DATABASE MANAGEMENT SYSTEM

GROU

GROUP 2

(Shopping Website)

GROUP MEMBERS

EMİRHAN ATAR

HAKAN BERKİTEN

Ş.HİLAL HOCAOĞLU

ABDULLAH SEVİNÇ

KEMAL ALTUĞ İKİZ

WHAT IS OUR MISSION

Our mission is to design and implement a robust, scalable, and secure e-commerce database that empowers businesses to deliver seamless shopping experiences

Who is our target audience ?

Our target audience is e-commerce businesses and developers who need a flexible, scalable database—complete with multi-currency support, saved payment methods, and dynamic product management—that can grow and adapt alongside their site.

What did we do for our goal?

We designed a database that keeps product, customer, order ,currency , adress and card information in a seamless and responsive manner. We used REACT, MYSQL , PYTHON & LUCIDCHART for our project.

Fact Finding Techniques

A

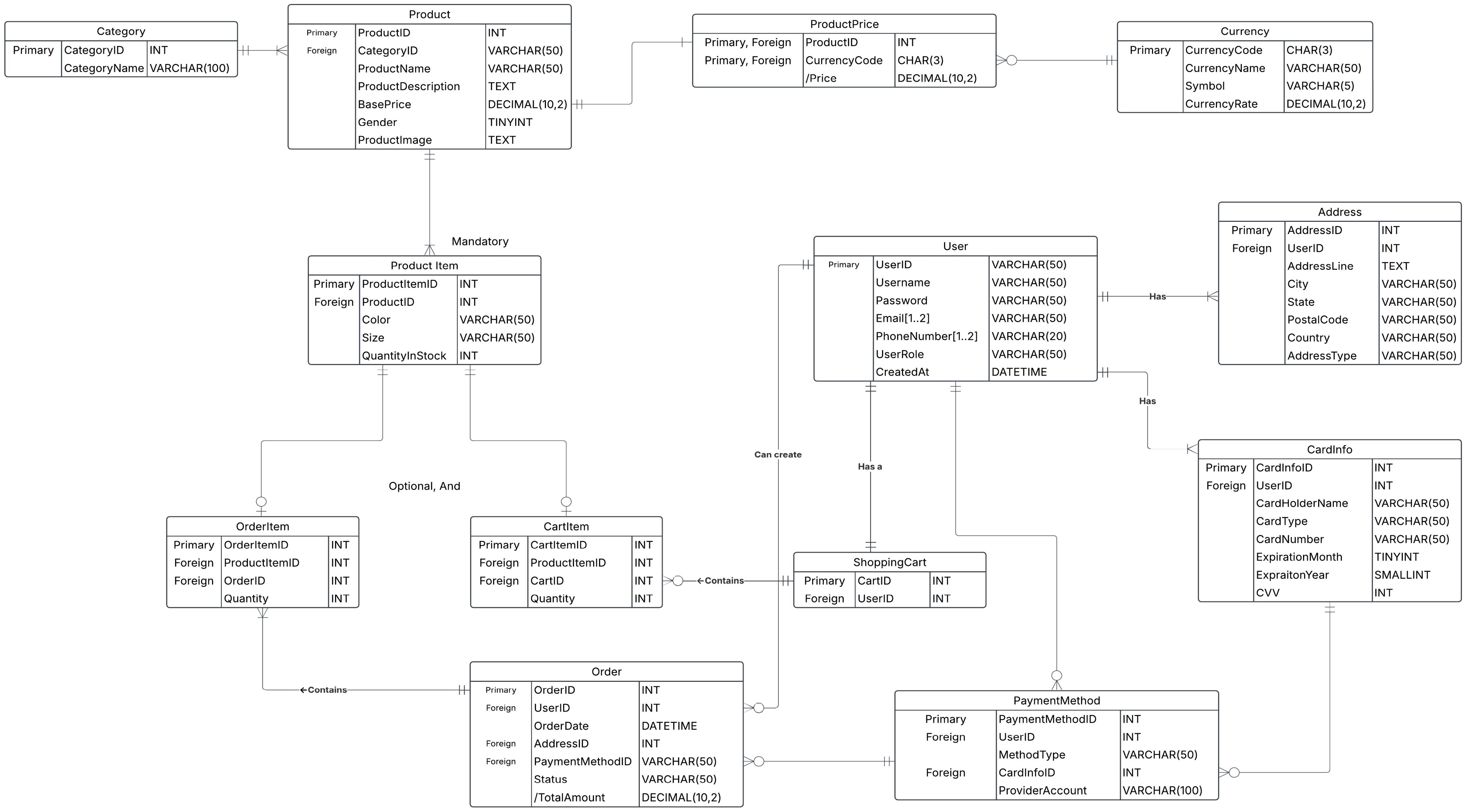
RESEARCH

C

OBSERVING

Why should you invest in this project ?

Imagine converting every visitor into a happy customer with instant feature discovery and seamless checkout. Our flexible, scalable database lets you launch new features effortlessly—driving growth and boosting returns.



Category

Primary	CategoryID	INT
	CategoryName	VARCHAR(100)

Product

Primary	ProductID	INT
Foreign	CategoryID	VARCHAR(50)
	ProductName	VARCHAR(50)
	ProductDescription	TEXT
	BasePrice	DECIMAL(10,2)
	Gender	TINYINT
	ProductImage	TEXT

ProductPrice

Primary, Foreign	ProductID	INT
Primary, Foreign	CurrencyCode	CHAR(3)
	/Price	DECIMAL(10,2)

Currency

Primary	CurrencyCode	CHAR(3)
	CurrencyName	VARCHAR(50)
	Symbol	VARCHAR(5)
	CurrencyRate	DECIMAL(10,2)

Product Item

Primary	ProductItemID	INT
Foreign	ProductID	INT
	Color	VARCHAR(50)
	Size	VARCHAR(50)
	QuantityInStock	INT

User

Primary	UserID	VARCHAR(50)
	Username	VARCHAR(50)
	Password	VARCHAR(50)
	Email[1..2]	VARCHAR(50)
	PhoneNumber[1..2]	VARCHAR(20)
	UserRole	VARCHAR(50)
	CreatedAt	DATETIME

Address

Primary	AddressID	INT
Foreign	UserID	INT
	AddressLine	TEXT
	City	VARCHAR(50)
	State	VARCHAR(50)
	PostalCode	VARCHAR(50)
	Country	VARCHAR(50)
	AddressType	VARCHAR(50)

CardInfo

Primary	CardInfoID	INT
Foreign	UserID	INT
	CardHolderName	VARCHAR(50)
	CardType	VARCHAR(50)
	CardNumber	VARCHAR(50)
	ExpirationMonth	TINYINT
	ExpraitonYear	SMALLINT
	CVV	INT

OrderItem

Primary	OrderItemID	INT
Foreign	ProductItemID	INT
Foreign	OrderID	INT
	Quantity	INT

CartItem

Primary	CartItemID	INT
Foreign	ProductItemID	INT
Foreign	CartID	INT
	Quantity	INT

ShoppingCart

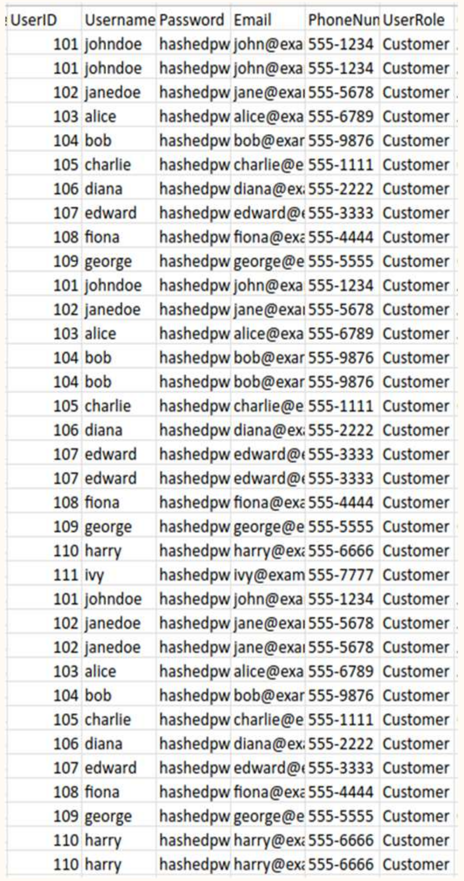
Primary	CartID	INT
Foreign	UserID	INT

Order

Primary	OrderID	INT
Foreign	UserID	INT
	OrderDate	DATETIME
Foreign	AddressID	INT
Foreign	PaymentMethodID	VARCHAR(50)
	Status	VARCHAR(50)
	/TotalAmount	DECIMAL(10,2)

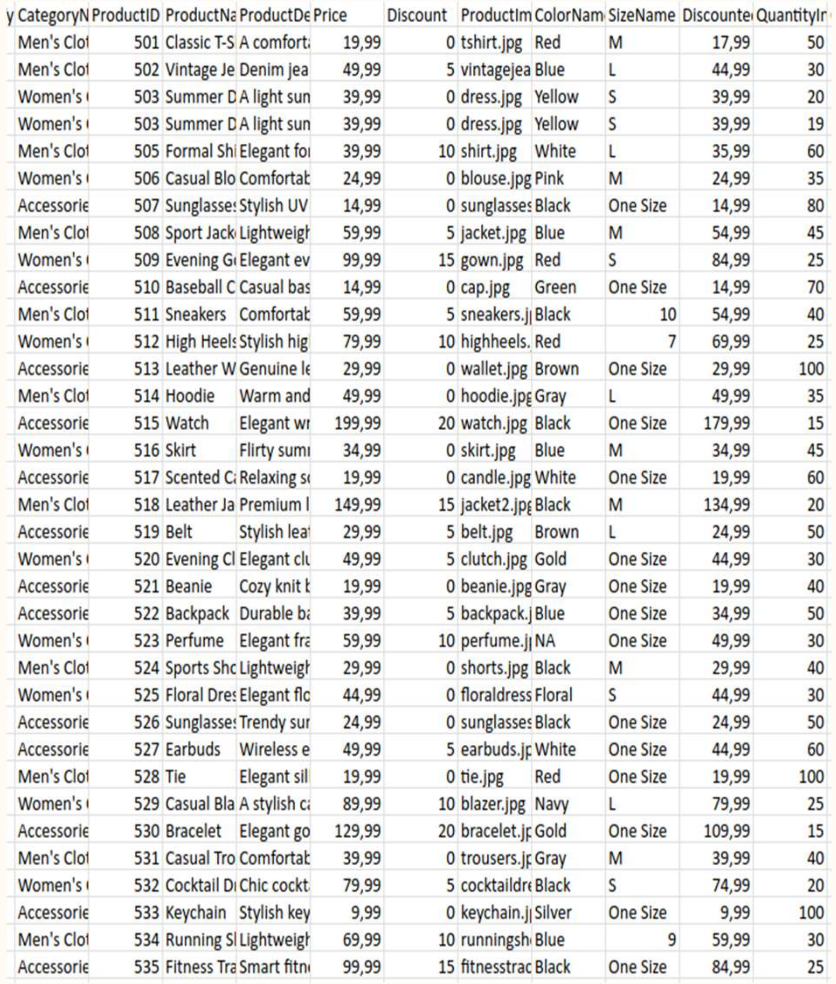
PaymentMethod

Primary	PaymentMethodID	INT
Foreign	UserID	INT
	MethodType	VARCHAR(50)
Foreign	CardInfoID	INT
	ProviderAccount	VARCHAR(100)



UserID	Username	Password	Email	PhoneNun	UserRole
101	johndoe	hashedpw	john@exa	555-1234	Customer
101	johndoe	hashedpw	john@exa	555-1234	Customer
102	janedoe	hashedpw	jane@exa	555-5678	Customer
103	alice	hashedpw	alice@exa	555-6789	Customer
104	bob	hashedpw	bob@exar	555-9876	Customer
105	charlie	hashedpw	charlie@e	555-1111	Customer
106	diana	hashedpw	diana@ex	555-2222	Customer
107	edward	hashedpw	edward@	555-3333	Customer
108	fiona	hashedpw	fiona@exa	555-4444	Customer
109	george	hashedpw	george@e	555-5555	Customer
101	johndoe	hashedpw	john@exa	555-1234	Customer
102	janedoe	hashedpw	jane@exa	555-5678	Customer
103	alice	hashedpw	alice@exa	555-6789	Customer
104	bob	hashedpw	bob@exar	555-9876	Customer
104	bob	hashedpw	bob@exar	555-9876	Customer
105	charlie	hashedpw	charlie@e	555-1111	Customer
106	diana	hashedpw	diana@ex	555-2222	Customer
107	edward	hashedpw	edward@	555-3333	Customer
107	edward	hashedpw	edward@	555-3333	Customer
108	fiona	hashedpw	fiona@exa	555-4444	Customer
109	george	hashedpw	george@e	555-5555	Customer
110	harry	hashedpw	harry@exa	555-6666	Customer
111	ivy	hashedpw	ivy@exam	555-7777	Customer
101	johndoe	hashedpw	john@exa	555-1234	Customer
102	janedoe	hashedpw	jane@exa	555-5678	Customer
102	janedoe	hashedpw	jane@exa	555-5678	Customer
103	alice	hashedpw	alice@exa	555-6789	Customer
104	bob	hashedpw	bob@exar	555-9876	Customer
105	charlie	hashedpw	charlie@e	555-1111	Customer
106	diana	hashedpw	diana@ex	555-2222	Customer
107	edward	hashedpw	edward@	555-3333	Customer
108	fiona	hashedpw	fiona@exa	555-4444	Customer
109	george	hashedpw	george@e	555-5555	Customer
110	harry	hashedpw	harry@exa	555-6666	Customer
110	harry	hashedpw	harry@exa	555-6666	Customer

```sql
CREATE TABLE `User` (
    UserID INT AUTO_INCREMENT,
    Username VARCHAR(50) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    PhoneNumber VARCHAR(20),
    UserRole VARCHAR(50),
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (UserID),
    UNIQUE (Username),
    UNIQUE (Email)
);
INSERT INTO `User` (UserID, Username, Password, Email, PhoneNumber,
UserRole)
SELECT DISTINCT
    UserID,
    Username,
    Password,
    Email,
    PhoneNumber,
    UserRole
FROM Unnormalized_Data;
```



y	CategoryN	ProductID	ProductNa	ProductDe	Price	Discount	ProductIm	ColorNam	SizeName	Discounte	QuantityIr
	Men's Clot	501	Classic T-S	A comfort	19,99	0	tshirt.jpg	Red	M	17,99	50
	Men's Clot	502	Vintage Je	Denim jea	49,99	5	vintagejea	Blue	L	44,99	30
	Women's	503	Summer D	A light sun	39,99	0	dress.jpg	Yellow	S	39,99	20
	Women's	503	Summer D	A light sun	39,99	0	dress.jpg	Yellow	S	39,99	19
	Men's Clot	505	Formal Shi	Elegant fo	39,99	10	shirt.jpg	White	L	35,99	60
	Women's	506	Casual Blo	Comfortab	24,99	0	blouse.jpg	Pink	M	24,99	35
	Accessorie	507	Sunglasses	Stylish UV	14,99	0	sunglasses	Black	One Size	14,99	80
	Men's Clot	508	Sport Jack	Lightweigh	59,99	5	jacket.jpg	Blue	M	54,99	45
	Women's	509	Evening G	Elegant ev	99,99	15	gown.jpg	Red	S	84,99	25
	Accessorie	510	Baseball C	Casual bas	14,99	0	cap.jpg	Green	One Size	14,99	70
	Men's Clot	511	Sneakers	Comfortab	59,99	5	sneakers.j	Black	10	54,99	40
	Women's	512	High Heels	Stylish hig	79,99	10	highheels.	Red	7	69,99	25
	Accessorie	513	Leather W	Genuine le	29,99	0	wallet.jpg	Brown	One Size	29,99	100
	Men's Clot	514	Hoodie	Warm and	49,99	0	hoodie.jpg	Gray	L	49,99	35
	Accessorie	515	Watch	Elegant wr	199,99	20	watch.jpg	Black	One Size	179,99	15
	Women's	516	Skirt	Flirty sum	34,99	0	skirt.jpg	Blue	M	34,99	45
	Accessorie	517	Scented Ca	Relaxing s	19,99	0	candle.jpg	White	One Size	19,99	60
	Men's Clot	518	Leather Ja	Premium l	149,99	15	jacket2.jpg	Black	M	134,99	20
	Accessorie	519	Belt	Stylish lea	29,99	5	belt.jpg	Brown	L	24,99	50
	Women's	520	Evening Cl	Elegant clu	49,99	5	clutch.jpg	Gold	One Size	44,99	30
	Accessorie	521	Beanie	Cozy knit b	19,99	0	beanie.jpg	Gray	One Size	19,99	40
	Accessorie	522	Backpack	Durable ba	39,99	5	backpack.j	Blue	One Size	34,99	50
	Women's	523	Perfume	Elegant fra	59,99	10	perfume.j	NA	One Size	49,99	30
	Men's Clot	524	Sports Sho	Lightweigh	29,99	0	shorts.jpg	Black	M	29,99	40
	Women's	525	Floral Dres	Elegant flo	44,99	0	floraldress	Floral	S	44,99	30
	Accessorie	526	Sunglasses	Trendy sur	24,99	0	sunglasses	Black	One Size	24,99	50
	Accessorie	527	Earbuds	Wireless e	49,99	5	earbuds.jp	White	One Size	44,99	60
	Men's Clot	528	Tie	Elegant sil	19,99	0	tie.jpg	Red	One Size	19,99	100
	Women's	529	Casual Bla	A stylish ca	89,99	10	blazer.jpg	Navy	L	79,99	25
	Accessorie	530	Bracelet	Elegant go	129,99	20	bracelet.jp	Gold	One Size	109,99	15
	Men's Clot	531	Casual Tro	Comfortab	39,99	0	trousers.jp	Gray	M	39,99	40
	Women's	532	Cocktail Di	Chic cockt	79,99	5	cocktaildre	Black	S	74,99	20
	Accessorie	533	Keychain	Stylish key	9,99	0	keychain.j	Silver	One Size	9,99	100
	Men's Clot	534	Running Sl	Lightweigh	69,99	10	runningsh	Blue	9	59,99	30
	Accessorie	535	Fitness Tra	Smart fitn	99,99	15	fitnesstrac	Black	One Size	84,99	25

```sql
CREATE TABLE Product (
    ProductID INT AUTO_INCREMENT,
    CategoryID INT NOT NULL,
    ProductName VARCHAR(50) NOT NULL,
    ProductDescription TEXT,
    Price DECIMAL(10,2) NOT NULL,
    Gender TINYINT,
    Discount DECIMAL(10,2) DEFAULT 0,
    ProductImage TEXT,
    PRIMARY KEY (ProductID),
    FOREIGN KEY (CategoryID) REFERENCES
Category(CategoryID)
);
INSERT INTO Product (ProductID, CategoryID, ProductName,
ProductDescription, Price, Gender, Discount, ProductImage)
SELECT DISTINCT
    ud.ProductID,
    (SELECT CategoryID FROM Category WHERE CategoryName =
ud.CategoryName) AS CategoryID,
    ud.ProductName,
    ud.ProductDescription,
    ud.Price,
    0 AS Gender,
    ud.Discount,
    ud.ProductImage
FROM Unnormalized_Data ud;
```

Last queries for customer request

```
CREATE TABLE ProductPrice (
    ProductID INT  NOT NULL,
    CurrencyCode  CHAR(3)  NOT NULL,
    Price  DECIMAL(10,2) NOT NULL,
    PRIMARY KEY (ProductItemID, CurrencyCode),
    FOREIGN KEY (ProductItemID) REFERENCES ProductItem(ProductItemID),
    FOREIGN KEY (CurrencyCode)  REFERENCES Currency(CurrencyCode)
);
CREATE TABLE Currency (
    CurrencyCode CHAR(3)  PRIMARY KEY,
    CurrencyName VARCHAR(50),
    Symbol  VARCHAR(5),
    CurrencyRate DECIMAL(10,2)
);
```

Keeping track of prices in multiple currencies for customers.

```sql
CREATE TABLE PaymentMethod (
    PaymentMethodID INT AUTO_INCREMENT,
    UserID    INT  NOT NULL,
    MethodType     VARCHAR(50)  NOT NULL,
    CardInfoID      INT NULL,
    ProviderAccount VARCHAR(100),
    PRIMARY KEY (PaymentMethodID),
    FOREIGN KEY (UserID)     REFERENCES `User`(UserID),
    FOREIGN KEY (CardInfoID) REFERENCES CardInfo(CardInfoID)
);

ALTER TABLE `Order`
    DROP COLUMN PaymentMethod,
    ADD  COLUMN PaymentMethodID INT NOT NULL,
    ADD  CONSTRAINT fk_Order_PaymentMethod
    FOREIGN KEY (PaymentMethodID) REFERENCES PaymentMethod(PaymentMethodID);
```

A customer applying one of their payment type for a particular order

CUSTOMER REQUESTS

1. List all customers who have ordered a particular product.
2. Show the maximum, minimum, and average price of all products based on category.
3. Order a product and change its stock value.
4. Add a new card for a customer.
5. Remove a product from the system and from all carts in which it appears.

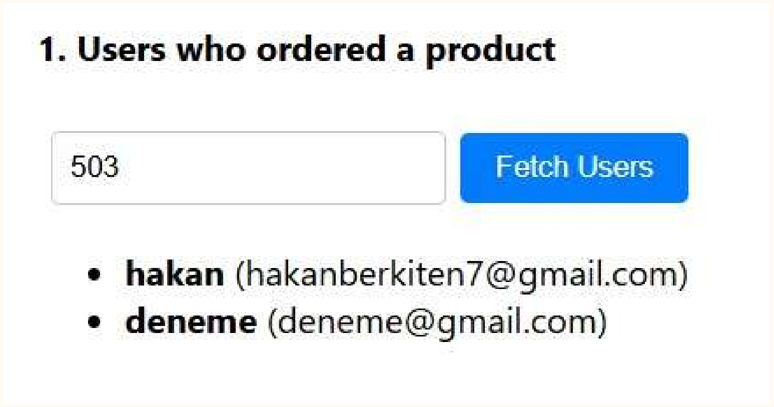
List all customers who have ordered a particular product.

//List each customer who has ordered a specific product

```
SELECT
   DISTINCT u.Username, u.Email

FROM order o
   JOIN orderitem oi ON o.OrderID = oi.OrderID
   JOIN productitem pi ON oi.ProductItemID = pi.ProductItemID
   JOIN product p ON pi.ProductID = p.ProductID
   JOIN user u ON o.UserID = u.UserID

WHERE  p.ProductID = 503;
```



1. Users who ordered a product

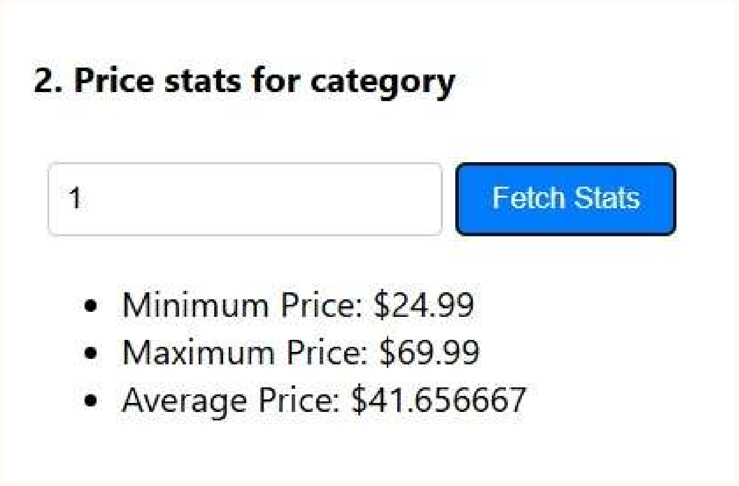
503 | Fetch Users

- **hakan** (hakanberkiten7@gmail.com)
- **deneme** (deneme@gmail.com)

Show the maximum, minimum, and average price of all products based on category.

// Calculate the highest, lowest, and average product prices for a specific category

```
SELECT
    MAX(Price) AS max_price,
    MIN(Price)  AS min_price,
    AVG(Price)  AS avg_price
FROM  product
GROUP BY CategoryID
HAVING CategoryID = 1;
```



2. Price stats for category

1

- Minimum Price: $24.99
- Maximum Price: $69.99
- Average Price: $41.656667

Order a product and change its stock value.

1) Create the order

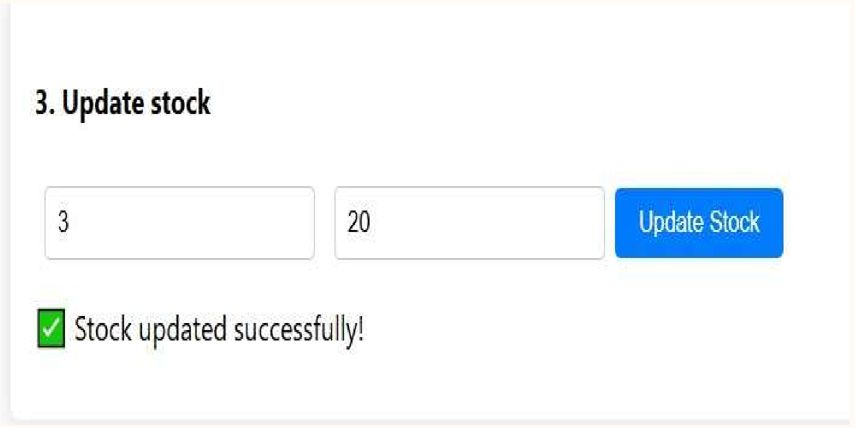
```
INSERT INTO `Order` (
            UserID,
            OrderDate,
            AddressID,
            PaymentMethodID,
            Status,
            TotalAmount)
VALUES (%s, NOW(), %s, %s, 'Placed', %s);
SET  @NewOrderID = LAST_INSERT_ID();
```

2) Add the item

```
INSERT INTO OrderItem (OrderID, ProductItemID, Quantity)
VALUES (@NewOrderID, %s, %s);
```

3) update stock

```
UPDATE
     ProductItem
SET
     QuantityInStock = QuantityInStock - %s
WHERE
     ProductItemID = %s;
```



3. Update stock

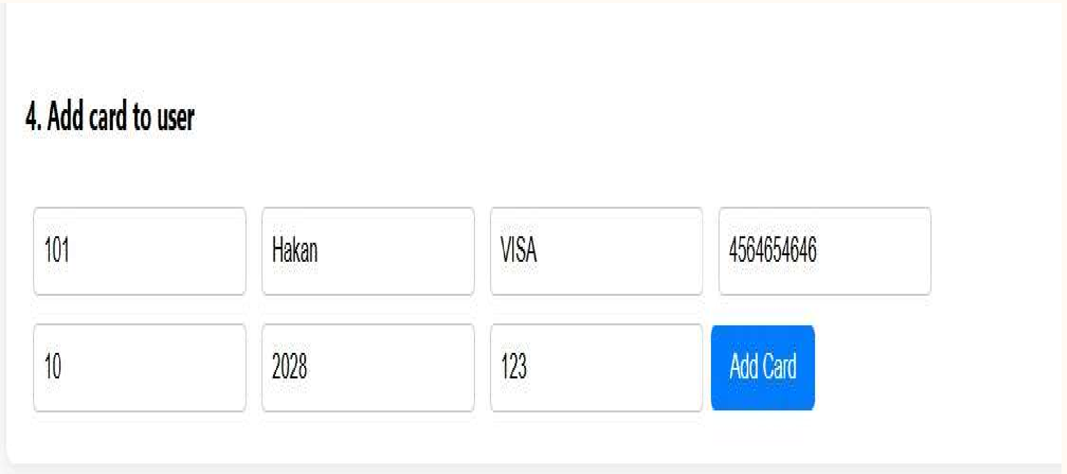
3 | 20

✓ Stock updated successfully!

Add a new card for a customer.

```
// Insert a new payment card for a customer

INSERT INTO cardinfo (
                     UserID,
                     CardHolderName,
                     CardType,
                     CardNumber,
                     ExpirationMonth,
                     ExpirationYear,
                    CVV)
VALUES (%s, %s, %s, %s, %s, %s, %s);
```



4. Add card to user

101	Hakan	VISA	4564654646
10	2028	123	Add Card

Remove a product from the system and from all carts in which it appears.

1) Remove any cart entries for the targeted product's variants

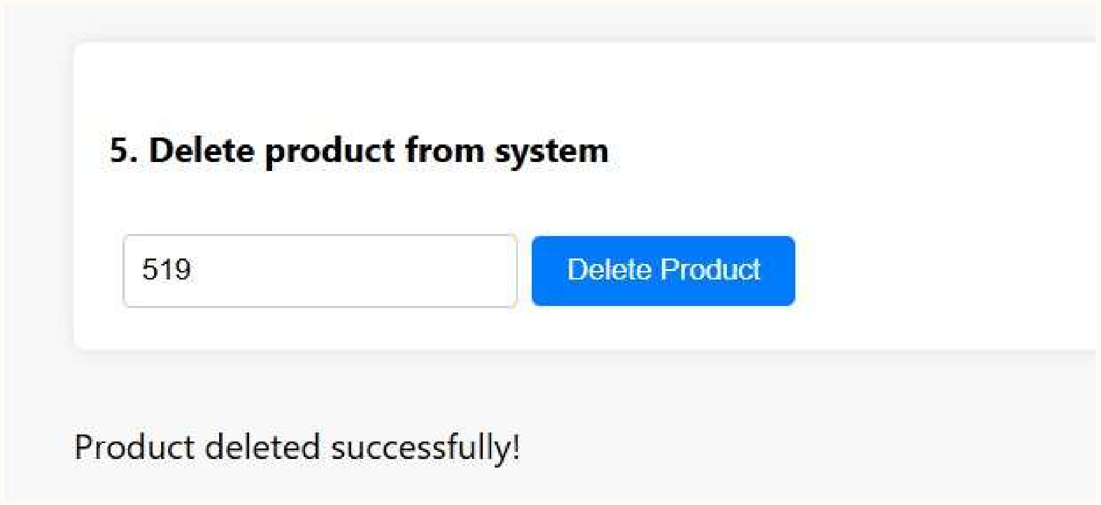
```
DELETE FROM `CartItem`
 WHERE ProductItemID IN (
                        SELECT ProductItemID
                        FROM `ProductItem`
                       WHERE ProductID = %s);
```

2) Delete all specific item variants for that product

```
DELETE FROM `ProductItem`
 WHERE ProductID = %s;
```

3) Finally, delete the product record itself

```
DELETE FROM `Product`
 WHERE ProductID = %s;
```

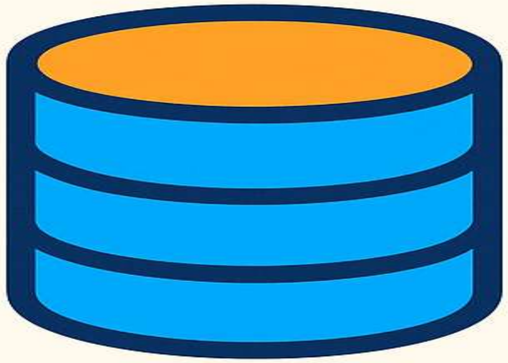


5. Delete product from system

519

Product deleted successfully!

Other queries we used



Getting customer cart

```
SELECT p.ProductID AS id, p.ProductName AS name,
p.Price, p.Discount, p.ProductImage AS image,   ci.Quantity
 FROM cartitem ci
 JOIN shoppingcart sc ON ci.CartID = sc.CartID
JOIN productitem pi ON ci.ProductItemID =
pi.ProductItemID
 JOIN product p ON pi.ProductID = p.ProductID
 WHERE sc.UserID = %s
```

This query provides signup for customer

```
INSERT INTO user (Username, Email, Password,
PhoneNumber, UserRole, CreatedAt)
 VALUES (%s, %s, %s, %s, %s, %s)
```

It saves the address information for the user

```
INSERT INTO address (UserID, AddressLine, City, State,
PostalCode, Country, AddressType)
 VALUES (%s, %s, %s, %s, %s, %s, %s)
```

THANK YOU FOR YOUR TIME