Name of Database in SQL Server: Team30_Vamsitha_Ajay_Sudeep

Scripts:

```
-- Creating the Employee table
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Position VARCHAR(50),
    SupervisorID INT,
    Address VARCHAR(255),
    Phone VARCHAR(15),
    Email VARCHAR(100),
    EmergencyContact VARCHAR(100)
);

-- Creating the Supervisor table with a foreign key to Employee
CREATE TABLE Supervisor (
    SupervisorID INT PRIMARY KEY,
    EmployeeID INT,
    SupervisorName VARCHAR(100),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

-- Creating the Wine table
CREATE TABLE Wine (
    WineID INT PRIMARY KEY,
    WineName VARCHAR(100),
    VintageYear INT,
    Category VARCHAR(50),
    AlcoholPercentage DECIMAL(4,2)
);

-- Creating the GrapeHarvest table
CREATE TABLE GrapeHarvest (
    HarvestID INT PRIMARY KEY,
    VineyardID INT,
    GrapeVarietyID INT,
    EmployeeInCharge INT,
    SizeInAcres DECIMAL(10,2),
    VintageYear INT,
    TotalWeight DECIMAL(10,2),
    RipenessPercentage DECIMAL(5,2),
    FOREIGN KEY (EmployeeInCharge) REFERENCES Employee(EmployeeID)
);

-- Creating the Vineyard table
CREATE TABLE Vineyard (
    VineyardID INT PRIMARY KEY,
    VineyardName VARCHAR(100),
    Location VARCHAR(255)
);

-- Creating the GrapeVariety table
CREATE TABLE GrapeVariety (
    GrapeVarietyID INT PRIMARY KEY,
    VarietyName VARCHAR(100),
    JuiceConversionRatio DECIMAL(5,3)
);
```

```sql
-- Creating the BottleType table
CREATE TABLE BottleType (
    BottleTypeID INT PRIMARY KEY,
    Capacity INT,
    Shape VARCHAR(50)
);

-- Creating the Vendor table
CREATE TABLE Vendor (
    VendorID INT PRIMARY KEY,
    VendorName VARCHAR(100),
    Address VARCHAR(255),
    Phone VARCHAR(15)
);

-- Creating the PurchaseOrder table with foreign keys
CREATE TABLE PurchaseOrder (
    OrderID INT PRIMARY KEY,
    VendorID INT,
    OrderDate DATE,
    FOREIGN KEY (VendorID) REFERENCES Vendor(VendorID)
);

-- Creating the Customer table
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    CustomerType VARCHAR(50),
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    Address VARCHAR(255),
    ShippingAddress VARCHAR(255),
    DateOfBirth DATE,
    Email VARCHAR(100),
    Phone VARCHAR(15)
);

-- Creating the Order table with foreign keys
CREATE TABLE OrderTable (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);
```

Output

```
Commands completed successfully.

Completion time: 2024-04-17T15:59:49.2917282-06:00
```

```sql
-- Creating the Intern table
CREATE TABLE Intern (
    InternID INT PRIMARY KEY,
    EmployeeID INT,
    InternshipRequirement TEXT,
    GraduationDate DATE,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

-- Creating the Bottling table
CREATE TABLE Bottling (
    BottlingID INT PRIMARY KEY,
    WineID INT,
    BottleTypeID INT,
    FOREIGN KEY (WineID) REFERENCES Wine(WineID),
    FOREIGN KEY (BottleTypeID) REFERENCES BottleType(BottleTypeID)
);

-- Creating the Product table
CREATE TABLE Product (
    ProductID INT PRIMARY KEY,
    WineID INT,
    BottleTypeID INT,
    CaseQuantity INT,
    Price DECIMAL(10,2),
    FOREIGN KEY (WineID) REFERENCES Wine(WineID),
    FOREIGN KEY (BottleTypeID) REFERENCES BottleType(BottleTypeID)
);

-- Creating the WineComposition table
CREATE TABLE WineComposition (
    CompositionID INT PRIMARY KEY,
    WineID INT,
    GrapeVarietyID INT,
    JuiceProportion DECIMAL(5,2),
    FOREIGN KEY (WineID) REFERENCES Wine(WineID),
    FOREIGN KEY (GrapeVarietyID) REFERENCES GrapeVariety(GrapeVarietyID)
);

-- Creating the OrderLine table
CREATE TABLE OrderLine (
    OrderLineID INT PRIMARY KEY,
    ProductID INT,
    OrderID INT,
    Quantity INT,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
    FOREIGN KEY (OrderID) REFERENCES OrderTable(OrderID)
);

-- Creating the Payment table
CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    OrderID INT,
    PaymentType VARCHAR(50),
```

```
    PaymentDate DATE,
    Amount DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES OrderTable(OrderID)
);

-- Creating the MarketingCampaign table
CREATE TABLE MarketingCampaign (
    CampaignID INT PRIMARY KEY,
    CampaignName VARCHAR(100)
);

-- Creating the CustomerMarketingActivity table
CREATE TABLE CustomerMarketingActivity (
    ActivityID INT PRIMARY KEY,
    CustomerID INT,
    CampaignID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (CampaignID) REFERENCES MarketingCampaign(CampaignID)
);

-- Creating the WineClub table
CREATE TABLE WineClub (
    ClubID INT PRIMARY KEY,
    MemberID INT,
    JoinDate DATE,
    LeaveDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Customer(CustomerID)
);

-- Creating the ClubOfferedWine table
CREATE TABLE ClubOfferedWine (
    OfferedWineID INT PRIMARY KEY,
    ClubID INT,
    WineID INT,
    FOREIGN KEY (ClubID) REFERENCES WineClub(ClubID),
    FOREIGN KEY (WineID) REFERENCES Wine(WineID)
);

-- Creating the CottageRooms table
CREATE TABLE CottageRooms (
    RoomID INT PRIMARY KEY,
    RoomType VARCHAR(100),
    Availability VARCHAR(50),
    Price DECIMAL(10,2),
    Amenities TEXT
);
```

Output:

```
Commands completed successfully.

Completion time: 2024-04-17T16:02:05.1199893-06:00
```

```sql
-- Creating the Certification table
CREATE TABLE Certification (
    CertificationID INT PRIMARY KEY,
    EmployeeID INT,
    CertificationLevel VARCHAR(50),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

-- Creating the DriverCertification table
CREATE TABLE DriverCertification (
    DriverCertificationID INT PRIMARY KEY,
    EmployeeID INT,
    CDL VARCHAR(20),
    CertificationDate DATE,
    ReCertificationOutcomes VARCHAR(255),
    CDLExpirationDate DATE,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

-- Creating the RestaurentWineShop table
CREATE TABLE RestaurentWineShop (
    CustomerID INT PRIMARY KEY,
    CompanyName VARCHAR(255),
    TaxID VARCHAR(20),
    ResaleLicense VARCHAR(20),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

-- Creating the Reservation table
CREATE TABLE Reservation (
    ReservationID INT PRIMARY KEY,
    CustomerID INT,
    RoomType VARCHAR(50),
    CheckInDate DATE,
    CheckOutDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);
```

Output:

```
Commands completed successfully.

Completion time: 2024-04-17T16:03:36.0185989-06:00
```

```
-- Foreign key examples on other relationships
ALTER TABLE GrapeHarvest ADD CONSTRAINT FK_GrapeHarvest_Vineyard FOREIGN KEY (VineyardID)
REFERENCES Vineyard(VineyardID);
ALTER TABLE GrapeHarvest ADD CONSTRAINT FK_GrapeHarvest_GrapeVariety FOREIGN KEY
(GrapeVarietyID) REFERENCES GrapeVariety(GrapeVarietyID);
ALTER TABLE Wine ADD CONSTRAINT FK_Wine_GrapeHarvest FOREIGN KEY (WineID) REFERENCES
GrapeHarvest(HarvestID);
-- Foreign key constraints for Employee and Supervisor
ALTER TABLE Employee ADD CONSTRAINT FK_Employee_Supervisor FOREIGN KEY (SupervisorID)
REFERENCES Supervisor(SupervisorID);

-- Foreign key constraints for Bottling processes linking to Wine and BottleType
ALTER TABLE Bottling ADD CONSTRAINT FK_Bottling_Wine FOREIGN KEY (WineID) REFERENCES
Wine(WineID);
ALTER TABLE Bottling ADD CONSTRAINT FK_Bottling_BottleType FOREIGN KEY (BottleTypeID)
REFERENCES BottleType(BottleTypeID);

-- Foreign key constraints for Product linking to Wine and BottleType
ALTER TABLE Product ADD CONSTRAINT FK_Product_Wine FOREIGN KEY (WineID) REFERENCES
Wine(WineID);
ALTER TABLE Product ADD CONSTRAINT FK_Product_BottleType FOREIGN KEY (BottleTypeID)
REFERENCES BottleType(BottleTypeID);

-- Foreign key constraints for WineComposition linking to Wine and GrapeVariety
ALTER TABLE WineComposition ADD CONSTRAINT FK_WineComposition_Wine FOREIGN KEY (WineID)
REFERENCES Wine(WineID);
ALTER TABLE WineComposition ADD CONSTRAINT FK_WineComposition_GrapeVariety FOREIGN KEY
(GrapeVarietyID) REFERENCES GrapeVariety(GrapeVarietyID);

-- Foreign key constraints for CustomerMarketingActivity linking to Customer and
MarketingCampaign
ALTER TABLE CustomerMarketingActivity ADD CONSTRAINT
FK_CustomerMarketingActivity_Customer FOREIGN KEY (CustomerID) REFERENCES
Customer(CustomerID);
ALTER TABLE CustomerMarketingActivity ADD CONSTRAINT
FK_CustomerMarketingActivity_Campaign FOREIGN KEY (CampaignID) REFERENCES
MarketingCampaign(CampaignID);

-- Foreign key constraints for ClubOfferedWine linking to WineClub and Wine
ALTER TABLE ClubOfferedWine ADD CONSTRAINT FK_ClubOfferedWine_Club FOREIGN KEY (ClubID)
REFERENCES WineClub(ClubID);
ALTER TABLE ClubOfferedWine ADD CONSTRAINT FK_ClubOfferedWine_Wine FOREIGN KEY (WineID)
REFERENCES Wine(WineID);

-- Foreign key constraints for OrderLine linking to Product and Order
ALTER TABLE OrderLine ADD CONSTRAINT FK_OrderLine_Product FOREIGN KEY (ProductID)
REFERENCES Product(ProductID);
ALTER TABLE OrderLine ADD CONSTRAINT FK_OrderLine_Order FOREIGN KEY (OrderID) REFERENCES
OrderTable(OrderID);

-- Foreign key constraints for Payment linking to Order
```

```
ALTER TABLE Payment ADD CONSTRAINT FK_Payment_Order FOREIGN KEY (OrderID) REFERENCES
OrderTable(OrderID);

-- Foreign key constraints for Reservation linking to Customer
ALTER TABLE Reservation ADD CONSTRAINT FK_Reservation_Customer FOREIGN KEY (CustomerID)
REFERENCES Customer(CustomerID);

-- Additional relationships linking WineClub to Customer
ALTER TABLE WineClub ADD CONSTRAINT FK_WineClub_Customer FOREIGN KEY (MemberID)
REFERENCES Customer(CustomerID);

-- Foreign key constraints for Intern linking back to Employee
ALTER TABLE Intern ADD CONSTRAINT FK_Intern_Employee FOREIGN KEY (EmployeeID) REFERENCES
Employee(EmployeeID);
```

Output:

```
Commands completed successfully.

Completion time: 2024-04-17T16:06:12.3290764-06:00
```

Database Diagram (Relationship Window):