

Migration

Migration দিয়ে আমরা টেবিল create,delete,update(column add,remove,update) করতে পারি।

- Migrations allow you to **create**, **delete**, and **update** tables.
- Updates include adding, removing, renaming, and modifying columns or tables, as well as handling indexes and constraints.

Using Laravel migrations, you can perform the following tasks:

1. Create Tables

- Define the structure of a new table in the database.

2. Delete Tables

- Remove a table from the database.

3. Update Tables

- Modify the structure of an existing table.

This includes:

- **Add Columns:** Add new columns to an existing table.
- **Remove Columns:** Remove existing columns from a table.
- **Rename Columns:** Change the name of an existing column.
- **Update Column Attributes:** Modify the data type, default value, or other attributes of an existing column.
- **Add/Drop Indexes:** Add or remove indexes (e.g., primary key, unique, foreign key).
- **Add/Drop Foreign Keys:** Define or remove foreign key constraints.

4. Rename Tables

- Change the name of an existing table.

5. Add/Remove Table Options

- Set or remove table-specific options (e.g., engine type, collation).

Migration Create করার জন্য Naming Format Follow করতে হয়:

Sanke Case Follow করতে হয়।

1. Creating a Table

Use the prefix `create` followed by the table name and suffix `_table` .

Command:

bash

Copy code

```
php artisan make:migration create_users_table
```

2. Adding a Column to a Table

Use the prefix `add` followed by the column name(s) and the table name.

Command:

bash

Copy code

```
php artisan make:migration add_phone_to_users_table --table=users
```

3. Updating a Column in a Table

Use the prefix `update` followed by the column name(s) and the table name.

Command:

bash

Copy code

```
php artisan make:migration update_email_in_users_table --table=users
```

4. Removing a Column from a Table

Use the prefix `remove` followed by the column name(s) and the table name.

Command:

bash


Copy code

```
php artisan make:migration remove_phone_from_users_table --table=users
```

Create Delete Update এর সকল Syntax:

1. Create a Table:


bash

 Copy code

```
php artisan make:migration create_users_table
```

2. Delete a Table:


bash

 Copy code

```
php artisan make:migration drop_users_table
```

3. Add a Column:


bash

 Copy code

```
php artisan make:migration add_phone_to_users_table --table=users
```

4. Remove a Column:


bash

 Copy code

```
php artisan make:migration remove_phone_from_users_table --table=users
```

5. Rename a Column: Use the doctrine/dbal package:


bash

 Copy code

```
composer require doctrine/dbal
```

Then:


bash

 Copy code

```
php artisan make:migration rename_email_to_contact_in_users_table --table=users
```

6. **Update Column Attributes:** Modify the column in a migration file using `change()` :


bash

 Copy code

```
php artisan make:migration update_column_in_users_table --table=users
```

7. **Rename a Table:**


bash

 Copy code

```
php artisan make:migration rename_users_to_members
```

8. **Add/Remove Indexes or Foreign Keys:**

bash

 Copy code

```
php artisan make:migration add_indexes_to_users_table --table=users
```

UP()

Php artisan:migrate execute হলে এটা call হবে।

1. `up()` Method


The `up()` method is executed when you run the migration using the `php artisan migrate` command. It contains the instructions to apply the intended changes to the database schema.

Example Scenario:

- Creating a table.
- Adding a column.
- Defining a foreign key.

Command:


bash

 Copy code

```
php artisan migrate
```

Code Example:

php

 Copy code

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

Down()

এটা দিয়ে create কৃত জিনিস ডিলিট/রিভার্স করার জন্য ব্যবহৃত হবে।

2. `down()` Method


The `down()` method is executed when you roll back the migration using the `php artisan migrate:rollback` command. It contains the instructions to revert the changes made by the `up()` method.

Example Scenario:

- Dropping a table.
- Removing a column.
- Undoing a foreign key definition.

Command:


bash

 Copy code

```
php artisan migrate:rollback
```

Code Example:

php

 Copy code

```
public function down()
{
    Schema::dropIfExists('users');
}
```

যে সকল ডাটাইপ ডাটাবেসে এর টেবিল এ দিতে পারব তার লিস্ট+কিভাবে ব্যবহার করব।

Integer:

Column Types in Laravel Migrations

1. Integer Types

- `$table->bigIncrements('id');`
 - Auto-incrementing unsigned big integer (primary key).
- `$table->bigInteger('votes');`
 - Big integer.
- `$table->increments('id');`
 - Auto-incrementing unsigned integer (primary key).

- `$table->integer('age');`
 - Integer.
- `$table->mediumInteger('population');`
 - Medium integer.
- `$table->smallInteger('score');`
 - Small integer.
- `$table->tinyInteger('level');`
 - Tiny integer.

- `$table->unsignedBigInteger('user_id');`
 - Unsigned big integer (useful for foreign keys).
- `$table->unsignedInteger('count');`
 - Unsigned integer.

String

2. String Types

- `$table->string('name', 100);`
 - Variable-length string with a maximum length (default is 255).
- `$table->char('code', 2);`
 - Fixed-length string.
- `$table->text('description');`
 - Text column for long text data.
- `$table->mediumText('content');`
 - Medium-length text data.
- `$table->longText('comments');`
 - Very long text data.



Date Time

3. Date and Time Types

- `$table->date('birth_date');`
 - Date column (YYYY-MM-DD).
- `$table->datetime('created_at');`
 - Date and time column (YYYY-MM-DD HH:MM:SS).
- `$table->timestamp('published_at');`
 - Timestamp column.
- `$table->time('event_time');`
 - Time column (HH:MM:SS).
- `$table->year('year');`
 - Year column (YYYY).



Boolean

4. Boolean Type

- `$table->boolean('is_active');`
 - Boolean value (`true` or `false`).

Floating Point And Decimal Types

5. Floating Point and Decimal Types

- `$table->float('rating', 8, 2);`
 - Floating-point number (total digits: 8, decimal places: 2).
- `$table->double('score', 15, 8);`
 - Double precision floating-point number.
- `$table->decimal('price', 10, 2);`
 - Decimal number (total digits: 10, decimal places: 2).

Json and Binary

6. JSON Types

- `$table->json('options');`
 - JSON column.
 - `$table->jsonb('attributes');`
 - JSONB column (if supported by the database).
-

7. Binary Types

- `$table->binary('data');`
 - Binary data.



Enumeration

8. Enumeration Types

- `$table->enum('status', ['draft', 'published', 'archived']);`
 - Enum column with predefined values.

IP address

9. IP Address and UUID Types

- `$table->ipAddress('visitor_ip');`
 - IP address column.
- `$table->macAddress('device_mac');`
 - MAC address column.
- `$table->uuid('unique_id');`
 - UUID column.

Timestamp

11. Timestamp Convenience

- `$table->timestamps();`
 - Adds `created_at` and `updated_at` columns.
- `$table->softDeletes();`
 - Adds `deleted_at` column for soft deletes.
- `$table->rememberToken();`
 - Adds `remember_token` column for authentication.

10. Other Special Types

- `$table->geometry('location');`
 - Spatial data column.
- `$table->point('coordinates');`
 - Geographic point column.
- `$table->lineString('path');`
 - Line string column.

- `$table->polygon('area');`
 - Polygon column.
- `$table->morphs('taggable');`
 - Adds `taggable_id` and `taggable_type` columns for polymorphic relationships.
- `$table->nullableMorphs('taggable');`
 - Nullable version of `morphs`.

```
class CreateExampleTable extends Migration
{
    public function up()
    {
        Schema::create('examples', function (Blueprint $table) {
            $table->id();
            $table->string('name', 100);
            $table->integer('age');
            $table->boolean('is_active')->default(true);
            $table->text('description');
            $table->float('rating', 8, 2);
            $table->json('meta_data');
            $table->timestamps();
            $table->softDeletes();
        });
    }

    public function down()
    {
        Schema::dropIfExists('examples');
    }
}
```

Column Attribute:

Column এ যে সকল Attribute apply করা যায় তার লিস্ট দেয়া হল:

Nullable:

1. Nullable

- Allows the column to accept `NULL` values.

php

Copy code

```
$table->string('email')->nullable();
```

2.Default

2. Default

- Sets a default value for the column.

php

```
$table->integer('status')->default(0);
```

3.Unsigned

3. Unsigned

- Specifies that an integer column should not allow negative values (useful for IDs or counts).

php

Copy code

```
$table->unsignedInteger('count');
```

4.Unique

4. Unique

- Enforces uniqueness for the column values.

php

```
$table->string('email')->unique();
```

5.Primary

5. Primary

- Sets the column as the primary key.

php


```
$table->bigInteger('id')->primary();
```

Index

6. Index

- Adds an index to the column for faster queries.

php

 Copy code


```
$table->string('name')->index();
```

After

7. After

- Specifies the order of the column by placing it after another column.

php

 Copy code


```
$table->string('last_name')->after('first_name');
```

First

8. First

- Adds the column as the first column in the table.

php

 Copy code


```
$table->string('first_name')->first();
```

Change

9. Change

- Modifies an existing column (requires the `doctrine/dbal` package).

php

 Copy code


```
$table->string('email', 150)->change();
```

Comment

10. Comment

- Adds a comment to the column.

php

 Copy code


```
$table->string('name')->comment('This is the user name');
```

Auto Increment

11. Auto-Increment

- Specifies that the column should auto-increment (used for IDs).

php

 Copy code

```
$table->increments('id');
```

Foreign

12. Foreign Key

- Sets a foreign key constraint for the column.

```
php Copy code  
  
$table->unsignedBigInteger('user_id');  
$table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
```

Collation

13. Collation

- Specifies the collation for a column (for strings).

```
php Copy code  
  
$table->string('name')->collation('utf8mb4_unicode_ci');
```

Charset

14. Charset

- Sets the character set for a string column.

```
php Copy code  
  
$table->string('name')->charset('utf8mb4');
```

Stored and Virtual Columns

15. Stored and Virtual Columns

- Creates a computed column (stored or virtual).


```
php Copy code  
  
$table->integer('total')->storedAs('price * quantity');
```

Use Current

16. Use Current

- Automatically sets the current timestamp for `datetime` or `timestamp` columns.

php

 Copy code


```
$table->timestamp('created_at')->useCurrent();
```

Generated As

17. Generated As

- Used for generated (computed) columns in databases like PostgreSQL.

php

 Copy code


```
$table->integer('score')->generatedAs('base_score + bonus_score');
```

Spatial Index

18. Spatial Index

- Creates a spatial index for geographic data.

php

 Copy code

```
$table->point('location')->spatialIndex();
```

```
class CreateUsersTable extends Migration
```

[Copy code](#)

```
{
```

```
    public function up()
```

```
    {
```

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

```
            $table->id();
```

```
            $table->string('name')->unique()->comment('User name');
```

```
            $table->string('email')->unique();
```

```
            $table->timestamp('email_verified_at')->nullable();
```

```
            $table->string('password');
```

```
            $table->rememberToken();
```

```
            $table->timestamps();
```

```
            $table->softDeletes();
```

```
        });
```

```
    }
```

```
    public function down()
```

```
    {
```

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

```
    }
```


Real Example:

1.Create Users Table

Php artisan make:migration create_users_table

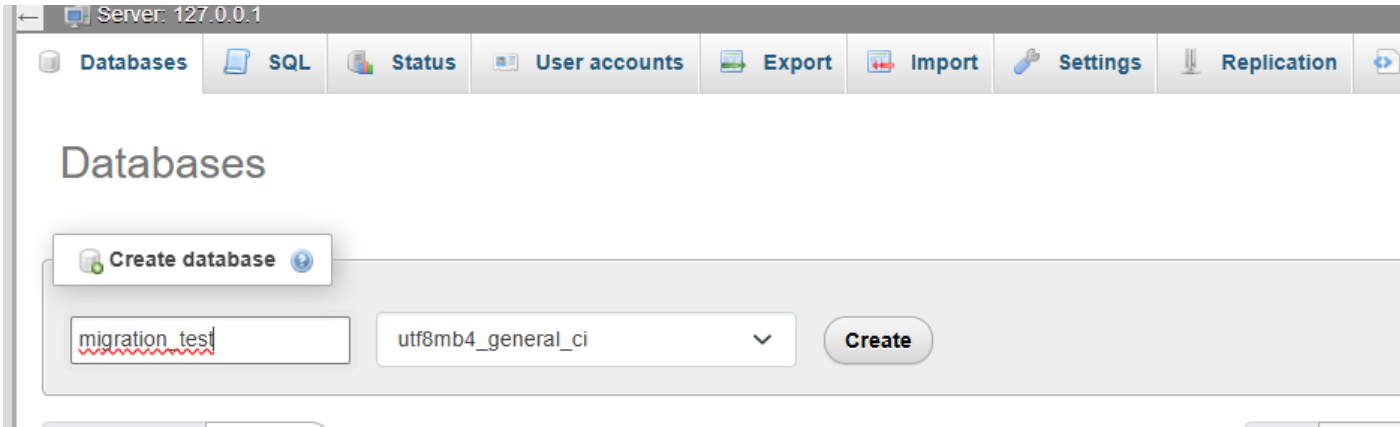
```
User@DESKTOP-TSJJMAQ MINGW64 /c:/xampp/htdocs/module_17_migration
• $ php artisan make:migration create_users_table

[INFO] Migration [C:\xampp\htdocs\module_17_migration\database\Migrations\2024_12_13_081315_create_users_table.php] created successfully.
```

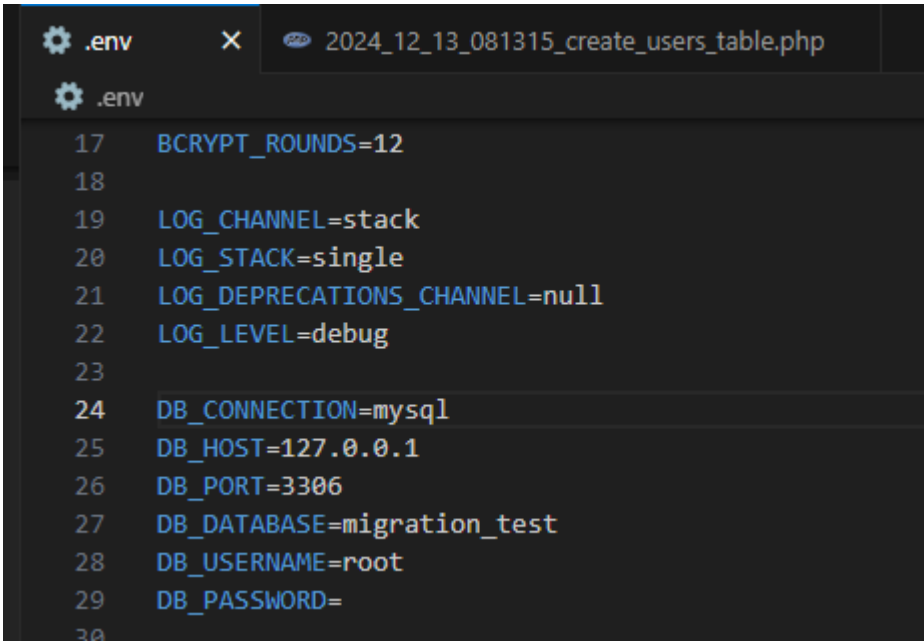
Create table

```
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('users', function (Blueprint $table) {
15             $table->id(); // Unsigned Big integer Primary key(auto increment)
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->string('password');
19             $table->timestamp('created_at')->useCurrent(); // Default to CURRENT_TIMESTAMP
20             $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate(); // Updates to CURRENT_TIMESTAMP on update
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      */
27     public function down(): void
28     {
29         Schema::dropIfExists('users');
30     }
31 };
32
```

Database Create



Database এর সাথে লিঙ্ক



Migration

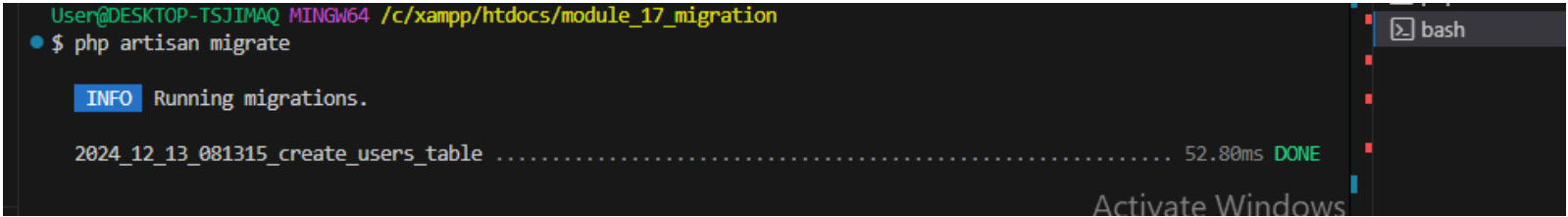
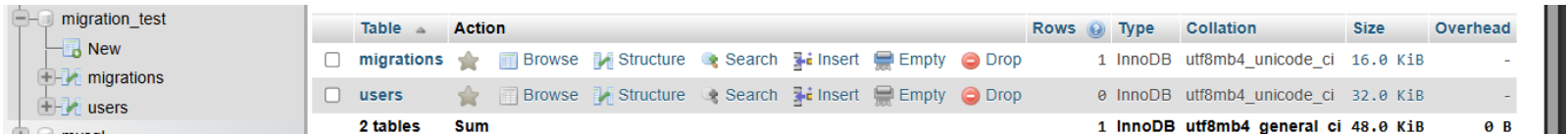


Table Created



Migration Table যা automatically create হবে এখানে migration এর History থাকে।

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

← T →

idmigrationbatch

☐ Edit Copy Delete 3 2024_12_13_081315_create_users_table 1

SELECT * FROM `users`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id name email password created_at updated_at

Query results operations

Create view

Table এ ডাটা ইনসার্ট:

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

SELECT * FROM `users`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

← T →

idnameemailpasswordcreated_atupdated_at

☐ Edit Copy Delete 1 utsab utasb@duet.ac.bd 12345 2024-12-13 14:34:21 2024-12-13 14:34:21

☐ Edit Copy Delete 2 babu babu@duet.ac.bd 1234 2024-12-13 14:34:21 2024-12-13 14:34:21

Rollback:

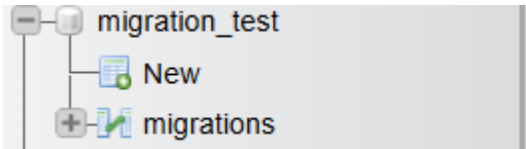
Rollback করলে পূর্বের migration delete হবে এবং যদি ঐ টেবিল এ ডাটা থাকে তাহলে ঐ ডাটাও ডিলিট হয়ে যাবে।

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
• $ php artisan migrate:rollback

INFO Rolling back migrations.

2024_12_13_081315_create_users_table ..... 19.18ms DONE
```

Table delete হয়ে যাবে:



পূর্বের Table ফিরে পেতে চাইলে আবার migration create করব(ডাটা insert করতে হবে আবার।)

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
• $ php artisan migrate

INFO Running migrations.

2024_12_13_081315_create_users_table ..... 43.94ms DONE
```

- If you want to rollback a specific number of batches:

bash

Copy code


php artisan migrate:rollback --step=1

Refresh: এর মাধ্যমে migration সব rollback(ডিলিট হবে/টেবিল ডিলিট হবে) হবে এবং নতুন করে সব migration create(টেবিল create) হবে।

4. Refreshing Migrations

To rollback and re-run all migrations in one step:


bash

 Copy code

```
php artisan migrate:refresh
```

- This is useful during development to reset the database and apply all migrations again.
- You can optionally specify the number of steps to refresh:

bash

 Copy code


```
php artisan migrate:refresh --step=2
```

Reset(সব migration delete হয়ে যাবে তবে নতুন করে migration চলবে না।)

5. Resetting Migrations

To rollback all migrations (all batches):

bash

 Copy code

```
php artisan migrate:reset
```

- This drops all tables created by migrations but does not reapply them.

Wiping The Database:

6. Wiping the Database

To completely clear the database:

```
bash Copy code  
  
php artisan db:wipe
```

- This is often used when the database schema has changed drastically.

Migration Status

Run `migrate:status` : Check the current status of migrations:

```
bash Copy code  
  
php artisan migrate:status
```

Add New Column:

Migration create:

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
• $ php artisan make:migration add_mobile_number_to_users --table=users
```

```
INFO Migration [C:\xampp\htdocs\module_17_migration\database\migrations\2024_12_13_104418_add_mobile_number_to_users.php] created successfully.
```

```
1  return new class extends Migration
2  {
3      /**
4       * Run the migrations.
5       */
6      public function up(): void
7      {
8          Schema::table('users', function (Blueprint $table) {
9              //add new column
10             $table->string('mobile_number')->nullable()->unique();
11         });
12     }
13
14     /**
15      * Reverse the migrations.
16      */
17     public function down(): void
18     {
19         Schema::table('users', function (Blueprint $table) {
20             //
21             $table->dropColumn('mobile_number');
22         });
23     }
24 };
```


Migration:

```
$ php artisan migrate

INFO Running migrations.

INFO Running migrations.
INFO Running migrations.

2024_12_13_104418_add_mobile_number_to_users ..... 26.49ms DONE
```

Migration Table এ সব রেকর্ড হচ্ছে:

SELECT * FROM `migrations`

☐ Profiling

[\[Edit inline \]](#)

[\[Edit \]](#)

[\[Explain SQL \]](#)

[\[Create PHP code \]](#)

[\[Refresh \]](#)

☐ Show all

Number of rows: 25

Filter rows:

Sort by key: None

Extra options

			id	migration	batch
<input type="checkbox"/>	Edit	Copy	Delete	5 2024_12_13_081315_create_users_table	1
<input type="checkbox"/>	Edit	Copy	Delete	6 2024_12_13_104027_add_column_name_to_table_name	2
<input type="checkbox"/>	Edit	Copy	Delete	7 2024_12_13_104418_add_mobile_number_to_users	3

Rename Column:

Rename এর জন্য package install করতে হবে

1. Install the package:

```
bash
```

[Copy code](#)

```
composer require doctrine/dbal
```

```
C:\Users\User>composer require doctrine/dbal
./composer.json has been created
Running composer update doctrine/dbal
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
- Locking doctrine/dbal (4.2.1)
- Locking doctrine/deprecations (1.1.4)
- Locking psr/cache (3.0.0)
- Locking psr/log (3.0.2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
Failed to download psr/log from dist: The zip extension and unzip/7z commands are both missing, skipping.
```

Migration Create করব:

```
User@DESKTOP-TSJIQAQ MINGW64 /c:/xampp/htdocs/module_17_migration
• $ php artisan make:migration rename_mobile_number_in_users --table=users

INFO Migration [C:\xampp\htdocs\module_17_migration\database\Migrations\2024_12_13_110745_rename_mobile_number_in_users.php] created successfully.
```

```
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::table('users', function (Blueprint $table) {
15             //
16             $table->renameColumn('mobile_number', 'mobile_number_ok');
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::table('users', function (Blueprint $table) {
26             //
27             $table->renameColumn('mobile_number_ok', 'mobile_number');
28         });
29     }
30 };
31
```

Migration Execute (database এ store):

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
• $ php artisan migrate

INFO Running migrations.

2024_12_13_110745_rename_mobile_number_in_users ..... 45.92ms DONE
```

Rename হয়েছে

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

id	name	email	password	created_at	updated_at	mobile_number_ok
----	------	-------	----------	------------	------------	------------------

Query results operations

Remove Column Name:

Migration Create

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
```

```
• $ php artisan make:migration remove_mobile_number_ok_from_users --table=users
```

```
INFO Migration [C:\xampp\htdocs\module_17_migration\database\Migrations\2024_12_13_125455_remove_mobile_number_ok_from_users.php] created successfully.
```

```
1  return new class extends Migration
2  {
3      /**
4       * Run the migrations.
5       */
6      public function up(): void
7      {
8          Schema::table('users', function (Blueprint $table) {
9              $table->dropColumn('mobile_number_ok');
10          });
11      }
12
13      /**
14       * Reverse the migrations.
15       */
16      public function down(): void
17      {
18          Schema::table('users', function (Blueprint $table) {
19              //
20              $table->string('mobile_number_ok'); // Recreate the column if needed
21          });
22      }
23  };
```

Migration Execute

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
```

```
• $ php artisan migrate
```

```
INFO Running migrations.
```

```
2024_12_13_125455_remove_mobile_number_ok_from_users ..... 23/24ms DONE
```

Activate Windows

Go to Settings to activate Windows.

Column Deleted

SELECT * FROM `users`

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

id	name	email	password	created_at	updated_at
----	------	-------	----------	------------	------------

Query results operations

Update Column:

Migration Create:

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
● $ php artisan make:migration update_mobile_number_ok_in_users --table=users

INFO Migration [C:\xampp\htdocs\module_17_migration\database\migrations\2024_12_13_131714_update_mobile_number_ok_in_users.php] created successfully.
```

```
1  return new class extends Migration
2  {
3      /**
4       * Run the migrations.
5       */
6      public function up(): void
7      {
8          Schema::table('users', function (Blueprint $table) {
9              $table->string('mobile_number_ok',255)->nullable()->unique()->change();
10         });
11     }
12
13     /**
14      * Reverse the migrations.
15      */
16     public function down(): void
17     {
18         Schema::table('users', function (Blueprint $table) {
19             //for rollback
20             $table->string('mobile_number_ok',100)->nullable(false)->change();
21         });
22     }
23 };
```

Table Store

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
○ $ php artisan migrate
```

Activate Windows
Go to Settings to activate Windows.

Table এর property change হয়ে গেছে:


<input type="checkbox"/>	7	mobile_number_ok		varchar(255)	utf8mb4_unicode_ci	Yes	NULL	
--------------------------	---	------------------	---	--------------	--------------------	-----	------	---

Table Rename:

Migration Create

User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration

• \$ php artisan make:migration rename_users_to_usersBoss

INFO Migration [C:\xampp\htdocs\module_17_migration\database\Migrations\2024_12_13_150023_rename_users_to_usersBoss.php] created successfully.

```
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::rename('users', 'usersBoss');
15     }
16
17     /**
18      * Reverse the migrations.
19      */
20     public function down(): void
21     {
22         Schema::table('users', function (Blueprint $table) {
23             Schema::rename('usersBoss', 'users');
24         });
25     }
26 };
```

Table Rename To Database(ডাটাবেসে পরিবর্তন।)

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
• $ php artisan migrate

INFO Running migrations.

2024_12_13_150023_rename_users_to_users_boss ..... 49.74ms DONE
```

migration_test

- New
- migrations
- usersboss

id	name	email	password	created_at	updated_at	mobile_number_ok
----	------	-------	----------	------------	------------	------------------

Query results operations

DROP TABLE

Migration Create:

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
● $ php artisan migrate

INFO Running migrations.

2024_12_13_152206_drop_users_boss_table ..... 16.23ms DONE
```

```
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::dropIfExists('usersBoss');
15     }
16
17     /**
18      * Reverse the migrations.
19      */
20     public function down(): void
21     {
22         Schema::create('users', function (Blueprint $table) {
23             $table->id(); // Unsigned Big integer Primary key(auto increment)
24             $table->string('name');
25             $table->string('email')->unique();
26             $table->string('password');
27             $table->timestamp('created_at')->useCurrent(); // Default to CURRENT_TIMESTAMP
28             $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate(); // Updates to CURRENT_TIMESTAMP on update
29         });
30     }
31 };
32
```

```
User@DESKTOP-TSJIMAQ MINGW64 /c/xampp/htdocs/module_17_migration
● $ php artisan migrate

INFO Running migrations.

2024_12_13_152206_drop_users_boss_table ..... 16.23ms DONE
```

Table Deleted

