

Module-16 Laravel Migration Concepts

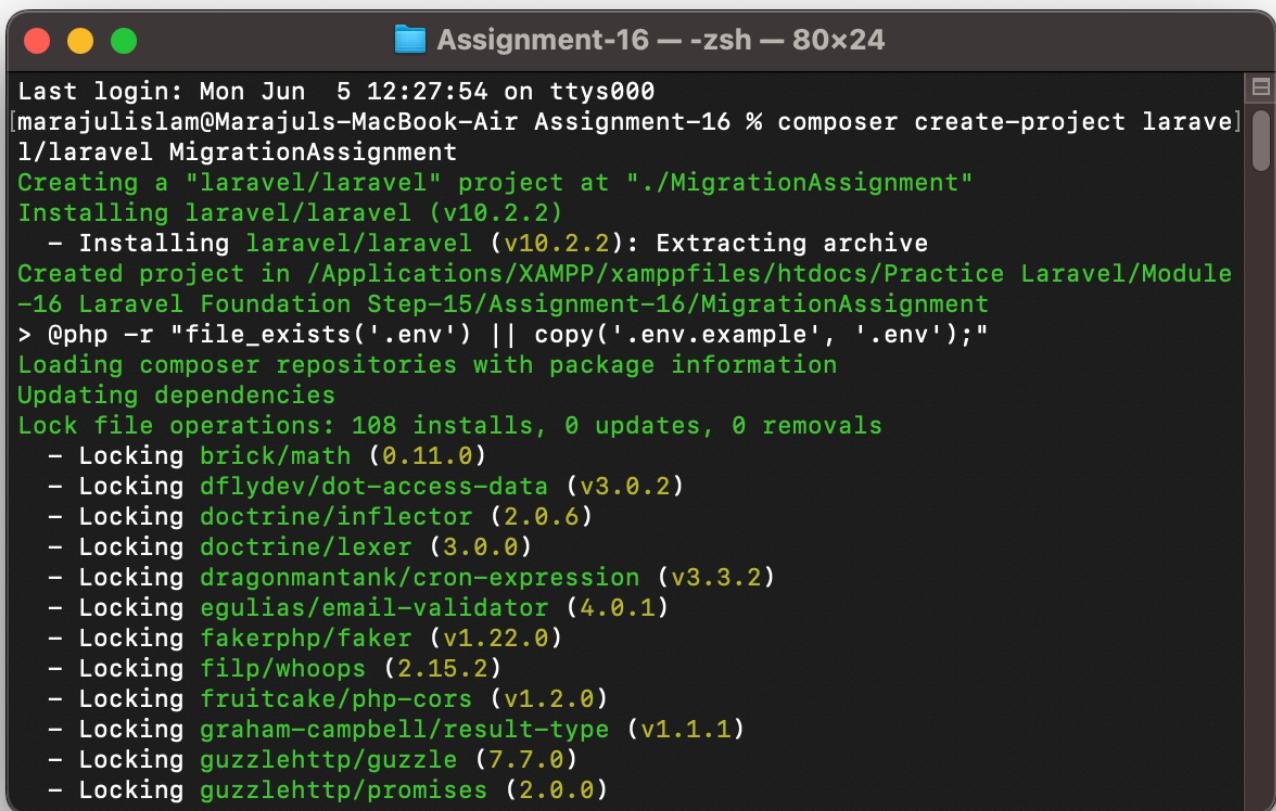
Assignment-16:

Task 1: Create a new Laravel project named "MigrationAssignment" using the Laravel command-line interface.

Step 1: Create a new Laravel Project

1. Open Terminal.
2. Run the following command:

```
composer create-project laravel/laravel MigrationAssignment
```



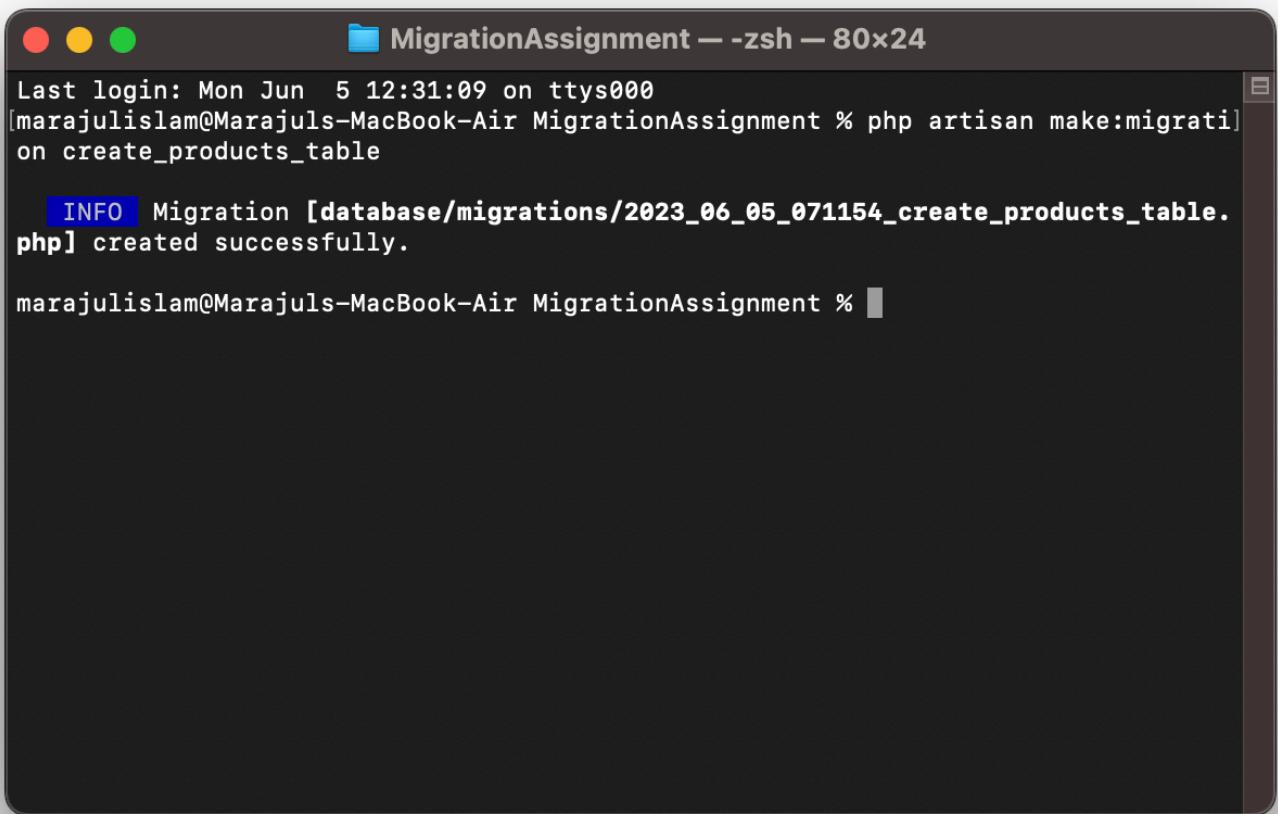
```
Last login: Mon Jun  5 12:27:54 on ttys000
[marajulislam@Marajuls-MacBook-Air Assignment-16 % composer create-project larave]
l/laravel MigrationAssignment
Creating a "laravel/laravel" project at "./MigrationAssignment"
Installing laravel/laravel (v10.2.2)
- Installing laravel/laravel (v10.2.2): Extracting archive
Created project in /Applications/XAMPP/xamppfiles/htdocs/Practice Laravel/Module
-16 Laravel Foundation Step-15/Assignment-16/MigrationAssignment
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 108 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (4.0.1)
- Locking fakerphp/faker (v1.22.0)
- Locking filp/whoops (2.15.2)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.1)
- Locking guzzlehttp/guzzle (7.7.0)
- Locking guzzlehttp/promises (2.0.0)
```

Task 2: Within the project, create a new migration file named "create_products_table" that will be responsible for creating a table called "products" in the database. The "products" table should have the following columns:

Step 1: Create a new migration file

1. Open Terminal.
2. Run the following command:

```
php artisan make:migration create_products_table
```



```
Last login: Mon Jun  5 12:31:09 on ttys000
[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php artisan make:migration
on create_products_table

  INFO  Migration [database/migrations/2023_06_05_071154_create_products_table.
php] created successfully.

marajulislam@Marajuls-MacBook-Air MigrationAssignment %
```

Step 2: Write needed columns in products table

1. Open products table.

The screenshot shows a code editor window with a dark theme. The title bar reads "2023_06_05_071154_create_products_table.php — Practice Laravel". The code editor displays a PHP migration file named "2023_06_05_071154_create_products_table.php". The file contains the following code:

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('products', function (Blueprint $table) {
15             $table->id();
16             $table->string('name',200);
17             $table->decimal('price');
18             $table->text('description');
19
20             $table->timestamp('created_at')->useCurrent();
21             $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('products');
31     }
32 };
33
```

The status bar at the bottom of the editor shows "Ln 19, Col 1 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⚡ Prettier".

Task 3: After creating the migration file, run the migration to create the "products" table in the database.

Step 1: Migrate the 'products' table.

1. Open Terminal.
2. Run the following command:

```
php artisan migrate
```

```

MigrationAssignment — zsh — 80x24
Last login: Mon Jun  5 13:11:50 on ttys000
[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php artisan migrate

INFO Preparing database.

Creating migration table ..... 33ms DONE

INFO Running migrations.

2014_10_12_000000_create_users_table ..... 18ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 18ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 13ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 18ms DONE
2023_06_05_071154_create_products_table ..... 7ms DONE

marajulislam@Marajuls-MacBook-Air MigrationAssignment %

```

3. Database:

The screenshot shows the phpMyAdmin interface running on a local host. The left sidebar displays a tree view of databases and tables, including 'Assignment-16' which contains tables like 'failed_jobs', 'migrations', 'password_reset_tokens', 'personal_access_tokens', and 'products'. The main panel shows a query results page for the 'products' table. The query executed was:

```

SELECT * FROM `products`

```

The results table shows the following data:

	id	name	price	description	created_at	updated_at

Below the table, there are buttons for 'Query results operations' such as 'Create view' and 'Bookmark this SQL query'. A message at the top of the main area states: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)".

Task 4: Modify the existing migration file "create_products_table" to add a new column called "quantity" to the "products" table. The "quantity" column should be an integer column and allow null values.

Step 1: RollBack the migration file

1. Open Terminal.
2. Run the following command:

```
php artisan migration:rollback
```

The screenshot shows a terminal window with the title bar 'MigrationAssignment -- zsh -- 80x24'. The terminal output is as follows:

```
Last login: Mon Jun  5 13:53:09 on ttys000
[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php artisan migrate:roll]
back

INFO Rolling back migrations.

2023_06_05_071154_create_products_table ..... 12ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 2ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 1ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 1ms DONE
2014_10_12_000000_create_users_table ..... 1ms DONE

[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php aritsan migrate
Could not open input file: aritsan
[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php artisan migrate

INFO Running migrations.

2014_10_12_000000_create_users_table ..... 27ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 18ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 12ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 15ms DONE
2023_06_05_071154_create_products_table ..... 7ms DONE
```

Step 2: Add a new coloums in products table:

1. Open products table.

2023_06_05_071154_create_products_table.php — Practice Laravel

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  You, 16 minutes ago | 1 author (You)
8  return new class extends Migration
9  {
10
11     /**
12      * Run the migrations.
13      */
14     public function up(): void
15     {
16         Schema::create('products', function (Blueprint $table) {
17             $table->id();
18             $table->string('name',200);
19             $table->decimal('price');
20             $table->text('description');
21             //Modify:
22             $table->integer('quantity')->nullable(); You, now + Uncommitted changes
23
24             $table->timestamp('created_at')->useCurrent();
25             $table->timestamp('updated_at')->useCurrent()>useCurrentOnUpdate();
26         });
27
28     /**
29      * Reverse the migrations.
30      */
31     public function down(): void
32     {
33         Schema::dropIfExists('products');
34     }
35 };

```

You, now Ln 20, Col 13 Spaces: 4 UTF-8 LF PHP Go Live Prettier

3. Updated Database:

The screenshot shows the phpMyAdmin interface for the 'Assignment-16' database. The left sidebar lists various databases and tables. The main area displays the 'products' table structure. The table has seven columns: id, name, price, description, quantity, created_at, and updated_at. The 'id' column is defined as a bigint(20) with an AUTO_INCREMENT attribute. The 'name' column is a varchar(200) using utf8mb4_unicode_ci collation. The 'price' column is a decimal(8,2). The 'description' column is a text type using utf8mb4_unicode_ci collation. The 'quantity' column is an int(11) with a NULL default value. The 'created_at' and 'updated_at' columns are timestamp types with current_timestamp() defaults.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	<i>None</i>		AUTO_INCREMENT	<i>Change</i> <i>Drop</i> <i>More</i>
2	name	varchar(200)	utf8mb4_unicode_ci		No	<i>None</i>			<i>Change</i> <i>Drop</i> <i>More</i>
3	price	decimal(8,2)			No	<i>None</i>			<i>Change</i> <i>Drop</i> <i>More</i>
4	description	text	utf8mb4_unicode_ci		No	<i>None</i>			<i>Change</i> <i>Drop</i> <i>More</i>
5	quantity	int(11)			Yes	<i>NULL</i>			<i>Change</i> <i>Drop</i> <i>More</i>
6	created_at	timestamp			No	current_timestamp()			<i>Change</i> <i>Drop</i> <i>More</i>
7	updated_at	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		<i>Change</i> <i>Drop</i> <i>More</i>

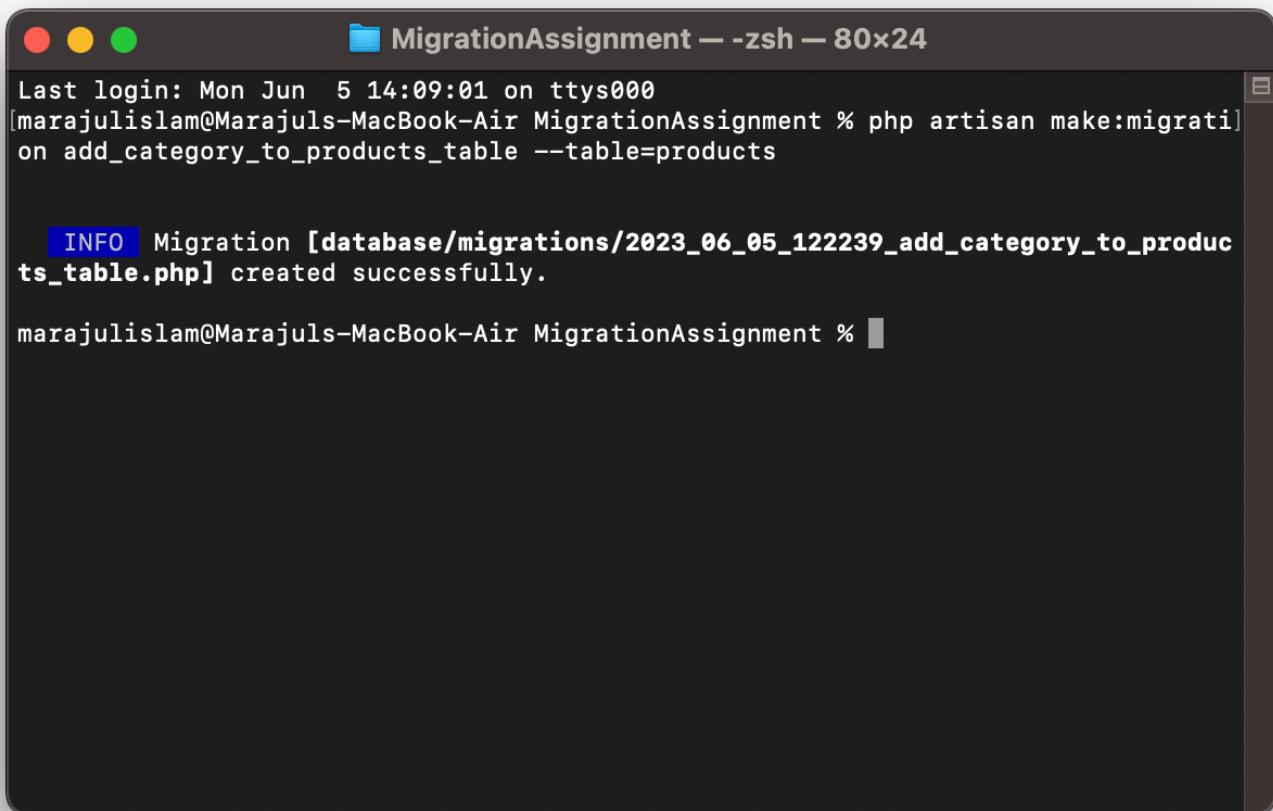
Task 5: Create a new migration file named 'add_category_to_products_table' that will be responsible for adding a new column called "category" to the "products" table.

The "category" column should be a string column with a maximum length of 50 characters:

Step 1: Create a new migration file

1. Open Terminal.
2. Run the following command:

```
php artisan make:migration add_category_to_products_table --table=products
```



The screenshot shows a terminal window with the title bar 'MigrationAssignment -- zsh -- 80x24'. The terminal displays the following text:

```
Last login: Mon Jun  5 14:09:01 on ttys000
[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php artisan make:migration
on add_category_to_products_table --table=products

INFO Migration [database/migrations/2023_06_05_122239_add_category_to_products_table.php] created successfully.

marajulislam@Marajuls-MacBook-Air MigrationAssignment %
```

Step 2: Write needed coloums in products table

1. Open products table.

The screenshot shows a code editor window with a dark theme. At the top, there are tabs for 'README.md' and two migration files: '2023_06_05_122239_add_category_to_products_table.php' and '2023_06_05_071154_create_products_table.php'. The main content area displays the PHP code for the migration '2023_06_05_122239_add_category_to_products_table.php'. The code defines a class that extends the 'Migration' class from Illuminate\Database\Migrations. It contains two methods: 'up()' and 'down()'. The 'up()' method uses the 'Schema::table' function to create a column named 'category' of type string(50) in the 'products' table. The 'down()' method uses the same function to drop the 'category' column. A tooltip for the '\$table' parameter in the 'up()' method indicates it is of type '\Illuminate\Database\Schema\Blueprint'. The bottom status bar shows the file is '15' lines long, 'Col 44', with 'Spaces: 4', 'UTF-8', 'LF', 'PHP', and other status indicators.

```
2023_06_05_122239_add_category_to_products_table.php — Practice Laravel
Module-16 Laravel Foundation Step-15 > Assignment-16 > MigrationAssignment > database > migrations > 2023_06_05_122239_add_category_to_products_table.php > class > up > Closure

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      */
13     public function up(): void
14     {
15         Schema::table('products', function (Blueprint $table) {
16             $table->string('category', 50);
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::table('products', function (Blueprint $table) {
26             // ...
27         });
28     }
29 };

Ln 15, Col 44  Spaces: 4  UTF-8  LF  PHP  Go Live  Prettier
```

Task 6: After creating the new migration file, run the migration to add the "category" column to the "products" table.

Step 1: Migrate the 'category' table.

1. Open Terminal.
2. Run the following command:

```
php artisan migrate
```

```

MigrationAssignment -- zsh -- 80x24
Last login: Mon Jun  5 18:11:09 on ttys000
[marajulislam@Marajuls-MacBook-Air MigrationAssignment % php artisan migrate

INFO  Running migrations.

2023_06_05_122239_add_category_to_products_table ..... 19ms DONE
marajulislam@Marajuls-MacBook-Air MigrationAssignment %

```

3. Database:

The screenshot shows the phpMyAdmin interface for the 'Assignment-16' database. The left sidebar lists various databases and tables. The 'products' table is selected in the main area. The 'Structure' tab is active, displaying the following table definition:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	<code>bigint(20)</code>	<code>UNSIGNED</code>	No	<code>None</code>		<code>AUTO_INCREMENT</code>		<code>Change</code> <code>Drop</code> <code>More</code>
2	<code>name</code>	<code>varchar(200)</code>	<code>utf8mb4_unicode_ci</code>	No	<code>None</code>				<code>Change</code> <code>Drop</code> <code>More</code>
3	<code>price</code>	<code>decimal(8,2)</code>		No	<code>None</code>				<code>Change</code> <code>Drop</code> <code>More</code>
4	<code>description</code>	<code>text</code>	<code>utf8mb4_unicode_ci</code>	No	<code>None</code>				<code>Change</code> <code>Drop</code> <code>More</code>
5	<code>quantity</code>	<code>int(11)</code>		Yes	<code>NULL</code>				<code>Change</code> <code>Drop</code> <code>More</code>
6	<code>created_at</code>	<code>timestamp</code>		No	<code>current_timestamp()</code>				<code>Change</code> <code>Drop</code> <code>More</code>
7	<code>updated_at</code>	<code>timestamp</code>		No	<code>current_timestamp()</code>	<code>ON UPDATE CURRENT_TIMESTAMP()</code>			<code>Change</code> <code>Drop</code> <code>More</code>
8	<code>category</code>	<code>varchar(50)</code>	<code>utf8mb4_unicode_ci</code>	No	<code>None</code>				<code>Change</code> <code>Drop</code> <code>More</code>

Below the table structure, there are buttons for `Check all`, `Browse`, `Change`, `Drop`, `Primary`, `Unique`, `Index`, `Spatial`, `Fulltext`, and `Add to central columns`. A search bar at the bottom allows adding a column with value '1' and action 'after category'. The 'Indexes' section shows one primary key index:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<code>Edit</code> <code>Rename</code> <code>Drop</code>	<code>PRIMARY</code>	<code>BTREE</code>	Yes	No	<code>id</code>	0	<code>A</code>	No	

Task 7: Create a new migration file named "create_orders_table" that will be responsible for creating a table called "orders" in the database. The "orders" table should have the following columns:

`id`: an auto-incrementing integer and primary key.

`product_id`: an unsigned integer column to establish a foreign key relationship with the "id" column of the "products" table.

`quantity`: an integer column to store the quantity of products ordered.

`created_at`: a timestamp column to store the creation date and time.

`updated_at`: a timestamp column to store the last update date and time.

Step 1: Create a new migration file

1. Open Terminal.
2. Run the following command:

```
php artisan make:migration create_orders_table --create=orders
```

The screenshot shows the phpMyAdmin interface for the 'Assignment-16' database. The 'orders' table is selected. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT		Change Drop More
2	product_id	int(10)	UNSIGNED	No	None				Change Drop More
3	quantity	int(11)		No	None				Change Drop More
4	created_at	timestamp		No	current_timestamp()				Change Drop More
5	updated_at	timestamp		No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()		Change Drop More

Below the table structure, there are buttons for Print, Propose table structure, Track table, Move columns, Normalize, Add (with a dropdown for '1 column(s) after updated_at'), and Go.

The 'Indexes' section shows one index:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	

Below the indexes, there is a 'Create an index on' field with '1 columns' and a 'Go' button. The 'Partitions' and 'Console' sections are also visible at the bottom.

Step 2: Write needed coloums in orders table

1. Open orders table.

The screenshot shows a code editor window with a dark theme. The title bar reads "2023_06_05_131712_create_orders_table.php — Practice Laravel". The left sidebar has various icons for file operations like copy, paste, search, and refresh. The main area contains the following PHP code:

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration {
8     /**
9      * Run the migrations.
10     */
11    public function up(): void
12    {
13        Schema::create('orders', function (Blueprint $table) {
14            $table->id();
15            $table->unsignedInteger('product_id');
16            $table->integer('quantity');
17            //Realtion
18            $table->foreign('product_id')->references('id')->on('products');
19
20            $table->timestamp('created_at')->useCurrent();
21            $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();
22        });
23    }
24
25    /**
26     * Reverse the migrations.
27     */
28    public function down(): void
29    {
30        Schema::dropIfExists('orders');
31    }
32 };
33
```

At the bottom, there are status indicators: "Practice-Laravel*", "0 ↓ 1↑", "Ln 17, Col 23", "Spaces: 4", "UTF-8 LF PHP", "Go Live", "Prettier", and a refresh icon.

Task 8: After creating the migration file for the "orders" table, run the migration to create the "orders" table in the database.

Step 1: Migrate the 'orders' table.

1. Open Terminal.
2. Run the following command:

```
php artisan migrate
```

3. Database:

The screenshot shows the phpMyAdmin interface for the 'Assignment-16' database. The left sidebar lists various databases and tables. The 'orders' table is selected under the 'Assignment-16' database. The main area displays the 'Table structure' for the 'orders' table, which has five columns: id, product_id, quantity, created_at, and updated_at. The 'Indexes' section shows a primary key named 'PRIMARY' on the 'id' column. The 'Partitions' section is currently empty.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	product_id	int(10)		UNSIGNED	No	None			Change Drop More
3	quantity	int(11)			No	None			Change Drop More
4	created_at	timestamp			No	current_timestamp()			Change Drop More
5	updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	

Create an index on columns

Partitions

Console

Author: Marajul Islam