E-COMMERCE

Database Design and Implementation Report

Author: JOSEPH KAYIJUKA

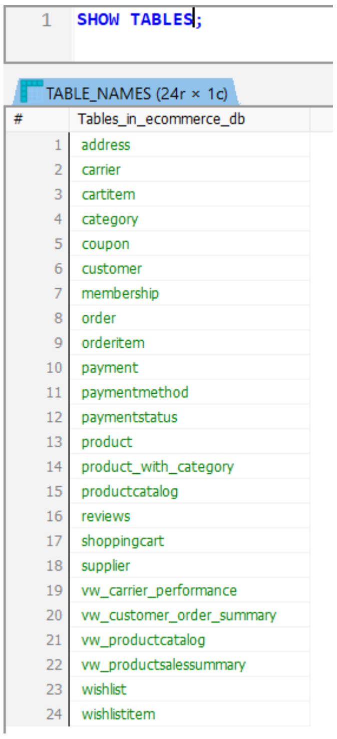
Date: 06-08-2025

Subject: Ecommerce Database

1. LOGICAL AND RELATIONAL DATABASE MODEL

1.1 Ecommerce Data Management

The e-commerce database consists of multiple entities designed to handle Customer Management ,Product Management ,Order Management ,Payment Management , Payment System , Shopping Experience and Marketing & Loyalty .



```
SHOW TABLES;
```

TABLE_NAMES (24r × 1c)

#	Tables_in_ecommerce_db
1	address
2	carrier
3	cartitem
4	category
5	coupon
6	customer
7	membership
8	order
9	orderitem
10	payment
11	paymentmethod
12	paymentstatus
13	product
14	product_with_category
15	productcatalog
16	reviews
17	shoppingcart
18	supplier
19	vw_carrier_performance
20	vw_customer_order_summary
21	vw_productcatalog
22	vw_productsalessummary
23	wishlist
24	wishlistitem

1.2 Entity Descriptions

Core Entities

------Customer Management------

1. CUSTOMER

Purpose: **Stores customer information for account management and order processing**

Attributes:

CustomerID (PK), FirstName, LastName, PhoneNumber, Email, Password, DateOfBirth, Gender, CreatedAt, UpdatedAt, MembershipID (FK)

2. MEMBERSHIP

Purpose: **Manages customer loyalty programs and membership levels**

Attributes:

MembershipID (PK), MembershipType, Description, DiscountRate, ShippingBenefit, IsActive

------Product Management------

3. **PRODUCT**

Purpose: **Manages product catalog with inventory and pricing information**

Attributes:

ProductID (PK), ProductName, ProductDesc, StockQuantity, Price, Size, Brand, SKU, ImageURL, CreatedAt, UpdatedAt, IsActive, SupplierID (FK), CategoryID (FK)

4. **CATEGORY**

Purpose: **Organizes products into logical categories**

Attributes:

CategoryID (PK), CategoryName, CreatedAt, CategoryDesc, IsActive

5. **SUPPLIER**

Purpose: **Manages supplier information for product sourcing**

Attributes:

SupplierID (PK), CompanyName, ContactName, supplier_Address, Phone, Email, CreatedAt, IsActive

------Order Management------

6. **ORDER**

Purpose: **Tracks customer orders and processing status**

Attributes:

OrderID (PK), OrderNumber, OrderDate, RequiredDate, TotalAmount, OrderStatus, PaymentMethod, PaymentStatus, CreatedAt, UpdatedAt, CustomerID (FK), CouponID (FK), Currency, ShippingCost, ShippedDate, DeliveredDate, ShippingMethod, CarrierID (FK)

7. **ORDERITEM**

Purpose: **Stores individual items within each order**

Attributes:

OrderItemID (PK), OrderNumber, Quantity, UnitPrice, TotalPrice, ShipDate, Status, VariantInfo, ProductID (FK), OrderID (FK)

8. **ADDRESS**

Purpose: **Stores shipping/billing addresses for orders**

Attributes:

AddressID (PK), FullName, Street, City, State, PostalCode, Country, Phone, AddressType, CustomerID (FK), OrderID (FK)

------Payment System------

9. **PAYMENT**

Purpose: **Records payment transactions**

Attributes:PaymentID (PK), AmountPaid, PaymentDate, OrderID (FK), PaymentStatusID (FK),PaymentMethodID (FK)

10. PAYMENTMETHOD

Purpose: Defines available payment methods

Attributes: PaymentMethodID (PK), Name

11. PAYMENTSTATUS

Purpose: Tracks payment processing states

Attributes: PaymentStatusID (PK), Name

------Shopping Experience------

12. SHOPPINGCART

Purpose: Manages temporary cart items before checkout

Attributes:

CartID (PK), OrderStatus, UpdatedAt, CreatedAt, TotalPrice, CustomerID (FK)

13. CARTITEM

Purpose: Stores products in shopping carts

Attributes:

CartItemID (PK), Quantity, UnitPrice, AddedAt, UpdatedAt, CartID (FK), ProductID (FK)

14. WISHLIST

Purpose: Manages customer wishlists

Attributes:

WishlistID (PK), Name, Description, CreatedAt, UpdatedAt, CustomerID (FK)

15. WISHLISTITEM

Purpose: Stores products in wishlists

Attributes:

WishlistItemID (PK), AddedAt, Notes, WishlistID (FK), ProductID (FK)

------Marketing & Loyalty------

16. COUPON

Purpose: Manages discount coupons and promotions

Attributes:

CouponID (PK), Code, Type, StartDate, EndDate, IsActive,

UsageCount, MembershipID (FK)

17. CARRIER

Purpose: Stores shipping carrier information

Attributes:

CarrierID (PK), CarrierName, ContactInfo, Website

Customer Feedback

18. **REVIEWS**

***Purpose*: Stores product ratings and reviews**

***Attributes*:**

Rating, Comment, CreatedAt, UpdatedAt, ProductID1 (FK), CustomerID1 (FK) [Composite PK]

1.2 Entity Relationships

1.2.1 Primary Relationships

CUSTOMER ⇔ ORDER (1:N)

- One customer can place multiple orders
- Each order belongs to exactly one customer
- *Foreign Key*: ORDER.CustomerID references CUSTOMER.CustomerID
- *Constraint*: ON DELETE RESTRICT (cannot delete customer with existing orders)

ORDER ⇔ ORDERITEM (1:N)

- One order can contain multiple items
- Each order item belongs to exactly one order
- *Foreign Key*: ORDERITEM.OrderID references ORDER.OrderID
- *Constraint*: ON DELETE CASCADE (deleting order removes all items)

PRODUCT ⇔ ORDERITEM (1:N)

- One product can appear in multiple order items
- Each order item references exactly one product
- *Foreign Key*: ORDERITEM.ProductID references PRODUCT.ProductID
- *Constraint*: ON DELETE RESTRICT (cannot delete product with existing orders)

CATEGORY ⇔ PRODUCT (1:N)

- One category can contain multiple products
- Each product belongs to exactly one category
- *Foreign Key*: PRODUCT.CategoryID references CATEGORY.CategoryID
- *Constraint*: ON DELETE RESTRICT (cannot delete category with existing products)

SUPPLIER ⇔ PRODUCT (1:N)

- One supplier can supply multiple products
- Each product has exactly one supplier
- *Foreign Key*: PRODUCT.SupplierID references SUPPLIER.SupplierID

- *Constraint*: ON DELETE RESTRICT (cannot delete supplier with existing products)

1.2.2 Customer-Centric Relationships

CUSTOMER ⇔ ADDRESS (1:N)

- One customer can have multiple addresses
- Each address belongs to exactly one customer
- *Foreign Key*: ADDRESS.CustomerID references CUSTOMER.CustomerID
- *Constraint*: ON DELETE CASCADE (deleting customer removes addresses)

CUSTOMER ⇔ SHOPPINGCART (1:1)

- One customer has one active shopping cart
- Each shopping cart belongs to exactly one customer
- *Foreign Key*: SHOPPINGCART.CustomerID references CUSTOMER.CustomerID
- *Constraint*: ON DELETE CASCADE (deleting customer removes cart)

CUSTOMER ⇔ WISHLIST (1:N)

- One customer can have multiple wishlists
- Each wishlist belongs to exactly one customer
- *Foreign Key*: WISHLIST.CustomerID references CUSTOMER.CustomerID
- *Constraint*: ON DELETE CASCADE (deleting customer removes wishlists)

CUSTOMER ⇔ MEMBERSHIP (N:1)

- Multiple customers can share the same membership level
- Each customer has exactly one membership
- *Foreign Key*: CUSTOMER.MembershipID references MEMBERSHIP.MembershipID
- *Constraint*: ON DELETE RESTRICT (cannot delete membership with active customers)

1.2.3 Payment & Fulfillment Relationships

ORDER ⇔ PAYMENT (1:N)

- One order can have multiple payment transactions
- Each payment belongs to exactly one order
- *Foreign Key*: PAYMENT.OrderID references ORDER.OrderID
- *Constraint*: ON DELETE CASCADE (deleting order removes payments)

PAYMENTMETHOD ⇔ PAYMENT (1:N)

- One payment method can be used in multiple payments
- Each payment uses exactly one payment method
- *Foreign Key*: PAYMENT.PaymentMethodID references PAYMENTMETHOD.PaymentMethodID

- *Constraint*: ON DELETE RESTRICT (cannot delete payment method in use)

PAYMENTSTATUS ⇔ PAYMENT (1:N)

- One status can apply to multiple payments
- Each payment has exactly one status
- *Foreign Key*: PAYMENT.PaymentStatusID references PAYMENTSTATUS.PaymentStatusID
- *Constraint*: ON DELETE RESTRICT (cannot delete status in use)

CARRIER ⇔ ORDER (1:N)

- One carrier can deliver multiple orders
- Each order uses exactly one carrier
- *Foreign Key*: ORDER.CarrierID references CARRIER.CarrierID
- *Constraint*: ON DELETE RESTRICT (cannot delete carrier with active orders)

1.2.4 Shopping Experience Relationships

SHOPPINGCART ⇔ CARTITEM (1:N)

- One cart can contain multiple items
- Each cart item belongs to exactly one cart
- *Foreign Key*: CARTITEM.CartID references SHOPPINGCART.CartID
- *Constraint*: ON DELETE CASCADE (deleting cart removes items)

WISHLIST ⇔ WISHLISTITEM (1:N)

- One wishlist can contain multiple items
- Each wishlist item belongs to exactly one wishlist
- *Foreign Key*: WISHLISTITEM.WishlistID references WISHLIST.WishlistID
- *Constraint*: ON DELETE CASCADE (deleting wishlist removes items)

PRODUCT ⇔ CARTITEM (1:N)

- One product can be in multiple carts
- Each cart item references exactly one product
- *Foreign Key*: CARTITEM.ProductID references PRODUCT.ProductID
- *Constraint*: ON DELETE CASCADE (deleting product removes from carts)

PRODUCT ⇔ WISHLISTITEM (1:N)

- One product can be in multiple wishlists
- Each wishlist item references exactly one product
- *Foreign Key*: WISHLISTITEM.ProductID references PRODUCT.ProductID
- *Constraint*: ON DELETE CASCADE (deleting product removes from wishlists)

1.2.5 Promotional Relationships

MEMBERSHIP ⇔ COUPON (1:N)

- One membership level can have multiple coupons
- Each coupon belongs to exactly one membership level
- *Foreign Key*: COUPON.MembershipID references MEMBERSHIP.MembershipID
- *Constraint*: ON DELETE CASCADE (deleting membership removes coupons)

COUPON ⇔ ORDER (1:N)

- One coupon can be used in multiple orders
- Each order can use at most one coupon
- *Foreign Key*: ORDER.CouponID references COUPON.CouponID
- *Constraint*: ON DELETE RESTRICT (cannot delete coupon used in orders)

1.2.6 Review Relationships

CUSTOMER ⇔ REVIEWS (1:N)

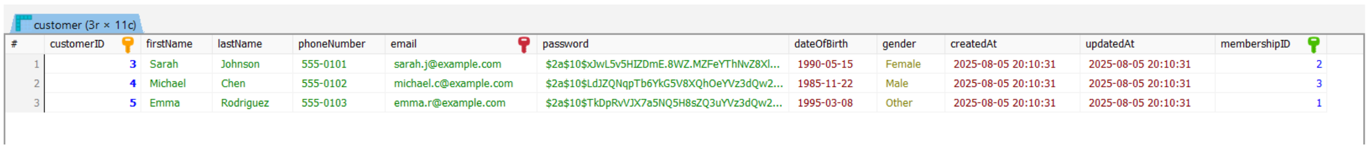
- One customer can write multiple reviews
- Each review is written by exactly one customer
- *Foreign Key*: REVIEWS.CustomerID1 references CUSTOMER.CustomerID
- *Constraint*: ON DELETE CASCADE (deleting customer removes reviews)

PRODUCT ⇔ REVIEWS (1:N)

- One product can have multiple reviews
- Each review is for exactly one product
- *Foreign Key*: REVIEWS.ProductID1 references PRODUCT.ProductID
- *Constraint*: ON DELETE CASCADE (deleting product removes reviews)

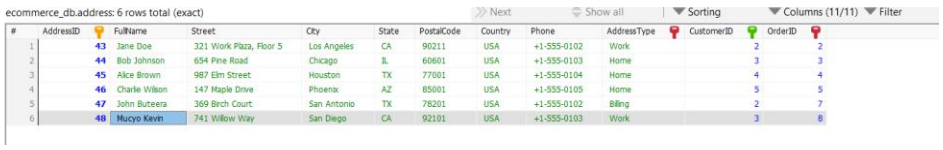
1.3 Entity Instance Examples

1.3.1 Tables

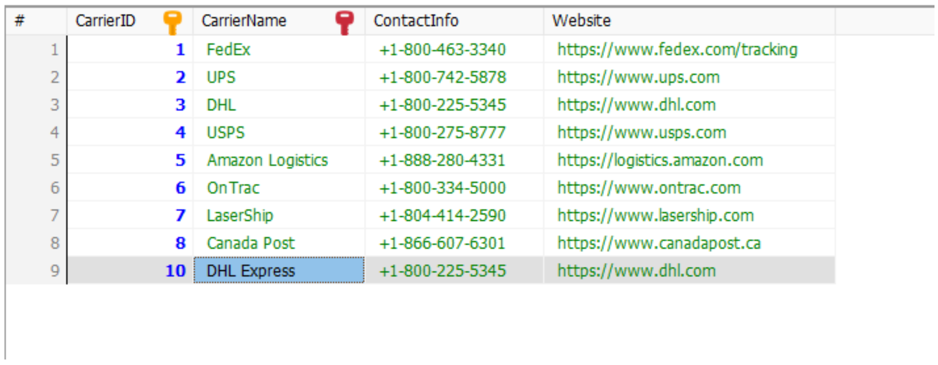


customer (3r × 11c)

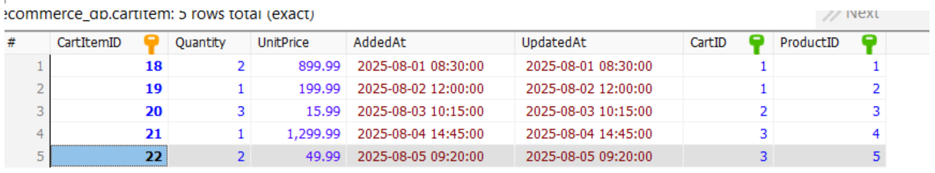
#	customerID	firstName	lastName	phoneNumber	email	password	dateOfBirth	gender	createdAt	updatedAt	membershipID
1	3	Sarah	Johnson	555-0101	sarah.j@example.com	$2a$10$xJwL5v5HIZDmE.8WZ.MZFeYThNvZ8Xl...	1990-05-15	Female	2025-08-05 20:10:31	2025-08-05 20:10:31	2
2	4	Michael	Chen	555-0102	michael.c@example.com	$2a$10$LdJZQNqpTb6YkG5V8XQhOeYVz3dQw2...	1985-11-22	Male	2025-08-05 20:10:31	2025-08-05 20:10:31	3
3	5	Emma	Rodriguez	555-0103	emma.r@example.com	$2a$10$TkDpRvVJX7a5NQ5H8sZQ3uYVz3dQw2...	1995-03-08	Other	2025-08-05 20:10:31	2025-08-05 20:10:31	1

1. 
ecommerce_db.address: 6 rows total (exact)

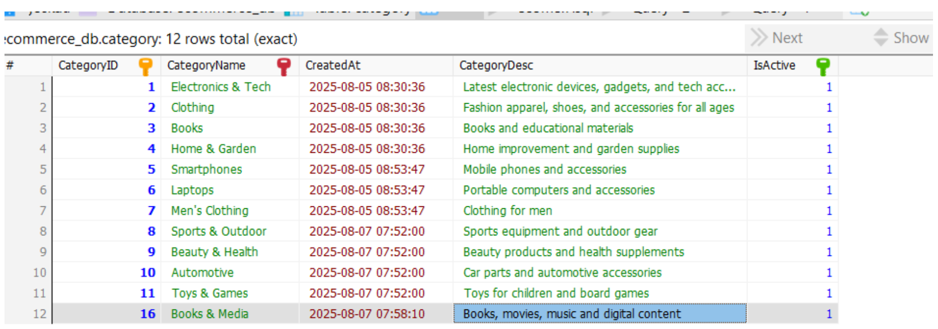
#	AddressID	FullName	Street	City	State	PostalCode	Country	Phone	AddressType	CustomerID	OrderID
1	43	Jane Doe	321 Work Plaza, Floor 5	Los Angeles	CA	90211	USA	+1-555-0102	Work	2	2
2	44	Bob Johnson	654 Pine Road	Chicago	IL	60601	USA	+1-555-0103	Home	3	3
3	45	Alice Brown	987 Elm Street	Houston	TX	77001	USA	+1-555-0104	Home	4	4
4	46	Charlie Wilson	147 Maple Drive	Phoenix	AZ	85001	USA	+1-555-0105	Home	5	5
5	47	John Buteera	369 Birch Court	San Antonio	TX	78201	USA	+1-555-0102	Billing	2	7
6	48	Mucyo Kevin	741 Willow Way	San Diego	CA	92101	USA	+1-555-0103	Work	3	8



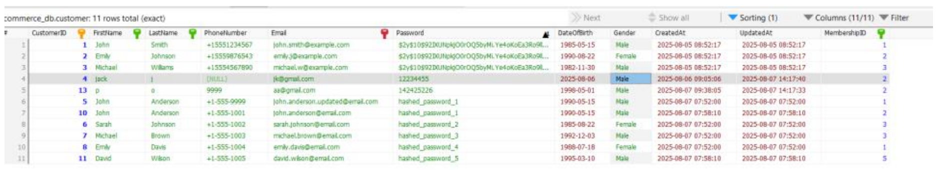
#	CarrierID	CarrierName	ContactInfo	Website
1	1	FedEx	+1-800-463-3340	https://www.fedex.com/tracking
2	2	UPS	+1-800-742-5878	https://www.ups.com
3	3	DHL	+1-800-225-5345	https://www.dhl.com
4	4	USPS	+1-800-275-8777	https://www.usps.com
5	5	Amazon Logistics	+1-888-280-4331	https://logistics.amazon.com
6	6	OnTrac	+1-800-334-5000	https://www.ontrac.com
7	7	LaserShip	+1-804-414-2590	https://www.lasership.com
8	8	Canada Post	+1-866-607-6301	https://www.canadapost.ca
9	10	DHL Express	+1-800-225-5345	https://www.dhl.com


commerce_db.cartitem: 5 rows total (exact)

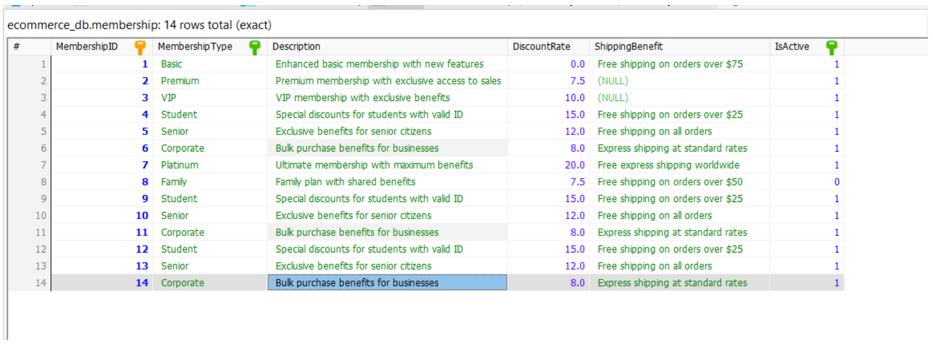
#	CartItemID	Quantity	UnitPrice	AddedAt	UpdatedAt	CartID	ProductID
1	18	2	899.99	2025-08-01 08:30:00	2025-08-01 08:30:00	1	1
2	19	1	199.99	2025-08-02 12:00:00	2025-08-02 12:00:00	1	2
3	20	3	15.99	2025-08-03 10:15:00	2025-08-03 10:15:00	2	3
4	21	1	1,299.99	2025-08-04 14:45:00	2025-08-04 14:45:00	3	4
5	22	2	49.99	2025-08-05 09:20:00	2025-08-05 09:20:00	3	5


commerce_db.category: 12 rows total (exact)

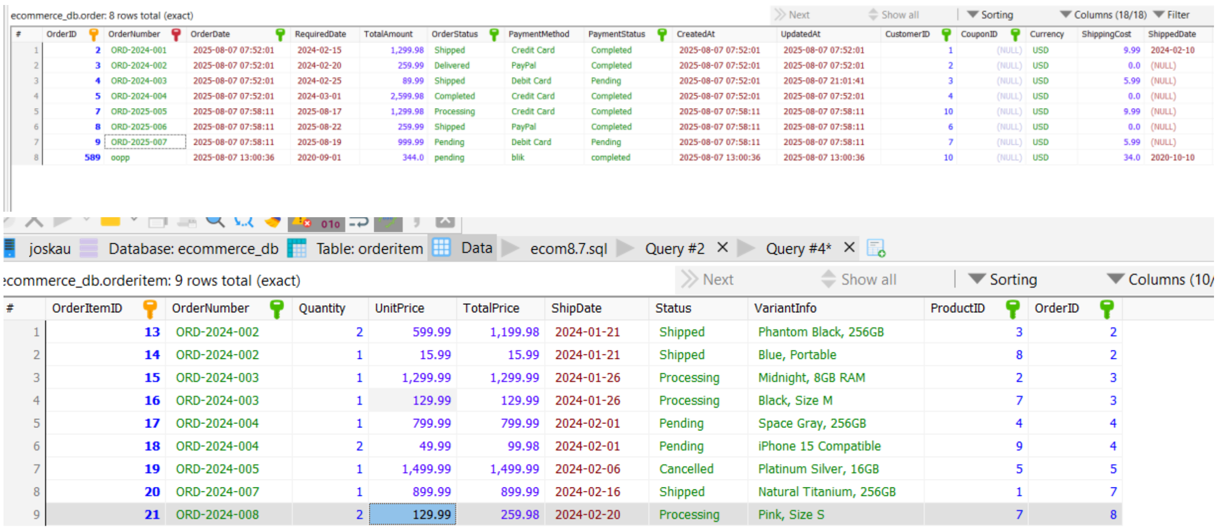
#	CategoryID	CategoryName	CreatedAt	CategoryDesc	IsActive
1	1	Electronics & Tech	2025-08-05 08:30:36	Latest electronic devices, gadgets, and tech acc...	1
2	2	Clothing	2025-08-05 08:30:36	Fashion apparel, shoes, and accessories for all ages	1
3	3	Books	2025-08-05 08:30:36	Books and educational materials	1
4	4	Home & Garden	2025-08-05 08:30:36	Home improvement and garden supplies	1
5	5	Smartphones	2025-08-05 08:53:47	Mobile phones and accessories	1
6	6	Laptops	2025-08-05 08:53:47	Portable computers and accessories	1
7	7	Men's Clothing	2025-08-05 08:53:47	Clothing for men	1
8	8	Sports & Outdoor	2025-08-07 07:52:00	Sports equipment and outdoor gear	1
9	9	Beauty & Health	2025-08-07 07:52:00	Beauty products and health supplements	1
10	10	Automotive	2025-08-07 07:52:00	Car parts and automotive accessories	1
11	11	Toys & Games	2025-08-07 07:52:00	Toys for children and board games	1
12	16	Books & Media	2025-08-07 07:58:10	Books, movies, music and digital content	1


commerce_db.customer: 11 rows total (exact)

#	CustomerID	FirstName	LastName	PhoneNumber	Email	Password	DateOfBirth	Gender	CreatedAt	UpdatedAt	MembershipID
1	1	John	Smith	+15551234567	john.smith@example.com	$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9l...	1985-05-15	Male	2025-08-05 08:52:17	2025-08-05 08:52:17	1
2	2	Emily	Johnson	+15559876543	emily.j@example.com	$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9l...	1990-08-22	Female	2025-08-05 08:52:17	2025-08-05 08:52:17	2
3	3	Michael	Williams	+15554567890	michael.w@example.com	$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9l...	1982-11-30	Male	2025-08-05 08:52:17	2025-08-05 08:52:17	3
4	4	jack	j	(NULL)	jk@gmail.com	12234455	2025-08-06	Male	2025-08-06 09:05:06	2025-08-07 14:17:40	2
5	13	p	o	9999	aa@gmail.com	142425226	1998-05-01	Male	2025-08-07 09:38:05	2025-08-07 14:17:33	2
6	5	John	Anderson	+1-555-9999	john.anderson.updated@email.com	hashed_password_1	1990-05-15	Male	2025-08-07 07:52:00	2025-08-07 07:52:00	1
7	10	John	Anderson	+1-555-1001	john.anderson@email.com	hashed_password_1	1990-05-15	Male	2025-08-07 07:58:10	2025-08-07 07:58:10	2
8	6	Sarah	Johnson	+1-555-1002	sarah.johnson@email.com	hashed_password_2	1985-08-22	Female	2025-08-07 07:52:00	2025-08-07 07:52:00	3
9	7	Michael	Brown	+1-555-1003	michael.brown@email.com	hashed_password_3	1992-12-03	Male	2025-08-07 07:52:00	2025-08-07 07:52:00	3
10	8	Emily	Davis	+1-555-1004	emily.davis@email.com	hashed_password_4	1988-07-18	Female	2025-08-07 07:52:00	2025-08-07 07:52:00	1
11	11	David	Wilson	+1-555-1005	david.wilson@email.com	hashed_password_5	1995-03-10	Male	2025-08-07 07:58:10	2025-08-07 07:58:10	5


ecommerce_db.membership: 14 rows total (exact)

#	MembershipID	MembershipType	Description	DiscountRate	ShippingBenefit	IsActive
1	1	Basic	Enhanced basic membership with new features	0.0	Free shipping on orders over $75	1
2	2	Premium	Premium membership with exclusive access to sales	7.5	(NULL)	1
3	3	VIP	VIP membership with exclusive benefits	10.0	(NULL)	1
4	4	Student	Special discounts for students with valid ID	15.0	Free shipping on orders over $25	1
5	5	Senior	Exclusive benefits for senior citizens	12.0	Free shipping on all orders	1
6	6	Corporate	Bulk purchase benefits for businesses	8.0	Express shipping at standard rates	1
7	7	Platinum	Ultimate membership with maximum benefits	20.0	Free express shipping worldwide	1
8	8	Family	Family plan with shared benefits	7.5	Free shipping on orders over $50	0
9	9	Student	Special discounts for students with valid ID	15.0	Free shipping on orders over $25	1
10	10	Senior	Exclusive benefits for senior citizens	12.0	Free shipping on all orders	1
11	11	Corporate	Bulk purchase benefits for businesses	8.0	Express shipping at standard rates	1
12	12	Student	Special discounts for students with valid ID	15.0	Free shipping on orders over $25	1
13	13	Senior	Exclusive benefits for senior citizens	12.0	Free shipping on all orders	1
14	14	Corporate	Bulk purchase benefits for businesses	8.0	Express shipping at standard rates	1

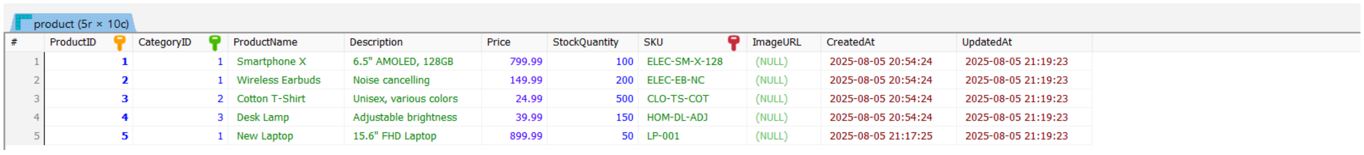


ecommerce_db.order: 8 rows total (exact)

#	OrderID	OrderNumber	OrderDate	RequiredDate	TotalAmount	OrderStatus	PaymentMethod	PaymentStatus	CreatedAt	UpdatedAt	CustomerID	CouponID	Currency	ShippingCost	ShippedDate
1	2	ORD-2024-001	2025-08-07 07:52:01	2024-02-15	1,299.98	Shipped	Credit Card	Completed	2025-08-07 07:52:01	2025-08-07 07:52:01	1	(NULL)	USD	9.99	2024-02-10
2	3	ORD-2024-002	2025-08-07 07:52:01	2024-02-20	259.99	Delivered	PayPal	Completed	2025-08-07 07:52:01	2025-08-07 07:52:01	2	(NULL)	USD	0.0	(NULL)
3	4	ORD-2024-003	2025-08-07 07:52:01	2024-02-25	89.99	Shipped	Debit Card	Pending	2025-08-07 07:52:01	2025-08-07 21:01:41	3	(NULL)	USD	5.99	(NULL)
4	5	ORD-2024-004	2025-08-07 07:52:01	2024-03-01	2,599.98	Completed	Credit Card	Completed	2025-08-07 07:52:01	2025-08-07 07:52:01	4	(NULL)	USD	0.0	(NULL)
5	7	ORD-2025-005	2025-08-07 07:58:11	2025-08-17	1,299.98	Processing	Credit Card	Completed	2025-08-07 07:58:11	2025-08-07 07:58:11	10	(NULL)	USD	9.99	(NULL)
6	8	ORD-2025-006	2025-08-07 07:58:11	2025-08-22	259.99	Shipped	PayPal	Completed	2025-08-07 07:58:11	2025-08-07 07:58:11	6	(NULL)	USD	0.0	(NULL)
7	9	ORD-2025-007	2025-08-07 07:58:11	2025-08-19	999.99	Pending	Debit Card	Pending	2025-08-07 07:58:11	2025-08-07 07:58:11	7	(NULL)	USD	5.99	(NULL)
8	589	oopp	2025-08-07 13:00:36	2020-09-01	344.0	pending	blik	completed	2025-08-07 13:00:36	2025-08-07 13:00:36	10	(NULL)	USD	34.0	2020-10-10

ecommerce_db.orderitem: 9 rows total (exact)

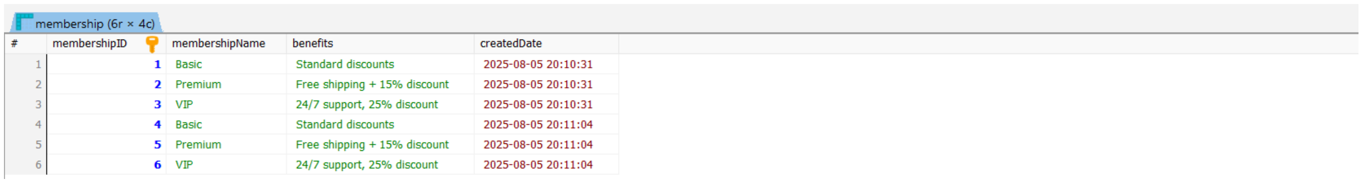
#	OrderItemID	OrderNumber	Quantity	UnitPrice	TotalPrice	ShipDate	Status	VariantInfo	ProductID	OrderID
1	13	ORD-2024-002	2	599.99	1,199.98	2024-01-21	Shipped	Phantom Black, 256GB	3	2
2	14	ORD-2024-002	1	15.99	15.99	2024-01-21	Shipped	Blue, Portable	8	2
3	15	ORD-2024-003	1	1,299.99	1,299.99	2024-01-26	Processing	Midnight, 8GB RAM	2	3
4	16	ORD-2024-003	1	129.99	129.99	2024-01-26	Processing	Black, Size M	7	3
5	17	ORD-2024-004	1	799.99	799.99	2024-02-01	Pending	Space Gray, 256GB	4	4
6	18	ORD-2024-004	2	49.99	99.98	2024-02-01	Pending	iPhone 15 Compatible	9	4
7	19	ORD-2024-005	1	1,499.99	1,499.99	2024-02-06	Cancelled	Platinum Silver, 16GB	5	5
8	20	ORD-2024-007	1	899.99	899.99	2024-02-16	Shipped	Natural Titanium, 256GB	1	7
9	21	ORD-2024-008	2	129.99	259.98	2024-02-20	Processing	Pink, Size S	7	8



product (5r × 10c)

#	ProductID	CategoryID	ProductName	Description	Price	StockQuantity	SKU	ImageURL	CreatedAt	UpdatedAt
1	1	1	Smartphone X	6.5" AMOLED, 128GB	799.99	100	ELEC-SM-X-128	(NULL)	2025-08-05 20:54:24	2025-08-05 21:19:23
2	2	1	Wireless Earbuds	Noise cancelling	149.99	200	ELEC-EB-NC	(NULL)	2025-08-05 20:54:24	2025-08-05 21:19:23
3	3	2	Cotton T-Shirt	Unisex, various colors	24.99	500	CLO-TS-COT	(NULL)	2025-08-05 20:54:24	2025-08-05 21:19:23
4	4	3	Desk Lamp	Adjustable brightness	39.99	150	HOM-DL-ADJ	(NULL)	2025-08-05 20:54:24	2025-08-05 21:19:23
5	5	1	New Laptop	15.6" FHD Laptop	899.99	50	LP-001	(NULL)	2025-08-05 21:17:25	2025-08-05 21:19:23

1.3.2 membership Table Sample Data



membership (6r × 4c)

#	membershipID	membershipName	benefits	createdDate
1	1	Basic	Standard discounts	2025-08-05 20:10:31
2	2	Premium	Free shipping + 15% discount	2025-08-05 20:10:31
3	3	VIP	24/7 support, 25% discount	2025-08-05 20:10:31
4	4	Basic	Standard discounts	2025-08-05 20:11:04
5	5	Premium	Free shipping + 15% discount	2025-08-05 20:11:04
6	6	VIP	24/7 support, 25% discount	2025-08-05 20:11:04

1.4 Design Assumptions and Constraints

1.4.1 Business Rules

Customer Uniqueness: Each customer must have a unique email address

Order Integrity: Orders cannot be deleted if they contain items

Product Availability: Products with zero stock can still be ordered (backorder)

Price Consistency: Order items store historical prices at time of purchase

Soft Deletion: Products and categories are deactivated, not physically deleted

Membership Exclusivity: Each customer can have only one active membership

1.4.2 Data Integrity Constraints

Referential Integrity: All foreign keys must reference existing records

Domain Constraints:

Prices must be non-negative

Stock quantities must be non-negative

Email addresses must follow valid format

Entity Integrity: All primary keys must be unique and non-null

Business Constraints:

Order total must equal sum of order items

Shipped date cannot be before order date

Required date cannot be before order date

1.4.3 Performance Considerations

Indexing Strategy:

Primary keys have clustered indexes

Foreign keys have non-clustered indexes

Email field has unique index

Order date has index for temporal queries

Partitioning: Large tables (ORDER, ORDERITEM) can be partitioned by date

Archiving: Historical data older than 2 years moved to archive tables

2. DATABASE VIEWS

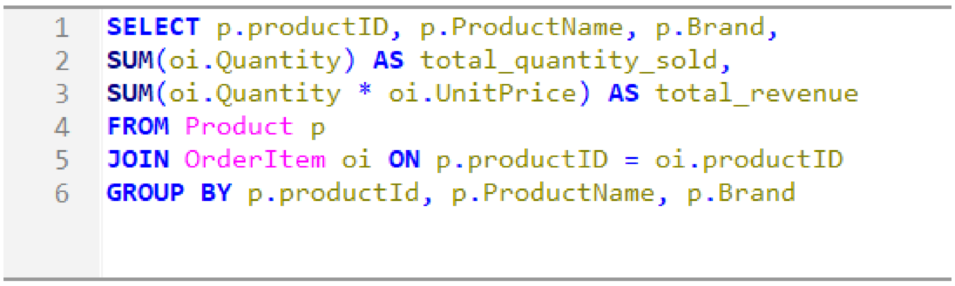
2.1 ProductCatalog View

Purpose: Provides a comprehensive view of active products with category and supplier information for catalog display.

```sql
CREATE VIEW ProductCatalog AS
SELECT
    p.ProductID,
    p.ProductName AS product_name,
    p.Brand,
    p.price,
    p.StockQuantity,
    s.SupplierID,
    s.ContactName AS supplier_name,
    s.Email
FROM Product p
JOIN Supplier s ON p.ProductID = s.SupplierID
WHERE p.StockQuantity > 0;  -- Only show products in stock
```

2.2 vw_productsalessummary

Purpose: Provides a comprehensive view of quantity and revenue of sold product.



```sql
SELECT p.productID, p.ProductName, p.Brand,
SUM(oi.Quantity) AS total_quantity_sold,
SUM(oi.Quantity * oi.UnitPrice) AS total_revenue
FROM Product p
JOIN OrderItem oi ON p.productID = oi.productID
GROUP BY p.productId, p.ProductName, p.Brand
```

2.2 CustomerOrderHistory View

Purpose: Displays complete order history for customers with order details and

status tracking.

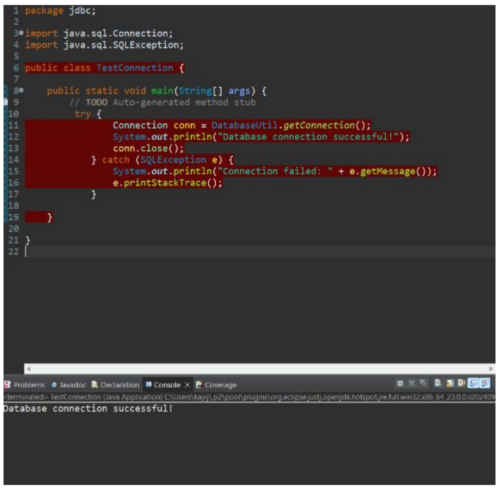
Usage Example:

-- Get recent orders for a specific customer

3. APPLICATION DESCRIPTION

Step 1: testing a database connection in Java using DatabaseUtil.getConnection(). The message "Database connection successful!" in the console confirms the connection was successful.

TestConnection.JAVA------FILE



```java
package jdbc;

import java.sql.Connection;
import java.sql.SQLException;

public class TestConnection {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         try {
                Connection conn = DatabaseUtil.getConnection();
                System.out.println("Database connection successful!");
                conn.close();
            } catch (SQLException e) {
                System.out.println("Connection failed: " + e.getMessage());
                e.printStackTrace();
            }

    }

}
```

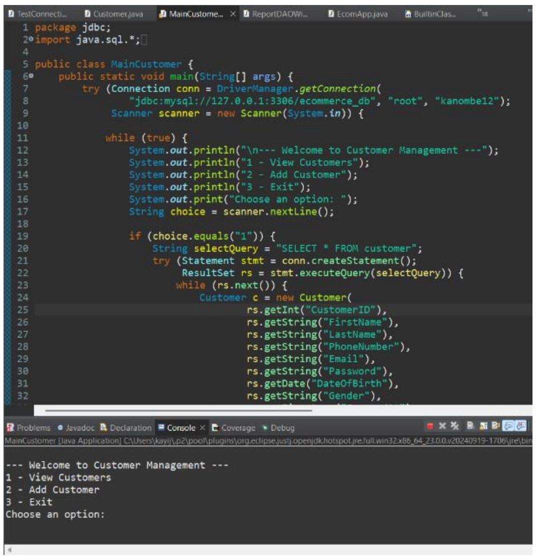
```
<terminated> TestConnection [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v202409
Database connection successful!
```

```java
try (Connection conn = DriverManager.getConnection(
                "jdbc:mysql://127.0.0.1:3306/ecommerce_db", "root", "kanombe12");
             Scanner scanner = new Scanner(System.in)) {
```

Step 2 :

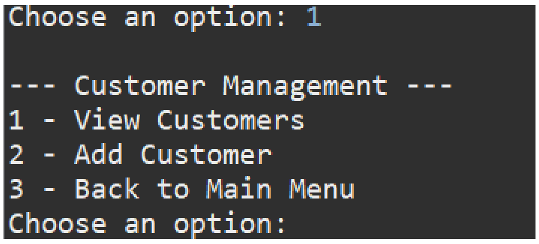
This program is for managing customer records inconsole-based.

MainCustomer.java----------

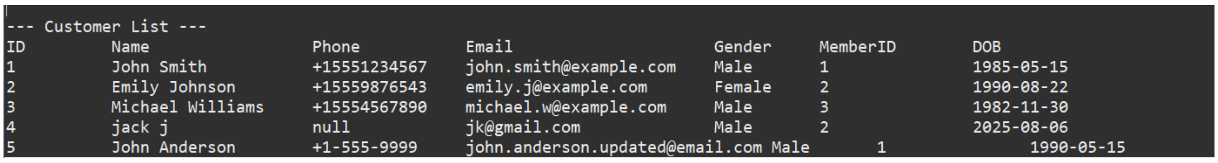


```java
package jdbc;
import java.sql.*;

public class MainCustomer {
    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection(
                "jdbc:mysql://127.0.0.1:3306/ecommerce_db", "root", "kanombe12");
             Scanner scanner = new Scanner(System.in)) {

            while (true) {
                System.out.println("\n--- Welcome to Customer Management ---");
                System.out.println("1 - View Customers");
                System.out.println("2 - Add Customer");
                System.out.println("3 - Exit");
                System.out.print("Choose an option: ");
                String choice = scanner.nextLine();

                if (choice.equals("1")) {
                    String selectQuery = "SELECT * FROM customer";
                    try (Statement stmt = conn.createStatement();
                         ResultSet rs = stmt.executeQuery(selectQuery)) {
                        while (rs.next()) {
                            Customer c = new Customer(
                                    rs.getInt("CustomerID"),
                                    rs.getString("FirstName"),
                                    rs.getString("LastName"),
                                    rs.getString("PhoneNumber"),
                                    rs.getString("Email"),
                                    rs.getString("Password"),
                                    rs.getDate("DateOfBirth"),
                                    rs.getString("Gender"),
```

```
MainCustomer [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin

--- Welcome to Customer Management ---
1 - View Customers
2 - Add Customer
3 - Exit
Choose an option:
```



```
Choose an option: 1

--- Customer Management ---
1 - View Customers
2 - Add Customer
3 - Back to Main Menu
Choose an option:
```

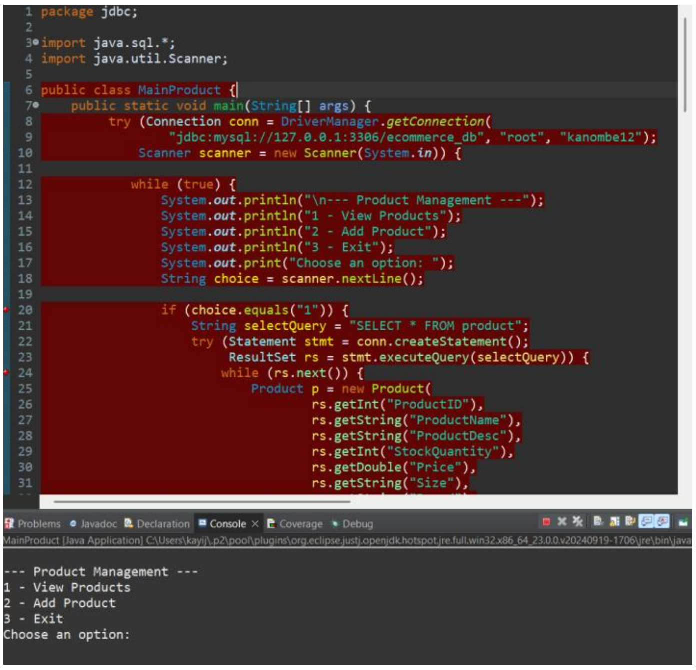


```
--- Customer List ---
ID          Name                Phone           Email                      Gender      MemberID        DOB
1           John Smith          +15551234567    john.smith@example.com     Male        1               1985-05-15
2           Emily Johnson       +15559876543    emily.j@example.com        Female      2               1990-08-22
3           Michael Williams    +15554567890    michael.w@example.com      Male        3               1982-11-30
4           jack j              null            jk@gmail.com               Male        2               2025-08-06
5           John Anderson       +1-555-9999     john.anderson.updated@email.com Male        1               1990-05-15
```

Next u could add up new customer and update

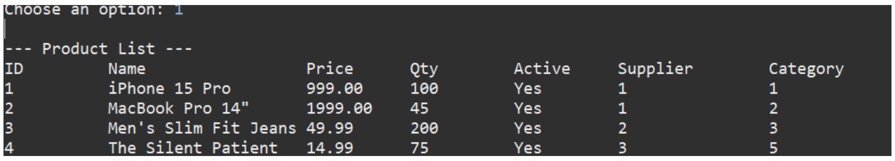
Managing product records in consol – based

MainProduct.java-----------



```java
package jdbc;

import java.sql.*;
import java.util.Scanner;

public class MainProduct {
    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection(
                "jdbc:mysql://127.0.0.1:3306/ecommerce_db", "root", "kanombe12");
             Scanner scanner = new Scanner(System.in)) {

            while (true) {
                System.out.println("\n--- Product Management ---");
                System.out.println("1 - View Products");
                System.out.println("2 - Add Product");
                System.out.println("3 - Exit");
                System.out.print("Choose an option: ");
                String choice = scanner.nextLine();

                if (choice.equals("1")) {
                    String selectQuery = "SELECT * FROM product";
                    try (Statement stmt = conn.createStatement();
                         ResultSet rs = stmt.executeQuery(selectQuery)) {
                        while (rs.next()) {
                            Product p = new Product(
                                    rs.getInt("ProductID"),
                                    rs.getString("ProductName"),
                                    rs.getString("ProductDesc"),
                                    rs.getInt("StockQuantity"),
                                    rs.getDouble("Price"),
                                    rs.getString("Size"),
```

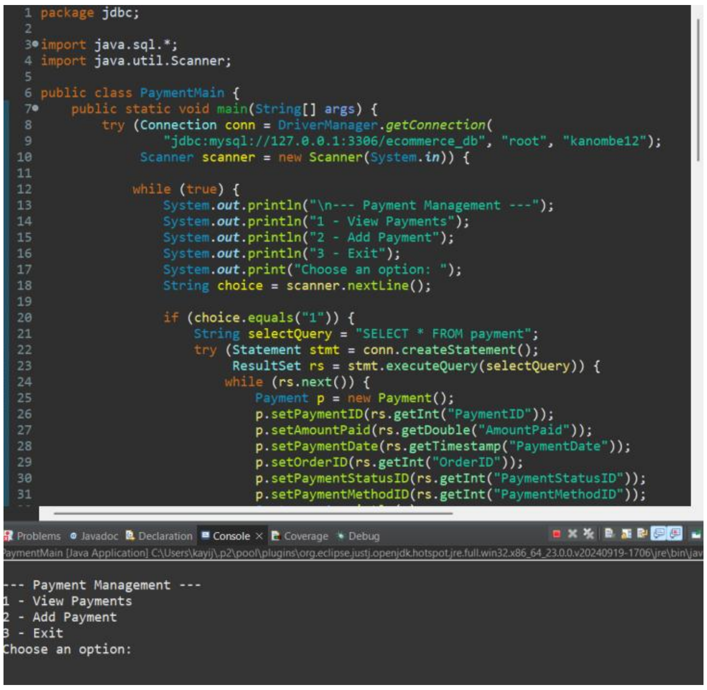
```
MainProduct [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\java

--- Product Management ---
1 - View Products
2 - Add Product
3 - Exit
Choose an option:
```



```
Choose an option: 1

--- Product List ---
ID          Name               Price   Qty       Active      Supplier         Category
1           iPhone 15 Pro      999.00  100       Yes         1                1
2           MacBook Pro 14"    1999.00 45        Yes         1                2
3           Men's Slim Fit Jeans 49.99 200       Yes         2                3
4           The Silent Patient 14.99   75        Yes         3                5
```

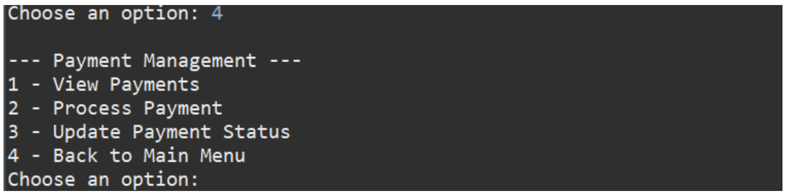
Managing order and payments records

PaymentMain.java----------

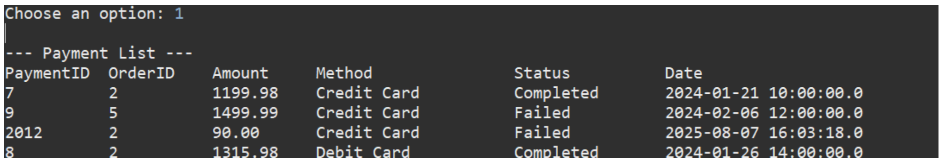


```java
package jdbc;

import java.sql.*;
import java.util.Scanner;

public class PaymentMain {
    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection(
                "jdbc:mysql://127.0.0.1:3306/ecommerce_db", "root", "kanombe12");
             Scanner scanner = new Scanner(System.in)) {

            while (true) {
                System.out.println("\n--- Payment Management ---");
                System.out.println("1 - View Payments");
                System.out.println("2 - Add Payment");
                System.out.println("3 - Exit");
                System.out.print("Choose an option: ");
                String choice = scanner.nextLine();

                if (choice.equals("1")) {
                    String selectQuery = "SELECT * FROM payment";
                    try (Statement stmt = conn.createStatement();
                         ResultSet rs = stmt.executeQuery(selectQuery)) {
                        while (rs.next()) {
                            Payment p = new Payment();
                            p.setPaymentID(rs.getInt("PaymentID"));
                            p.setAmountPaid(rs.getDouble("AmountPaid"));
                            p.setPaymentDate(rs.getTimestamp("PaymentDate"));
                            p.setOrderID(rs.getInt("OrderID"));
                            p.setPaymentStatusID(rs.getInt("PaymentStatusID"));
                            p.setPaymentMethodID(rs.getInt("PaymentMethodID"));
```

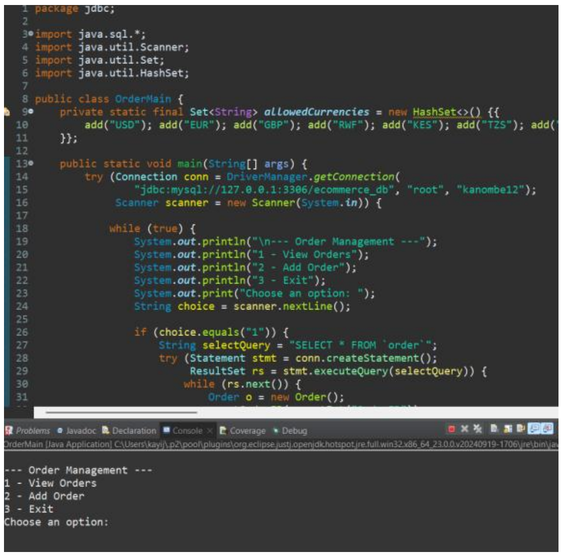
```
PaymentMain [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\jav

--- Payment Management ---
1 - View Payments
2 - Add Payment
3 - Exit
Choose an option:
```



```
Choose an option: 4

--- Payment Management ---
1 - View Payments
2 - Process Payment
3 - Update Payment Status
4 - Back to Main Menu
Choose an option:
```



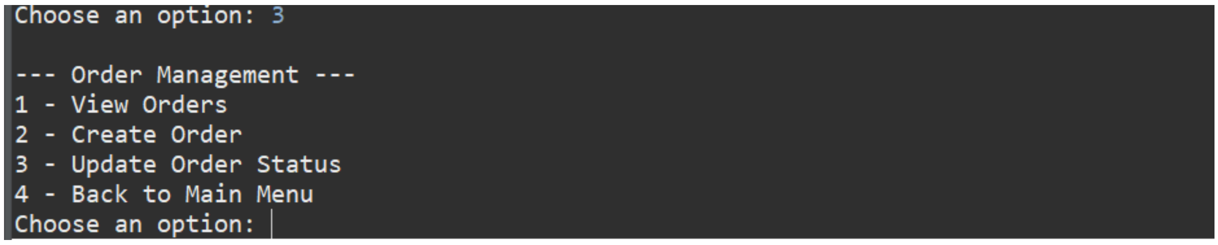
```
Choose an option: 1

--- Payment List ---
PaymentID  OrderID    Amount     Method              Status         Date
7          2          1199.98    Credit Card         Completed      2024-01-21 10:00:00.0
9          5          1499.99    Credit Card         Failed         2024-02-06 12:00:00.0
2012       2          90.00      Credit Card         Failed         2025-08-07 16:03:18.0
8          2          1315.98    Debit Card          Completed      2024-01-26 14:00:00.0
```



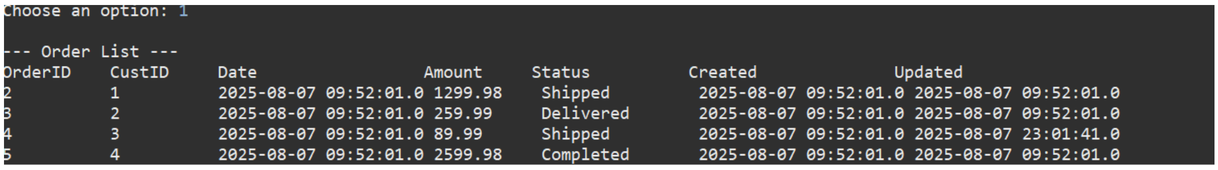
```java
package jdbc;

import java.sql.*;
import java.util.Scanner;
import java.util.Set;
import java.util.HashSet;

public class OrderMain {
    private static final Set<String> allowedCurrencies = new HashSet<>() {{
        add("USD"); add("EUR"); add("GBP"); add("RWF"); add("KES"); add("TZS"); add(
    }};

    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection(
                "jdbc:mysql://127.0.0.1:3306/ecommerce_db", "root", "kanombe12");
             Scanner scanner = new Scanner(System.in)) {

            while (true) {
                System.out.println("\n--- Order Management ---");
                System.out.println("1 - View Orders");
                System.out.println("2 - Add Order");
                System.out.println("3 - Exit");
                System.out.print("Choose an option: ");
                String choice = scanner.nextLine();

                if (choice.equals("1")) {
                    String selectQuery = "SELECT * FROM `order`";
                    try (Statement stmt = conn.createStatement();
                         ResultSet rs = stmt.executeQuery(selectQuery)) {
                        while (rs.next()) {
                            Order o = new Order();
```

```
OrderMain [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\jav

--- Order Management ---
1 - View Orders
2 - Add Order
3 - Exit
Choose an option:
```

OrderMain.java--------------



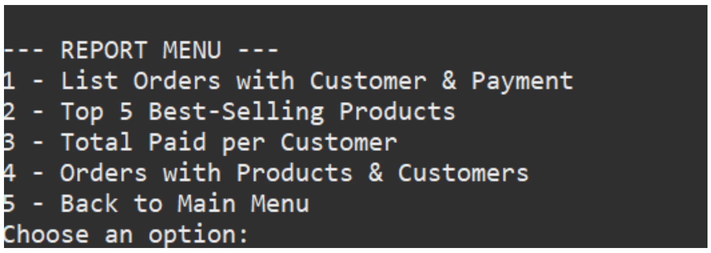
```
Choose an option: 3

--- Order Management ---
1 - View Orders
2 - Create Order
3 - Update Order Status
4 - Back to Main Menu
Choose an option:
```



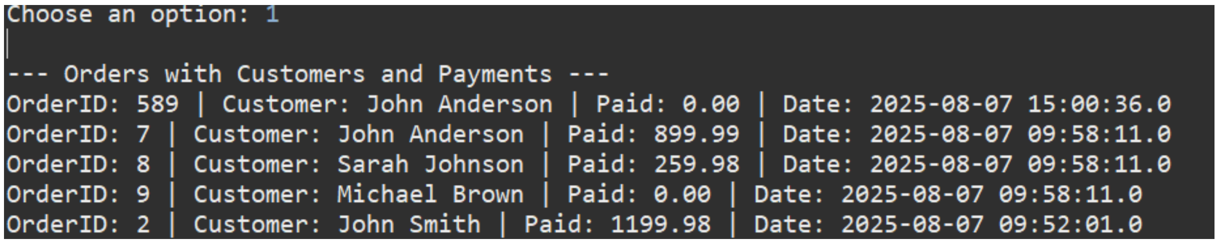
```
Choose an option: 1

--- Order List ---
OrderID    CustID     Date                  Amount     Status          Created               Updated
2          1          2025-08-07 09:52:01.0 1299.98     Shipped         2025-08-07 09:52:01.0 2025-08-07 09:52:01.0
3          2          2025-08-07 09:52:01.0 259.99      Delivered       2025-08-07 09:52:01.0 2025-08-07 09:52:01.0
4          3          2025-08-07 09:52:01.0 89.99       Shipped         2025-08-07 09:52:01.0 2025-08-07 23:01:41.0
5          4          2025-08-07 09:52:01.0 2599.98     Completed       2025-08-07 09:52:01.0 2025-08-07 09:52:01.0
```

And later update the status if goods where being shipped.

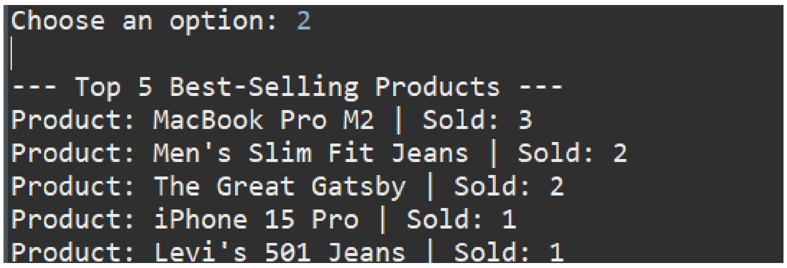
REPORTS



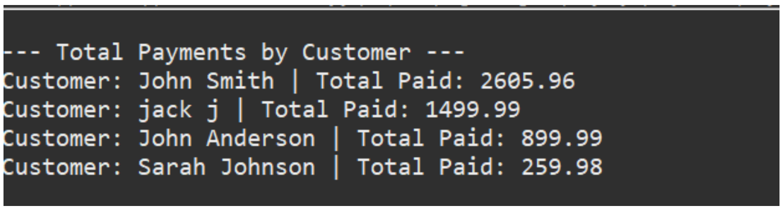
```
--- REPORT MENU ---
1 - List Orders with Customer & Payment
2 - Top 5 Best-Selling Products
3 - Total Paid per Customer
4 - Orders with Products & Customers
5 - Back to Main Menu
Choose an option:
```



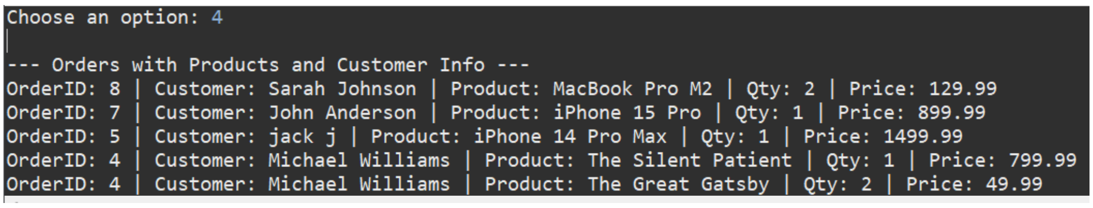
```
Choose an option: 1

--- Orders with Customers and Payments ---
OrderID: 589 | Customer: John Anderson | Paid: 0.00 | Date: 2025-08-07 15:00:36.0
OrderID: 7 | Customer: John Anderson | Paid: 899.99 | Date: 2025-08-07 09:58:11.0
OrderID: 8 | Customer: Sarah Johnson | Paid: 259.98 | Date: 2025-08-07 09:58:11.0
OrderID: 9 | Customer: Michael Brown | Paid: 0.00 | Date: 2025-08-07 09:58:11.0
OrderID: 2 | Customer: John Smith | Paid: 1199.98 | Date: 2025-08-07 09:52:01.0
```



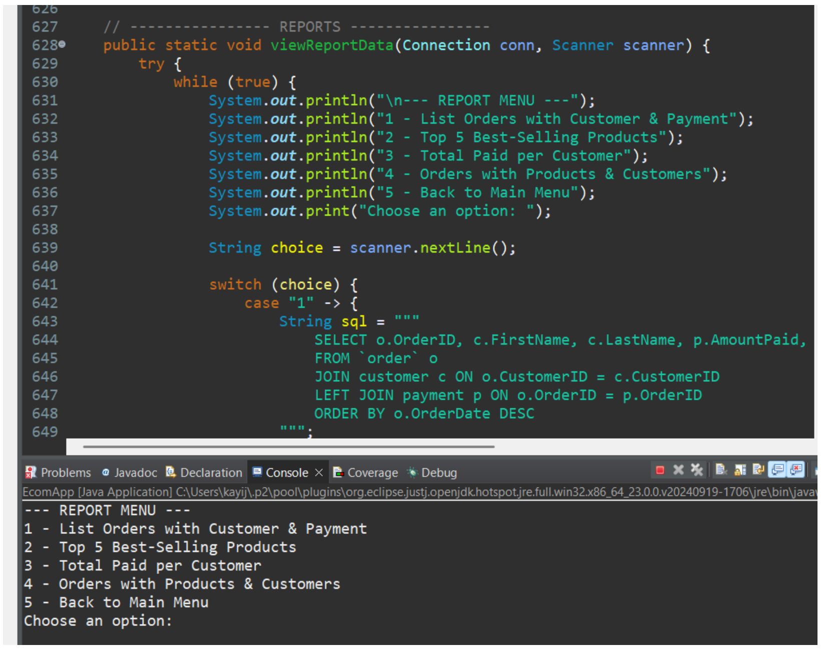
```
Choose an option: 2

--- Top 5 Best-Selling Products ---
Product: MacBook Pro M2 | Sold: 3
Product: Men's Slim Fit Jeans | Sold: 2
Product: The Great Gatsby | Sold: 2
Product: iPhone 15 Pro | Sold: 1
Product: Levi's 501 Jeans | Sold: 1
```



```
--- Total Payments by Customer ---
Customer: John Smith | Total Paid: 2605.96
Customer: jack j | Total Paid: 1499.99
Customer: John Anderson | Total Paid: 899.99
Customer: Sarah Johnson | Total Paid: 259.98
```



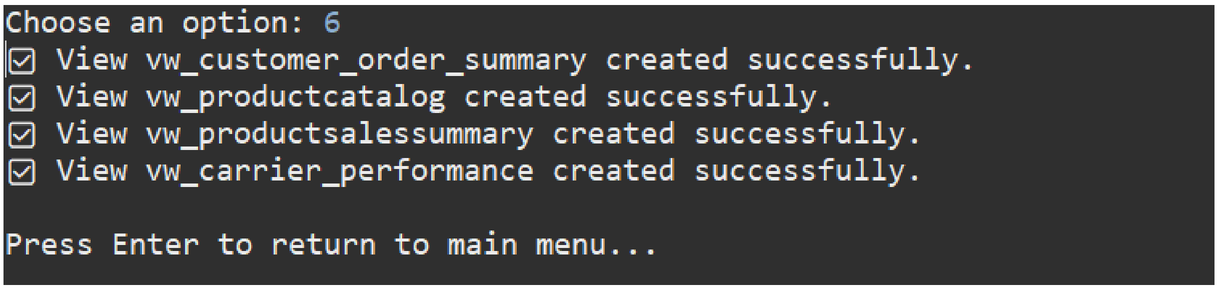
```
Choose an option: 4

--- Orders with Products and Customer Info ---
OrderID: 8 | Customer: Sarah Johnson | Product: MacBook Pro M2 | Qty: 2 | Price: 129.99
OrderID: 7 | Customer: John Anderson | Product: iPhone 15 Pro | Qty: 1 | Price: 899.99
OrderID: 5 | Customer: jack j | Product: iPhone 14 Pro Max | Qty: 1 | Price: 1499.99
OrderID: 4 | Customer: Michael Williams | Product: The Silent Patient | Qty: 1 | Price: 799.99
OrderID: 4 | Customer: Michael Williams | Product: The Great Gatsby | Qty: 2 | Price: 49.99
```



```java
    // ---------------- REPORTS ----------------
    public static void viewReportData(Connection conn, Scanner scanner) {
        try {
            while (true) {
                System.out.println("\n--- REPORT MENU ---");
                System.out.println("1 - List Orders with Customer & Payment");
                System.out.println("2 - Top 5 Best-Selling Products");
                System.out.println("3 - Total Paid per Customer");
                System.out.println("4 - Orders with Products & Customers");
                System.out.println("5 - Back to Main Menu");
                System.out.print("Choose an option: ");

                String choice = scanner.nextLine();

                switch (choice) {
                    case "1" -> {
                        String sql = """
                            SELECT o.OrderID, c.FirstName, c.LastName, p.AmountPaid,
                            FROM `order` o
                            JOIN customer c ON o.CustomerID = c.CustomerID
                            LEFT JOIN payment p ON o.OrderID = p.OrderID
                            ORDER BY o.OrderDate DESC
                        """;
```

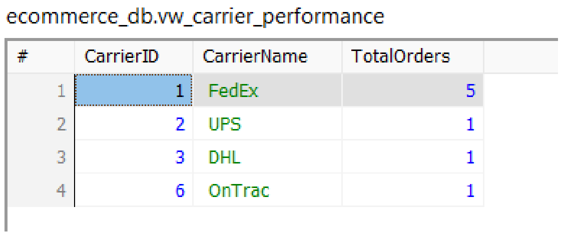
```
EcomApp [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\javav
--- REPORT MENU ---
1 - List Orders with Customer & Payment
2 - Top 5 Best-Selling Products
3 - Total Paid per Customer
4 - Orders with Products & Customers
5 - Back to Main Menu
Choose an option:
```

VIEW



```
Choose an option: 6
☑ View vw_customer_order_summary created successfully.
☑ View vw_productcatalog created successfully.
☑ View vw_productsalessummary created successfully.
☑ View vw_carrier_performance created successfully.

Press Enter to return to main menu...
```

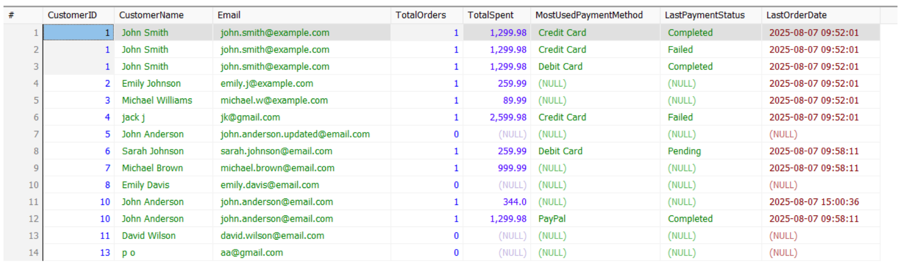
vw_carrier_performance



ecommerce_db.vw_carrier_performance

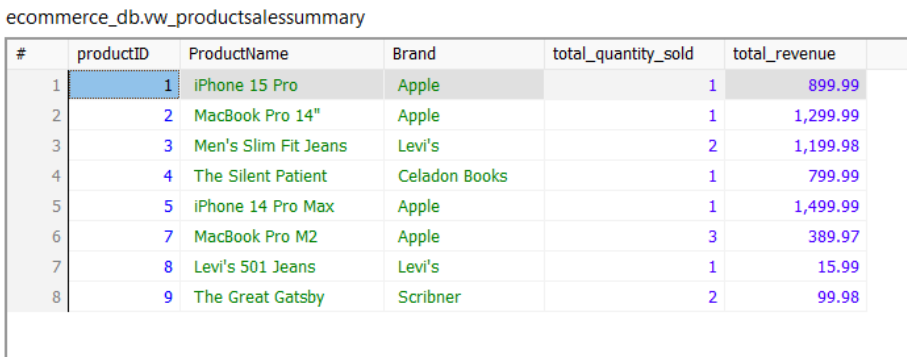
#	CarrierID	CarrierName	TotalOrders
1	1	FedEx	5
2	2	UPS	1
3	3	DHL	1
4	6	OnTrac	1

vw_customer_order_summary



#	CustomerID	CustomerName	Email	TotalOrders	TotalSpent	MostUsedPaymentMethod	LastPaymentStatus	LastOrderDate
1	1	John Smith	john.smith@example.com	1	1,299.98	Credit Card	Completed	2025-08-07 09:52:01
2	1	John Smith	john.smith@example.com	1	1,299.98	Credit Card	Failed	2025-08-07 09:52:01
3	1	John Smith	john.smith@example.com	1	1,299.98	Debit Card	Completed	2025-08-07 09:52:01
4	2	Emily Johnson	emily.j@example.com	1	259.99	(NULL)	(NULL)	2025-08-07 09:52:01
5	3	Michael Williams	michael.w@example.com	1	89.99	(NULL)	(NULL)	2025-08-07 09:52:01
6	4	jack j	jk@gmail.com	1	2,599.98	Credit Card	Failed	2025-08-07 09:52:01
7	5	John Anderson	john.anderson.updated@email.com	0	(NULL)	(NULL)	(NULL)	(NULL)
8	6	Sarah Johnson	sarah.johnson@email.com	1	259.99	Debit Card	Pending	2025-08-07 09:58:11
9	7	Michael Brown	michael.brown@email.com	1	999.99	(NULL)	(NULL)	2025-08-07 09:58:11
10	8	Emily Davis	emily.davis@email.com	0	(NULL)	(NULL)	(NULL)	(NULL)
11	10	John Anderson	john.anderson@email.com	1	344.0	(NULL)	(NULL)	2025-08-07 15:00:36
12	10	John Anderson	john.anderson@email.com	1	1,299.98	PayPal	Completed	2025-08-07 09:58:11
13	11	David Wilson	david.wilson@email.com	0	(NULL)	(NULL)	(NULL)	(NULL)
14	13	p o	aa@gmail.com	0	(NULL)	(NULL)	(NULL)	(NULL)

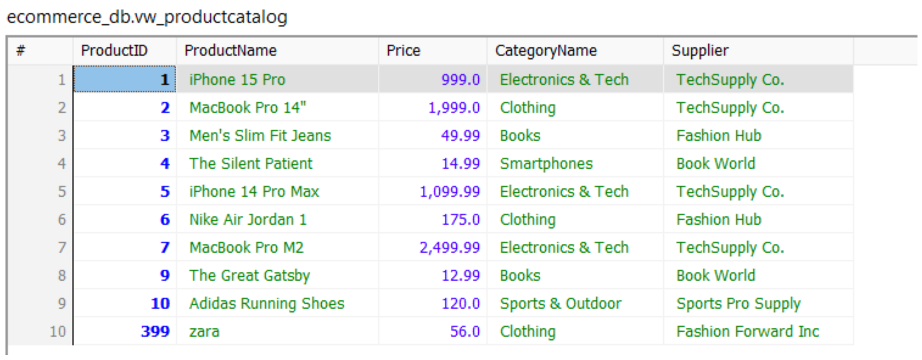
vw_productsalessummary



ecommerce_db.vw_productsalessummary

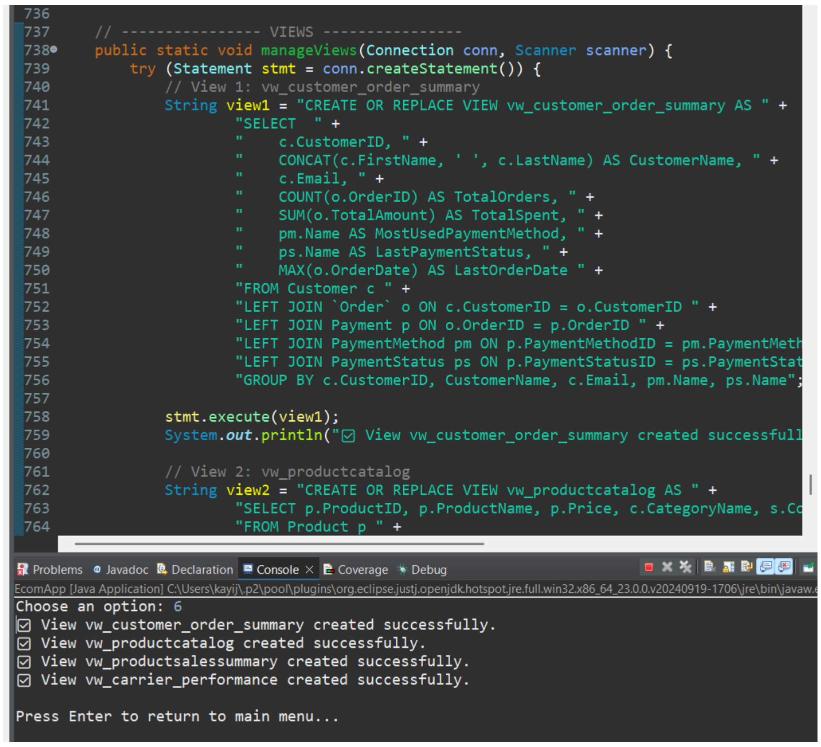
#	productID	ProductName	Brand	total_quantity_sold	total_revenue
1	1	iPhone 15 Pro	Apple	1	899.99
2	2	MacBook Pro 14"	Apple	1	1,299.99
3	3	Men's Slim Fit Jeans	Levi's	2	1,199.98
4	4	The Silent Patient	Celadon Books	1	799.99
5	5	iPhone 14 Pro Max	Apple	1	1,499.99
6	7	MacBook Pro M2	Apple	3	389.97
7	8	Levi's 501 Jeans	Levi's	1	15.99
8	9	The Great Gatsby	Scribner	2	99.98

vw_productcatalog



ecommerce_db.vw_productcatalog

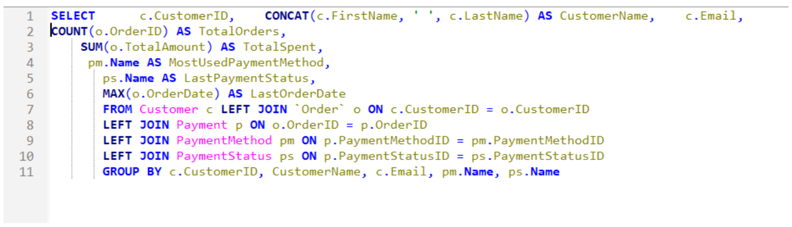
#	ProductID	ProductName	Price	CategoryName	Supplier
1	1	iPhone 15 Pro	999.0	Electronics & Tech	TechSupply Co.
2	2	MacBook Pro 14"	1,999.0	Clothing	TechSupply Co.
3	3	Men's Slim Fit Jeans	49.99	Books	Fashion Hub
4	4	The Silent Patient	14.99	Smartphones	Book World
5	5	iPhone 14 Pro Max	1,099.99	Electronics & Tech	TechSupply Co.
6	6	Nike Air Jordan 1	175.0	Clothing	Fashion Hub
7	7	MacBook Pro M2	2,499.99	Electronics & Tech	TechSupply Co.
8	9	The Great Gatsby	12.99	Books	Book World
9	10	Adidas Running Shoes	120.0	Sports & Outdoor	Sports Pro Supply
10	399	zara	56.0	Clothing	Fashion Forward Inc



```
    // --------------- VIEWS ---------------
    public static void manageViews(Connection conn, Scanner scanner) {
        try (Statement stmt = conn.createStatement()) {
            // View 1: vw_customer_order_summary
            String view1 = "CREATE OR REPLACE VIEW vw_customer_order_summary AS " +
                    "SELECT  " +
                    "    c.CustomerID, " +
                    "    CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, " +
                    "    c.Email, " +
                    "    COUNT(o.OrderID) AS TotalOrders, " +
                    "    SUM(o.TotalAmount) AS TotalSpent, " +
                    "    pm.Name AS MostUsedPaymentMethod, " +
                    "    ps.Name AS LastPaymentStatus, " +
                    "    MAX(o.OrderDate) AS LastOrderDate " +
                    "FROM Customer c " +
                    "LEFT JOIN `Order` o ON c.CustomerID = o.CustomerID " +
                    "LEFT JOIN Payment p ON o.OrderID = p.OrderID " +
                    "LEFT JOIN PaymentMethod pm ON p.PaymentMethodID = pm.PaymentMeth
                    "LEFT JOIN PaymentStatus ps ON p.PaymentStatusID = ps.PaymentStat
                    "GROUP BY c.CustomerID, CustomerName, c.Email, pm.Name, ps.Name";

            stmt.execute(view1);
            System.out.println("☑ View vw_customer_order_summary created successfull

            // View 2: vw_productcatalog
            String view2 = "CREATE OR REPLACE VIEW vw_productcatalog AS " +
                    "SELECT p.ProductID, p.ProductName, p.Price, c.CategoryName, s.Co
                    "FROM Product p " +
```

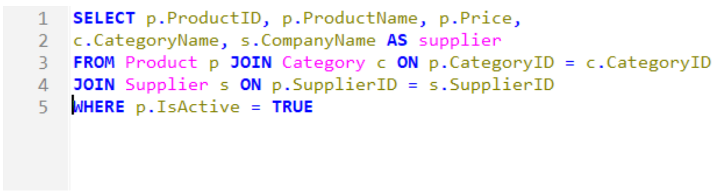
```
EcomApp [Java Application] C:\Users\kayij\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.0.v20240919-1706\jre\bin\javaw.e
Choose an option: 6
☑ View vw_customer_order_summary created successfully.
☑ View vw_productcatalog created successfully.
☑ View vw_productsalessummary created successfully.
☑ View vw_carrier_performance created successfully.

Press Enter to return to main menu...
```

View 1: vw_customer_order_summary



```
SELECT      c.CustomerID,     CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,     c.Email,
COUNT(o.OrderID) AS TotalOrders,
    SUM(o.TotalAmount) AS TotalSpent,
   pm.Name AS MostUsedPaymentMethod,
      ps.Name AS LastPaymentStatus,
      MAX(o.OrderDate) AS LastOrderDate
      FROM Customer c LEFT JOIN `Order` o ON c.CustomerID = o.CustomerID
      LEFT JOIN Payment p ON o.OrderID = p.OrderID
      LEFT JOIN PaymentMethod pm ON p.PaymentMethodID = pm.PaymentMethodID
      LEFT JOIN PaymentStatus ps ON p.PaymentStatusID = ps.PaymentStatusID
      GROUP BY c.CustomerID, CustomerName, c.Email, pm.Name, ps.Name
```

View 2: vw_productcatalog

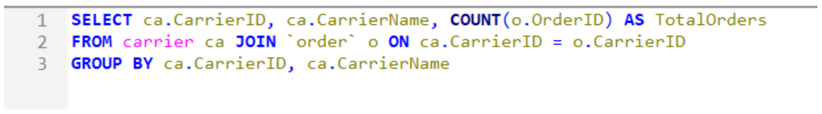


```
SELECT p.ProductID, p.ProductName, p.Price,
c.CategoryName, s.CompanyName AS supplier
FROM Product p JOIN Category c ON p.CategoryID = c.CategoryID
JOIN Supplier s ON p.SupplierID = s.SupplierID
WHERE p.IsActive = TRUE
```

View 3 :vw_productsalessummary

```
SELECT p.productID, p.ProductName, p.Brand,
SUM(oi.Quantity) AS total_quantity_sold,
SUM(oi.Quantity * oi.UnitPrice) AS total_revenue
FROM Product p
JOIN OrderItem oi ON p.productID = oi.productID
GROUP BY p.productId, p.ProductName, p.Brand
```

View 4: vw_carrier_performance



```sql
SELECT ca.CarrierID, ca.CarrierName, COUNT(o.OrderID) AS TotalOrders
FROM carrier ca JOIN `order` o ON ca.CarrierID = o.CarrierID
GROUP BY ca.CarrierID, ca.CarrierName
```

3.1 Application Architecture

The E-commerce Management System is built using a three-tier architecture with the following components:

Data Access Layer:

DAO (Data Access Object) pattern implementation

JDBC connectivity to MariaDB database

Connection pooling for performance

SQL query optimization

3.2 Technology Stack

Frontend:

Java 22 (OpenJDK)

JFreeChart for data visualization

Database:

MariaDB 10.6 (MySQL compatible)

Build Tools:

mysql-connector-j-8.4.0