

TABLE CREATION WITH CONSTRAINTS

AIM

To ensure data integrity and consistency in a database by enforcing rules such as NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY and DEFAULT.

DESCRIPTION

NOT NULL Constraint

The NOT NULL constraint enforces that a particular column must always contain a value when a record is inserted. If no value is provided, the database will throw an error.

Syntax:

```
CREATE TABLE table_name ( column_name datatype NOT NULL);
```

CHECK Constraint

The CHECK constraint is used to limit the range of values that can be inserted into a column. If the condition is violated, the insert/update fails.

Syntax:

```
CREATE TABLE table_name (column_name datatype CHECK (condition));
```

UNIQUE Constraint

The UNIQUE constraint prevents duplicate values in one or more columns. Multiple NULLs are allowed unless combined with NOT NULL

Syntax:

```
CREATE TABLE table_name (column_name datatype UNIQUE);
```

PRIMARY KEY Constraint

The PRIMARY KEY constraint is a combination of NOT NULL and UNIQUE. A table can have only one primary key, which may consist of a single column or multiple columns (composite key).

Syntax:

```
CREATE TABLE table_name (column_name datatype PRIMARY KEY);
```

Or for composite primary key:

```
CREATE TABLE table_name ( col1 datatype, col2 datatype,  
    CONSTRAINT pk_name PRIMARY KEY (col1, col2) );
```

FOREIGN KEY Constraint

The FOREIGN KEY constraint ensures that the value in one table corresponds to a valid primary key value in another table. This maintains consistency across related tables.

Syntax:

```
CREATE TABLE table_name (column_name datatype,  
    FOREIGN KEY (column_name) REFERENCES parent_table(parent_column));
```

DEFAULT Constraint

The DEFAULT constraint automatically assigns a predefined value when no explicit value is given during insertion.

Syntax:

```
CREATE TABLE table_name(column_name datatype DEFAULT default_value);
```