The Codd's Rules

- Jayendra Khatod

Overview of CODD's 12 rules

- **Codd's twelve rules are a set of thirteen rules (numbered zero to twelve) proposed by Dr.Edgar F. Codd, a pioneer of the relational model for databases, designed to define what is required from a database management system in order for it to be considered *relational*,**

  i.e. a relational database management system (RDBMS).

Rule (0)

- **The system must qualify as *relational*, as a *database*, and as a *management system*.**

- **For a system to qualify as a relational database management system (RDBMS), that system must use its *relational* facilities (exclusively) to *manage* the *database*.**

Rule 1 : The information Rule

“All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in column positions within rows of tables."

Rule 1 (Continued..)

- **All information including table names, column names and column data types should be available in some table with in the database.**
- **Tables that hold such information constitute the data dictionary.**
- **Data are represented only one way: as values within columns and rows.**
- **Simple, consistent and versatile.**
- **The basic requirement of the relational model.**

Rule 2 : Guaranteed access Rule

“Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name."

Rule 2 (Continued..)

- **Every value can be accessed by providing table name, column name and key.**
- **All data are uniquely identified and accessible via this identity.**
- **Most RDBMS do not make the definition of the primary key mandatory and are deficient to that extent**

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 DBMS for representing missing information and inapplicable information in a systematic way, independent of data type."

Rule 3 (Continued..)

- **Separate handling of missing and/or non applicable data.**
- **This is distinct to zero or empty strings**

Rule 4 : Dynamic on-line catalog based on the relational model

The data base 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 4 (Continued..)

- **Catalog (data dictionary) can be queried by authorized users as part of the database.**
- **The catalog is part of the database.**

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 the following items

Data Definition

View Definition

Data Manipulation (Interactive and by program).

Integrity Constraints

Rule 5 (Continued..)

- **Every RDBMS should provide a language to allow the user to query the contents of the RDBMS and also manipulate the contents of the RDBMS.**

Rule 6 : View updating Rule

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

All theoretically possible view updates should be possible.

RDBMS should allow the update of simple theoretically updatable views, but disallow attempts to update complex views.

Rule 7 : High-level insert, update and delete

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

Rule 7 (Continued..)

***High-level insert, update, and delete*:**

- **The system must support set-at-a-time *insert*, *update*, and *delete* operators.**
- **This means that data can be retrieved from a relational database in sets constructed of data from multiple rows and/or multiple tables.**
- **This rule states that insert, update, and delete operations should be supported for any retrievable set rather than just for a single row in a single table.**

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 8 (Continued..)

Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure.

Rule 9 : Logical data independence.

“Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit un-impairment are made to the base tables."

Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure. Logical data independence is more difficult to achieve than physical data independence.

Rule 10 : Integrity independence.

“Integrity constraints specific to a particular relational data base must be definable in the relational data sub-language and storable in the catalog, not in the application programs."

Rule 10 (Continued..)

- **Integrity constraints are to be stored in the catalog not the programs.**
- **Alterations to integrity constraints should not affect application programs.**
- **This simplifies the programs.**
- **It is not always possible to do this.**

Rule 11 : Distribution independence.

The data manipulation sub language of an RDBMS must enable application programs and queries to remain logically unchanged whether & whenever data is physically centralized or distributed.

Distribution independence:

Applications should still work in a distributed database (DDB).

This means an Application program that accesses the DBMS on a single computer should also work, without modification, even if the data is moved from one computer to another in a network environment.

Rule 12: The *non-subversion rule*:

If the system provides a low-level (record-at-a-time) interface, then that interface cannot be used to subvert the system, for example, bypassing a relational security or integrity constraint.

Rule 12 (Continued..)

All the database access must be controlled through the DBMS so that the integrity of the database cannot be compromised without the knowledge of the user.

Summary

In this lesson should have learned the following:

- **Rule (0): The system must qualify as *relational*, as a *database*, and as a *management system*.**
- **Rule 1: The *information rule***
- **Rule 2: The *guaranteed access rule***
- **Rule 3: *Systematic treatment of null values***
- **Rule 4: *Active online catalog based on the relational model***
- **Rule 5: The *comprehensive data sublanguage rule***

Summary (continued..)

- **Rule 6: The *view updating rule***
- **Rule 7: *High-level insert, update, and delete***
- **Rule 8: *Physical data independence***
- **Rule 9: *Logical data independence***
- **Rule 10: *Integrity independence***
- **Rule 11: *Distribution independence***
- **Rule 12: The *nonsubversion rule***

Thank You !