Part 21

Criteria for Relational Databases
Aspects of Implementation

Logical Database Design

Relational Database Management System Design

Index Creation and Tuning
Objectives of the Relational Approach

(Codd & Date, 1974)

1. Provide a high degree of data independence
2. Provide a community view of the data of Spartan simplicity, so that a wide variety of users in an enterprise (ranging from the most computer-naive to the most computer-sophisticated) can interact with a common model (while not prohibiting superimposed user views for specialized purposes)
3. Simplify the potentially formidable job of the database administrator
4. Introduce a theoretical foundation (albeit modest) into database management so as to reduce the likelihood of surprise
5. Merge the fact retrieval and file management fields in preparation for the addition at a later time of inferential services in the commercial world
6. Lift database application programming to a new level—a level in which sets (and more specifically relations) are treated as operands instead of being processed element by element
Levels of Relational

NOT RELATIONAL

1. Mistakenly relational
   • “Supports relational operators such as <, >, =, etc.”
2. Born-again relational
   • a few relational features added - basically non-relational in concept, implementation, and usage
3. Tabular
   • supports tables without user-visible links
   • programmer must do the navigation

RELATIONAL

3 Major Parts

Structural  Manipulative  Integrity

4. Minimally relational
   • full structural, minimal relational processing capability
5. Relationally complete
   • full structural, relationally complete processing
6. Fully relational
   • full structural, full manipulative, full integrity
Classification of DBMS's

Legend:
S - Structural
M - Manipulative
I - Integrity

c = Relational completeness
m = Minimal relational processing capability

Tabular
(Previously called semi-relational)

Minimally Relational

Relationally Complete

Fully Relational
Criteria for Fully Relational

(Codd) For any system that is advertised as, or claimed to be, a relational database management system, that system must be able to manage databases entirely through its relational capabilities

- must support database insert, update, and delete *multiple records at-a-time* (relation-at-a-time)
Codd’s 12 Rules

1. The Information Rule
2. Guaranteed Access Rule
3. Systematic Treatment of Null Values
4. Dynamic On-line Catalog Based on the Relational Model
5. Comprehensive Data Sub-language Rule
6. View Updating Rule
7. High-level Insert, Update, and Delete
8. Physical Data Independence
9. Logical Data Independence
10. Integrity Independence
11. Distribution Independence
12. Non-subversion Rule
Codd’s Rules 1 and 2

Codd Rule 1: The Information Rule
All information in a relational database is represented explicitly at the logical level and in exactly one way - by values in tables
• including table names, column names, domain names
• including database administration data

Codd Rule 2: Guaranteed Access Rule
Each and every datum (atomic value) in a relational database is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name
• must have a primary key
• associative addressing scheme
Codd’s Rules 3 and 4

Codd Rule 3: Systematic Treatment of Null Values

Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational database management systems for representing missing information and inapplicable information in a systematic way, independent of data type

- “nulls not allowed” must be available
- not just a “special value”, type independent
- need to support 3-valued logic

Codd Rule 4: Dynamic On-line Catalog Based on the Relational Model

The database description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data

- need only learn one model
- can patch vendor’s design oversights
Codd’s Rules 5 and 6

Codd Rule 5: Comprehensive Data Sub-language Rule
A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all of the following items:
• data definition
• view definition
• data manipulation
• integrity constraints
• authorization
• transaction boundaries
(should not need several languages)

Codd Rule 6: View Updating Rule
All views that are theoretically updateable are also updateable by the system
• insertion, deletion, modification
• not required to invert arithmetic or functions
Codd’s Rules 7 and 8

Codd Rule 7: High-level Insert, Update, and Delete

The capability of handling a base relation as a single operand applies not only to the retrieval of data but also to the insertion, update, and deletion of data

- permits performance enhancements
- non-navigational
- system free to cluster distributed transactions by site

Codd Rule 8: Physical Data Independence

Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods

- boundaries between logical and semantic aspects
- boundaries between physical and performance aspects
Codd’s Rules 9 and 10

Codd Rule 9: Logical Data Independence
Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables
Ex: split table in 2 by row selection
Ex: split table in 2 by columns, repeating key
Ex: combine 2 tables by “lossless” join
• can “make mistakes” in design and fix it

Codd Rule 10: Integrity Independence
Integrity constraints specific to a particular relational database must be definable in the relational data sub-language and storable in the catalog, not in the application programs
• entity integrity (no component of a primary key is allowed to be null)
• referential integrity (non-null foreign key values must have matching primary key values)
• additional integrity
• must be part of the database management system engine
Codd’s Rules 11 and 12

Codd Rule 11: Distribution Independence
A relational database management system has distribution independence.
DBMS has a data sub-language with logically unimpaired application programs and terminal activity under
- the first occurrence of data distribution
- data redistribution

Codd Rule 12: Non-subversion Rule
If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher-level relational language (multiple-records-at-a-time)
- no bypass of integrity