ANKARA BİLİM ÜNİVERSİTESİ

(CENG-351) DATABASE MANAGEMENT SYSTEMS

PROJECT: Flowershop Database

Group Members:

- Fevzi Samed Ünal – 220205032
- Hamit Güneş – 220204042
- Anıl Güngör - 220205003

Phase 1 – Database Design

1. Topic Selection

This project focuses on designing and implementing a relational database system for a flower shop that operates through both physical stores and online sales channels. The system manages customers, flowers, orders, inventory, suppliers, payments, and delivery processes in an integrated and structured manner.

Modern flower shops handle a wide variety of operational data, including customer information, product details, stock levels, order records, supplier entries, payment transactions, and delivery schedules. Managing this data manually or across separate platforms often leads to inconsistencies, errors, and operational inefficiencies. This project aims to address these challenges by providing a centralized database solution.

2. Problem Analysis

Flower shops typically receive orders from multiple channels such as in-store purchases, phone calls, websites, and online messaging platforms. When these orders, stock movements, and delivery processes are handled using manual methods, spreadsheets, or non-integrated systems, several critical problems arise.

2.1 Inconsistency and Data Integrity Issues

When customer, order, and product information is stored across different files or systems, data duplication and inconsistency become common.

Examples:

- A customer’s contact or address information may differ across records
- Stock quantities may not reflect real-time sales
- Orders may reference outdated or incorrect flower information

These inconsistencies can result in incorrect deliveries, billing errors, and loss of customer trust.

2.2 Stock Management and Availability Problems

Without a centralized stock tracking system:

- Flower inventory may not be updated automatically after orders
- Out-of-stock products may still be sold
- Supplier deliveries may not be recorded accurately

This can lead to delays in order fulfillment and customer dissatisfaction, especially for time-sensitive deliveries such as events and special occasions.

2.3 Order Processing and Delivery Tracking Issues

Managing delivery details manually increases the risk of:

- Missing or incorrect delivery addresses
- Unclear delivery time slots
- Inability to track delivery status (prepared, en route, delivered)

Lack of proper delivery tracking reduces operational efficiency and affects service quality.

2.4 Payment Tracking and Financial Reporting Problems

Flower shops require accurate financial tracking to monitor:

- Payment status of orders
- Different payment methods (cash, credit card, etc.)
- Total revenue and order history

When payments are recorded manually, generating reliable financial reports becomes time-consuming and error-prone.

2.5 Supplier and Product Entry Challenges

Suppliers play a critical role in maintaining stock availability. Without a proper system:

- Supplier-product relationships are difficult to track
- Incoming stock entries may be overlooked
- Inventory updates may be delayed

This affects both operational planning and cost management.

2.6 Purpose of the Project

The main purpose of this project is to design a centralized and reliable database system that enables the flower shop to:

- Manage customer information and multiple addresses efficiently
- Track flowers and categories in a structured manner
- Monitor stock levels automatically
- Process orders from both physical and online channels
- Record payments and delivery details accurately
- Track supplier-product relationships

The system aims to ensure data integrity, reduce redundancy, and improve overall operational efficiency.

2.7 Project Scope

The system will cover:

- Customer and address management
- Flower and category management
- Inventory and stock tracking
- Order creation and order item management
- Payment and delivery tracking
- Supplier and supplier-flower relationships
- Basic reporting and order history analysis

The system will not include:

- Online payment gateway integration
- Mobile application development
- Advanced ERP or logistics integrations

3. System Users

User	Explain
Admin	Defines product categories, manages pricing policies, and oversees overall data consistency across the system.
Staff	Creates orders, records stock entries for incoming products, and updates inventory levels.
Customer	Manages personal information and multiple delivery addresses; places both physical and online orders.
Supplier	Represents external stakeholders who provide flowers and products, allowing the system to track supply processes
Courier *(optional)*	Updates the delivery status (e.g., Picked Up, En Route, Delivered) of assigned orders in real-time.

4. Functional Requirements

- Customers must be able to be registered in the system
- Customers must be able to have multiple addresses
- Flowers and bouquets must be defined based on categories
- The stock quantity of each flower must be maintained
- Stock levels should be automatically updated based on orders
- Customers should be able to select more than one flower/bouquet in an order
- Orders should be distinguishable between physical store and online
- Orders should be marked as delivered to address or collected from store
- Address, delivery date and time slot should be recorded for delivery
- The delivery status of orders (being prepared, en route, delivered, etc.) should be trackable
- Payment details for orders should be recorded
- An order should be able to contain one or more payments (optional)
- Product entries from suppliers should be trackable

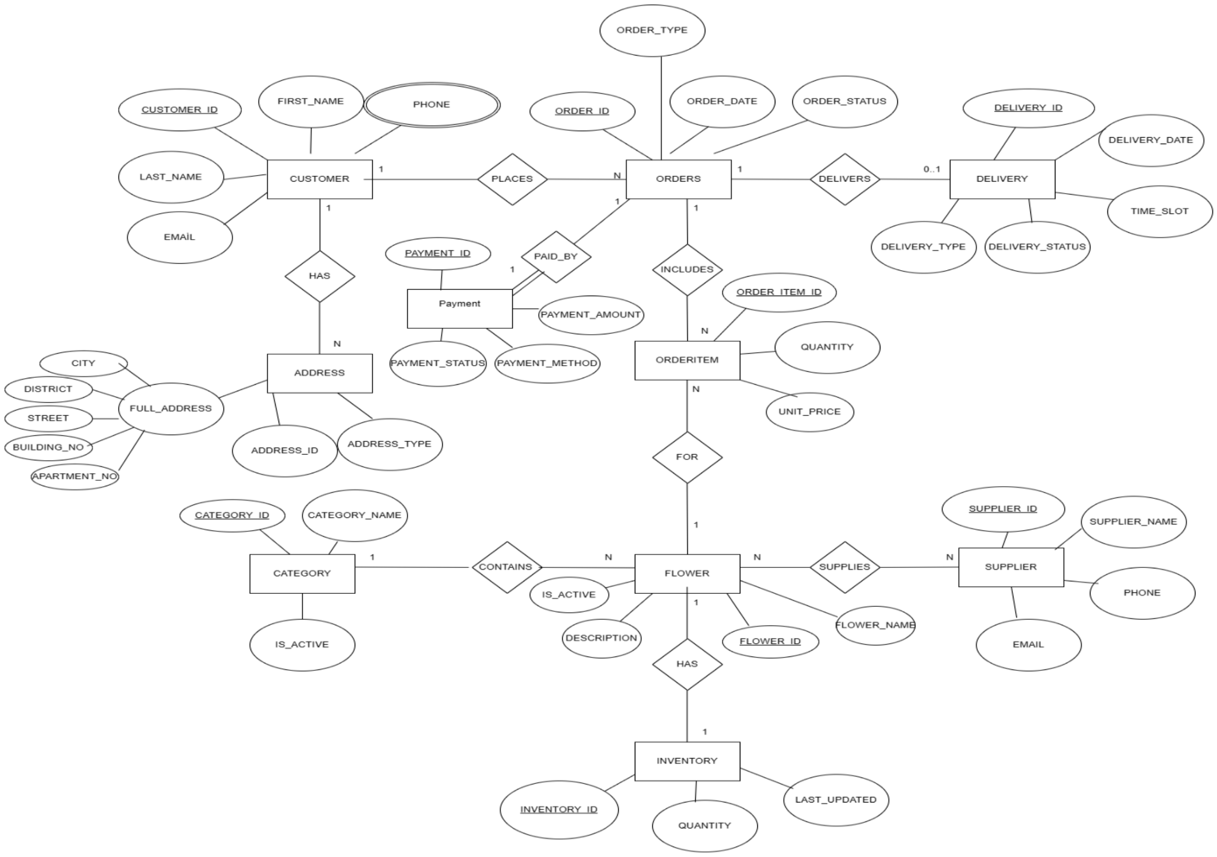
5. Non-Functional Requirements

- Data integrity must be ensured through Primary Key and Foreign Key constraints
- Tables must comply with at least Third Normal Form (3NF)
- Data redundancy must be minimised
- The system must be scalable (new products and payment types must be able to be added)
- Delivery and payment details must be protected with consistent and mandatory fields

6. Basic Data Types to be Stored

- Customer information
- Address information
- Flower and bouquet information
- Category information
- Stock information
- Order information
- Order details
- Payment information
- Delivery information
- Supplier information

Conceptual Data Model



LOGICAL DATA MODEL

Data / Information Requirements

Entity	Stored Attributes
Customer	CustomerID (PK), FirstName, LastName, Email
CustomerPhone	PhoneID (PK), Phone, CustomerID (FK)
Address	AddressID (PK), City, District, Street, BuildingNo, ApartmentNo, AddressType, CustomerID (FK)
Orders	OrderID (PK), OrderDate, OrderStatus, OrderType, CustomerID (FK)
Payment	PaymentID (PK), PaymentMethod, PaymentStatus, PaymentAmount, OrderID (FK)
Delivery	DeliveryID (PK), DeliveryDate, TimeSlot, DeliveryType, DeliveryStatus, OrderID (FK)
OrderItem	OrderItemID (PK), Quantity, UnitPrice, OrderID (FK), FlowerID (FK)
Category	CategoryID (PK), CategoryName, IsActive
Flower	FlowerID (PK), FlowerName, Description, UnitPrice, IsActive, CategoryID (FK)
Inventory	InventoryID (PK), Quantity, LastUpdated, FlowerID (FK)
Supplier	SupplierID (PK), SupplierName, Phone, Email
SupplierFlower	SupplierID (PK, FK), FlowerID (PK, FK)

Normalization Analysis

Instead of restructuring existing tables, the design was built "normalization-first" to prevent insertion, deletion, and update anomalies from the outset. The following systematic checks confirm that the schema satisfies 3NF requirements:

First Normal Form (1NF) Compliance: All attributes across all tables (such as Customer, Flower, and Orders) contain only atomic values. For instance, multi-valued attributes like customer phone numbers were moved to the CustomerPhone table to eliminate repeating groups.

Second Normal Form (2NF) Compliance: Each table satisfies 1NF, and every non-key attribute is fully functionally dependent on the entire primary key. By using single-column surrogate keys (e.g., FlowerID, OrderID), partial dependencies were structurally impossible to occur.

Third Normal Form (3NF) Compliance: The schema ensures that no transitive dependencies exist; every non-key attribute depends "only on the key". For example, Category details were separated from the Flower table to ensure that category names are determined solely by the CategoryID, not by a FlowerID.

The analysis confirms that the use of associative tables (e.g., Order_Item, Supplier_Flower) effectively resolved complex relationships while maintaining 3NF standards. Consequently, the database provides a streamlined architecture that minimizes redundancy and ensures reliable transaction handling

Detailed normalization checks for each table are provided below:

Detailed Normalization Checks

1. Customer Table

1NF:
All attributes (FirstName, LastName, Email) contain atomic values.

2NF:
CustomerID is a single-column primary key; therefore, no partial dependency exists.

3NF:
All non-key attributes depend directly on CustomerID. No transitive dependency exists.

2. CustomerPhone Table

1NF:
Phone values are atomic; no repeating groups exist.

2NF:
PhoneID is a single primary key.

3NF:
Phone depends only on PhoneID. CustomerID is a foreign key and does not create transitive dependency.

3. Address Table

1NF:
All address fields are atomic (City, Street, ApartmentNo, etc.).

2NF:
AddressID is a single-column primary key.

3NF:
All attributes describe the address entity itself and depend only on AddressID.

4. Orders Table

1NF:
OrderDate, OrderStatus, and OrderType are atomic.

2NF:
OrderID is a single primary key.

3NF:
All non-key attributes depend directly on OrderID. CustomerID is a foreign key.

5. Payment Table

1NF:
PaymentAmount and PaymentMethod contain atomic values.

2NF:
PaymentID is the primary key; no partial dependency exists.

3NF:
All attributes describe the payment transaction itself. OrderID is a foreign key and does not cause transitive dependency.

6. Delivery Table

1NF:
DeliveryDate and TimeSlot are atomic values.

2NF:
DeliveryID is a single-column primary key.

3NF:
Delivery attributes depend only on DeliveryID. OrderID is a reference attribute.

7. OrderItem Table

1NF:
Quantity and UnitPrice are atomic.

2NF:
OrderItemID is a surrogate primary key.

3NF:
Attributes describe the order-item relationship directly. OrderID and FlowerID are foreign keys.

8. Category Table

1NF:
CategoryName and IsActive contain atomic values.

2NF:
CategoryID is the primary key.

3NF:
No non-key attribute depends on another non-key attribute.

9. Flower Table

1NF:
FlowerName, Description, and UnitPrice are atomic.

2NF:
FlowerID is the primary key.

3NF:
All attributes describe the flower entity. CategoryID is a foreign key.

10. Inventory Table

1NF:
Quantity and LastUpdated are atomic.

2NF:
InventoryID is the primary key.

3NF:
Inventory attributes depend only on InventoryID.

11. Supplier Table

1NF:
SupplierName, Phone, and Email are atomic.

2NF:
SupplierID is the primary key.

3NF:
No transitive dependency exists.

12. SupplierFlower Table

1NF:
Each record represents a single supplier–flower relationship.

2NF:
Composite primary key (SupplierID, FlowerID) is used correctly.

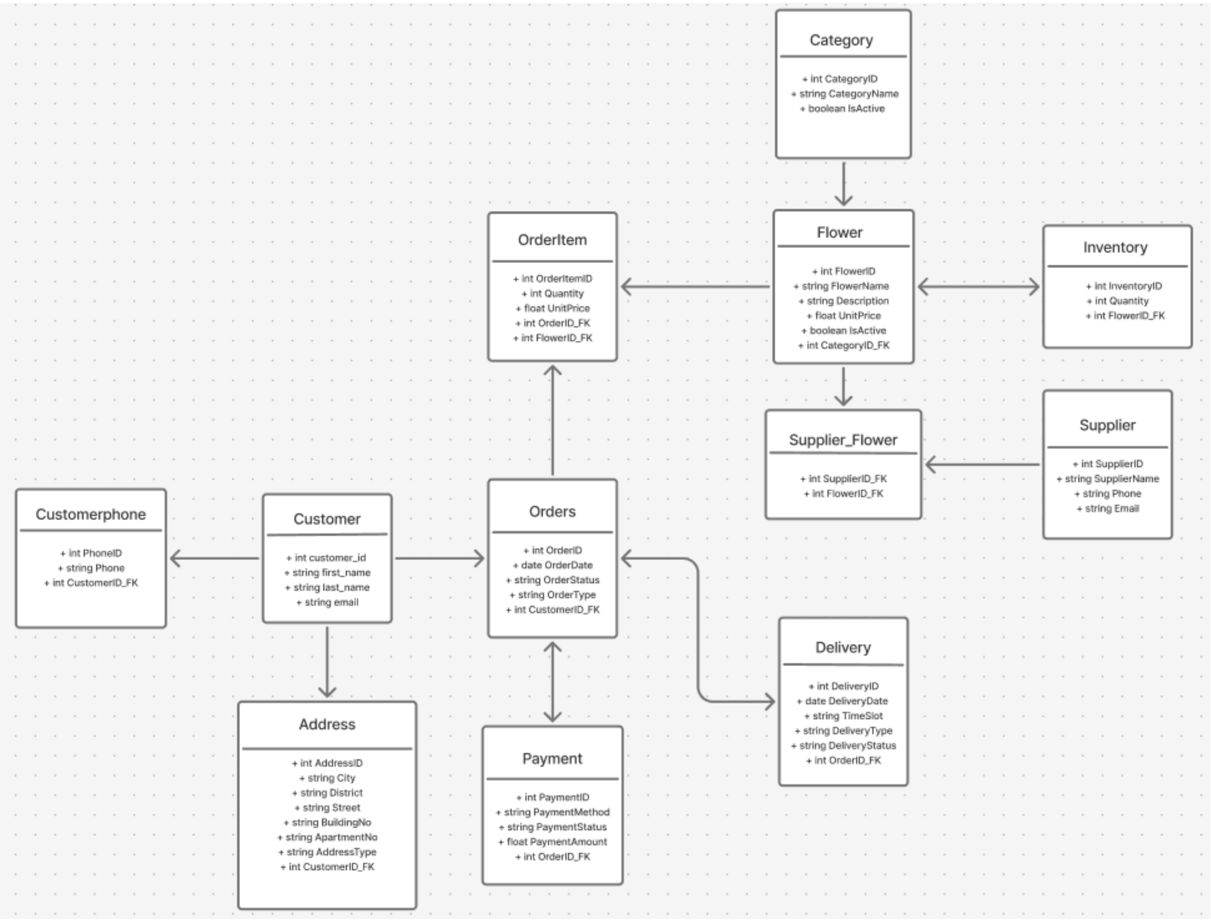
3NF:
No non-key attributes exist; therefore, the table is fully normalized.

Conclusion

The Florist Order, Delivery and Stock Management System has been successfully designed and implemented to provide a robust digital solution to the operational challenges faced by modern floristry businesses. By using surrogate keys (e.g. Customer ID, Order ID, Payment ID) and resolving all many-to-many (M:N) relationships through relational tables such as Order_Item and Supplier_Flower, the database schema complies with Third Normal Form (3NF) requirements. This normalised structure effectively eliminates data redundancy and prevents insertion, update, and deletion anomalies.

The implementation phase carried out in Microsoft SQL Server has confirmed that the physical model is consistent with the initial conceptual design. As a result, the system serves not only as a secure data repository but also as an efficient management tool that can seamlessly support both physical and online sales channels.

Physical Data Model

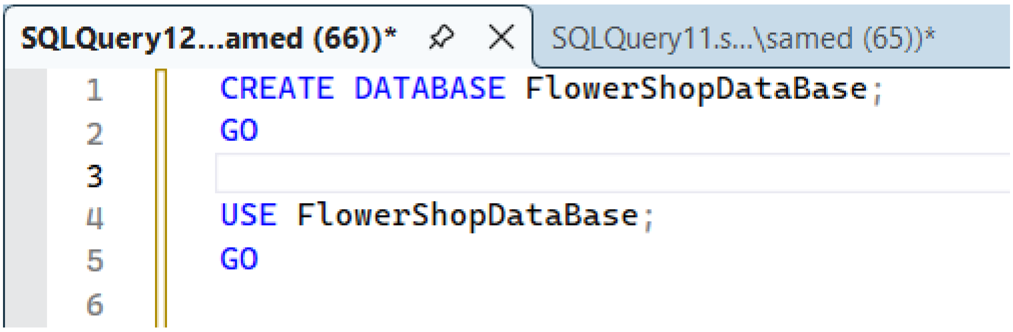


Phase 2 – Database Implementation

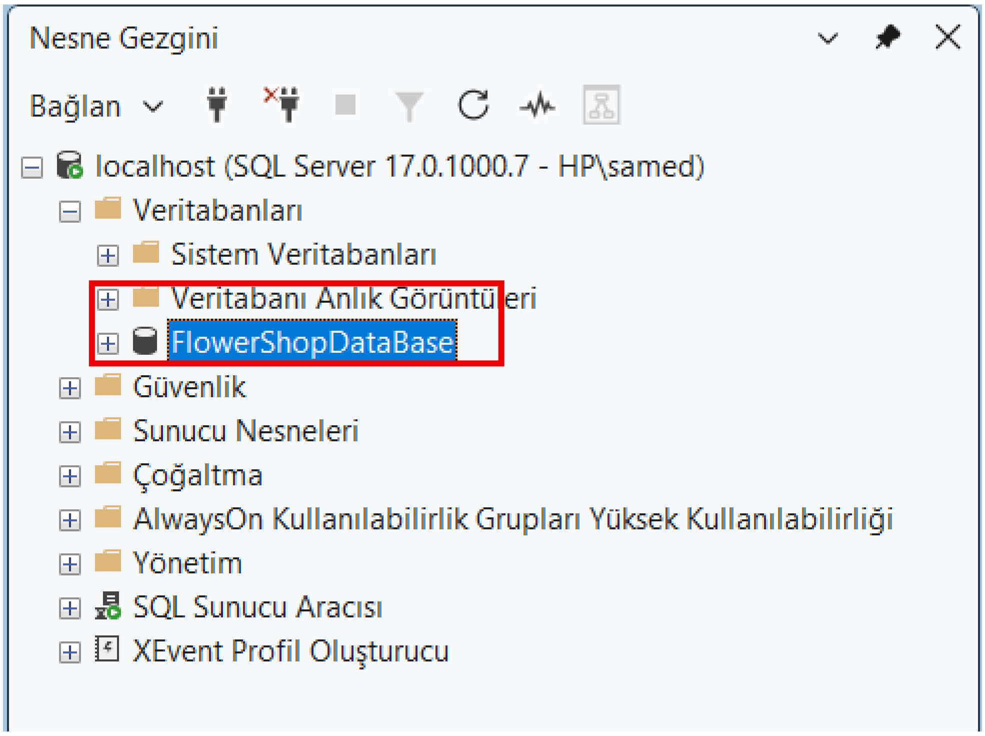
In this phase, the physical data model designed in Phase 1 was implemented using Microsoft SQL Server. The database schema, tables, and relationships were created via SQL scripts, and the tables were populated with sample data to verify the system's functionality.

1.Database Creation

The following SQL statement was executed to create the FlowerShopDB database



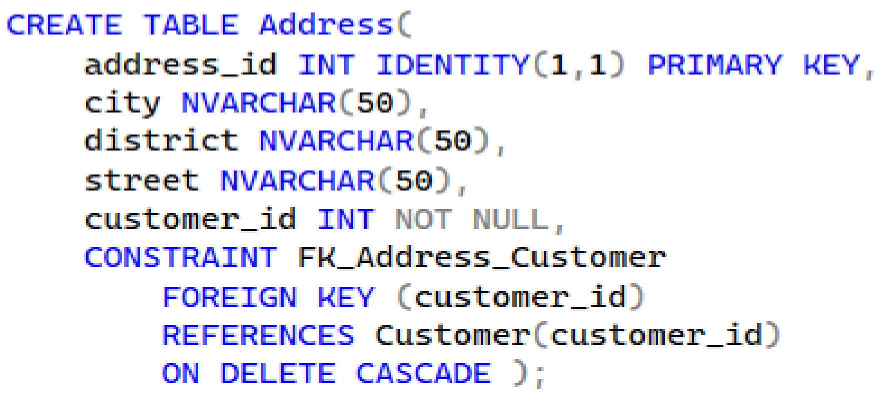
```sql
CREATE DATABASE FlowerShopDataBase;
GO

USE FlowerShopDataBase;
GO

```



localhost (SQL Server 17.0.1000.7 - HP\samed)
Veritabanları
Sistem Veritabanları
Veritabanı Anlık Görüntüleri
FlowerShopDataBase
Güvenlik
Sunucu Nesneleri
Çoğaltma
AlwaysOn Kullanılabilirlik Grupları Yüksek Kullanılabilirliği
Yönetim
SQL Sunucu Aracısı
XEvent Profil Oluşturucu

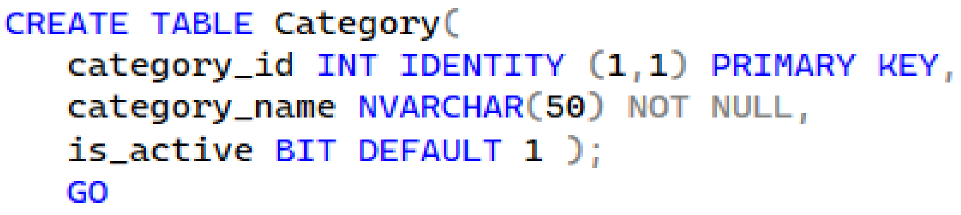
2.Table Creation

1) Address



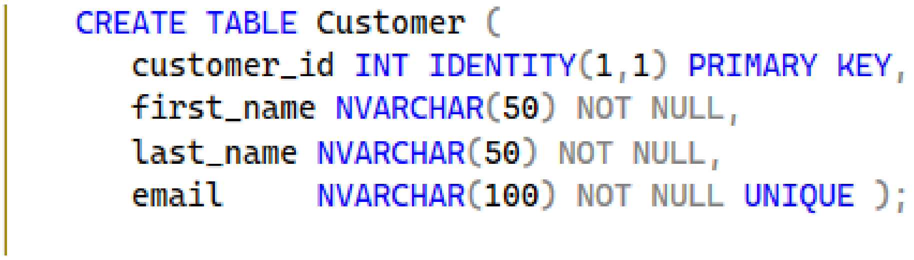
```sql
CREATE TABLE Address(
    address_id INT IDENTITY(1,1) PRIMARY KEY,
    city NVARCHAR(50),
    district NVARCHAR(50),
    street NVARCHAR(50),
    customer_id INT NOT NULL,
    CONSTRAINT FK_Address_Customer
        FOREIGN KEY (customer_id)
        REFERENCES Customer(customer_id)
        ON DELETE CASCADE );
```

2)Category



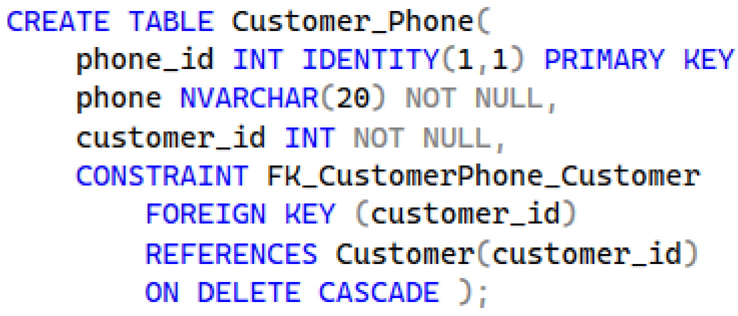
```sql
CREATE TABLE Category(
    category_id INT IDENTITY (1,1) PRIMARY KEY,
    category_name NVARCHAR(50) NOT NULL,
    is_active BIT DEFAULT 1 );
    GO
```

3)Customer



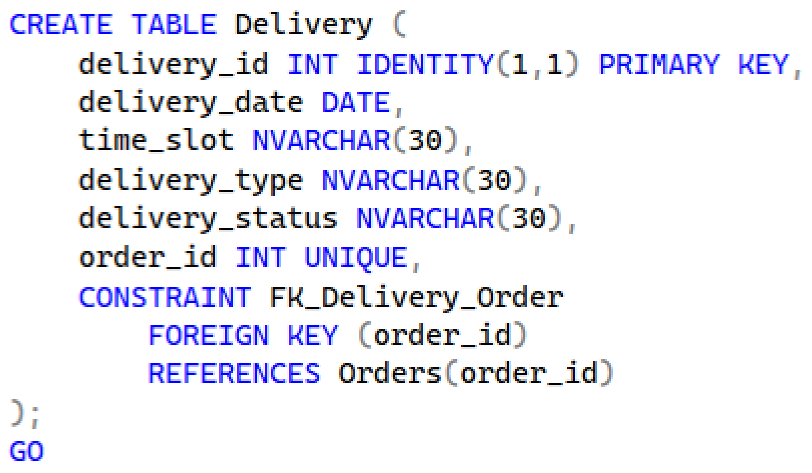
```sql
CREATE TABLE Customer (
    customer_id INT IDENTITY(1,1) PRIMARY KEY,
    first_name NVARCHAR(50) NOT NULL,
    last_name NVARCHAR(50) NOT NULL,
    email     NVARCHAR(100) NOT NULL UNIQUE );
```

4) Customer Phone



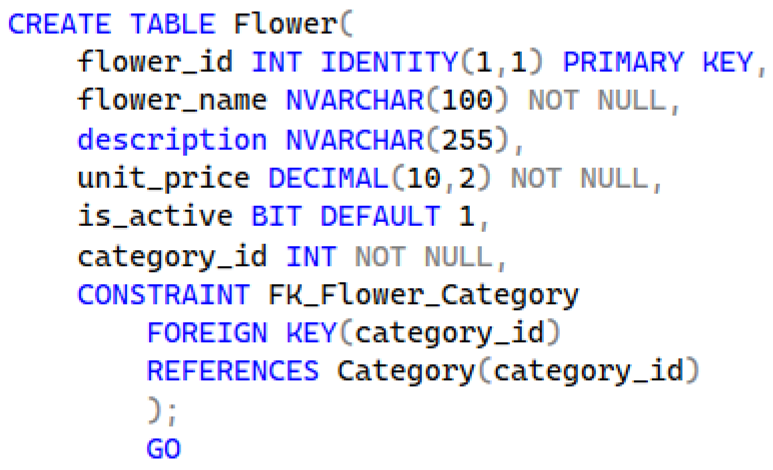
```sql
CREATE TABLE Customer_Phone(
    phone_id INT IDENTITY(1,1) PRIMARY KEY,
    phone NVARCHAR(20) NOT NULL,
    customer_id INT NOT NULL,
    CONSTRAINT FK_CustomerPhone_Customer
        FOREIGN KEY (customer_id)
        REFERENCES Customer(customer_id)
        ON DELETE CASCADE );
```

5)Delivery



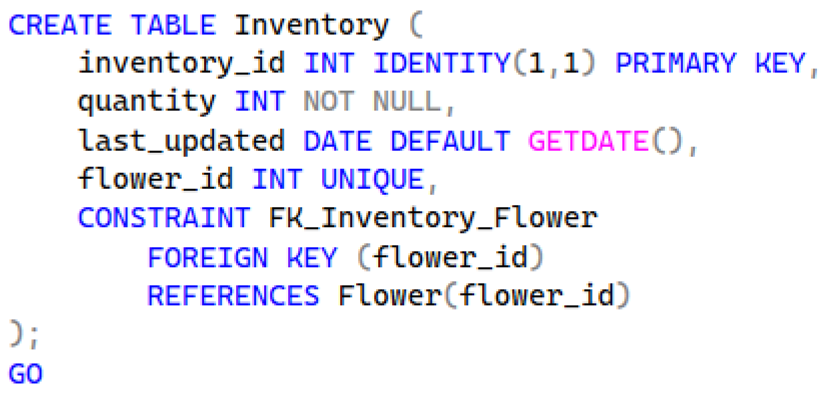
```
CREATE TABLE Delivery (
    delivery_id INT IDENTITY(1,1) PRIMARY KEY,
    delivery_date DATE,
    time_slot NVARCHAR(30),
    delivery_type NVARCHAR(30),
    delivery_status NVARCHAR(30),
    order_id INT UNIQUE,
    CONSTRAINT FK_Delivery_Order
        FOREIGN KEY (order_id)
        REFERENCES Orders(order_id)
);
GO
```

6)Flower



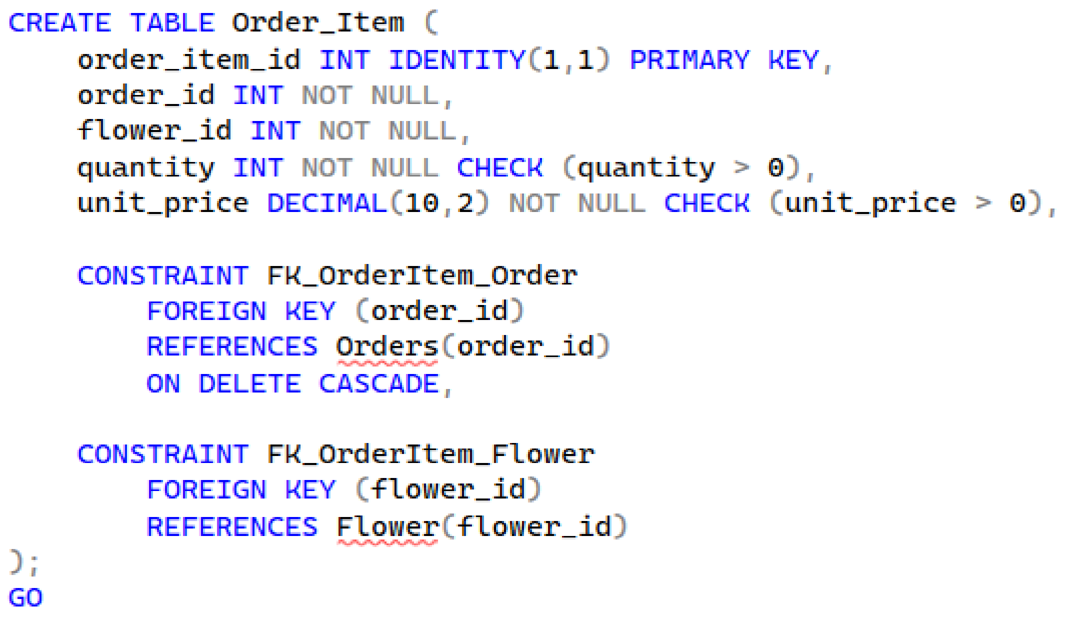
```
CREATE TABLE Flower(
    flower_id INT IDENTITY(1,1) PRIMARY KEY,
    flower_name NVARCHAR(100) NOT NULL,
    description NVARCHAR(255),
    unit_price DECIMAL(10,2) NOT NULL,
    is_active BIT DEFAULT 1,
    category_id INT NOT NULL,
    CONSTRAINT FK_Flower_Category
        FOREIGN KEY(category_id)
        REFERENCES Category(category_id)
        );
        GO
```

7)Inventory



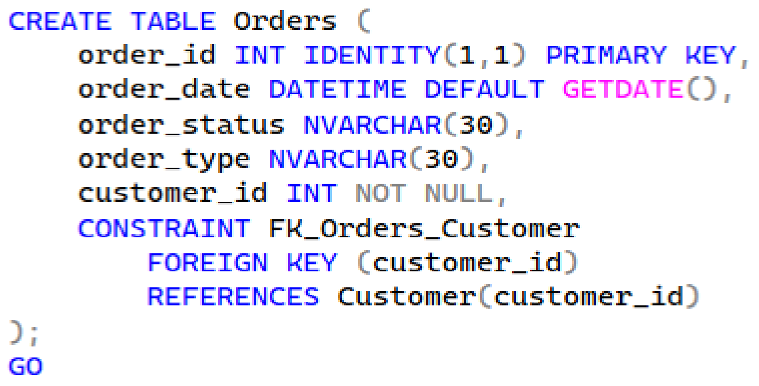
```
CREATE TABLE Inventory (
    inventory_id INT IDENTITY(1,1) PRIMARY KEY,
    quantity INT NOT NULL,
    last_updated DATE DEFAULT GETDATE(),
    flower_id INT UNIQUE,
    CONSTRAINT FK_Inventory_Flower
        FOREIGN KEY (flower_id)
        REFERENCES Flower(flower_id)
);
GO
```

8)Order Item



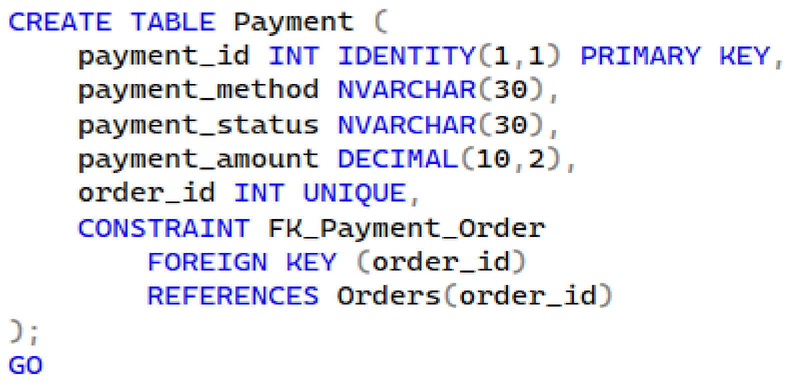
```
CREATE TABLE Order_Item (
    order_item_id INT IDENTITY(1,1) PRIMARY KEY,
    order_id INT NOT NULL,
    flower_id INT NOT NULL,
    quantity INT NOT NULL CHECK (quantity > 0),
    unit_price DECIMAL(10,2) NOT NULL CHECK (unit_price > 0),

    CONSTRAINT FK_OrderItem_Order
        FOREIGN KEY (order_id)
        REFERENCES Orders(order_id)
        ON DELETE CASCADE,

    CONSTRAINT FK_OrderItem_Flower
        FOREIGN KEY (flower_id)
        REFERENCES Flower(flower_id)
);
GO
```

9)Orders



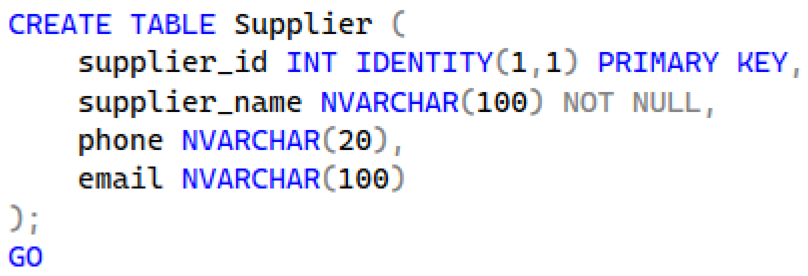
```
CREATE TABLE Orders (
    order_id INT IDENTITY(1,1) PRIMARY KEY,
    order_date DATETIME DEFAULT GETDATE(),
    order_status NVARCHAR(30),
    order_type NVARCHAR(30),
    customer_id INT NOT NULL,
    CONSTRAINT FK_Orders_Customer
        FOREIGN KEY (customer_id)
        REFERENCES Customer(customer_id)
);
GO
```

10)Payment



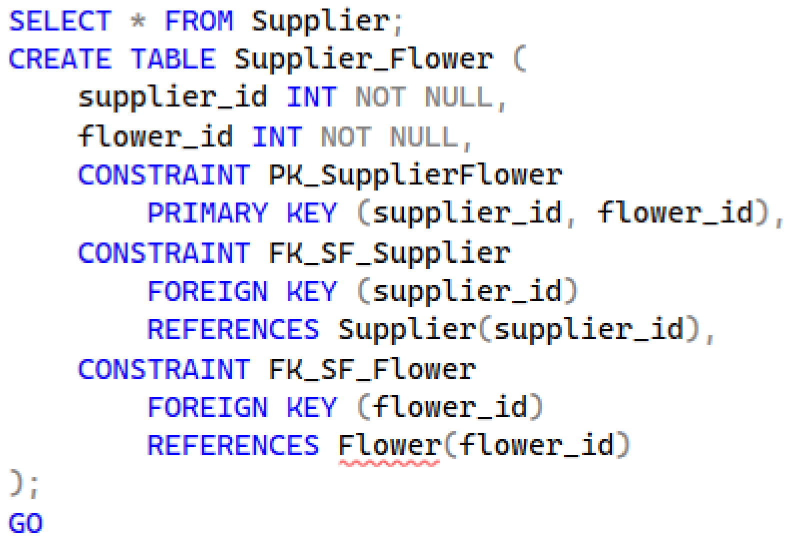
```
CREATE TABLE Payment (
    payment_id INT IDENTITY(1,1) PRIMARY KEY,
    payment_method NVARCHAR(30),
    payment_status NVARCHAR(30),
    payment_amount DECIMAL(10,2),
    order_id INT UNIQUE,
    CONSTRAINT FK_Payment_Order
        FOREIGN KEY (order_id)
        REFERENCES Orders(order_id)
);
GO
```

11)Supplier



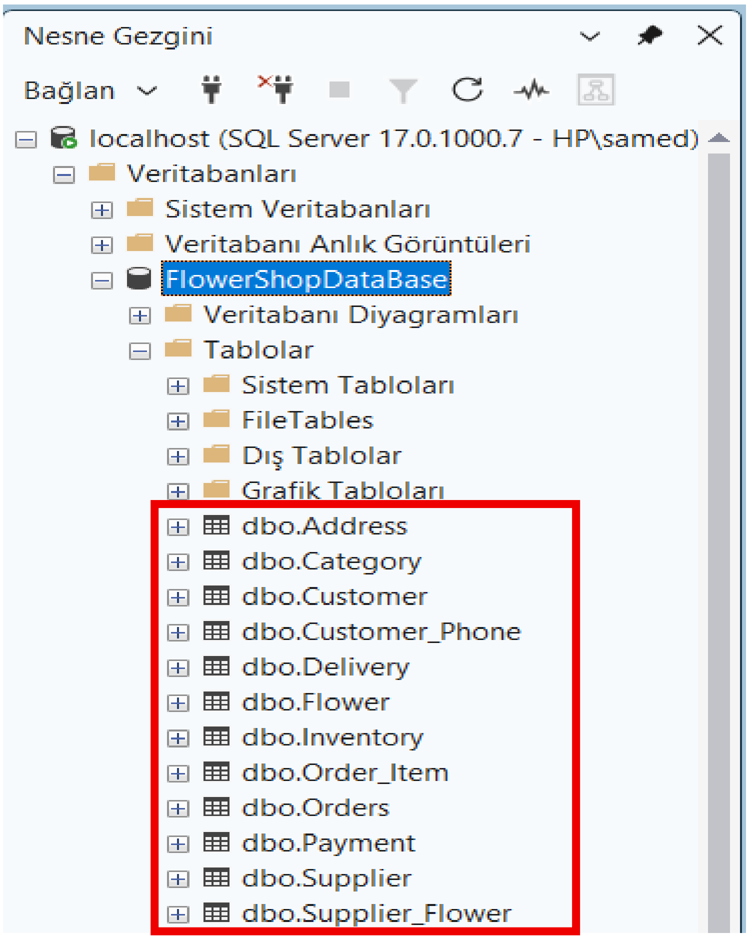
```
CREATE TABLE Supplier (
    supplier_id INT IDENTITY(1,1) PRIMARY KEY,
    supplier_name NVARCHAR(100) NOT NULL,
    phone NVARCHAR(20),
    email NVARCHAR(100)
);
GO
```

12)Supplier Flower



```
SELECT * FROM Supplier;
CREATE TABLE Supplier_Flower (
    supplier_id INT NOT NULL,
    flower_id INT NOT NULL,
    CONSTRAINT PK_SupplierFlower
        PRIMARY KEY (supplier_id, flower_id),
    CONSTRAINT FK_SF_Supplier
        FOREIGN KEY (supplier_id)
        REFERENCES Supplier(supplier_id),
    CONSTRAINT FK_SF_Flower
        FOREIGN KEY (flower_id)
        REFERENCES Flower(flower_id)
);
GO
```

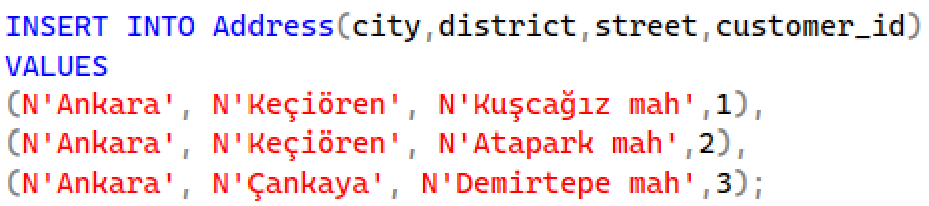
All Tables



- localhost (SQL Server 17.0.1000.7 - HP\samed)
  - Veritabanları
    - Sistem Veritabanları
    - Veritabanı Anlık Görüntüleri
    - FlowerShopDataBase
      - Veritabanı Diyagramları
      - Tablolar
        - Sistem Tabloları
        - FileTables
        - Dış Tablolar
        - Grafik Tabloları
        - dbo.Address
        - dbo.Category
        - dbo.Customer
        - dbo.Customer_Phone
        - dbo.Delivery
        - dbo.Flower
        - dbo.Inventory
        - dbo.Order_Item
        - dbo.Orders
        - dbo.Payment
        - dbo.Supplier
        - dbo.Supplier_Flower

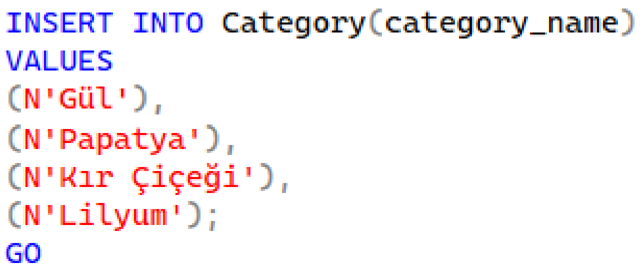
3. Data Insertion

1)Address



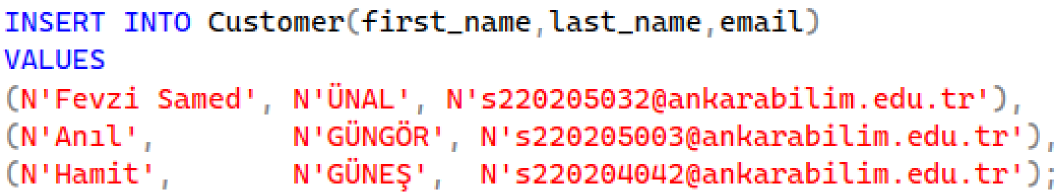
```
INSERT INTO Address(city,district,street,customer_id)
VALUES
(N'Ankara', N'Keçiören', N'Kuşcağız mah',1),
(N'Ankara', N'Keçiören', N'Atapark mah',2),
(N'Ankara', N'Çankaya', N'Demirtepe mah',3);
```

2)Category



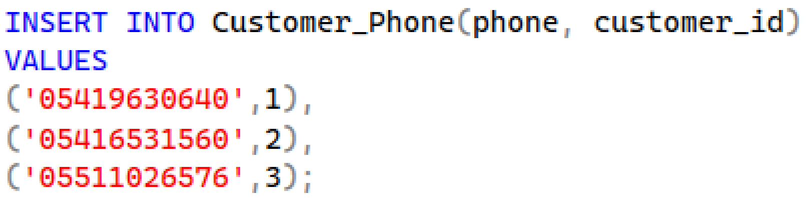
```
INSERT INTO Category(category_name)
VALUES
(N'Gül'),
(N'Papatya'),
(N'Kır Çiçeği'),
(N'Lilyum');
GO
```

3)Customer



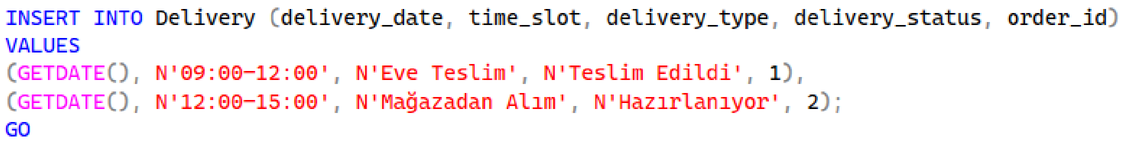
```
INSERT INTO Customer(first_name,last_name,email)
VALUES
(N'Fevzi Samed', N'ÜNAL', N's220205032@ankarabilim.edu.tr'),
(N'Anıl',        N'GÜNGÖR', N's220205003@ankarabilim.edu.tr'),
(N'Hamit',       N'GÜNEŞ',  N's220204042@ankarabilim.edu.tr');
```

4)Customer Phone



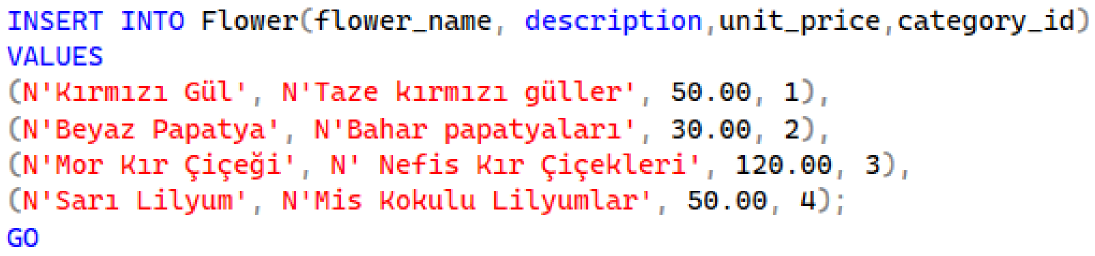
```
INSERT INTO Customer_Phone(phone, customer_id)
VALUES
('05419630640',1),
('05416531560',2),
('05511026576',3);
```

5)Delivery



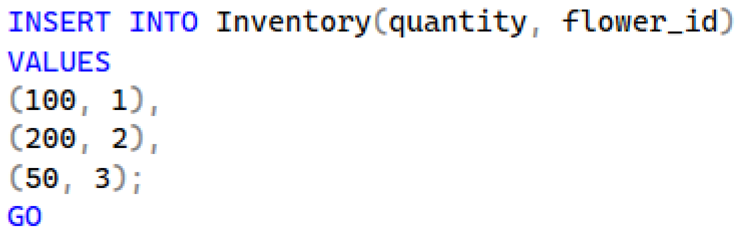
```
INSERT INTO Delivery (delivery_date, time_slot, delivery_type, delivery_status, order_id)
VALUES
(GETDATE(), N'09:00-12:00', N'Eve Teslim', N'Teslim Edildi', 1),
(GETDATE(), N'12:00-15:00', N'Mağazadan Alım', N'Hazırlanıyor', 2);
GO
```

6)Flower



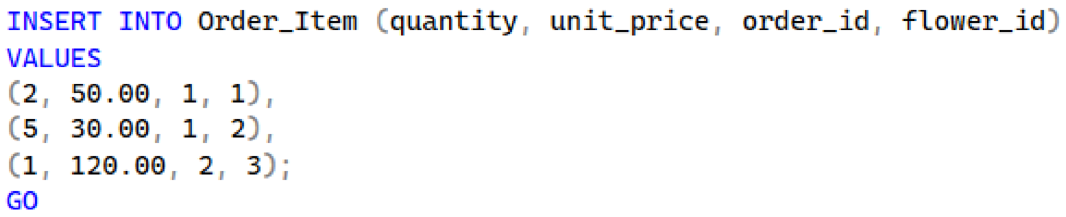
```
INSERT INTO Flower(flower_name, description,unit_price,category_id)
VALUES
(N'Kırmızı Gül', N'Taze kırmızı güller', 50.00, 1),
(N'Beyaz Papatya', N'Bahar papatyaları', 30.00, 2),
(N'Mor Kır Çiçeği', N' Nefis Kır Çiçekleri', 120.00, 3),
(N'Sarı Lilyum', N'Mis Kokulu Lilyumlar', 50.00, 4);
GO
```

7)Inventory



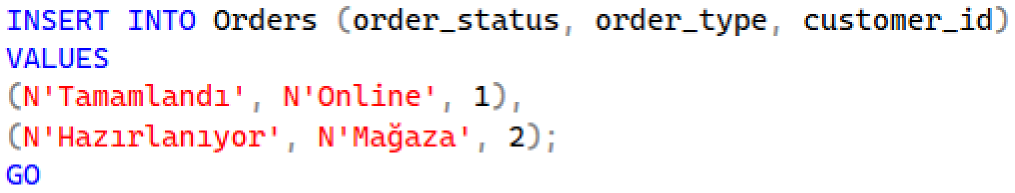
```
INSERT INTO Inventory(quantity, flower_id)
VALUES
(100, 1),
(200, 2),
(50, 3);
GO
```

8)Order Item



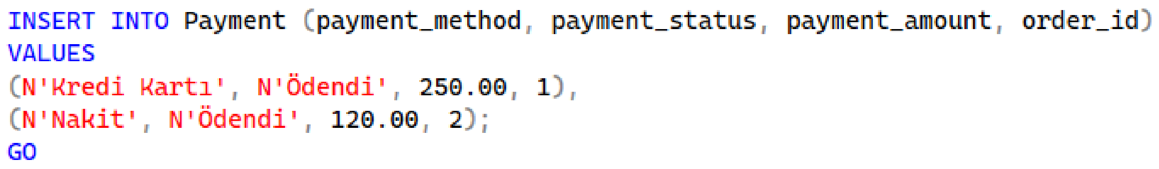
```
INSERT INTO Order_Item (quantity, unit_price, order_id, flower_id)
VALUES
(2, 50.00, 1, 1),
(5, 30.00, 1, 2),
(1, 120.00, 2, 3);
GO
```

9)Orders



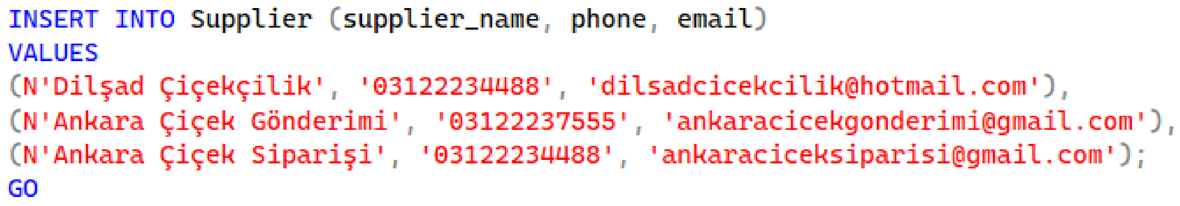
```
INSERT INTO Orders (order_status, order_type, customer_id)
VALUES
(N'Tamamlandı', N'Online', 1),
(N'Hazırlanıyor', N'Mağaza', 2);
GO
```

10)Payment



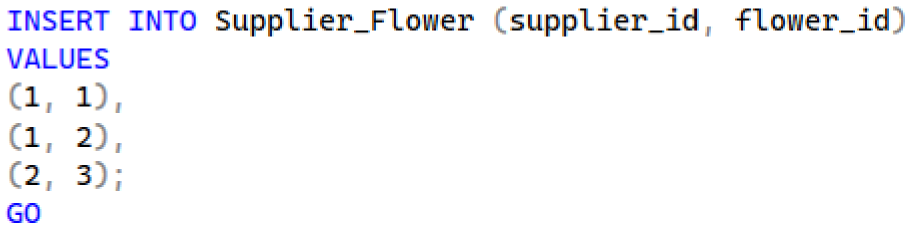
```
INSERT INTO Payment (payment_method, payment_status, payment_amount, order_id)
VALUES
(N'Kredi Kartı', N'Ödendi', 250.00, 1),
(N'Nakit', N'Ödendi', 120.00, 2);
GO
```

11)Supplier



```
INSERT INTO Supplier (supplier_name, phone, email)
VALUES
(N'Dilşad Çiçekçilik', '03122234488', 'dilsadcicekcilik@hotmail.com'),
(N'Ankara Çiçek Gönderimi', '03122237555', 'ankaracicekgonderimi@gmail.com'),
(N'Ankara Çiçek Siparişi', '03122234488', 'ankaraciceksiparisi@gmail.com');
GO
```

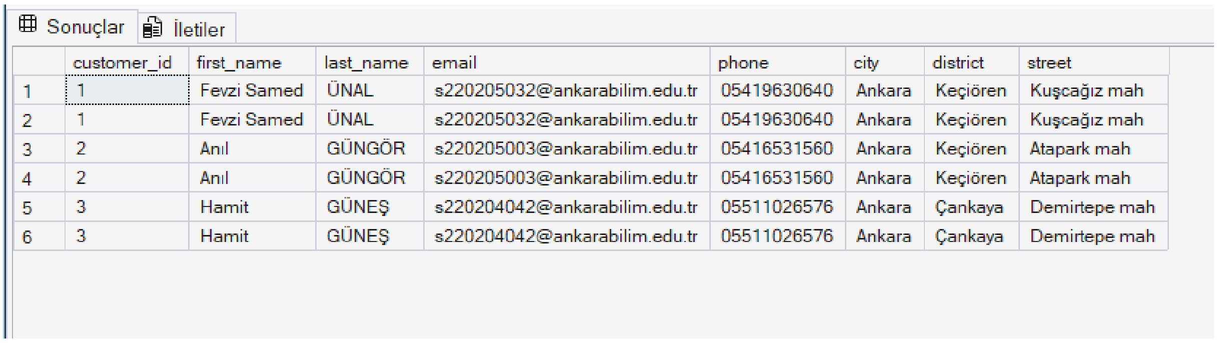
12)Supplier Flower



```
INSERT INTO Supplier_Flower (supplier_id, flower_id)
VALUES
(1, 1),
(1, 2),
(2, 3);
GO
```

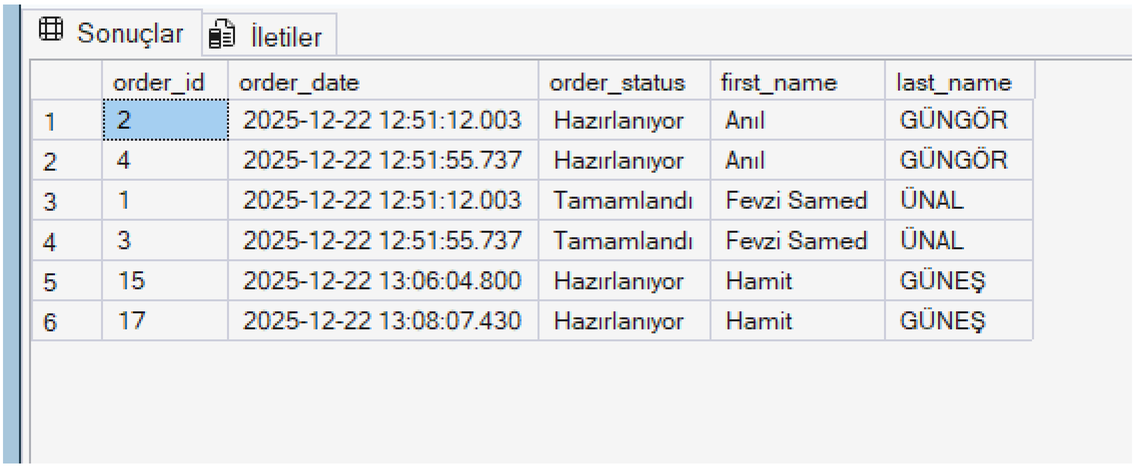
4. Successful Table Views

1) Customer Information



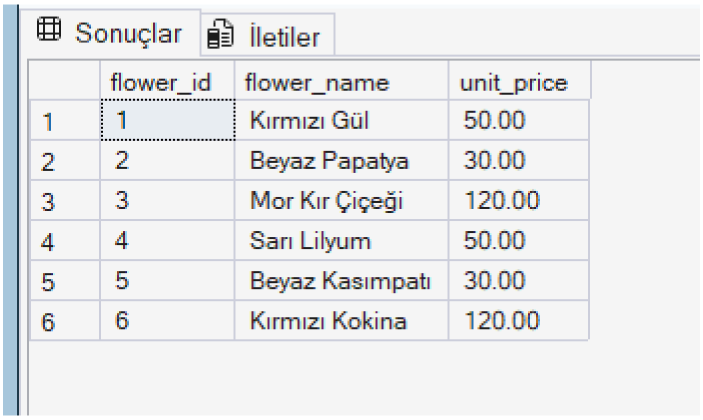
	customer_id	first_name	last_name	email	phone	city	district	street
1	1	Fevzi Samed	ÜNAL	s220205032@ankarabilim.edu.tr	05419630640	Ankara	Keçiören	Kuşcağız mah
2	1	Fevzi Samed	ÜNAL	s220205032@ankarabilim.edu.tr	05419630640	Ankara	Keçiören	Kuşcağız mah
3	2	Anıl	GÜNGÖR	s220205003@ankarabilim.edu.tr	05416531560	Ankara	Keçiören	Atapark mah
4	2	Anıl	GÜNGÖR	s220205003@ankarabilim.edu.tr	05416531560	Ankara	Keçiören	Atapark mah
5	3	Hamit	GÜNEŞ	s220204042@ankarabilim.edu.tr	05511026576	Ankara	Çankaya	Demirtepe mah
6	3	Hamit	GÜNEŞ	s220204042@ankarabilim.edu.tr	05511026576	Ankara	Çankaya	Demirtepe mah

2)ORDER RECORDS



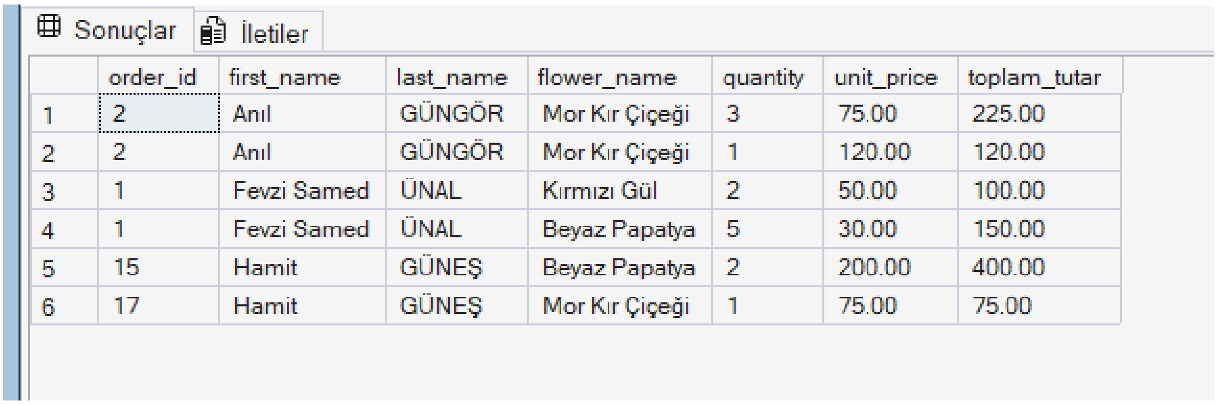
	order_id	order_date	order_status	first_name	last_name
1	2	2025-12-22 12:51:12.003	Hazırlanıyor	Anıl	GÜNGÖR
2	4	2025-12-22 12:51:55.737	Hazırlanıyor	Anıl	GÜNGÖR
3	1	2025-12-22 12:51:12.003	Tamamlandı	Fevzi Samed	ÜNAL
4	3	2025-12-22 12:51:55.737	Tamamlandı	Fevzi Samed	ÜNAL
5	15	2025-12-22 13:06:04.800	Hazırlanıyor	Hamit	GÜNEŞ
6	17	2025-12-22 13:08:07.430	Hazırlanıyor	Hamit	GÜNEŞ

3)FLOWER INFORMATION



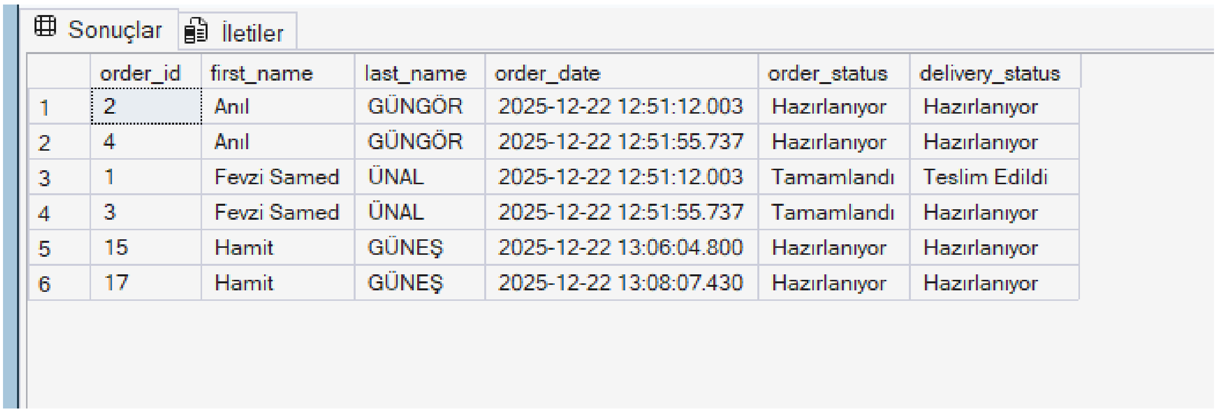
	flower_id	flower_name	unit_price
1	1	Kırmızı Gül	50.00
2	2	Beyaz Papatya	30.00
3	3	Mor Kır Çiçeği	120.00
4	4	Sarı Lilyum	50.00
5	5	Beyaz Kasımpatı	30.00
6	6	Kırmızı Kokina	120.00

4) ORDER HISTORY



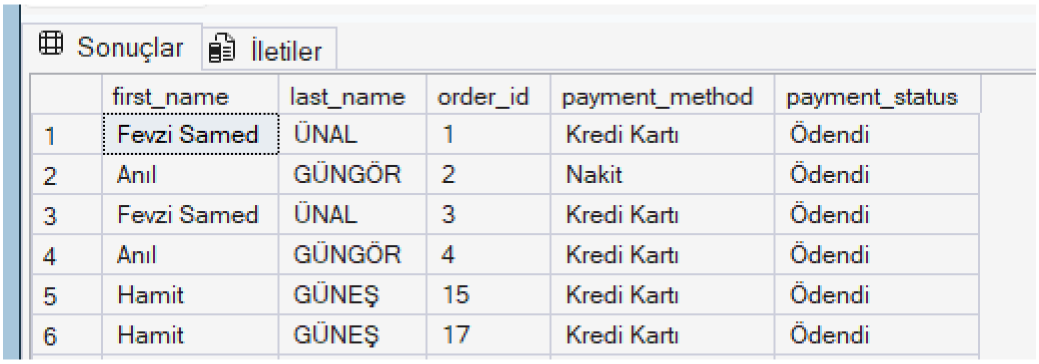
	order_id	first_name	last_name	flower_name	quantity	unit_price	toplam_tutar
1	2	Anıl	GÜNGÖR	Mor Kır Çiçeği	3	75.00	225.00
2	2	Anıl	GÜNGÖR	Mor Kır Çiçeği	1	120.00	120.00
3	1	Fevzi Samed	ÜNAL	Kırmızı Gül	2	50.00	100.00
4	1	Fevzi Samed	ÜNAL	Beyaz Papatya	5	30.00	150.00
5	15	Hamit	GÜNEŞ	Beyaz Papatya	2	200.00	400.00
6	17	Hamit	GÜNEŞ	Mor Kır Çiçeği	1	75.00	75.00

5) ORDER TIME & DELIVERY



	order_id	first_name	last_name	order_date	order_status	delivery_status
1	2	Anıl	GÜNGÖR	2025-12-22 12:51:12.003	Hazırlanıyor	Hazırlanıyor
2	4	Anıl	GÜNGÖR	2025-12-22 12:51:55.737	Hazırlanıyor	Hazırlanıyor
3	1	Fevzi Samed	ÜNAL	2025-12-22 12:51:12.003	Tamamlandı	Teslim Edildi
4	3	Fevzi Samed	ÜNAL	2025-12-22 12:51:55.737	Tamamlandı	Hazırlanıyor
5	15	Hamit	GÜNEŞ	2025-12-22 13:06:04.800	Hazırlanıyor	Hazırlanıyor
6	17	Hamit	GÜNEŞ	2025-12-22 13:08:07.430	Hazırlanıyor	Hazırlanıyor

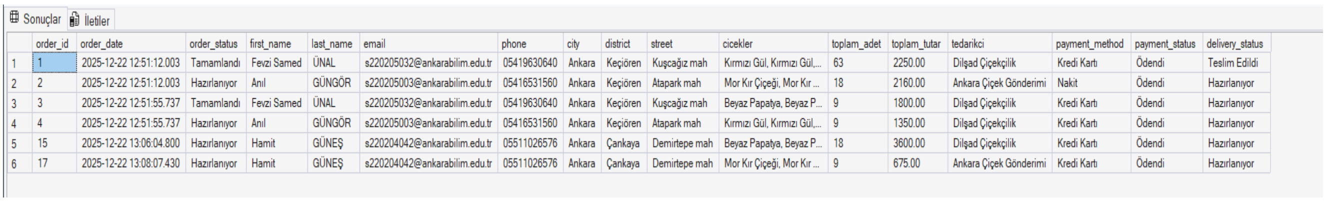
6) PAYMENT HISTORY



	first_name	last_name	order_id	payment_method	payment_status
1	Fevzi Samed	ÜNAL	1	Kredi Kartı	Ödendi
2	Anıl	GÜNGÖR	2	Nakit	Ödendi
3	Fevzi Samed	ÜNAL	3	Kredi Kartı	Ödendi
4	Anıl	GÜNGÖR	4	Kredi Kartı	Ödendi
5	Hamit	GÜNEŞ	15	Kredi Kartı	Ödendi
6	Hamit	GÜNEŞ	17	Kredi Kartı	Ödendi

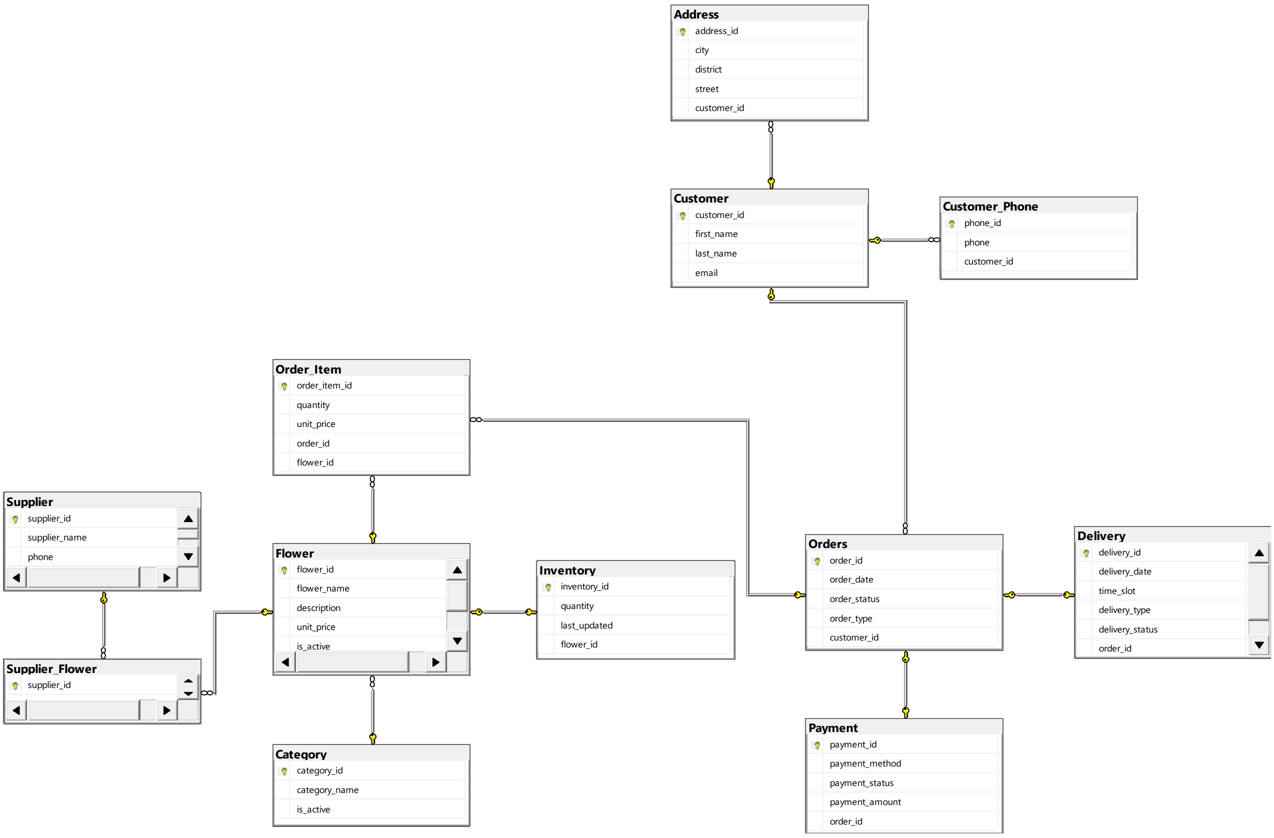
7) Total Successful

All order payment and delivery processes have been linked to the Payment and Delivery tables, and missing records have been completed.



	order_id	order_date	order_status	first_name	last_name	email	phone	city	district	street	cicekler	toplam_adet	toplam_tutar	tedarikci	payment_method	payment_status	delivery_status
1	1	2025-12-22 12:51:12.003	Tamamlandı	Fevzi Samed	ÜNAL	s220205032@ankarabilim.edu.tr	05419630640	Ankara	Keçiören	Kuşcağız mah	Kırmızı Gül, Kırmızı Gül,...	63	2250.00	Dilşad Çiçekçilik	Kredi Kartı	Ödendi	Teslim Edildi
2	2	2025-12-22 12:51:12.003	Hazırlanıyor	Anıl	GÜNGÖR	s220205003@ankarabilim.edu.tr	05416531560	Ankara	Keçiören	Atapark mah	Mor Kır Çiçeği, Mor Kır ...	18	2160.00	Ankara Çiçek Gönderimi	Nakit	Ödendi	Hazırlanıyor
3	3	2025-12-22 12:51:55.737	Tamamlandı	Fevzi Samed	ÜNAL	s220205032@ankarabilim.edu.tr	05419630640	Ankara	Keçiören	Kuşcağız mah	Beyaz Papatya, Beyaz P...	9	1800.00	Dilşad Çiçekçilik	Kredi Kartı	Ödendi	Hazırlanıyor
4	4	2025-12-22 12:51:55.737	Hazırlanıyor	Anıl	GÜNGÖR	s220205003@ankarabilim.edu.tr	05416531560	Ankara	Keçiören	Atapark mah	Kırmızı Gül, Kırmızı Gül,...	9	1350.00	Dilşad Çiçekçilik	Kredi Kartı	Ödendi	Hazırlanıyor
5	15	2025-12-22 13:06:04.800	Hazırlanıyor	Hamit	GÜNEŞ	s220204042@ankarabilim.edu.tr	05511026576	Ankara	Çankaya	Demirtepe mah	Beyaz Papatya, Beyaz P...	18	3600.00	Dilşad Çiçekçilik	Kredi Kartı	Ödendi	Hazırlanıyor
6	17	2025-12-22 13:08:07.430	Hazırlanıyor	Hamit	GÜNEŞ	s220204042@ankarabilim.edu.tr	05511026576	Ankara	Çankaya	Demirtepe mah	Mor Kır Çiçeği, Mor Kır ...	9	675.00	Ankara Çiçek Gönderimi	Kredi Kartı	Ödendi	Hazırlanıyor

Database Diagram



Phases	Tasks	Fevzi Samed ÜNAL	Anıl GÜNGÖR	Hamit GÜNEŞ
Phase-1	Problem Definition	✓	✓	✓
	Requirement Analysis	✓	✓	✓
	Identification of Entities and Attributes	✓	✓	✓
	Conceptual Data Model (ER Diagram)	✓	✓	✓
	Relationship Identification	✓	✓	✓
	Cardinality and Participation Analysis	✓	✓	✓
	Logical Data Model Design	✓	✓	✓
	Normalization (1NF, 2NF, 3NF)	✓	✓	✓
	Physical Data Model Design	✓	✓	✓
	Attribute Type Definition	✓	✓	✓
	Primary Key & Foreign Key Definition	✓	✓	✓
	Phase-1 Documentation	✓	✓	✓

Phases	Tasks	Fevzi Samed ÜNAL	Anıl GÜNGÖR	Hamit GÜNEŞ
Phase-2	Database Creation	✓	✓	✓
	Table Creation	✓	✓	✓
	Constraint Implementation (PK, FK, NOT NULL)	✓	✓	✓
	Relationship Implementation	✓	✓	✓
	Sample Data Design	✓	✓	✓
	Data Insertion	✓	✓	✓
	Data Integrity Testing	✓	✓	✓
	Verification of Database Structure	✓	✓	✓
	SQL Query Testing	✓	✓	✓
	Phase-2 Documentation	✓	✓	✓
	Advanced SQL Queries	✓	✓	✓
	Multi-table JOIN Queries	✓	✓	✓
	Order Detail Query Design	✓	✓	✓
	Data Consistency Checks	✓	✓	✓
	Performance Testing	✓	✓	✓
	Error Handling & Corrections	✓	✓	✓
	Final Report Preparation	✓	✓	✓
	Final Review & Submission	✓	✓	✓