Term Project: Flower Store Management System

Group 3

[redacted] George Kaplan, [redacted]

MIS 380 Section 1

SAN DIEGO STATE UNIVERSITY

Step 1: Topic/Domain Selection

Selected Topic: Flower Store Management System.

Introduction:

Our topic is based on a flower store. With the use of a database and the information that will be provided from it, employees and managers that work in the store will have structured data that can help them determine what is happening day-to-day. The database can help show which flowers are being sold in the store, how many are being sold, overall inventory, and additional information about the flowers. This database is also used to show which employee made the corresponding sale, who set up the flowers, the amount the flowers were sold at, and the buyer's information. Each flower sale is linked to a sales invoice that shows information about the customer and sales representative in the transaction. This flower store produces and sells individual flowers, as well as multiple in one order to fulfill all requests such as for big events. For the bigger orders for events, the database will help with determining the different flower sets to make so that they are not all the same.

Advantages:

This database management system is beneficial for the flower shop because it will allow the managers to see how each of the operations are running with the vendors and customers. One way this database is helpful is by allowing the shop managers to see how much money they make on a month by month basis and track how much they are spending on inventory. Additionally, the managers could monitor which orders from the vendors have been completed or are still processing. In the event that a customer is given the wrong flowers, the managers would

be able to look back on the order ticket and see what flowers they were supposed to receive in order to fix the mistake. By looking at which flowers were popular in the past, this data can be used to indicate flowers that were used the most, helping the managers determine what they should order again from their vendor. If the store wanted to give a loyalty discount to customers who have spent a certain amount of money in the past, the database would be able to track total sales from each customer on file. Overall, this database would be beneficial in allowing the managers to oversee all logistical aspects of the business by having management of vendors, customers, sales, and inventory all in one place.

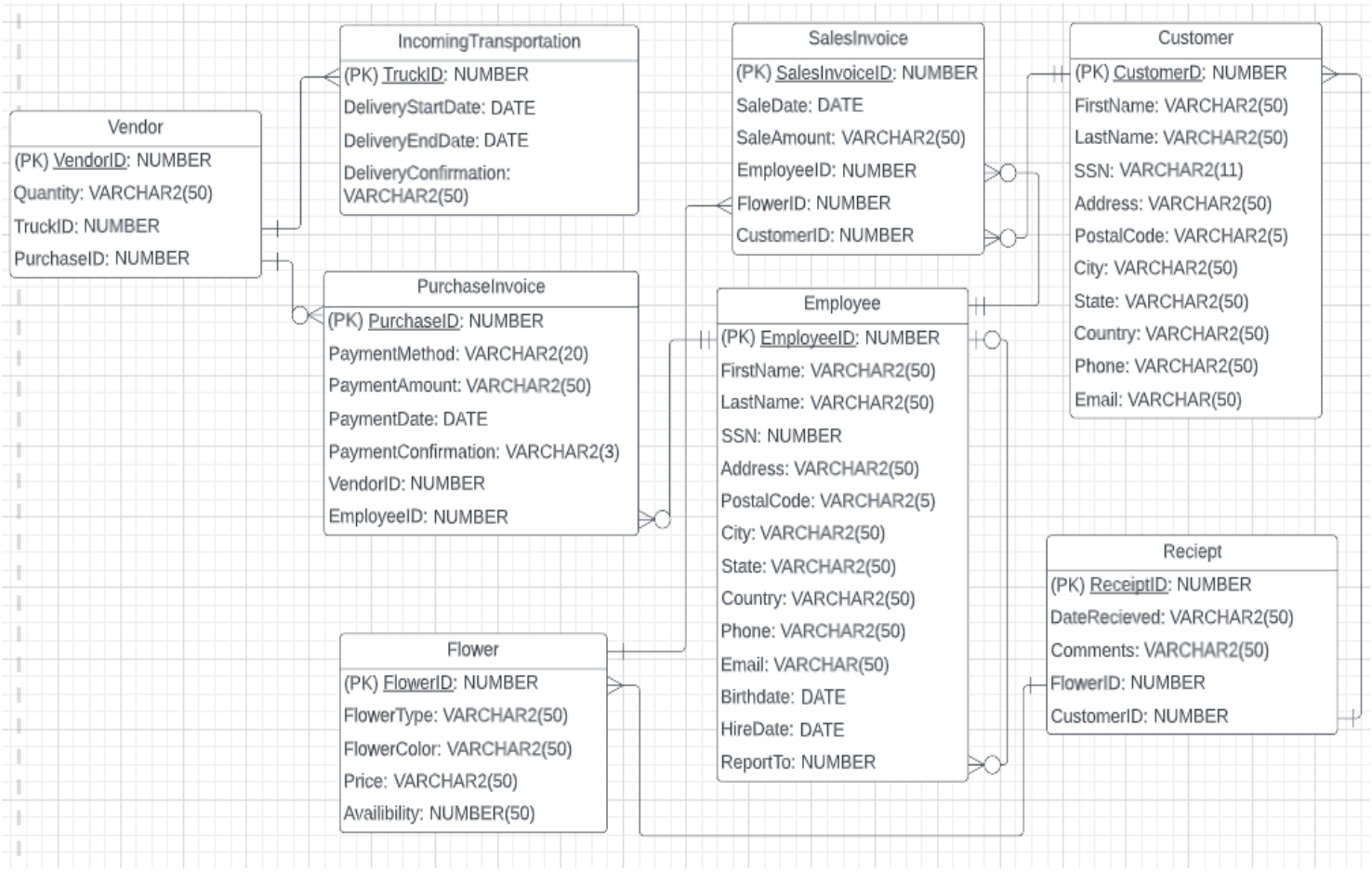
Uses Cases: The general purpose of this database is to help the flower store simplify and structure loads of data for managers to make better and more efficient decisions. The database will give all relevant information into neatly packaged format and allow for answering questions about the businesses performance. This database can help with tracking all transactions that happen daily such as customer information from each sale, all aspects of the sales transaction, inventory coming in through their vendors and out through selling from business to customer, as well as track employee performance. This database will also reduce the amount of repetitive information that is not needed and gives data that is relevant to run the flower store. The way this database is set up allows for other kiosk type flower stores to be able to use the same database with no problems. This database is also very efficient since with small changes in the database entities, it can be used by many other companies with similar transactions and can file and organize data efficiently.

Step 2: Conceptual Data Modeling and Database Design

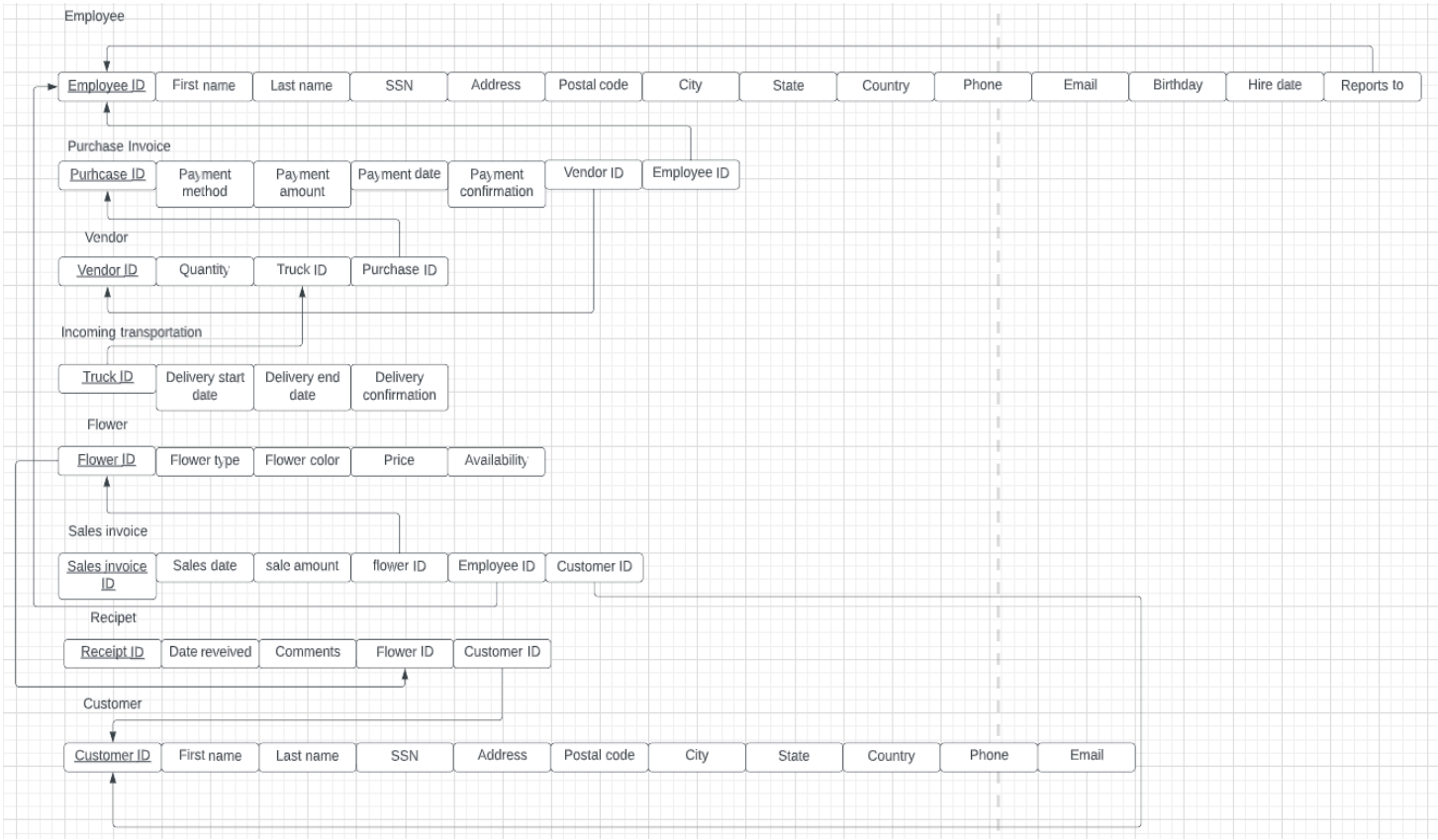
Business Rules:

1. Every employee who is not a manager reports to a manager. Managers can have multiple employees reporting to them. Managers do not report to report to employees
2. A sales invoice is always specific to a specific customer.
3. Each sales invoice relates to exactly one receipt. With that, a sales invoice will have a specific customer ID that correlates with an individual customer. There is only one customer ID number for each customer, but one customer can have multiple sales invoices by purchasing different products at different times.
4. Each receipt describes a specific flower to be made and its details including flowers used, colors, and price. A receipt always relates to exactly one customer and one sales invoice.
5. Each incoming transportation has one specific truck that delivers the products to the flower store.
6. The flower store can have one or many purchases from the vendor at one time. Once the flower store finds what they want to be given to them, a purchase invoice will be created.
7. There is only one purchase ID for every time flowers are bought from the vendor.
8. For every purchase, one receipt is given to a customer and a customer can have multiple receipts from multiple purchases.
9. There can be one or two employees that create zero to many purchase invoices as well as a sales invoice. At least one employee must be responsible for each invoice that is made.

ER/EER Diagram:



Relational Model:



Step 3: Database Implementation

DDL SQL commands to create tables: Below are the SQL commands that were used to set up the database in SQLite. The code can also be found in an attached file (MIS380TermProject.db).

```
CREATE TABLE Vendor
(
VendorID NUMBER NOT NULL,
Quantity VARCHAR2(50),
TruckID NUMBER NOT NULL,
PurchaseID Number NOT NULL,
CONSTRAINT PK_Vendor PRIMARY KEY (VendorID)
);

CREATE TABLE IncomingTransportation
(
TruckID NUMBER NOT NULL,
DeliveryStartDate DATE,
DeliveryEndDate DATE,
DeliveryConfirmation VARCHAR2(50) NOT NULL,
CONSTRAINT PK_IncomingTransportation PRIMARY KEY (TruckID)
);

CREATE TABLE PurchaseInvoice
(
PurchaseID NUMBER NOT NULL,
PaymentMethod VARCHAR2(20),
PaymentAmount VARCHAR2(50),
PaymentDate DATE,
PaymentConfirmation VARCHAR2(3),
VendorIF NUMBER,
EmployeeID NUMBER,
CONSTRAINT PK_PurchaseInvoice PRIMARY KEY (PurchaseID)
);

CREATE TABLE Employee
(
EmployeeID NUMBER NOT NULL,
FirstName VARCHAR2(50) NOT NULL,
LastName VARCHAR2(50) NOT NULL,
SSN VARCHAR2(11) NOT NULL,
Address VARCHAR2(50) NOT NULL,
PostalCode VARCHAR2(5) NOT NULL,
City VARCHAR2(50) NOT NULL,
```

```
City VARCHAR2(50) NOT NULL,
State VARCHAR2(50) NOT NULL,
Country VARCHAR2(50) NOT NULL,
Phone VARCHAR2(50) NOT NULL,
Email VARCHAR2(50) NOT NULL,
Birthdate DATE,
HireDate DATE,
ReportTo NUMBER,
CONSTRAINT PK_Employee Primary Key (EmployeeID),
FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)
);

CREATE TABLE SalesInvoice
(
SalesInvoiceID NUMBER,
SaleDate DATE,
SaleAmount VARCHAR2(50),
EmployeeID NUMBER,
FlowerID NUMBER,
CustomerID NUMBER,
CONSTRAINT PK_SalesInvoice PRIMARY KEY (SalesInvoiceID),
FOREIGN KEY (FlowerID) REFERENCES Flower (FlowerID),
FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
);

CREATE TABLE Customer
(
CustomerID NUMBER NOT NULL,
FirstName VARCHAR2(50) NOT NULL,
LastName VARCHAR2(50) NOT NULL,
SSN VARCHAR2(11) NOT NULL,
Address VARCHAR2(50) NOT NULL,
PostalCode VARCHAR2(5) NOT NULL,
City VARCHAR2(50) NOT NULL,
State VARCHAR2(50) NOT NULL,
Country VARCHAR2(50) NOT NULL,
Phone VARCHAR2(50) NOT NULL,
Email VARCHAR2(50) NOT NULL,
CONSTRAINT PK_Customer PRIMARY KEY (CustomerID)
);
```

```

CREATE TABLE Flower
(
FlowerID NUMBER,
FlowerType VARCHAR2(50) NOT NULL,
FlowerColor VARCHAR2(50) NOT NULL,
Price VARCHAR2(50) NOT NULL,
Availability VARCHAR2(50) NOT NULL,
CONSTRAINT PK_Flower PRIMARY KEY (FlowerID)
);

CREATE TABLE Receipt
(
ReceiptID NUMBER,
DateReceived VARCHAR2(50) NOT NULL,
Comments VARCHAR2(50) NOT NULL,
FlowerID NUMBER,
CustomerID NUMBER,
CONSTRAINT PK_Receipt PRIMARY KEY (ReceiptID),
FOREIGN KEY (FlowerID) REFERENCES Flower (FlowerID),
FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
);
```

SQL Commands to insert the data into the database: A fake data set with a total of 159 rows was created using MS Excel (MIS380SampleData.xlsx). A file with the complete database with the fake data is also attached (MIS380TermProject.db).

Analytical Questions: Below are exemplary analytic questions and corresponding SQL statements.

SINGLE

What was the total revenue for the Flower Type Orchid?

SELECT Flower.FlowerType, SUM (SalesInvoice.SaleAmount) AS Total Sale,

FROM SalesInvoice INNER JOIN Flower

ON SalesInvoice.FlowerID = Flower.FlowerID

WHERE Flower.FlowerType = 'Orchid';

Which Customers had the highest and lowest sales in 2020?

SELECT CustomerID, FirstName, LastName

MAX (SaleAmount) as Max_Invoice,

MIN (SaleAmount) as Min_Invoice

FROM SalesInvoice *INNERJOIN* Customer

ON Customer.CustomerID = SalesInvoice.CustomerID

WHERE SaleDate IN ('2020')

GROUP BY CustomerID

MULTI TABLE

Which Customer spent the most on flowers in 2021 (ordered from highest to lowest)?

SELECT Customer.CustomerID, Flower.FlowerID

*WITH A Q (SUM (Flower.Price x SalesInvoice.Quantity) AS FlowerSales

*W/O: SalesInvoice.SaleAmount (assuming this is per flower)

FROM Customer, Flower, SalesInvoice

WHERE Customer.CustomerID = SalesInvoice.CustomerID

AND Flower.FlowerID = SalesInvoice.FlowerID

AND SalesInvoice.SalesDate = “2021”

GROUP BY Customer.CustomerID, Flower.FlowerID, SalesInvoice.SaleAmount

ORDER BY SalesInvoice.SaleAmount **DESC;**

Step 4: Enterprise (Web) Database Dashboard

The database dashboard was made on Tableau and can be found under the following link:

[Flower Bouquet DashBoard.twb]