# Relational Data Model

Ortiz Vega Angelo
Course Code: CE3101
Name: Databases.
Academic Area of Computer Engineering.
Cartago, Costa Rica.

**-Keywords:** Relational Data Model, DataBases, SQL, DBMS, DBA, Data Abstraction, Data Model, Database Schema, Schema Diagram, Snapshot, Database Languages.

**-Content:**

The relational data model was first introduced by Ted Codd of IBM Research in 1970. The model uses the concept of a mathematical relation. The first commercial implementation of the relational model became available in the early 1980.

**Concepts:**

1. The relational model represents the database as a collection of relations.
2. Informally, each relation resembles a table of values or, to some extent, a flat file of records.
3. A row represents a fact that typically corresponds to a real-world entity or relationship.

**Terminology:**

1. A **row** is called a **tuple**.
2. A **column header** is called an **attribute**.
3. A **table** is called a **relation**.

**Domains, Attributes, Tuples and Relations.**

**Domain:**

A domain D is a **set of atomic values.** A **data type** or **format** is also specified for each domain.

**Relation:**

A relation schema R denoted by $R(A_1, A_2, ..., A_n)$, is made up of a relation name R and a list of attributes $A_1, A_2, ..., A_n$. Each attribute $A_i$ is the name of a role played by some domain D in the relation schema R. The degree of a relation is the number of attributes n of its relation schema.

**Characteristics of Relations:**

1. Ordering of tuples in relation. Elements of a set have no order among them.
2. Ordering of values within a tuple. An -tuple is an ordered list of n values.
3. Values and NULL is the tuples. Each value in a tuple is an atomic value.
4. Interpretation of a relation: The

relation schema can be interpreted as a declaration or a type of assertion.

**Relational Model Constraints:**

It is a feature of databases using the relational model. Exist three principal categories:

1. Inherent model-based constraints or implicit constraints.
2. Schema-based constraints or explicit constraints.
3. Application-based or semantic constraints or business rules.

**1. Inherent model-based constraints or implicit constraints.**

Restrict the values that can be stored considering the model definition.

*Example: A relation cannot have duplicate tuples is an inherent constraint.*

**2. Schema-based constraints of explicit constraints.**

Are defined when creating the schema in the deployment model (relational in this case). Are expressen in the language DDL.

Categories:

a. Domain Constraints.
   i. Specify that the value of each attribute A must be an atomic value from the domain dom(A).
   ii. The data types associated with the domains typically include standard numeric data types for integers.
   iii. Other possible domains may be described by a subrange of values from a data type or as an enumerated data type which all possible values are explicitly listed.

b. Key Constraints and Constraints on NULL Values.
   i. A relation is defined as a set of tuples.
   ii. By definition all elements of a set are distinct, hence, all tuples in a relation must also be distinct.
   iii. Denote one subset of attributes by $SK$, then for any two distinct tuples $t_1$ and $t_2$ in a relation state r of $R$, we have the constraint that $t_1[SK] \neq t_2[SK]$ .
   iv. Any such set of attributes $SK$ is called a superkey of the relation schema $R$.
   v. A superkey $SK$ specifies a uniqueness constraint that no two distinct tuples in any state r of R can have the same value of $SK$.
   vi. Every relation has at least one default super key - the set of all its attributes.
   vii. A key K of a relation schema R is a superkey of R with the additional property that removing any attribute A from K leaves a set of attributes K that is not a superkey of R any more.
   viii. It is a minimal superkey

-that is a super key from which we cannot remove any attributes and still have the uniqueness constraint

c. Integrity, Referential and Foreign Keys.
   i. **Integrity:** The entity integrity constraint states that no primary key value can be NULL.
   ii. **Referential Integrity:** The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in two relations. Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.

**3. Application-based or semantic constraints or business rules.**

Some restrictions reflect complex business rules.Due to their complexity can not be expressed in the data model, so that should be implemented at the level of code in the application or in the trigger.

*Example: The salary of an employee may not decrease, only increase.*