Add New Table

Tables are the foundation of your database design. Each table represents a collection of related data, like users, products, or orders.

What is a Table?

A table is like a spreadsheet with:

• **Rows:** Individual records (each user, product, order)

• **Columns:** Fields (name, email, price)

• **Primary Key:** Unique identifier for each row

How to Create a Table

Step 1: Open Create Table Dialog

Click the [Create Table] button in the toolbar OR press Ctrl+N

Step 2: Enter Table Name

Use lowercase, meaningful names without spaces.

Good table names:

• users

• products

• orders

• customers

• inventory

Bad table names:

• table1

• data

• tbl_user

• USERS (use lowercase)

• user data (no spaces)

Step 3: Specify Number of Columns

Enter how many fields this table needs. You can have 1 to 50 columns.

Examples:

• users table: 4 columns (id, username, email, created_at)

• products table: 5 columns (id, name, price, stock, description)

• orders table: 4 columns (id, user_id, total, date)

Step 4: Define Each Column

For each column, enter: Column Name, Data Type (INT, VARCHAR, DECIMAL, etc.), and Constraints (Primary Key, Auto Increment, NOT NULL, UNIQUE)

Complete Example: Creating a Users Table

Table Name: users

Columns: 4

Column 1: id

• Type: INT

• Primary Key: Yes

• Auto Increment: Yes

Column 2: username

• Type: VARCHAR(255)

• Unique: Yes

• NOT NULL

Column 3: email

• Type: VARCHAR(255)

• Unique: Yes

• NOT NULL

Column 4: created_at

• Type: TIMESTAMP

• NOT NULL

Result on Canvas

After creating the users table, a blue box appears on the canvas showing all columns with their types. A gold circle indicates the Primary Key (id).

Important Tips

Always make the first column a Primary Key

• Data type: INT

• Enable Auto Increment

• This creates unique identifiers for each row

Use clear, descriptive names

• first_name (not fname or f_name)

- user_id (not uid or user)
- created_at (not date or d)

Don't use special characters or spaces

- Good: user_address
- Bad: user address (space)
- Bad: user-address (hyphen)

Manage Columns

Columns define what data your table stores. Each column has a name, type, and optional constraints.

Data Types Reference

INT (Integer)

Used for: Whole numbers

Examples: age, count, id, quantity, stock

VARCHAR(n)

Used for: Text strings

Examples: name, email, username, address

DECIMAL(m,n)

Used for: Decimal numbers

Examples: price, rating, percentage

TIMESTAMP

Used for: Date and Time

Examples: created_at, updated_at, deleted_at

BOOLEAN

Used for: True/False values

Examples: is_active, is_deleted, is_verified

TEXT

Used for: Long text

Examples: description, comment, bio, content

Column Constraints

Constraints are rules that control what data can be stored in a column.

Primary Key (PK)

- Uniquely identifies each row in the table
- Usually placed on first column named 'id'
- Value must be unique and not NULL
- Usually paired with Auto Increment
- Only one Primary Key per table allowed

Auto Increment

- Automatically generates sequential numbers (1, 2, 3...)
- Used with INT data type and Primary Key
- Ensures every row has unique identifier
- You cannot manually set auto increment values

NOT NULL

- Field must always have a value
- Cannot be left empty
- Use for required fields like username, email
- Leave unchecked if field can be optional

UNIQUE

- No duplicate values allowed
- Used for username, email, phone number
- Different from Primary Key (PK is unique AND identifies row)
- Multiple rows can be NULL if not checked

Column Design Patterns

User ID Column

• Name: id : INT

• Primary Key: Yes

• Auto Increment: Yes

Email Column

• Name: email : VARCHAR(255)

• Nullable: No

• Unique: Yes

Username Column

• Name: username : VARCHAR(255)

• Nullable: No

• Unique: Yes

Price Column

• Name: price : DECIMAL(10,2)

• Nullable: No

• Unique: No

Date Column

• Name: created_at : TIMESTAMP

• Nullable: No

• Unique: No

Good vs Bad Column Design

Good Column Design

✓ Column names are lowercase: first_name

✓ Uses underscores for spacing: created_at

✓ Appropriate data types used

✓ Required fields marked NOT NULL

✓ Unique fields have UNIQUE constraint

✓ First column is always ID (INT, PK, Auto Increment)

Bad Column Design

✗ Column names with spaces: 'First Name'

✗ Uppercase column names: FirstName

✗ Using VARCHAR for everything

✗ No constraints specified

✗ Missing Primary Key

✗ Wrong data types (VARCHAR for numbers)

Naming Conventions

What to do

✓ Use lowercase: user, not User

✓ Use underscores: first_name, not firstName

✓ Use full words: created_at, not c_at

✓ Use consistent style throughout

✓ Be descriptive: user_address, not addr

What to avoid

✗ Spaces: 'first name'

✗ Special characters: first-name, first@name

✗ Abbreviations: fname instead of first_name

✗ Mixed case: firstName

✗ Numbers at start: 1name