

Integrity Constraints

Integrity constraints are rules defined in a database to ensure that the data stored remains accurate, consistent, and meaningful throughout its lifetime. Since databases are shared by many users and applications, these constraints prevent invalid data from being entered and help maintain the correctness of the database even when multiple operations occur. These constraints are discussed below.

a) Entity Integrity Constraint

This is one of the most fundamental integrity constraints. This rule states that every relation must have a primary key and that the primary key value cannot be null. The primary key uniquely identifies each record in a table, so allowing null or duplicate values would make it impossible to distinguish one record from another. Entity integrity therefore ensures that every tuple in a relation represents a real, identifiable entity.

b) Referential Integrity Constraint

Referential integrity maintains consistency between related tables. This rule states that a foreign key in one relation must either be null or match an existing primary key value in the related relation. Referential integrity ensures that relationships between tables remain valid. For example, if a student is registered for a course, the course referenced must actually exist in the course table. This prevents situations where records refer to non-existent data.

Employee				Department	
EID	Name	Salary	Dept-ID	Dept-ID	Dept-Name
1	Amit	6,000	1A	1A	Accounts
2	Sumit	10,000	2C	2C	Computer
3	Lalit	15,000	4F	3E	Electrical
4	Deepak	9,000	3F	4C	Civil
5	Sandeep	4,000	—		

Null value is allowed

This is not allowed because Dept-ID is a foreign key and the value 4F is not present in attribute Dept-ID of relation Department.

Fig 1: Referential Integrity

c) Domain constraints

Domain constraints restrict the values that an attribute can take. These constraints define the data type, size, format, and allowable range of values for each attribute. For instance, an age attribute may be restricted to integer values within a specific range, and a date attribute must follow a valid date format. Domain constraints help ensure that values stored in the database are meaningful and appropriate.

Employee			
EID	Name	Age	Salary
1	Aditya	22	10,000
2	Dinesh	-18	5,600
3	Sumit	25	8,000
4	Lalit	20	ABC
5	Gaurav	23	11,000

Annotations:

- An arrow points from the value '-18' in the Age column of row 2 to the text "Not allowed because age must be greater than 18".
- An arrow points from the value 'ABC' in the Salary column of row 4 to the text "Not allowed because salary has integer data type".

Fig 2: Domain Constraints

d) Key Constraints

In any relation R, if attribute A is primary key then A must have unique value or you can say that primary key attribute must have unique value. Duplicate values in primary key are invalid.

Employee		
EID	Name	Age
1	Aditya	22
2	Sumit	19
3	Deepak	25
2	Manoj	24
4	Dheeraj	28

Annotations:

- An arrow points from the value '2' in the EID column of row 5 to the text "Invalid".

Fig 3: Key Constraints

e) Tuple Uniqueness Constraints

In any relation R, all tuples in relation R must have distinct values. In other words, duplicate tuples within a single relation are not allowed.

Employee		
EID	Name	Age
1	Aditya	22
2	Sumit	19
3	Deepak	25
2	Sumit	19

Fig 4: Tuple Uniqueness Constraints