

Laravel Database: Migrations and Seeding

ToC

Migrations

- What are Migrations?
- Structure
- Commands
- Table/Column/Index modification
 - Create, Update, Rename, Drop
- Live Demonstration

Seeding

- What is Seeding?
- Structure
- Commands
- Live Demonstration

Questions

Migrations

What are Migrations?

- Version control for a database
- Define/Share an app's database schema definition
- Create and modify database:
 - Add/update/rename/drop tables/columns/indexes

Structure

File name format:

YYYY_MM_DD_HHMMSS_descriptive_migration_name

Stored under database\migrations

up():

- add/update/rename/drop tables/columns/indexes

down():

- reverse operations performed in up()

Commands

Create a migration:

```
php artisan make:migration name_of_migration
```

Install migrations table:

```
php artisan migrate:install
```

Run migrations:

```
php artisan migrate
```

Commands

Rollback last migration operations:

```
php artisan migrate:rollback  
php artisan migrate:rollback --step=3
```

Rollback all migrations:

```
php artisan migrate:reset
```

Commands

Reset && rerun migrations:

```
php artisan migrate:refresh
```

Drop tables && rerun migrations:

```
php artisan migrate:fresh
```


Commands

Which migrations have been run so far?:

```
php artisan migrate:status
```

Table, Column, Index modification

Commands

Database manipulation via:

- Schema Facade
- Blueprint class

Create

```
//table
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email');
    $table->timestamps();
});
```

Create

//column

```
Schema::table('users', function (Blueprint $table) {  
    $table->integer('votes');  
});
```

//index

```
Schema::table('users', function (Blueprint $table) {  
    $table->string('email')->unique();  
});
```

Update

```
//column  
Schema::table('users', function (Blueprint $table) {  
    $table->string('name', 50)->change();  
});
```

Rename

```
//table
```

```
Schema::rename($from, $to);
```

```
//column
```

```
Schema::table('users', function (Blueprint $table) {  
    $table->renameColumn('from', 'to');  
});
```

Rename

```
//index  
$table->renameIndex('from', 'to')
```


Drop

//table

```
Schema::drop('users');
```

```
Schema::dropIfExists('users');
```

//column

```
Schema::table('users', function (Blueprint $table) {
```

```
    $table->dropColumn('votes');
```

```
});
```

Drop

```
//index  
$table->dropUnique('users_email_unique');  
$table->dropPrimary('users_id_primary');
```

Live Demonstration

Seeding

What is Seeding?

- Insert (random) data into database tables

Structure

Only one method: `run()`

Insert data however you like:

- Query builder: insert records manually
- Model factories: generate & insert lots of records

Custom seeders under `database\seeders`

Default seeder class `Database\Seeders\DatabaseSeeder`:

- `run()`: call multiple seeder classes

Commands

Create seeder class:

```
php artisan make:seeder SeederClassName
```

Seed a database:

```
php artisan db:seed  
php artisan db:seed --class=SeederClassName
```

Commands

(Re)fresh and seed a database:

```
php artisan migrate:fresh --seed  
php artisan migrate:refresh --seed
```


Live Demonstration

Questions

Sources

Laravel 9.x Documentation:

- laravel.com/docs/9.x/migrations
- laravel.com/docs/9.x/seeding