Term Project: Coffee Shop Database

Group 3

Spencer Soalt, Layne Consales, Ricardo Romero, Kevin Sanchez

MIS 380 Section 02

Step 1: Topic/Domain Selection

Selected Topic: Coffee Shop

Introduction: Our database is designed to support the operations of a coffee shop. It includes records for all employees, past and present, and allows us to identify each employee's role (manager, cashier, or barista) and the coffee shop they are assigned to. It also tracks the reporting structure of the employees, so we can see who each employee reports to. In addition to employee records, the database contains information about customers and their orders. To place an order, a customer must create an account with the coffee shop by providing their name, email, phone number, and an account name. Once they have an account, they can use it to place orders and view details about their past orders, including the items they ordered, the quantities of each, and the payment method they used. The database also allows us to see the current inventory of each coffee shop location, so we can track which items are available at each location and in what quantities. Finally, the database maintains transactional records for each customer account, including purchase histories and product usage information. This data can be used to offer personalized recommendations and discounts to customers based on their preferences and past purchases.

Advantages: The database management system will provide significant benefits to the coffee shop by streamlining the process of managing order data. It will allow employees to input orders quickly and accurately, which will reduce the time and effort required to process payments and track orders. This will increase the overall productivity and efficiency of the coffee shop, as more orders can be processed in a given time frame. The database also provides a centralized system for tracking and fixing orders, which can improve customer satisfaction by addressing any issues that may arise. Additionally, the database will provide benefits to employees through the HR department, such as access to information about their own performance and opportunities for growth. It will also track the company's revenue, allowing managers to make data-driven decisions about the business. Furthermore, the database eliminates redundancy by providing a single location for both online and in-person orders, which simplifies the process of managing and tracking orders. Overall, the database management system will provide numerous benefits to the coffee shop, including increased productivity, improved customer satisfaction, and streamlined data management.
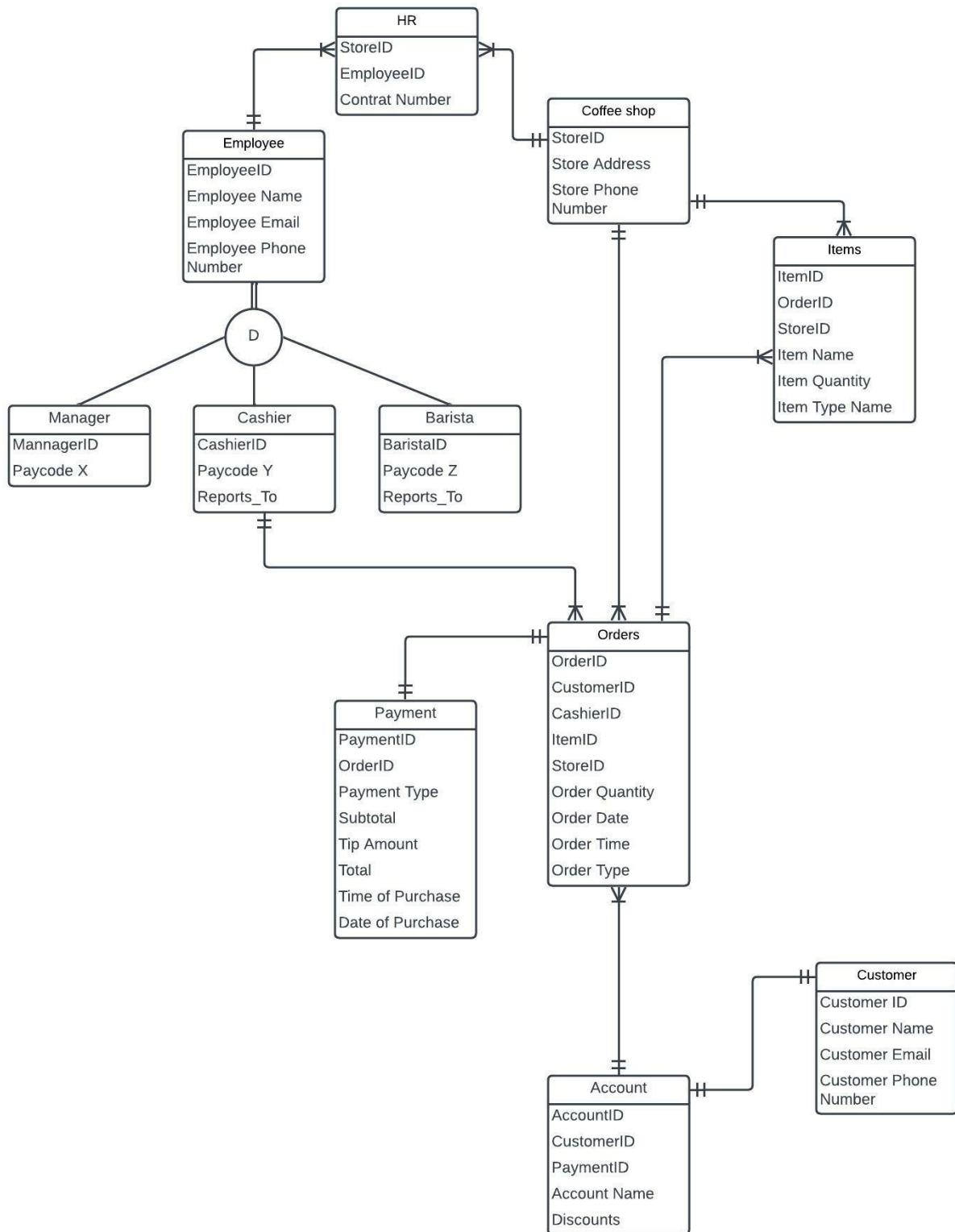
Uses **Cases:** The proposed database is designed to support the operations of coffee shop businesses. It includes a user-friendly online platform that allows customers to easily place orders and track data related to their purchase history, coffee customization, payment methods, and even coffee recommendations. This will help coffee shops to attract and retain customers who may be hesitant to place orders online because of outdated or confusing databases. Additionally, the database provides a centralized system for tracking and managing orders, inventory, employee information, and other data related to the coffee shop's operations. This will streamline the process of managing and analyzing data, which can help to improve the efficiency and productivity of the business. In addition, the database can be customized by businesses to suit their specific needs, such as by including service descriptions, receipts, and other analytics. Overall, the proposed database offers a comprehensive solution for managing and tracking data related to coffee shop operations, which will benefit both customers and businesses.

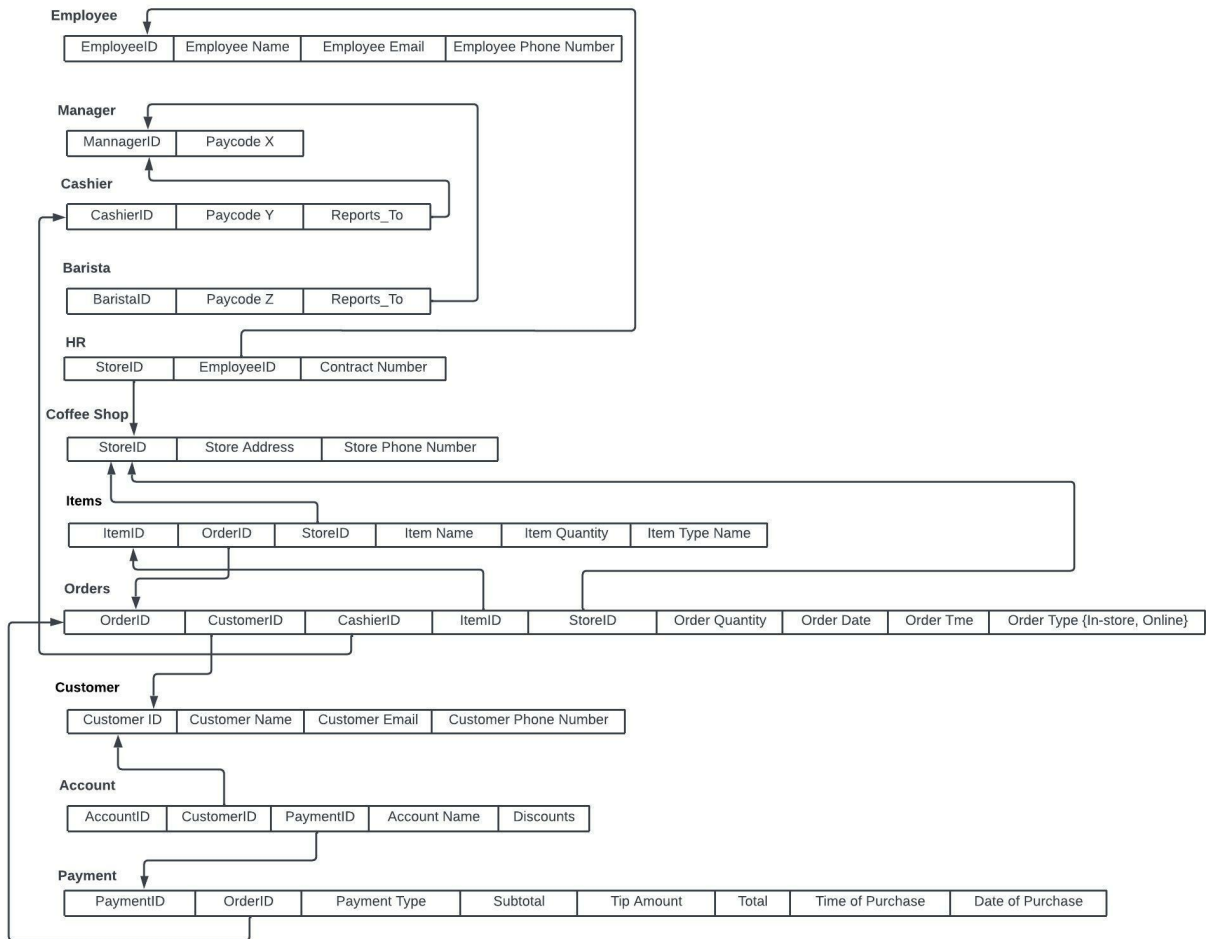## Step 2: Conceptual Data Modeling and Database Design

Business Rules:

1. Each employee must be assigned to exactly one coffee shop.
2. Each employee has to be either a manager, cashier, or barista.
3. Each Employee that is not a manager must report to at least one manager.
4. Every manager, cashier, and barista must be registered as an employee.
5. All managers must be paid paycode x, all cashiers must be paid paycode y, and all baristas must be paid paycode z.
6. Each coffee shop must have at least one employee working.
7. All Coffee Shops are only open from 6:00am to 12:00pm.
8. Each employee must be assigned to exactly one coffee shop.
9. HR hold records for store ID's, employee's ID's, and Contract # for each employee.
10. Each item possesses an item ID, name, quantity, description and price.
11. Each item belongs to exactly one Coffee Shop location.
12. Each item is used for exactly one order.
13. Each store location can have one to many of each item.
14. Each order has a cashier attached to it but no barista or manager.
15. Each order can only be placed by exactly one customer.
16. Each order must have a date/time of order.
17. Each order must have a receipt number.
18. Orders can be placed either instore or online. (must register first)
19. Every payment must be connected to exactly one order.
20. Payments can either be completed by cash or card.
21. All payments must be marked with a time and date of purchase.
22. Each account must be connected to exactly one and only one customer.
23. Each account can place one to many orders.
24. Each account has transactional details such as purchase histories and product usage information (e.g. repeated actions)
25. Customers can qualify for a discount type dependent on the number of previous orders they have placed
26. Every customer can have exactly one account, to create an account customer must provide name, email, and phone number.

ER/EER Diagram:

HR
- StoreID
- EmployeeID
- Contrat Number

Coffee shop
- StoreID
- Store Address
- Store Phone Number

Employee
- EmployeeID
- Employee Name
- Employee Email
- Employee Phone Number

Items
- ItemID
- OrderID
- StoreID
- Item Name
- Item Quantity
- Item Type Name

D

Manager
- MannagerID
- Paycode X

Cashier
- CashierID
- Paycode Y
- Reports_To

Barista
- BaristaID
- Paycode Z
- Reports_To

Orders
- OrderID
- CustomerID
- CashierID
- ItemID
- StoreID
- Order Quantity
- Order Date
- Order Time
- Order Type

Payment
- PaymentID
- OrderID
- Payment Type
- Subtotal
- Tip Amount
- Total
- Time of Purchase
- Date of Purchase

Customer
- Customer ID
- Customer Name
- Customer Email
- Customer Phone Number

Account
- AccountID
- CustomerID
- PaymentID
- Account Name
- Discounts

Relational Model:

Employee

EmployeeID	Employee Name	Employee Email	Employee Phone Number

Manager

MannagerID	Paycode X

Cashier

CashierID	Paycode Y	Reports_To

Barista

BaristaID	Paycode Z	Reports_To

HR

StoreID	EmployeeID	Contract Number

Coffee Shop

StoreID	Store Address	Store Phone Number

Items

ItemID	OrderID	StoreID	Item Name	Item Quantity	Item Type Name

Orders

OrderID	CustomerID	CashierID	ItemID	StoreID	Order Quantity	Order Date	Order Tme	Order Type {In-store, Online}

Customer

Customer ID	Customer Name	Customer Email	Customer Phone Number

Account

AccountID	CustomerID	PaymentID	Account Name	Discounts

Payment

PaymentID	OrderID	Payment Type	Subtotal	Tip Amount	Total	Time of Purchase	Date of Purchase

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 be found in an attached .sql file. (MIS 380 Term Project Create Tables.sql)

```sql
CREATE TABLE Accounts (
    AccountID      NUMBER          NOT NULL,
    CustomerID     NUMBER          NOT NULL,
    PaymentID      NUMBER          NOT NULL,
    Account_Name VARCHAR2 (15) NOT NULL,
    Discounts      NUMBER          NOT NULL,
    CONSTRAINT PK_Accounts PRIMARY KEY (
        AccountID
    ),
    FOREIGN KEY (
        CustomerID
    )
    REFERENCES Customer (CustomerID),
    FOREIGN KEY (
        PaymentID
    )
    REFERENCES Payment (PaymentID)
);

CREATE TABLE Barista (
    BaristaID   NUMBER          NOT NULL,
    PaycodeZ    VARCHAR2 (50) NOT NULL,
    Reports_To VARCHAR2 (20) NOT NULL,
    FOREIGN KEY (
        BaristaID
    )
    REFERENCES Employee (EmployeeID)
);

CREATE TABLE Cashier (
    CashierID   NUMBER          NOT NULL,
    PaycodeY    VARCHAR2 (50) NOT NULL,
    Reports_To VARCHAR2 (20) NOT NULL,
    FOREIGN KEY (
        CashierID
    )
    REFERENCES Employee (EmployeeID)
);

CREATE TABLE Coffee_Shop (
    StoreID            NUMBER                      NOT NULL,
    Store              [ADDRESS VARCHAR2] (100) NOT NULL,
    Store_Phone_Number VARCHAR2 (75)              NOT NULL,
    CONSTRAINT PK_Coffee_Shop PRIMARY KEY (
        StoreID
    )
);

CREATE TABLE Customer (
    CustomerID           NUMBER          NOT NULL,
    Customer_Name        VARCHAR2 (40) NOT NULL,
    Customer_Email       VARCHAR2 (40) NOT NULL,
    Customer_Phone_Number VARCHAR2 (25) NOT NULL,
    CONSTRAINT PK_Customer PRIMARY KEY (
        CustomerID
    )
);

CREATE TABLE Employee (
    EmployeeID     NUMBER          NOT NULL,
    Employee_Name VARCHAR2 (50) NOT NULL,
    Email          VARCHAR2 (50) NOT NULL,
    Phone          VARCHAR2 (50) NOT NULL,
    CONSTRAINT PK_Employee PRIMARY KEY (
        EmployeeID
    )
);
```

```sql
CREATE TABLE HR (
    StoreID         NUMBER NOT NULL,
    EmployeeID      NUMBER NOT NULL,
    Contract_Number NUMBER NOT NULL,
    FOREIGN KEY (
        EmployeeID
    )
    REFERENCES Employee (EmployeeID),
    FOREIGN KEY (
        StoreID
    )
    REFERENCES Coffee_Shop (StoreID)
);

CREATE TABLE Items (
    ItemID          NUMBER          NOT NULL,
    OrderID         NUMBER          NOT NULL,
    StoreID         NUMBER          NOT NULL,
    Item_Name       VARCHAR2 (20)   NOT NULL,
    Item_Quantity   NUMBER          NOT NULL,
    Item_Type_Name  VARCHAR2 (30)   NOT NULL,
    CONSTRAINT PK_Items PRIMARY KEY (
        ItemID
    ),
    FOREIGN KEY (
        OrderID
    )
    REFERENCES Orders (OrderID),
    FOREIGN KEY (
        StoreID
    )
    REFERENCES Coffee_Shop (StoreID)
);

CREATE TABLE Manager (
    ManagerID NUMBER          NOT NULL,
    PaycodeX  VARCHAR2 (50) NOT NULL,
    FOREIGN KEY (
        ManagerID
    )
    REFERENCES Employee (EmployeeID)
);

CREATE TABLE Orders (
    OrderID         NUMBER          NOT NULL,
    CustomerID      NUMBER          NOT NULL,
    CashierID       NUMBER          NOT NULL,
    ItemID          NUMBER          NOT NULL,
    StoreID         NUMBER          NOT NULL,
    Order_Quantity  NUMBER          NOT NULL,
    Order_Date      VARCHAR2 (30)   NOT NULL,
    Order_Time      TIME (30)       NOT NULL,
    Order_Type      VARCHAR2 (75)   NOT NULL,
    CONSTRAINT PK_Orders PRIMARY KEY (
        OrderID
    ),
    FOREIGN KEY (
        CustomerID
    )
    REFERENCES Customer (CustomerID),
    FOREIGN KEY (
        ItemID
    )
    REFERENCES Items (ItemID),
    FOREIGN KEY (
        CashierID
    )
    REFERENCES Cashier (CashierID),
    FOREIGN KEY (
        StoreID
    )
    REFERENCES Coffee_Shop (StoreID)
);
```

```
CREATE TABLE Payment (
    PaymentID        NUMBER          NOT NULL,
    OrderID          NUMBER          NOT NULL,
    Payment_Type     VARCHAR2 (20)   NOT NULL,
    Subtotal         NUMBER          NOT NULL,
    Tip_Amount       NUMBER          NOT NULL,
    Total            NUMBER          NOT NULL,
    Time_of_Purchase VARCHAR2 (8)    NOT NULL,
    Date_of_Purchase VARCHAR2 (10)   NOT NULL,
    CONSTRAINT PK_Payment PRIMARY KEY (
        PaymentID
    ),
    FOREIGN KEY (
        OrderID
    )
    REFERENCES Orders (OrderID)
);
```

SQL commands to insert data into the database: To populate our database we created a fake dataset on google sheets. This dataset consists of 11 tables and 200 rows. The dataset can be seen in the attached excel file (MIS 380 Term Project Dataset.xlsx). We used the INSERT command to transfer the data from excel to the database. The individual INSERT commands can be found in the attached SQL file (MIS 380 Term Project Create Tables.sql). Furthermore the finished tables can be viewed in the SQLite Database (MIS 380 Term Project.db).

Analytic questions: Below are some examples of the analytical questions and the SQL queries we created to answer them.

Who are the customers with the highest average order price, and how do their spending habits compare to other customers?
```
SELECT c.Customer_Name as Customer, AVG(p.Total) as Avg_Order
FROM Payment p
JOIN Orders o ON p.OrderID = o.OrderID
JOIN Customer c ON o.CustomerID = c.CustomerID
GROUP BY c.Customer_Name
order by AVG(p.Total) desc
```

What are the sales totals for each cashier, and how do these compare to overall sales for each store and the company as a whole?
```
SELECT e.Employee_Name as Cashier, s.StoreID as
Store,SUM(p.Total) as Total
```

```
FROM Payment p
JOIN Orders o ON p.OrderID = o.OrderID
JOIN Cashier c ON o.CashierID = c.CashierID
JOIN Employee e ON c.CashierID = e.EmployeeID
JOIN Coffee_Shop s ON o.StoreID = s.StoreID
GROUP BY e.Employee_Name, s.StoreID
ORDER BY SUM(p.Total) DESC
```

In step 4, we used our Database Dashboard to conduct a deeper analysis of these questions and many others. The SQL queries for the remaining questions can be found in the python dashboard file (MIS 380 Term Project.ipynb). Here are some of the other analytical questions we explored:

- How many employees are currently employed at all coffee shop locations in the company?
- What is the total number of customers that have visited any of the coffee shops in the company?
- How many individual coffee shops are there within the company?
- What is the total number of orders that have been placed by customers at any of the coffee shops?
- What is the total number of orders per store, and how does this vary by store location?
- What is the total number of orders per item type, and how does this vary?
- What is the average price per order for all customers, and what does the distribution look like?
- How many orders are placed per cashier, on average, and how does this vary by time of day?
- What is the busiest time of day, in terms of number of orders, and how does this vary by store location and customer?
- How many discounts are offered at each discount rate, and which are most prevalent?

Step 4: Enterprise (web) Database Dashboard

The database dashboard can be found under the following link:

https://cloud.datapane.com/apps/qkWVYj7/mis-380-term-project-interactive-dashboard/

The database dashboard provides information on the business aspects of the coffee shop. The code that was used to create the dashboard was built on the provided template (MIS 380-simple Dashboard.ipynb). The final code has been attached as a python file (MIS 380 Term Project Dashboard.ipynb).