

All SQL Constraints (Most Important)

1 PRIMARY KEY

- Uniquely identifies each row in a table
- Cannot be NULL
- Only one primary key per table (can be composite)

Example:

```
CREATE TABLE students (  
    id INT PRIMARY KEY,  
    name VARCHAR(50)  
);
```

2 FOREIGN KEY

- Links two tables
- Ensures referenced value exists in parent table
- Maintains referential integrity

Example:

```
CREATE TABLE orders (  
    order_id INT PRIMARY KEY,  
    customer_id INT,  
    FOREIGN KEY (customer_id) REFERENCES customers(id)  
);
```

3 UNIQUE

- Ensures all values in a column are different
- Unlike PRIMARY KEY, a table can have multiple UNIQUE constraints

Example:

```
CREATE TABLE users (  
    email VARCHAR(100) UNIQUE  
);
```

4 NOT NULL

- Prevents a column from storing NULL values

Example:

```
CREATE TABLE employees (  
    name VARCHAR(50) NOT NULL  
);
```

5 CHECK

- Validates input based on a condition

Example:

```
CREATE TABLE products (  
    price DECIMAL(10,2),  
    CHECK (price > 0)  
);
```

6 DEFAULT

- Sets default value when no value is provided

Example:

```
CREATE TABLE accounts (  
    status VARCHAR(20) DEFAULT 'ACTIVE'  
);
```

7 AUTO_INCREMENT (MySQL) / SERIAL (PostgreSQL)

- Automatically generates unique numbers for each row

Example (MySQL):

```
CREATE TABLE orders (  
    order_id INT AUTO_INCREMENT PRIMARY KEY  
);
```

Example (PostgreSQL):

```
order_id SERIAL PRIMARY KEY
```

8 INDEX (Not exactly a constraint, but improves speed)

- Speeds up searching and sorting

Example:

```
CREATE INDEX idx_name ON students(name);
```

SUMMARY TABLE OF CONSTRAINTS

Constraint	Purpose
PRIMARY KEY	Unique + Not Null identifier
FOREIGN KEY	Maintain relationship between tables
UNIQUE	No duplicate values
NOT NULL	Cannot be NULL
CHECK	Adds condition validation
DEFAULT	Supplies default value
AUTO_INCREMENT / SERIAL	Auto-generated key
INDEX	Improves search speed