**Codd's Rule**

E.F Codd was a Computer Scientist who invented **Relational model** for Database management. Based on relational model, **Relation database** was created. Codd proposed 13 rules popularly known as **Codd's 12 rules** to test DBMS's concept against his relational model. Codd's rule actualy define what quality a DBMS requires in order to become a Relational Database Management System(RDBMS). Till now, there is hardly any commercial product that follows all the 13 Codd's rules. Even **Oracle** follows only eight and half out(8.5) of 13. The Codd's 12 rules are as follows.

**Rule zero**

This rule states that for a system to qualify as an **RDBMS**, it must be able to manage database entirely through the relational capabilities.

**Rule 1 : Information rule**

All information(including metadata) is to be represented as stored data in cells of tables. The rows and columns have to be strictly unordered.

**Rule 2 : Guaranted Access**

Each unique piece of data(atomic value) should be accesible by : **Table Name + primary key(Row) + Attribute(column)**.

**NOTE :**Ability to directly access via POINTER is a violation of this rule.

**Rule 3 : Systemetic treatment of NULL**

**Null** has several meanings, it can mean missing data, not applicable or no value. It should be handled consistently. Primary key must not be null. Expression on **NULL** must give null.

**Rule 4 : Active Online Catalog**

Database dictionary(catalog) must have description of **Database**. Catalog to be governed by same rule as rest of the database. The same query language to be used on catalog as on application database.

**Rule 5 : Powerful language**

One well defined language must be there to provide all manners of access to data. Example: **SQL**. If a file supporting table can be accessed by any manner except SQL interface, then its a violation to this rule.

**Rule 6 : View Updation rule**

All view that are theoretically updatable should be updatable by the system.

**Rule 7 : Relational Level Operation**

There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

**Rule 8 : Physical Data Independence**

The physical storage of data should not matter to the system. If say, some file supporting table were renamed or moved from one disk to another, it should not effect the application.

**Rule 9 : Logical Data Independence**

If there is change in the logical structure(table structures) of the database the user view of data should not change. Say, if a table is split into two tables, a new view should give result as the join of the two tables. This rule is most difficult to satisfy.

**Rule 10 : Integrity Independence**

The database should be able to conforce its own integrity rather than using other programs. Key and Check constraints, trigger etc should be stored in Data Dictionary. This also make **RDBMS** independent of front-end.

**Rule 11 : Distribution Independence**

A database should work properly regardless of its distribution across a network. This lays foundation of distributed database.

**Rule 12 : Nonsubversion rule**

If low level access is allowed to a system it should not be able to subvert or bypass integrity rule to change data. This can be achieved by some sort of looking or encryption.