|  |  |
| --- | --- |
| BloomCart  *By: Kavita Boywah* | From Our Online Cart to Their Heart—Celebrate Life’s Moments with Flowers. |

## Project Selection, System Design Requirements, Process

**Mini World:**

Online Florist (modeled on services like 1-800-Flowers)

**Objective:**

Design a database system that supports an online florist business with functionality to manage customers, floral products, orders, deliveries, and payments.

**Entities:**

**1.Customer**

- CustomerID (PK)

- FirstName

- LastName

- Email

- Phone

- Address (Street, City, State, Zip)

**2. Product (Floral Arrangement)**

- ProductID (PK)

- ProductName

- Description

- Price

- Category (e.g., Birthday, Sympathy, Wedding)

**3. Order**

- OrderID (PK)

- CustomerID (FK)

- OrderDate

- DeliveryDate

- TotalAmount

- PaymentStatus (Paid/Unpaid)

**4. OrderDetails**

- OrderDetailID (PK)

- OrderID (FK)

- ProductID (FK)

- Quantity

- UnitPrice

**5. Payment**

- PaymentID (PK)

- OrderID (FK)

- PaymentMethod (Credit, PayPal, etc.)

- PaymentDate

- Amount

**6. Delivery**

- DeliveryID (PK)

- OrderID (FK)

- DeliveryAddress

- DeliveryStatus (Pending, In Transit, Delivered)

- DeliveryTimeWindow

**Documentation of Process:**

I worked for a florist from 2014 to 2020, which gave me firsthand experience with how the operations of a floral business work—from taking orders and arranging products to scheduling deliveries and handling payments. I drew from that experience to determine the essential entities and relationships for this project. I remembered how we kept detailed customer records, product catalogs with seasonal availability, and how important it was to accurately manage deliveries. Payment tracking and multiple delivery windows were also common, especially for weddings and special events. All of this influenced what I decided to include in the database design.

## ENTITY-RELATIONSHIP (ER) DIAGRAMS-

Hand drawn using IPAD

A diagram of a company

Description automatically generated

**MICROSOFT ACCESS DATABASE**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

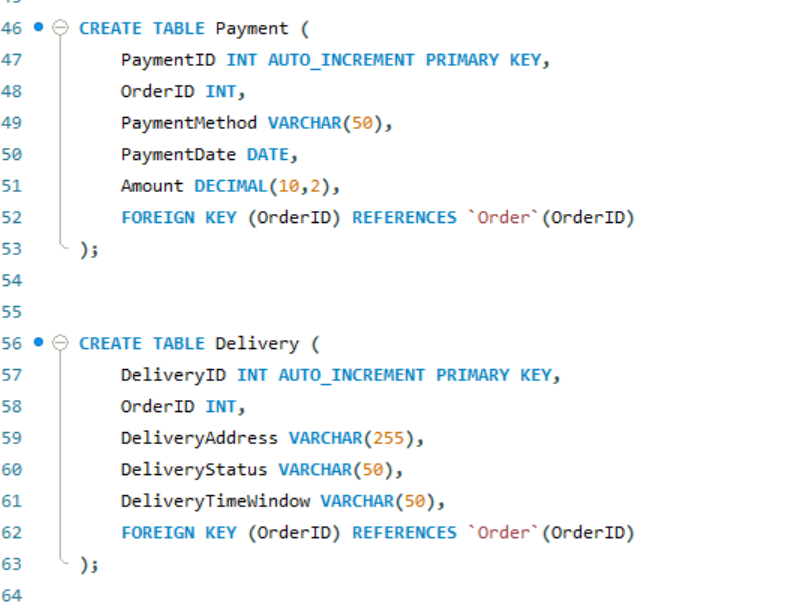
Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**SQL ScriptA screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated**

**PROJECT REPORT**

Designing and developing the BloomCart database project was both a practical and rewarding experience. The goal of this project was to create a robust and normalized relational database for an online florist business modeled after companies like 1-800-Flowers. Drawing upon my prior work experience at a florist shop from 2014 to 2020, I was able to bring real-world insight into the system design, ensuring that it reflected the actual operational needs of a floral business, including customer management, inventory tracking, order processing, payment handling, and delivery logistics.

The first phase of the project involved identifying the essential components of the system. I designed a Chen-style Entity-Relationship Diagram (ERD) that included six primary entities: Customer, Product, Order, OrderDetails, Payment, and Delivery. These entities were chosen to capture the full scope of the business process. For instance, the OrderDetails table served as a junction table to facilitate the many-to-many relationship between orders and products. Each entity was assigned appropriate attributes, with clearly defined primary keys and foreign keys to maintain referential integrity.

Once the data model was finalized, I used Microsoft Access to create and implement the tables. I defined field types carefully, ensuring they matched the nature of the data they would store. Establishing foreign keys in Access required the use of the “Relationships” window rather than defining them directly in the table design view, which was a learning curve at first. With the relationships established and integrity enforced, I then populated each table with sample data—five to ten records per table—to simulate real business scenarios, such as placing orders, making payments, and coordinating deliveries.

Following the development in Access, the next step was to transition the database to MySQL. This part of the project was particularly insightful. Because direct export from Access to MySQL is not natively supported, I manually recreated the schema using CREATE TABLE scripts in MySQL Workbench. This process involved translating Access data types to MySQL-compatible ones, such as converting AutoNumber fields to INT AUTO\_INCREMENT. While this required precision and attention to syntax, it was an excellent exercise in understanding cross-platform database architecture and migration.

Several challenges emerged throughout the project. Setting up foreign keys correctly in Access, especially without the ability to declare them directly in table definitions, required patience and experimentation. Additionally, the lack of a seamless export path to MySQL meant I had to be meticulous in manually replicating the database structure. Matching data types between Access and MySQL also posed difficulties, as small mismatches could lead to import errors or integrity issues. Despite these hurdles, the BloomCart project was a valuable opportunity to apply theoretical knowledge in a practical setting. It enhanced my understanding of relational database design, SQL scripting, and platform interoperability. Most importantly, it reinforced the importance of planning, consistency, and testing when developing systems intended to support real business functions. Through this project, I not only refined my technical skills but also deepened my appreciation for the role of databases in modern commerce.