Perfect 👍 You want a **complete list of all tables** in the e-commerce database design with their **properties** and **descriptions**.  
Here’s a structured breakdown:

**📌 Relational Database Design for Mid-Level E-Commerce**

**1. users**

Holds customer, admin, and seller details.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Unique user identifier |
| name | VARCHAR(150) | Full name of the user |
| email | VARCHAR(150), UNIQUE | User login/communication email |
| password\_hash | VARCHAR(255) | Hashed password for authentication |
| phone | VARCHAR(20) | User contact number |
| role | ENUM(customer, admin, seller) | Defines user role |
| created\_at | TIMESTAMP | Record creation time |
| updated\_at | TIMESTAMP | Last update time |

**2. categories**

Supports recursive categories (subcategory to N levels).

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Category ID |
| name | VARCHAR(150) | Category name |
| parent\_id | BIGINT (FK → categories.id) | Points to parent category (NULL for root) |
| description | TEXT | Details about the category |
| created\_at | TIMESTAMP | Record creation time |
| updated\_at | TIMESTAMP | Last update time |

**3. brands**

Stores product brands.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Brand ID |
| name | VARCHAR(150) | Brand name |
| description | TEXT | Brand description |
| created\_at | TIMESTAMP | Record creation time |
| updated\_at | TIMESTAMP | Last update time |

**4. sizes**

Standard product sizes.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Size ID |
| name | VARCHAR(50) | Size name (e.g., S, M, L, XL) |
| created\_at | TIMESTAMP | Record creation time |

**5. products**

Main product details.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Product ID |
| category\_id | BIGINT (FK → categories.id) | Product category |
| brand\_id | BIGINT (FK → brands.id) | Product brand |
| name | VARCHAR(255) | Product name |
| sku | VARCHAR(100), UNIQUE | Stock keeping unit |
| description | TEXT | Product details |
| base\_price | DECIMAL(10,2) | Base price |
| stock\_quantity | INT | Total stock available |
| weight | DECIMAL(8,2) | Weight (optional for shipping) |
| is\_seasonal | BOOLEAN | Seasonal product flag |
| seasonal\_start\_date | DATE | Seasonal start date |
| seasonal\_end\_date | DATE | Seasonal end date |
| status | ENUM(active, inactive) | Product availability |
| created\_at | TIMESTAMP | Record creation time |
| updated\_at | TIMESTAMP | Last update time |

**6. product\_images**

Stores multiple images for each product.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Image ID |
| product\_id | BIGINT (FK → products.id) | Related product |
| image\_url | VARCHAR(255) | Image path/URL |
| is\_primary | BOOLEAN | Marks main image |
| created\_at | TIMESTAMP | Record creation time |

**7. product\_variants**

Stores size/color variations of a product.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Variant ID |
| product\_id | BIGINT (FK → products.id) | Related product |
| size\_id | BIGINT (FK → sizes.id) | Product size |
| color | VARCHAR(50) | Product color |
| additional\_price | DECIMAL(10,2) | Extra cost for variant |
| stock\_quantity | INT | Available stock |
| created\_at | TIMESTAMP | Record creation time |

**8. wishlists**

Stores user’s saved products.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Wishlist ID |
| user\_id | BIGINT (FK → users.id) | Related user |
| product\_id | BIGINT (FK → products.id) | Related product |
| created\_at | TIMESTAMP | Record creation time |

**9. orders**

Stores purchase orders.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Order ID |
| user\_id | BIGINT (FK → users.id) | Related customer |
| total\_amount | DECIMAL(10,2) | Subtotal amount |
| discount\_amount | DECIMAL(10,2) | Total discount applied |
| shipping\_fee | DECIMAL(10,2) | Shipping charge |
| final\_amount | DECIMAL(10,2) | Total after discounts/shipping |
| status | ENUM(pending, paid, shipped, delivered, cancelled) | Order status |
| created\_at | TIMESTAMP | Order date |
| updated\_at | TIMESTAMP | Last update |

**10. order\_items**

Stores items in an order.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Item ID |
| order\_id | BIGINT (FK → orders.id) | Related order |
| product\_id | BIGINT (FK → products.id) | Related product |
| variant\_id | BIGINT (FK → product\_variants.id) | Specific variant |
| quantity | INT | Number of units |
| price\_at\_purchase | DECIMAL(10,2) | Price at purchase time |
| discount\_applied | DECIMAL(10,2) | Discount per item |
| created\_at | TIMESTAMP | Record creation time |

**11. payments**

Stores payment transactions.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Payment ID |
| order\_id | BIGINT (FK → orders.id) | Related order |
| payment\_method | ENUM(card, mobile\_banking, COD) | Payment method |
| transaction\_id | VARCHAR(150) | Payment gateway transaction ID |
| amount | DECIMAL(10,2) | Paid amount |
| status | ENUM(pending, success, failed) | Payment status |
| created\_at | TIMESTAMP | Payment date |

**12. coupons**

Coupon details for discounts.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Coupon ID |
| code | VARCHAR(50), UNIQUE | Coupon code |
| description | TEXT | Coupon details |
| discount\_type | ENUM(percentage, fixed) | Type of discount |
| discount\_value | DECIMAL(10,2) | Value of discount |
| min\_purchase\_amount | DECIMAL(10,2) | Minimum order value |
| max\_discount\_amount | DECIMAL(10,2) | Maximum discount (optional) |
| valid\_from | DATE | Start date |
| valid\_to | DATE | End date |
| usage\_limit | INT | Max number of uses |
| status | ENUM(active, inactive) | Status |
| created\_at | TIMESTAMP | Creation time |

**13. coupon\_usages**

Tracks coupon usage.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Usage ID |
| coupon\_id | BIGINT (FK → coupons.id) | Related coupon |
| user\_id | BIGINT (FK → users.id) | Who used the coupon |
| order\_id | BIGINT (FK → orders.id) | Order where coupon applied |
| used\_at | TIMESTAMP | When coupon was used |

**14. discounts**

Special discounts on categories/products.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Discount ID |
| product\_id | BIGINT NULL (FK → products.id) | Discounted product |
| category\_id | BIGINT NULL (FK → categories.id) | Discounted category |
| discount\_type | ENUM(percentage, fixed) | Discount type |
| discount\_value | DECIMAL(10,2) | Value of discount |
| valid\_from | DATE | Start date |
| valid\_to | DATE | End date |
| created\_at | TIMESTAMP | Creation time |

**15. shipping\_methods**

Defines shipping options.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Shipping ID |
| name | VARCHAR(150) | Shipping name |
| description | TEXT | Details |
| fee | DECIMAL(10,2) | Shipping fee |
| is\_free\_shipping | BOOLEAN | Free shipping flag |
| created\_at | TIMESTAMP | Record creation time |

**16. order\_shipping**

Stores shipping details for each order.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Shipping record ID |
| order\_id | BIGINT (FK → orders.id) | Related order |
| shipping\_method\_id | BIGINT (FK → shipping\_methods.id) | Selected method |
| address | TEXT | Delivery address |
| tracking\_number | VARCHAR(100) | Courier tracking number |
| created\_at | TIMESTAMP | Creation time |

**17. reviews**

Product reviews and ratings.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Review ID |
| user\_id | BIGINT (FK → users.id) | Reviewer |
| product\_id | BIGINT (FK → products.id) | Reviewed product |
| rating | INT (1–5) | Star rating |
| comment | TEXT | Customer review |
| created\_at | TIMESTAMP | Review date |

**18. inventory\_logs**

Tracks stock changes.

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| id | BIGINT (PK) | Log ID |
| product\_id | BIGINT (FK → products.id) | Related product |
| variant\_id | BIGINT NULL (FK → product\_variants.id) | Related variant |
| change\_type | ENUM(in, out) | Stock in/out |
| quantity\_changed | INT | Amount changed |
| note | TEXT | Reason (sale, restock, return) |
| created\_at | TIMESTAMP | Date of log |

**19. notifications**

System notifications for users.

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id | BIGINT (PK) | Notification ID |
| user\_id | BIGINT (FK → users.id) | Receiver |
| message | TEXT | Notification content |
| status | ENUM(unread, read) | Read status |
| created\_at | TIMESTAMP | Notification time |

✅ This is a **full professional e-commerce schema** with **all features you requested + standard tables** (sizes, brands, reviews, logs, notifications).

In **Laravel 12**, you can manage **soft delete** easily using the SoftDeletes trait.  
Soft deleting means instead of permanently removing a record from the database, Laravel will just set a deleted\_at timestamp. That way you can restore it later if needed.

### 🔹 Steps to Manage Soft Delete in Laravel 12

#### 1. **Add** deleted\_at **column in migration**

When creating a table (for example products), add the softDeletes() column:

public function up(): void

{

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

$table->id();

$table->string('name');

$table->decimal('price', 10, 2);

$table->integer('stock\_quantity');

$table->timestamps();

$table->softDeletes(); // 👈 adds deleted\_at column

});

}

Then run:

php artisan migrate

#### 2. **Enable SoftDeletes in Model**

In your model (e.g., Product.php):

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\SoftDeletes;

class Product extends Model

{

use SoftDeletes; // 👈 enables soft delete

protected $fillable = ['name', 'price', 'stock\_quantity'];

protected $dates = ['deleted\_at']; // optional in Laravel 12 (auto-detected)

}

#### 3. **Using Soft Deletes in Queries**

* **Delete a record (soft delete):**

$product = Product::find(1);

$product->delete(); // sets deleted\_at timestamp

* **Get all records except soft deleted:**

$products = Product::all();

* **Get only soft deleted records:**

$trashed = Product::onlyTrashed()->get();

* **Get both active and soft deleted records:**

$allProducts = Product::withTrashed()->get();

* **Restore a soft deleted record:**

$product = Product::withTrashed()->find(1);

$product->restore(); // sets deleted\_at to NULL

* **Permanently delete a record:**

$product = Product::withTrashed()->find(1);

$product->forceDelete(); // removes row from DB permanently

✅ This is the **standard way** to manage soft deletes in **Laravel 12**.

Do you want me to also give you a **real-world example for your ecommerce project** (like soft delete for orders, products, and users)?