

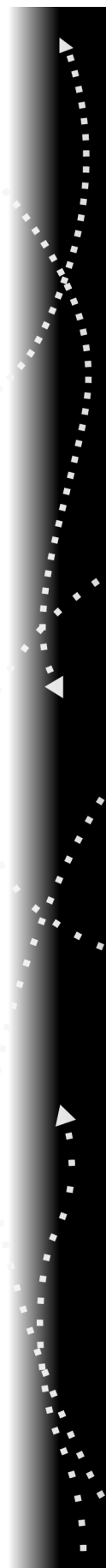
Introduction

Back in the early and mid-1990s, I wrote regular magazine columns in *Database Programming & Design* and later in *DBMS* magazine. The gimmick I used to attract reader responses was to end each column with a SQL programming puzzle. Ten years later, those two magazines were consolidated into *Intelligent Enterprise*. My SQL puzzles moved to some smaller publications and then finally faded away. Today, I throw out a puzzle or two on the www.dbazine.com Web site and other places on the Internet rather than in print media.

Over the years, college students had all kinds of programming contests that used the procedural language du jour—C, Pascal, then Java and C++ today. There is not much for database programmers to test themselves against, except my little puzzle book.

I would often find my puzzles showing up in homework assignments because I was the only source that teachers knew about for SQL problems. I would then get an e-mail from a lazy student wanting me to do his homework for him, unaware of the source of the assignment.

Back in those early days, the de facto standard was SQL-86, and the SQL-92 standard was a design goal for the database vendors. Today, most vendors have gotten most of SQL-92 into their products. The design goal is now the SQL-99 standard's OLAP features.





A decade ago, college students took RDBMS courses, and becoming an SQL programmer required some expertise. SQL products were expensive and the best ones lived on mainframes.

Today, colleges are not teaching RDBMS theory in the undergrad curriculum. SQL is not as exotic as it once was, and you can get cheap or open-source SQL databases. The Internet is full of newsgroups where you can get help for particular products.

The bad news is that the quality of SQL programmers has gotten worse because people who have no foundations in RDBMS or training in SQL are being asked to write SQL inside their native programming languages.

This collection of puzzles includes the original puzzles, so that the original readers can look up their favorites. But now many of them have new solutions, some use the older syntax, and some use the newer features. Many of the original solutions have been cooked by other people over the years. The term “cooked” is a puzzler’s term for finding a better solution than the proposer of the problem presented. The original book contained 50 puzzles; this edition has 75 puzzles.

In the first edition, I tried to organize the puzzles by categories rather than in chronological order or by complexity. This edition, I have given up my informal category scheme because it made no sense. A problem might be solved by a change to the DDL or a query, so should it be categorized as a DDL puzzle or a DML puzzle?

I have tried to credit the people involved with each problem, but if I missed someone, I apologize.

Acknowledgments, Corrections, and Future Editions

I will be glad to receive corrections, new tricks and techniques, and other suggestions for future editions of this book. Send your ideas to or contact me through the publisher, Morgan Kaufmann.

I would like to thank Diane Cerra of Morgan Kaufmann, David Kalman of *DBMS* magazine, Maurice Frank of *DBMS* magazine, David Stodder of *Database Programming & Design*, Phil Chapnick of Miller-Freeman, Frank Sweet of *Boxes & Arrows*, and Dana Farver at www.dbazine.com.

Special thanks to Richard Romley of Smith Barney for cooking so many of my early puzzles, all the people on CompuServe and SQL newsgroups who sent me e-mail all these years, and the people who are posting on the newsgroups today (I used your newsgroup handles, so people can search for your postings). These include, but are not limited to, Raymond D’Anjou, Dieter Noeth, Alexander Kuznetsov, Andrey



Odegov, Steve Kass, Tibor Karaszi, David Portas, Hugo Kornelis, Aaron Bertrand, Itzik Ben-Gan, Tom Moreau, Serge Rielau, Erland Sommarskog, Mikito Harakiri, Adam Machanic, and Daniel A. Morgan.