

# SQL Data Science for Beginners

## Creating Tables: CREATE TABLE

The `CREATE TABLE` statement defines a new table structure in your database.

### Basic Syntax:

```
CREATE TABLE table_name (  
    column1 datatype constraints,  
    column2 datatype constraints,  
    ...  
);
```

### Example 1: E-commerce Products Table

```
CREATE TABLE products (  
    product_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    price DECIMAL(10,2) NOT NULL,  
    stock_quantity INT DEFAULT 0,  
    added_date DATE NOT NULL,  
    is_active BOOLEAN DEFAULT TRUE  
);
```

### Example 2: Social Media Users Table

```
CREATE TABLE users (  
    user_id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) UNIQUE NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password_hash VARCHAR(255) NOT NULL,  
    join_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
    is_verified BOOLEAN DEFAULT FALSE  
);
```

# SQL Data Science for Beginners

## Example 3: Hospital Patients Table

```
CREATE TABLE patients (  
  patient_id INT AUTO_INCREMENT PRIMARY KEY,  
  first_name VARCHAR(50) NOT NULL,  
  last_name VARCHAR(50) NOT NULL,  
  birth_date DATE NOT NULL,  
  blood_type VARCHAR(3),  
  last_visit_date DATE,  
  has_insurance BOOLEAN DEFAULT FALSE  
);
```

## Data Types

### Numeric Types

1. **INT** - Whole numbers (-2,147,483,648 to 2,147,483,647)

**age INT NOT NULL**

2. **FLOAT/DOUBLE** - Floating-point numbers

**temperature FLOAT**

3. **DECIMAL** - Fixed-point numbers (for precise values like money)

**DECIMAL(10,2) -- 10 digits total, 2 after decimal**

### String Types

1. **VARCHAR(n)** - Variable-length strings (1-65,535 characters)

**username VARCHAR(50)**

2. **CHAR(n)** - Fixed-length strings (pads with spaces)

# SQL Data Science for Beginners

`country_code CHAR(2) -- Like 'US', 'CA'`

3. **TEXT** - Long text (up to 65,535 characters)

`article_content TEXT`

## Date/Time Types

1. **DATE** - Stores date only in the format YYYY-MM-DD

`birth_date DATE`

2. **DATETIME** - Stores date and time in the format YYYY-MM-DD HH:MM:SS

`created_at DATETIME DEFAULT CURRENT_TIMESTAMP`

3. **TIMESTAMP** - Stores a Unix timestamp and auto-updates on record change

`last_updated TIMESTAMP ON UPDATE CURRENT_TIMESTAMP`

## Boolean

1. **BOOLEAN/TINYINT(1)** - True/False values

`is_active BOOLEAN DEFAULT TRUE`

## Table Constraints

### PRIMARY KEY

Uniquely identifies each record in a table.

- Single column:  
`product_id INT PRIMARY KEY`

# SQL Data Science for Beginners

- Auto-incrementing primary key:

**order\_id INT AUTO\_INCREMENT PRIMARY KEY**

- Composite key (multiple columns):

**PRIMARY KEY (student\_id, course\_id)**

## AUTO\_INCREMENT

Automatically generates sequential numbers for a column.

- Simple auto-increment:

**customer\_id INT AUTO\_INCREMENT PRIMARY KEY**

- Starting from a specific number (MySQL syntax):

**invoice\_id INT AUTO\_INCREMENT PRIMARY KEY START WITH 1000**

- With specific increment:

**log\_id INT AUTO\_INCREMENT PRIMARY KEY INCREMENT BY 2**

## NOT NULL

Ensures a column cannot contain NULL values.

- Required name field:

**first\_name VARCHAR(50) NOT NULL**

- Mandatory date field:

**order\_date DATE NOT NULL**

# SQL Data Science for Beginners

- Essential numeric field:

**quantity INT NOT NULL**

## DEFAULT

Sets a default value if no value is provided during insert.

- Default quantity:

**stock INT DEFAULT 0**

- Default timestamp:

**created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP**

- Default status:

**status VARCHAR(20) DEFAULT 'pending'**

## UNIQUE

Ensures all values in a column (or combination) are different.

- Unique email:

**email VARCHAR(100) UNIQUE**

- Unique product code:

**sku VARCHAR(20) UNIQUE**

- Unique combination of columns:

**UNIQUE (department\_id, employee\_code)**

# SQL Data Science for Beginners

## Complete Real-World Examples

### Example 1: Online Banking System

```
CREATE TABLE accounts (  
  account_id INT AUTO_INCREMENT PRIMARY KEY,  
  account_number VARCHAR(20) UNIQUE NOT NULL,  
  customer_id INT NOT NULL,  
  account_type VARCHAR(20) NOT NULL,  
  balance DECIMAL(15,2) DEFAULT 0.00,  
  opened_date DATE NOT NULL,  
  is_active BOOLEAN DEFAULT TRUE,  
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

### Example 2: University Course Registration

```
CREATE TABLE courses (  
  course_id INT AUTO_INCREMENT PRIMARY KEY,  
  course_code VARCHAR(10) UNIQUE NOT NULL,  
  title VARCHAR(100) NOT NULL,  
  credits TINYINT NOT NULL,  
  department_id INT NOT NULL,  
  max_capacity INT DEFAULT 30,  
  is_offered BOOLEAN DEFAULT TRUE,  
  CONSTRAINT chk_credits CHECK (credits BETWEEN 1 AND 5)  
);
```

### Example 3: Inventory Management

```
CREATE TABLE inventory (  
  item_id INT AUTO_INCREMENT PRIMARY KEY,  
  sku VARCHAR(15) UNIQUE NOT NULL,  
  name VARCHAR(100) NOT NULL,  
  category VARCHAR(30) NOT NULL,  
  supplier_id INT NOT NULL,  
  unit_price DECIMAL(10,2) NOT NULL,  
  quantity_in_stock INT DEFAULT 0,
```

# SQL Data Science for Beginners

```
reorder_level INT DEFAULT 10,  
last_restocked DATE,  
FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id)  
);
```