Unit:1

Codd’s rules

RDBMS

- Relational Database Management System
- For defining a fully relational database
  - Dr. E. F. Codd's 12 rules is used.

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

Dr. E.F. Codd's 12 Rule

- Rule 0 : Foundation Rule
  - For a system to qualify as a relational database management system (RDBMS), that system must use its *relational* facilities (Relation between table) to *manage* the *database*.
- Rule 1: The Information Rule
  - All data should be presented to the user in table form.
- Rule 2: Guaranteed Access Rule
  - All data should be accessible without ambiguity.
  - This can be achieved through a combination of the table name, primary key, and column name.

- Rule 3: Systematic Treatment of Null Values:
  - A field should be allowed to remain empty.
  - This involves the support of a null value, which is distinct from an empty string or a number with a value of zero.
  - Of course, this can't apply to primary keys.
- Rule 4: Active online catalog based on the relational model:
  - The system must support an online, inline, relational catalog that is accessible to authorized users by means of their regular query language.
  - Users must be able to access the database's structure (catalog) using the same query language that they use to access the database's data.

- Rule 5: Comprehensive Data Sublanguage
  - The database must support at least one clearly defined language that includes functionality for data definition, data manipulation, data integrity, and database transaction control.
  - All commercial relational databases use forms of the standard SQL (Structured Query Language) as their supported comprehensive language.
  - Supported Language :
    - Data definition
    - View definition
    - Data manipulation (interactive and by program)
    - Integrity constraints
    - Authorization
    - Transaction boundaries (begin, commit, and rollback).

- Rule 6: View Updating Rule
  - View : Data can be presented to the user in different logical combinations, called views.
  - Each view should support the same full range of data manipulation that direct-access to a table has available.
- Rule 7: 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.

- Rule 8: Physical Data Independence
  - 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
  - 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 must be specified separately from application programs and stored in the structure/catalog.
  - No component of a primary key can have a null value. (see rule 3) If a foreign key is defined in one table, any value in it must exist as a primary key in another table.
  - Key and Check constraints, trigger etc should be stored in Data Dictionary.

- Rule 11: Distribution Independence
  - A user should be totally unaware of whether or not the database is distributed (whether parts of the database exist in multiple locations).
  - A variety of reasons make this rule difficult to implement;
- Rule 12: Non subversion Rule
  - *If a relational system has or supports a low-level (single-record-at-a-time) language, that low-level language cannot be used to subvert or bypass the integrity rules or constraints expressed in the higher-level (multiple-records-at-a-time) relational language.*