SVKM's NMIMS

School of Technology Management & Engineering, Chandigarh

A.Y. 2023 - 24

Course: Database Management Systems

Project Report

Program	B.tech Computer Engineering	
Semester	4	
Name of the Project:	E-COMMERCE MANAGEMENT	
Details of Project Members		
Batch 1	Roll No- A163 Roll No- A176 Roll No- A177	Devanshi Srivastava Yash Patil Srushti Mishrikotkar
Date of Submission: 02-04-2024		

Contribution of each project Members:

Roll No.:	**Name**	**Contribution**
A163	**Devanshi Srivastava**	**Query/Information/ Conclusion**
A176	**Yash Patil**	**E-R Diagram/Relational Model/Query**
A177	**Srushti Mishrikotkar**	**Query/Storyline/Filled up the information/Reviews**

Rubrics for the Project evaluation:

First phase of evaluation: Innovative Ideas (5 Marks) Design and Partial implementation (5 Marks)	10 marks
Final phase of evaluation Implementation, presentation and viva, Self-Learning and Learning Beyond classroom	10 marks

Table of Contents

STORYLINE-

"E-Commerce Emporium" is an innovative online marketplace that leverages a sophisticated database management system (DBMS) to provide customers with a seamless shopping experience. Customers register for accounts, browse through a diverse product catalog, and effortlessly place orders. The DBMS efficiently manages product inventory, order processing, and payment transactions, ensuring accuracy and security throughout. Sellers benefit from streamlined operations, timely order notifications, and access to valuable customer feedback. Continuous analysis of data stored in the DBMS allows for ongoing optimization, driving improvements in inventory management, user experience, and overall performance. As a result, "E-Commerce Emporium" emerges as a trusted destination for online shopping, renowned for its reliability, convenience, and commitment to customer satisfaction.

II. Components of Database Design

User (customer_id, name, email, address, phone_number, created_at, updated_at)

Order (order_id, customer_id, order_date, total_amount, status, created_at, updated_at)

Order Item (order_item_id, order_id, product_id, quantity, price, subtotal)

Product (product_id, name, description, price, stock_quantity, category_id, seller_id, created_at, updated_at)

Review (review_id, product_id, customer_id, rating, comment, created_at, updated_at)

Cart (cart_id, customer_id, created_at, updated_at)

Category (category_id, name, description)

Seller (seller_id, name, email, address, phone_number)

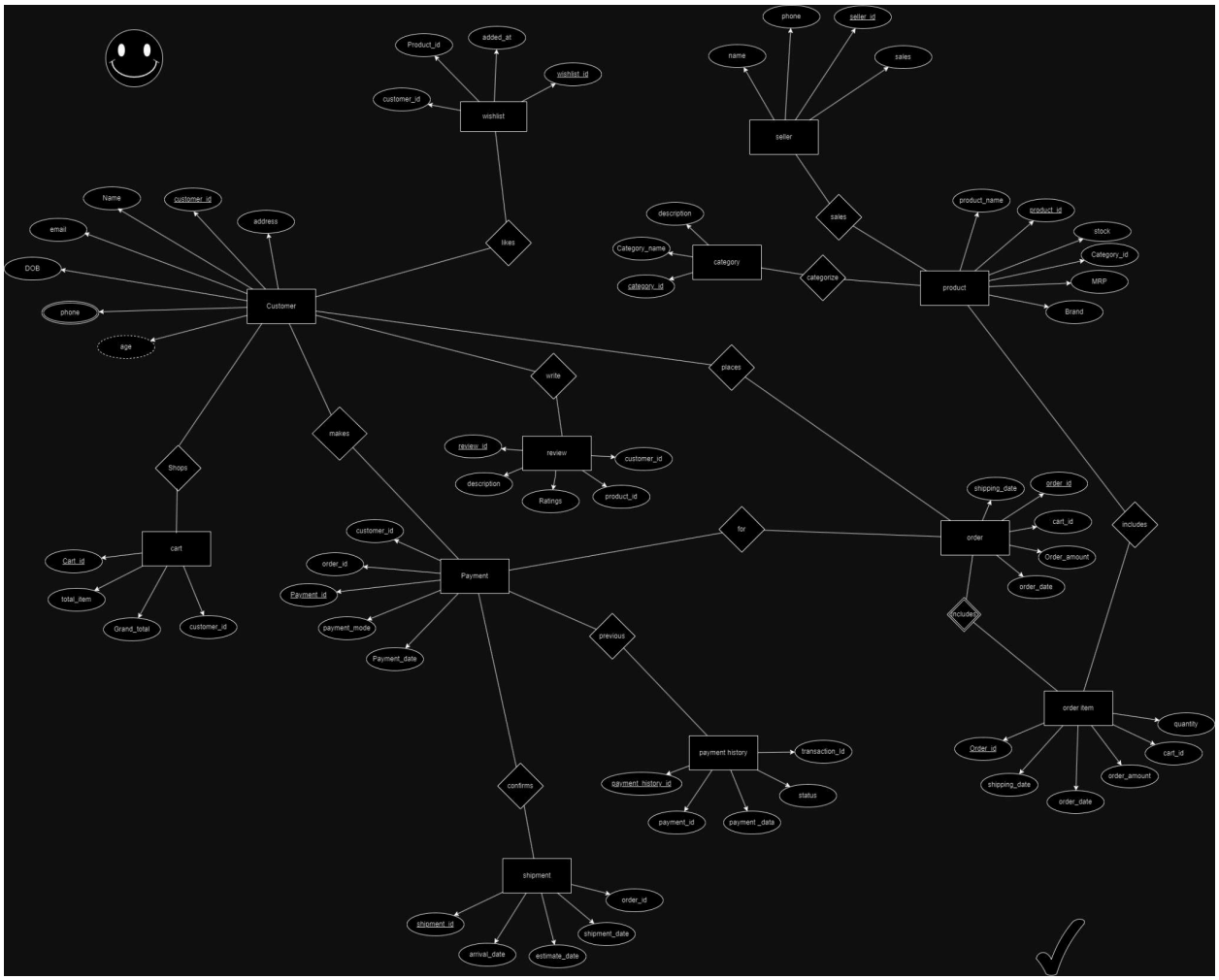
Payment (payment_id, order_id, amount, payment_method, status, created_at, updated_at)

Payment History (payment_history_id, payment_id, transaction_id, payment_date, status, created_at, updated_at)

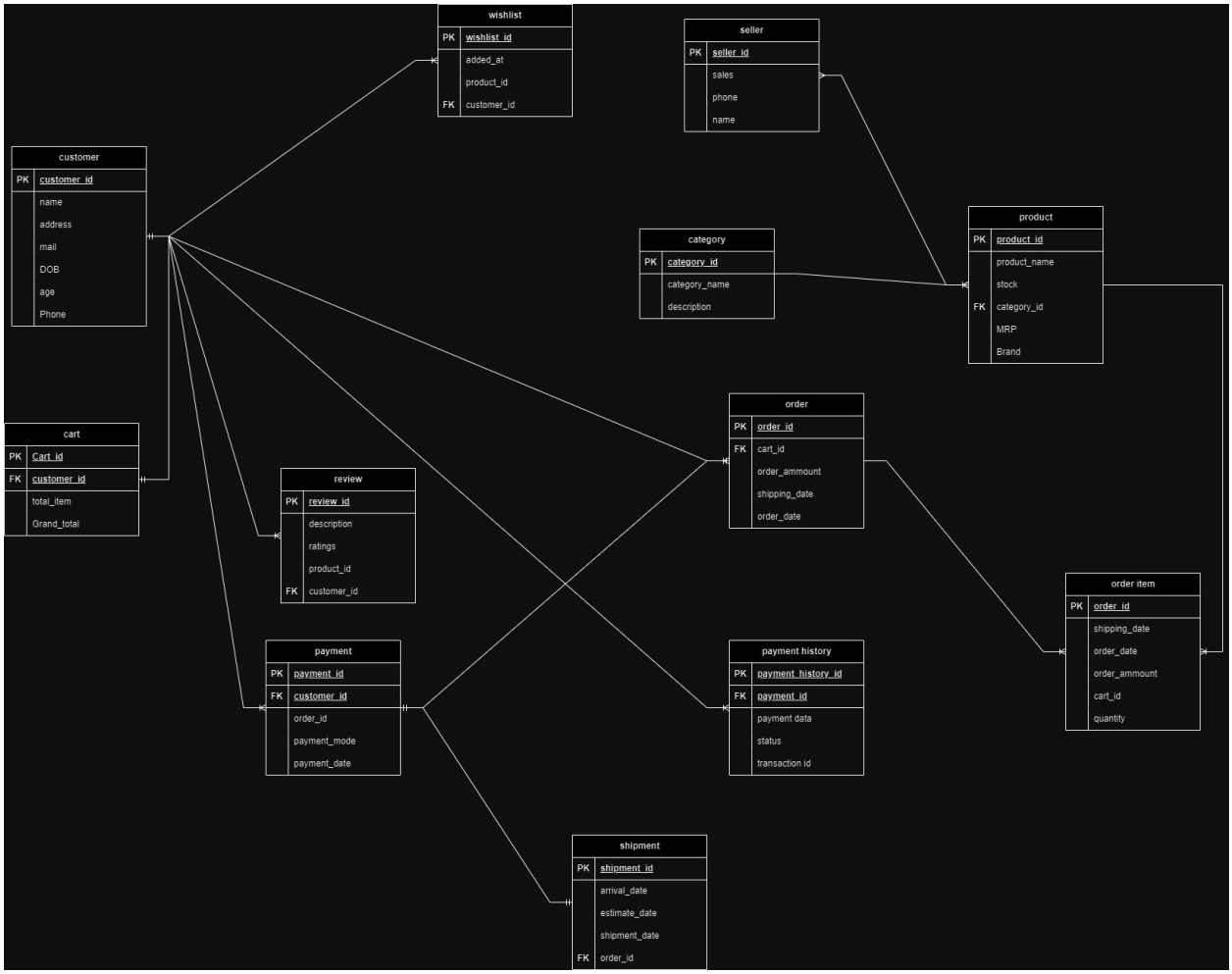
Shipment (shipment_id, order_id, shipment_date, estimated_arrival_date, actual_arrival_date, status, created_at, updated_at)

Wishlist (wishlist_id, customer_id, product_id, added_at)

III. Entity Relationship Diagram



IV. Relational Model



V. Normalization

1. User Table: All columns contain atomic values. No repeating groups. Attributes depend on the primary key (customer_id).
2. Order Table: Similarly, all conditions for 3NF are met.
3. Order Item Table: Same as above.
4. Product Table: All conditions for 3NF are met.
5. Review Table: All conditions for 3NF are met.
6. Cart Table: All conditions for 3NF are met.
7. Category Table: All conditions for 3NF are met.
8. Seller Table: All conditions for 3NF are met.
9. Payment Table: All conditions for 3NF are met.
10. Payment History Table: All conditions for 3NF are met.
11. Shipment Table: All conditions for 3NF are met.
12. Wishlist Table:All conditions for 3NF are met.

VI. SQL Queries

Output of Tables:

Table -1 CUSTOMER

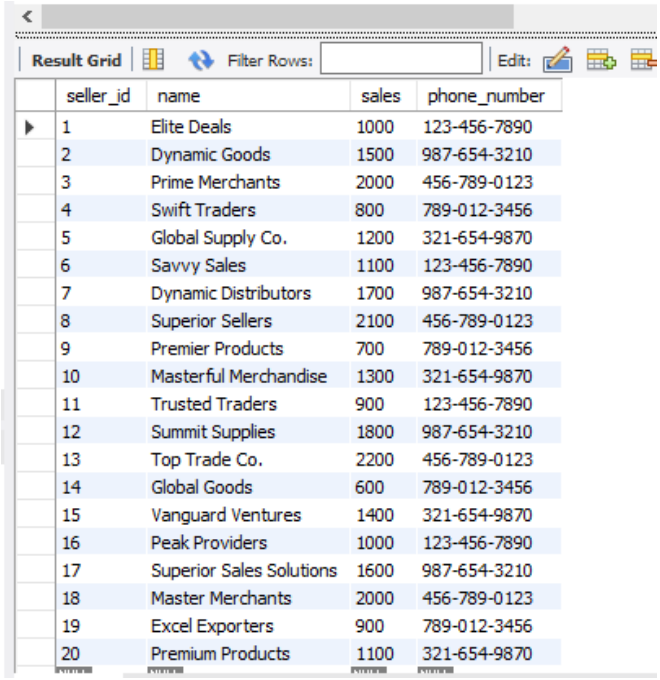
customer_id	name	email	address	phone_number	age
4	Emma Davis	emma@example.com	101 Pine St	789-012-3456	28
5	Michael Wilson	michael@example.com	202 Maple St	321-654-9870	32
6	Sarah Brown	sarah@example.com	303 Cedar St	123-456-7890	27
7	David Miller	david@example.com	404 Oak St	987-654-3210	33
8	Jessica Taylor	jessica@example.com	505 Elm St	456-789-0123	29
9	Chris Harris	chris@example.com	606 Pine St	789-012-3456	31
10	Rachel Clark	rachel@example.com	707 Maple St	321-654-9870	26
11	Matthew Mar...	matthew@example.com	808 Ceda	7890	34
12	Lauren Garcia	lauren@example.com	909 Oak St	987-654-3210	30
13	Kevin Rodriguez	kevin@example.com	1010 Elm St	456-789-0123	28
14	Amanda Hern...	amanda@example.com	1111 Pine St	789-012-3456	32
15	Justin Nelson	justin@example.com	1212 Maple St	321-654-9870	27
16	Stephanie King	stephanie@example....	1313 Cedar St	123-456-7890	29
17	Brandon Adams	brandon@example.com	1414 Oak St	987-654-3210	33
18	Nicole Thomas	nicole@example.com	1515 Elm St	456-789-0123	31
19	Jonathan White	jonathan@example.com	1616 Pine St	789-012-3456	25
20	Melissa Scott	melissa@example.com	1717 Maple St	321-654-9870	28
NULL	NULL	NULL	NULL	NULL	NULL

Table -2 ORDER

Result Grid | Filter Rows: | Edit:

order_id	customer_id	order_date	total_amount	cart_id
1	1	2024-01-01	100.00	1
2	2	2024-01-02	150.00	2
3	3	2024-01-03	200.00	3
4	4	2024-01-04	75.00	4
5	5	2024-01-05	120.00	5
6	6	2024-01-06	90.00	6
7	7	2024-01-07	80.00	7
8	8	2024-01-08	110.00	8
9	9	2024-01-09	70.00	9
10	10	2024-01-10	120.00	10
11	11	2024-01-11	85.00	11
12	12	2024-01-12	95.00	12
13	13	2024-01-13	135.00	13
14	14	2024-01-14	105.00	14
15	15	2024-01-15	125.00	15
16	16	2024-01-16	100.00	16
17	17	2024-01-17	145.00	17
18	18	2024-01-18	95.00	18
19	19	2024-01-19	130.00	19
20	20	2024-01-20	110.00	20

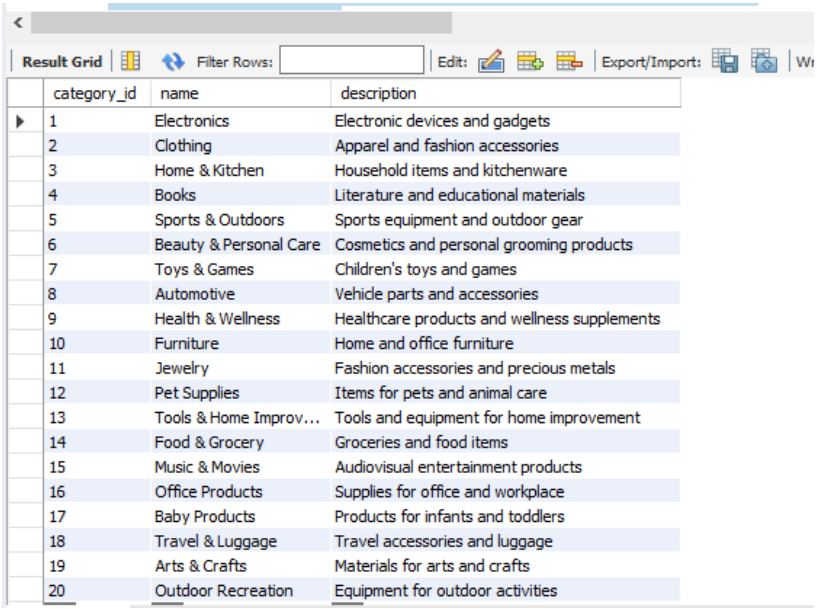
Table -3 SELLER



Result Grid | Filter Rows: | Edit:

seller_id	name	sales	phone_number
1	Elite Deals	1000	123-456-7890
2	Dynamic Goods	1500	987-654-3210
3	Prime Merchants	2000	456-789-0123
4	Swift Traders	800	789-012-3456
5	Global Supply Co.	1200	321-654-9870
6	Savvy Sales	1100	123-456-7890
7	Dynamic Distributors	1700	987-654-3210
8	Superior Sellers	2100	456-789-0123
9	Premier Products	700	789-012-3456
10	Masterful Merchandise	1300	321-654-9870
11	Trusted Traders	900	123-456-7890
12	Summit Supplies	1800	987-654-3210
13	Top Trade Co.	2200	456-789-0123
14	Global Goods	600	789-012-3456
15	Vanguard Ventures	1400	321-654-9870
16	Peak Providers	1000	123-456-7890
17	Superior Sales Solutions	1600	987-654-3210
18	Master Merchants	2000	456-789-0123
19	Excel Exporters	900	789-012-3456
20	Premium Products	1100	321-654-9870

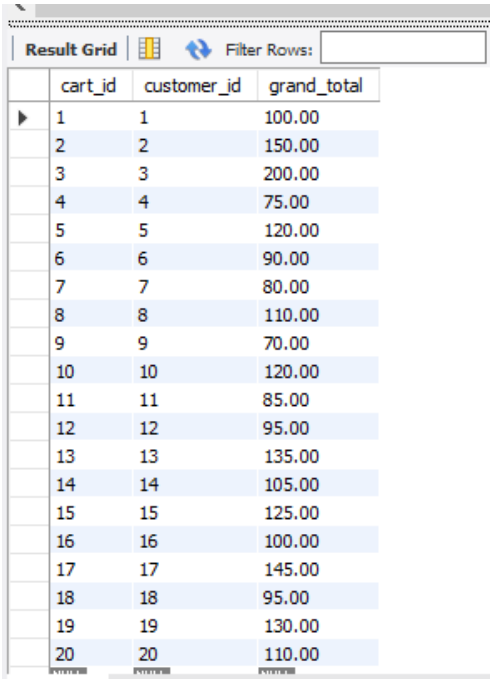
Table -4 CATEGORY



Result Grid | Filter Rows: | Edit: | Export/Import:

category_id	name	description
1	Electronics	Electronic devices and gadgets
2	Clothing	Apparel and fashion accessories
3	Home & Kitchen	Household items and kitchenware
4	Books	Literature and educational materials
5	Sports & Outdoors	Sports equipment and outdoor gear
6	Beauty & Personal Care	Cosmetics and personal grooming products
7	Toys & Games	Children's toys and games
8	Automotive	Vehicle parts and accessories
9	Health & Wellness	Healthcare products and wellness supplements
10	Furniture	Home and office furniture
11	Jewelry	Fashion accessories and precious metals
12	Pet Supplies	Items for pets and animal care
13	Tools & Home Improv...	Tools and equipment for home improvement
14	Food & Grocery	Groceries and food items
15	Music & Movies	Audiovisual entertainment products
16	Office Products	Supplies for office and workplace
17	Baby Products	Products for infants and toddlers
18	Travel & Luggage	Travel accessories and luggage
19	Arts & Crafts	Materials for arts and crafts
20	Outdoor Recreation	Equipment for outdoor activities

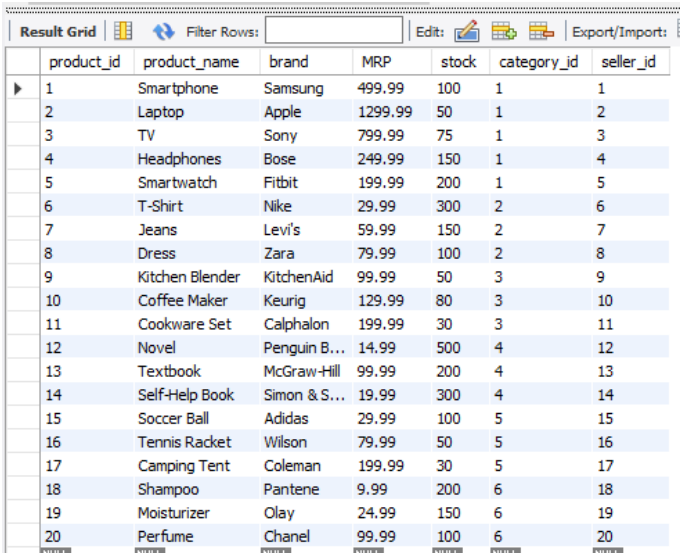
Table - 5 CART



Result Grid | Filter Rows:

cart_id	customer_id	grand_total
1	1	100.00
2	2	150.00
3	3	200.00
4	4	75.00
5	5	120.00
6	6	90.00
7	7	80.00
8	8	110.00
9	9	70.00
10	10	120.00
11	11	85.00
12	12	95.00
13	13	135.00
14	14	105.00
15	15	125.00
16	16	100.00
17	17	145.00
18	18	95.00
19	19	130.00
20	20	110.00

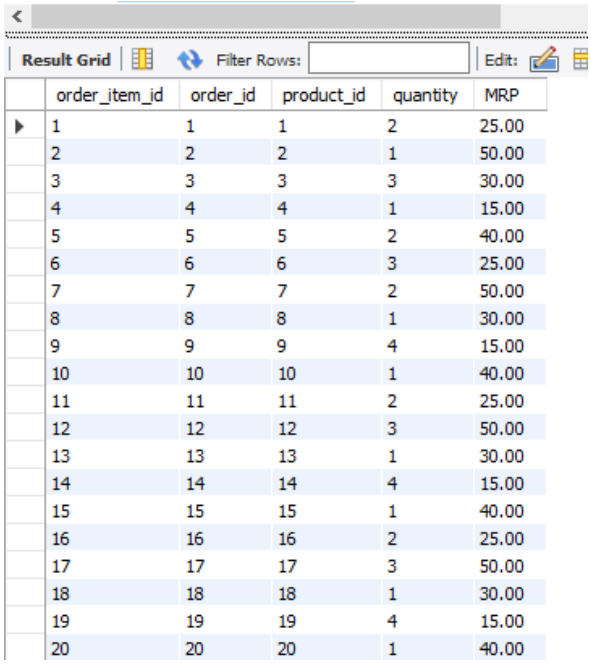
Table - 6 PRODUCT



Result Grid | Filter Rows: | Edit: | Export/Import:

product_id	product_name	brand	MRP	stock	category_id	seller_id
1	Smartphone	Samsung	499.99	100	1	1
2	Laptop	Apple	1299.99	50	1	2
3	TV	Sony	799.99	75	1	3
4	Headphones	Bose	249.99	150	1	4
5	Smartwatch	Fitbit	199.99	200	1	5
6	T-Shirt	Nike	29.99	300	2	6
7	Jeans	Levi's	59.99	150	2	7
8	Dress	Zara	79.99	100	2	8
9	Kitchen Blender	KitchenAid	99.99	50	3	9
10	Coffee Maker	Keurig	129.99	80	3	10
11	Cookware Set	Calphalon	199.99	30	3	11
12	Novel	Penguin B...	14.99	500	4	12
13	Textbook	McGraw-Hill	99.99	200	4	13
14	Self-Help Book	Simon & S...	19.99	300	4	14
15	Soccer Ball	Adidas	29.99	100	5	15
16	Tennis Racket	Wilson	79.99	50	5	16
17	Camping Tent	Coleman	199.99	30	5	17
18	Shampoo	Pantene	9.99	200	6	18
19	Moisturizer	Olay	24.99	150	6	19
20	Perfume	Chanel	99.99	100	6	20

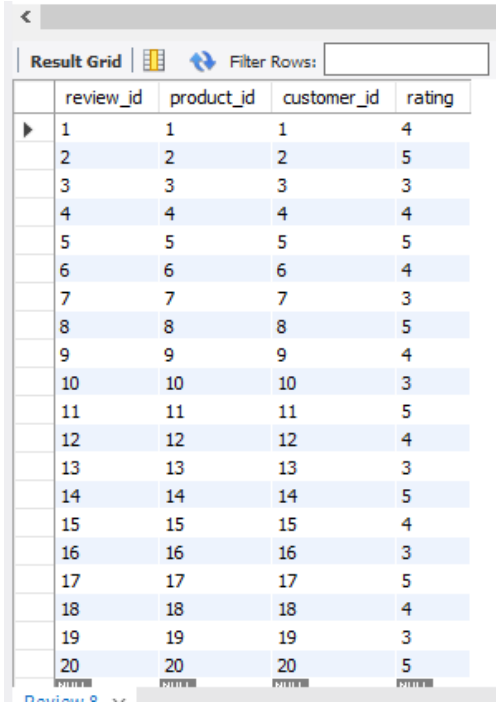
Table - 7 OrderItem



Result Grid | Filter Rows: | Edit:

order_item_id	order_id	product_id	quantity	MRP
1	1	1	2	25.00
2	2	2	1	50.00
3	3	3	3	30.00
4	4	4	1	15.00
5	5	5	2	40.00
6	6	6	3	25.00
7	7	7	2	50.00
8	8	8	1	30.00
9	9	9	4	15.00
10	10	10	1	40.00
11	11	11	2	25.00
12	12	12	3	50.00
13	13	13	1	30.00
14	14	14	4	15.00
15	15	15	1	40.00
16	16	16	2	25.00
17	17	17	3	50.00
18	18	18	1	30.00
19	19	19	4	15.00
20	20	20	1	40.00

Table - 8 REVIEW



Result Grid | Filter Rows:

review_id	product_id	customer_id	rating
1	1	1	4
2	2	2	5
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	4
7	7	7	3
8	8	8	5
9	9	9	4
10	10	10	3
11	11	11	5
12	12	12	4
13	13	13	3
14	14	14	5
15	15	15	4
16	16	16	3
17	17	17	5
18	18	18	4
19	19	19	3
20	20	20	5

Table - 9 PAYMENT

payment_id	order_id	customer_id	payment_method	payment_date
1	1	1	Credit Card	2024-01-01
2	2	2	PayPal	2024-01-02
3	3	3	Cash on Delivery	2024-01-03
4	4	4	Credit Card	2024-01-04
5	5	5	Cash on Delivery	2024-01-05
6	6	6	Credit Card	2024-01-06
7	7	7	PayPal	2024-01-07
8	8	8	Cash on Delivery	2024-01-08
9	9	9	Credit Card	2024-01-09
10	10	10	Cash on Delivery	2024-01-10
11	11	11	Credit Card	2024-01-11
12	12	12	PayPal	2024-01-12
13	13	13	Cash on Delivery	2024-01-13
14	14	14	Credit Card	2024-01-14
15	15	15	Cash on Delivery	2024-01-15
16	16	16	Credit Card	2024-01-16
17	17	17	PayPal	2024-01-17
18	18	18	Cash on Delivery	2024-01-18
19	19	19	Credit Card	2024-01-19
20	20	20	Cash on Delivery	2024-01-20
NULL	NULL	NULL	NULL	NULL

Table - 10 PAYMENT_HISTORY

payment_history_id	payment_id	transaction_id	payment_date	status
1	1	12345	2024-01-01	Success
2	2	67890	2024-01-02	Success
3	3	23456	2024-01-03	Pending
4	4	78901	2024-01-04	Success
5	5	34567	2024-01-05	Pending
6	6	12345	2024-01-06	Success
7	7	67890	2024-01-07	Success
8	8	23456	2024-01-08	Pending
9	9	78901	2024-01-09	Success
10	10	34567	2024-01-10	Pending
11	11	12345	2024-01-11	Success
12	12	67890	2024-01-12	Success
13	13	23456	2024-01-13	Pending
14	14	78901	2024-01-14	Success
15	15	34567	2024-01-15	Pending
16	16	12345	2024-01-16	Success
17	17	67890	2024-01-17	Success
18	18	23456	2024-01-18	Pending
19	19	78901	2024-01-19	Success
20	20	34567	2024-01-20	Pending
NULL	NULL	NULL	NULL	NULL

Table - 11 SHIPMENT

Result Grid | Filter Rows: | Edit: | Export/Import:

shipment_id	order_id	shipment_date	estimated_arrival_date	actual_arrival_date
2	2	2024-01-02	2024-01-04	2024-01-03
3	3	2024-01-03	2024-01-05	2024-01-06
4	4	2024-01-04	2024-01-06	2024-01-05
5	5	2024-01-05	2024-01-07	2024-01-08
6	6	2024-01-06	2024-01-08	2024-01-07
7	7	2024-01-07	2024-01-09	2024-01-08
8	8	2024-01-08	2024-01-10	2024-01-09
9	9	2024-01-09	2024-01-11	2024-01-10
10	10	2024-01-10	2024-01-12	2024-01-11
11	11	2024-01-11	2024-01-13	2024-01-12
12	12	2024-01-12	2024-01-14	2024-01-13
13	13	2024-01-13	2024-01-15	2024-01-14
14	14	2024-01-14	2024-01-16	2024-01-15
15	15	2024-01-15	2024-01-17	2024-01-16
16	16	2024-01-16	2024-01-18	2024-01-17
17	17	2024-01-17	2024-01-19	2024-01-18
18	18	2024-01-18	2024-01-20	2024-01-19
19	19	2024-01-19	2024-01-21	2024-01-20
20	20	2024-01-20	2024-01-22	2024-01-21
NULL	NULL	NULL	NULL	NULL

Table - 12 WISHLIST

Result Grid | Filter Rows:

wishlist_id	customer_id	product_id
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
NULL	NULL	NULL

SQL Queries:

Simple query-

```
470
471 •  SELECT AVG(MRP) AS average_MRP FROM Product;
472
473
474
475
476
```

Result Grid | Filter Rows: | Export: | Wrap Cell

average_MRP
211.490000

Query 1 | SQL File 3* | SQL File 4*

Limit to 1000 rows

```
459 •  UPDATE Customer
460    SET name = 'John Smith'
461    WHERE customer_id = 1;
462 •  UPDATE Customer
463    SET name = 'David Watson'
464    WHERE customer_id = 2;
465 •  Select * from Customer;
466
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

customer_id	name	email	address	phone_number	age
1	John Smith	john@example.com	123 Main St	123-456-7890	30
2	David Watson	alice@example.com	456 Elm St	987-654-3210	25
3	Bob Johnson	bob@example.com	789 Oak St	456-789-0123	35
4	Emma Davis	emma@example.com	101 Pine St	789-012-3456	28
5	Michael Wilson	michael@example.com	202 Maple St	321-654-9870	32

```
9 •  SELECT * FROM Product ORDER BY stock DESC ;
```

sult Grid | Filter Rows: | Edit: | Export/Import:

product_id	product_name	brand	MRP	stock	category_id	seller_id
12	Novel	Penguin Books	14.99	500	4	12
6	T-Shirt	Nike	29.99	300	2	6
14	Self-Help Book	Simon & Schuster	19.99	300	4	14
5	Smartwatch	Fitbit	199.99	200	1	5
13	Textbook	McGraw-Hill	99.99	200	4	13
18	Shampoo	Pantene	9.99	200	6	18
4	Headphones	Bose	249.99	150	1	4
7	Jeans	Levi's	59.99	150	2	7
19	Moisturizer	Jeans	24.99	150	6	19
1	Smartphone	Samsung	499.99	100	1	1
8	Dress	Zara	79.99	100	2	8
15	Soccer Ball	Adidas	29.99	100	5	15
20	Perfume	Chanel	99.99	100	6	20
10	Coffee Maker	Keurig	129.99	80	3	10
3	TV	Sony	799.99	75	1	3
2	Laptop	Apple	1299.99	50	1	2
9	Kitchen Blender	KitchenAid	99.99	50	3	9
16	Tennis Racket	Wilson	79.99	50	5	16
11	Cookware Set	Calphalon	199.99	30	3	11
17	Camping Tent	Coleman	199.99	30	5	17

```sql
SELECT * FROM Orders WHERE total_amount > 100.00;
```

Result Grid | Filter Rows: | Edit: | Exp

order_id	customer_id	order_date	total_amount	cart_id
2	2	2024-01-02	150.00	2
3	3	2024-01-03	200.00	3
5	5	2024-01-05	120.00	5
8	8	2024-01-08	110.00	8
10	10	2024-01-10	120.00	10
13	13	2024-01-13	135.00	13
14	14	2024-01-14	105.00	14
15	15	2024-01-15	125.00	15
17	17	2024-01-17	145.00	17
19	19	2024-01-19	130.00	19
20	20	2024-01-20	110.00	20
NULL	NULL	NULL	NULL	NULL

```sql
/*QUERY 1
CUSTOMER TO FIND THE PRODUCT WITH HIGHEST RATING FOR GIVEN CATEGORY*/
SELECT p.product_name, AVG(r.rating) AS avg_rating
FROM Product p
JOIN Review r ON p.product_id = r.product_id
JOIN Category c ON p.category_id = c.category_id
WHERE c.name = 'Electronics'
GROUP BY p.product_name
ORDER BY avg_rating DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_name	avg_rating
Smartwatch	5.0000

```sql
/* QUERY 2
Customers to filter out the products according to their brand and price.( products of Samsung having MRP less than 300)*/
SELECT *
FROM Product
WHERE brand = 'Samsung' AND MRP < 1000.00;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_id	product_name	brand	MRP	stock	category_id	seller_id
1	Smartphone	Samsung	499.99	100	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
/*QUERY 3
 Customers to find the best seller of a particular product.*/
 SELECT s.name AS best_seller
FROM Seller s
JOIN Product p ON s.seller_id = p.seller_id
WHERE p.product_id = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

best_seller
Elite Deals

```sql
/*QUERY 5 List the category of product which has been sold the highest on a particular day ( 01-01-2024). */
SELECT c.name, SUM(oi.quantity) AS total_sales
FROM Orders o
JOIN OrderItem oi ON o.order_id = oi.order_id
JOIN Product p ON oi.product_id = p.product_id
JOIN Category c ON p.category_id = c.category_id
WHERE o.order_date = '2024-01-01'
GROUP BY c.name
ORDER BY total_sales DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	total_sales
Electronics	2

```sql
/* QUERY 6
Customers to compare the products based on their ratings and reviews. .*/
SELECT p.product_name, AVG(r.rating) AS avg_rating, COUNT(r.review_id) AS total_reviews
FROM Product p
JOIN Review r ON p.product_id = r.product_id
GROUP BY p.product_name
ORDER BY avg_rating DESC, total_reviews DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	avg_rating	total_reviews
Laptop	5.0000	1
Smartwatch	5.0000	1
Dress	5.0000	1
Cookware Set	5.0000	1
Self-Help Book	5.0000	1
Camping Tent	5.0000	1
Perfume	5.0000	1
Smartphone	4.0000	1
Headphones	4.0000	1
T-Shirt	4.0000	1
Kitchen Blender	4.0000	1
Novel	4.0000	1
Soccer Ball	4.0000	1
Shampoo	4.0000	1
TV	3.0000	1
Jeans	3.0000	1
Coffee Maker	3.0000	1
Textbook	3.0000	1
Tennis Racket	3.0000	1
Moisturizer	3.0000	1

```sql
/* QUERY 7 List the sellers with total sales greater than $1000*/
SELECT *
FROM Seller
WHERE seller_id IN (
    SELECT seller_id
    FROM Product
    GROUP BY seller_id
    HAVING SUM(stock * MRP) > 30000
);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

seller_id	name	sales	phone_number
1	Elite Deals	1000	123-456-7890
2	Dynamic Goods	1500	987-654-3210
3	Prime Merchants	2000	456-789-0123
4	Swift Traders	800	789-012-3456
5	Global Supply Co.	1200	321-654-9870
NULL	NULL	NULL	NULL

```sql
/* QUERY 8
   List the seller who has the highest stock of a particular product. */
   SELECT s.name AS seller_name, p.stock
FROM Seller s
JOIN Product p ON s.seller_id = p.seller_id
WHERE p.product_id = 1
ORDER BY p.stock DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

seller_name	stock
Elite Deals	100

```sql
/* QUERY 9 Display the orders made by customers aged 30 or above*/
SELECT *
FROM Orders
WHERE customer_id IN (
    SELECT customer_id
    FROM Customer
    WHERE age >= 30
);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

order_id	customer_id	order_date	total_amount	cart_id
1	1	2024-01-01	100.00	1
3	3	2024-01-03	200.00	3
5	5	2024-01-05	120.00	5
7	7	2024-01-07	80.00	7
9	9	2024-01-09	70.00	9
11	11	2024-01-11	85.00	11
12	12	2024-01-12	95.00	12
14	14	2024-01-14	105.00	14
17	17	2024-01-17	145.00	17
18	18	2024-01-18	95.00	18
NULL	NULL	NULL	NULL	NULL

```sql
/*  QUERY 10
Find customers who have made payments using PayPal */
SELECT customer_id, name
FROM Customer
WHERE customer_id IN (
    SELECT DISTINCT customer_id
    FROM Payment
    WHERE payment_method = 'PayPal'
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	name
2	Alice Smith
7	David Miller
12	Lauren Garcia
17	Brandon Adams
NULL	NULL

```sql
/* QUERY 11 List the products with ratings greater than 4.5*/
SELECT *
FROM Product
WHERE product_id IN (
    SELECT product_id
    FROM Review
    WHERE rating > 4.5
);
```

ult Grid | Filter Rows: | Edit: | Export/Import:

product_id	product_name	brand	MRP	stock	category_id	seller_id
2	Laptop	Apple	1299.99	50	1	2
5	Smartwatch	Fitbit	199.99	200	1	5
8	Dress	Zara	79.99	100	2	8
11	Cookware Set	Calphalon	199.99	30	3	11
14	Self-Help Book	Simon & Schuster	19.99	300	4	14
17	Camping Tent	Coleman	199.99	30	5	17
20	Perfume	Chanel	99.99	100	6	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
/* QUERY 12
Find orders where the shipment was delayed beyond the estimated arrival date */
SELECT order_id, order_date
FROM Orders
WHERE order_id IN (
    SELECT order_id
    FROM Shipment
    WHERE actual_arrival_date > estimated_arrival_date
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

order_id	order_date
3	2024-01-03
5	2024-01-05
NULL	NULL

```sql
/*QUERY 13 Find the customer who made the highest total purchase amount*/
SELECT name
FROM Customer
WHERE customer_id = (
    SELECT customer_id
    FROM Orders
    GROUP BY customer_id
    ORDER BY SUM(total_amount) DESC
    LIMIT 1
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name
Bob Johnson

```sql
/* QUERY 14
Find products that have stock less then 100 */
SELECT product_id, product_name
FROM Product
WHERE stock < 100;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_id	product_name
2	Laptop
3	TV
9	Kitchen Blender
10	Coffee Maker
11	Cookware Set
16	Tennis Racket
17	Camping Tent
21	sdeer
NULL	NULL

```sql
/* QUERY 15
Find the total revenue generated in each month of the year */
SELECT EXTRACT(MONTH FROM order_date) AS month, SUM(total_amount) AS monthly_revenue
FROM Orders
GROUP BY EXTRACT(MONTH FROM order_date)
ORDER BY month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month	monthly_revenue
1	2240.00

```sql
/*  QUERY 16
Find the average age of customers who have made a purchase */
SELECT AVG(age) AS average_age
FROM Customer
WHERE customer_id IN (
    SELECT DISTINCT customer_id
    FROM Orders
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

average_age
29.6500

```sql
/* QUERY 17
Find the average rating given by customers for each product category: */
SELECT c.name AS category_name, AVG(r.rating) AS average_rating
FROM Category c
LEFT JOIN Product p ON c.category_id = p.category_id
LEFT JOIN Review r ON p.product_id = r.product_id
GROUP BY c.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category_name	average_rating
Electronics	4.2000
Clothing	4.0000
Home & Kitchen	4.0000
Books	4.0000
Sports & Outdoors	4.0000
Beauty & Personal Care	4.0000
Toys & Games	NULL
Automotive	NULL
Health & Wellness	NULL
Furniture	NULL
Jewelry	NULL
Pet Supplies	NULL
Tools & Home Improv...	NULL
Food & Grocery	NULL
Music & Movies	NULL
Office Products	NULL
Baby Products	NULL
Travel & Luggage	NULL
Arts & Crafts	NULL
Outdoor Recreation	NULL

```sql
/*  QUERY 18
Find the top 3 best-selling products*/
SELECT p.product_id, p.product_name, SUM(oi.quantity) AS total_sold
FROM Product p
JOIN OrderItem oi ON p.product_id = oi.product_id
GROUP BY p.product_id, p.product_name
ORDER BY total_sold DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_id	product_name	total_sold
14	Self-Help Book	4
19	Moisturizer	4
9	Kitchen Blender	4

```sql
/* QUERY 19
Find the products that have not been sold yet */
SELECT p.product_id, p.product_name
FROM Product p
LEFT JOIN OrderItem oi ON p.product_id = oi.product_id
WHERE oi.order_id IS NULL;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

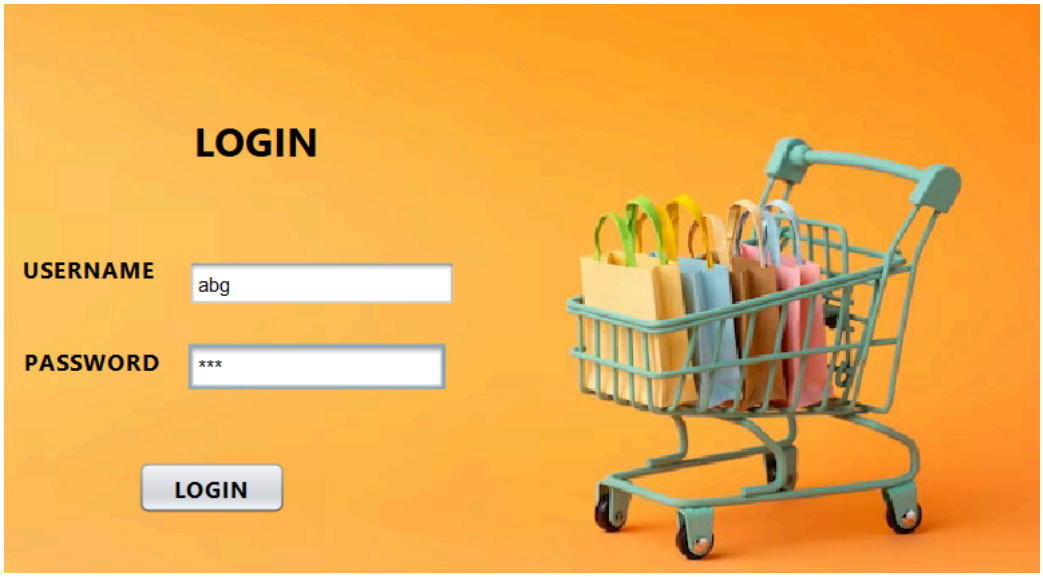
product_id	product_name
21	sdeer

```sql
/* QUERY 20
Find the customers who have made purchases on weekdays (Monday to Friday) */
SELECT DISTINCT c.customer_id, c.name AS customer_name
FROM Customer c
JOIN Orders o ON c.customer_id = o.customer_id
WHERE DAYOFWEEK(o.order_date) BETWEEN 2 AND 6;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name
1	John Doe
2	Alice Smith
3	Bob Johnson
4	Emma Davis
5	Michael Wilson
8	Jessica Taylor
9	Chris Harris
10	Rachel Clark
11	Matthew Martinez
12	Lauren Garcia
15	Justin Nelson
16	Stephanie King
17	Brandon Adams
18	Nicole Thomas
19	Jonathan White

VI. Project demonstration



VII. Self -Learning beyond classroom:

1.Database Design: Learned the intricacies of designing a relational database schema, including entity identification, attribute definition, and table normalization to minimize redundancy and ensure data integrity.

2.Entity Relationship Modeling: Gained expertise in defining relationships between entities, such as one-to-one, one-to-many, and many-to-many relationships, to accurately represent the data structure and facilitate efficient data retrieval.
3.SQL Proficiency: Improved SQL skills through writing complex queries to retrieve, insert, update, and delete data, enabling effective data manipulation and management within the database.
4.Query Optimization: Mastered techniques for optimizing SQL queries to enhance database performance, including indexing, query restructuring, and use of efficient join strategies to minimize query execution time.
5.Problem Solving: Faced challenges in resolving database-related issues, which honed problem-solving abilities and fostered a systematic approach to debugging and troubleshooting.
6.Understanding Data Integrity: Acquired a deeper understanding of maintaining data integrity through constraints, such as primary keys, foreign keys, and unique constraints, to enforce data consistency and accuracy.
7.Collaboration Skills: Collaborated effectively with team members to discuss database design decisions, resolve conflicts, and ensure alignment with project requirements, thereby enhancing teamwork and communication skills.

Baaki Maza aa gaya
Learned a lot and it also helped in understanding the concept very clearly and those concept those were a bit blurred got crystal cleared -Yash Patil

VIII. Learning from the Project

1.SQL Queries Mastery: You've acquired proficiency in writing SQL queries, which is essential for interacting with databases. This skill enables you to retrieve, insert, update, and delete data effectively, catering to various requirements of database management.

2.Understanding Entity-Relationship Model: Through the project, you've grasped the concept of the entity-relationship model, which forms the basis of designing a database schema. Understanding entities, attributes, relationships, and cardinalities is crucial for creating a well-structured and normalized database.

3.Database Design Principles: By designing and implementing the database schema for the ecommerce system, you've learned about database normalization, data integrity, and other design principles. This knowledge is vital for ensuring efficient storage, retrieval, and management of data.

4.Integration of Java with DBMS: You've successfully connected Java applications with a DBMS (MySQL) using NetBeans IDE. This integration allows you to develop dynamic and interactive applications that interact seamlessly with databases, enabling functionalities like data retrieval, manipulation, and storage.

5.Hands-on Learning: Through practical application and hands-on experience, you've reinforced your understanding of database concepts and programming skills. This hands-on approach has provided you with real-world experience, enhancing your problem-solving abilities and critical thinking skills.

6.Continuous Learning: As you continue to explore and work on projects involving databases and Java applications, you'll encounter new challenges and opportunities for learning. Embrace these opportunities for growth and continually expand your knowledge and skills in database management and application development.

IX. Challenges Faced

Designing an efficient and scalable database schema that accurately represents the relationships between different entities (e.g., customers, products, orders) complex. Deciding on the appropriate data types, keys, and constraints requires careful consideration.

X. Conclusion

Designing and implementing the database schema for the ecommerce system has provided me with a comprehensive understanding of database normalization, relationships, and data modeling principles. Through practical application of programming skills, I strengthened my proficiency in SQL for database management and learned to develop user-friendly interfaces. The project also underscored the importance of data security and integrity, leading me to implement robust security measures. Additionally, I honed my problem-solving abilities and learned the value of effective teamwork and collaboration. Overall, this project has equipped me with essential skills and knowledge that will undoubtedly benefit me in future academic and professional pursuits.