Online Retail Management System

Author:Boiko Yaroslav

1. Short Description:

The Online Retail Management System is designed to manage customer information, product inventory, orders, deliveries, and payments for an online retail store. It allows customers to browse products, place orders, and choose delivery methods, while providing administrators with tools to track sales, manage inventory, and handle customer interactions.

2. Goal:

The Online Retail Management System aims to streamline and enhance the online shopping experience for both customers and administrators. It facilitates a seamless customer journey from product browsing to order placement and delivery, while providing administrators with comprehensive tools for inventory management, order tracking, and customer engagement.

3. Users:

Customers

Administrators

4. Functions (Use Cases) of the System:

1. Place an Order: Customers can place orders for products.

2. View Order History: Customers and administrators can view the order history.

3. Manage Product Inventory: Administrators can add, update, or remove products from the inventory.

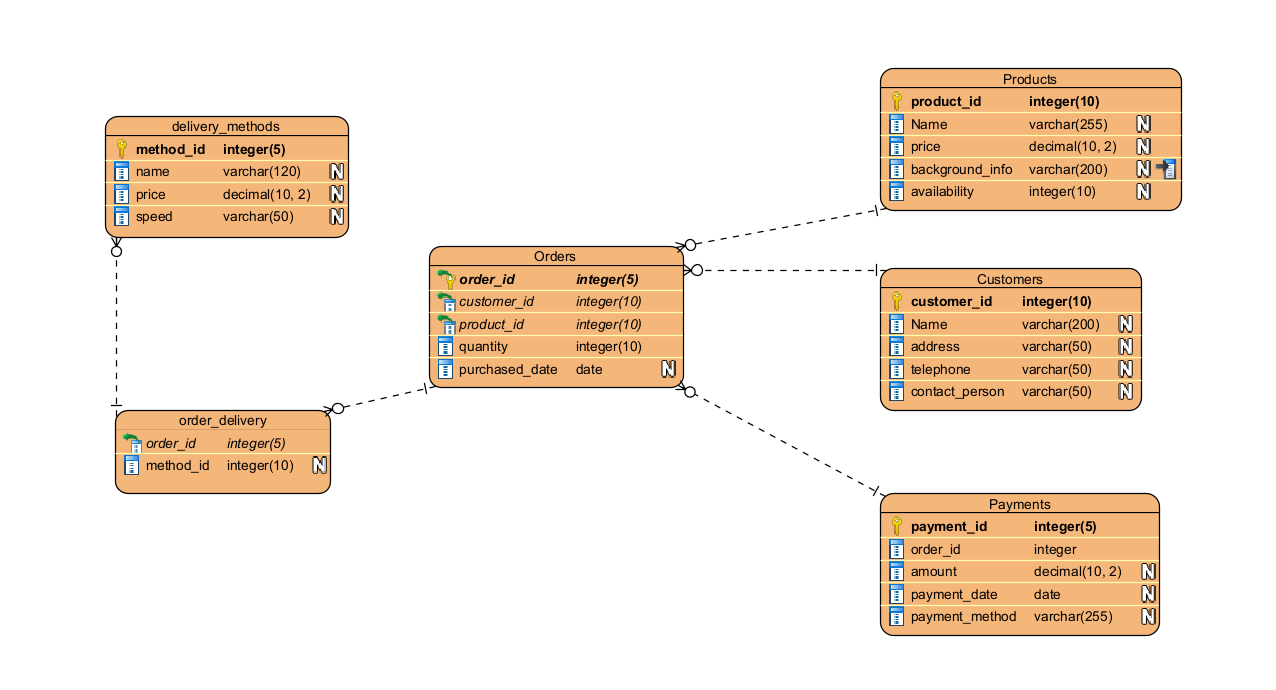
4. Track Delivery Status: Customers and administrators can track the status of deliveries.

5. Handle Payments: System can handle various payment methods for orders.

6. Generate Sales Reports: Administrators can generate reports on product sales.

7. Manage Customer Information: Administrators can manage customer information.

8. Offer Promotions: System can apply promotions or discounts during checkout.



**5. Data Diagram in SQL EXPRESS (SSMS) (3NF):**

**Customer Table (3NF):**

CREATE DATABASE my\_db;

GO

USE my\_db;

GO

CREATE TABLE customers

(

customer\_id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100),

address NVARCHAR(255),

telephone NVARCHAR(15),

contact\_person NVARCHAR(100)

);

GO

**INSERT INTO** customers (name, address, telephone, contact\_person)

VALUES ('John Doe', '123 Main St', '555-1234', 'Jane Smith'),

('Alice Johnson', '456 Oak St', '555-5678', 'Bob Miller'),

('Eva White', '789 Maple Ln', '555-8765', 'Charlie Davis'),

('Michael Brown', '101 Cedar Dr', '555-4321', 'Sophia Johnson'),

('David Smith', '202 Birch Ave', '555-6789', 'Olivia Taylor'),

('James Miller', '303 Pine Rd', '555-9876', 'Emma Davis');

**Products Table (3NF):**

CREATE TABLE products

(

product\_id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100),

price DECIMAL(10, 2),

background\_info NVARCHAR(MAX),

availability BIT

);

GO

**INSERT INTO** products (name, price, background\_info, availability)

VALUES ('Product A', 19.99, 'Some information', 1),

('Product B', 29.99, 'More details about Product B', 1),

('Product C', 39.99, 'Additional details for Product C', 1),

('Product D', 49.99, 'Details about Product D', 1),

('Product E', 59.99, 'Description for Product E', 1),

('Product F', 69.99, 'Details about Product F', 1);

**Orders Table (3NF):**

CREATE TABLE orders

(

order\_id INT IDENTITY(1,1) PRIMARY KEY,

customer\_id INT,

product\_id INT,

quantity INT,

purchase\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES customers (customer\_id),

FOREIGN KEY (product\_id) REFERENCES products (product\_id)

);

GO

**INSERT INTO orders** (customer\_id, product\_id, quantity, purchase\_date)

VALUES (1, 1, 2, CONVERT(DATE, GETDATE())),

(2, 2, 3, CONVERT(DATE, GETDATE())),

(3, 4, 1, CONVERT(DATE, GETDATE())),

(1, 5, 2, CONVERT(DATE, GETDATE())),

(4, 3, 3, CONVERT(DATE, GETDATE())),

(5, 1, 1, CONVERT(DATE, GETDATE()));

**Delivery Methods Table (3NF):**

CREATE TABLE delivery\_methods

(

method\_id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100),

price DECIMAL(10, 2),

speed NVARCHAR(50)

);

GO

**INSERT INTO delivery\_methods** (name, price, speed)

VALUES ('Standard', 5.00, '2-5 days'),

('Priority', 15.00, '1 day'),

('Economy', 3.50, '5-7 days'),

('Same Day', 25.00, 'Within hours'),

('Two-Day', 8.00, '2 days'),

('Super Express', 30.00, '4-6 hours'),

('Express', 10.00, '1-2 days');

**Order Delivery Table (3NF):**

CREATE TABLE order\_delivery

(

order\_id INT,

method\_id INT,

PRIMARY KEY (order\_id, method\_id),

FOREIGN KEY (order\_id) REFERENCES orders (order\_id),

FOREIGN KEY (method\_id) REFERENCES delivery\_methods (method\_id)

);

GO

**INSERT INTO order\_delivery** (order\_id, method\_id)

VALUES (1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5);

**Payments table (3NF):**

CREATE TABLE payments

(

payment\_id INT IDENTITY(1,1) PRIMARY KEY,

order\_id INT,

amount DECIMAL(10, 2),

payment\_date DATE,

payment\_method NVARCHAR(50),

FOREIGN KEY (order\_id) REFERENCES orders (order\_id)

);

GO

**INSERT INTO payments** (order\_id, amount, payment\_date, payment\_method)

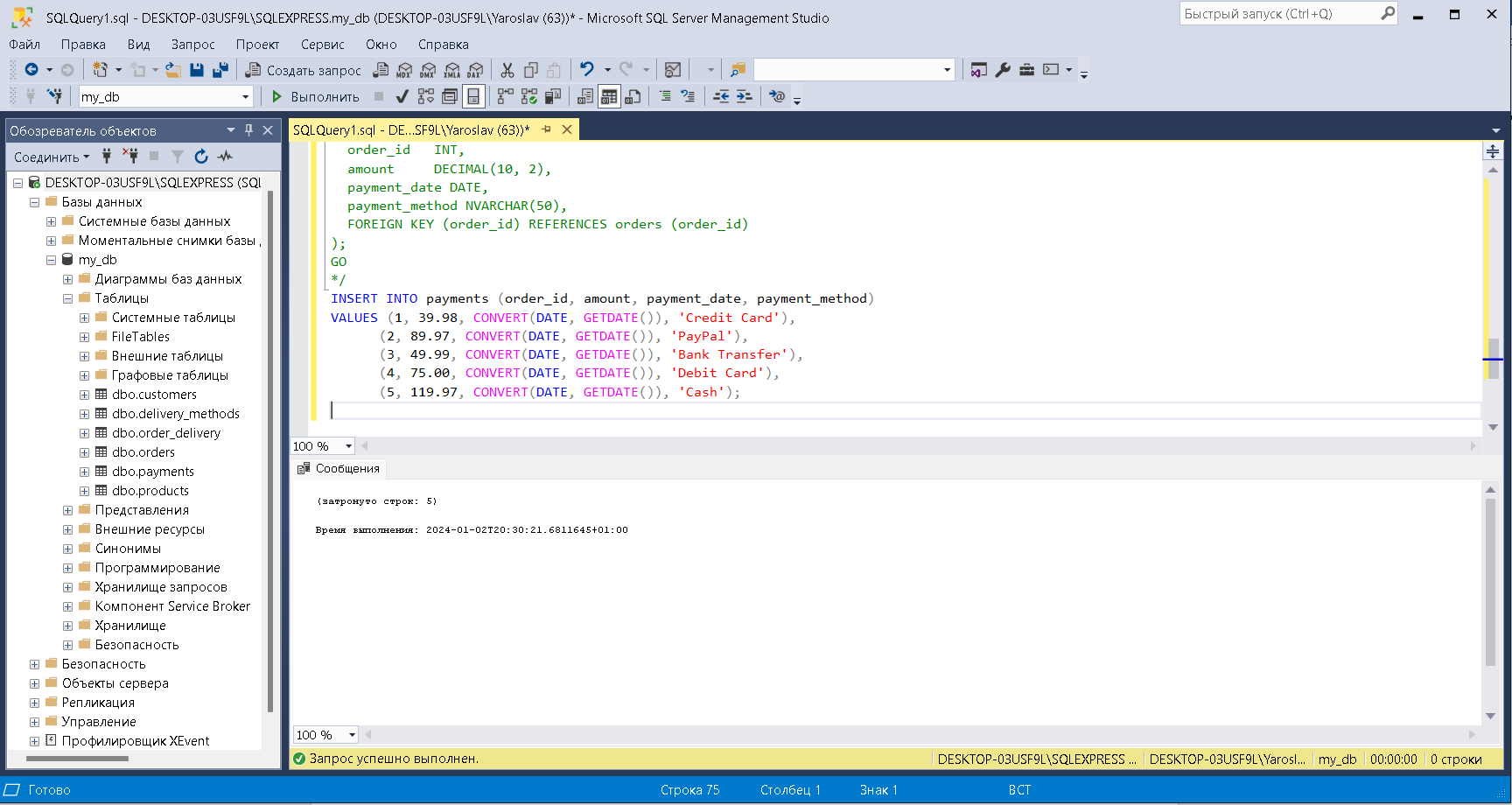
VALUES (1, 39.98, CONVERT(DATE, GETDATE()), 'Credit Card'),

(2, 89.97, CONVERT(DATE, GETDATE()), 'PayPal'),

(3, 49.99, CONVERT(DATE, GETDATE()), 'Bank Transfer'),

(4, 75.00, CONVERT(DATE, GETDATE()), 'Debit Card'),

(5, 119.97, CONVERT(DATE, GETDATE()), 'Cash');



6. Verification of Normal Forms:

Every table is in Third Normal Form (3NF) as they meet the criteria:

Each table is in 1NF (atomic values in each column).

Each table is in 2NF (no partial dependencies on the primary key).

Each table is in 3NF (no transitive dependencies).

7. Views:

Explanation and Sql query of Views are provided ***in 8 views.txt file***

Assumptions:

- Views are used for reporting and analytics.

- Views are read-only and do not modify underlying data.

These views provide insights into different aspects of the online retail management system, allowing users to analyze and retrieve specific information without directly interacting with the underlying tables. Adjust the queries based on specific business requirements.