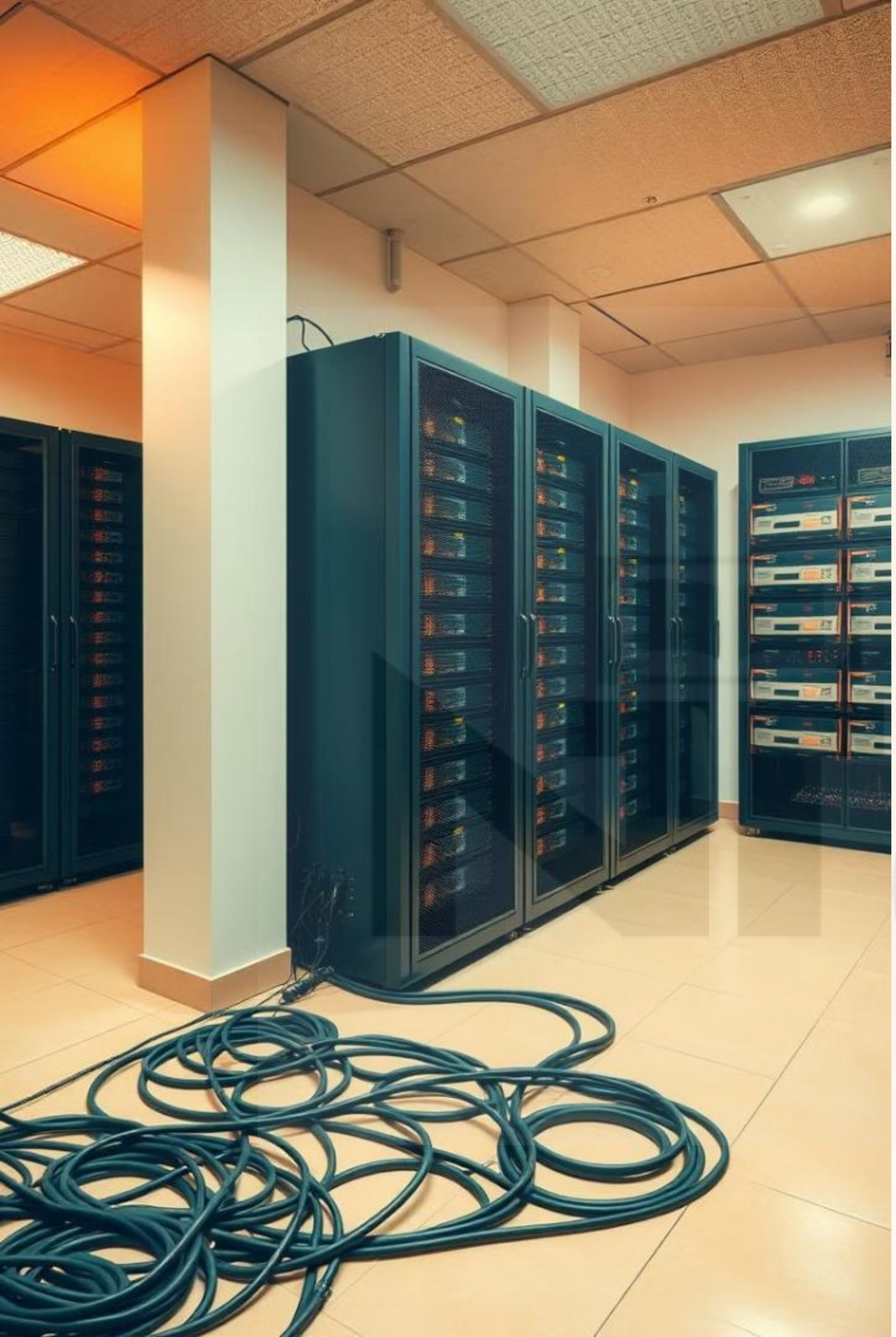


Introduction to SQL Constraints

Constraints are rules applied to data. They ensure data integrity and consistency.

MUKESH KUMAR

What are SQL Constraints?

- SQL **constraints** are rules applied to table columns to ensure **data accuracy, consistency, and integrity**.

- They restrict the type of data that can be inserted into a table.

Constrains in SQL

- **Constraints** are rules applied to **table columns** in SQL to enforce **data integrity, accuracy, and reliability**.

- They make sure that the data entered into a table is **valid**, **consistent**, and **follows business rules**.

Common Types of SQL Constraints

Constraint	Purpose
NOT NULL	Prevents NULL (empty) values
UNIQUE	Ensures all values in a column are different
PRIMARY KEY	Uniquely identifies each row (implies NOT NULL + UNIQUE)
FOREIGN KEY	Links a column to another table's primary key
CHECK	Ensures values meet a condition
DEFAULT	Sets a default value when no value is provided

Why Use Constraints?

- Prevent bad data (e.g., empty usernames)
- Enforce rules automatically (e.g., price > 0)
- Maintain referential integrity (via foreign keys)

What is Primary key?

- A **primary key** is a column or a set of columns in a table that is used to **uniquely identify each row** in that table

- A table can have **only one primary key**

- Primary keys often have the **NOT NULL** constraint implicitly or explicitly applied, meaning that the primary key column cannot contain null values

- Think of it like a student's roll number in a college or a user_id in a users table; it provides a unique identifier for each record

Primary Key Characteristics

Uniqueness

Each value in the primary key must be unique.

Not Null

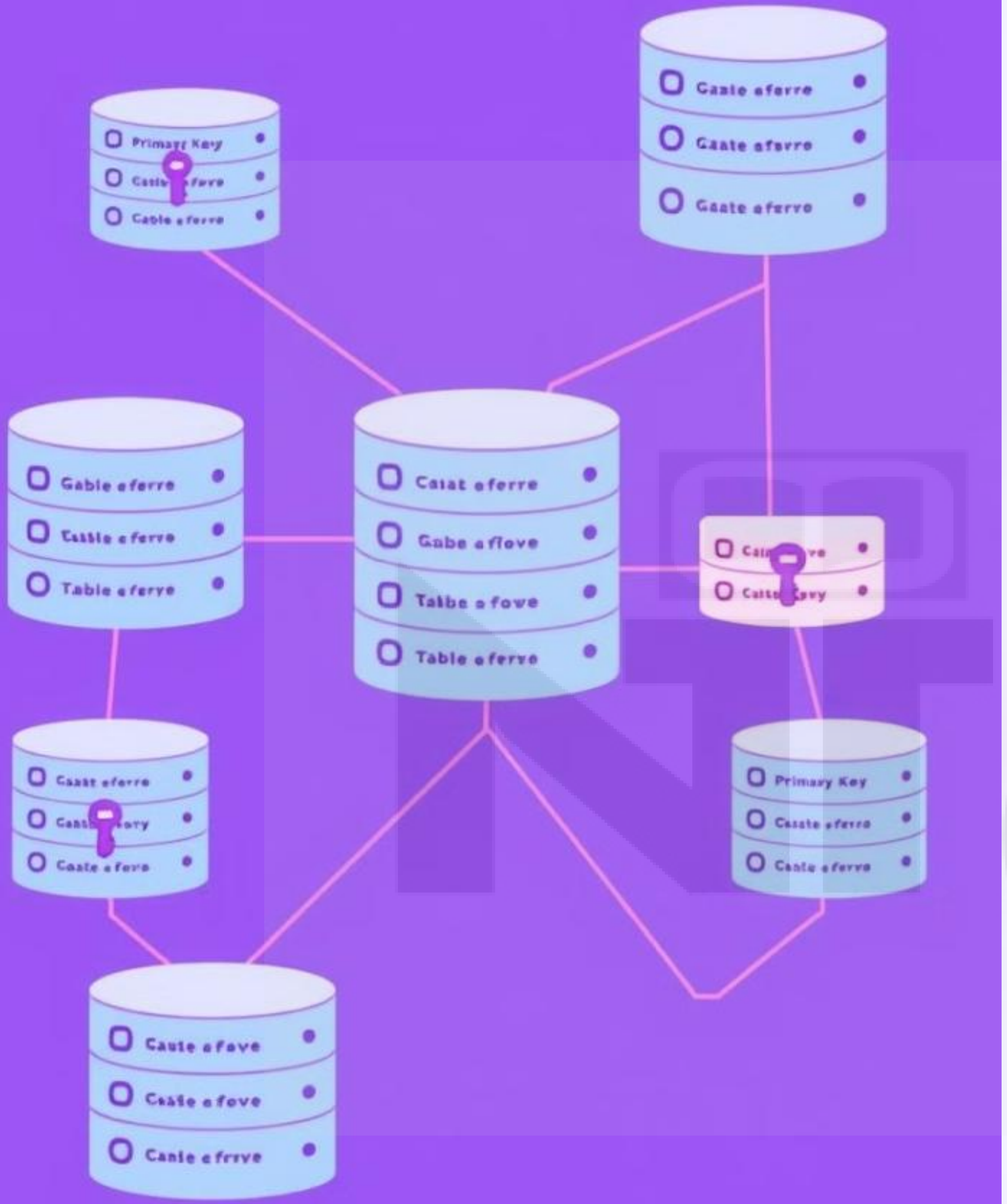
Fields containing the primary key cannot be empty.

Indexing

Automatically indexed for faster data retrieval.

One Per Table

A table can have only one primary key



Benefits of Primary Key

Data Integrity

Prevents duplicate records.

Data Retrieval

Faster queries and data manipulation.

Relationships

Supports connections with other tables.

Foreign Key: Establishing Relationships

Referential Integrity

Ensures consistency between tables.

Cascade Operations

Updates and deletes propagate across tables.

What is Foreign Key?

- A **foreign key** is a column (or a set of columns) in one table that **references the primary key of another table**
- Foreign keys are used to **establish and enforce relationships** between tables
- A foreign key column in one table doesn't need to have a unique value; it can have multiple rows referencing the same primary key value in the other table (reflecting one-to-many or many-to-many relationships

Benefits of Foreign Key

Integrity

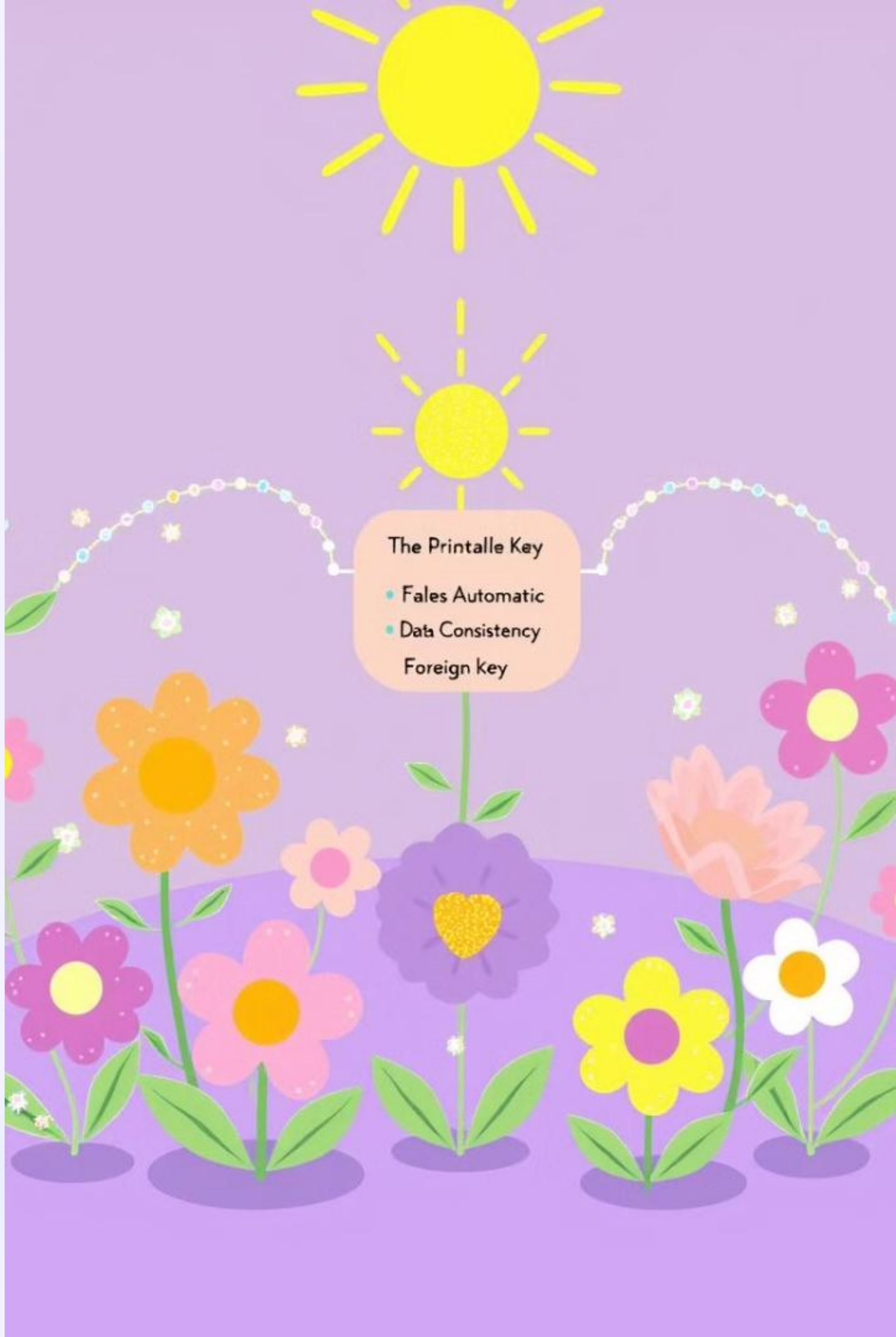
Maintains data consistency across the database.

Queries

Facilitates complex queries and joins.

Consistency

Data consistency through cascade operations.



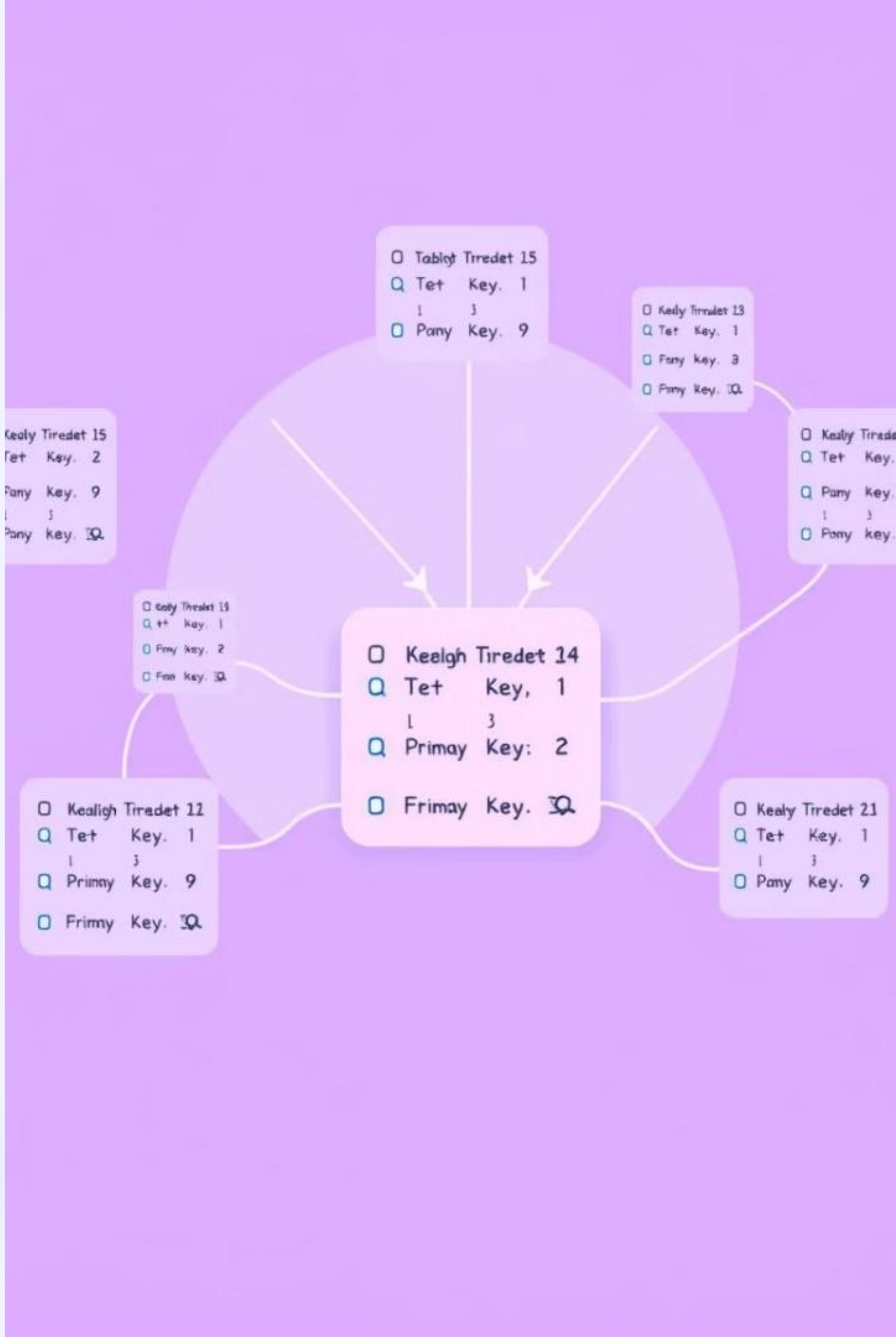
Primary Key vs Foreign Key

Feature	Primary Key	Foreign Key
Purpose	Unique identifier	Establishes relationships
Uniqueness	Must be unique	Not necessarily unique
Nullability	Not null	Can be null
Indexing	Automatically indexed	Not automatically indexed

Key Takeaways

Primary keys ensure data uniqueness.

Foreign keys maintain data consistency.



Constraints

1 NOT NULL

Column cannot have a NULL value.

2 UNIQUE

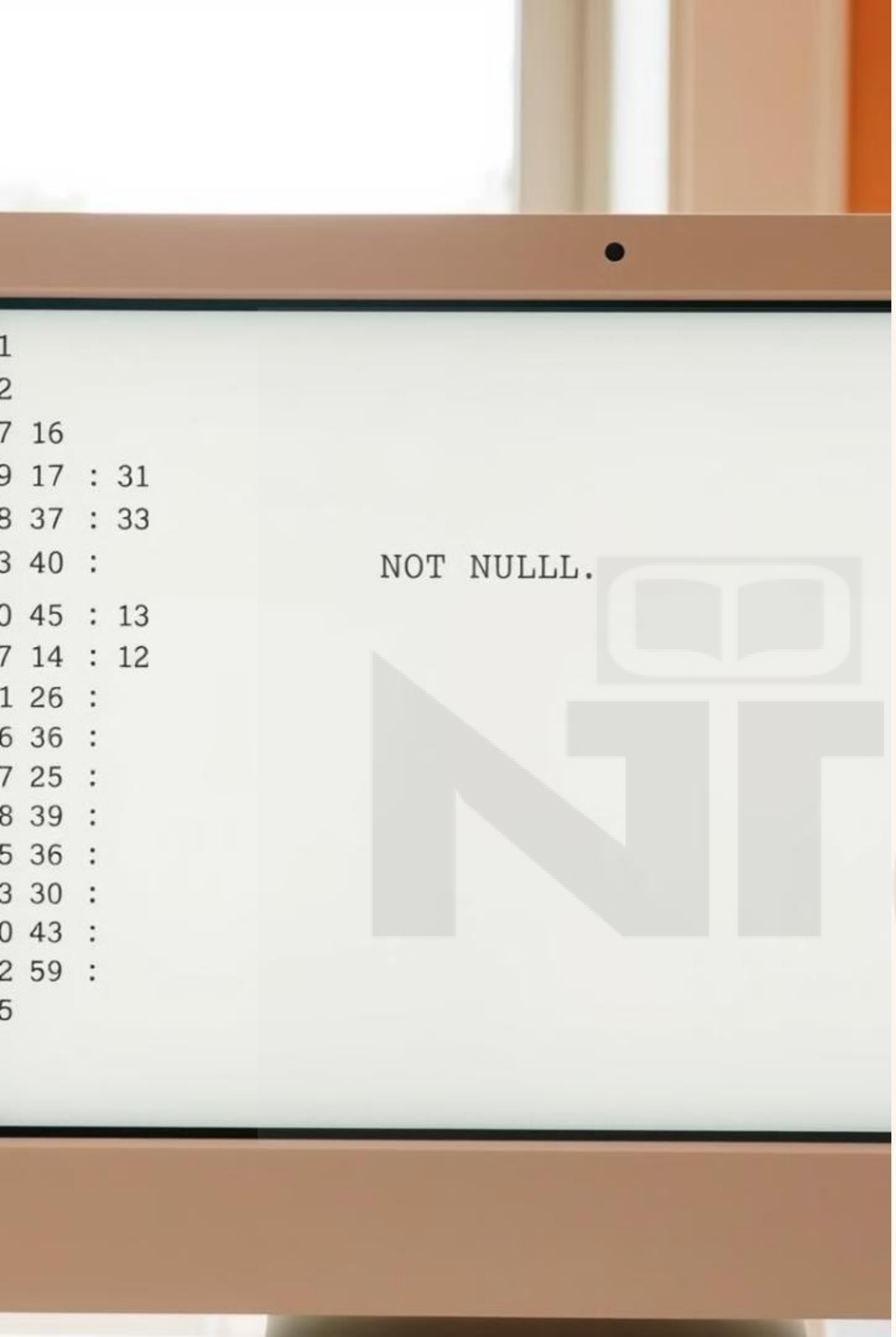
All values in a column are different.

3 CHECK

Data meets certain conditions.

4 DEFAULT

Sets a default value if none given.



NOT NULL Constraint

Ensures that a column cannot have a NULL value.

```
CREATE TABLE Students ( StudentID INT NOT NULL, Name
VARCHAR(255) );
```

UNIQUE Constraint

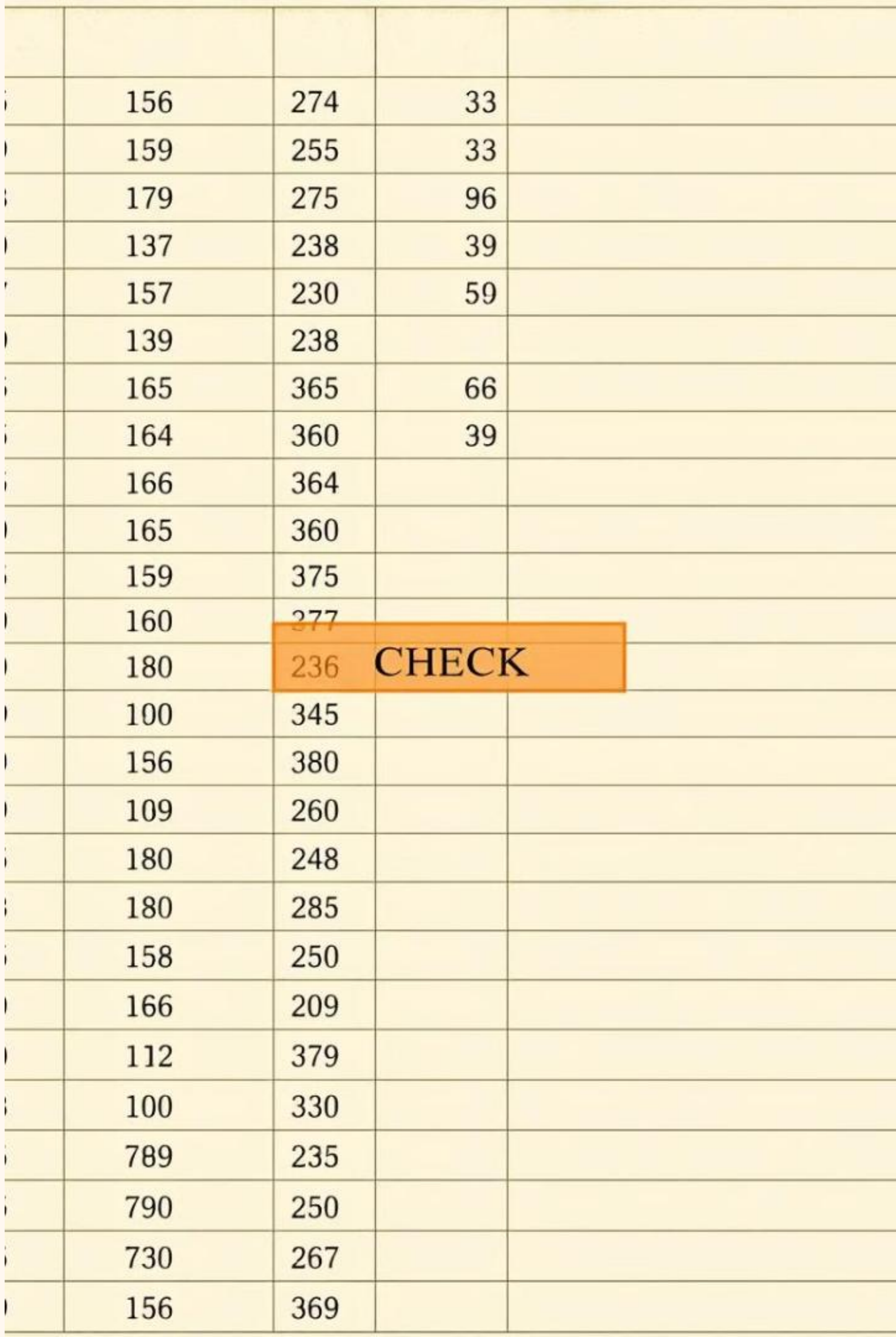
Ensures that all values in a column are different.

```
CREATE TABLE Products ( ProductID INT UNIQUE, ProductName
VARCHAR(255) );
```

CHECK Constraint

Ensures data meets conditions before insertion.

```
CREATE TABLE Employees ( Salary DECIMAL(10, 2) CHECK (Salary > 0) );
```



156	274	33
159	255	33
179	275	96
137	238	39
157	230	59
139	238	
165	365	66
164	360	39
166	364	
165	360	
159	375	
160	277	
180	236	CHECK
100	345	
156	380	
109	260	
180	248	
180	285	
158	250	
166	209	
112	379	
100	330	
789	235	
790	250	
730	267	
156	369	

DEFAULT Constraint

Sets a default value for a column if no value is specified.

```
CREATE TABLE Employees ( Salary DECIMAL(10, 2) DEFAULT
50000.00 );
```

Auto Increment

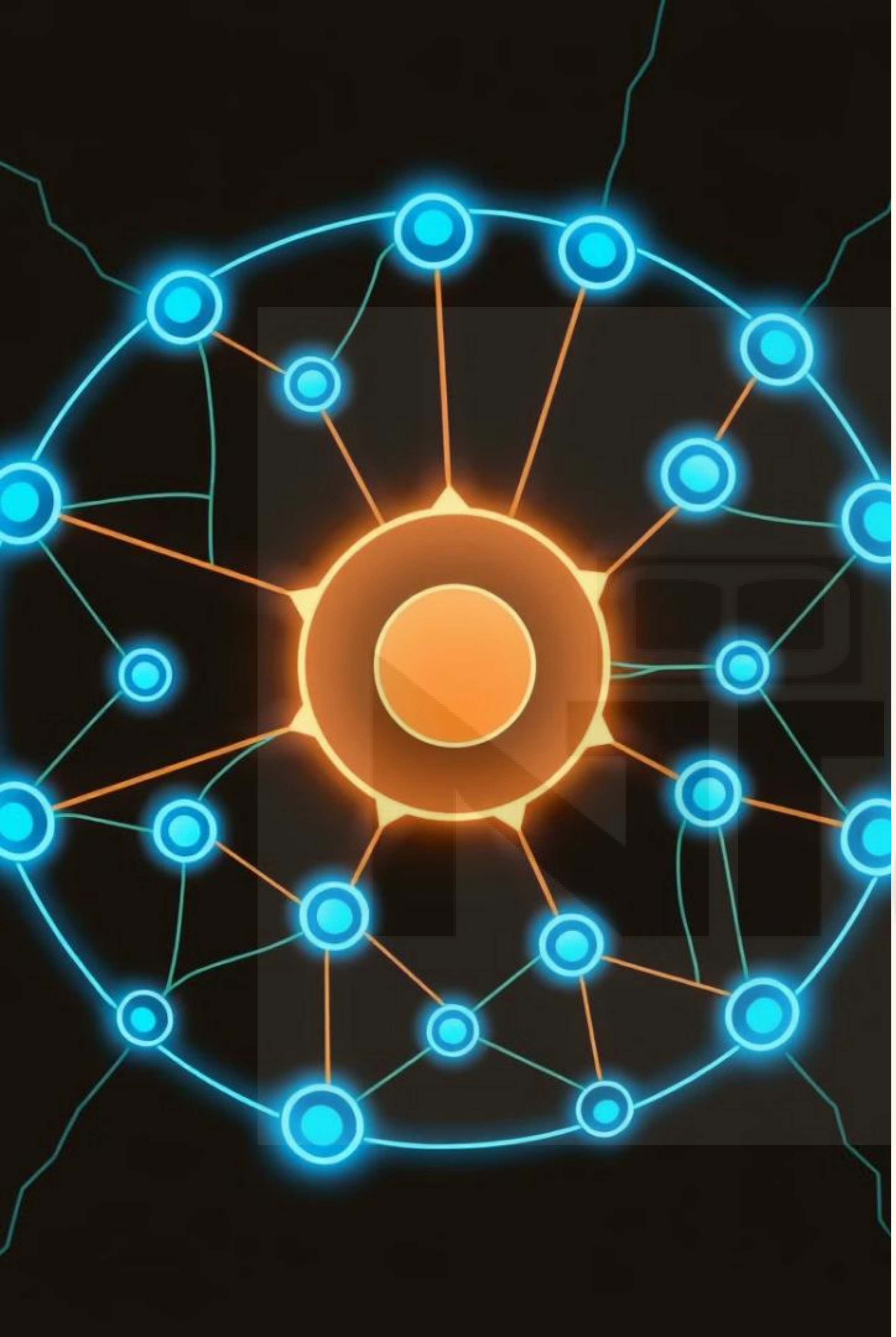
Assigns a unique integer to each new row.

```
CREATE TABLE Employees ( EmployeeID INT AUTO_INCREMENT
PRIMARY KEY, Name VARCHAR(255) );
```

Syntax varies by database system.

Comparison of Constraints

NOT NULL	Ensures no NULL values.	StudentID INT NOT NULL
UNIQUE	Ensures unique values.	ProductID INT UNIQUE
CHECK	Ensures data meets conditions.	Salary DECIMAL CHECK (Salary > 0)
DEFAULT	Sets default values.	Salary DECIMAL DEFAULT 50000.00
Auto Increment	Automatically assigns unique IDs.	EmployeeID INT AUTO_INCREMENT



Conclusion

Constraints are essential for data integrity.

Each type serves a specific purpose.

From unique values to default settings.