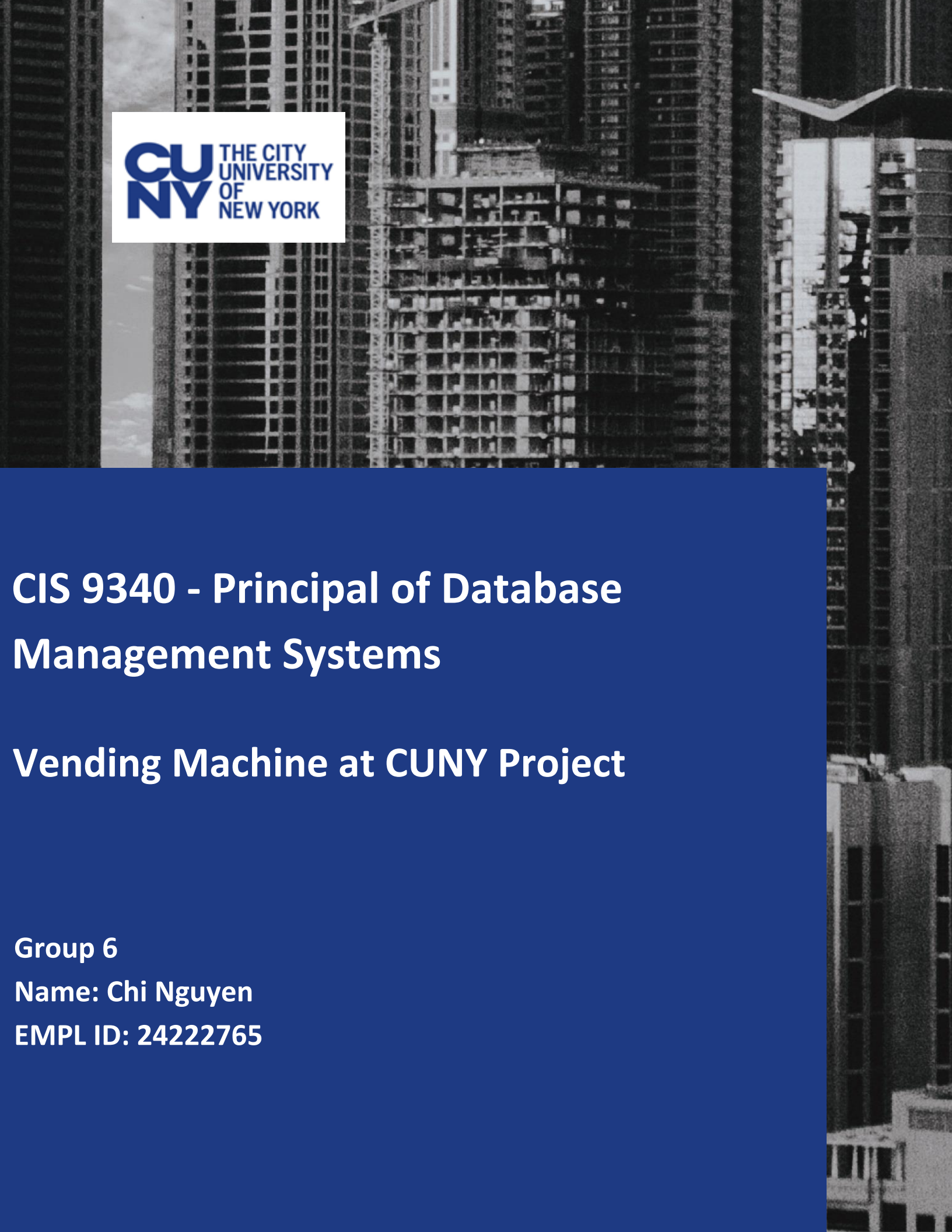


CIS 9340 - Principal of Database Management Systems

Vending Machine at CUNY Project

Group 6

Name: Chi Nguyen

EMPL ID: 24222765

Table of Contents

I. Introduction

1. Project Purpose

CUNY has a wide range of vending machines to provide its staff and students with various types of snacks and beverages. In order to manage the machines effectively and bring employees and students the best services, the school has decided to research and develop a database to keep track of all the information relating to the machines.

2. Methodology

This report covers steps that describe how the final vending machine database management model was developed. My project begins with a description of the CUNY vending machine Entity-Relationship model. Then, based on the ER model, I will convert the ER model to relational model and conduct the normalization process. After the normalization, the ER model and relational models will be changed accordingly. Lastly, by applying SQL, the database is created in MS Access to record all the information for future control and analysis.

3. Scenario

The vending machines are positioned across the campuses of the CUNY colleges. CUNY signs contracts to buy the vending machines from several manufacturers. Machine maintenance, which takes place every three months, is the responsibility of these manufacturers. They are also in charge of fixing the machines when machines encounter technical issues. Users frequently face a few technical problems, such as vending machines not returning change or goods after accepting payment, cards that cannot be read, etc. Each of these problems is logged together with the identity of the reporter so that the schools may get in touch with them after the issue has been resolved.

The details of products sold in each vending machine are recorded. To get the snacks or beverages, each person should make successful payments to the machines. Only successful transactions are logged into the system.

The products in vending machines decrease after each successful transaction, and they need to be refilled every working day to avoid product shortages. The refilling process will be handled by the school staff responsible for maintaining the school facilities. These staff will control the product inventories and place orders with suppliers when the items run out of stock.

After receiving the orders from the school staff, the suppliers deliver the ordered products to the school. The information of suppliers and shipment details are recorded in the database system to keep track of the inventory level.

II. Before Normalization - System Analysis and ER Model

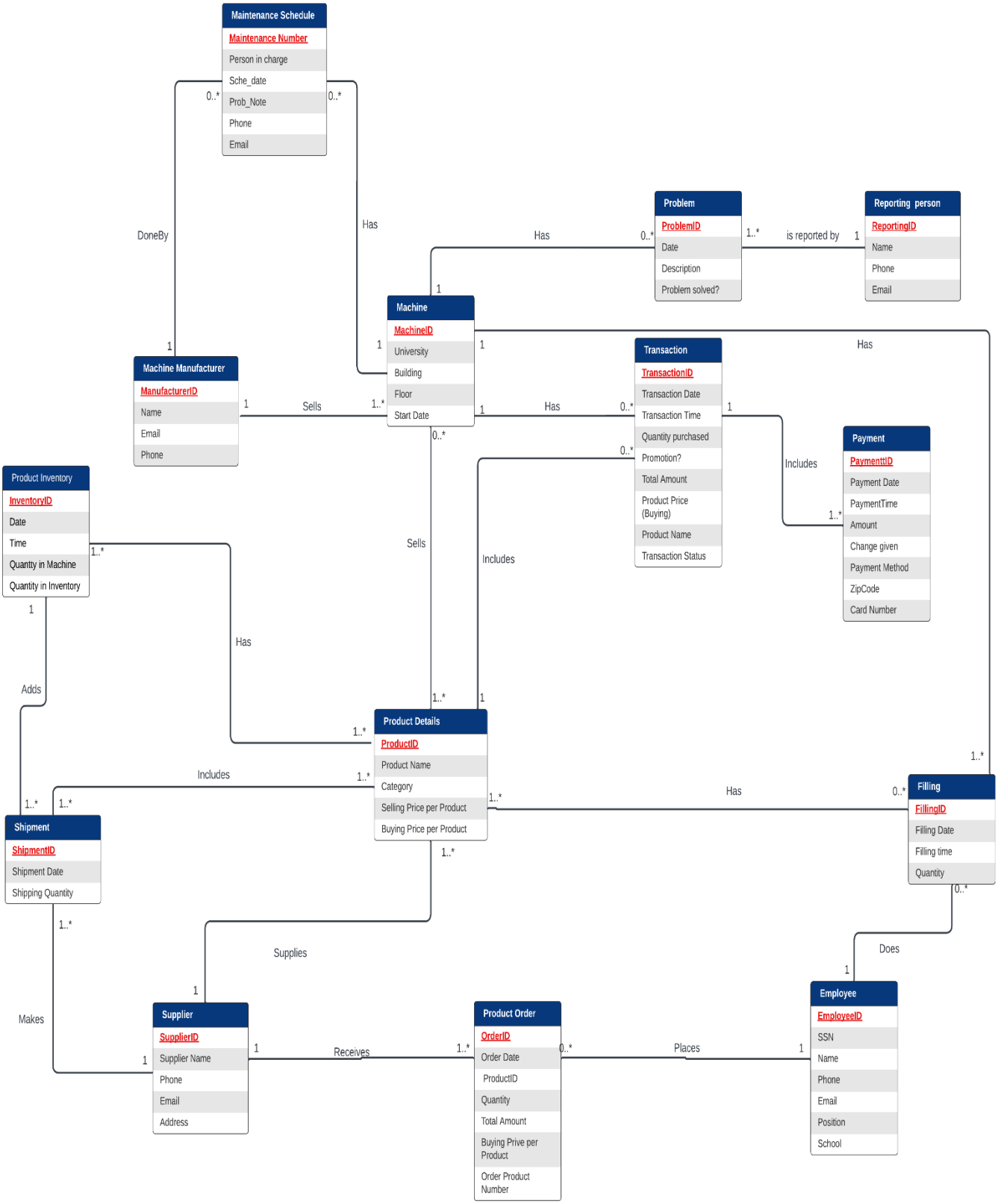
1. Assumptions

CUNY has a wide range of universities and colleges, and each university has its inventory to store products supplied for vending machines. Assume that each CUNY worker and student has a unique EMPL ID that helps to distinguish them, and that a general department is in charge of handling all tasks related to vending machines.

Entity	Entity	Description and Assumptions	Cardinality
Machine	Machine Manufacturer	A machine is manufactured by a manufacturer, but a manufacturer can sell many machines to CUNY	1:*
Machine	Maintenance Schedule	There are 0 (for new machines) or numerous maintenance schedules for each machine. One machine will be evaluated throughout each maintenance schedule	1:*
Machine Manufacturer	Maintenance Schedule	Manufacturer must assign its employee to do the maintenance. A manufacturer can do 0-many maintenance (because some of the machines are still new, so 0 is possible in this case). Each maintenance can be done by only 1 person	1:*
Machine	Problem	A machine while operating may cause zero or many issues. A problem recorded is caused by a machine only	1:*
Reporting Person	Problem	As said, the information of people reporting the issues will be recorded so that the school may get in touch with them after the issues have been resolved. Each issue recorded is associated with a reporting individual, who may inform one or many problems.	1:*
Machine	Transaction	A machine can have many 0 (if it's new) or many transactions. One transaction is made to a machine only.	1:*
Transaction	Payment	A transaction includes 1 or many times of payments. Each payment is conducted to a transaction only.	1:*

Entity	Entity	Description and Assumptions	Cardinality
Supplier	Shipment	A supplier can make 1 or many shipments to CUNY, and each shipment is conducted by a supplier	1:*
Supplier	Product Details	A supplier can supply only 1 product or many ones. However, each product is provided by a supplier only	1:*
Supplier	Product Order	A supplier can receive 1 or many orders from CUNY, but an order includes only 1 supplier.	1:*
Product Order	Employee	An employee can order nothing or place many orders with the suppliers. Each order is made by only 1 employee	*:1
Employee	Filling	Each employee is in charge of refilling the vending machine one or more times. Each filling task is completed by a single employee	1:*
Product Details	Shipment	Suppliers make shipments to CUNY to refill the items in vending machines. A product can be shipped multiple times, and a shipment contains various products	*:*
Product Inventory	Product Details	An inventory has lots of products, and a product can be included in many universities' inventories	*:*
Shipment	Product Inventory	Each shipment adds more products to an inventory. Over the time, there are many shipments made to an inventory	*:1
Machine	Filling	A vending machine may be refilled once or often. Yet only one vending machine is used for each filling activity that is recorded	1:*
Product Details	Transaction	A productid is purchased for each transaction. A product may be included in 0 or many transactions	1:*
Machine	Product Details	At CUNY, a product may be placed in 0 or many machines and a machine can sell 1 or many items	*:*
Product Details	Filling	The vending machine is restocked with different items daily. Each filling includes 1 or many items, and each item may be restocked 0 (unpopular items) or many times	*:*

2. Entities and Attributes



Entities	Attributes
Machine	MachineID, University, Building, Floor, Start Date
Machine Manufacturer	ManufacturerID, Name, Email, Phone
Maintenance Schedule	Maintenance Number, Person in charge, Sche_Datetime, Prob_Note, Phone, Email
Problem	ProblemID, Date, Description, Probolem Solved?
Reporting Person	ReportingID, Name, Phone, Email
Payment	PaymentID, Payment Date, Payment Time, Amount, Change Given, Payment Method, ZipCode, (E.g: Cash, Debit Card, Credit Card), Card Number
Product Details	ProductID, Product Name, Category, Selling Price per Product, Buying Price per Product
Shipment	ShipmentID, Shipment Date, Shipment Quantity
Supplier	SupplierID, Supplier Name, Phone, Emai, Address
Product Order	OrderID, Order Date, ProductID, Quantity, Total Amount, Buying Price per Product, Order Product Number
Employee	EmployeeID, SSN, Name, Phone, Email, Position
Filling	FillingID, Filling DateTime, Quantity
Product Inventory	InventoryID, DateTime, Quantity in Machine, Quantity in Inventory
Transaction	TransactionID, Transaction Date, Transaction Time, Quantity purchased, Promotion?, Total Amount, Product Price, Product Name, Transaction Status

3. Relational Model

Relation	Attribute
Machine	MachineID, University, Building, Floor, Start Date, ManufaturerID (fk)
Machine Manufacturer	ManufacturerID, Name, Email, Phone
Maintenance Schedule	Maintenance Number, Person in charge, Sche_Datetime, Prob_Note, Phone, Email, MachineID (fk), ManufaturerID(fk)
Problem	ProblemID, Date, Description, Probolem Solved?, EMPLID (fk), MachineID (fk)
Reporting Person	ReportingID, Name, Phone, Email
Payment	PaymentID, Payment Date, Payment Time, Amount, Change Given, Payment Method (E.g: Cash, Debit Card, Credit Card), Card Number, ZipCode, TransactionID (fk)
Product Details	ProductID, Product Name, Category, Selling Price per Product, Buying Price per Product, SupplierID (fk))

Shipment	ShipmentID. Shipment Date, Shipment Quantity, SupplierID (fk), InventoryID (fk)
Supplier	SupplierID, Supplier Name, Phone, Emai, Address
Product Order	OrderID, Order Date, ProductID, Quantity, Total Amount, Buying Price per Product, Order Product Number, EmployeeID (fk), SupplierID (fk)
Employee	EmployeeID, SSN, Name, Phone, Email, Position
Filling	FillingID, Filling DateTime, Quantity, EmployeeID (fk), MachineID (fk)
Product Inventory	InventoryID , Quantity in Machine, Quantity in Inventory, DateTime
Product_Machine	MachineProductID, MachineID (fk), ProductID (fk)
Product_Filling	ProductFillingID, ProductID (fk), FillingID (fk)
Product_Shipment	ProductShipmentID, ProductID (fk), ShipmentID (fk)
Inventory_Product	InventoryProductID, InventoryID (fk), ProductID (fk)
Transaction	TransactionID, Transaction Date, Transaction Time, Quantity purchased, Promotion?, Total Amount, Product Price, Product Name, Transaction Status, MachineID (fk), ProductID (fk)

III. Normalization and Relational Model

1. Normalization

a. Payment

- 1NF: Payment (PaymentID, Payment Date, Payment Time, Amount, Change Given, Payment Method (E.g: Cash, Debit Card, Credit Card), Card Number, ZipCode, TransactionID (fk))
- For 2NF, there is no partial dependency.
- For 3NF, I remove transitive dependent attributes from the relation, and create a new relation based on those attributes with a copy of their determinant.
    - Payment (PaymentID, Payment Date, Payment Time, Amount, Change Given, Card Number, TransactionID (fk))
    - Card (Card Number, Payment Method, ZipCode)
- I will create a new relation called “Card”.

b. Product Order

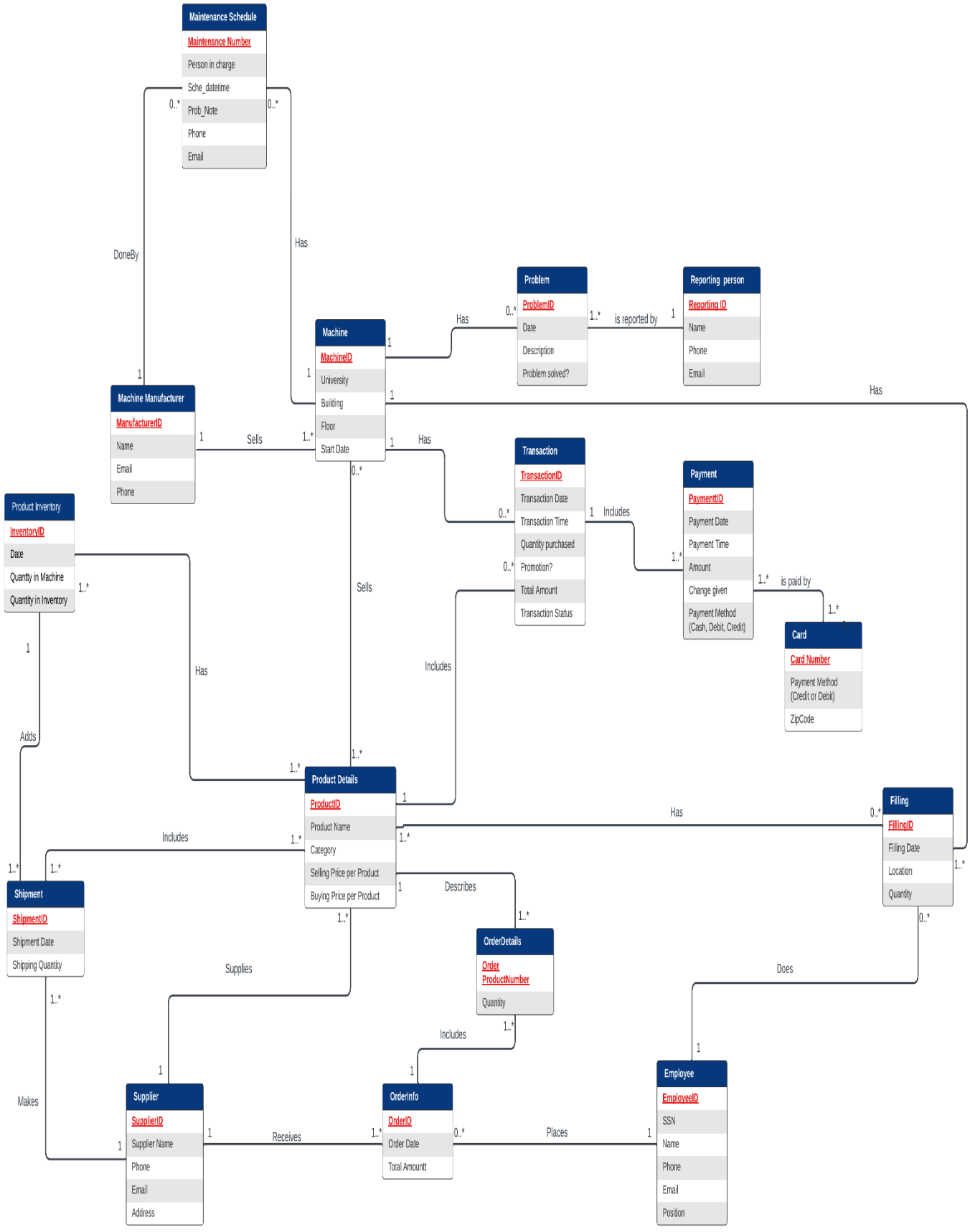
- 1NF: Product Order (OrderID, Order Date, Order Product Number, ProductID, Product Name, Quantity, Total Amount, Buying Price per Product, EmployeeID (fk), SupplierID (fk))
- For 2NF, there is no partial dependency.
- For 3NF, I remove transitive dependent attributes from the relation, and create a new relation based on those attributes with a copy of their determinant.
    - Order Info (OrderID, Order Date, Order Product Number, ProductID, Quantity, Total Amount, Buying Price per Product, EmployeeID (fk), SupplierID (fk))
    - Order Details (Order Product Number, Quantity)
    - Product (Product ID, Product Name, Buying Price per Product)
- I will combine Product relation with the Product Details Relation that already exists, and create a new relation called Order Details.

c. **Transaction**

- 1NF: Transaction (TransactionID, Transaction Date, Transaction Time, Quantity purchased, Promotion?, Total Amount, Product Price (Buying), Product Name, Transaction Status, MachineID (fk), ProductID (fk))
- For 2NF, there is no partial dependency.
- For 3NF, I remove transitive dependent attributes from the relation, and create a new relation based on those attributes with a copy of their determinant.
    - Transaction (TransactionID, Transaction Date, Transaction Time, Quantity Purchased, Promotion?, Total Amount, Transaction Status, MachineID (fk), ProductID (fk))
    - Product (Product ID, Product Name, Buying Price per Product)
- I will combine Product relation with the Product Details Relation that already existed.

2. Entities and Attributes after Normalization

Entities	**Attributes**
Machine	MachineID, University, Building, Floor, Start Date
Machine Manufacturer	ManufacturerID, Name, Email, Phone
Maintenance Schedule	Maintenance Number, Person in charge, Sche_Datetime, Prob_Note, Phone, Email
Problem	ProblemID, Date, Description, Probolem Solved?
Reporting Person	ReportingID, Name, Phone, Email
Payment	PaymentID, Payment Date, Payment Time, Amount, Change Given, Payment Method (E.g: Cash, Debit Card, Credit Card)
Product Details	ProductID, Product Name, Category, Selling Price per Product, Buying Price per Product
Shipment	ShipmentID. Shipment Date, Shipment Quantity
Supplier	SupplierID, Supplier Name, Phone, Emai, Address
Order Info	OrderID, Order Date, Total Amount
Employee	EmployeeID, SSN, Name, Phone, Email, Position
Filling	FillingID, Filling DateTime, Quantity
Product Inventory	InventoryID, DateTime, Quantity in Machine, Quantity in Inventory
Card	Card Number, Credit/Debit?, ZipCode
Order Details	Order Product Number, Quantity
Transaction	TransactionID, Transaction Date, Transaction Time, Quantity Purchased, Promotion?, Total Amount, Transaction Status


Maintenance Schedule
Maintenance Number
Person in charge
Sche_datetime
Prob_Note
Phone
Email
DoneBy
Has
Problem
ProblemID
Date
Description
Problem solved?
Has
is reported by
Reporting_person
Reporting ID
Name
Phone
Email
Machine
MachineID
University
Building
Floor
Start Date
Machine Manufacturer
ManufacturerID
Name
Email
Phone
Sells
Has
Has
Transaction
TransactionID
Transaction Date
Transaction Time
Quantity purchased
Promotion?
Total Amount
Transaction Status
Includes
Payment
PaymentID
Payment Date
Payment Time
Amount
Change given
Payment Method (Cash, Debit, Credit)
is paid by
Card
Card Number
Payment Method (Credit or Debit)
ZipCode
Product Inventory
InventoryID
Date
Quantity in Machine
Quantity in Inventory
Has
Sells
Includes
Adds
Product Details
ProductID
Product Name
Category
Selling Price per Product
Buying Price per Product
Has
Filling
FillingID
Filling Date
Location
Quantity
Includes
Describes
Shipment
ShipmentID
Shipment Date
Shipping Quantity
Supplies
OrderDetails
Order ProductNumber
Quantity
Does
Includes
Makes
Supplier
SupplierID
Supplier Name
Phone
Email
Address
Receives
OrderInfo
OrderID
Order Date
Total Amountt
Places
Employee
EmployeeID
SSN
Name
Phone
Email
Position

3. Relational Model after Normalization

Relation	Attribute
Machine	MachineID, University, Building, Floor, Start Date, ManufaturerID (fk)
Machine Manufacturer	ManufacturerID, Name, Email, Phone
Maintenance Schedule	Maintenance Number, Person in charge, Sche_Datetime, Prob_Note, Phone, Email, MachineID (fk), ManufaturerID(fk)
Problem	ProblemID, Date, Description, Probolem Solved?, ReportingID (fk), MachineID (fk)
Reporting Person	ReportingID, Name, Phone, Email
Payment	PaymentID, Payment Date, Amount, Change Given, Payment Method (E.g: Cash, Debit Card, Credit Card), TransactionID (fk)
Product Details	ProductID, Product Name, Category, Selling Price per Product, Buying Price per Product, SupplierID (fk))
Shipment	ShipmentID. Shipment Date, Shipment Quantity, SupplierID (fk), InventoryID (fk)
Supplier	SupplierID, Supplier Name, Phone, Emai, Address
Order Info	OrderID, Order Date, Total Amount, EmployeeID (fk), SupplierID (fk)
Order Details	Order Product Number, Quantity, ProductID (fk), OrderID (fk)
Employee	EmployeeID, SSN, Name, Phone, Email, Position
Filling	FillingID, Filling Datetime, Quantity, EmployeeID (fk), MachineID (fk)
Product Inventory	InventoryID , Quantity in Machine, Quantity in Inventory, DateTime
Transaction	TransactionID, Transaction Date, Quantity Purchased, Promotion?, Total Amount, Transaction Status, MachineID (fk), ProductID (fk)
Product_Machine	MachineProductID, MachineID (fk), ProductID (fk)
Product_Filling	ProductFillingID, ProductID (fk), FillingID (fk)
Product_Shipment	ProductShipmentID, ProductID (fk), ShipmentID (fk)
Inventory_Product	InventoryProductID, InventoryID (fk), ProductID (fk)
Card	CardNumber, Credit/Debit?, ZipCode
Payment_Card	PaymentCard, Card Number (fk), PaymentID (fk)

IV. Creating the Database Schema

```
CREATE TABLE Card (
CardNumber VARCHAR(20) NOT NULL,
Credit_Debit BIT NULL,
ZipCode VARCHAR(10) NOT NULL,
PRIMARY KEY (CardNumber)
);
```

```
CREATE TABLE Machine_Manufacturer (
ManufacturerID INT NOT NULL,
Name VARCHAR(45) NULL,
Email VARCHAR(45) NULL,
Phone VARCHAR(10) NULL,
PRIMARY KEY (ManufacturerID)
);

CREATE TABLE Machine (
MachineID INT NOT NULL,
University VARCHAR(45) NULL,
Building VARCHAR(45) NULL,
Floor INT NULL,
StartDate DATETIME NULL,
ManufacturerID INT NOT NULL,
PRIMARY KEY (MachineID),
FOREIGN KEY (ManufacturerID) REFERENCES Machine_Manufacturer(ManufacturerID)
);

CREATE TABLE Maintenance_Schedule (
Maintenance_Number INT NOT NULL,
PersonInCharge VARCHAR(45) NULL,
Sche_Datetime DATETIME NULL,
Phone VARCHAR(10) NULL,
Email VARCHAR(45) NULL,
Prob_Note TEXT NOT NULL,
ManufacturerID INT NOT NULL,
MachineID INT NOT NULL,
PRIMARY KEY (Maintenance_Number),
FOREIGN KEY (ManufacturerID) REFERENCES Machine_Manufacturer(ManufacturerID),
FOREIGN KEY (MachineID) REFERENCES Machine(MachineID)
);

CREATE TABLE Reporting_Person (
ReportingID INT NOT NULL,
Name VARCHAR(45) NULL,
Email VARCHAR(45) NULL,
Phone VARCHAR(10) NULL,
PRIMARY KEY (ReportingID)
);
```

```
CREATE TABLE Problem (
ProblemID INT NOT NULL,
ReportDate DATETIME NULL,
Description TEXT NULL,
Problem_Solved BIT NOT NULL,
EMPLID INT NOT NULL,
MachineID INT NOT NULL,
PRIMARY KEY (ProblemID),
FOREIGN KEY (ReportingID) REFERENCES Reporting_Person(ReportingID),
FOREIGN KEY (MachineID) REFERENCES Machine(MachineID)
);

CREATE TABLE Supplier (
SupplierID INT NOT NULL,
Phone VARCHAR(10) NULL,
Supplier_Name VARCHAR(45) NULL,
Emai VARCHAR(45) NULL,
Address TEXT NULL,
PRIMARY KEY (SupplierID)
);

CREATE TABLE Product_Details (
ProductID INT NOT NULL,
Category VARCHAR(45) NULL,
Product_Name VARCHAR(45) NOT NULL,
Selling_Price FLOAT NOT NULL,
Buying_Price FLOAT NOT NULL,
SupplierID INT NOT NULL,
PRIMARY KEY (ProductID),
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
);

CREATE TABLE Transaction (
TransactionID INT NOT NULL,
TransactionDate DATETIME NULL,
PromotionCode VARCHAR(10) NULL,
TotalAmount FLOAT NULL,
QuantityPurchased INT NULL,
TransactionStatus VARCHAR(10) NULL,
MachineID INT NOT NULL,
ProductID INT NOT NULL,
```

```
PRIMARY KEY (TransactionID),
FOREIGN KEY (MachineID) REFERENCES Machine(MachineID),
FOREIGN KEY (ProductID) REFERENCES Product_Details(ProductID)
);

CREATE TABLE Payment (
PaymentID INT NOT NULL,
Payment_Date DATETIME NOT NULL,
Payment_Method VARCHAR NOT NULL,
Amount FLOAT NULL,
Change_Given FLOAT NULL,
TransactionID INT NOT NULL,
PRIMARY KEY (PaymentID),
FOREIGN KEY (TransactionID) REFERENCES Transaction(TransactionID)
);

CREATE TABLE Payment_Card (
PaymentCard INT NOT NULL,
CardNumber VARCHAR NOT NULL,
PaymentID INT NOT NULL,
PRIMARY KEY (PaymentCard),
FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID),
FOREIGN KEY (CardNumber) REFERENCES Card(CardNumber)
);

CREATE TABLE Product_Inventory (
InventoryID INT NOT NULL,
DateTime_Inventory DATETIME NOT NULL,
Quantity_in_Machine INT NULL,
Quantity_in_Inventory INT NULL,
PRIMARY KEY (InventoryID)
);

CREATE TABLE Shipment (
ShipmentID INT NOT NULL,
Shipment_Date DATETIME NULL,
Shipment_Quantity INT NULL,
SupplierID INT NOT NULL,
InventoryID INT NOT NULL,
PRIMARY KEY (ShipmentID),
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),
FOREIGN KEY (InventoryID) REFERENCES Product_Inventory(InventoryID)
```

```
);

CREATE TABLE Order_Info (
OrderID INT NOT NULL,
Order_Date DATETIME NULL,
Total_Amount FLOAT NULL,
EmployeeID INT NOT NULL,
SupplierID INT NOT NULL,
PRIMARY KEY (OrderID),
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
);

CREATE TABLE Order_Details (
OrderProductNumber INT NOT NULL,
Quantity INT NULL,
ProductID INT NOT NULL,
OrderID INT NOT NULL,
PRIMARY KEY (OrderProductNumber),
FOREIGN KEY (ProductID) REFERENCES Product_Details(ProductID),
FOREIGN KEY (OrderID) REFERENCES Order_Info(OrderID)
);

CREATE TABLE Filling (
FillingID INT NOT NULL,
Filling_Date DATETIME NULL,
Quantity INT NULL,
EmployeeID INT NOT NULL,
MachineID  INT NOT NULL,
PRIMARY KEY (FillingID),
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
FOREIGN KEY (MachineID) REFERENCES Machine(MachineID)
);

CREATE TABLE Employee (
EmployeeID INT NOT NULL,
SSN VARCHAR(20) NOT NULL,
Name VARCHAR(50) NOT NULL,
Phone VARCHAR(20) NULL,
Email VARCHAR(50) NOT NULL,
Position VARCHAR(50) NOT NULL,
FillingID INT NOT NULL,
```

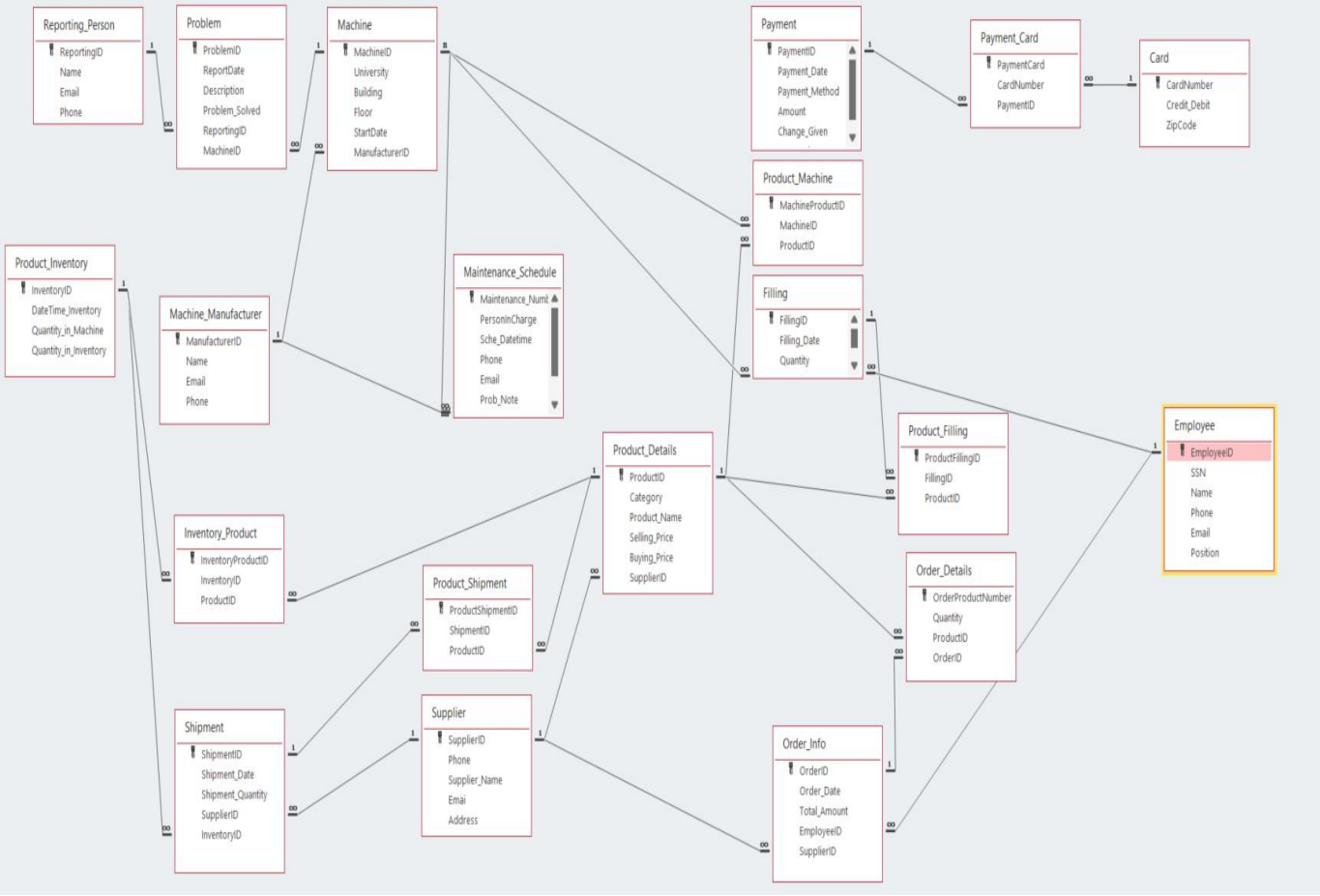
```
OrderID INT NOT NULL,
PRIMARY KEY (EmployeeID),
FOREIGN KEY (OrderID) REFERENCES Order_Info(OrderID),
FOREIGN KEY (FillingID) REFERENCES Filling(FillingID)
);

CREATE TABLE Product_Filling  (
ProductFillingID INT NOT NULL,
FillingID INT NOT NULL,
ProductID INT NOT NULL,
PRIMARY KEY (ProductFillingID),
FOREIGN KEY (FillingID) REFERENCES Filling(FillingID),
FOREIGN KEY (ProductID) REFERENCES Product_Details(ProductID)
);

CREATE TABLE Product_Machine  (
MachineProductID INT NOT NULL,
MachineID INT NOT NULL,
ProductID INT NOT NULL,
PRIMARY KEY (MachineProductID),
FOREIGN KEY (MachineID) REFERENCES Machine(MachineID),
FOREIGN KEY (ProductID) REFERENCES Product_Details(ProductID)
);

CREATE TABLE Product_Shipment  (
ProductShipmentID INT NOT NULL,
ShipmentID INT NOT NULL,
ProductID INT NOT NULL,
PRIMARY KEY (ProductShipmentID),
FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID),
FOREIGN KEY (ProductID) REFERENCES Product_Details(ProductID)
);

CREATE TABLE Inventory_Product  (
InventoryProductID INT NOT NULL,
InventoryID INT NOT NULL,
ProductID INT NOT NULL,
PRIMARY KEY (InventoryProductID),
FOREIGN KEY (InventoryID) REFERENCES Product_Inventory(InventoryID),
FOREIGN KEY (ProductID) REFERENCES Product_Details(ProductID)
);
```



V. Conclusion

Initially, I had some issues with my ERD since I was unsure and confused of which attributes to put into the entities. This made it challenging for me to manage the normalization process afterwards. I had to make a lot of adjustments to the relational models and ERD at the same time to leverage the benefits of normalization.

As a result, CUNY now can effectively track all the information relating to the outflow and inflow of the vending machines' products thanks to the optimized database. Without a doubt, this system will offer a lot of benefits to the CUNY community, which includes students and staff using the vending machines as well as the school's management in controlling costs and inventory.