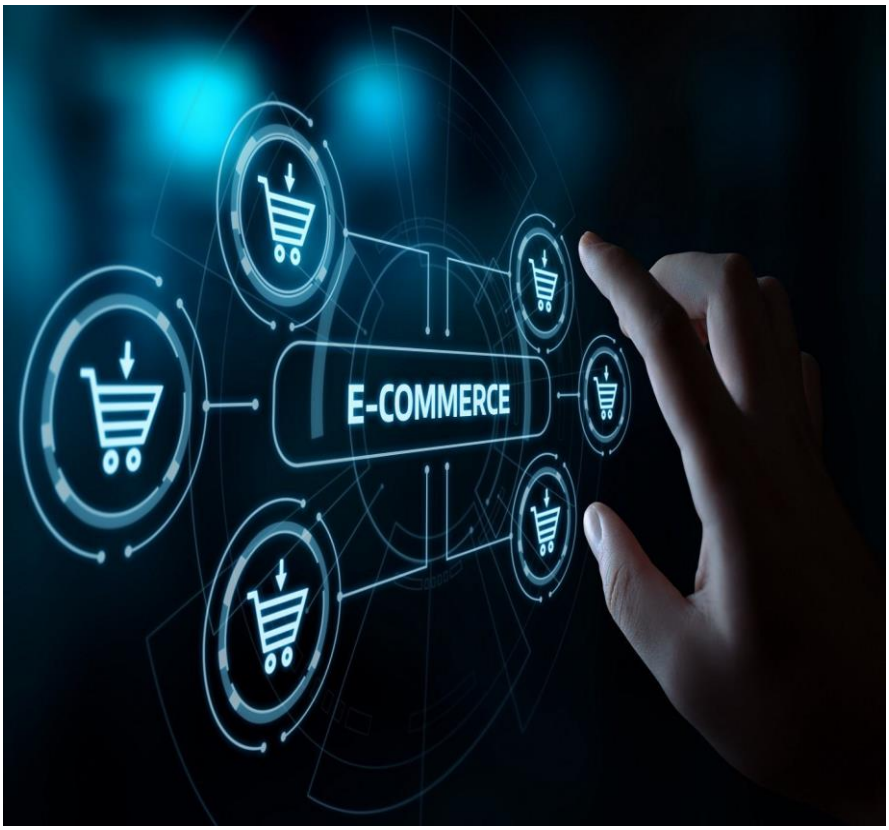
BIRMINGHAM CITY University

Advanced Database System
E-Commerce Database Management System



Module Code: **CMP7214**
Module Co-ordinator: Parnia Samimi

Group Members
Mohd Bilal *- 21161821*
Abdullah Aijaz *- 21169137*
Subhaan Nadaf *- 21172155*

TABLE OF CONTENTS

Contents **Pages**

1. Domain Description:

This report will provide a description of the ecommerce management system. The goal is to provide comprehensive understandings of how the online e-commerce website works, its entities and a plan of attack of how the database will be created based of the plan. Online shopping is the most advanced and used technique by which consumers buy goods, services, and other products from a seller in real time without the use of an intermediary service via the internet. The process of purchasing goods and services from merchants who sell on the Internet is known as E commerce(Electronic Commerce) also known as online shopping in most simplest form. Since the advent of the World Wide Web, merchants have sought to sell their wares to Internet users. Shoppers can visit web stores while sitting in front of their computers from the comfort of their own homes(REAL WORLD DESIGN AND IMPLEMENTATION DATABASE COURSE, 2013).

2. Scope of E commerce System:

The scope of the project is to build a relational database for all the core functionalities of an e-commerce website which explains how an e-commerce platform works.

The 6 major scope of Ecommerce are as follows:

1. Digital information exchange: The interchange of digitised information can represent two-party conversations, the coordination of the flow of products and services, or the transmission of electronic orders. These exchanges might take place between organisations or between people.

2. technologically enabled: E-commerce refers to technologically facilitated transactions. Web browsers maybe the most well-known of these technologically assisted client interactions. Other interfaces, such as automated teller machines (ATMs), are also classified as e-commerce. Previously, businesses handled transactions with consumers and markets solely via human connection; now, similar transitions may be handled using technology(Czurylo, 2021).

3. Customer loyalty: E-commerce allows firms to obtain categorised and tailored market information, which aids in client retention through quick order fulfilment and effective customer relationship management (CRM). End-to-end management of supply chains in e-commerce allows for the whole flow of market dynamics, resulting in profitable client retention.

4. Accounting: Because of the integrated database, financial accounting, treasury management, and asset management are as good as they could be in e-commerce. E-commerce makes financial strategies and planning decision easier.

5. Supplier connectivity: Supplier networks can be connected with EDI to enable just-in-time (JIT) inventory management, cutting inventory-carrying costs and increasing material and opportunity availability.

6. Encourage the interchange: E-commerce encompasses intra- and inter-organizational operations that facilitate the exchange of goods and services. E-commerce encompasses any electronic communications intra and inter - organizational information operations that promote marketplace transaction directly or indirectly. In this sense, we are discussing a phenomena that impacts both how business organisations interact with external parties such as customers, suppliers, partners, rivals, and markets, as well as how they manage activities, processes, and systems inside(Truitt, 2022).

3. Database Analysis

Pandemic-driven internet buying habits appear to be here to stay, entering in a data-driven future in which new channels and newcomers constantly raise the stakes on consumer and industry expectations.
The whole database and the system would be managed by Admin.
The project will aim to create a coherent database that works well with a managing all the details that go with running an e-commerce application. There will also be constraints added to the database which will stop things like NULL DATA or not validated data being entered into the database(Sumalatha, Vookanti and Vannala, 2021).

3.1 Business Situation:

There is a website that sells different kinds of products through an e-commerce platform. In this system, Admin(**id, name, email, password, created_by, updated_by**) is in-charge of managing other admins, Product Categories (**id, name, description, is_active, created_by**), Products (**id,category_id, name, stock quantity, description, sku, price, is_active, created_by**) and Discount Vouchers (**id, name, description, discount_percentage, is_active, created_by**). The Users (**id, name, email, password, contact number**) who have multiple addresses (**id, user_id, address_line1, address_line2, postal_code, city, country, is default**) stored .The Product Category (**id, name, description, is_active, created_by**) and Product (**id, name, stock quantity, description, sku, price, is_active, created_by**). The user can add products to the Shopping cart (**id, user_id**) which contains Cart Items (**id, product_id, price per piece, quantity**) and then chose the payment mode(Cash on Delivery/Pay via Card) checkout the order (**id, user_id, delivery status, total amount, payment status)** to place the order. order item

(**id,order_id,price_per_piece, quantity)** will be store in the separate entities. User can also apply active Discount Vouchers to the order to get discount.

3.2 List of Entity/Attributes:

A database management system's key keywords are entity and attributes. The key distinction between an entity and an attribute is that an entity is a physical thing, whereas attributes describe the qualities of an entity. The Entity might be both tangible and immaterial.
Any actual item is regarded an entity with self-existence, and these entities have their own traits and features known as attributes in relational databases. Attributes provide extra information about entities and aid in the investigation of their relationships within the given system. In an ER (Entity Relationship) model, attributes are always expressed in an elliptical form. In DBMS, attributes are classified as Simple, Composite, Single Valued, Multi-Valued, Stored, Derived, Key, and Complex(Editor, 2016).
In DBMS, attributes are usually represented in an elliptical form.
In this database there are in total 11 Entities and 76 attributes.

Entity: Admin:
- Attributes: id, name, email, password, created_by, updated_by.

Entity: Product Category:
- Attributes: id, name, description, is_active, created_by

Entity: Product:
- Attributes: id, category_id, name, description, stock_quantity, sku, price, is_active, created_by, edited_by, created_at, updated_at

Entity: Users:
- Attributes: id, name, email, password, contact_number, created_at, updated_at

Entity User Addresses:
- Attributes: id, user_id, address_line1, address_line2,postcode, city,country, is_default, created_at, updated_at

Entity User_carts:
- Attributes: id, user_id, created_at, updated_at

Entity: Cart Item:
- Attributes: id, cart_id, product_id, quantity, price_per_piece

Entity: Orders:
- Attributes: id, user_id, voucher_id, total_amount, payment_status, payment_mode, delivery_status, payment_date, address (address_line1, address_line2, postal_code, city, country), created_at, updated_at

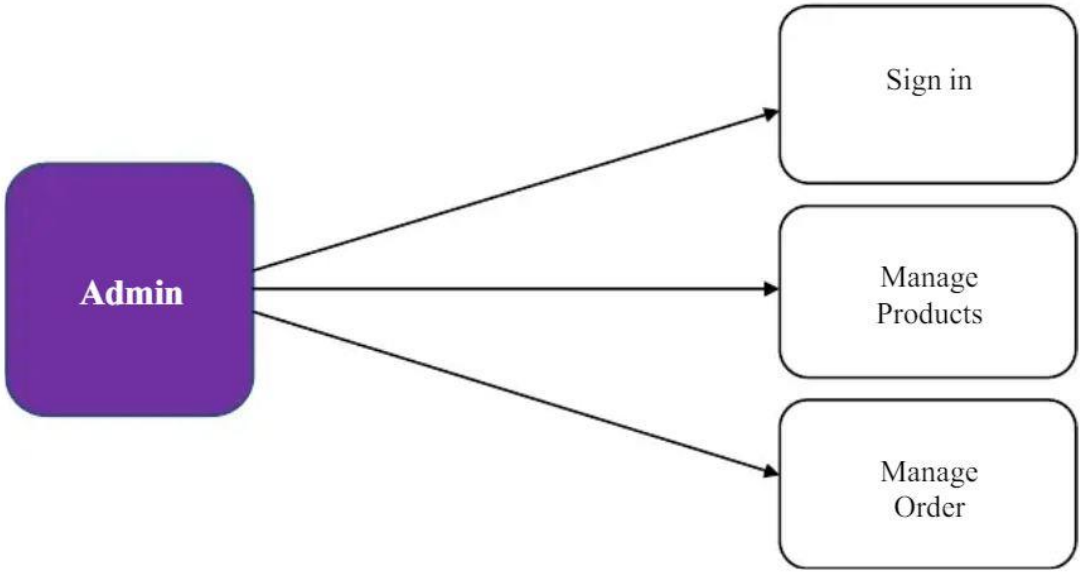
Entity: Order Items:
- Attributes: id, order_id, product_id, quantity, price_per_piece

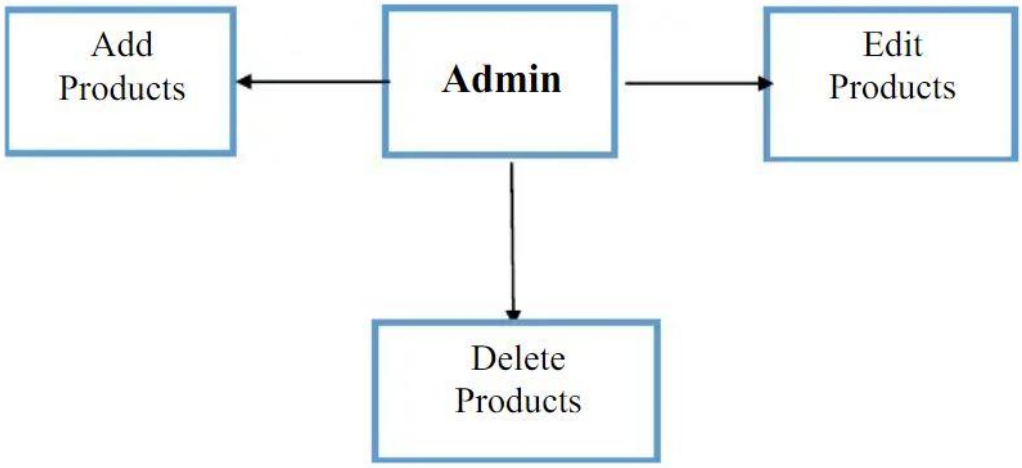
Entity: Payment:
- Attributes: id, order_id, amount, status, mode_of_payment, date_of_payment

Entity: Voucher:
- Attributes: id, name, description, discount_percentage, is_active, created_by

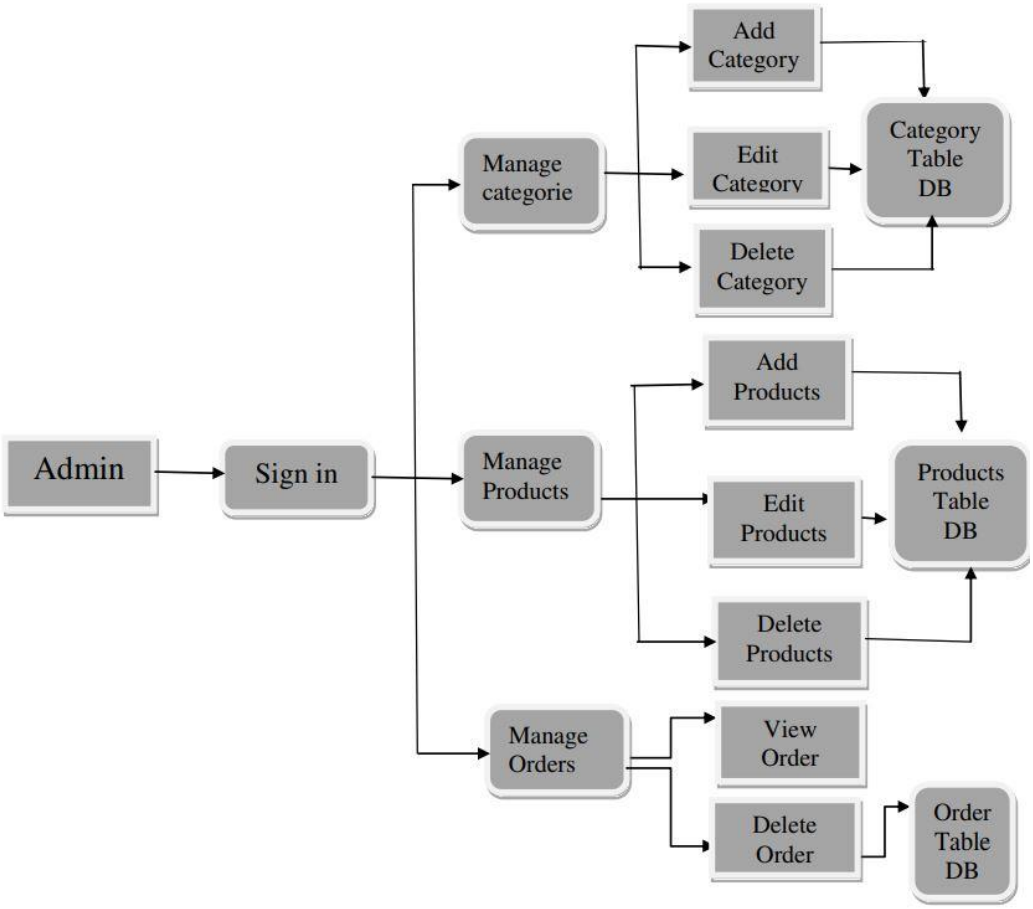
3.3 Business Rules:

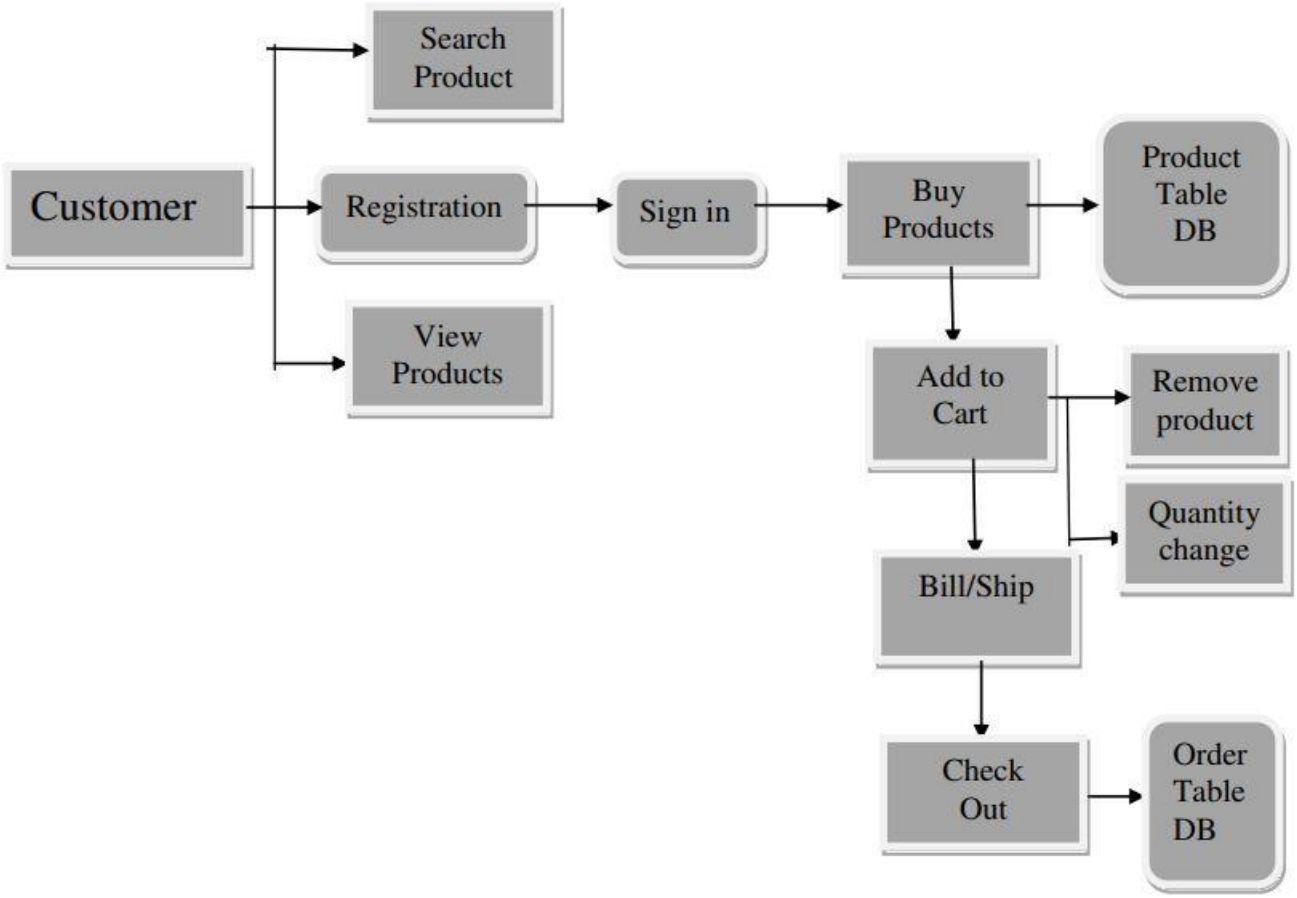
- Admin must manage and look after other admins.
- Admin must manage not less than one Product Category.
- Admin must manage not less than one Product.
- Admin can add discount vouchers.
- User can sign up and login using email address and password in this system.
- A user can add multiple delivery addresses but can choose only one specific address for delivery of the order.
- User can add products to the shopping cart.
- An order must be placed by choosing only one payment option.
- A user can order at least one and not more than the product's stock quantity.
- An order can include many products, but it must have at least one product.
- Only one active discount voucher can be applied to the order.
- Voucher is only valid during specific duration as created and updated by super admin.
- Only after the payment method is selected and processed a order can be processed for shipping.
- Order Addresses must contain hose/flat number and the postcode.
- Postcode must be a valid postcode.
- Shipping of the products is just limited across the United Kingdom.
- Admin must update the Sale prices for the products.
- All products under sale are limited until the stocks last.
- All products price are subject to changes by the admin.
- Admin is responsible to keep a track of the stocks of the product categories and has to update it regularly.
- Users are themselves responsible to cross check the details before placing the order.
- Admin must look after the shipping and dispatch of the orders.
- Voucher code can be used multiple times by a specific user for multiple orders.


Admin
Sign in
Manage Products
Manage Order


Add Products
Admin
Edit Products
Delete Products


View Order
Admin
Delete Order


Admin
Sign in
Manage categorie
Add Category
Edit Category
Delete Category
Category Table DB
Manage Products
Add Products
Edit Products
Delete Products
Products Table DB
Manage Orders
View Order
Delete Order
Order Table DB


Customer
Search Product
Registration
Sign in
Buy Products
Product Table DB
View Products
Add to Cart
Remove product
Quantity change
Bill/Ship
Check Out
Order Table DB

3.4 Simple Relationships:

According to the relational paradigm, logical data structures (data tables, views, and indexes) are distinct from physical storage structures. Because of this separation, database managers may adjust physical data storage without influencing logical data access.
In the framework of databases, a relationship is basically a scenario that occurs between two relational database tables when one table includes a foreign key that references the primary key of the other table. Relationships enable relational databases to divide and store data in various tables while also connecting distinct data pieces. A relationship may be defined as any affiliation, linkage, or connection between entities of importance to the company. A connection is often denoted by a verb that connects two or more items(Silberschatz, Korth and Sudarshan, n.d.).

[Admin] M	<manages>	N [Admin]
[Admin] M	<manages>	N [Product Category]
[Admin] M	<manages>	N [Product]
[Admin] M	<manages>	N [Voucher]
[Product Category] 1	<has>	N [Product]
[User] 1	<add product to>	1 [Shopping Cart]
[Shopping Cart] 1	<made of>	N [Cart Item]
[Product] 1	<added to>	N [Cart Item]
[User] 1	<places>	N [Order]
[Order] 1	<made of>	N [Order Item]
[Order] 1	<has>	1 [Payment]
[Voucher] 1	<applied>	1 [Order]

3.5 Connectivities and Cardinalities:

A connection's connectivity defines the mapping of connected data items in the relationship. The connection values are "one" or "many." A relationship's cardinality is the actual number of connected occurrences for each of the two entities.

User has one shopping cart
One to One Relation.

User can add different Address.
One to Many Relation

User can add multiple products to cart.
One to Many Relation

Admin manages Admin
Many to Many

Admin manages Product Category
Many to Many.

Admin manages Product

Many to Many

Admin manages Vouchers
Many to Many

Product Category has Many Products
Many to Many

Order will have payment details.
One to One

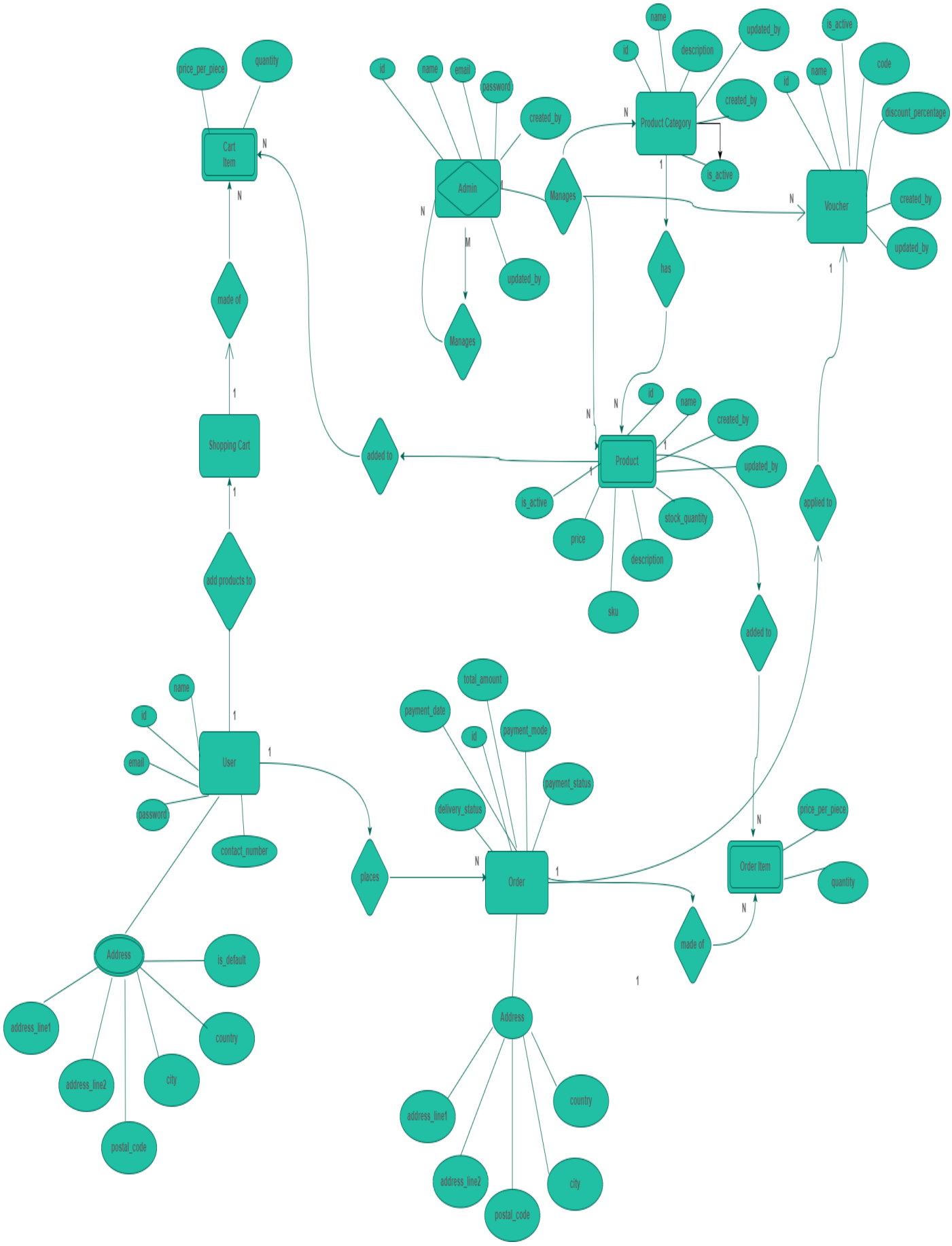
Voucher will be applied to the order.
One to One

One user can have multiple orders.
One to Many.

Order can have multiple order items
One to Many

Voucher has one unique code for orders

4. ERD


price_per_piece
quantity
Cart Item
made of
Shopping Cart
add products to
User
id
name
email
password
contact_number
Address
address_line1
address_line2
city
postal_code
country
is_default
places
Order
total_amount
payment_date
payment_mode
payment_status
delivery_status
Address
address_line1
address_line2
city
postal_code
country
made of
Order Item
price_per_piece
quantity
added to
Product
id
name
created_by
updated_by
price
stock_quantity
description
sku
is_active
applied to
Voucher
id
name
code
discount_percentage
created_by
updated_by
is_active
has
Product Category
id
name
description
created_by
updated_by
is_active
Manages
Admin
id
name
email
password
created_by
updated_by
Manages
added to

5.Database Normalization

Following the initial exercise of selecting the data items that should be contained in the relational database, establishing their relationships, and defining the rows and columns inside each table, database normalisation is often a refining process.

Database normalisation follows a few guidelines. Each rule is referred to as a "normal form." The database is considered to be in "first normal form" if the first rule is followed. The database is regarded to be in "third normal form" if the first three requirements are followed. Although higher degrees of normalisation are feasible, the third normal form is thought to be the most important for most purposes.

Real-world circumstances, like many technical rules and standards, may not always allow for flawless compliance. Normalization, in general, necessitates the creation of new tables, which some customers find inconvenient. If you decide to break the first three criteria of normalisation, make sure your system anticipates any potential difficulties, such as duplicate data and incoherent dependencies.

In the system normalization in made on 6 entities.

1. Admin

id	name	email	password	created_by	updated_by	created_at	updated_by
1	admin1	admin1@ecommerce.co.uk	12345678			10-10-2021	
2	admin2	admin2@ecommerce.co.uk	12345678	1		10-12-2021	
3	admin3	admin3@ecommerce.co.uk	12345678	1		10-12-2021	
4	admin4	admin4@ecommerce.co.uk	12345678	1		10-12-2021	
5	admin5	admin5@ecommerce.co.uk	12345678	1		10-12-2021	

1NF

id is the PK

Entity is already in 1NF form.

2NF

Entity is already in 2NF form and and all attributes depend on PK

3NF

Entity is in 3NF, there are no transitive dependencies

2.Product Categories

id	name	description	is_active	created_by	updated_by	created_at	updated_at
1	Kids	All clothing and accessories related to kids.	1	1		14/08/2022	
2	Men	All clothing and accessories related to Men	1	1		14/08/2022	
3	Women	All clothing and accessories related to Women	1	1		14/08/2022	
4	Electronics	All Electronic Items	1	1		14/08/2022	
5	Summer Sale	All products under summer sale	1	1		14/08/2022	

1NF

id is the PK

Entity is already in 1NF form.

2NF

Entity is already in 2NF form and and all attributes depend on PK

3NF

Entity is in 3NF, there are no transitive dependencies

3. Products

id	category	name	description	is_active	created_by	updated_by	created_at	updated_at
1	Kids	Soft Toys	all kid toys including in soft toys	1	1		26/08/2022	
2	Men	Clothing top wear	T shirts and Shirts	1	1		26/08/2022	
3	Women	Topweear	Tops, t shirts and shirts	1	1		26/08/2022	
4	Electronics	Mobiles	Different Brands and accessories	1	1		26/08/2022	
5	Summer Sale	Electronics	Electrical Applicances	1	1		26/08/2022	

1NF

id is the PK

Entity is already in 1NF form.

2NF

id	category_id	name	description	is_active	created_by	updated_by	created_at	updated_at
1	1	Soft Toys	all kid toys including in soft toys	1	1		26/08/2022	
2	2	Clothing top wear	T shirts and Shirts	1	1		26/08/2022	
3	3	Topweear	Tops, t shirts and shirts	1	1		26/08/2022	
4	4	Mobiles	Different Brands and accessories	1	1		26/08/2022	
5	5	Electronics	Electrical Applicances	1	1		26/08/2022	

category_id is the foreign key

3NF

Entity is in 3NF, there are no transitive dependencies

4.Users

id	name	email	password	contact_number	created_at	updated_at	addresses
1	user11	user11@bcu.ac.uk	12345678	7894654321	08-10-2022		103 east avenue john road b27 1qw Birmingham United Kingdom, 8 cardigan street Birmingham United Kingdom
2	user12	user12@bcu.ac.uk	12345678	7894654321	08-10-2022		250 west avenue john road b13 asf Birmingham United Kingdom, 8 broadway street Birmingham United Kingdom
3	user13	user13@bcu.ac.uk	12345678	7894654321	08-10-2022		251 south avenue broadroad B13 POT Birmingham United Kingdom, 10 cape hill street Birmingham United Kingdom
4	user14	user14@bcu.ac.uk	12345678	7894654321	08-10-2022		103 east avenue john road b27 1qw Birmingham United Kingdom, 8 cardigan street Birmingham United Kingdom
5	user15	user15@bcu.ac.uk	12345678	7894654321	08-10-2022		987 east avenue peter road B6Y OLY Birmingham United Kingdom

1NF

id	name	email	password	contact_number	created_at	updated_at	Address
1	user11	user11@bcu.ac.uk	12345678	7894654321	08-10-2022		103 east avenue john road b27 1qw Birmingham United Kingdom
1	user11	user11@bcu.ac.uk	12345678	7894654321	08-10-2022		8 cardigan street Birmingham B43 8KT United Kingdom
2	user12	user12@bcu.ac.uk	12345678	7894654321	08-10-2022		250 west avenue john road b13 asf Birmingham United Kingdom
2	user12	user12@bcu.ac.uk	12345678	7894654321	08-10-2022		8 broadway street Birmingham B18 KJG United Kingdom
3	user13	user13@bcu.ac.uk	12345678	7894654321	08-10-2022		10 cape hill street Birmingham B65 8ZQ United Kingdom
3	user13	user13@bcu.ac.uk	12345678	7894654321	08-10-2022		251 south avenue broadroad B13 POT Birmingham United Kingdom
4	user14	user14@bcu.ac.uk	12345678	7894654321	08-10-2022		103 east avenue john road b27 1qw Birmingham United Kingdom, 8 cardigan street B43 8KT Birmingham United Kingdom
4	user14	user14@bcu.ac.uk	12345678	7894654321	08-10-2022		10 cardigan street B43 8KT Birmingham United Kingdom
5	user15	user15@bcu.ac.uk	12345678	7894654321	08-10-2022		987 east avenue peter road B6Y OLY Birmingham United Kingdom

Each column is now have single value

2NF

users table

id is PK in the users table

id	name	email	password	contact_number	created_at	updated_at
1	user11	user11@bcu.ac.uk	12345678	7894654321	08-10-2022	
2	user12	user12@bcu.ac.uk	12345678	7894654321	08-10-2022	
3	user13	user13@bcu.ac.uk	12345678	7894654321	08-10-2022	
4	user14	user14@bcu.ac.uk	12345678	7894654321	08-10-2022	
5	user15	user15@bcu.ac.uk	12345678	7894654321	08-10-2022	

users_addresses table

id	user_id	address_line_1	address_line_2	postal code	city	country	is_default
2	1	103 east avenue john road		b27 1qw	Birmingham	United Kingdom	1
3	1	8 cardigan street		b27 1qw	Birmingham	United Kingdom	0
4	2	250 west avenue		b27 1qw	Birmingham	United Kingdom	1
3	2	8 broadway street		b27 1qw	Birmingham	United Kingdom	0
6	3	10 cape hill street		b27 1qw	Birmingham	United Kingdom	1
7	3	251 south avenue		b27 1qw	Birmingham	United Kingdom	0
8	4	103 east avenue	john road	b27 1qw	Birmingham	United Kingdom	1
9	4	8 cardigan street		b27 1qw	Birmingham	United Kingdom	0
10	5	987 east avenue	Peter Road	B6Y OLY	Birmingham	United Kingdom	1

users address which was the mulitvalued attribute is now divided into different table. Id is the PK for the user_address table and user_id is the foreign key.
Each column is now have single value

3NF

Entities are in 3NF, there are no transitive dependencies

5. Vouchers

id	name	code	discount percenage	is_active	created_by	updated_by	created_at	updated_at
1	End of season sale	EOSS	30	1	1		27-08-2022 20:28	
2	Summer Sale 2022	SS22	40	1	1		27-08-2022 20:28	

1NF

id is the PK

Entity is already in 1NF form.

2NF

Entity is already in 2NF form.

3NF

Entity is in 3NF, there are no transitive dependencies

6.Orders

id	user	voucher	payment_mode	payment_status	delivery_status	total_amount	payment_date	created_at	updated_at	address_line_1	address_line_2	postal_code	city	country
1	john	Summer Sale 2022	CARD	PAID	DELIVERED	1000	25/08/2022	24/05/2022		103 East Aveneue	Accocks Jazz Road	B26 KJH	Birmingham	United Kingdom
2	doe	End of season sale	COD	PAID	DELIVERED	2000	26/08/2022	24/05/2022		510 South Aveneue	Green Road	B78 0KH	Birmingham	United Kingdom
3	kim	Summer Sale 2022	CARD	PAID	DELIVERED	3000	26/08/2022	24/05/2022		625 Richard Aveneue	Cape Hill	B90 8JK	Birmingham	United Kingdom

1NF

id	user_id	voucher_id	payment_mode	payment_status	delivery_status	total_amount	payment_date	created_at	updated_at	address_line_1	address_line_2	postal_code	city	country
1	1	2	CARD	PAID	DELIVERED	1000	25/08/2022	24/05/2022		103 East Aveneue	Accocks Jazz Road	B26 KJH	Birmingham	United Kingdom
2	2	1	COD	PAID	DELIVERED	2000	26/08/2022	24/05/2022		510 South Aveneue	Green Road	B78 0KH	Birmingham	United Kingdom
3	3	1	CARD	PAID	DELIVERED	3000	26/08/2022	24/05/2022		625 Richard Aveneue	Cape Hill	B90 8JK	Birmingham	United Kingdom

id is the PK, user_id is the foreign key for the users, and voucher_id is the foreign key for vouchers

2NF

users table

id	user_id	voucher_id	payment_mode	payment_status	delivery_status	total_amount	payment_date	created_at	updated_at					
1	1	2	CARD	PAID	DELIVERED	1000	25/08/2022	24/05/2022		103 East Aveneue	Accocks Jazz Road	B26 KJH	Birmingham	United Kingdom
2	2	1	COD	PAID	DELIVERED	2000	26/08/2022	24/05/2022		510 South Aveneue	Green Road	B78 0KH	Birmingham	United Kingdom
3	3	1	CARD	PAID	DELIVERED	3000	26/08/2022	24/05/2022		625 Richard Aveneue	Cape Hill	B90 8JK	Birmingham	United Kingdom

order_addresses

id	order_id	address_line_1	address_line_2	postal_code	city	country
1	1	103 East Aveneue	Accocks Jazz Road	B26 KJH	Birmingham	United Kingdom
2	2	510 South Aveneue	Green Road	B78 0KH	Birmingham	United Kingdom
3	3	625 Richard Aveneue	Cape Hill	B90 8JK	Birmingham	United Kingdom

id is PK of the order_addresses table and order_id is the foreign key of the

3NF

Entity is in 3NF, there are no transitive dependencies

6.Database Implementation

Table Creation:

1- Creation of the Users Table

```
CREATE TABLE `ecommerce`.`users` (`id` INT NOT NULL AUTO_INCREMENT ,
`name` VARCHAR(50) NOT NULL , `email` VARCHAR(50) NOT NULL ,
`password` VARCHAR(50) NOT NULL , `contact_number` VARCHAR(15) NOT
NULL , `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,
`updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,
PRIMARY KEY (`id`), UNIQUE (`email`));
```

2- Creation of the User Addresses Table

```
CREATE TABLE `ecommerce`.`user_addresses` (`id` INT NOT NULL AUTO_INCREMENT
,`user_id` INT NOT NULL, `address_line1` VARCHAR(50) NOT NULL , `address_line2`
VARCHAR(50) NOT NULL , `postal_code` VARCHAR(20) NOT NULL , `city`
VARCHAR(20) NOT NULL , `country` VARCHAR(20) NOT NULL , `is_default` BOOLEAN
NOT NULL DEFAULT TRUE , `created_at` TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP , `updated_at` TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP , PRIMARY KEY (`id`));
```

```
ALTER TABLE `user_addresses` ADD FOREIGN KEY (`user_id`) REFERENCES
`users`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

3- Creation of the Admins Table

```
CREATE TABLE `ecommerce`.`admins` (`id` INT NOT NULL AUTO_INCREMENT ,
`name` VARCHAR(50) NOT NULL , `email` VARCHAR(50) NOT NULL, `password`
VARCHAR(50) NOT NULL , `created_by` INT NULL , `updated_by` INT NULL ,
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , `updated_at`
TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`id`),
UNIQUE (`email`));
```

```
ALTER TABLE `admins` ADD FOREIGN KEY (`created_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT; ALTER TABLE `admins` ADD FOREIGN KEY (`updated_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

4- Creation of the Product Categories Table

```
CREATE TABLE `ecommerce`.`product_categories` (`id` INT NOT NULL AUTO_INCREMENT , `name` VARCHAR(50) NOT NULL , `description` VARCHAR(255) NOT NULL , `is_active` BOOLEAN NOT NULL DEFAULT TRUE , `created_by` INT NOT NULL , `updated_by` INT NULL , `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , `updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`id`));
```

```
ALTER TABLE `product_categories` ADD FOREIGN KEY (`created_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT; ALTER TABLE `product_categories` ADD FOREIGN KEY (`updated_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

5- Creation of the Products Table

```
CREATE TABLE `ecommerce`.`products` (`id` INT NOT NULL AUTO_INCREMENT , `category_id` INT NOT NULL , `name` VARCHAR(50) NOT NULL , `description` VARCHAR(255) NOT NULL , `stock_quantity` INT UNSIGNED NOT NULL , `sku` VARCHAR(20) NOT NULL , `price` DECIMAL NOT NULL , `created_by` INT NOT NULL ,`updated_by` INT NULL , `is_active` BOOLEAN NOT NULL DEFAULT TRUE , `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , `updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`id`), UNIQUE (`sku`));
```

```
ALTER TABLE `products` ADD FOREIGN KEY (`category_id`) REFERENCES `product_categories`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT; ALTER TABLE `products` ADD FOREIGN KEY (`created_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT; ALTER TABLE `products` ADD FOREIGN KEY (`updated_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

6- Creation of the User Carts Table

```
CREATE TABLE `ecommerce`.`user_carts` (`id` INT NOT NULL AUTO_INCREMENT , `user_id` INT NOT NULL ,`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , `updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (`id`));

ALTER TABLE `user_carts` ADD FOREIGN KEY (`user_id`) REFERENCES `users`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

7- Creation of the Cart Items Table

```
CREATE TABLE `ecommerce`.`cart_items` (`id` INT NOT NULL AUTO_INCREMENT , `cart_id` INT NOT NULL , `product_id` INT NOT NULL , `quantity` INT UNSIGNED NOT NULL , `price_per_piece` DECIMAL NOT NULL , PRIMARY KEY (`id`));

ALTER TABLE `cart_items` ADD FOREIGN KEY (`cart_id`) REFERENCES `user_carts`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT; ALTER TABLE `cart_items` ADD FOREIGN KEY (`product_id`) REFERENCES `products`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

8- Creation of the Vouchers Table

```
CREATE TABLE `ecommerce`.`vouchers` (`id` INT NOT NULL AUTO_INCREMENT , `name` VARCHAR(50) NOT NULL , `code` VARCHAR(10) NOT NULL , `discount_percentage` FLOAT UNSIGNED NOT NULL , `is_active` BOOLEAN NOT NULL DEFAULT TRUE , `created_by` INT NOT NULL , `updated_by` INT NULL , `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , `updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`id`));

ALTER TABLE `vouchers` ADD FOREIGN KEY (`created_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT; ALTER TABLE `vouchers` ADD FOREIGN KEY (`updated_by`) REFERENCES `admins`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

9- Creation of the Orders Table

```
CREATE TABLE `ecommerce`.`orders` (`id` INT NOT NULL AUTO_INCREMENT ,
`user_id` INT NOT NULL ,`voucher_id` INT NULL, `payment_mode` ENUM('cod',
'card') NOT NULL DEFAULT 'cod' , `payment_status` ENUM('pending', 'paid') NOT
NULL DEFAULT 'pending' , `delivery_status` ENUM('in_progress', 'despatced',
'delivered') NOT NULL DEFAULT 'in_progress' ,`total_amount` DECIMAL NOT NULL ,
`payment_date` DATETIME NULL ,`created_at` TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP , `updated_at` TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP, PRIMARY KEY (`id`));
```

```
ALTER TABLE `orders` ADD FOREIGN KEY (`user_id`) REFERENCES `users`(`id`) ON
DELETE RESTRICT ON UPDATE RESTRICT;
ALTER TABLE `orders` ADD FOREIGN KEY (`voucher_id`) REFERENCES `vouchers`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

10- Creation of the Order Addresses Table

```
CREATE TABLE `ecommerce`.`order_addresses` (`id` INT NOT NULL
AUTO_INCREMENT ,`order_id` INT NOT NULL, `address_line1` VARCHAR(50) NOT
NULL , `address_line2` VARCHAR(50) NOT NULL , `postal_code` VARCHAR(20) NOT
NULL , `city` VARCHAR(20) NOT NULL , `country` VARCHAR(20) NOT NULL ,
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , `updated_at`
TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`id`));
```

```
ALTER TABLE `order_addresses` ADD FOREIGN KEY (`order_id`) REFERENCES
`orders`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

11- Creation of the Order Items Table

```
CREATE TABLE `ecommerce`.`order_items` (`id` INT NOT NULL AUTO_INCREMENT ,
`order_id` INT NOT NULL , `product_id` INT NOT NULL , `quantity` INT UNSIGNED
NOT NULL , `price_per_piece` DECIMAL NOT NULL , PRIMARY KEY (`id`));
```

```
ALTER TABLE `order_items` ADD FOREIGN KEY (`order_id`) REFERENCES `orders`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;
ALTER TABLE `order_items` ADD FOREIGN KEY (`product_id`) REFERENCES
`products`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

Inserting Data in Tables.

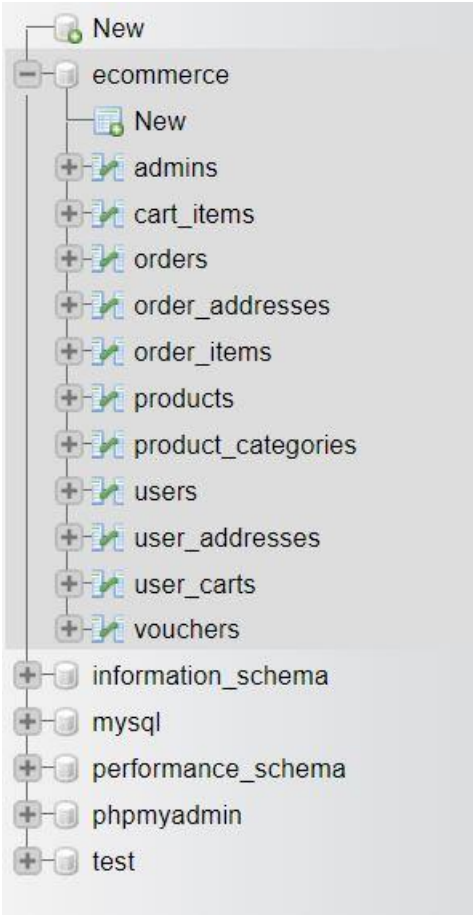


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	admins	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	64.0 KiB	-
☐	cart_items	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	orders	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	order_addresses	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	order_items	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	products	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	64.0 KiB	-
☐	product_categories	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	user_addresses	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	user_carts	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	vouchers	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
	11 tables	Sum	0	InnoDB	utf8mb4_general_ci	480.0 KiB	0 B

Inserting Data Into Admin Table

```
INSERT INTO `admins` (`id`, `name`, `email`, `password`, `created_by`, `updated_by`, `created_at`, `updated_at`) VALUES (NULL,
'admin11', 'admin11@ecommerce.co.uk', '2936bfe965123c61adce74a24201502e', NULL, NULL, '2022-08-27 19:24:26', '2022-08-27 19:24:26'),
(NULL, 'admin12', 'admin12@ecommerce.co.uk', '3761611798c1d7f32921deb6a6cabb78', NULL, NULL, '2022-08-27 19:24:26', '2022-08-27
19:24:26'), (NULL, 'admin13', 'admin13@ecommerce.co.uk', '6047f6092ea16137c8bbe5b62e02b2af', NULL, NULL, '2022-08-27 19:27:03', '2022-
08-27 19:27:03'), (NULL, 'admin14', 'admin14@ecommerce.co.uk', 'f6e30368cd63f36281a775676c52fa7f', NULL, NULL, '2022-08-27 19:27:03',
'2022-08-27 19:27:03'), (NULL, 'admin15', 'admin15@ecommerce.co.uk', '2e42c1973238078b456318d1c9770a3a', NULL, NULL, '2022-08-27
19:28:53', '2022-08-27 19:28:53'), (NULL, 'admin16', 'admin16@ecommerce.co.uk', '4b657d6e980b2f466eb3ebb7021dde32', NULL, NULL, '2022-
08-27 19:28:53', '2022-08-27 19:28:53'), (NULL, 'admin17', 'admin17@ecommerce.co.uk', '7ed05ae3c861b8e9b8093f735e32369f', NULL, NULL,
'2022-08-27 19:30:16', '2022-08-27 19:30:16'), (NULL, 'admin18', 'admin18@ecommerce.co.uk', 'c4e426591145c1a547ab3cade3c1bce8', NULL,
NULL, '2022-08-27 19:30:16', '2022-08-27 19:30:16'), (NULL, 'admin19', 'admin19@ecommerce.co.uk', '0a7377e9db8b67a32c04183dacdee114',
NULL, NULL, '2022-08-27 19:31:35', '2022-08-27 19:31:35'), (NULL, 'admin20', 'admin20@ecommerce.co.uk',
'a4436f6eb3ca231b5c8bb4926fc2dade', NULL, NULL, '2022-08-27 19:31:35', '2022-08-27 19:31:35');
```

id
name
email
password
created_by
updated_by
created_at
updated_at

Inserting Data Into Users Table

```
INSERT INTO `users` (`id`, `name`, `email`, `password`, `contact_number`, `created_at`, `updated_at`) VALUES (NULL, 'user11',
'user11@bcu.ac.uk', MD5('buc@11user'), '+447894654321', current_timestamp(), current_timestamp()), (NULL, 'user12',
'user12@bcu.ac.uk', MD5('bcu@12user'), '+447633641313', current_timestamp(), current_timestamp()), (NULL, 'user13',
'user13@bcu.ac.uk', MD5('bcu@13user'), '+447568465468', current_timestamp(), current_timestamp()), (NULL, 'user14',
'user14@bcu.ac.uk', MD5('bcu@14user'), '+447568452316', current_timestamp(), current_timestamp()), (NULL, 'user15',
'user15@bcu.ac.uk', MD5('bcu@15user'), '+447645132576', current_timestamp(), current_timestamp()), (NULL, 'user16',
'user16@bcu.ac.uk', MD5('bcu@16user'), '+447613294527', current_timestamp(), current_timestamp()), (NULL, 'user17',
'user17@bcu.ac.uk', MD5('bcu@17user'), '+447613495846', current_timestamp(), current_timestamp()), (NULL, 'user18',
'user18@bcu.ac.uk', MD5('bcu@18user'), '+4475134698513', current_timestamp(), current_timestamp()), (NULL, 'user19',
'user19@bcu.ac.uk', MD5('bcu@19user'), '+447215469231', current_timestamp(), current_timestamp()), (NULL, 'user20',
'user20@bcu.ac.uk', MD5('bcu@20user'), '+447132546981', current_timestamp(), current_timestamp());
```

id
name
email
password
contact_number
created_at
updated_at

Inserting Data Into User Addresses Table

```
INSERT INTO `user_addresses` (`id`, `user_id`, `address_line1`, `address_line2`, `postal_code`, `city`, `country`, `is_default`, `created_at`, `updated_at`) VALUES (NULL, '1', '103 east avenue', 'john road', 'b27 1qw', 'birmingham', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '2', '8 curzon street', 'curzon road', 'b5 1ew', 'birmingham', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '3', '55 park lane', 'park street', 'p12 2nl', 'paddington', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '4', '27-B almero student beech gardens', 'elvetham road', 'b15 2nl', 'birmgham', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '5', '5 street avenue', 'park lane', 'w12 2pn', 'london', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '6', '24 harrow lane', '', 'w25 2pl', 'london ', 'unitedkingom', '1', current_timestamp(), current_timestamp()), (NULL, '7', '222 east side building', 'edgbaston street', 'b16 2ll', 'birmingham', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '8', '15 park side close', 'avenue park', 'e21 2ws', 'london', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '9', '12 selly park', 'pershore road', 'm2 1ew', 'manchester', 'united kingdom', '1', current_timestamp(), current_timestamp()), (NULL, '10', '12 colway avenue', 'harry road', 'm12 2ka', 'manchester', 'united kingdom', '1', current_timestamp(), current_timestamp());
```

id
user_id
address_line1
address_line2
postal_code
city
country
is_default
created_at
updated_at

Insert Data Into Product Categories Table

```
INSERT INTO `product_categories` (`id`, `name`, `description`, `is_active`, `created_by`, `updated_by`, `created_at`, `updated_at`) VALUES (NULL, 'Kids', 'All clothing and accessories related to kids.', '1', '26', '26', current_timestamp(), current_timestamp()), (NULL, 'Men', 'All clothing and accessories related to Men', '1', '27', '27', current_timestamp(), current_timestamp()), (NULL, 'Women', 'All clothing and accessories related to Women', '1', '30', '30', current_timestamp(), current_timestamp()), (NULL, 'Electronics', 'All Electronic Items', '1', '31', '31', current_timestamp(), current_timestamp()), (NULL, 'Summer Sale', 'All products under summer sale', '1', '25', '25', current_timestamp(), current_timestamp());
```

id
name
description
is_active
created_by
updated_by
created_at
updated_at

Insert Data Into Products Table

```
INSERT INTO `products` (`id`, `category_id`, `name`, `description`, `stock_quantity`, `sku`, `price`, `created_by`, `updated_by`, `is_active`,
`created_at`, `updated_at`) VALUES (NULL, '1', 'Soft Toys', 'all kid toys including in soft toys', '1000', 'ST123', '10', '29', '29', '1',
current_timestamp(), current_timestamp()), (NULL, '1', 'Hard Toys', 'all kid toys including in hard toys', '15000', 'sku354', '20', '29', '29', '1',
current_timestamp(), current_timestamp()), (NULL, '2', 'Clothing top wear', 'T shirts and Shirts', '3000', 'sku421', '25', '32', '32', '1',
current_timestamp(), current_timestamp()), (NULL, '2', 'clothing bottomwear', 'Jeans and Trousers', '5000', 'sku456', '30', '26', '26', '1',
current_timestamp(), current_timestamp()), (NULL, '2', 'Sports topwear', 'T shirts', '2500', 'sku849', '30', '27', '27', '1', current_timestamp(),
current_timestamp()), (NULL, '2', 'Sportswear bottomwear', 'Tracks and trainers', '3200', 'sku322', '45', '28', '28', '1', current_timestamp(),
current_timestamp()), (NULL, '2', 'Shoes', 'Casual and Sneakers', '6000', 'sku520', '60', '34', '34', '1', current_timestamp(), current_timestamp()),
(NULL, '2', 'Accessories', 'Watches, Belts, Wallets', '3000', 'sku444', '30', '27', '27', '1', current_timestamp(), current_timestamp()), (NULL, '3',
'Topweear', 'Tops, t shirts and shirts', '5000', 'sku566', '25', '29', '29', '1', current_timestamp(), current_timestamp()), (NULL, '3', 'Bottomwear',
'Jeans, Trousers and Skirts', '6000', 'sku221', '20', '29', '29', '1', current_timestamp(), current_timestamp()), (NULL, '3', 'Accessories', 'Handbags,
Earing, Makeup', '5000', 'sku554', '30', '28', '28', '1', current_timestamp(), current_timestamp()), (NULL, '3', 'Footwear', 'Shoes, Sandals, clogs',
'5000', 'sku222', '30', '31', '31', '1', current_timestamp(), current_timestamp()), (NULL, '4', 'Mobiles', 'Different Brands and accessories', '6000',
'sku201', '400', '32', '32', '1', current_timestamp(), current_timestamp()), (NULL, '4', 'Home Applicances', 'Electrical Applicances', '3000', 'sku746',
'300', '29', '29', '1', current_timestamp(), current_timestamp()), (NULL, '4', 'Laptops', 'Dekstops and Pcs', '200', 'sku999', '500', '29', '29', '1',
current_timestamp(), current_timestamp()), (NULL, '4', 'Printer ', 'Printers and Scanners', '300', 'sku516', '200', '34', '34', '1', current_timestamp(),
current_timestamp()), (NULL, '5', 'Mens Clothing', 'Selected Items', '3000', 'skuSS22', '20', '25', '25', '1', current_timestamp(), current_timestamp()),
(NULL, '1', 'Toys', 'Soft and Hard Toys', '2000', 'skuSS22', '15', '25', '25', '1', current_timestamp(), current_timestamp()), (NULL, '5', 'Women',
'Selected categories and items', '6000', 'skuSS22', '30', '25', '25', '1', current_timestamp(), current_timestamp()), (NULL, '4', 'Electronics',
'Selected Items', '3000', 'skuSS22', '400', '25', '25', '1', current_timestamp(), current_timestamp());
```

id
category_id
name
description
stock_quantity
sku
price
created_by
updated_by
is_active
created_at
updated_at

<<

Insert Into Vouchers Table

```
INSERT INTO `vouchers` (`id`, `name`, `code`, `discount_percentage`, `is_active`, `created_by`, `updated_by`, `created_at`, `updated_at`) VALUES (NULL,
'End of season sale', 'EOSS', '30', '1', '32', '32', current_timestamp(), current_timestamp()), (NULL, 'Summer Sale 2022', 'SS22', '40', '1', '26', '26',
current_timestamp(), current_timestamp());
```

id
name
code
discount_percentage
is_active
created_by
updated_by
created_at
updated_at

Insert Into Cart_items Table

```
INSERT INTO `cart_items` (`id`, `cart_id`, `product_id`, `quantity`, `price_per_piece`) VALUES (NULL, '1', '7', '1', '30'), (NULL, '3', '9', '2', '40'),
(NULL, '2', '14', '1', '300'), (NULL, '5', '1', '5', '10'), (NULL, '7', '6', '2', '40');
```

id
cart_id
product_id
quantity
price_per_piece

Insert into Orders Table

```
INSERT INTO `orders` (`id`, `user_id`, `voucher_id`, `payment_mode`, `payment_status`, `delivery_status`, `total_amount`, `payment_date`, `created_at`, `updated_at`) VALUES (NULL, '1', NULL, 'cod', 'pending', 'in_progress', '20', NULL, current_timestamp(), current_timestamp()), (NULL, '2', NULL, 'card', 'paid', 'in_progress', '30', NULL, current_timestamp(), current_timestamp()), (NULL, '3', '2', 'cod', 'pending', 'in_progress', '60', NULL, current_timestamp(), current_timestamp()), (NULL, '7', '2', 'card', 'paid', 'despatced', '70', NULL, current_timestamp(), current_timestamp()), (NULL, '10', NULL, 'card', 'paid', 'delivered', '200', NULL, current_timestamp(), current_timestamp());
```

id
user_id
voucher_id
payment_mode
payment_status
delivery_status
total_amount
payment_date
created_at
updated_at

Insert into Order_items table

```
INSERT INTO `order_items` (`id`, `order_id`, `product_id`, `quantity`, `price_per_piece`) VALUES (NULL, '20', '12', '2', '30'), (NULL, '22', '13', '1', '30'), (NULL, '25', '20', '', '40'), (NULL, '23', '6', '3', '90'), (NULL, '24', '9', '1', '60');
```

id
order_id
product_id
quantity
price_per_piece

Insert Into Order_Addresses Table

```
INSERT INTO `order_addresses` (`id`, `order_id`, `address_line1`, `address_line2`, `postal_code`, `city`, `country`, `created_at`, `updated_at`) VALUES (NULL, '20', '19 colmore row', 'new street', 'b1 2as', 'birmingham', 'united kingdom', current_timestamp(), current_timestamp()), (NULL, '22', '11 new street', 'park lane', 'b2 3aq', 'birmingham', 'united kingdom', current_timestamp(), current_timestamp()), (NULL, '23', '12 avenue close', 'elvetham road', 'b15 2nl', 'birmingham', 'united kingdom', current_timestamp(), current_timestamp()), (NULL, '24', '222 cardigan street', 'west london', 'w2 1aw', 'london', 'united kingdom', current_timestamp(), current_timestamp()), (NULL, '25', '90 east end road', 'manchester drive', 'm23 1ap ', 'manchester', 'united kingdom', current_timestamp(), current_timestamp());
```

id
order_id
address_line1
address_line2
postal_code
city
country
created_at
updated_at

Insert Into user_carts table

```
INSERT INTO `user_carts` (`id`, `user_id`, `created_at`, `updated_at`) VALUES (NULL, '1', current_timestamp(), current_timestamp()), (NULL, '2', current_timestamp(), current_timestamp()), (NULL, '3', current_timestamp(), current_timestamp()), (NULL, '4', current_timestamp(), current_timestamp()), (NULL, '5', current_timestamp(), current_timestamp()), (NULL, '6', current_timestamp(), current_timestamp()), (NULL, '7', current_timestamp(), current_timestamp()), (NULL, '8', current_timestamp(), current_timestamp()), (NULL, '9', current_timestamp(), current_timestamp()), (NULL, '10', current_timestamp(), current_timestamp());
```

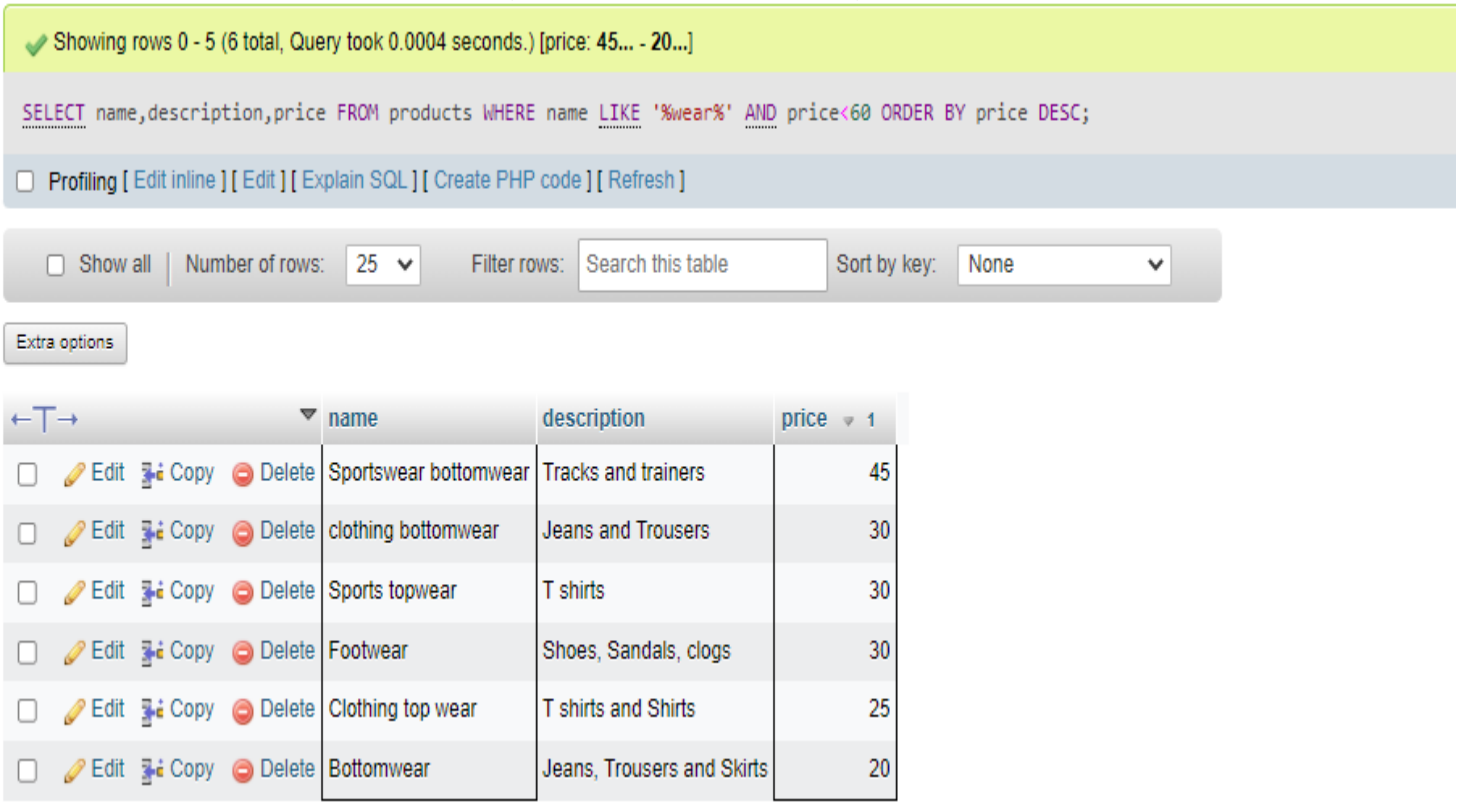
id
user_id
created_at
updated_at

7.Database Queries

7.1 Queries Written by Abdullah

1 - Query that prints name, description and the price of the products which have price less than 60 and have "wear" in their name in price greater to lowest.

SELECT name,description,price FROM products WHERE name LIKE '%wear%' AND price<60 ORDER BY price DESC;



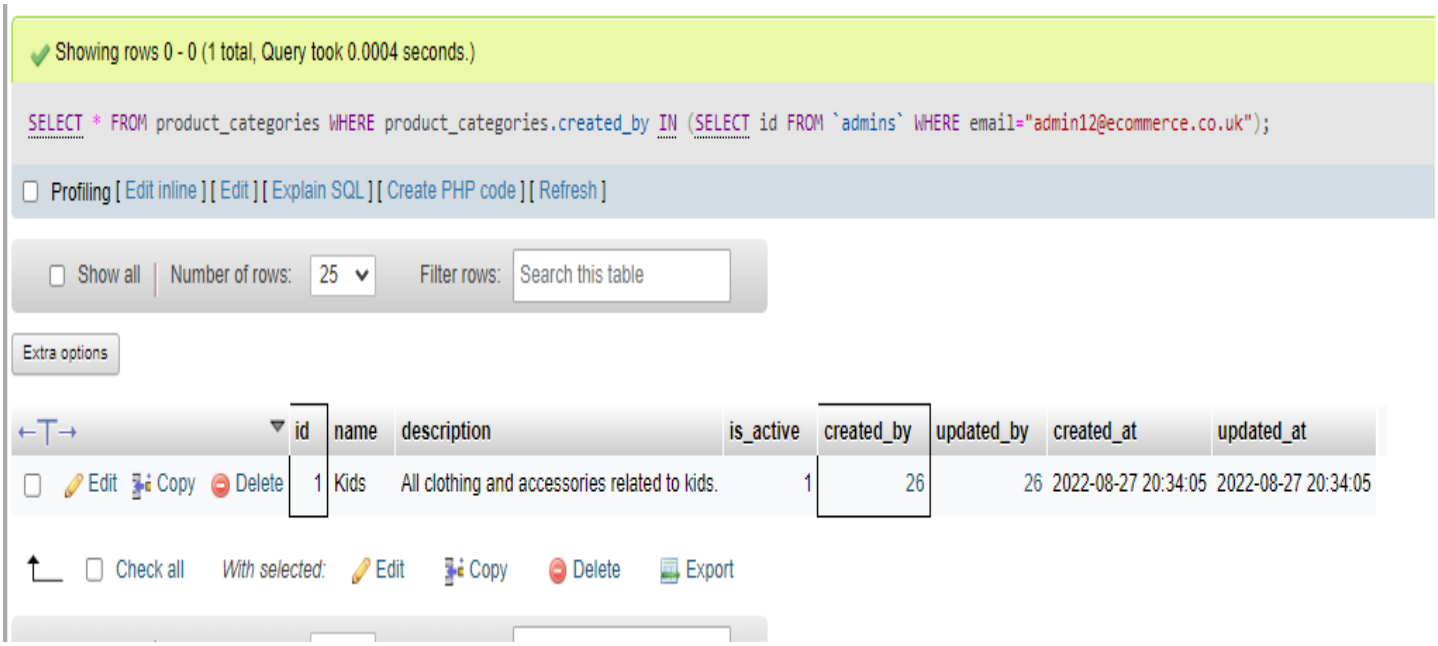
	name	description	price
Edit Copy Delete	Sportswear bottomwear	Tracks and trainers	45
Edit Copy Delete	clothing bottomwear	Jeans and Trousers	30
Edit Copy Delete	Sports topwear	T shirts	30
Edit Copy Delete	Footwear	Shoes, Sandals, clogs	30
Edit Copy Delete	Clothing top wear	T shirts and Shirts	25
Edit Copy Delete	Bottomwear	Jeans, Trousers and Skirts	20

6. Query that shows the orders of the London whose payments are done and have been despatched according to the latest dates.

SELECT * FROM orders JOIN order_addresses ON orders.id=order_addresses.order_id WHERE order_addresses.city="london" AND orders.delivery_status="despatced" AND orders.payment_status="paid" ORDER BY orders.created_at DESC;

5. Query that shows the categories created by admin with email_address="admin12@ecommerce.co.uk";

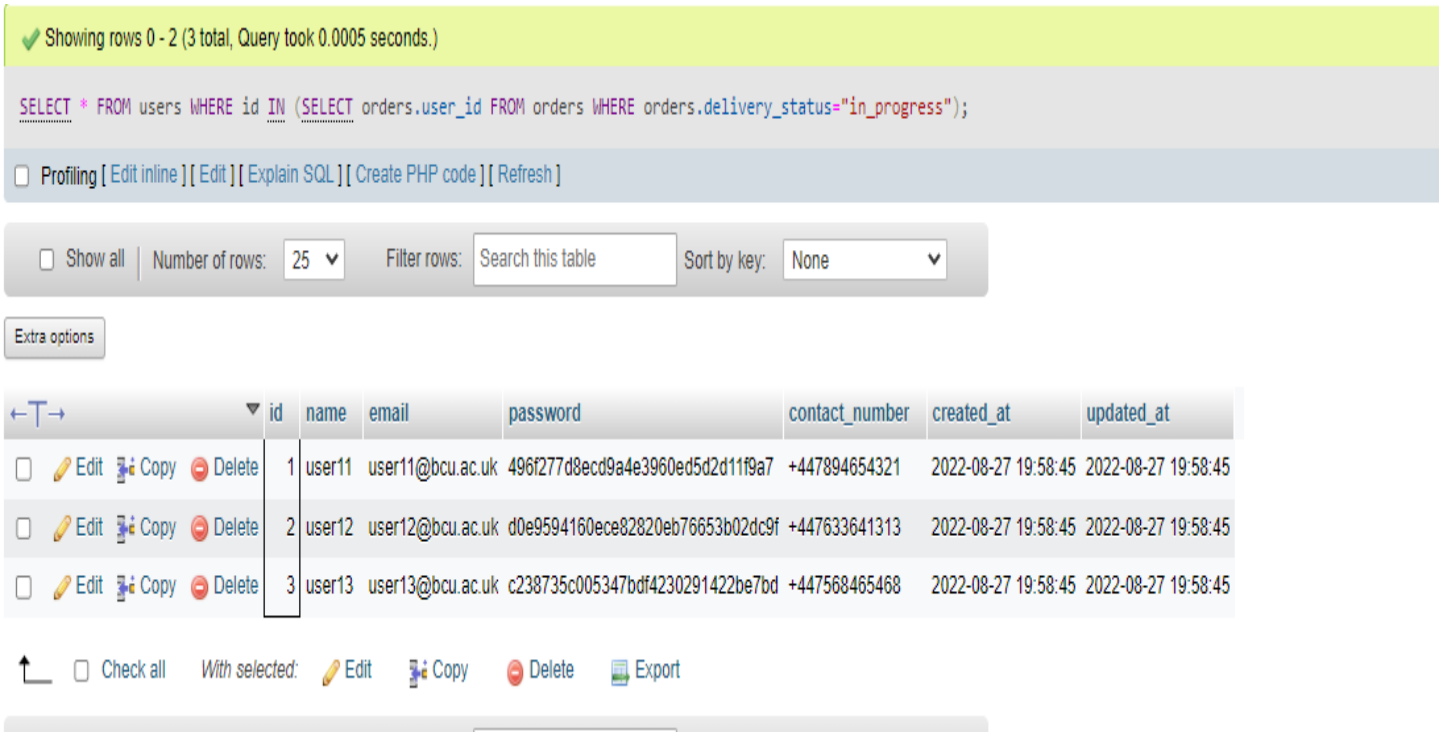
SELECT * FROM product_categories WHERE product_categories.created_by IN (SELECT id FROM `admins` WHERE email="admin12@ecommerce.co.uk");



7.2 Queries Written by Mohd Bilal

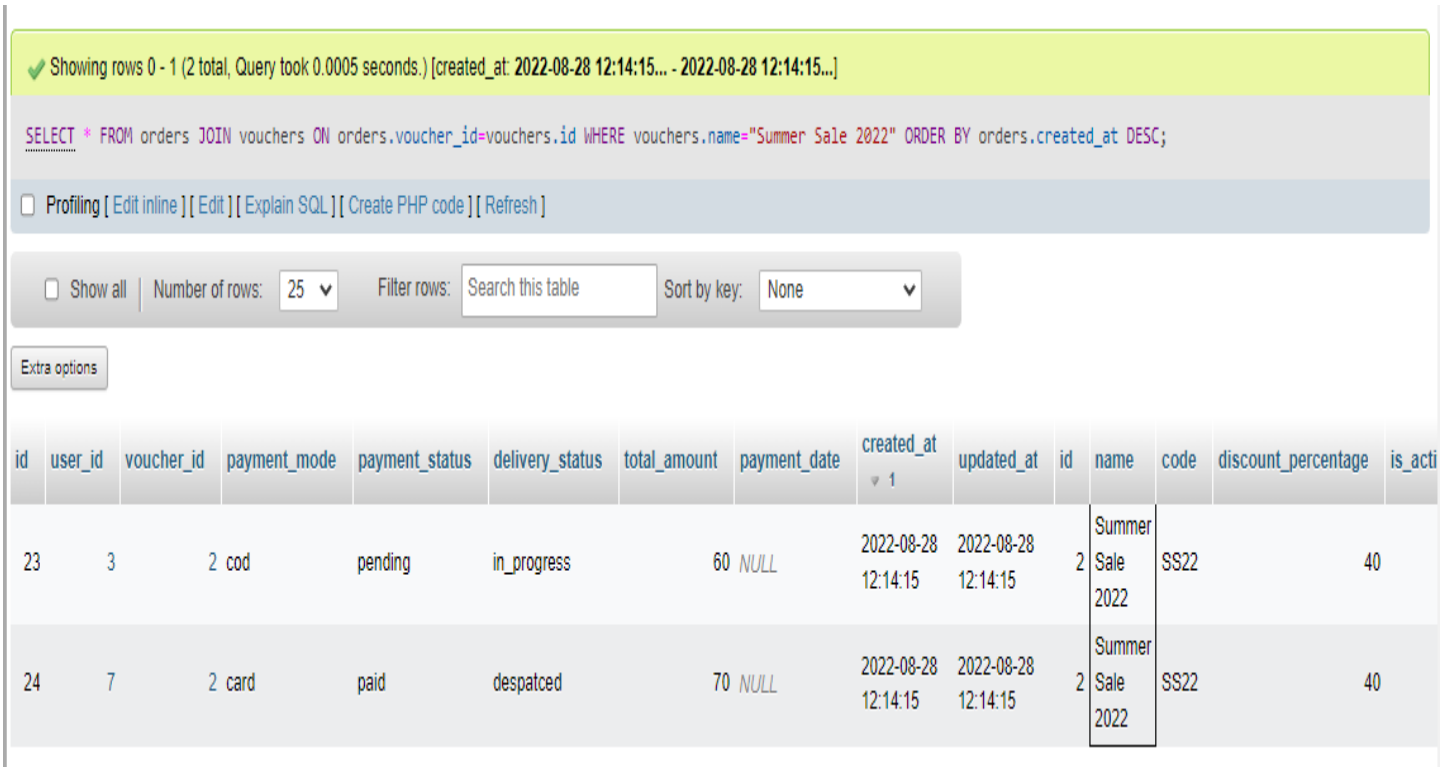
4. The query which shows the user details of the users whose orders delivery status are still in progress.

SELECT * FROM users WHERE id IN (SELECT orders.user_id FROM orders WHERE orders.delivery_status="in_progress");



8. Query that shows all the orders of the voucher "Summer Sale 2022" sorted according to the latest date.

SELECT * FROM orders JOIN vouchers ON orders.voucher_id=vouchers.id WHERE vouchers.name="Summer Sale 2022" ORDER BY orders.created_at DESC;



Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.) [created_at: 2022-08-28 12:14:15... - 2022-08-28 12:14:15...]

```
SELECT * FROM orders JOIN vouchers ON orders.voucher_id=vouchers.id WHERE vouchers.name="Summer Sale 2022" ORDER BY orders.created_at DESC;
```

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

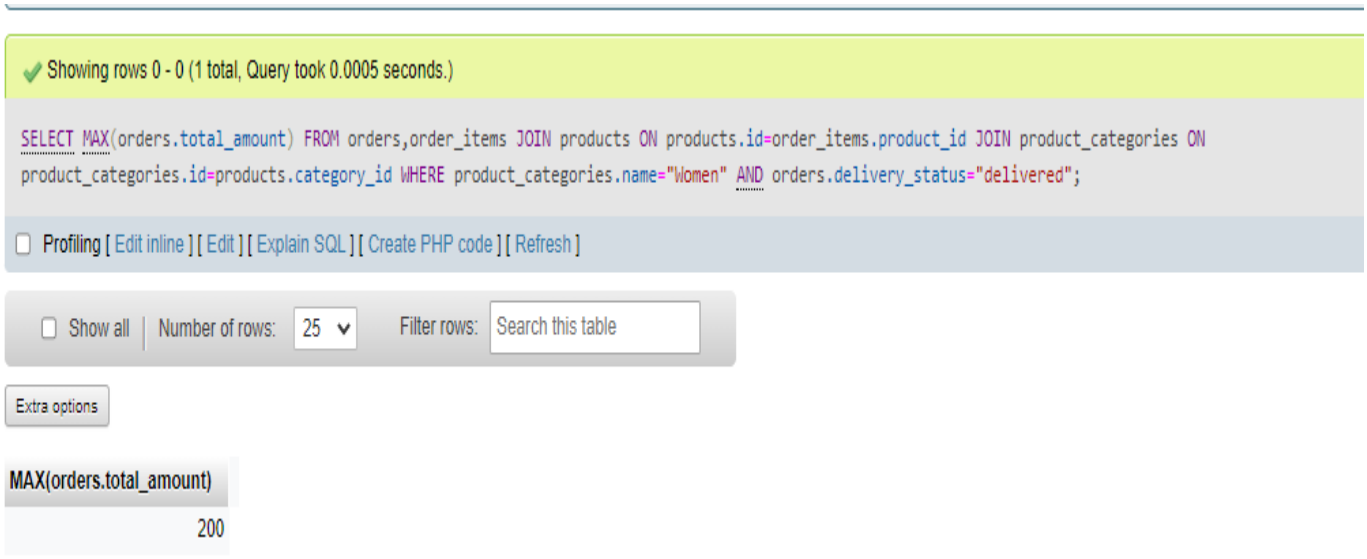
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

id	user_id	voucher_id	payment_mode	payment_status	delivery_status	total_amount	payment_date	created_at ▼ 1	updated_at	id	name	code	discount_percentage	is_acti
23	3	2	cod	pending	in_progress	60	NULL	2022-08-28 12:14:15	2022-08-28 12:14:15	2	Summer Sale 2022	SS22	40	
24	7	2	card	paid	despatced	70	NULL	2022-08-28 12:14:15	2022-08-28 12:14:15	2	Summer Sale 2022	SS22	40	

3.Query which shows maximum amount of the delivered order of the product category in Women.

SELECT MAX(orders.total_amount) FROM orders,order_items JOIN products ON products.id=order_items.product_id JOIN product_categories ON product_categories.id=products.category_id WHERE product_categories.name="Women" AND orders.delivery_status="delivered";



Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT MAX(orders.total_amount) FROM orders,order_items JOIN products ON products.id=order_items.product_id JOIN product_categories ON
product_categories.id=products.category_id WHERE product_categories.name="Women" AND orders.delivery_status="delivered";
```

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

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

Extra options

MAX(orders.total_amount)
200

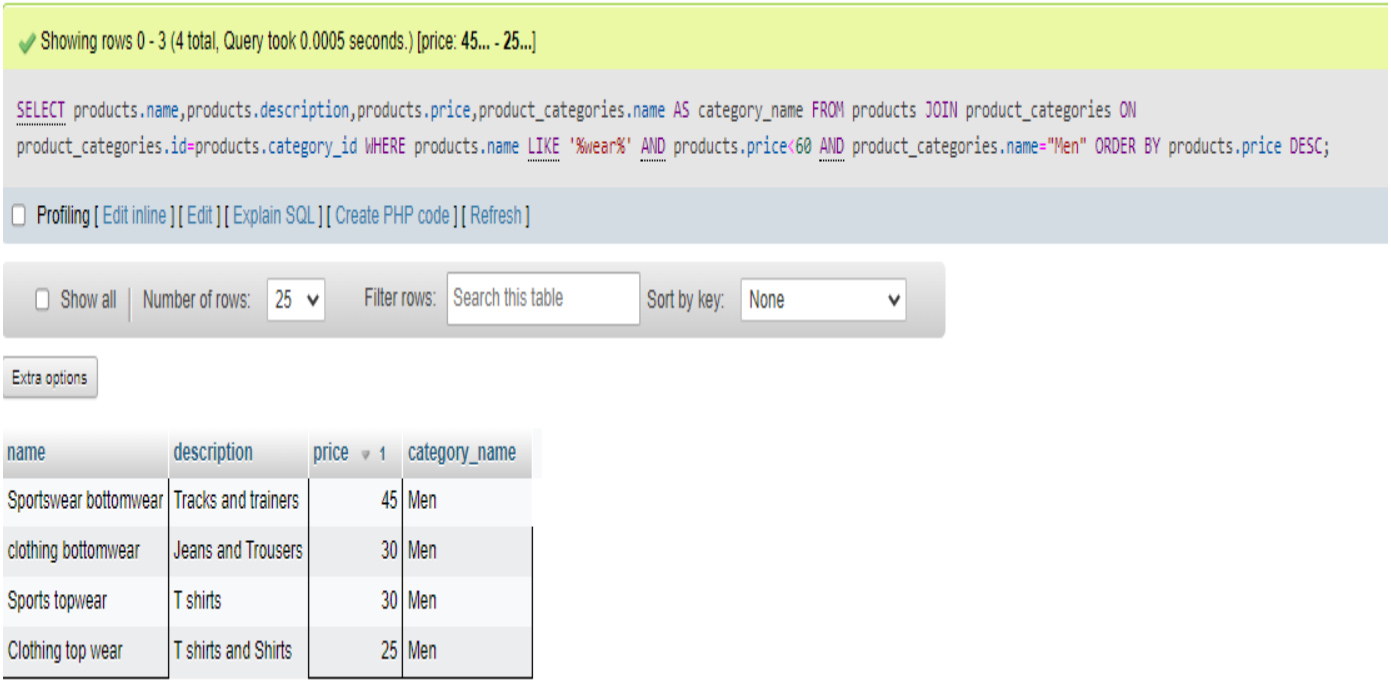
7.3 Queries Written by Subhaan

9- Query that shows the user having the maximum amount of the order

SELECT * FROM users WHERE users.id IN (SELECT orders.user_id FROM orders HAVING MAX(orders.total_amount));

2-Query that prints name, description and the price and category of the products which have price less than 60 have "wear" in their name in price greater to lowest only for products for men.

SELECT products.name,products.description,products.price,product_categories.name AS category_name FROM products JOIN product_categories ON product_categories.id=products.category_id WHERE products.name LIKE '%wear%' AND products.price<60 AND product_categories.name="Men" ORDER BY products.price DESC;



name	description	price	category_name
Sportswear bottomwear	Tracks and trainers	45	Men
clothing bottomwear	Jeans and Trousers	30	Men
Sports topwear	T shirts	30	Men
Clothing top wear	T shirts and Shirts	25	Men

7- Query that shows the name of the product category, product and stock quantity whose stock quantity is minimum in the stock.

SELECT product_categories.name AS product_category_name,products.name AS product_name,MIN(products.stock_quantity) FROM products JOIN product_categories ON products.category_id=product_categories.id WHERE product_categories.name="Men";

product_category_name	product_name	MIN(products.stock_quantity)
Men	Clothing top wear	2500

Conclusion

This developed e commerce database achieves the goal of constructing a system indicated in the domain definition and business condition. The ERD was created with the relationships and their limits in mind. To guarantee that the design is complete, the tables were normalised to the third normal form. Then, SQL was used to construct tables, enter data into them, add foreign keys to link the entities, and test constraints. Finally, basic and sophisticated queries were developed to show the complexity of the database design, which was then adjusted to improve the efficiency of these searches.

References

- *Issues In Information Systems*, 2013. REAL WORLD DESIGN AND IMPLEMENTATION IN THE STUDENTS FIRST DATABASE COURSE.
- Czurylo, T., 2021. Online Retail Me Yes: E-Commerce and Employment during the COVID-19 Pandemic. *SSRN Electronic Journal*,.
- Editor, S., 2016. Secure Data Storage for Electronic Commerce Network Payment. *International Journal of Simulation: Systems, Science & Technology*,.
- Sumalatha, A., Vookanti, R. and Vannala, S., 2021. Study on Applications of SQL and Not only SQL Databases used for Big Data Analytics. *SSRN Electronic Journal*,.
- Truitt, L., 2022. *How to Build a (Successful) Ecommerce Website (2022)*. [online] The BigCommerce Blog. Available at: <https://www.bigcommerce.com/blog/build-an-ecommerce-website/> [Accessed 29 August 2022].
- Silberschatz, A., Korth, H. and Sudarshan, S., n.d. *Database system concepts*.