# CONSTRAINTS

Constraints are set of rules or protocol which are applied on all parts of database to maintain and control the operations performed on database. A database should be in Integrated and consistent state at every instance. Constraints make sure that after every operation performed on database the database remain consistent. Also, constraints maintain user authorization.

Constraints are required to maintain consistency, integrity, security, and reliability. Constraints are applied on each and every tuple of the database table.

Actually, when any kind of operation performed on any table of a database like insert, delete, update, modify then these constraints pretend like procedure for performing such operations. Each and every user should follow these.

Note: Few constraints are system generated and few constraints are defined by user.

Constraints are divided in these main categories:

1. Mapping Constraints
2. Participation Constraints
3. Integrity Constraints
    1. Domain constraints
    2. Entity Integrity constraints
        1. Not null
        2. Primary key
    3. Referential integrity constraints
        1. Foreign key
4. Key Constraints

On broad level we can classify the constraints as per given below:

(1) NOT NULL: No column should hold null values.

(2) UNIQUE: try to ensure that each and every column should hold unique values, means no duplicate values.

(3) DEFAULT: Only allowed values should be inserted in any specific column. Like in student database the school's name for every student should be same, so the name of school is a default value. If school name is RMDPS then in school name column all values should be RMDPS like below given table. No other value is allowed.

| SID | SNAME | SCHOOL NAME |
|---|---|---|
| 1 | Raj | RMDPS |
| 2 | Jyoti | RMDPS |
| 3 | Harsh | RMDPS |
| 4 | Vikash | RMDPS |

(4) CHECK: Insert the values in any specific column form the allowed range only. Like the student ID should be less than 1000.
(5) INDEX: used to search and fetch data as quickly as possible.
(6) KEY CONSTRAINTS – PRIMARY KEY, FOREIGN KEY
(7) DOMAIN CONSTRAINTS: Each column in a table has its own datatypes. No other datatypes data should be inserted in that column.
(8) MAPPING CONSTRAINTS: Show the limits of entity-to-entity relationship.

We have already read about Mapping Constraints and Participation Constraints in previous videos, here we will read about Integrity Constraints.

# INTEGRITY CONSTRAINTS:

Integrity constraints is a protocol or set of rules which are applied on database. The main objective of integrity constraints is to secure the database against any accidental or unwanted damage. The data should be remained accurate and consistent even after each and every operation performed on the database. Integrity constrains are used to do so.

Types of Integrity constraints:

1. Domain Constraints
2. Entity Integrity Constraints
3. Referential Integrity Constraints
4. Key Constraints

## 1. Domain Constraints:

In simple words in a relational table a domain is nothing else just an attribute or a column of a table.

| SID | Name | Age |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |

Here SID, Name, Age are domains. So, in short domain constraints are nothing else just rules applied on a domain, and every value entered in that domain needs to follow the rules.

Also, these all attributes have some rules to follow:

(1) Every attribute has a specific datatype like integer, variable-character, string, time, date, currency etc. An attribute can allow only its datatype values in its domain. In above given table the datatypes of the attributes or domains are as follows:

SID = Integer
Name = Varchar
Age = Integer

So, if we insert any other datatype in any column then it is not allowed. Like in given below table "Age" column's datatype is integer so if we insert any varchar value in the table then the database will not allow it.

| SID | Name | Age |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |
| 03 | Jyoti | Twenty-two |

(2) If we applied any constraints on any attribute/ Null, Unique etc, then each value entered in the column should follow the constraint's restriction.

Not Allowed, only integer is allowed

| SID (unique) | Name | Age (Not Null) |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |

In this table the SID is unique means the values entered in this column should be unique.

Also, Age is Not Null means we can't leave this column blank.

So, in short domain constraints are nothing else just rules applied on a domain.

## 2. Entity Integrity Constraints:

(1) According to the entity integrity constraint primary key value should not be NULL. Means we need to give some value in each row's primary key attribute.

This is followed because the other attributes are functionally dependent on their primary key attribute for their recognition in a relation and if the primary key has a null value, then we can't identify those attributes or rows.

Also, if two rows have Null Values then we can't distinguish between both rows.

Except primary key a table can keep a null value in any other attribute.

| SID (Primary Key) | Name | Age |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |

Here in this table SID is primary key so remaining two attributes such as Name and Age are dependent on SID.

| SID (Primary Key) | Name | Age |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |
|  | Jyoti | 22 |

The Primary Key should not be Null or Empty.

(2) Also, the Primary key should be Unique. Means no repeated values are allowed, because if two primary keys hold similar values then we can't distinguish between both the rows.

| SID (Primary Key) | Name | Age |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |
| 02 | Jyoti | 22 |



The Primary Key should be Unique.

## 3. Referential Integrity Constraints

A referential integrity constraint is specified between two tables, which are related via Primary Key-Foreign key concept.

In the Referential integrity constraints, if a foreign key is applied in a table no. 2, refers to the Primary Key of table no. 1, then every value of the Foreign Key in Table 2 must follow some rules:

(1) Both Columns should have same values for similar entities, means the primary key value for an entity should be used as foreign key in another table for the same entity.
(2) We can't delete or change the value of primary key if the value is existed in other table as foreign key value.
(3) We can't insert any random value in foreign key which not available in primary key.
(4) A foreign key may be allowed to hold Null values.

Table 1

| SID (Primary Key) | Name | Age |
|---|---|---|
| 01 | Raj | 21 |
| 02 | Harsh | 19 |
| 03 | Jyoti | 22 |

Table 2

| SID (Foreign Key) | Class | Subject |
|---|---|---|

| | | |
|---|---|---|
| 01 | 11 | Maths |
| 02 | 10 | Commerce |
| 03 | 12 | Biology |

## 4. Key Constraints:

We have already read about key constraints in a separate video. Please visit our channel for video or notes link.