Generate SQL Code

Convert your visual database design into SQL code that you can run on any database system.

What is SQL?

SQL (Structured Query Language) is the standard language for databases. It works with:

• MySQL

• PostgreSQL

• SQL Server

• SQLite

• Oracle

• And many others

How to Generate SQL

Step 1: Click Generate SQL Button

Click the [SQL] button in toolbar OR press Ctrl+G

A dialog opens showing the generated SQL code.

Step 2: Review the Code

Read through the SQL to verify it matches your design. Check for:

• All tables are included

• All columns with correct types

• All relations with correct ON DELETE actions

• Foreign keys properly defined

Step 3: Copy to Clipboard

Click [COPY] button to copy all SQL to clipboard. Button changes to "Copied!" for 2 seconds to confirm.

Step 4: Paste in Database Tool

Open your database application:

• MySQL Workbench

• pgAdmin (PostgreSQL)

• SQL Server Management Studio

• Or any other database tool

Paste the SQL code and execute it.

What SQL is Generated

CREATE TABLE Statements

For each table, CREATE TABLE statement is generated:

```
CREATE TABLE users ( id INT PRIMARY KEY AUTO_INCREMENT, username VARCHAR(255) NOT NULL UNIQUE, email VARCHAR(255) NOT NULL UNIQUE, created_at TIMESTAMP NOT NULL );
```

This creates:

• Table named users

• Column id as primary key

• Auto increment enabled

• Columns with proper types

• NOT NULL constraints applied

• UNIQUE constraints applied

ALTER TABLE for Foreign Keys

After all tables are created, relations are added:

```
ALTER TABLE orders ADD CONSTRAINT fk_users_user_id FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE;
```

This creates:

• Foreign key named fk_users_user_id

• user_id column references users.id

• ON DELETE CASCADE action specified

Complete Example: E-Commerce Database

Tables Created

1. users table
2. products table
3. orders table

Generated SQL

```
CREATE TABLE users ( id INT PRIMARY KEY AUTO_INCREMENT, username VARCHAR(255) NOT NULL UNIQUE, email VARCHAR(255) NOT NULL UNIQUE ); CREATE TABLE products ( id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255) NOT NULL, price DECIMAL(10,2) NOT NULL, stock INT NOT NULL ); CREATE TABLE orders ( id INT PRIMARY KEY AUTO_INCREMENT, user_id INT NOT NULL, total DECIMAL(10,2) NOT NULL ); ALTER TABLE orders ADD CONSTRAINT fk_users_user_id FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE;
```

Using Generated SQL

Option 1: MySQL Workbench

1. Open MySQL Workbench
2. Connect to your database
3. Open new SQL script
4. Paste the SQL code
5. Click Execute (or Ctrl+Shift+Enter)

Option 2: Command Line

1. Save SQL to file: database.sql
2. Open terminal
3. Run: mysql -u username -p < database.sql
4. Enter password when prompted

Option 3: SQL Server Management Studio

1. Open SSMS
2. Connect to server
3. Open new query
4. Paste SQL code
5. Click Execute (or F5)

Option 4: PostgreSQL (pgAdmin)

1. Open pgAdmin

2. Right-click database -> Query Tool
3. Paste SQL code
4. Click Execute (or F5)

SQL Updates Automatically

When you delete tables or relations, the SQL code updates automatically:

• Deleted table CREATE statement removed

• Deleted relation ALTER statement removed

• References to deleted tables removed

Saving SQL Code

Option 1: Save as File

1. Copy SQL from dialog
2. Open text editor
3. Paste SQL
4. Save as: database.sql
5. Keep safe for later use

Option 2: Copy and Paste Directly

1. Click COPY in dialog
2. Go to your database tool
3. Right-click and Paste
4. Execute the SQL

Option 3: Email or Share

1. Copy SQL code
2. Email to team member
3. They can paste into their database tool
4. Both have same schema

Important Notes

Database Compatibility

The generated SQL uses standard SQL commands that work on most databases. However, some syntax may differ slightly between databases. Review the SQL before executing.

Always Backup First

Before executing SQL on a production database, always backup your data. Test on a test database first.

Review Before Executing

Always review the generated SQL code to ensure it matches your database design. Check all table names, column names, types, and constraints.