

Dr Edgar F. Codd, after his extensive research on the Relational Model of database systems, came up with **twelve rules** of his own, which according to him, a database must obey in order to be regarded as a true relational database.

Rule 1: Information Rule

The data stored in a database, may it be **user data or metadata**, must be a value of some table cell. **Everything** in a database must be stored **in a table format**.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of **table-name**, **primary-key** (row value), and **attribute-name** (column value). **No other means, such as pointers**, can be used to access data.

Rule 3: Systematic Treatment of NULL Values

The **NULL values** in a database must be given a systematic and **uniform treatment**. This is a very important rule because a NULL can be interpreted as one of the following – data is **missing**, data is **not known**, or data is **not applicable**.

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an **online catalog**, known as **data dictionary**, which can be accessed by authorized users. Users can use the **same query language to access** the catalog which they use to access the database itself.

Rule 5: Comprehensive Data Sub-Language Rule

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. (**create table**, alter table, drop table)
- View definition. (**create view**)
- Data manipulation (**insert**, delete, update).
- Integrity constraints. (**primary key**, foreign key, check)
- Authorization. (**grant**, revoke)
- Transaction boundaries (**commit**, rollback, savepoint).

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, **must also be updatable** by the system.

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

A database must support **high-level insertion, update, and deletion**. This must not be limited to a single row, that is, it must also support union, intersection, and minus operations to yield **sets of data records**.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. **Any change in the physical structure** of a database must not have any impact on how the data is being accessed by external applications.

Rule 9: Logical Data Independence

The logical data in a database must be independent of its user's view (application). **Any change in logical data** must not affect the applications using it. For example, if **two tables are merged** or one is **split into two** different tables, there should be no impact or change on the user application.

Rule 10: Integrity Independence

Integrity constraints specific to a particular relational data base must be definable in the relational data sublanguage and storable in the catalog, **not in the application programs**.

Rule 11: Distribution Independence

The **end-user must not be able to see that the data is distributed** over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the **foundation of distributed database systems**.

Rule 12: Non-Subversion Rule

If a system has an interface that provides access to low-level records, then the interface **must not be able to** subvert the system and **bypass** security and integrity **constraints**.