Database

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.

Here is the detailed database schema for your e-commerce website:  
```sql  
CREATE TABLE users (  
 id SERIAL PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password\_hash VARCHAR(255) NOT NULL,  
 email VARCHAR(100) NOT NULL UNIQUE,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 phone\_number VARCHAR(20),  
 address TEXT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE products (  
 id SERIAL PRIMARY KEY,  
 title VARCHAR(200) NOT NULL,  
 description TEXT,  
 price DECIMAL(10,2) NOT NULL,  
 image\_url VARCHAR(255),  
 category VARCHAR(50),  
 subcategory VARCHAR(50),  
 brand VARCHAR(50),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE categories (  
 id SERIAL PRIMARY KEY,  
 name VARCHAR(50) NOT NULL,  
 parent\_category\_id INTEGER,  
 FOREIGN KEY (parent\_category\_id) REFERENCES categories(id)  
);  
  
CREATE TABLE orders (  
 id SERIAL PRIMARY KEY,  
 customer\_id INTEGER NOT NULL,  
 total\_amount DECIMAL(10,2) NOT NULL,  
 order\_status VARCHAR(50) NOT NULL DEFAULT 'pending',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (customer\_id) REFERENCES users(id)  
);  
  
CREATE TABLE order\_items (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 unit\_price DECIMAL(10,2) NOT NULL,  
 subtotal DECIMAL(10,2) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE payments (  
 id SERIAL PRIMARY KEY,  
 order\_id INTEGER NOT NULL,  
 payment\_method\_id INTEGER NOT NULL,  
 amount DECIMAL(10,2) NOT NULL,  
 transaction\_date DATE NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (order\_id) REFERENCES orders(id),  
 FOREIGN KEY (payment\_method\_id) REFERENCES payment\_methods(id)  
);  
  
CREATE TABLE payment\_methods (  
 id SERIAL PRIMARY KEY,  
 method\_name VARCHAR(50) NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE inventories (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 quantity INTEGER NOT NULL,  
 reorder\_level INTEGER NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
  
CREATE TABLE reviews (  
 id SERIAL PRIMARY KEY,  
 product\_id INTEGER NOT NULL,  
 reviewer\_name VARCHAR(50) NOT NULL,  
 review\_text TEXT NOT NULL,  
 rating INTEGER NOT NULL CHECK(rating >=1 AND rating <=5),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (product\_id) REFERENCES products(id)  
);  
```  
This database schema includes the following tables:  
  
\* Users: stores user information such as username, password hash, email, etc.  
\* Products: stores product information such as title, description, price, etc.  
\* Categories: stores category information such as name, parent category ID, etc.  
\* Orders: stores order information such as customer ID, total amount, order status, etc.  
\* Order Items: stores order item information such as order ID, product ID, quantity, etc.  
\* Payments: stores payment information such as order ID, payment method ID, amount, etc.  
\* Payment Methods: stores payment method information such as method name, etc.  
\* Inventories: stores inventory information such as product ID, quantity, reorder level, etc.  
\* Reviews: stores review information such as product ID, reviewer name, review text, rating, etc.  
  
The relationships between these tables include:  
  
\* One-to-One: between users and orders (a user can have multiple orders, but an order belongs to only one user)  
\* One-To-Many: between products and order items (a product can appear in multiple order items), between products and reviews (a product can have multiple reviews), and between users and orders (a user can place multiple orders)  
\* Many-To-Many: between products and categories (a product can belong to multiple categories, and a category can contain multiple products)  
  
I hope this helps! Let me know if you need any further assistance.