CODD's 12 RULES OF RELATIONAL DATABASE

OVERVIEW OF CODD’s RULE

- A **relational database management system (RDBMS)** is a database management system (DBMS) that is based on the relational model as introduced by *E. F. Codd*.
- A short definition of an RDBMS may be a DBMS in which data is stored ***in the form of tables and the relationship among the data is also stored in the form of tables.***
- ***E.F. Codd*,** the famous mathematician has introduced 12 rules (0-12)for the relational model for databases commonly known as **Codd's rules**. The rules mainly define what is required for a DBMS for it to be considered *relational*, i.e., an RDBMS.

Rule 0

- This rule states that all subsequent rules are based on the notation that in order for a database to be considered relational, it must use it’s relational facilities exclusively to manage the database.

Rule 1: INFORMATION RULE

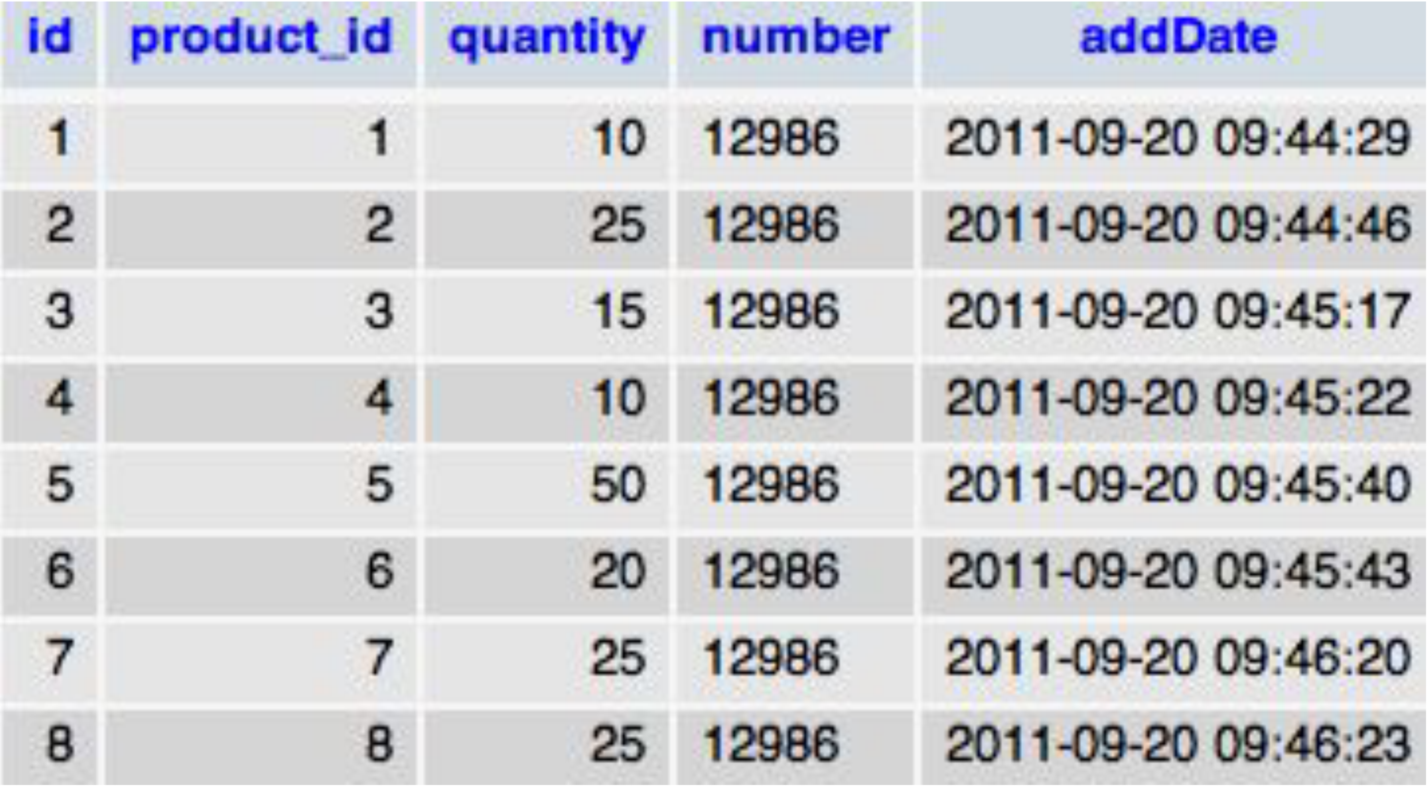
All information in the database is to be represented in one and only one way.

All information in an RDB is **represented as values in the tables.**

This is achieved by values in column and rows of tables.

All information including table names, column names and column data types should be available in same table within the database.

The basic requirement of the relational model.

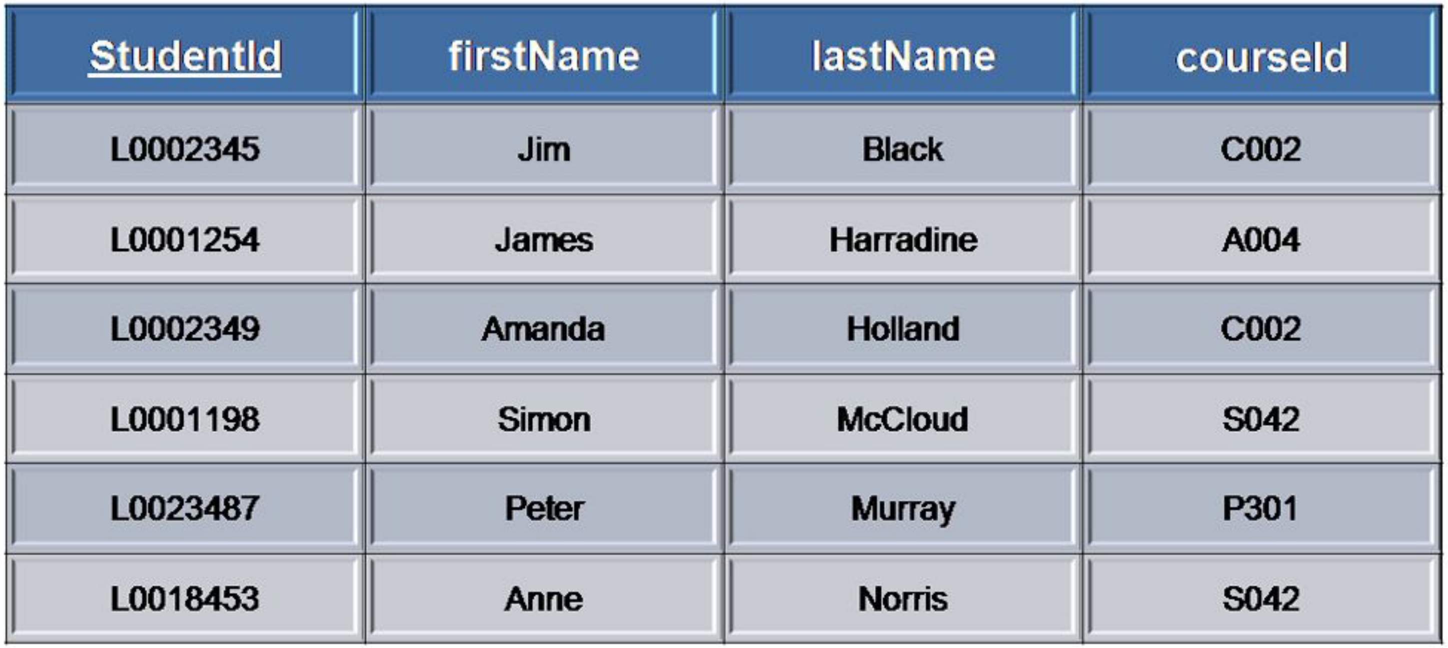


id	product_id	quantity	number	addDate
1	1	10	12986	2011-09-20 09:44:29
2	2	25	12986	2011-09-20 09:44:46
3	3	15	12986	2011-09-20 09:45:17
4	4	10	12986	2011-09-20 09:45:22
5	5	50	12986	2011-09-20 09:45:40
6	6	20	12986	2011-09-20 09:45:43
7	7	25	12986	2011-09-20 09:46:20
8	8	25	12986	2011-09-20 09:46:23

Rule 2: GUARANTEED ACCESS RULE

- Each unique piece of data should be accessible by:table name+primary key(row) + attribute(column).
- 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 .

Primary Keys

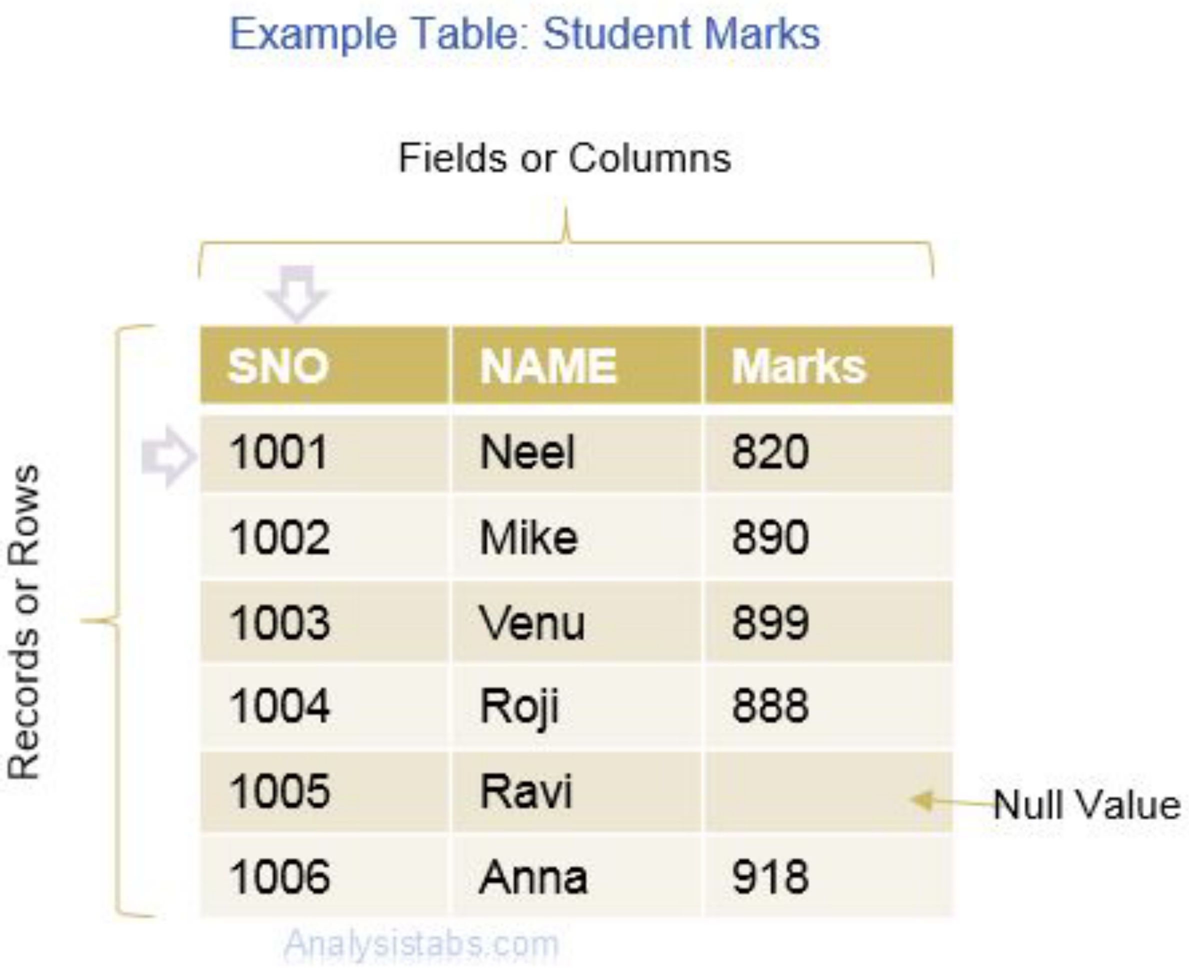


StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

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."

- NULLs may mean: Missing data, Not applicable
- Should be handled consistently - Not Zero or Blank
- Primary keys — Not NULL
- Expressions on NULL should give NULL.
- Separate handling of missing and/or non applicable data.
- This is distinct to zero or empty strings



Example Table: Student Marks

SNO	NAME	Marks
1001	Neel	820
1002	Mike	890
1003	Venu	899
1004	Roji	888
1005	Ravi	
1006	Anna	918

Fields or Columns
Records or Rows
Null Value

Analysistabs.com

Rule 4:DATABASE DESCRIPTION RULE

- 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.
- The authorized users can access the database structure by using common language i.e. SQL

Rule 5: COMPREHENSIVE DATA SUBLANGUAGE

- A relational system may support several languages and various modes of terminal use .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 (create,insert,update)
- View Definition
- Data Manipulation (alter,delete,truncate)
- Integrity Constraints (primary key,foreign key,null values)
- Authorization (GRANT , REVOKE)
- Transaction boundaries (begin,commit,rollbacketc)

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

- View = ”Virtual table”, temporarily derived from base tables.
- Example: If a view is formed as join of 3 tables, changes to view should be reflected in base tables.
- Not updatable: View does not have NOT-NULL attribute of base table.
- It allow the update of simple theoretically updatable views, but disallow attempts to update complex views.

BELOW IS A TABLE NAMED ‘STUDENT‘. RDBMS GIVES US THE FACILITY TO VIEW ONLY SOME PARTICULAR FIELDS ACCORDING TO OUR NEED WHICH ARE DIRECTLY ACCESSED FROM BASE TABLES WHEN REQUIRED.

name	class	Marks	PUPIN NUMBER
SONALI	BCA-2	95	17231
TAMANNA	BCA-2	90	17236
RAJWINDER	BCA-2	90	17267
SAKSHI	BCA-2	86	17893
SADHANA	BCA-2	82	17453

TO VIEW ONLY THE NAME AND MARKS OF THE STUDENT TABLE WE CAN WRITE THE FOLLOWING SYNTAX:

```
CREATE VIEW RECORD
AS SELECT NAME,MARKS FROM STUDENT;
```

VIEW IS CREATED.

```
SELECT * FROM RECORD;
```

name	Marks
SONALI	95
TAMANNA	90
RAJWINDER	90
SAKSHI	86
SADHANA	82

RULE 7

- HIGH-LEVEL INSERT,UPDATE AND DELETE

RULE 7 : HIGH-LEVEL INSERT , UPDATE AND DELETE

- 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.
- It also perform the operation on multiple row simultaneously .
- There must be delete, updating and insertion at the each level of operation. Set operation like union, all union , insertion and minus should also supported.
- **EXAMPLE:**

Suppose if we need to change ID then it will reflect everywhere automatically.

EXAMPLE

TABLE CREATED

NAME	ROLL_NO	CLASS

- **Create table:**

```
SQL>CREATE TABLE STUDENT_DATA
{
NAME VARCHAR 2(20),
ROLL_NO VARCHAR 2(10
),
CLASS VARCHAR 2(20);
};
INSERT ION:
SQL>INSERT  INTO STUDENT_DATA(‘&NAME’,&ROLL_NO,’&CLASS’);
SQL>ENTER VALUE FOR NAME:KIRAN
 SQL>ENTER VALUE FOR ROLL_NO:4556
SQL>ENTER VALUE FOR CLASS:BCA
SQL>/
```

NAME	ROLL_NO	CLASS
KIRAN	4566	BCA
RAHUL	3455	BCA

RULE 8

- LOGICAL DATA INDEPENDENCE

RULE 8 :LOGICAL DATA INDEPENDENCE RULE

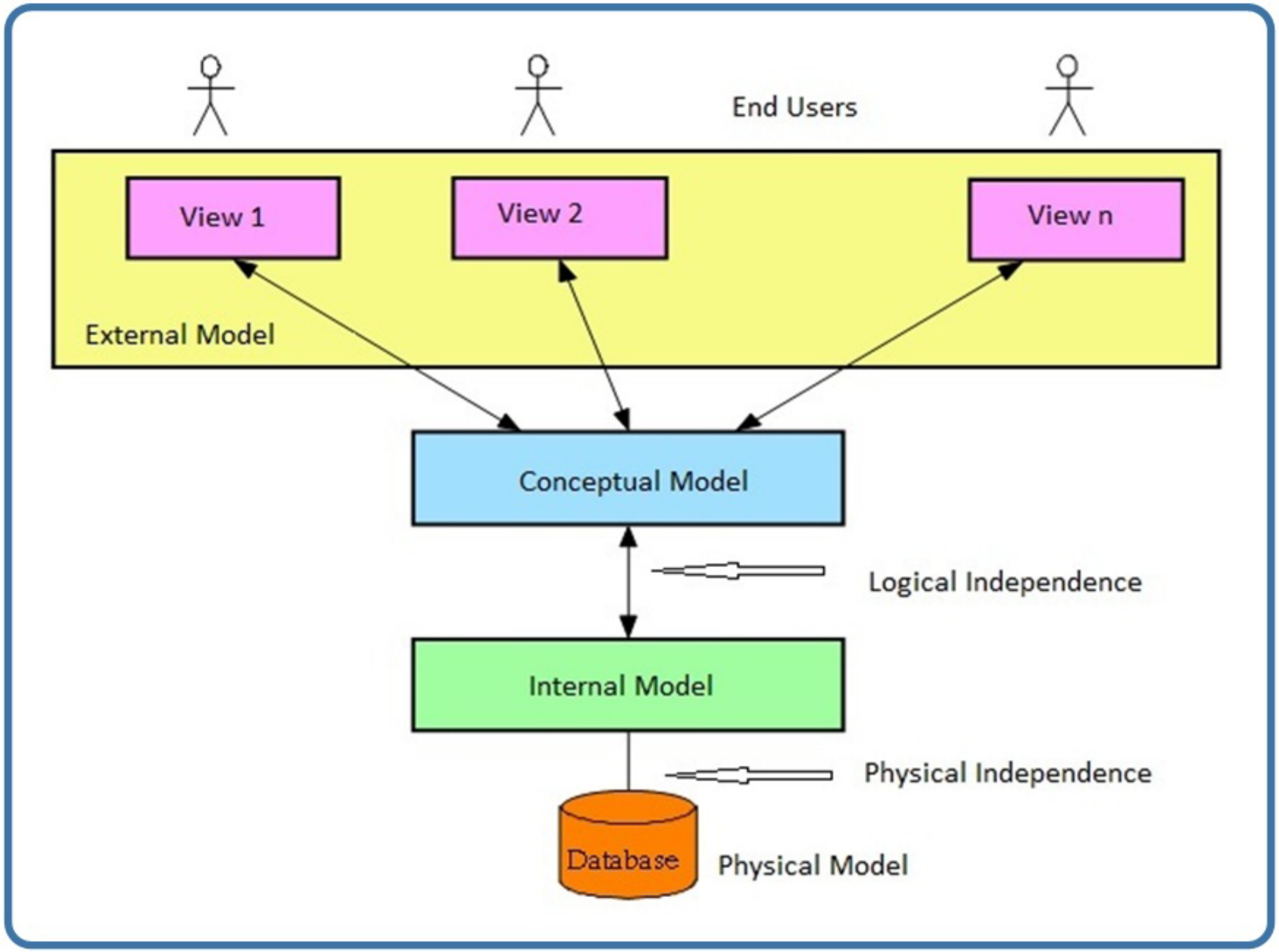
- **What is independence?**

  The ability to modify schema definition in on level without affecting schema definition in the next higher level is called data independence

- **The ability to change the logical (conceptual) schema without changing the External schema (User View) is called logical data independence.**

- **EXAMPLE:**

  The addition or removal of new entities, attributes, or relationships to the conceptual schema should be possible without having to change existing external schemas or having to rewrite existing application programs.


End Users
View 1
View 2
View n
External Model
Conceptual Model
Logical Independence
Internal Model
Physical Independence
Database
Physical Model

RULE 9

- PHYSICAL DATA INDEPENDENCE

RULE 9:PHYSICAL DATA INDEPENDENCE

- The ability to change the physical schema without changing the logical schema is called physical data independence.

- This is saying that users shouldn’t be concerned about how the data is stored or how it’s accessed. In fact, users of the data need only be able to get the basic definition of the data they need.

- **EXAMPLE:**

  A change to the *internal schema*, such as using different file organization or storage structures, storage devices, or indexing strategy, should be possible without having to change the conceptual or external schemas.

RULE 10

- INTEGRITY INDEPENDENCE RULE

RULE 10 : INTEGRITY INDEPENDENCE RULE

Data integrity refers to maintaining assuring the accuracy and consistency of data over its entire life cycle.

1. First insure that correct data type is used.
2. Check constraints: these allow column value to be checked agenized other column before insertion is allowed.

RULE 11

- DISTRIBUTION INDEPENDENCE RULE

RULE 11 : DISTRIBUTION INDEPENDECE RULE

- **“THE RELATION DATA BASE MANAGEMENT HAS DISTRIBUTION INDEPENDENCE”**

- **Distribution independence implies that user should not have to be aware of whether a database is distributed at different sites or not.**
- **Application program and adhoc request are not affected by the change in distribution of physical data. Application program will work even if the programs and data are moved on different site**

- **The RDBMS may spread across the more one system or several networks**

RULE 12

- NON-SUBVERSION RULE

RULE 12 : NON-SUBVERSION RULE

There should be no way to modify to database structure other then through the multiple row data base language(SQL).

<u>Example:</u>

A relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity Rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).”

MULTIPLE

CHOICE

QUESTIONS

Question 1

What is true regarding NULL value requirements confirming to Codd rule?

- a. Null value should be zero.
  b. Null value should be space.
  c. Null value should represent missing information.
  d. Either a or b.

Answer :

- c. Null value should represent missing information. According to Codd rule, Null value should not be any regular data like zero, spaces etc, but should represent that data is not available.

<u>Question 2</u>

According to Codd rule, how one should be able to access information about data structures like databases, tables etc.

- a. Should be directly accessible via all programming languages.
  b. Should be directly accessible via same query language used for data manipulation (e.g. select, update statements etc)
  c. Should be directly accessible via XML
  d. All of the above

<u>Answer :</u>

- b. Should be directly accessible via same query language used for data manipulation. For example, SQL can be used to retrieve information about the tables, the column types etc.

Question 3

Which is true regarding multi row update?

- a. Multiple row updates should be prohibited.
  b. Multiple row updates should be allowed only on tables without null values.
  c. Multiple row updates should be possible.
  d. Multiple row updates should be allowed only on integer data types

Answer :

- c. Multiple row updates should be possible.