

Codd's Relational Rules

In 1985, Dr. E. F. Codd, the acknowledged “father” of the relational data model, presented the following 12 rules that a database must obey if it is to be considered truly relational.

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.

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.

Rule 3: Systematic treatment of null values Null values (distinct from an empty character string or a string of blank characters and distinct from zero or any other number) are supported for representing missing information and inapplicable information in a systematic way, independent of the data type.

Rule 4: Dynamic online 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.

Rule 5: Comprehensive data sublanguage 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 definitions
- View definitions
- Data manipulation (interactive and by program)
- Integrity constraints
- Authorization
- Transaction boundaries (begin, commit, and rollback)

Rule 6: View updating rule All views that are theoretically updateable are also updateable by the system.

Rule 7: High-level insert, update, and delete The capability of handling a base relation or a derived relation (that is, a view) as a single operand applies not only to the retrieval of data but also to the insertion, update, and deletion of data.

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.

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.

Rule 10: Integrity independence Integrity constraints specific to a particular relational database must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.

Rule 11: Distribution independence The data manipulation sublanguage of a relational DBMS must enable application programs and inquiries to remain logically the same whether and whenever data are physically centralized or distributed.

Rule 12: Nonsubversion 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).