In the context of Laravel, "seeders" refer to classes that are used to populate the database with initial data. This is particularly useful for testing purposes or for providing some default data when you first deploy your application.

Seeders are typically used in combination with Laravel's database migrations. Migrations allow you to define the structure of your database schema in code, and seeders allow you to populate it with data.

Here's a basic overview of how seeders work in Laravel:

1. \*\*Create a Seeder:\*\*

You can create a seeder using the following Artisan command:

```bash

php artisan make:seeder NameOfSeeder

```

This will generate a new seeder class in the `database/seeders` directory.

2. \*\*Define the Data:\*\*

Open the generated seeder file and use it to define the data you want to insert into your database. You'll typically use database queries (e.g., Eloquent or raw SQL) to accomplish this.

For example, you might create a seeder to populate a `users` table with some default users.

3. \*\*Run the Seeder:\*\*

You can run the seeder using the following Artisan command:

```bash

php artisan db:seed --class=NameOfSeeder

```

If you want to run all seeders, you can use:

```bash

php artisan db:seed

```

4. \*\*Automatic Seeding with Migrations:\*\*

You can also specify which seeders should be run after running a migration. This is done by using the `--seed` option with the `migrate` command:

```bash

php artisan migrate --seed

```

This will run all outstanding migrations and then execute all the seeders.

5. \*\*Database Seeding Classes:\*\*

In Laravel, database seeding classes are stored in the `database/seeders` directory. By default, Laravel provides a `DatabaseSeeder` class which you can use to call other seeders.

You can run this seeder using:

```bash

php artisan db:seed

```

Remember that seeders are mainly intended for development and testing purposes. In a production environment, you would typically use a database migration to structure your database and then use other means (e.g., manually entering data, using an admin interface) to populate it.

Keep in mind that as of my last training data in September 2021, this information is accurate. If there have been changes or updates to Laravel after that date, I would not be aware of them.