Relational Model

Definition

The **Relational Model** is the foundation of modern databases and was introduced by **E.F. Codd** in 1970.

It organizes data into **relations**, which are represented as **tables**. Each table consists of **rows (tuples)** and **columns (attributes)**.

Key Features

Simplicity

- Data is stored in a tabular format, making it easy to understand

Mathematical Foundation

- Based on set theory

Data Independence

- Logical and physical data are separated, ensuring flexibility

Key Concepts

Domain

Attribute

Tuple

Relation

Domain

A **domain** is the set of permissible values for an attribute

Think of it as the **data type** or the range of acceptable values

Examples

- A StudentID might have a domain of integers from 1 to 9999
- A DateOfBirth might have a domain of valid dates

Attribute

An **attribute** is a **column** in a table and represents a specific property of the entity

Examples

- **name, salary, joining_date** can be termed as attributes of an **employee**
- **author, genre, price** can be termed as attributes of a **book**
- **brand, RAM, processor** can be termed as attributes of a **PC**

Tuple

A **tuple** is a single **row** in a table, representing a single record

Examples

- A tuple in a student table can be (1, ‘Alice’, 17, ‘CSE’, ‘AEC’)
- A tuple in a book table can be (174, ‘Wings Of Fire’, ‘Autobiography’, 1500)
- A tuple in a item table can be (12, ‘Laptop’, 55000)

Relation

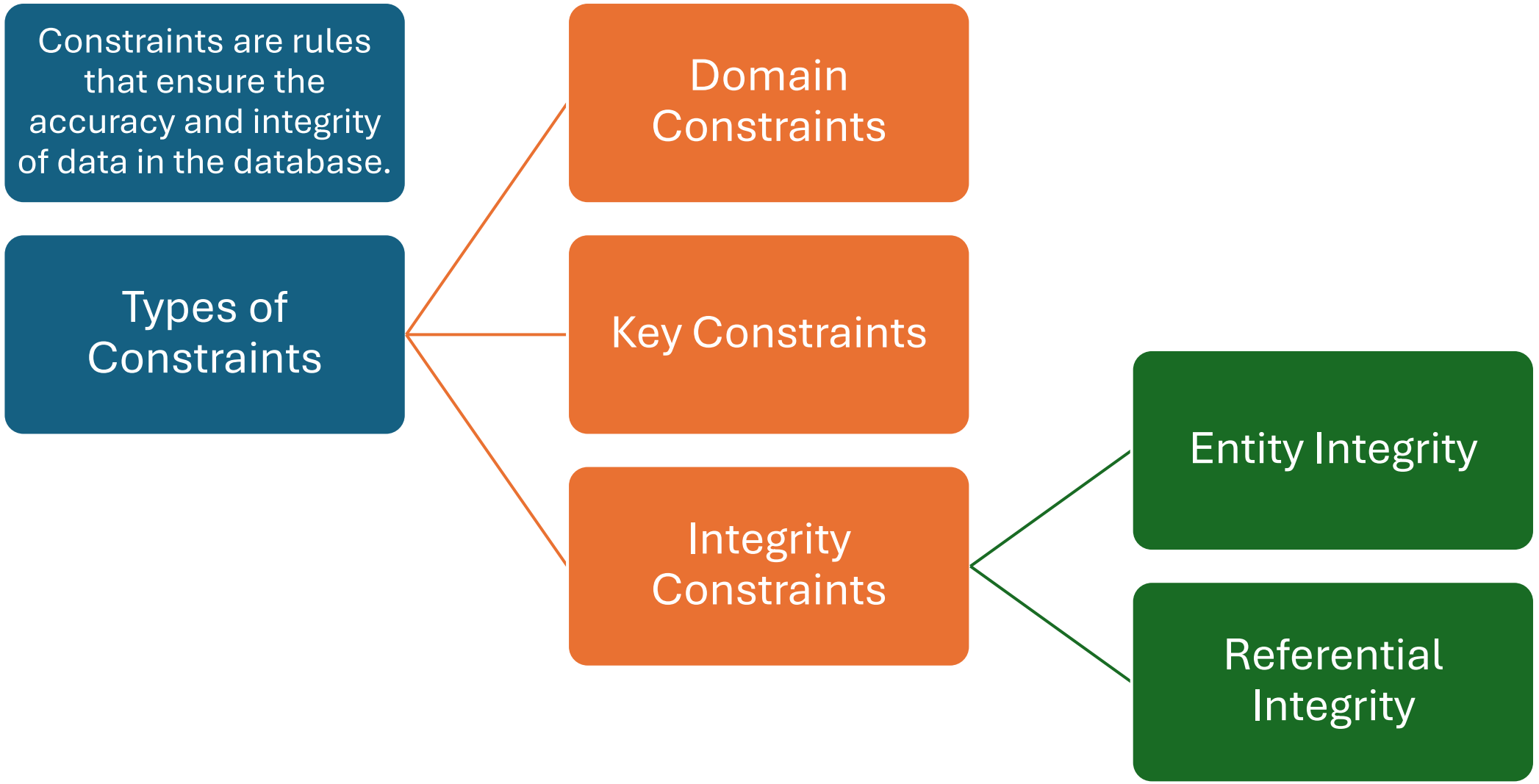
A **relation** is a **table** consisting of tuples (rows) and attributes (columns)

It's characterized by

- A unique name
- Each tuple being distinct
- Attribute values conforming to their domain

Constraints in Relational Model

Constraints in Relational Model



Domain Constraints

Ensure that attribute values fall within the specified domain

Examples

- If the **age** attribute has a domain of integers from 18 to 60, any value outside this range in invalid (Can be enforced with **CHECK()**)
- Ensuring **joining_date** attribute of an employee is greater than or equals to the **date_of_establishment** of organization
- Ensuring **grade** attribute of a student falls under a valid grade of **1 to 10**
- Ensuring **rating** attribute of a movie will never have a value less than 0 and greater than 10

Key Constraints

Ensure that each tuple in a relation is uniquely identifiable

Candidate Key: A set of attributes that are capable uniquely identifying a tuple

Primary Key: One chosen candidate key

Examples

- A set of **candidate key**s to identify an employee can be
  - {emp_id, aadhaar_number, PAN, email (If set to UNIQUE}
- One of these set of candidate keys can be chosen as a **Primary Key** upon developers choice

Integrity Constrains

Entity Integrity

- Primary Key attributes cannot have null values
- Ensuring every tuple is uniquely identifiable

Referential Integrity

- Maintains consistency between two related tables
- Often achieved through **foreign key** that refers to a **primary key** or any other **unique** valued attribute from another table
- Ensuring only existing user from **users** table can make **orders**
- Ensuring only existing student from **students** table able to **enroll** into an existing course from **courses** table

Real World Analogy

Domain: Defines acceptable inputs, e.g., roll numbers should be integers, and names should be strings, age cannot be more than 100 and so on

Attributes: Represent student details like roll number, name, and branch

Tuples: Each student is a unique row

Constraints: Ensure data is consistent. For instance, a course enrollment record must refer to a valid student and course

Relational Algebra

Relational Algebra (Procedural Query Language)

Definition: Relational Algebra is a **formal query language** that is used in relational databases

Relational Algebra uses a set of operations to manipulate relations (tables) to produce desired results

Basic Symbols (Operators) in Relation Algebraic Expressions

Operator Name	Symbol Used	Description
Selection	σ	Filters rows based on a condition
Projection	π	Selects specific columns from a table
Union	∪	Combines tuples from two relations (tables) without duplicates
Intersection	∩	Retrieves common tuples from two relations
Difference	–	Retrieves tuples in one relation but not in the other
Cartesian Product	×	Combines tuples from two relations (tables) in every possible way
Join	⋈	Combines tuples from two relations based on a condition
Rename	ρ	Renames a relation or it's attributes
Grouping / Aggregation	γ	Used to represent groups and aggregations
Ordering	τ	Used to perform ordering of projected data
AND, OR, NOT	∧, ∨, ¬	Combines conditions in Selection or Join

LaTeX Commands for the Symbols used in Relation Algebra

Operator Name	Symbol Used	Latex Command
Selection	σ	\sigma_{condition}
Projection	π	\pi_{attributes}
Union	∪	\cup
Intersection	∩	\cap
Difference	–	-
Cartesian Product	×	\times
Join	⋈	\bowtie
Rename	ρ	\rho_{new_name}
Grouping / Aggregation	γ	\gamma_{group_by, agg}
Ordering	τ	\tau_{attributes}
AND, OR, NOT	∧, ∨, ¬	\land, \lor, \neg