**Step 1:**

**Import the database**

**Then install the package through composer command**

**Then run**

php artisan migrate:generate --tables="table1,table2,table3,table4,table5"

**it will ask a option of log do “no” it will create all schema migration everything**

[**Laravel Migrations Generator**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#laravel-migrations-generator)

Generate Laravel Migrations from an existing database, including indexes and foreign keys!

This package is a modified version of <https://github.com/Xethron/migrations-generator> that has been updated to support Laravel 5.6 and beyond, along with additional features.

[**Supported Database**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#supported-database)

Laravel Migrations Generator supports all five Laravel first-party support databases:

*  MariaDB
*  MySQL
*  PostgreSQL
*  SQL Server
*  SQLite

[**Version Compatibility**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#version-compatibility)

| **Laravel** | **Version** |
| --- | --- |
| 10.x | 6.x |
| 9.x | 6.x |
| 8.x | 6.x |
| 7.x | 6.x |
| 6.x | 6.x |
| 5.8.x | 6.x |
| 5.7.x | 6.x |
| 5.6.x | 6.x |
| 5.5 and below | <https://github.com/Xethron/migrations-generator> |

[**Install**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#install)

The recommended way to install this is through composer:

composer require --dev kitloong/laravel-migrations-generator

[**Laravel Setup**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#laravel-setup)

Laravel will automatically register service provider for you.

[**Lumen Setup**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#lumen-setup)

Expand

[**Usage**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#usage)

To generate migrations from a database, you need to have your database setup in Laravel's config (config/database.php).

To create migrations for all the tables, run:

php artisan migrate:generate

You can specify the tables you wish to generate using:

php artisan migrate:generate --tables="table1,table2,table3,table4,table5"

You can also ignore tables with:

php artisan migrate:generate --ignore="table3,table4,table5"

Laravel Migrations Generator will first generate all the tables, columns and indexes, and afterwards setup all the foreign key constraints.

So make sure you include all the tables listed in the foreign keys so that they are present when the foreign keys are created.

You can also specify the connection name if you are not using your default connection with:

php artisan migrate:generate --connection="connection\_name"

[**Squash Migrations**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#squash-migrations)

By default, Generator will generate multiple migration files for each table.

You can squash all migrations into a single file with:

php artisan migrate:generate --squash

[**Options**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#options)

Run php artisan help migrate:generate for a list of options.

| **Options** | **Description** |
| --- | --- |
| -c, --connection[=CONNECTION] | The database connection to use |
| -t, --tables[=TABLES] | A list of tables or views you wish to generate migrations for separated by a comma: users,posts,comments |
| -i, --ignore[=IGNORE] | A list of tables or views you wish to ignore, separated by a comma: users,posts,comments |
| -p, --path[=PATH] | Where should the file be created? |
| -tp, --template-path[=TEMPLATE-PATH] | The location of the template for this generator |
| --date[=DATE] | Migrations will be created with specified date. Views and foreign keys will be created with + 1 second. Date should be in format supported by Carbon::parse |
| --table-filename[=TABLE-FILENAME] | Define table migration filename, default pattern: [datetime]\\_create\_[name]\_table.php |
| --view-filename[=VIEW-FILENAME] | Define view migration filename, default pattern: [datetime]\\_create\_[name]\_view.php |
| --proc-filename[=PROC-FILENAME] | Define stored procedure filename, default pattern: [datetime]\\_create\_[name]\_proc.php |
| --fk-filename[=FK-FILENAME] | Define foreign key migration filename, default pattern: [datetime]\\_add\_foreign\_keys\_to\_[name]\_table.php |
| --log-with-batch[=LOG-WITH-BATCH] | Log migrations with given batch number. We recommend using batch number 0 so that it becomes the first migration |
| --default-index-names | Don't use DB index names for migrations |
| --default-fk-names | Don't use DB foreign key names for migrations |
| --use-db-collation | Generate migrations with existing DB collation |
| --skip-log | Don't log into migrations table |
| --skip-views | Don't generate views |
| --skip-proc | Don't generate stored procedures |
| --squash | Generate all migrations into a single file |
| --with-has-table | Check for the existence of a table using hasTable |

[**SQLite Alter Foreign Key**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#sqlite-alter-foreign-key)

The generator first generates all tables and then adds foreign keys to existing tables.

However, SQLite only supports foreign keys upon creation of the table and not when tables are altered. *add\_foreign\_keys* migrations will still be generated, however will get omitted if migrate to SQLite type database.

[**User Defined Custom Column Type**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#user-defined-custom-column-type)

The generator will register custom data type from the schema, then generate migration as

public function up()

{

Schema::create('table', function (Blueprint $table) {

...

});

DB::statement("ALTER TABLE table ADD column custom\_type NOT NULL");

}

Note that the new column is always added at the end of the created table which means the ordering of the column generated in migration will differ from what we have from the schema.

Supported database with custom types:

*  PostgreSQL
*  SQL Server

[**Thank You**](https://github.com/kitloong/laravel-migrations-generator/tree/6.x#thank-you)

Thanks to Bernhard Breytenbach for his great work. This package is based on <https://github.com/Xethron/migrations-generator>.