

Using the CREATE DATABASE statement (SQL Client):

Syntax:

```
CREATE DATABASE database_name;
```

Optional:

```
CREATE DATABASE IF NOT EXISTS database_name;
```

Verify creation:

You can verify the database creation by using the **SHOW DATABASES;**

Creating a Table:

```
CREATE TABLE [IF NOT EXISTS] table_name (  
    column1_name datatype [constraints],  
    column2_name datatype [constraints],  
    ...  
    columnN_name datatype [constraints]  
);
```

Example:

```
CREATE TABLE Employees (  
    employee_id INT PRIMARY KEY ,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    hire_date DATE  
);
```

SQL Constraints:

NOT NULL: Ensures that a column cannot contain NULL values. Every row must have a value for a column with this constraint.

```
CREATE TABLE Persons (  
    ID INT NOT NULL,  
    LastName VARCHAR(255) NOT NULL  
);
```

UNIQUE: Ensures that all values in a column are unique. No two rows can have the same value in a column with this constraint.

```
CREATE TABLE Persons (  
    ID INT UNIQUE,  
    Email VARCHAR(255) UNIQUE  
);
```

PRIMARY KEY: Uniquely identifies each record in a table. A primary key must contain UNIQUE values and cannot contain NULL values. Each table can have only one primary key, which can consist of single or multiple columns.

```
CREATE TABLE Persons (  
    ID INT PRIMARY KEY,  
    LastName VARCHAR(255)  
);
```

FOREIGN KEY: Ensures referential integrity by linking data in one table to data in another table. It establishes a relationship between two tables, where the foreign key in one table references the primary key in another.

-- Parent Table

```
CREATE TABLE users (  
    user_id INT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE  
);
```

-- Child Table with Foreign Key

```
CREATE TABLE orders (  
    order_id INT PRIMARY KEY,  
    user_id INT,  
    order_date DATE,  
    total_amount DECIMAL(10, 2),  
    FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

CHECK: Limits the range of values that can be placed in a column. It can be defined on a single column to allow only certain values or on a table to limit values based on other columns in the same row.

```
CREATE TABLE Persons (  
    ID INT PRIMARY KEY,  
    Age INT CHECK (Age >= 18)  
);
```

DEFAULT: Provides a default value for a column when no value is explicitly specified during an INSERT operation.

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    OrderDate DATE DEFAULT CURRENT_DATE()  
);
```