edizione	[1st edition]
Descrizione fisica	1 online resource (236 p.)
Collana	The Morgan Kaufmann Series in Data Management Systems
Disciplina	005.13/3
Soggetti	SQL (Computer program language) Query languages (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references (p. [203]-206) and index.
Nota di contenuto	Front cover; Title page; Copyright page; Table of contents; front matter; Introduction; 1.1 Purpose of the Book; 1.2 Acknowledgments; 1.3 Corrections, Comments, and Future Editions; First chapter; 1. Names and Data Elements; 1.1 Names; 1.1.1 Watch the Length of Names; 1.1.2 Avoid All Special Characters in Names; 1.1.3 Avoid Quoted Identifiers; 1.1.4 Enforce Capitalization Rules to Avoid Case- Sensitivity Problems; 1.2 Follow the ISO-11179 Standards Naming Conventions; 1.2.1 ISO-11179 for SQL; 1.2.2 Levels of Abstraction; 1.2.3 Avoid Descriptive Prefixes; 1.2.4 Develop Standardized Postfixes 1.2.5 Table and View Names Should Be Industry Standards, Collective, Class, or Plural Nouns1.2.6 Correlation Names Follow the Same Rules as Other Names . . . Almost; 1.2.7 Relationship Table Names Should Be Common Descriptive Terms; 1.2.8 Metadata Schema Access Objects Can Have Names That Include Structure Information; 1.3 Problems in Naming Data Elements; 1.3.1 Avoid Vague Names; 1.3.2 Avoid Names That Change from Place to Place; 1.3.3 Do Not Use Proprietary Exposed Physical Locators; 2. Fonts, Punctuation, and Spacing; 2.1 Typography and Code 2.1.1 Use Only Upper- and Lowercase Letters, Digits, and Underscores for Names2.1.2 Lowercase Scalars Such as Column Names, Parameters,

and Variables; 2.1.3 Capitalize Schema Object Names; 2.1.4 Uppercase the Reserved Words; 2.1.5 Avoid the Use of CamelCase; 2.2 Word Spacing; 2.3 Follow Normal Punctuation Rules; 2.4 Use Full Reserved Words; 2.5 Avoid Proprietary Reserved Words if a Standard Keyword Is Available in Your SQL Product; 2.6 Avoid Proprietary Statements if a Standard Statement Is Available; 2.7 Rivers and Vertical Spacing; 2.8 Indentation; 2.9 Use Line Spacing to Group Statements

3. Data Declaration Language

3.1 Put the Default in the Right Place; 3.2 The Default Value Should Be the Same Data Type as the Column; 3.3 Do Not Use Proprietary Data Types; 3.4 Place the PRIMARY KEY Declaration at the Start of the CREATE TABLE Statement; 3.5 Order the Columns in a Logical Sequence and Cluster Them in Logical Groups; 3.6 Indent Referential Constraints and Actions under the Data Type; 3.7 Give Constraints Names in the Production Code; 3.8 Put CHECK() Constraint Near what they Check; 3.8.1 Consider Range Constraints for Numeric Values

3.8.2 Consider LIKE and SIMILAR TO Constraints for Character Values

3.8.3 Remember That Temporal Values Have Duration; 3.8.4 REAL and FLOAT Data Types Should Be Avoided; 3.9 Put Multiple Column Constraints as Near to Both Columns as Possible; 3.10 Put Table-Level CHECK() Constraints at the End of the Table Declaration; 3.11 Use CREATE ASSERTION for Multi-table Constraints; 3.12 Keep CHECK() Constraints Single Purposed; 3.13 Every Table Must Have a Key to Be a Table; 3.13.1 Auto-Numbers Are Not Relational Keys; 3.13.2 Files Are Not Tables; 3.13.3 Look for the Properties of a Good Key

3.14 Do Not Split Attributes

Sommario/riassunto

Are you an SQL programmer that, like many, came to SQL after learning and writing procedural or object-oriented code? Or have switched jobs to where a different brand of SQL is being used, or maybe even been told to learn SQL yourself? If even one answer is yes, then you need this book. A "Manual of Style" for the SQL programmer, this book is a collection of heuristics and rules, tips, and tricks that will help you improve SQL programming style and proficiency, and for formatting and writing portable, readable, maintainable SQL code. Based on many years of experience consulting in SQ
