

# Codd's Rules

---

Rajeev Srivastava  
rajeevs@cdac.in

# ToC

---

- Codd's Rules
- Data Independence
- ACID Properties
  - Atomicity
  - Consistency
  - Isolation
  - Durability

# Codd's Rules for RDBMS

---

- Defined by and named after Scientist Edgar Frank Codd (Father of RDBMS)
- A set of 13 rules (numbered from 0 to 12)
- Defines characteristics that a DBMS is required to qualify as an RDBMS

**Rule 0 : The foundation Rule :** A system to qualify as a relational database management system, the system must use its relational facilities (exclusively) to manage the database.

**Rule 1: Information Rule :** All information in a database must be represented as data stored in cell. Rows and Columns have to be strictly unordered.

**Rule 2: Guaranteed Access :** All data must be accessible. Each Unique Piece of data must be accessible through a combination of Table Name, Attribute and Primary Key.

# Codd's Rules...

---

**Rule 3: Systematic Treatment of NULL values** : Database must have a consistent way to handle null values. A NULL value is a missing, unknown or Not Applicable value. It must not be represented by empty string or zero.

**Rule 4: Active/Dynamic Online Catalogue** : Database catalogue (Data Dictionary) must be stored in form of tables, and must be accessible using the same query language which is used to access data tables.

**Rule 5: Advanced Language Support (Comprehensive Data Sublanguage Rule)** : A single well defined language to provide all manner access to data. The language must have define, manipulate and access capabilities.

**Rule 6: View Updation Rule** : All views that are theoretically updatable should be updatable.

# Codd's Rules...

---

**Rule 7: Relational Level Operations (High Level Insert, Update and Delete) :**

Insert/Update/Delete operations must be capable of affecting multiple rows from multiple tables (Base Table & Derived Table). Set operations like Union, Intersection and Minus should be supported.

**Rule 8: Physical Data Independence :** Changes in Physical Storage should not affect the logical design of the database. Moving data from one disk to another or change in underlying data structure (Arrays, Linked List etc.) should not matter to the application.

**Rule 9: Logical Data Independence :** Changes in logical database structure (table, column, row etc.) should be hidden from application.

# Data Independence

---

**Physical Data Independence** : Modifications in the physical schema should not affect implementation of Logical Schema.

**Logical Data Independence** : Modification in the logical schema should not affect implementation at View/ Application Level

View/External Level

- Logical Data Independence

Logical/Conceptual Level

- Physical Data Independence

Physical/Internal Level

# Codd's Rules...

---

**Rule 10: Integrity Independence** : Database should be able to define and implement its own integrity rules to filter correct/ intended data and must not depend on other sources. (Implemented through Primary Key and Foreign Key).

**Rule 11: Distribution Independence** : The distribution of portions of the database to various locations should be invisible to users - Existing applications should continue to operate successfully in case of distribution or redistribution of database.

**Rule 12: Non-subversion Rule** : If a low-level access interface is allowed, then it cannot be used to subvert (bypass) system rules (ex. integrity constraint) to change data.

# ACID Properties of a Transaction

---

Set of properties of that defines that every transaction is reliable.

**Atomicity** : Atomicity defines that each transaction be "all or nothing". i.e. . if one part of the transaction fails, the entire transaction fails.

**Consistency** : The consistency property ensures that any transaction will bring the database from one valid state to another. Valid states of database are defined by integrity constraints/triggers etc.

**Isolation** : Isolation property ensures that other operations cannot access or see the data in an intermediate state during a transaction.

**Durability** : Committed data, even in the event of a failure and system restart, is available in its correct state. In other words, all committed changes must be permanent.