Codd's 12 Rules for a Truly Relational System

Rule Zero

- 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.

Rule 1: Information Representation

- 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

- 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 the empty character string or a blank characters and distinct from zero or any number) are supported for representing missing information and inapplicable information in a systematic way, independent of data type.

Rule 4: Dynamic On-Line Catalog Based on Relation Model

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

Rule 5: Comprehensive Data Sublanguage

- A relational system may support several languages and various modes of terminal use. However, there must be at least one language whose statements can express all of the following items: (1) data definition, (2) view definitions, (3) data manipulation (interactive and by program), (4) integrity constraints, (5) authorization, and (6) transaction boundaries (begin, commit, and rollback).

Rule 6: View Updating

- All views that are theoretically updatable are also updatable by the system.

Rule 7: High-Level Insert, Update, and Delete

- The capability of handling a base relation or a derived relation (that is, 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 table.

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 program.

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

- 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-record-at-a-time).

Summary

- *Rules 1 and 4* allow a database administrator to always know exactly what kinds of data are recorded. Hence, they minimize the time needed to determine the data available while also reducing the data redundancy that would result if this information were unknowable.

Summary

- *Rule 3* helps users of all types to avoid making foolish and costly mistakes (for example, miscalculating summary data when null values are coded in some uniterpretable way).

Summary

- *Rule 5* supports interactive program testing, which can mean improved programmer productivity.

Summary

- *Rules 8-11* contribute to lower program development and maintenance costs due to the inevitable system changes that occur in decision support and information retrieval applications systems.