# SQL INSERT INTO Statement ➕

The INSERT INTO statement is used to add new records (rows) to an existing table in a database.

## Basic Syntax

There are two main ways to use the INSERT INTO statement:

### 1. Specifying both column names and values:

This is the recommended approach as it makes your SQL more readable and less prone to errors if the table structure changes (e.g., if a new column is added).

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

* INSERT INTO table\_name: Specifies the table you want to insert data into.
* (column1, column2, ...): (Optional but recommended) A list of the columns you are inserting data into. If you omit this list, you must provide values for all columns in the table, in their defined order.
* VALUES (value1, value2, ...): The values to be inserted, corresponding to the columns listed (or all columns if no list is provided). String values and dates should be enclosed in single quotes (' ').

### 2. Specifying only values (for all columns):

This method requires you to provide values for every column in the table, in the exact order they were defined when the table was created. It's less flexible and generally not recommended for long-term use.

INSERT INTO table\_name

VALUES (value1, value2, value3, ...);

## Example 1: Inserting Data into the **Customers** Table

Let's insert some data into the Customers table we created earlier. Remember its structure:

Customers (CustomerID INT PRIMARY KEY AUTO\_INCREMENT, FirstName VARCHAR(50) NOT NULL, LastName VARCHAR(50) NOT NULL, Email VARCHAR(100) UNIQUE)

-- Inserting a new customer, letting AUTO\_INCREMENT handle CustomerID

INSERT INTO Customers (FirstName, LastName, Email)

VALUES ('Alice', 'Smith', 'alice.smith@example.com');

-- Inserting another customer without an email

INSERT INTO Customers (FirstName, LastName)

VALUES ('Bob', 'Johnson');

-- Inserting a third customer with all explicit values (CustomerID will be ignored if AUTO\_INCREMENT is set)

INSERT INTO Customers (CustomerID, FirstName, LastName, Email)

VALUES (101, 'Charlie', 'Brown', 'charlie.b@example.com');

**Explanation:**

* The first INSERT statement provides values for FirstName, LastName, and Email. The CustomerID is automatically generated because of AUTO\_INCREMENT.
* The second INSERT statement only provides FirstName and LastName. Since Email allows NULL values (it only has a UNIQUE constraint, not NOT NULL), it's perfectly valid to omit it.
* The third INSERT statement explicitly tries to set CustomerID to 101. In most SQL databases with AUTO\_INCREMENT, this value will be ignored, and the database will still assign the next available auto-incremented ID. If AUTO\_INCREMENT wasn't present, 101 would be used.

## Example 2: Inserting Data into the **Products** Table

Now, let's add some products to our Products table. Recall its structure:

Products (ProductID INT PRIMARY KEY AUTO\_INCREMENT, ProductName VARCHAR(255) NOT NULL UNIQUE, Description TEXT, Price DECIMAL(10, 2) NOT NULL CHECK (Price > 0), StockQuantity INT DEFAULT 0, DateAdded DATE DEFAULT CURRENT\_DATE)

-- Inserting a product with all required details

INSERT INTO Products (ProductName, Description, Price, StockQuantity)

VALUES ('Laptop Pro', 'High-performance laptop with 16GB RAM', 1200.00, 50);

-- Inserting a product with only mandatory fields; StockQuantity and DateAdded will use defaults

INSERT INTO Products (ProductName, Price)

VALUES ('Wireless Mouse', 25.50);

-- Inserting a product with a specific date and stock quantity

INSERT INTO Products (ProductName, Description, Price, StockQuantity, DateAdded)

VALUES ('Mechanical Keyboard', 'RGB backlit, tactile switches', 99.99, 20, '2024-07-15');

**Explanation:**

* The first INSERT provides values for ProductName, Description, Price, and StockQuantity. ProductID and DateAdded will be handled by AUTO\_INCREMENT and DEFAULT CURRENT\_DATE respectively.
* The second INSERT only provides ProductName and Price (the NOT NULL columns). Description will be NULL, StockQuantity will default to 0, and DateAdded will default to the current date.
* The third INSERT explicitly sets all values, including DateAdded, overriding its default behavior.