

Database in Depth:
Relational Theory for Practitioners

a technical seminar for DBAs, data architects,
DBMS implementers, database application programmers,
and other database professionals

*based on the book of the same name
(O'Reilly Media Inc., 2005)*

by

C. J. Date

ABOUT THIS SEMINAR

Years of experience in working with the database community strongly suggest a need for a seminar that covers relational principles in a way not tainted by the quirks and peculiarities of existing products, commercial practice, or the current version of the SQL standard. This seminar has been designed to meet that need. It's aimed primarily at database practitioners (that is, people working in the database field, perhaps on a daily basis) who feel they don't have as much understanding of the theory underlying their own field as they might. That theory is, of course, the relational model--and while the fundamental ideas of that model are all quite simple, they're widely misrepresented in the trade press and elsewhere; indeed, they're widely misunderstood, and often not understood at all. For example, can you answer the following questions?

1. What exactly is first normal form?
2. What's the connection between relations and predicates?
3. What's semantic optimization?
4. What's a join dependency?
5. Why is semidifference important?
6. Why doesn't deferred integrity checking make sense?
7. What's a relation variable?
8. What's nonloss decomposition?
9. Can a relation have an attribute whose values are relations?
10. What's the difference between SQL and the relational model?
11. Why is *The Information Principle* important?
12. How does XML fit with the relational model?

This seminar provides the answers to these and many related questions. Overall, the intent is to help true database professionals understand relational theory in depth and make use of that understanding in their day-to-day database activities.

TOPIC OUTLINE

1. Preliminaries

- SQL ≠ the relational model!
- Terminology: principles not products
- The original model: overview
- Model vs. implementation
- Properties of relations
- Relations vs. relvars
- Values vs. variables

2. Relations vs. types

- Domains are types ...
- ... of arbitrary complexity
- Types have operators
- System- vs. user-defined types
- Scalar vs. nonscalar types
- Generated types

3. Tuples and relations

- What's a tuple?
- Important consequences
- What's a relation?
- Further consequences
- Why duplicates are prohibited
- Why nulls are prohibited
- TABLE_DUM and TABLE_DEE

4. Relation variables

- Updating is set-at-a-time
- Candidate and foreign keys
- Views and view processing
- *The Principle of Interchangeability*
- Predicates and propositions
- More on relations vs. types

5. Relational algebra

- Importance of closure
- Familiar operators (JOIN etc.)
- Which operators are primitive?
- Evaluating SQL expressions
- RENAME, EXTEND, SUMMARIZE
- SEMIJOIN, SEMIMINUS
- GROUP, UNGROUP
- Optimization
- Relational comparisons
- Assignment revisited
- What about ORDER BY?

6. Integrity constraints

- Type and database constraints
- The role of transactions
- When are constraints checked?
- Multiple assignment
- Constraints and predicates
- **The Golden Rule**
- Correctness vs. consistency

7. Database design

- The place of design theory
- BCNF and 5NF; join dependencies
- A useful normalization theorem
- Why normalization is not a panacea
- But don't denormalize!
- *The Principle of Orthogonal Design*
- Orthogonality and normalization
- Remarks on physical design

8. What *is* the relational model?

- Five components and their significance
- *The Information Principle*
- Relational systems and others
- Where next?
- Suggestions for further reading

DURATION

One day (approximately 6 lecture-hours).

WHO SHOULD ATTEND

- DBMS designers, implementers, and other vendor personnel
- Database consultants
- Data and database administrators
- Information modelers and database designers
- Database application designers and implementers
- Computer science professors specializing in database matters
- People responsible for DBMS product evaluation and acquisition

The seminar is *not* meant for beginners: Attendees will be expected to have at least an elementary familiarity with database concepts in general and the SQL language in particular.

OBJECTIVES

On completion of this seminar, attendees will:

- Have a solid appreciation of the importance of sound principles and foundations
- Understand why and how the relational model is still directly relevant to modern database technology and will remain so for as far as anyone can see
- See why and how the SQL standard is seriously deficient
- Be able to use the best current theoretical knowledge in the design of their databases and database applications
- Be able to make informed decisions in their daily database professional activities

DOCUMENTATION

Attendees will receive a workbook containing copies of the speaker's slides, together with a selection of technical papers and other background reading material.

SPEAKER: Chris Date

C. J. Date is an independent author, lecturer, researcher, and consultant, specializing in relational database technology. He is best known for his book *An Introduction to Database Systems* (eighth edition, Addison-Wesley, 2004), which has sold some 725,000 copies and is used by several hundred colleges and universities worldwide. He is also the author of many other books on database management, including most recently:

- From Morgan Kaufmann: *Temporal Data and the Relational Model* (coauthored with Hugh Darwen and Nikos A. Lorentzos, 2003)
- From O'Reilly: *Database in Depth: Relational Theory for Practitioners* (2005)
- From Addison-Wesley: *Databases, Types, and the Relational Model: The Third Manifesto* (coauthored with Hugh Darwen, to appear 2006)

Another book, *Go Faster! The TransRelationaltm Approach to DBMS Implementation*, is also due for publication in the near future.

Mr. Date enjoys a reputation that is second to none for his ability to communicate complex technical subjects in a clear and understandable fashion.

----- ◆◆◆◆◆ -----