Constraint (Rule)

Constraint are the predefined rules and restrictions that are enforced in single or multiple columns, regarding the values allowed in columns, to maintain integrity, accuracy and reliability of that columns data.

If the inserted data meets the constraint rule, it will be inserted into table successfully. If inserted data violates any of the defined constraint, the data will not be inserted and insert operation will be aborted.

Constraint in SQL server can be defined:

1. **Column Level Constraint:**

The constraint applied to only one column of table are called column level constraint. Column level constraints refers to a single column in table and do not specify a column name (Except check constraint). These refers to the column that they follow.

1. **Table Level Constraint**:

The constraint applied on two or more columns in table are called Table level constraint. Table level constraints specify the names of the columns to which they apply.

E.g.: Check Constraint

SQL constraints are used to specify rules for the data.

The constraint can be created within Create Table command while creating table or added using Alter Table command after creating table.

**Note:**

Adding constraint after creating table, the existing data will be checked for the specified constraints rule before creating those constraints.

Constraints used in SQL are:

1. Nullability Constraint/ Not Null Constraint
2. Unique
3. Primary Key
4. Foreign Key
5. Check
6. Default
7. Create Index

Nullability constraint / Set NULL Constraint

If we don’t define any nullability constraint on any column of table then, by default, the columns are able to hold NULL values.

A NOT NULL constraint in SQL is used to prevent inserting NULL values into the specified column, considering it as a not accepted value for that column. This means that you should provide a valid SQL NOT NULL value to that column in the INSERT or UPDATE statements, as the column will always contain data.

If we try to insert null inside Not Null Column the, there will be Error.

Create Nullability Constraint on any column

1. At time of Creating Table

Create Table Person

(

[Id] nvarchar(10) not null,

[Name] nvarchar(100) not null,

[Aadhar] nvarchar(12) null,

[Email] nvarchar(50)

)

1. Alter table:
2. Alter table <tableName>
3. Constraint

Unique Constraint

Unique constraint enforces uniqueness of column i.e., the column should not allow any duplicate values.

Primary Key VS Unique:

Both primary key and unique are used to enforce uniqueness of a column but:

A table can have only one primary key but can have more than one unique key.

If we want to enforce uniqueness on 2 or more column then we should use unique.

1. A table can have only 1 primary key column.

A table can have more than 1 unique key column.

So, if we want to enforce uniqueness on 2 or more column then we should use unique.

1. Primary key column doesn’t allow NULL

Unique key column allows one NULL.

Can unique key act as primary key?

Declare unique key column as Not Null then it will act as primary key. Then, it will allow only unique and not null values in it.

Primary Key Constraint

When we apply primary key on column(s) in a table then column will not contain any NULL value & not allow duplicate value.

**Primary Key Column = No Duplicate + Not Null**

Primary key column is used to identify a record uniquely from a table. Primary key act as a row header which identifies a particular row uniquely.

The PRIMARY KEY constraint consists of one column or multiple columns with values that uniquely identify each row in the table.

The SQL PRIMARY KEY constraint combines between the UNIQUE and SQL NOT NULL constraints.

where the column or set of columns that are participating in the PRIMARY KEY cannot accept a NULL value.

If the PRIMARY KEY is defined in multiple columns, you can insert duplicate values on each column individually, but the combination values of all PRIMARY KEY columns must be unique.

Take into consideration that you can define only one PRIMARY KEY per each table, and it is recommended to use small or INT columns in the PRIMARY KEY.

In addition to providing fast access to the table data, the index that is automatically created, when defining the SQL PRIMARY KEY, will enforce the data uniqueness. The PRIMARY KEY is used mainly to enforce the entity integrity of the table. Entity integrity ensures that each row in the table is a uniquely identifiable entity.

PRIMARY KEY constraint differs from the UNIQUE constraint in that; you can create multiple UNIQUE constraints in a table, with the ability to define only one SQL PRIMARY KEY per each table. Another difference is that the UNIQUE constraint allows for one NULL value, but the PRIMARY KEY does not allow NULL values.

Foreign Key

Foreign key is used to enforce database integrity. A foreign key column in one table refers (points) to primary key column of another table.

Foreign key is used to enforce database integrity. Foreign key constraint prevents invalid data from being inserted into the foreign key column.

The values entered into the foreign key column has to be one of the values contained in the column it points (refers) to. (i.e., foreign key column can accept null or values from the primary key column & does not accept remaining values).

Foreign key column can have: NULL (if specified null), Duplicate values (Only the values from primary key column).

**Note:**

Primary Key 🡪 Foreign Key relationship is permanent & physical relationship between two tables.

To break this relation: We need to remove foreign key constraint between tables than tables will be no more related.

Cascading Referential Integrity Constraint

Cascading referential integrity constraint allows to define the actions SQL server should take when a user attempts to delete/update a primary key to which an existing foreign key point (refers) to.

By default, we get an error if we try to delete/update primary key column & delete/update statement is rolled back because by default cascading referential integrity constraint is set to NO ACTIONS.

Options when setting up cascading referential integrity constraints:

1. No Action (Default Behaviour)

It specifies that if attempt is made to delete/update primary key column in another table (i.e., a row with a key referenced by foreign keys in the existing rows in another table), An Error is raised & Update/Delete statement is rolled back.

1. Cascade

It specifies that if attempt is made to delete/update primary key column in another table then, all rows containing those foreign keys are also deleted/updated.

1. Set NULL

It specifies that if attempt is made to delete/update primary key column in another table then, all rows containing those foreign keys are also set to NULL.

1. Set Default

It specifies that if attempt is made to delete/update primary key column in another table then, all rows containing those foreign keys are set to default values.

Relationship is required for cascading update/delete statement. These relations are defined through foreign key constraint (i.e., on the table which has foreign key defined).

Constraint <foreignKey\_Constraint\_Name> Foreign Key (ForeignKeyColumnName)

References TableName(<PrimaryColumnName>)

1. ON DELETE Set NULL
2. ON UPDATE set NULL
3. ON DELETE Cascade
4. ON UPDATE Cascade
5. ON DELETE No Action
6. ON UPDATE No Action
7. ON DELETE set Default
8. ON UPDATE set Default

(For 7 & 8 cascading referential integrity constraint, foreign key column must have defined default constraint defined on them)

For 1 & 2 cascading referential integrity constraint, foreign key column must not have defined Not Null.

Default Constraint

A column default value can be specified using default constraint (i.e., used to insert default value into the column).

If while submitting data, we don’t insert (supply) any values for default constraint column, then default specified value will get inserted into that column.

But if we insert null or ‘different’ value then that specified value will get inserted (i.e.,

If we supply (insert) value for default constraint column event NULL then, that specified value will get inserted (NULL also)).

Check Constraint

Check constraint is used to limit the range of the values that can be entered for a column.

**Constraint <constraintName> Check (<Boolean Expression>)**

If Boolean expression, returns True or ‘Unknown Result’ then the check constraint allows the value to insert into the table otherwise it doesn’t (if return False).

If the column having check constraint is nullable when inserting a row, it is possible to pass null for this column. When we insert null, then Boolean expression evaluates to unknown and allows null to insert.