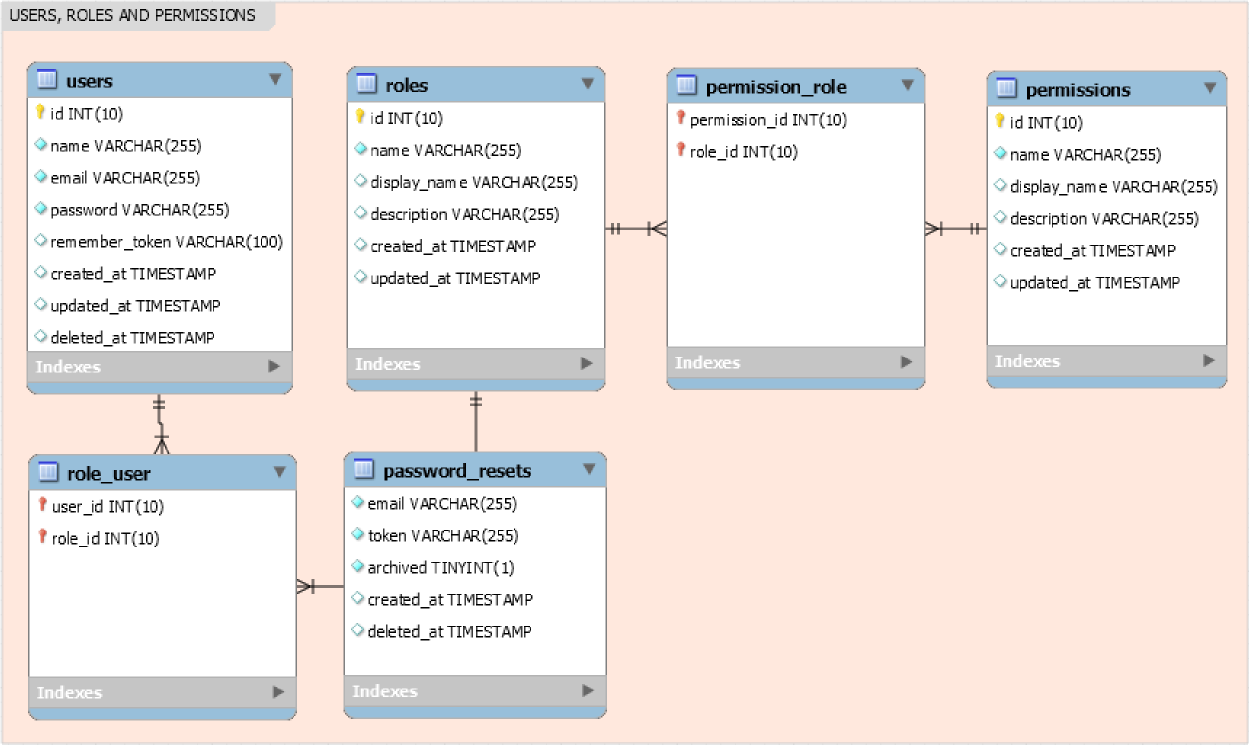
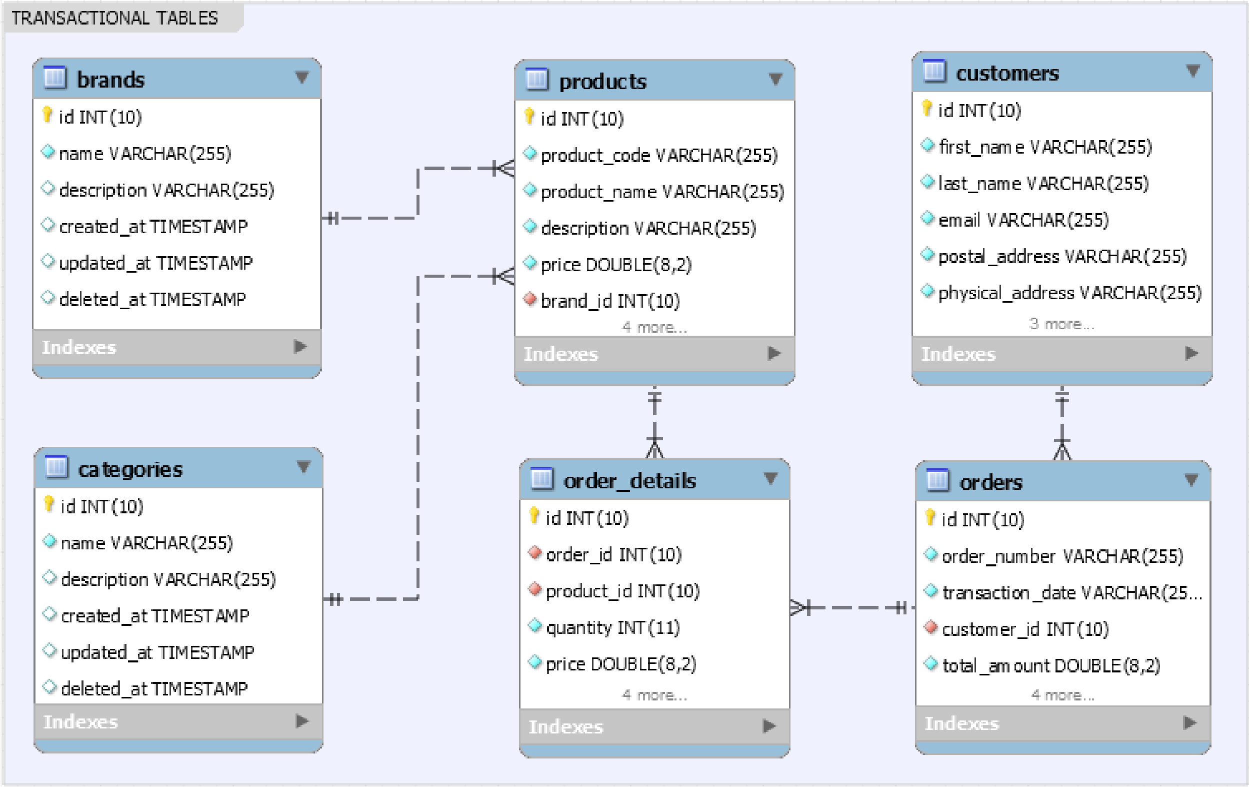

USERS, ROLES AND PERMISSIONS
users
id INT(10)
name VARCHAR(255)
email VARCHAR(255)
password VARCHAR(255)
remember_token VARCHAR(100)
created_at TIMESTAMP
updated_at TIMESTAMP
deleted_at TIMESTAMP
Indexes
roles
id INT(10)
name VARCHAR(255)
display_name VARCHAR(255)
description VARCHAR(255)
created_at TIMESTAMP
updated_at TIMESTAMP
Indexes
permission_role
permission_id INT(10)
role_id INT(10)
Indexes
permissions
id INT(10)
name VARCHAR(255)
display_name VARCHAR(255)
description VARCHAR(255)
created_at TIMESTAMP
updated_at TIMESTAMP
Indexes
role_user
user_id INT(10)
role_id INT(10)
Indexes
password_resets
email VARCHAR(255)
token VARCHAR(255)
archived TINYINT(1)
created_at TIMESTAMP
deleted_at TIMESTAMP
Indexes


TRANSACTIONAL TABLES
brands
id INT(10)
name VARCHAR(255)
description VARCHAR(255)
created_at TIMESTAMP
updated_at TIMESTAMP
deleted_at TIMESTAMP
Indexes
products
id INT(10)
product_code VARCHAR(255)
product_name VARCHAR(255)
description VARCHAR(255)
price DOUBLE(8,2)
brand_id INT(10)
4 more...
Indexes
customers
id INT(10)
first_name VARCHAR(255)
last_name VARCHAR(255)
email VARCHAR(255)
postal_address VARCHAR(255)
physical_address VARCHAR(255)
3 more...
Indexes
categories
id INT(10)
name VARCHAR(255)
description VARCHAR(255)
created_at TIMESTAMP
updated_at TIMESTAMP
deleted_at TIMESTAMP
Indexes
order_details
id INT(10)
order_id INT(10)
product_id INT(10)
quantity INT(11)
price DOUBLE(8,2)
4 more...
Indexes
orders
id INT(10)
order_number VARCHAR(255)
transaction_date VARCHAR(25...
customer_id INT(10)
total_amount DOUBLE(8,2)
4 more...
Indexes

USERS, ROLES AND PERMISSIONS

Table: Users |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	VARCHAR(255)	Primary key
2	name	VARCHAR(255)	The user's name
3	email	VARCHAR(255)	The user's email address
4	password	VARCHAR(255)	The user's password
5	remember_token	VARCHAR(100)	The remember me token for logged in users

Tables: Password Resets |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	email	VARCHAR(255)	User's registered email address
2	token	VARCHAR(255)	Password reset token

Table: Roles |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	VARCHAR(255)	Primary key
2	name	VARCHAR(255)	The name of the role i.e. admin
3	display_name	VARCHAR(255)	The name that we will display when listing the roles
4	description	VARCHAR(255)	The description of the user role

Table: Role User |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	user_id	INT (10)	User's id in users table
2	role_id	INT (10)	Role id in roles table

Table: Permissions |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	name	VARCHAR(255)	Permission name
3	display_name	VARCHAR(255)	Human readable permission name

Table: Permission Role |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	permission_id	INT	Permission id
2	role_id		

TRANSACTIONS TABLES

Table: Brands |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	name	VARCHAR(255)	The name of the brand
3	description	VARCHAR(255)	The description of the brand

Table: Categories |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	name	VARCHAR(255)	The name of the category
3	description	VARCHAR(255)	The description of the category

Table: Products |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	product_code	VARCHAR(255)	A unique code that identifies the product
3	product_name	VARCHAR(255)	The name of the product
4	description	VARCHAR(255)	The description of the product
5	price	VARCHAR(255)	The price of the product
6	brand_id	INT(10)	The brand id foreign key
7	category_id	INT(10)	The category id foreign key

Table: Customers |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	first_name	VARCHAR(255)	Customer's first name

3	last_name	VARCHAR(255)	Customer's lastname
4	email	VARCHAR(255)	Email address for the customer
5	postal_address	VARCHAR(255)	Postal address for the customer
6	physical_address	VARCHAR(255)	Customer’s physical address for delivery

Table: Orders |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	order_number	VARCHAR(255)	Auto generated order unique number
3	transaction_date	TIMESTAMP	Date that the order was placed
4	customer_id	INT	The id for the customer who placed the order
5	status	VARCHAR(45)	Status can either be pending or processed
6	total_amount	DOUBLE(8,2)	

Table: Order_Details |

S/N	FIELD	DATA TYPE	DESCRIPTION
1	id	INT	Primary key
2	order_id	INT(10)	Order id
3	product_id	INT(10)	Product primary key value
4	quantity	INT(11)	The quantity of the ordered product
5	price	DOUBLE(8,2)	The price of the product at the time of the order
6	sub_total	DOUBLE(8,2)	The total of the quantity multiplied by the price

MODELS AND MIGRATIONS

```
CREATE SCHEMA DemoLaravel;
```

```
DB_HOST=localhost
DB_DATABASE= DemoLaravel
DB_USERNAME=root
DB_PASSWORD=
```

```
php artisan make:model Models\Brand –m
php artisan make:model Models\Category -m
php artisan make:model Models\Product -m
php artisan make:model Models\Customer -m
php artisan make:model Models\Order -m
php artisan make:model Models\OrderDetail -m
```

Database migration files

/database/migrations/ 2019_10_28_124527_create_brands_table.php

```php
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateBrandsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('brands', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name')->unique();
            $table->string('description')->nullable();
            $table->timestamps();
            $table->softDeletes();
        });
    }

    /**
```

```
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('brands');
    }
}
```

/database/migrations/2019_10_28_124655_create_categories_table.php

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateCategoriesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name')->unique();
            $table->string('description')->nullable();
            $table->timestamps();
            $table->softDeletes();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('categories');
    }
}
```

/database/migrations/2019_10_28_124707_create_products_table.php

```php
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->increments('id');
            $table->string('product_code')->unique();
            $table->string('product_name')->unique();
            $table->string('description');
            $table->float('price')->default(0);
            $table->integer('brand_id')->unsigned()->index();
            $table->integer('category_id')->unsigned()->index();
            $table->timestamps();
            $table->softDeletes();

            $table->foreign('brand_id')->references('id')->on('brands');
            $table->foreign('category_id')->references('id')->on('categories');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('products');
    }
}
```

/database/migrations/2019_10_28_124726_create_customers_table.php

```php
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateCustomersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('customers', function (Blueprint $table) {
            $table->increments('id');
            $table->string('first_name');
            $table->string('last_name');
            $table->string('email')->unique();
            $table->string('postal_address');
            $table->string('physical_address');
            $table->timestamps();
            $table->softDeletes();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('customers');
    }
}
```

/database/migrations/2019_10_28_124736_create_orders_table.php

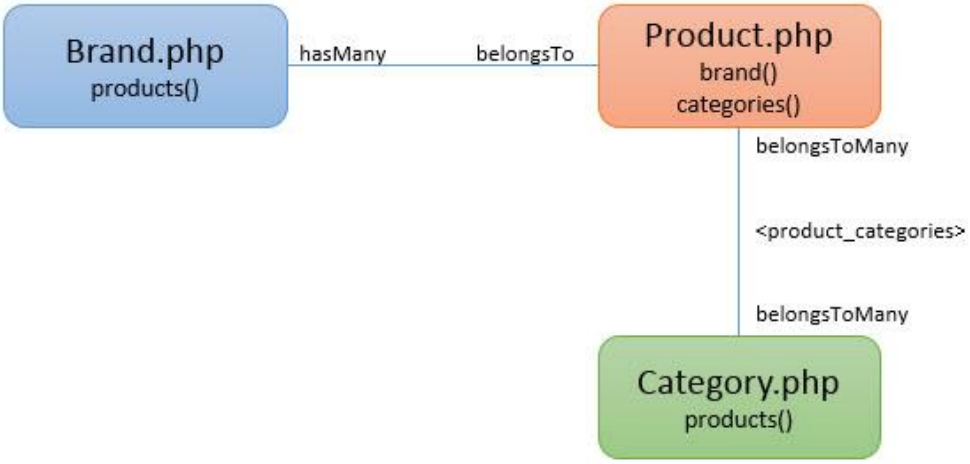
```php
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateOrdersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('orders', function (Blueprint $table) {
            $table->increments('id');
            $table->string('order_number')->unique();
            $table->string('transaction_date');
            $table->integer('customer_id')->unsigned()->index();
            $table->float('total_amount');
            $table->string('status');
            $table->timestamps();
            $table->softDeletes();

            $table->foreign('customer_id')->references('id')->on('customers');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('orders');
    }
}
```

/database/migrations/2019_10_28_124756_create_order_details_table.php

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateOrderDetailsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('order_details', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('order_id')->unsigned();
            $table->integer('product_id')->unsigned()->index();
            $table->integer('quantity');
            $table->float('price');
            $table->float('sub_total');
            $table->timestamps();
            $table->softDeletes();

            $table->foreign('order_id')->references('id')->on('orders');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('order_details');
    }
}
```

```
php artisan migrate
```

ELOQUENT ONE TO MANY RELATIONSHIP



- Brand.php -> function products()
- Product.php -> function brand()
- Product.php -> function categories()
- Category.php -> function products()

/app/Models/Brand.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Brand extends Model
{
    use SoftDeletes;

    /**
     * The attributes that should be mutated to dates.
     *
     * @var array
     */
    protected $dates = ['deleted_at'];

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name',
        'description',
    ];

    /**
     * Get the products for the brand.
     */
    public function products()
    {
        return $this->hasMany('Models\Product','brand_id','id');
    }
}
```

/app/Models/Product.php

```php
<?php

namespace Larashop\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Product extends Model
{
    use SoftDeletes;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'product_code',
        'product_name',
        'description',
        'price',
        'brand_id',
        'category_id',
    ];

    /**
     * Get the brand that the product belongs to.
     */
    public function brand()
    {
        return $this->belongsTo('Models\Brand','brand_id');
    }

    /**
     * Get the category that the product belongs to.
     */
    public function category()
    {
        return $this->belongsTo('Models\Category','category_id');
    }

     /* many - many
     public function categories()
     {
          return $this->belongsToMany('Models\Category', 'product_category',
               'product_id', 'category_id');
     }
     */
}
```

/app/Models/Category.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
```

```php
use Illuminate\Database\Eloquent\SoftDeletes;

class Category extends Model
{
    use SoftDeletes;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name',
        'description',
    ];

    /**
     * Get the products for the category.
     */
    public function products()
    {
        return $this->hasMany('Models\Product', category_id','id');
    }
}
```

/app/Models/Customer.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Customer extends Model
{
    use SoftDeletes;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'first_name',
        'last_name',
        'email',
        'postal_address',
        'physical_address',
    ];
}
```

/app/Model/Order.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Order extends Model
{
    use SoftDeletes;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'order_number',
        'transaction_date',
        'customer_id',
        'total_amount',
        'status',
    ];
}
```

/app/Models/OrderDetail.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class OrderDetail extends Model
{
    use SoftDeletes;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'order_id',
        'product_id',
        'quantity',
        'price',
        'sub_total',
    ];
}
```

LARAVEL TINKER

```
php artisan tinker
```

Tinker là một tiện ích dòng lệnh cho phép bạn tương tác với môi trường Laravel. Nó là một công cụ tốt để kiểm tra các chức năng mà không cần sử dụng trình duyệt web.

```
$brand = new App\Models\Brand;
```

Thêm 1 dòng

```
$brand->name = "Chanel";
$brand->description = "It is Chanel";
$brand->save();
```

Xem brand

```
$brand->all();
```

```
$product = new Larashop\Models\Product;

$product->product_code = "T-Shirt";
$product->product_name = "T-Shirt";
$product->description = "Men's cool T-Shirt";
$product->price = "65";
$product->brand_id = "1";
$product->category_id = "1";
$product->save();
```

```
$brand::find(1)->products;
```