Relational Data model and Codd's Rules

- **Relational Data Model**

  E.F.Codd from IBM proposed Relational Data model in year 1970.He is called father of RDBMS. The main elements of Relational Data model are based on Codd's 13 rules. It is the most common model for a relation systems in Industry today. Majority of the current database systems based on relational model are namely IBM's DB2 and Informix, Oracle, MySQL, Microsoft SQL Server. Before the advent of Relational model, there were other models Hierarchical model and Network model, but relational model is popular because it is simple and Elegant model. This model is based on mathematical Relation, so the name Relational Data model, this relation is nothing but what we are familiar in mathematics that is subset of cross product of sets (A relation from a set A to a set B is a subset of A×B).Finally this model represents database as a collection of relations, so the name relational data model.

- **Relational Data Model-Basic Terminology**
  - **Relation**

    Given sets A_1, A_2 , A_3 …….. A_n ,

    A relation R is subset of $A_1 X A_2 X A_3 X \ldots\ldots X A_n$

    i.e $R \subseteq A_1 X A_2 X A_3 X \ldots\ldots X A_n$
  - **Example**

    If Student_Id={ 501,502,503,504…}

    Student_Name={ Kaushal,Teja, Pavan,Nidhi,…}

    Student_Rank={8,6,7,4,…} Then the Relation

    $R=\left\{\begin{matrix}(501,Kaushal,8)\\(502,Teja,6)\\(503,Pavan,6)\\(504,Nidhi,4)\end{matrix}\right\} \subseteq$ Student_Id X Student_Name X Student_Rank
  - **Table**

    Relation is represented by Two-dimensional table.The elements of Student relation are represented as

Sample Table for extended Student Relation

Student Relation			
SID	**SNAME**	**SCORE**	**AGE**
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

  - **Attribute**

    Attributes are the properties that define a relation. Each column of the table is a attribute. In the following STUDENT table SID, SNAME, SCORE and AGE are attributes.

Student Relation			
SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

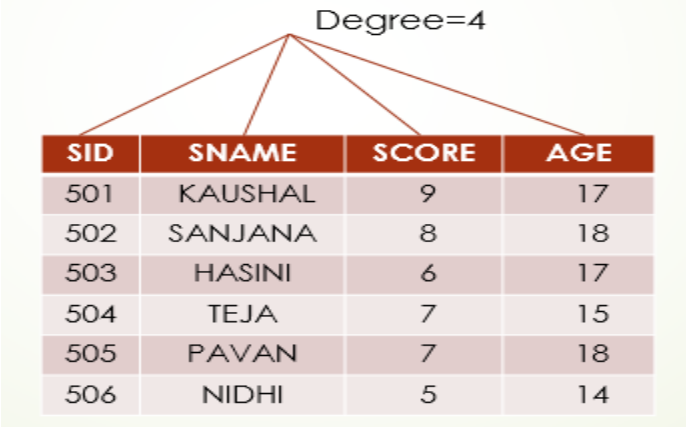
- **Domain**
  Domain specifies the values that appear in a column must be drawn from the domain associated with that column. Thus, the domain of a field is essentially the type of that field, in programming language terms, and restricts the values that can appear in the field. In this Student relation if domain of ID is integer with 501 to 575 as range, then a value outside this range will not be valid. Domain value can be non-continuous.

- **Relation Schema**
  Name of the relation with its attributes
  Student (**SID, SNAME, SCORE, AGE**)

- **Relation Instance**
  Set of tuples of a relation at a particular instance of time. Simply it is a snapshot of relation. It can change for every insertion, deletion or updation in database. Relation is a combination of Relation Schema and Relation Instance.
- **Tuple**
  Each row in a relation is tuple. For example highlighted part in Student relation represents on tuple. Student relation contains 6 tuples.

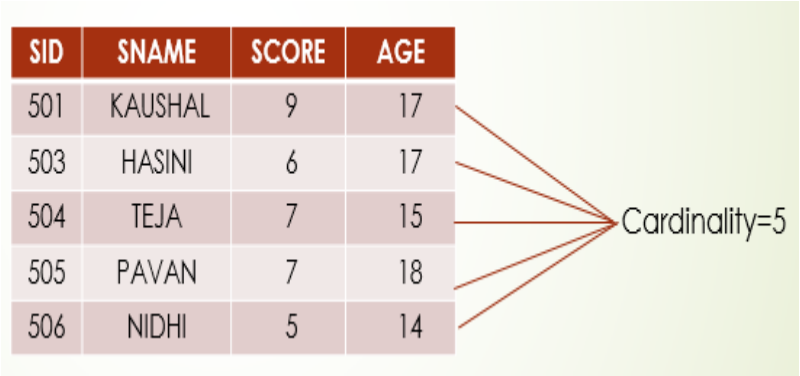
Student Relation			
SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

- **Degree**
  Number of attributes in a relation is known as Degree of relation.



SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

- **Cardinality**
  Number of Tuples in a relation is knowns as cardinality.



SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

- **Codd' s Rules**
  Codd proposed 13 rules popularly referred as Codd's 12 rules(0-12), which the database must follow to be regarded as Relational database.
  - **Rule 0: Foundation Rule**
    The system must use its relational capabilities exclusively to manage the database. The other 12 rules are derived from this rule.
  - **Rule 1: Information Rule**
    Data stored (user data or meta data) in Relational model must be a value of some cell of a table. Whatever data we are storing it should be in cell of a table.

SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

  - **Rule 2: Guaranteed Access Rule**
    Every data element must be accessible by combination of table Name, Row Value (primary key) and Column Value(attribute name).In the Student relation to access age of SID 503, we have to access through Student Relation+ 503 tuple + age attribute.

Student Relation			
SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15
505	PAVAN	7	18
506	NIDHI	5	14

- **Rule 3: Systematic Treatment of NULL values**
  NULL value can be treated as- Missing Value, UnKnown Value or No Value. It should be treated uniformly and systematically. Primary Key can't be NULL.

	Student Relation			
SID	SNAME	SCORE	Phone No	AGE
501	KAUSHAL	9	9492543638	17
502	SANJANA	8	9949122414	18
503	HASINI	6	NULL	17
504	TEJA	7	9853678966	15

- **Rule 4: Active Online Catalog**
  The metadata must be stored and managed as ordinary data, that is, in tables within the database. Such data must be available to authorized users using the standard database relational language

- **Rule 5: Comprehensive Data Sub-language Rule**
  A database should be accessible by a language to define, manipulate and perform transaction operations on data. The relational database must support one well-defined, declarative language with support for data definition, view definition, data manipulation integrity constraints, authorization, and transaction management
  Example:SQL

- **Rule 6: View Updating Rule**
  This rule states that all views of database, which can theoretically be updated, must also be updatable by the system. In the following table if we create view over attributes (SID,SNAME),any changes made on this view, should reflect in the main relation Student.

Student			
SID	SNAME	SCORE	AGE
501	KAUSHAL	9	17
502	SANJANA	8	18
503	HASINI	6	17
504	TEJA	7	15

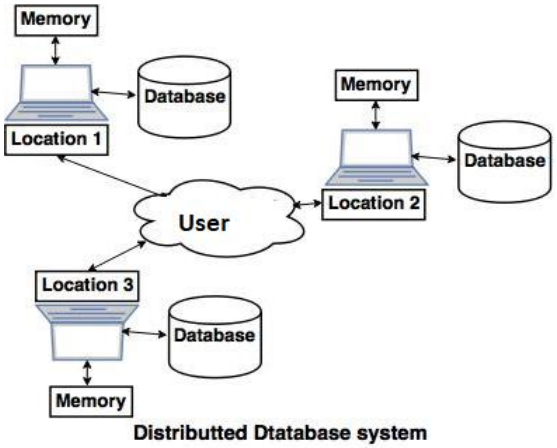
- **Rule 7: High level insert, update and delete rule**
  The database must support set-level inserts, updates, and deletes.

- **Rule 8: Physical data independence**
  Any Changes to the physical level (storage representations or access methods) must not effect at application level.

- **Rule 9: Logical data independence**
  Logical data must be independent of its application. Any modification in conceptual schema must not imply any change in the application using it. For example if change the order of columns or insert any new column, it should not effect the corresponding application program.

- **Rule 10: Integrity independence**
  The database should be able to implement integrity constraints than using application programs.

- **Rule 11: Distribution Independence**
  Distribution of the database to various locations should be invisible to users of the database. This is the Foundation for Distributed database.



Distributted Dtatabase system

- **Rule 12: Non-Subversion Rule**
  If a system has an interface that can access to low level records, then that interface cannot be used to subvert the system, for example, bypassing a relational security or integrity constraint.

- **References**
  - **TEXT BOOKS**
    1. "Database System Concepts" by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill, 2010, 9780073523323
    2. "Fundamentals of Relational Database Management Systems "by S. Sumathi,S. Esakkirajan, Springer Publications
    3. Data base systems Design,implementation and management by CARLOS ORONEL , STEVEN MORRIS , PETER ROB,Ninth Edition

  - **WEBLINKS**
    1. **Lecture 2:** **https://nptel.ac.in/courses/106/104/106104135/**
    2. **Lecture 7:** **https://nptel.ac.in/courses/106/106/106106220/**