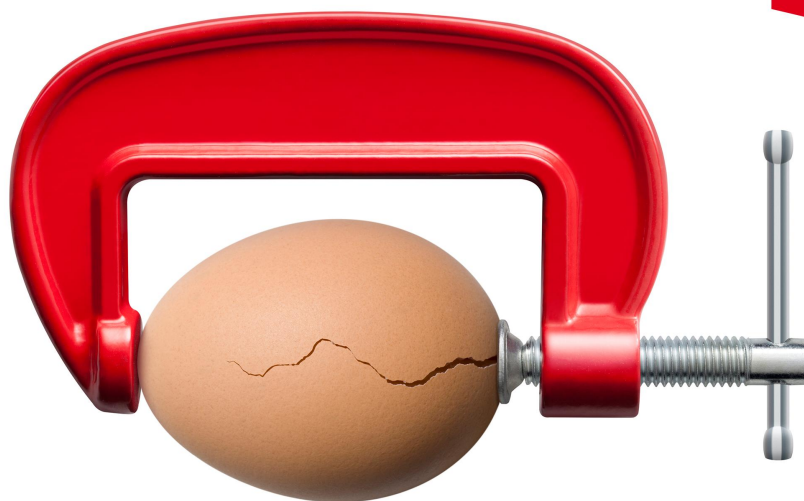


The  
Pragmatic  
Programmers

B  
E  
T  
A

# More SQL Antipatterns



*Bill Karwin*

*edited by  
Jacquelyn Carter*

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

# Contents

|    |                                                                        |   |
|----|------------------------------------------------------------------------|---|
|    | <a href="#">Change History</a>                                         | ? |
|    | <a href="#">Preface</a>                                                | ? |
| 1. | <a href="#">Why SQL?</a>                                               | ? |
|    | <a href="#">SQL Is Ubiquitous</a>                                      | ? |
|    | <a href="#">Does AI Make Learning SQL Obsolete?</a>                    | ? |
|    | <a href="#">What Is an Antipattern?</a>                                | ? |
|    | <a href="#">Example Database</a>                                       | ? |
|    | <a href="#">What's Next</a>                                            | ? |
|    | <b>Part I — SQL Language Antipatterns</b>                              |   |
| 2. | <a href="#">Fear of JOINS</a>                                          | ? |
|    | <a href="#">Objective: Combining Matching Data in Two Tables</a>       | ? |
|    | <a href="#">Antipattern: Fearing JOINS</a>                             | ? |
|    | <a href="#">How to Recognize the Antipattern</a>                       | ? |
|    | <a href="#">Legitimate Uses of the Antipattern</a>                     | ? |
|    | <a href="#">Solution: Preferring JOINS</a>                             | ? |
|    | <a href="#">Mini-Antipattern: Comma-Style Join Syntax</a>              | ? |
|    | <a href="#">Mini-Antipattern: NATURAL JOIN</a>                         | ? |
| 3. | <a href="#">Relational Division</a>                                    | ? |
|    | <a href="#">Objective: Matching a List of Values</a>                   | ? |
|    | <a href="#">Antipattern: Searching Multiple Rows in a WHERE Clause</a> | ? |
|    | <a href="#">How to Recognize the Antipattern</a>                       | ? |

|    |                                                             |   |
|----|-------------------------------------------------------------|---|
|    | Legitimate Uses of the Antipattern                          | ? |
|    | Solution: Using Relational Division                         | ? |
|    | Mini-Antipattern: Aggregation Functions in the WHERE Clause | ? |
| 4. | <b>Dynamic Pivot</b>                                        | ? |
|    | Objective: Turning the Table                                | ? |
|    | Antipattern: Using a Single Query                           | ? |
|    | How to Recognize the Antipattern                            | ? |
|    | Legitimate Uses of the Antipattern                          | ? |
|    | Solution: Using Code to Assist                              | ? |
|    | Mini-Antipattern: Operand Must Contain One Column           | ? |
| 5. | <b>Pagination: Take It from the Top</b>                     | ? |
|    | Objective: Presenting Query Results in Pages                | ? |
|    | Antipattern: Mistaking Offset for Optimization              | ? |
|    | How to Recognize the Antipattern                            | ? |
|    | Legitimate Uses of the Antipattern                          | ? |
|    | Solution: Changing the Rules                                | ? |
|    | Mini-Antipattern: LIMIT without ORDER BY                    | ? |

## Part II — Optimization Antipatterns

|    |                                                      |   |
|----|------------------------------------------------------|---|
| 6. | <b>Non-Sargable Queries</b>                          | ? |
|    | Objective: Optimizing Complex Queries                | ? |
|    | Antipattern: Using Queries That Can't Be Optimized   | ? |
|    | How to Recognize the Antipattern                     | ? |
|    | Legitimate Uses of the Antipattern                   | ? |
|    | Solution: Refactoring the Query                      | ? |
|    | Mini-Antipattern: Redundant Indexes                  | ? |
| 7. | <b>Premature Denormalization</b>                     | ? |
|    | Objective: Breaking the Rules to Improve Performance | ? |
|    | Antipattern: Breaking the Rules Recklessly           | ? |
|    | How to Recognize the Antipattern                     | ? |
|    | Legitimate Uses of the Antipattern                   | ? |

|     |                                                             |   |
|-----|-------------------------------------------------------------|---|
|     | Solution: Breaking the Rules Carefully                      | ? |
|     | Mini-Antipattern: Query Snipers                             | ? |
| 8.  | <b>NoSQL: The Grass Is Always Greener on the Other Side</b> | ? |
|     | Objective: Relief from the Challenges of SQL                | ? |
|     | Antipattern: Switching to NoSQL Is Easy                     | ? |
|     | How to Recognize the Antipattern                            | ? |
|     | Legitimate Uses of the Antipattern                          | ? |
|     | Solution: Match Your Queries to Your NoSQL Models           | ? |
|     | Mini-Antipattern: N+1 Queries                               | ? |
| 9.  | <b>JSON: Matryoshka Dolls</b>                               | ? |
|     | Objective: Unlimited Flexibility                            | ? |
|     | Antipattern: Semi-Structured Data in a Relational Database  | ? |
|     | How to Recognize the Antipattern                            | ? |
|     | Legitimate Uses of the Antipattern                          | ? |
|     | Solution: Be Mindful and Moderate About Flexibility         | ? |
|     | Mini-Antipattern: Quotes for All Occasions                  | ? |
| 10. | <b>Cacheless Transactions</b>                               | ? |
|     | Objective: A Magic Trick to Improve Performance             | ? |
|     | Antipattern: The Performance Vanishing Act                  | ? |
|     | How to Recognize the Antipattern                            | ? |
|     | Legitimate Uses of the Antipattern                          | ? |
|     | Solution: Use Sleight of Hand                               | ? |
|     | Mini-Antipattern: Query Caching                             | ? |

## Part III — Application Design Antipatterns

|     |                                                |   |
|-----|------------------------------------------------|---|
| 11. | <b>Polling: Are We There Yet?</b>              | ? |
|     | Objective: Notify of Changes to Data           | ? |
|     | Antipattern: Polling for Changes               | ? |
|     | How to Recognize the Antipattern               | ? |
|     | Legitimate Uses of the Antipattern             | ? |
|     | Solution: Just Wake Me Up When We Get There    | ? |
|     | Mini-Antipattern: Enqueueing Before Committing | ? |

|                                                         |   |
|---------------------------------------------------------|---|
| <b>12. Transaction Encapsulation</b>                    | ? |
| Objective: Transactions for Multiple Models             | ? |
| Antipattern: Every Model Class Manages Its Transactions | ? |
| How to Recognize the Antipattern                        | ? |
| Legitimate Uses of the Antipattern                      | ? |
| Solution: Simplify, Simplify                            | ? |
| Mini-Antipattern: ORMs that Pluralize Table Names       | ? |
| <b>13. Fear of Deadlocks</b>                            | ? |
| Objective: Don't See Deadlock Errors                    | ? |
| Antipattern: Trying to Prevent All Deadlocks            | ? |
| How to Recognize the Antipattern                        | ? |
| Legitimate Uses of the Antipattern                      | ? |
| Solution: Mitigate Deadlocks                            | ? |
| Mini-Antipattern: Sharing Connections Between Threads   | ? |
| <b>14. Data Hoarding</b>                                | ? |
| Objective: Cope with Data Bloat                         | ? |
| Antipattern: The Belief That You Need All Data          | ? |
| How to Recognize the Antipattern                        | ? |
| Legitimate Uses of the Antipattern                      | ? |
| Solution: An Intervention to Manage Data Retention      | ? |
| <b>15. Epilogue</b>                                     | ? |
| Optimization Is About Priorities                        | ? |
| Expand Your Toolbox                                     | ? |
| Think Holistically                                      | ? |
| Be Prepared for Questions                               | ? |
| Signing Off                                             | ? |
| <b>A1. ACID Transactions</b>                            | ? |
| Atomicity                                               | ? |
| Consistency                                             | ? |
| Isolation                                               | ? |
| Durability                                              | ? |
| ACID Is a Good Thing                                    | ? |

A2. The CAP Theorem . . . . . ?

Bibliography . . . . . ?