Online Shopping Center

Database Design Document

By

Zohaib Khalid BSSE-FA22-108

Submitted to

Mr. Umar Latif

Bachelor of Science in Software Engineering

DEPARTMENT OF COMPUTER SCIENCE

The University of Faisalabad

1. Project Description

The project involves creating a database system for an e-commerce platform. This platform handles various functionalities, such as:

- **User Management:** Managing user accounts, profiles, addresses, and payment methods.
- **Product Catalog:** Organizing products into categories, storing product details, and images.
- **Shopping Cart:** Allowing users to add, update, and remove items in their cart.
- **Order Processing:** Recording order details, tracking order status, and managing payments.
- **Reviews and Ratings:** Enabling users to leave reviews and ratings for products.
- **Promotional Activities:** Managing discounts and promotions on products or categories.

The database ensures data integrity, efficient retrieval, and smooth handling of all user interactions and transactions.

2. Entities with Types and Attributes

1. Country

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **country_name**

2. Site User

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **email_address**
  - **phone_number**
  - **password**

3. Address

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **unit_number**
  - **street_number**
  - **address_line1**
  - **address_line2**
  - **city**
  - **region**
  - **postal_code**
  - **country_id** (Foreign Key)

4. User Address

- **Type**: Weak Entity
- **Attributes**:
  - **id** (Primary Key)
  - **user_id** (Foreign Key)
  - **address_id** (Foreign Key)
  - **is_default**

5. Payment Type

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)

    - **value**

6. User Payment Method

- **Type**: Weak Entity
- **Attributes**:
    - **id** (Primary Key)
    - **user_id** (Foreign Key)
    - **payment_type_id** (Foreign Key)
    - **provider**
    - **account_number**
    - **expiry_date**
    - **is_default**

7. Product Category

- **Type**: Strong Entity, Hierarchical
- **Attributes**:
    - **id** (Primary Key)
    - **parent_category_id** (Foreign Key, Nullable)
    - **category_name**

8. Product

- **Type**: Strong Entity
- **Attributes**:
    - **id** (Primary Key)
    - **category_id** (Foreign Key)
    - **name**
    - **description**

9. Product Item

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **product_id** (Foreign Key)
  - **SKU**
  - **qty_in_stock**
  - **product_image**
  - **price**

10. Variation

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **category_id** (Foreign Key)
  - **name**

11. Variation Option

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **variation_id** (Foreign Key)
  - **value**

12. Product Configuration

- **Type**: Weak Entity
- **Attributes**:
    - **id** (Primary Key)
    - **product_item_id** (Foreign Key)
    - **variation_option_id** (Foreign Key)

13. Promotion

- **Type**: Strong Entity
- **Attributes**:
    - **id** (Primary Key)
    - **name**
    - **description**
    - **discount_rate**
    - **start_date**
    - **end_date**

14. Promotion Category

- **Type**: Weak Entity, Associative
- **Attributes**:
    - **category_id** (Foreign Key)
    - **promotion_id** (Foreign Key)

15. Shopping Cart

- **Type**: Strong Entity
- **Attributes**:
    - **id** (Primary Key)

    - **user_id** (Foreign Key)

16. Shopping Cart Item

- **Type**: Weak Entity
- **Attributes**:
    - **id** (Primary Key)
    - **cart_id** (Foreign Key)
    - **product_item_id** (Foreign Key)
    - **qty**

17. Shop Order

- **Type**: Strong Entity
- **Attributes**:
    - **id** (Primary Key)
    - **user_id** (Foreign Key)
    - **order_date**
    - **payment_method_id** (Foreign Key)
    - **shipping_address**
    - **shipping_method**
    - **order_total**
    - **order_status_id** (Foreign Key)

18. Order Line

- **Type**: Weak Entity
- **Attributes**:
    - **id** (Primary Key)
    - **product_item_id** (Foreign Key)

- **order_id** (Foreign Key)
- **qty**
- **price**

19. Order Status

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **status_name**

20. Shipping Method

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **method_name**
  - **cost**

21. Product Review

- **Type**: Strong Entity
- **Attributes**:
  - **id** (Primary Key)
  - **product_id** (Foreign Key)
  - **user_id** (Foreign Key)
  - **rating**
  - **review_text**
  - **review_date**

3. Data type of attributes related to different entities

1. Country

Attributes:

- **id** (INT)
- **country_name** (VARCHAR(100))

2. Site User

Attributes:

- **id** (INT)
- **email_address** (VARCHAR(100))
- **phone_number** (VARCHAR(15))
- **password** (VARCHAR(100))

3. Address

Attributes:

- **id** (INT)
- **unit_number** (VARCHAR(50))
- **street_number** (VARCHAR(50))
- **address_line1** (VARCHAR(100))
- **address_line2** (VARCHAR(100))
- **city** (VARCHAR(50))
- **region** (VARCHAR(50))
- **postal_code** (VARCHAR(10))
- **country_id** (INT)

4. User Address

Attributes:

- **id** (INT)
- **user_id** (INT)
- **address_id** (INT)
- **is_default** (BOOLEAN)

5. Payment Type

Attributes:

- **id** (INT)
- **value** (VARCHAR(50))

6. User Payment Method

Attributes:

- **id** (INT)
- **user_id** (INT)
- **payment_type_id** (INT)
- **provider** (VARCHAR(50))
- **account_number** (VARCHAR(50))
- **expiry_date** (VARCHAR(10))
- **is_default** (BOOLEAN)

7. Product Category

Attributes:

- **id** (INT)

- **parent_category_id** (INT, Nullable)
- **category_name** (VARCHAR(50))

8. Product

Attributes:

- **id** (INT)
- **category_id** (INT)
- **name** (VARCHAR(100))
- **description** (TEXT)

9. Product Item

Attributes:

- **id** (INT)
- **product_id** (INT)
- **SKU** (VARCHAR(50))
- **qty_in_stock** (INT)
- **product_image** (VARCHAR(100))
- **price** (DECIMAL(10, 2))

10. Variation

Attributes:

- **id** (INT)
- **category_id** (INT)
- **name** (VARCHAR(50))

11. Variation Option

Attributes:

- **id** (INT)
- **variation_id** (INT)
- **value** (VARCHAR(50))

12. Product Configuration

Attributes:

- **id** (INT)
- **product_item_id** (INT)
- **variation_option_id** (INT)

13. Promotion

Attributes:

- **id** (INT)
- **name** (VARCHAR(100))
- **description** (TEXT)
- **discount_rate** (DECIMAL(5, 2))
- **start_date** (DATE)
- **end_date** (DATE)

14. Promotion Category

Attributes:

- **category_id** (INT)
- **promotion_id** (INT)

15. Shopping Cart

Attributes:

- **id** (INT)
- **user_id** (INT)

16. Shopping Cart Item

Attributes:

- **id** (INT)
- **cart_id** (INT)
- **product_item_id** (INT)
- **qty** (INT)

17. Shop Order

Attributes:

- **id** (INT)
- **user_id** (INT)
- **order_date** (DATE)
- **payment_method_id** (INT)
- **shipping_address** (VARCHAR(255))
- **shipping_method** (VARCHAR(50))
- **order_total** (DECIMAL(10, 2))
- **order_status_id** (INT)

18. Order Line

Attributes:

- **id** (INT)

- **product_item_id** (INT)
- **order_id** (INT)
- **qty** (INT)
- **price** (DECIMAL(10, 2))

19. Order Status

Attributes:

- **id** (INT)
- **status_name** (VARCHAR(50))

20. Shipping Method

Attributes:

- **id** (INT)
- **method_name** (VARCHAR(50))
- **cost** (DECIMAL(10, 2))

21. Product Review

Attributes:

- **id** (INT)
- **product_id** (INT)
- **user_id** (INT)
- **rating** (INT)
- **review_text** (TEXT)
- **review_date** (DATE)

4. Relationships Between Entities Along with Cardinalities

1. **Country - Address**: One-to-Many
    - One country can have multiple addresses.
    - **country_id** in **address** references **id** in **country**.

2. **Site User - User Address**: One-to-Many
    - One user can have multiple addresses.
    - **user_id** in **user_address** references **id** in **site_user**.

3. **Address - User Address**: One-to-Many
    - One address can belong to multiple users.
    - **address_id** in **user_address** references **id** in **address**.

4. **Site User - User Payment Method**: One-to-Many
    - One user can have multiple payment methods.
    - **user_id** in **user_payment_method** references **id** in **site_user**.

5. **Payment Type - User Payment Method**: One-to-Many
    - One payment type can be used by multiple user payment methods.
    - **payment_type_id** in **user_payment_method** references **id** in **payment_type**.

6. **Product Category - Product**: One-to-Many
    - One product category can have multiple products.
    - **category_id** in **product** references **id** in **product_category**.

7. **Product - Product Item**: One-to-Many
    - One product can have multiple product items.
    - **product_id** in **product_item** references **id** in **product**.

8. **Product Category - Variation**: One-to-Many
    - One product category can have multiple variations.
    - **category_id** in **variation** references **id** in **product_category**.

9. **Variation - Variation Option**: One-to-Many
    - One variation can have multiple variation options.
    - **variation_id** in **variation_option** references **id** in **variation**.

10. **Product Item - Product Configuration**: One-to-Many
    - One product item can have multiple configurations.
    - **product_item_id** in **product_configuration** references **id** in **product_item**.

11. **Variation Option - Product Configuration**: One-to-Many
    - One variation option can be used in multiple product configurations.
    - **variation_option_id** in **product_configuration** references **id** in **variation_option**.

12. **Product Category - Promotion Category**: Many-to-Many
    - One product category can have multiple promotions and vice versa.
    - **category_id** in **promotion_category** references **id** in **product_category**.
    - **promotion_id** in **promotion_category** references **id** in **promotion**.

13. **Site User - Shopping Cart**: One-to-One
    - One user can have one shopping cart.
    - **user_id** in **shopping_cart** references **id** in **site_user**.

14. **Shopping Cart - Shopping Cart Item**: One-to-Many
    - One shopping cart can have multiple items.
    - **cart_id** in **shopping_cart_item** references **id** in **shopping_cart**.

15. **Product Item - Shopping Cart Item**: One-to-Many
    - One product item can be in multiple shopping cart items.
    - **product_item_id** in **shopping_cart_item** references **id** in **product_item**.

16. **Site User - Shop Order**: One-to-Many
    - One user can have multiple orders.
    - **user_id** in **shop_order** references **id** in **site_user**.

17. **User Payment Method - Shop Order**: One-to-Many
    - One payment method can be used in multiple orders.
    - **payment_method_id** in **shop_order** references **id** in **user_payment_method**.

18. **Order Status - Shop Order**: One-to-Many
    - One order status can apply to multiple orders.
    - **order_status_id** in **shop_order** references **id** in **order_status**.

19. **Shop Order - Order Line**: One-to-Many
    - One shop order can have multiple order lines.
    - **order_id** in **order_line** references **id** in **shop_order**.

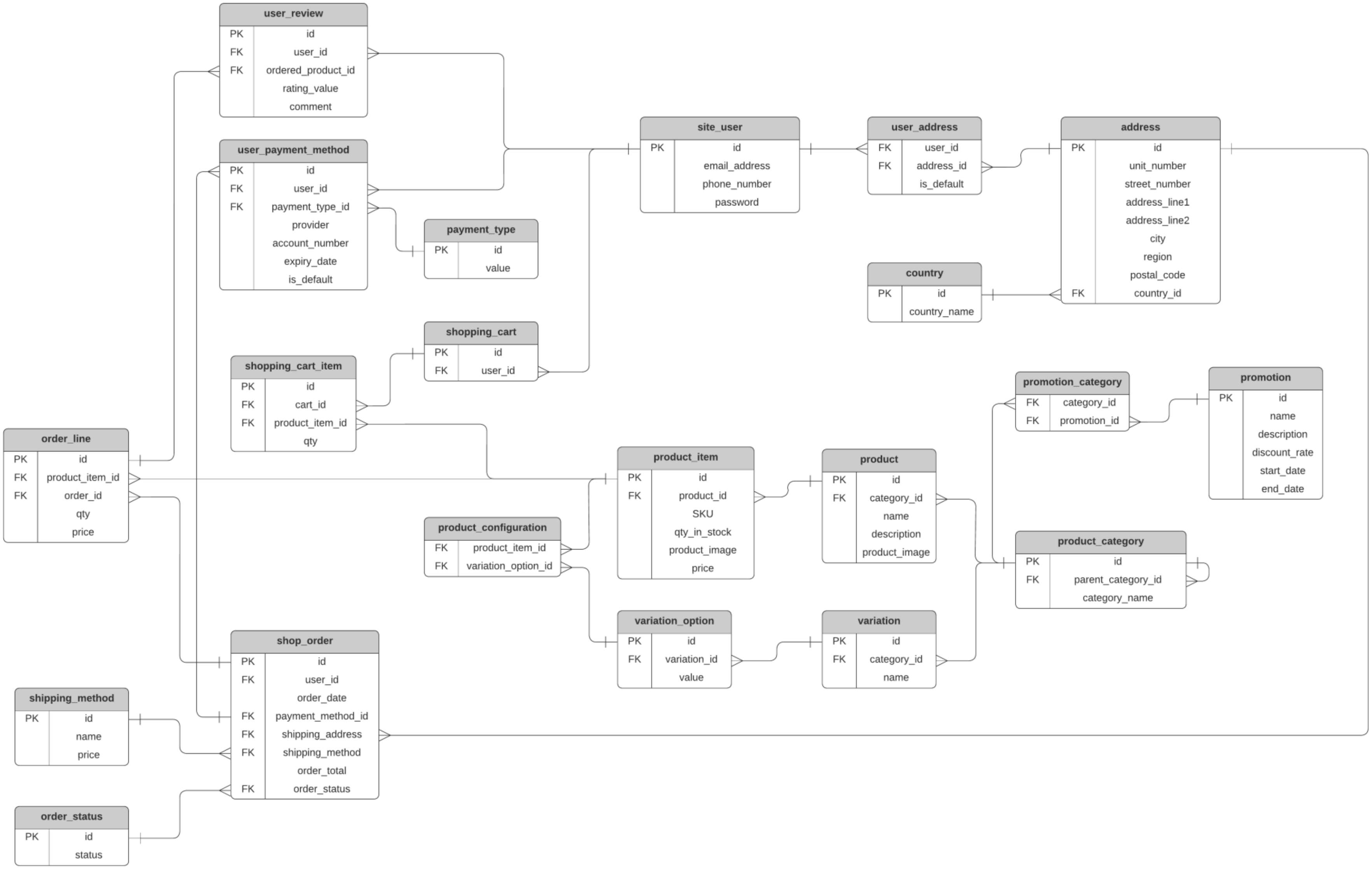
20. **Product Item - Order Line**: One-to-Many
    - One product item can be in multiple order lines.
    - **product_item_id** in **order_line** references **id** in **product_item**.

21. **Product - Product Review**: One-to-Many
    - One product can have multiple reviews.
    - **product_id** in **product_review** references **id** in **product**.

22. **Site User - Product Review**: One-to-Many
    - One user can write multiple reviews.
    - **user_id** in **product_review** references **id** in **site_user**.

5. ER Diagram of the system

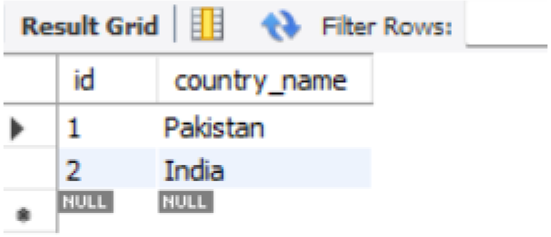
(The diagram is on the next page cause the diagram is too big, Heading and the ER diagram can't fit within the same page)


user_review
PK id
FK user_id
FK ordered_product_id
rating_value
comment
user_payment_method
PK id
FK user_id
FK payment_type_id
provider
account_number
expiry_date
is_default
payment_type
PK id
value
shopping_cart
PK id
FK user_id
shopping_cart_item
PK id
FK cart_id
FK product_item_id
qty
site_user
PK id
email_address
phone_number
password
user_address
FK user_id
FK address_id
is_default
address
PK id
unit_number
street_number
address_line1
address_line2
city
region
postal_code
FK country_id
country
PK id
country_name
promotion_category
FK category_id
FK promotion_id
promotion
PK id
name
description
discount_rate
start_date
end_date
order_line
PK id
FK product_item_id
FK order_id
qty
price
product_item
PK id
FK product_id
SKU
qty_in_stock
product_image
price
product
PK id
FK category_id
name
description
product_image
product_configuration
FK product_item_id
FK variation_option_id
product_category
PK id
FK parent_category_id
category_name
variation_option
PK id
FK variation_id
value
variation
PK id
FK category_id
name
shop_order
PK id
FK user_id
order_date
FK payment_method_id
FK shipping_address
FK shipping_method
order_total
FK order_status
shipping_method
PK id
name
price
order_status
PK id
status

6. Identify Tables

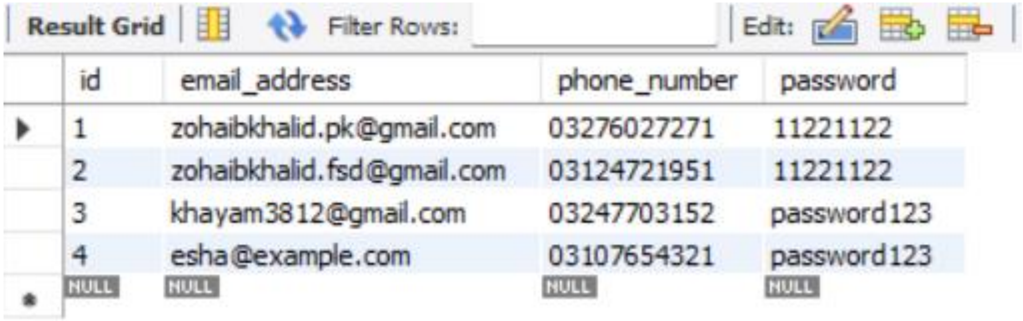
All the tables which will be in the Database are given below

1. Country



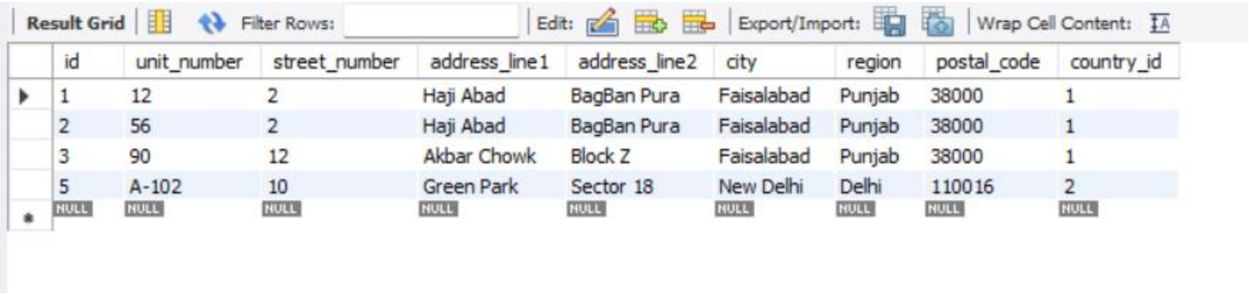
id	country_name
1	Pakistan
2	India
NULL	NULL

2. Site User



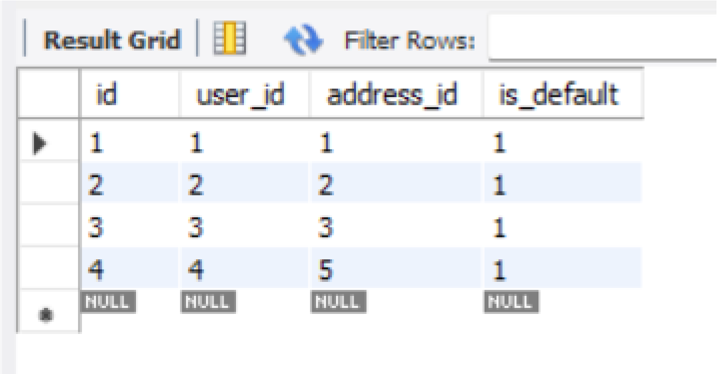
id	email_address	phone_number	password
1	zohaibkhalid.pk@gmail.com	03276027271	11221122
2	zohaibkhalid.fsd@gmail.com	03124721951	11221122
3	khayam3812@gmail.com	03247703152	password123
4	esha@example.com	03107654321	password123
NULL	NULL	NULL	NULL

3. Address



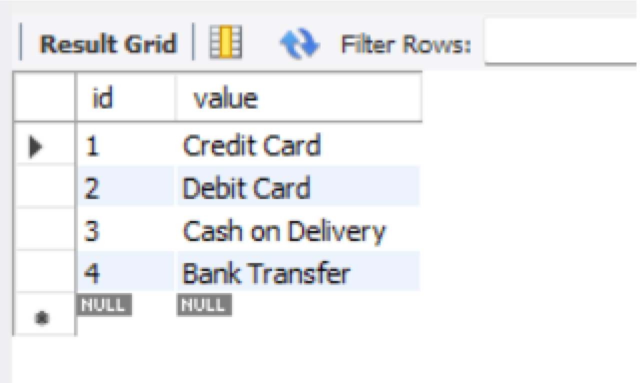
id	unit_number	street_number	address_line1	address_line2	city	region	postal_code	country_id
1	12	2	Haji Abad	BagBan Pura	Faisalabad	Punjab	38000	1
2	56	2	Haji Abad	BagBan Pura	Faisalabad	Punjab	38000	1
3	90	12	Akbar Chowk	Block Z	Faisalabad	Punjab	38000	1
5	A-102	10	Green Park	Sector 18	New Delhi	Delhi	110016	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. User Address



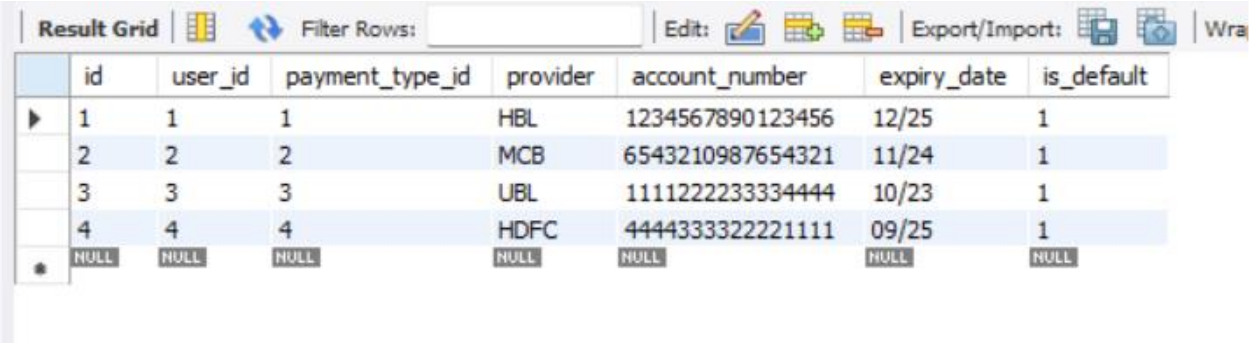
id	user_id	address_id	is_default
1	1	1	1
2	2	2	1
3	3	3	1
4	4	5	1
NULL	NULL	NULL	NULL

5. Payment Type



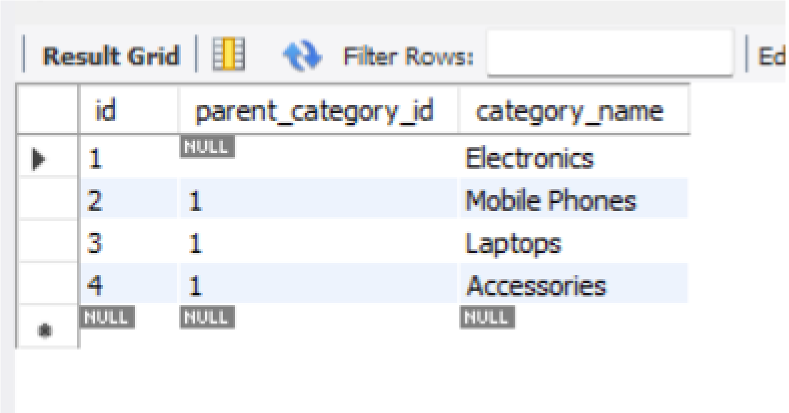
id	value
1	Credit Card
2	Debit Card
3	Cash on Delivery
4	Bank Transfer
NULL	NULL

6. User Payment Method



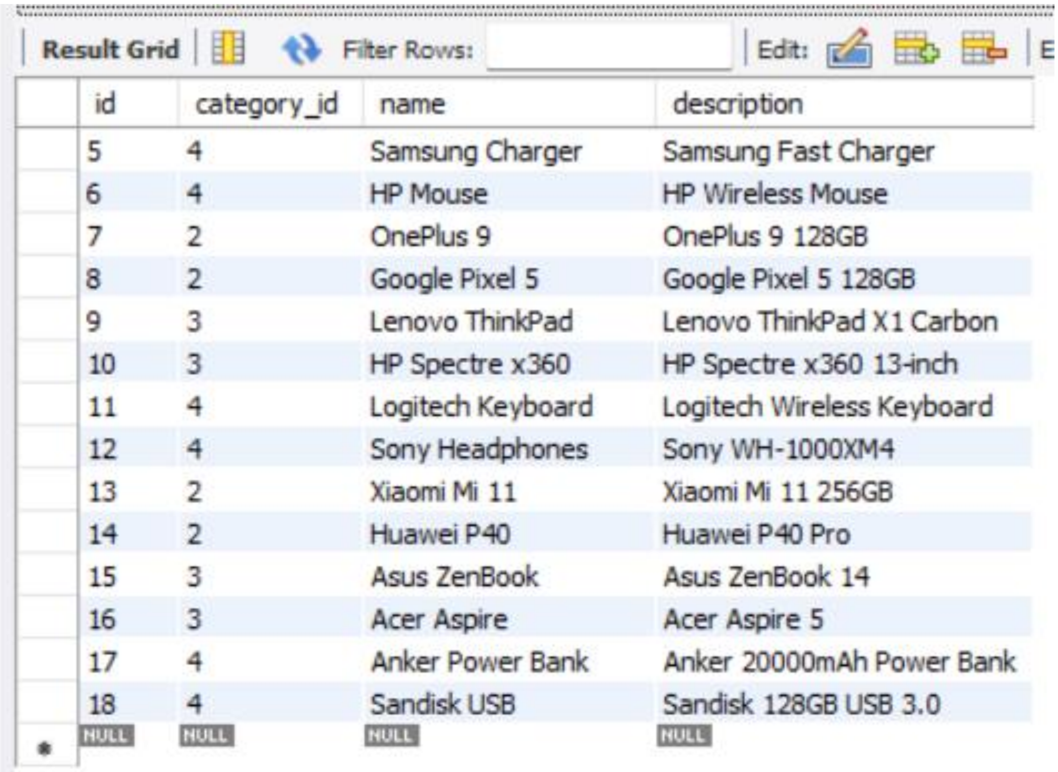
id	user_id	payment_type_id	provider	account_number	expiry_date	is_default
1	1	1	HBL	1234567890123456	12/25	1
2	2	2	MCB	6543210987654321	11/24	1
3	3	3	UBL	1111222233334444	10/23	1
4	4	4	HDFC	4444333322221111	09/25	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

7. **Product Category**



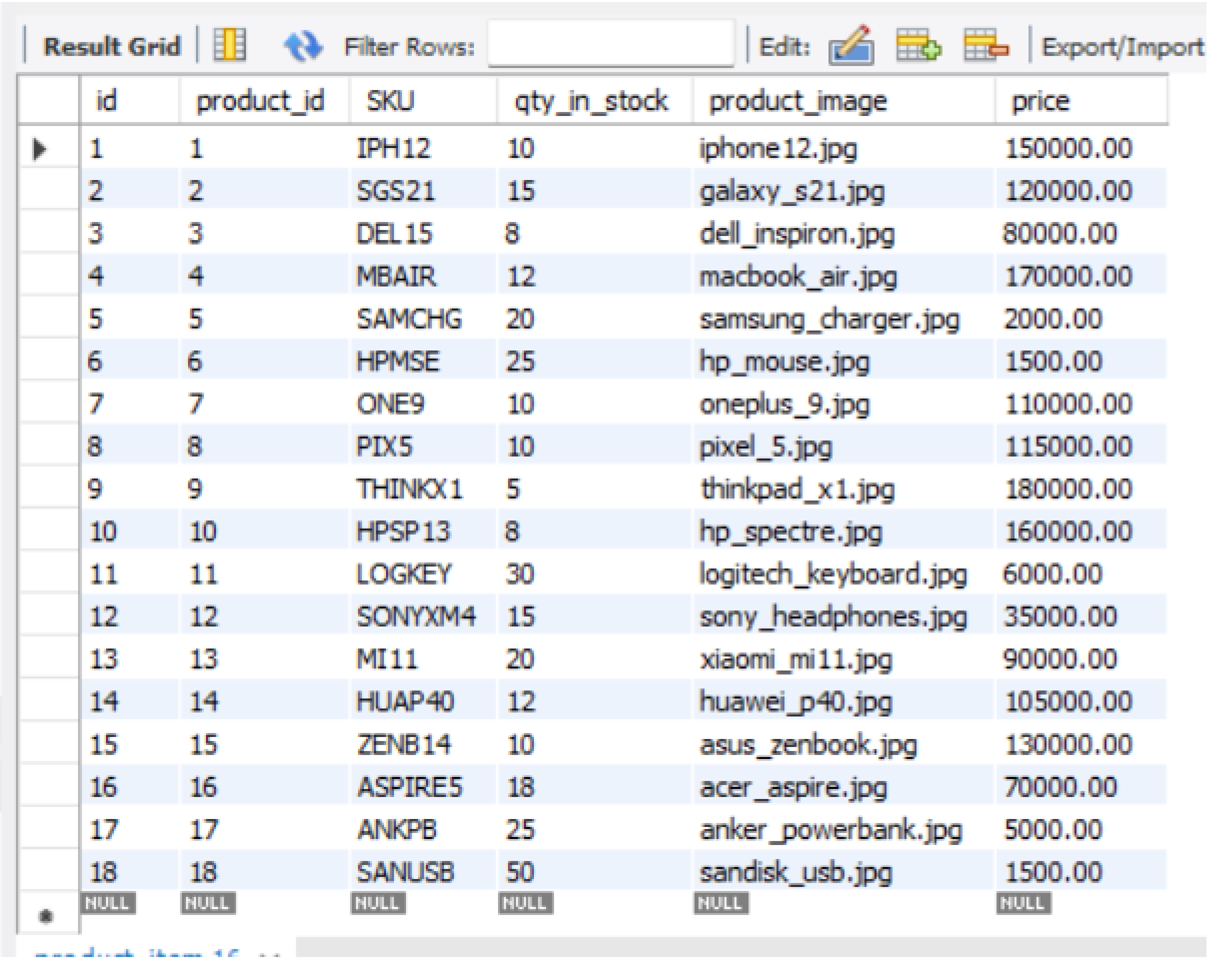
	id	parent_category_id	category_name
▶	1	NULL	Electronics
	2	1	Mobile Phones
	3	1	Laptops
	4	1	Accessories
*	NULL	NULL	NULL

8. **Product**



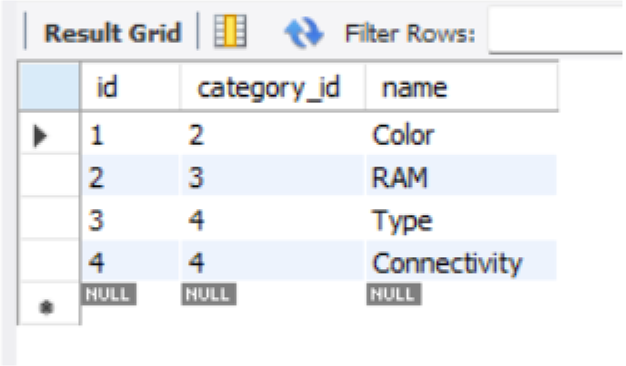
	id	category_id	name	description
	5	4	Samsung Charger	Samsung Fast Charger
	6	4	HP Mouse	HP Wireless Mouse
	7	2	OnePlus 9	OnePlus 9 128GB
	8	2	Google Pixel 5	Google Pixel 5 128GB
	9	3	Lenovo ThinkPad	Lenovo ThinkPad X1 Carbon
	10	3	HP Spectre x360	HP Spectre x360 13-inch
	11	4	Logitech Keyboard	Logitech Wireless Keyboard
	12	4	Sony Headphones	Sony WH-1000XM4
	13	2	Xiaomi Mi 11	Xiaomi Mi 11 256GB
	14	2	Huawei P40	Huawei P40 Pro
	15	3	Asus ZenBook	Asus ZenBook 14
	16	3	Acer Aspire	Acer Aspire 5
	17	4	Anker Power Bank	Anker 20000mAh Power Bank
	18	4	Sandisk USB	Sandisk 128GB USB 3.0
*	NULL	NULL	NULL	NULL

9. Product Item



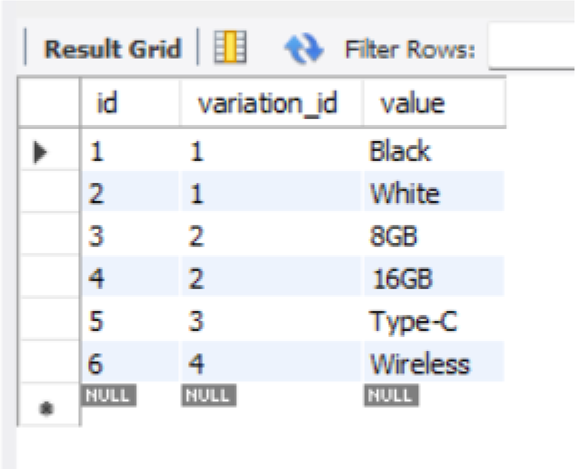
id	product_id	SKU	qty_in_stock	product_image	price
1	1	IPH12	10	iphone12.jpg	150000.00
2	2	SGS21	15	galaxy_s21.jpg	120000.00
3	3	DEL15	8	dell_inspiron.jpg	80000.00
4	4	MBAIR	12	macbook_air.jpg	170000.00
5	5	SAMCHG	20	samsung_charger.jpg	2000.00
6	6	HPMSE	25	hp_mouse.jpg	1500.00
7	7	ONE9	10	oneplus_9.jpg	110000.00
8	8	PIX5	10	pixel_5.jpg	115000.00
9	9	THINKX1	5	thinkpad_x1.jpg	180000.00
10	10	HPSP13	8	hp_spectre.jpg	160000.00
11	11	LOGKEY	30	logitech_keyboard.jpg	6000.00
12	12	SONYXM4	15	sony_headphones.jpg	35000.00
13	13	MI11	20	xiaomi_mi11.jpg	90000.00
14	14	HUAP40	12	huawei_p40.jpg	105000.00
15	15	ZENB14	10	asus_zenbook.jpg	130000.00
16	16	ASPIRE5	18	acer_aspire.jpg	70000.00
17	17	ANKPB	25	anker_powerbank.jpg	5000.00
18	18	SANUSB	50	sandisk_usb.jpg	1500.00
NULL	NULL	NULL	NULL	NULL	NULL

10. Variation



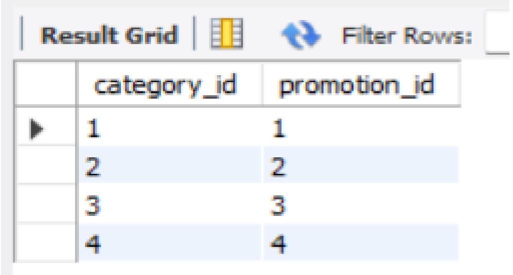
id	category_id	name
1	2	Color
2	3	RAM
3	4	Type
4	4	Connectivity
NULL	NULL	NULL

11. Variation Option



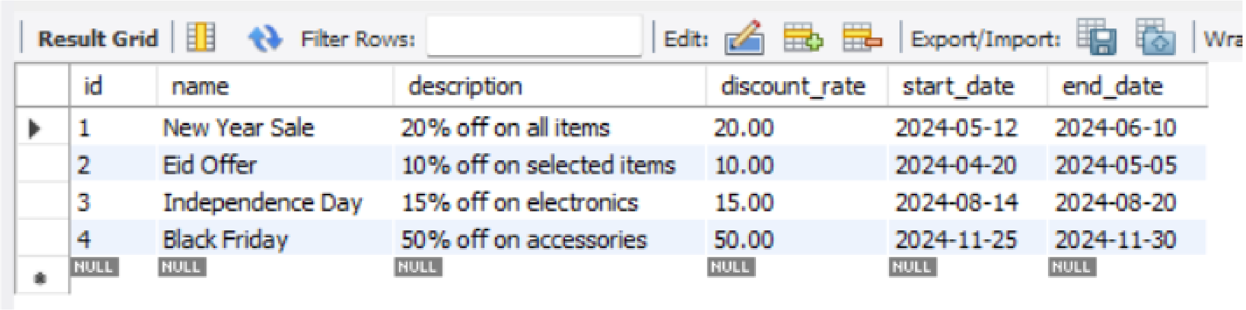
id	variation_id	value
1	1	Black
2	1	White
3	2	8GB
4	2	16GB
5	3	Type-C
6	4	Wireless
NULL	NULL	NULL

12. Product Configuration



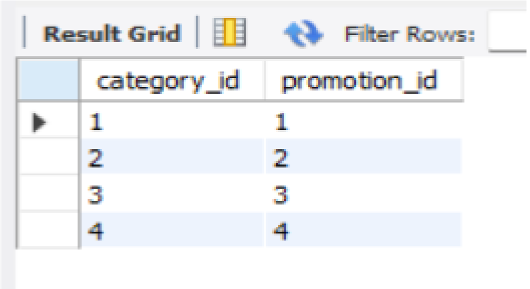
category_id	promotion_id
1	1
2	2
3	3
4	4

13. Promotion



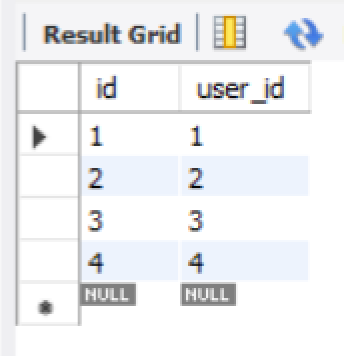
id	name	description	discount_rate	start_date	end_date
1	New Year Sale	20% off on all items	20.00	2024-05-12	2024-06-10
2	Eid Offer	10% off on selected items	10.00	2024-04-20	2024-05-05
3	Independence Day	15% off on electronics	15.00	2024-08-14	2024-08-20
4	Black Friday	50% off on accessories	50.00	2024-11-25	2024-11-30
NULL	NULL	NULL	NULL	NULL	NULL

14. Promotion Category



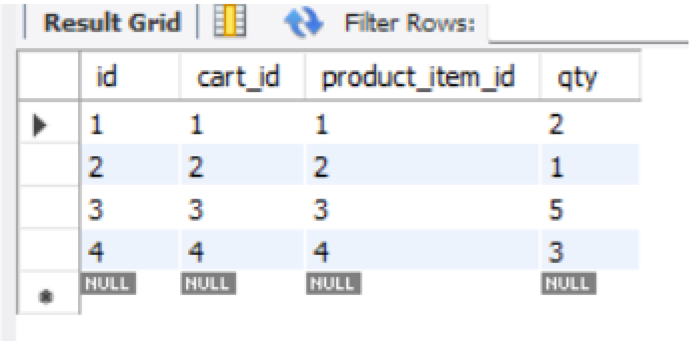
category_id	promotion_id
1	1
2	2
3	3
4	4

15. Shopping Cart



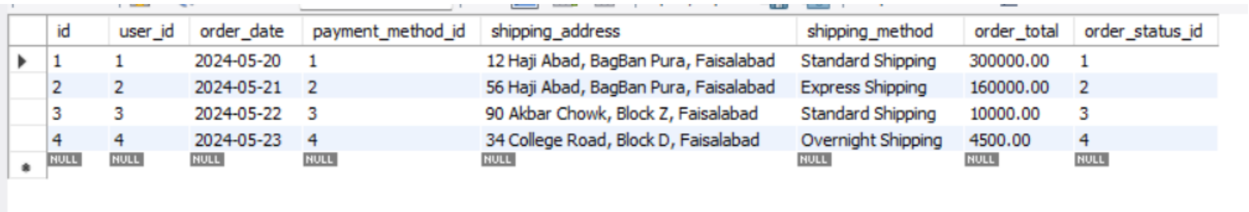
id	user_id
1	1
2	2
3	3
4	4
NULL	NULL

16. Shopping Cart Item



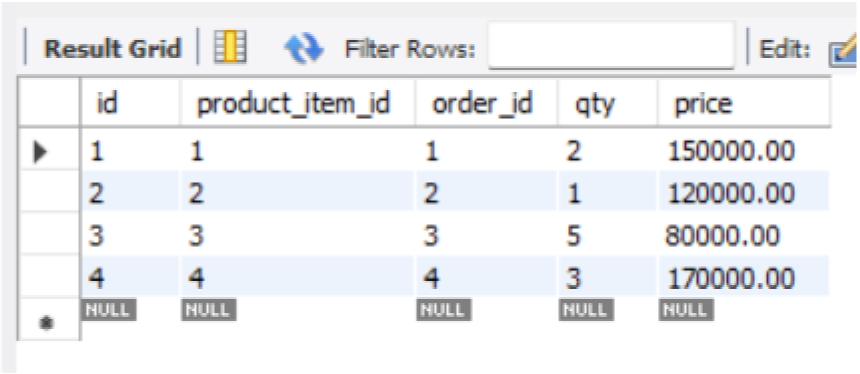
id	cart_id	product_item_id	qty
1	1	1	2
2	2	2	1
3	3	3	5
4	4	4	3
NULL	NULL	NULL	NULL

17. Shop Order



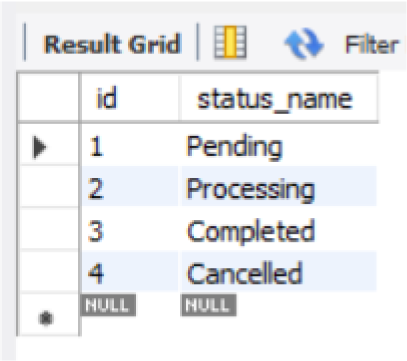
id	user_id	order_date	payment_method_id	shipping_address	shipping_method	order_total	order_status_id
1	1	2024-05-20	1	12 Haji Abad, BagBan Pura, Faisalabad	Standard Shipping	300000.00	1
2	2	2024-05-21	2	56 Haji Abad, BagBan Pura, Faisalabad	Express Shipping	160000.00	2
3	3	2024-05-22	3	90 Akbar Chowk, Block Z, Faisalabad	Standard Shipping	10000.00	3
4	4	2024-05-23	4	34 College Road, Block D, Faisalabad	Overnight Shipping	4500.00	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

18. **Order Line**



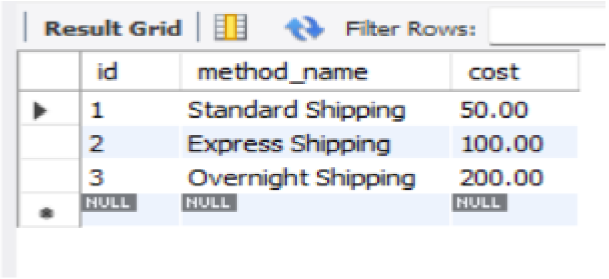
id	product_item_id	order_id	qty	price
1	1	1	2	150000.00
2	2	2	1	120000.00
3	3	3	5	80000.00
4	4	4	3	170000.00
NULL	NULL	NULL	NULL	NULL

19. **Order Status**



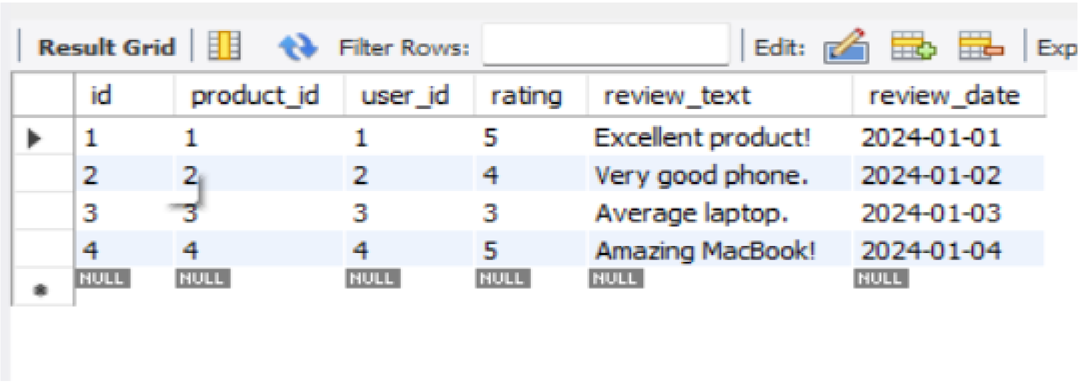
id	status_name
1	Pending
2	Processing
3	Completed
4	Cancelled
NULL	NULL

20. **Shipping Method**



id	method_name	cost
1	Standard Shipping	50.00
2	Express Shipping	100.00
3	Overnight Shipping	200.00
NULL	NULL	NULL

21. **Product Review**



id	product_id	user_id	rating	review_text	review_date
1	1	1	5	Excellent product!	2024-01-01
2	2	2	4	Very good phone.	2024-01-02
3	3	3	3	Average laptop.	2024-01-03
4	4	4	5	Amazing MacBook!	2024-01-04
NULL	NULL	NULL	NULL	NULL	NULL

7. Normalize Tables:

All tables are normalized up to the third normal form (3NF) to ensure there is no data redundancy and integrity is maintained.

8. Queries to create Tables in Database:

1. Create Country Table

```
CREATE TABLE country (
    id INT PRIMARY KEY,
    country_name VARCHAR(100) NOT NULL
);
```

2. Create site_user Table

```
CREATE TABLE site_user (
    id INT PRIMARY KEY,
    email_address VARCHAR(100) NOT NULL,
    phone_number VARCHAR(15) NOT NULL,
    password VARCHAR(100) NOT NULL
);
```

3. Create address Table

```
CREATE TABLE address (
    id INT PRIMARY KEY,
    unit_number VARCHAR(50),
    street_number VARCHAR(50),
    address_line1 VARCHAR(100),
    address_line2 VARCHAR(100),
    city VARCHAR(50),
    region VARCHAR(50),
    postal_code VARCHAR(10),
    country_id INT,
    FOREIGN KEY (country_id) REFERENCES country(id)
);
```

4. Create user_address table

```
CREATE TABLE user_address (
    id INT PRIMARY KEY,
    user_id INT,
    address_id INT,
    is_default BOOLEAN,
    FOREIGN KEY (user_id) REFERENCES site_user(id),
    FOREIGN KEY (address_id) REFERENCES address(id)
);
```

5. Create payment_type table

```
CREATE TABLE payment_type (
    id INT PRIMARY KEY,
    value VARCHAR(50) NOT NULL
);
```

6. Create user_payment_method table

```
CREATE TABLE user_payment_method (
    id INT PRIMARY KEY,
    user_id INT,
    payment_type_id INT,
    provider VARCHAR(50),
    account_number VARCHAR(50),
    expiry_date VARCHAR(10),
    is_default BOOLEAN,
    FOREIGN KEY (user_id) REFERENCES site_user(id),
    FOREIGN KEY (payment_type_id) REFERENCES payment_type(id)
);
```

7. Create product_category table

```
CREATE TABLE product_category (
    id INT PRIMARY KEY,
    parent_category_id INT,
    category_name VARCHAR(50) NOT NULL,
    FOREIGN KEY (parent_category_id) REFERENCES product_category(id)
);
```

8. Create product table

```
CREATE TABLE product (
    id INT PRIMARY KEY,
    category_id INT,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    FOREIGN KEY (category_id) REFERENCES product_category(id)
);
```

9. Create product_item table

```

CREATE TABLE product_item (
    id INT PRIMARY KEY,
    product_id INT,
    SKU VARCHAR(50),
    qty_in_stock INT,
    product_image VARCHAR(100),
    price DECIMAL(10, 2),
    FOREIGN KEY (product_id) REFERENCES product(id)
);
```

10.Create variation table

```
CREATE TABLE variation (
    id INT PRIMARY KEY,
    category_id INT,
    name VARCHAR(50) NOT NULL,
    FOREIGN KEY (category_id) REFERENCES product_category(id)
);
```

11.Create variation_option table

```
CREATE TABLE variation_option (
    id INT PRIMARY KEY,
    variation_id INT,
    value VARCHAR(50) NOT NULL,
    FOREIGN KEY (variation_id) REFERENCES variation(id)
);
```

12.Create product_configuration table

```
CREATE TABLE product_configuration (
    id INT PRIMARY KEY,
    product_item_id INT,
    variation_option_id INT,
    FOREIGN KEY (product_item_id) REFERENCES product_item(id),
    FOREIGN KEY (variation_option_id) REFERENCES variation_option(id)
);
```

13.Create promotion table

```
CREATE TABLE promotion (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    description TEXT,
    discount_rate DECIMAL(5, 2),
    start_date DATE,
    end_date DATE
);
```

14.Create promotion_category table

```
CREATE TABLE promotion_category (
    category_id INT,
    promotion_id INT,
    FOREIGN KEY (category_id) REFERENCES product_category(id),
    FOREIGN KEY (promotion_id) REFERENCES promotion(id)
);
```

15.Create shopping_cart table

```
CREATE TABLE shopping_cart (
    id INT PRIMARY KEY,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES site_user(id)
);
```

16.Create shopping_cart_item table

```
CREATE TABLE shopping_cart_item (
    id INT PRIMARY KEY,
    cart_id INT,
    product_item_id INT,
    qty INT,
    FOREIGN KEY (cart_id) REFERENCES shopping_cart(id),
    FOREIGN KEY (product_item_id) REFERENCES product_item(id)
);
```

17.Create shop_order table

```
CREATE TABLE shop_order (
    id INT PRIMARY KEY,
    user_id INT,
    order_date DATE,
    payment_method_id INT,
    shipping_address VARCHAR(255),
    shipping_method VARCHAR(50),
    order_total DECIMAL(10, 2),
    order_status_id INT
    FOREIGN KEY (user_id) REFERENCES site_user(id),
    FOREIGN KEY (payment_method_id) REFERENCES user_payment_method(id),
    FOREIGN KEY (order_status_id) REFERENCES order_status(id)
);
```

18.Create order_line table

```
CREATE TABLE order_line (
    id INT PRIMARY KEY,
    product_item_id INT,
    order_id INT,
    qty INT,
    price DECIMAL(10, 2),
    FOREIGN KEY (product_item_id) REFERENCES product_item(id),
    FOREIGN KEY (order_id) REFERENCES shop_order(id)
);
```

19.Create order_status table

```
CREATE TABLE order_status (
    id INT PRIMARY KEY,
    status_name VARCHAR(50) NOT NULL
);
```

20.Create shipping_method table

```
CREATE TABLE shipping_method (
    id INT PRIMARY KEY,
    method_name VARCHAR(50) NOT NULL,
    cost DECIMAL(10, 2) NOT NULL
);
```

21.Create product_review table

```
CREATE TABLE product_review (
    id INT PRIMARY KEY,
    product_id INT,
    user_id INT,
    rating INT,
    review_text TEXT,
    review_date DATE,
    FOREIGN KEY (product_id) REFERENCES product(id),
    FOREIGN KEY (user_id) REFERENCES site_user(id)
);
```

9. Queries to Insert date in to Tables:

Insert into Country Table

```
INSERT INTO country (id, country_name) VALUES
(1, 'Pakistan'),
(2, 'India');
```

Insert into site_user Table

```
INSERT INTO site_user (id, email_address, phone_number, password) VALUES
(1, 'zohaibkhalid.pk@gmail.com', '03276027271', '11221122'),
(2, 'zohaibkhalid.fsd@gmail.com', '03124721951', '11221122'),
(3, 'khayam3812@gmail.com', '03247703152', 'password123'),
(4, 'esha@example.com', '03107654321', 'password123');
```

Insert into address Table

```
INSERT INTO address (id, unit_number, street_number, address_line1,
address_line2, city, region, postal_code, country_id) VALUES
(1, '12', '2', 'Haji Abad', 'BagBan Pura', 'Faisalabad', 'Punjab', '38000', 1),
(2, '56', '2', 'Haji Abad', 'BagBan Pura', 'Faisalabad', 'Punjab', '38000', 1),
(3, '90', '12', 'Akbar Chowk', 'Block Z', 'Faisalabad', 'Punjab', '38000', 1),
(5, 'A-102', '10', 'Green Park', 'Sector 18', 'New Delhi', 'Delhi', '110016', 2);
```

Insert into user_address table

```
INSERT INTO user_address (id, user_id, address_id, is_default) VALUES
(1, 1, 1, 1),
(2, 2, 2, 1),
(3, 3, 3, 1),
(4, 4, 5, 1);
```

Insert into payment_type table

```
INSERT INTO payment_type (id, value) VALUES
(1, 'Credit Card'),
(2, 'Debit Card'),
(3, 'Cash on Delivery'),
(4, 'Bank Transfer');
```

Insert into user_payment_method table

```
INSERT INTO user_payment_method (id, user_id, payment_type_id, provider,
account_number, expiry_date, is_default) VALUES
(1, 1, 1, 'HBL', '1234567890123456', '12/25', 1),
(2, 2, 2, 'MCB', '6543210987654321', '11/24', 1),
(3, 3, 3, 'UBL', '1111222233334444', '10/23', 1),
(4, 4, 4, 'HDFC', '4444333322221111', '09/25', 1);
```

Insert into product_category table

```
INSERT INTO product_category (id, parent_category_id, category_name) VALUES
(1, NULL, 'Electronics'),
(2, 1, 'Mobile Phones'),
(3, 1, 'Laptops'),
(4, 1, 'Accessories');
```

Insert into product table

```
INSERT INTO product (id, category_id, name, description) VALUES
(1, 2, 'iPhone 12', 'Apple iPhone 12 64GB'),
(2, 2, 'Samsung Galaxy S21', 'Samsung Galaxy S21 128GB'),
(3, 3, 'Dell Inspiron', 'Dell Inspiron 15 5000'),
(4, 3, 'MacBook Air', 'Apple MacBook Air M1'),
(5, 4, 'Samsung Charger', 'Samsung Fast Charger'),
(6, 4, 'HP Mouse', 'HP Wireless Mouse'),
(7, 2, 'OnePlus 9', 'OnePlus 9 128GB'),
(8, 2, 'Google Pixel 5', 'Google Pixel 5 128GB'),
(9, 3, 'Lenovo ThinkPad', 'Lenovo ThinkPad X1 Carbon'),
(10, 3, 'HP Spectre x360', 'HP Spectre x360 13-inch'),
(11, 4, 'Logitech Keyboard', 'Logitech Wireless Keyboard'),
(12, 4, 'Sony Headphones', 'Sony WH-1000XM4'),
(13, 2, 'Xiaomi Mi 11', 'Xiaomi Mi 11 256GB'),
(14, 2, 'Huawei P40', 'Huawei P40 Pro'),
(15, 3, 'Asus ZenBook', 'Asus ZenBook 14'),
(16, 3, 'Acer Aspire', 'Acer Aspire 5'),
(17, 4, 'Anker Power Bank', 'Anker 20000mAh Power Bank'),
(18, 4, 'Sandisk USB', 'Sandisk 128GB USB 3.0');
```

Insert into product_item table

```sql
INSERT INTO product_item (id, product_id, SKU, qty_in_stock, product_image,
price) VALUES
(1, 1, 'IPH12', 10, 'iphone12.jpg', 150000.00),
(2, 2, 'SGS21', 15, 'galaxy_s21.jpg', 120000.00),
(3, 3, 'DEL15', 8, 'dell_inspiron.jpg', 80000.00),
(4, 4, 'MBAIR', 12, 'macbook_air.jpg', 170000.00),
(5, 5, 'SAMCHG', 20, 'samsung_charger.jpg', 2000.00),
(6, 6, 'HPMSE', 25, 'hp_mouse.jpg', 1500.00),
(7, 7, 'ONE9', 10, 'oneplus_9.jpg', 110000.00),
(8, 8, 'PIX5', 10, 'pixel_5.jpg', 115000.00),
(9, 9, 'THINKX1', 5, 'thinkpad_x1.jpg', 180000.00),
(10, 10, 'HPSP13', 8, 'hp_spectre.jpg', 160000.00),
(11, 11, 'LOGKEY', 30, 'logitech_keyboard.jpg', 6000.00),
(12, 12, 'SONYXM4', 15, 'sony_headphones.jpg', 35000.00),
(13, 13, 'MI11', 20, 'xiaomi_mi11.jpg', 90000.00),
(14, 14, 'HUAP40', 12, 'huawei_p40.jpg', 105000.00),
(15, 15, 'ZENB14', 10, 'asus_zenbook.jpg', 130000.00),
(16, 16, 'ASPIRE5', 18, 'acer_aspire.jpg', 70000.00),
(17, 17, 'ANKPB', 25, 'anker_powerbank.jpg', 5000.00),
(18, 18, 'SANUSB', 50, 'sandisk_usb.jpg', 1500.00);
```

Insert into variation table

```sql
INSERT INTO variation (id, category_id, name) VALUES
(1, 2, 'Color'),
(2, 3, 'RAM'),
(3, 4, 'Type'),
(4, 4, 'Connectivity');
```

Insert into variation_option table

```sql
INSERT INTO variation_option (id, variation_id, value) VALUES
(1, 1, 'Black'),
(2, 1, 'White'),
(3, 2, '8GB'),
(4, 2, '16GB'),
(5, 3, 'Type-C'),
(6, 4, 'Wireless');
```

Insert into product_configuration table

```sql
INSERT INTO product_configuration (id, product_item_id, variation_option_id)
VALUES
(1, 1, 1),
(2, 1, 2),
(3, 2, 1),
(4, 2, 2),
(5, 3, 3),
(6, 3, 4),
(7, 4, 3),
(8, 4, 4),
(9, 5, 5),
(10, 6, 6);
```

Insert into promotion table

```sql
INSERT INTO promotion (id, name, description, discount_rate, start_date,
end_date) VALUES
(1, 'New Year Sale', '20% off on all items', 20.00, '2024-05-12', '2024-06-10'),
(2, 'Eid Offer', '10% off on selected items', 10.00, '2024-04-20', '2024-05-05'),
(3, 'Independence Day', '15% off on electronics', 15.00, '2024-08-14', '2024-08-
20'),
(4, 'Black Friday', '50% off on accessories', 50.00, '2024-11-25', '2024-11-30');
```

Insert into promotion_category table

```sql
INSERT INTO promotion_category (category_id, promotion_id) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4);
```

Insert into shopping_cart table

```sql
INSERT INTO shopping_cart (id, user_id) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4);
```

Insert into shopping_cart_item table

```
INSERT INTO shopping_cart_item (id, cart_id, product_item_id, qty) VALUES
(1, 1, 1, 2),
(2, 2, 2, 1),
(3, 3, 3, 5),
(4, 4, 4, 3);
```

Insert into shop_order table

```
INSERT INTO shop_order (id, user_id, order_date, payment_method_id,
shipping_address, shipping_method, order_total, order_status_id) VALUES
(1, 1, '2024-05-20', 1, '12 Haji Abad, BagBan Pura, Faisalabad', 'Standard
Shipping', 300000.00, 1),
(2, 2, '2024-05-21', 2, '56 Haji Abad, BagBan Pura, Faisalabad', 'Express
Shipping', 160000.00, 2),
(3, 3, '2024-05-22', 3, '90 Akbar Chowk, Block Z, Faisalabad', 'Standard
Shipping', 10000.00, 3),
(4, 4, '2024-05-23', 4, '34 College Road, Block D, Faisalabad', 'Overnight
Shipping', 4500.00, 4);
```

Insert into order_line table

```
INSERT INTO order_line (id, product_item_id, order_id, qty, price) VALUES
(1, 1, 1, 2, 150000.00),
(2, 2, 2, 1, 120000.00),
(3, 3, 3, 5, 80000.00),
(4, 4, 4, 3, 170000.00);
```

Insert into order_status table

```
INSERT INTO order_status (id, status_name) VALUES
(1, 'Pending'),
(2, 'Processing'),
(3, 'Completed'),
(4, 'Cancelled');
```

Insert into shipping_method table

```sql
INSERT INTO shipping_method (id, method_name, cost) VALUES
(1, 'Standard Shipping', 50.00),
(2, 'Express Shipping', 100.00),
(3, 'Overnight Shipping', 200.00);
```

Insert into product_review table

```sql
INSERT INTO product_review (id, product_id, user_id, rating, review_text,
review_date) VALUES
(1, 1, 1, 5, 'Excellent product!', '2024-01-01'),
(2, 2, 2, 4, 'Very good phone.', '2024-01-02'),
(3, 3, 3, 3, 'Average laptop.', '2024-01-03'),
(4, 4, 4, 5, 'Amazing MacBook!', '2024-01-04');
```

10. Queries to select data and apply functions

1-

```sql
SELECT * FROM site_user;

SELECT * FROM site_user WHERE email_address = 'specific_email@example.com';

UPDATE site_user SET last_login = CURRENT_TIMESTAMP WHERE email_address =
'specific_email@example.com';

UPDATE site_user SET password = 'new_hashed_password' WHERE email_address =
'specific_email@example.com';

SELECT is_admin FROM site_user WHERE email_address =
'specific_email@example.com';
```

2-

```sql
SELECT * FROM product_category;

SELECT * FROM product_category WHERE id = 'category_id';

SELECT * FROM product_category WHERE category_name = 'category_name';

UPDATE product_category SET category_name = 'new_category_name' WHERE id = 'category_id';

DELETE FROM product_category WHERE id = 'category_id';
```

3-

```sql
SELECT * FROM product;

SELECT * FROM product WHERE id = 'product_id';

SELECT * FROM product WHERE name LIKE '%product_name%';

UPDATE product SET name = 'new_product_name' WHERE id = 'product_id';

DELETE FROM product WHERE id = 'product_id';
```

4-

```sql
SELECT * FROM product_item;

SELECT * FROM product_item WHERE id = 'product_item_id';

SELECT * FROM product_item WHERE SKU = 'SKU';

UPDATE product_item SET qty_in_stock = 100 WHERE id = 'product_item_id';

DELETE FROM product_item WHERE id = 'product_item_id';
```

5-

```
SELECT * FROM shopping_cart;

SELECT * FROM shopping_cart WHERE user_id = 'user_id';

DELETE FROM shopping_cart WHERE user_id = 'user_id';
```

6-

```
SELECT * FROM shopping_cart_item;

SELECT * FROM shopping_cart_item WHERE cart_id = 'cart_id';

DELETE FROM shopping_cart_item WHERE cart_id = 'cart_id' AND product_item_id =
'product_item_id';
```

7-

```
SELECT * FROM shop_order;

SELECT * FROM shop_order WHERE id = 'order_id';

UPDATE shop_order SET order_status_id = 2 WHERE id = 'order_id';

DELETE FROM shop_order WHERE id = 'order_id';
```

8-

```
SELECT * FROM order_line;

SELECT * FROM order_line WHERE order_id = 'order_id';

DELETE FROM order_line WHERE order_id = 'order_id' AND product_item_id =
'product_item_id';
```

9-

```
SELECT * FROM product_review;

SELECT * FROM product_review WHERE product_id = 'product_id';

DELETE FROM product_review WHERE product_id = 'product_id' AND user_id =
'user_id';
```

10-

```
SELECT * FROM product_configuration;

SELECT * FROM product_configuration WHERE product_item_id = 'product_item_id';

DELETE FROM product_configuration WHERE product_item_id = 'product_item_id' AND
variation_option_id = 'variation_option_id';
SELECT * FROM variation;

SELECT * FROM variation WHERE id = 'variation_id';

DELETE FROM variation WHERE id = 'variation_id';
```

11-

```
SELECT * FROM variation_option;

SELECT * FROM variation_option WHERE id = 'variation_option_id';

DELETE FROM variation_option WHERE id = 'variation_option_id';
```

12-

```
SELECT * FROM promotion;

SELECT * FROM promotion WHERE id = 'promotion_id';

UPDATE promotion SET name = 'new_promotion_name' WHERE id = 'promotion_id';

DELETE FROM promotion WHERE id = 'promotion_id';
```

13

```sql
SELECT * FROM promotion_category;

SELECT * FROM promotion_category WHERE category_id = 'category_id';

DELETE FROM promotion_category WHERE category_id = 'category_id' AND promotion_id = 'promotion_id';
```

14-

```sql
SELECT * FROM country;

SELECT * FROM country WHERE id = 'country_id';

DELETE FROM country WHERE id = 'country_id';
```

15-

```sql
SELECT * FROM address;

SELECT * FROM address WHERE id = 'address_id';

UPDATE address SET city = 'new_city' WHERE id = 'address_id';

DELETE FROM address WHERE id = 'address_id';
SELECT * FROM user_address;

SELECT * FROM user_address WHERE user_id = 'user_id';

DELETE FROM user_address WHERE user_id = 'user_id' AND address_id = 'address_id';
```

16-

```sql
SELECT * FROM payment_type;

SELECT * FROM payment_type WHERE id = 'payment_type_id';

DELETE FROM payment_type WHERE id = 'payment_type_id';
```

17-

```
SELECT * FROM user_payment_method;

SELECT * FROM user_payment_method WHERE user_id = 'user_id';

DELETE FROM user_payment_method WHERE user_id = 'user_id' AND payment_type_id =
'payment_type_id';
```

18-

```
SELECT * FROM shipping_method;

SELECT * FROM shipping_method WHERE id = 'shipping_method_id';

DELETE FROM shipping_method WHERE id = 'shipping_method_id';
```

19-

```
SELECT * FROM order_status;

SELECT * FROM order_status WHERE id = 'order_status_id';

DELETE FROM order_status WHERE id = 'order_status_id';
```

(The other 3 tables have just numbers in it, as primary and foreign keys so I have'nt selected data from them**)**

Appling SQL Functions

- Determine user type based on is_admin flag

```
SELECT email_address,
CASE WHEN is_admin = 1 THEN 'Admin' ELSE 'Regular User' END AS user_type
FROM site_user;
```

- calculate total number of products

```
SELECT SUM(id) AS total_products FROM product;
```

- DISTINCT: Select distinct cities from address table

```
SELECT DISTINCT city FROM address;
```

- Calculate the total number of users

```
SELECT COUNT(*) AS total_users FROM site_user;
```

- Calculate the average rating for a specific product

```
SELECT AVG(rating) AS average_rating FROM product_review WHERE product_id = 1;
```

- Calculate the total revenue from all orders

```
SELECT SUM(order_total) AS total_revenue FROM shop_order;
```

- Convert product names to uppercase

```
SELECT UPPER(name) AS product_name_uppercase FROM product;
```

12.Queries to select data from Multiple tables

- Retrieve the product names and their corresponding category names

```
SELECT p.name AS product_name, c.category_name AS category_name
FROM product p
INNER JOIN product_category c ON p.category_id = c.id;
```

- Retrieve the cart details including user, product item SKU, and quantity

```sql
SELECT sc.id AS cart_id, sc.user_id, pi.SKU AS product_item_sku, sci.qty
FROM shopping_cart sc
INNER JOIN shopping_cart_item sci ON sc.id = sci.cart_id
INNER JOIN product_item pi ON sci.product_item_id = pi.id;
```

- Retrieve the order details including order line items and product information

```sql
SELECT o.id AS order_id, ol.id AS order_line_id, p.name AS product_name, pi.SKU
AS product_item_sku, ol.qty, ol.price
FROM shop_order o
INNER JOIN order_line ol ON o.id = ol.order_id
INNER JOIN product_item pi ON ol.product_item_id = pi.id
INNER JOIN product p ON pi.product_id = p.id;
```

- Retrieve product reviews along with the corresponding product and user details

```sql
SELECT p.name AS product_name, r.rating, r.review_text, r.review_date,
u.email_address
FROM product_review r
INNER JOIN product p ON r.product_id = p.id
INNER JOIN site_user u ON r.user_id = u.id;
```

- Retrieve shopping cart details including number of items per category

```sql
SELECT sc.id AS cart_id, c.id AS category_id, c.category_name,
COUNT(sci.product_item_id) AS num_items
FROM shopping_cart sc
INNER JOIN shopping_cart_item sci ON sc.id = sci.cart_id
INNER JOIN product_item pi ON sci.product_item_id = pi.id
INNER JOIN product p ON pi.product_id = p.id
INNER JOIN product_category c ON p.category_id = c.id
GROUP BY sc.id, c.id, c.category_name;
```

- Retrieve order details along with user and order status, ordered by order date

```sql
SELECT o.id AS order_id, o.order_date, u.email_address, os.status_name
FROM shop_order o
INNER JOIN site_user u ON o.user_id = u.id
INNER JOIN order_status os ON o.order_status_id = os.id
ORDER BY o.order_date;
```

- Retrieve shopping cart items along with product details

```sql
SELECT p.name AS product_name, pi.SKU AS product_item_sku, sci.qty
FROM shopping_cart sc
INNER JOIN shopping_cart_item sci ON sc.id = sci.cart_id
INNER JOIN product_item pi ON sci.product_item_id = pi.id
INNER JOIN product p ON pi.product_id = p.id;
```

- Retrieve product reviews along with user email addresses

```sql
SELECT p.name AS product_name, r.rating, u.email_address
FROM product_review r
INNER JOIN product p ON r.product_id = p.id
INNER JOIN site_user u ON r.user_id = u.id;
```

13.Future work

Future Work to Optimize the System

1. **Indexing:** Add indexes to frequently queried columns to improve search speed.
2. **Query Optimization**: Optimize SQL queries to reduce execution time.
3. **Data Partitioning:** Partition large tables to enhance performance and manageability.
4. **Caching**: Implement caching for frequently accessed data to reduce database load.
5. **Load Balancing:** Distribute database load across multiple servers for better performance.
6. **Sharding:** Use sharding to handle large datasets and high traffic efficiently.
7. **Regular** Maintenance: Schedule re-indexing and data cleanup tasks regularly.
8. **Security Enhancements:** Strengthen data encryption and backup processes.
9. **Scalability:** Design the system to accommodate future growth in data and traffic.