

Integrity Rules

- **Integrity constraints** are used to ensure accuracy and consistency of the data in a relational database.
- **Integrity constraints** are **set of rules** that the database is not permitted to violate.
- **Constraints** may apply to each **attribute** or they may apply to **relationships between tables**.
- **Integrity constraints** ensure that changes (update, deletion, insertion) made to the database by authorized users do not result in a loss of data consistency. Thus, **integrity constraints guard against accidental damage to the database**.
 - **Example** - A blood group must be 'A' or 'B' or 'AB' or 'O' only (can not any other

Types of Integrity Constraint

1. **Domain Constraint** / A
2. **Entity Integrity Constraint** - A
3. **Referential Integrity Constraint** - R-R
4. **Key Constraints** / A

Domain Constraint

- **Domain constraints** defines the domain or the valid set of values for an attribute.
- The **data type of domain** includes string, character, integer, time, date, currency, etc. *The value of the attribute must be available in the corresponding domain.*

STUDENT_ID	NAME	SEMESTER	AGE
101	Manish	1st	18
102	Rohit	3rd	19
103	Badal	5th	20
104	Amit	7th	A

Not allowed. Because AGE is an integer value

Entity Integrity Constraint

- The **entity integrity constraint** states that **primary key value can't be null**.
- This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- A table can contain a null value other than the primary key field.

EMP_ID	EMP_NAME	SALARY
111	Mohan	20000
112	Rohan	30000
113	Sohan	35000
	Logan	20000

↑
Not allowed as Primary Key can't contain NULL value

Referencial Integrity Constraint

- A **referential integrity constraint** is specified between two tables.
- **Referential Integrity constraint** is enforced when a foreign key references the primary key of a table.
- In the **Referential integrity constraints**, if a foreign key in Table 1 refers to the Primary Key of Table 2, then **either** every value of the foreign Key in Table 1 must be available in primary key value of Table 2 **or** it must be null.
- The **rules** are:
 - ▣ You can't **delete** a record from a *primary table* if matching records exist in a related table.
 - ▣ You can't **change** a primary key value in the *primary table* if that record has related records.
 - ▣ You can't **insert** a value in the foreign key field of the *related table* that doesn't exist in the primary key of the *primary table*.
 - ▣ However, you can enter a **Null** value in the foreign key, specifying that the records are unrelated.

EMP (Table 1): Related /Referencing Table

<u>EMP_ID</u>	EMP_NAME	AGE	DEP_NO
111	Mohan	21	1
112	Rohan	33	2
113	Sohan	27	3
114	Logan	25	5

Foreign Key

Relationships

Primary Key

DEP (Table 2): Primary/Ref Table

<u>DEP_NO</u>	LOCATION
1	Mumbai
2	Delhi
3	Noida

Not allowed as DEP_NO 5 is not defined as a Primary Key of Table 2 and In Table 1, DEP_NO is a foreign key defined

Key Constraint

- An entity set can have multiple keys or candidate keys (minimal superkey), but out of which one key will be the primary key.
- **Key constraint** specifies that in any relation-
 - ▣ All the values of primary key must be **unique**.
 - ▣ The value of primary key must **not be null**.

STUDENT_ID	NAME	SEMESTER	AGE
101	Manish	1st	18
102	Rohit	3rd	19
103	Badal	5th	20
102	Amit	7th	21

↑
Not allowed. Because all rows must be **unique**