EF Codd’s rules in DBMS

- ***EF Codd*** is a computer scientist who first ***outlined the relational model*** which now became the most popular and one and only database model
- Codd Proposed `13 rules` (listed from 0 to 12) popularly known as ***codd’s 12 rules*** which are used as a yardstick to test the quality of relational database management system(RDBMS)
- Till now none of the commercial product has followed all the 13 rules ***even Oracle has followed 8.5 out of 13***

Rule 0: Foundation rule

- If a system is said to be an RDBMS then the ***database should be managed using only relational capabilities***

Rule 1: Information rule

- All the ***data*** including metadata ***must be stored in some cell of the table*** in the form of rows and columns .

Rule 2: Guaranteed access

- Each ***data element*** in a table ***must be accessed through a combination of table name + primary key (row)+ attribute(column)***
- Example : emp+empid+ename,sal
- Strictly the ***data must not be accessed via a pointer .***

Rule 3: Systematic treatment of null values

- ***Null values*** represent different situations it may be ***missing data or not applicable or no value situation.***
- Null values ***must be handled consistently*** and also ***primary key must not be null*** and any expression on null must give null.

Rule 4: Active online catalog

- Database dictionary is a catalog which ***shows structural description of the complete database*** and it must be stored online
- This rule states that a database dictionary ***must be governed by the same rules and same query language as used for general database***.

Rule 5: Powerful and well-structured language

- The database should be accessible through a language **which supports definition, manipulation and all transaction management activities**, such a language is called structured language
- For example, SQL, ***if database uses a different language for data access and manipulation then it is a violation of the rule.***

Rule 6: View updation rule

- Difference ***views*** created for different purposes ***should be automatically updated*** by the system itself.

Rule 7: Relational level operation

- Operations like ***insert, delete and update operations must be supported*** at each level of relation even though it might be a ***nested relation or a complex relation***
- ***Set operations*** like union, intersection, minus ***must be supported.***

Rule 8: Physical data independence

- Any ***change in the physical location*** of the table ***should not reflect the change at the application level***
- Example: If you rename or move a file from one disk to another then it should not affect the application.

Rule 9: Logical data independence

- If there are any ***changes done to the logical structure*** of the database table, then ***users view of data should not be changed***
- If the ***table is split into two tables,*** then a ***new view should give result as the join of these two tables*** but this rule is very difficult to satisfy.

Rule 10: Integrity independence

- Database table should ***design itself on integrity*** rather than using external programs.
- It should use ***primary keys, check constants triggers***, etc which makes our DBMS independent of the front end application.

Rule 11: Distribution Independence

- Data distribution over various geographical locations over a network should not reflect the end-user i.e ***you should feel that all the data is stored in a single place***
- This rule ***laid the foundation for the distributed database.***

Rule 12: Non-Subversion Rule

- Any access given to the ***data that is present in the lowest level must not give a chance to authenticate constraints and change data*** ,this can be achieved through some kind of encryption techniques.