

# import an SQL file

While there is no built-in

`php artisan` command to directly import an SQL file, you can achieve this by either:

- Creating a **custom Artisan command** that executes the SQL file.
- Using the standard **MySQL command-line interface (CLI)** tool directly.

## Method 1: Using a Custom Artisan Command (Recommended for Laravel Integration)

This method allows you to import the SQL file as part of your application's deployment or seeding process using a simple `php artisan` command.

### 1. Create a new Artisan command:

bash

2. `php artisan make:command ImportSqlFile`

This creates `app/Console/Commands/ImportSqlFile.php`.

### 3. Edit the `handle` method in the new file:

Place your SQL file in a directory (e.g., `database/sql/dump.sql`) and use the `DB` facade to run the raw SQL content.

php

```
4. <?php

namespace App\Console\Commands;

use Illuminate\Console\Command;
use Illuminate\Support\Facades\DB;

class ImportSqlFile extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
```

```

protected $signature = 'db:import-sql'; // Define your command name

/**
 * The console command description.
 *
 * @var string
 */
protected $description = 'Import a specific SQL file into the database';

/**
 * Execute the console command.
 *
 * @return int
 */
public function handle()
{
    $path = database_path('sql/dump.sql'); // Adjust the path as necessary

    if (!file_exists($path)) {
        $this->error('SQL file not found at: ' . $path);
        return Command::FAILURE;
    }

    $this->info('Starting SQL file import...');

    // Execute the raw SQL content
    DB::unprepared(file_get_contents($path));

    $this->info('Database import successful.');
```

```

    return Command::SUCCESS;
}
}

```

## 5. Run the command:

bash

```
php artisan db:import-sql
```

**Note:** For very large SQL files, directly using `DB::unprepared()` might hit PHP memory or execution time limits. In such cases, the MySQL CLI method (Method 2) is more robust.

## Method 2: Using the MySQL Command-Line Interface

You can run the `mysql` client directly from your terminal, which is the most reliable way to import large SQL dumps.

Open your terminal, navigate to your project root (or the directory containing the `.sql` file), and run the following command.

bash

```
mysql -u [username] -p [database_name] < /path/to/yourfile.sql
```

### Replace the placeholders:

- `[username]` with your MySQL username (e.g., `root`).
- `[database_name]` with the name of your database as defined in your Laravel `.env` file.
- `/path/to/yourfile.sql` with the actual path to your SQL dump file.

When prompted, enter your database password. The process will then import the file directly into your database

---

Revision #1

Created 14 December 2025 14:31:24 by AI Channel

Updated 14 December 2025 14:32:09 by AI Channel