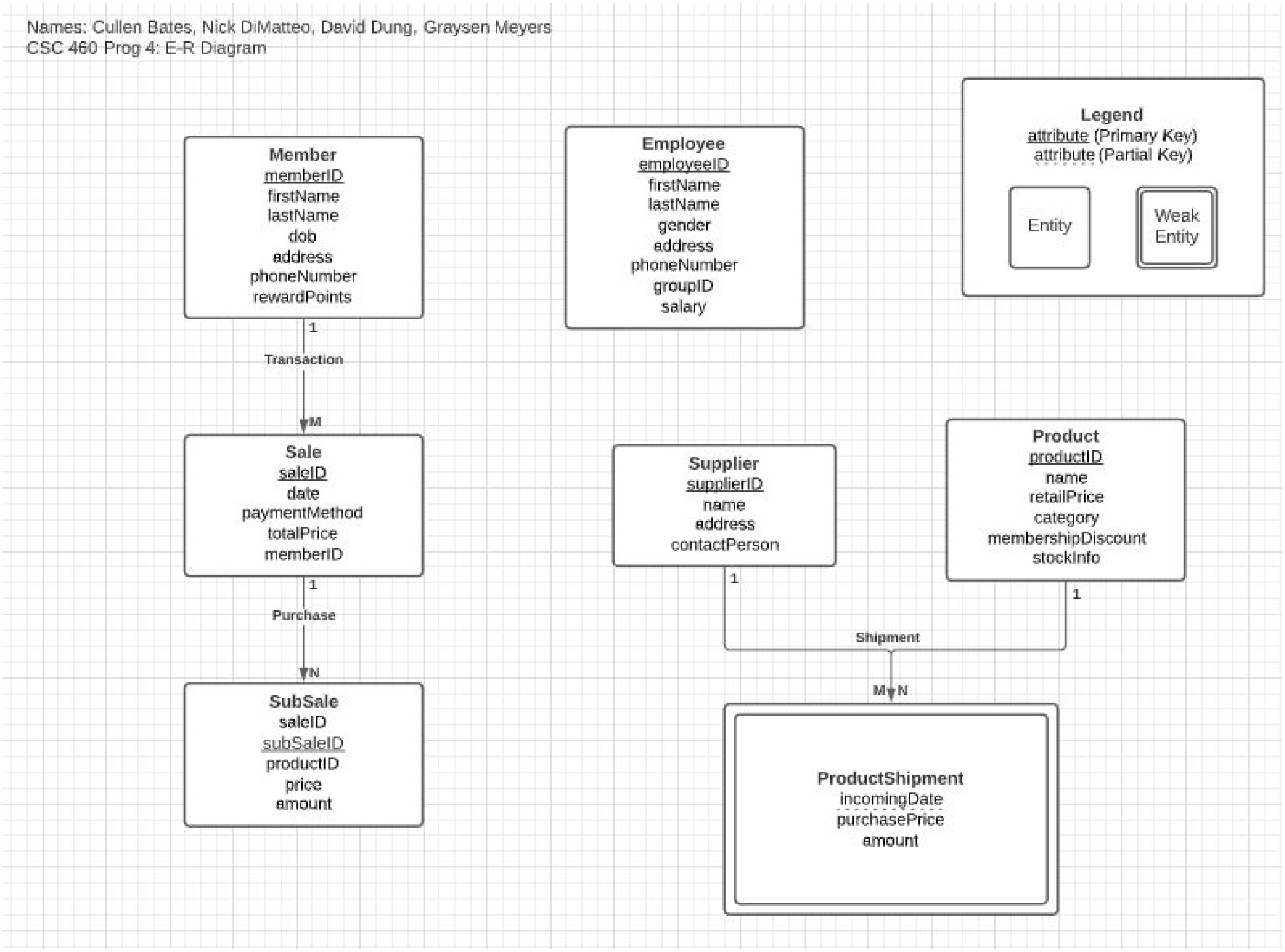
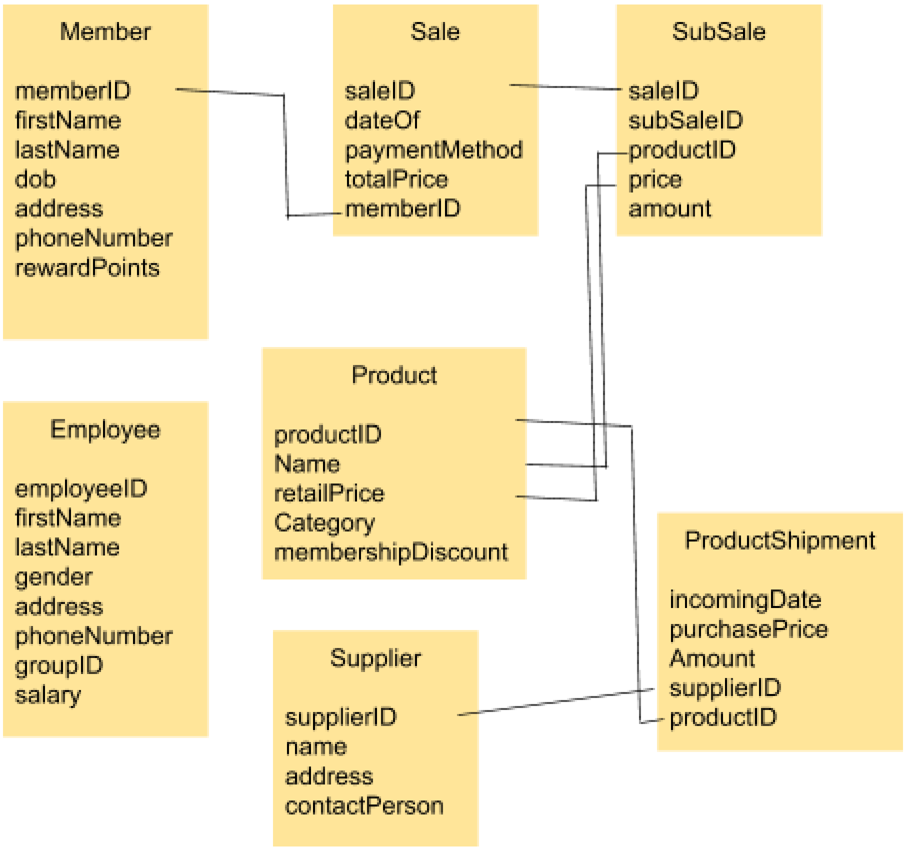
Program 4 Documentation

1. Conceptual database design: Final ER diagram along with your design rationale and any necessary high–level text description of the data model (e.g., constraints or anything not able to show in the ER diagram but is necessary to help people understand your database design).



2. Logical database design: Converting an ER schema into a relational database schema. Show the schemas of the tables resulted in this step.



Member

memberID
firstName
lastName
dob
address
phoneNumber
rewardPoints

Sale

saleID
dateOf
paymentMethod
totalPrice
memberID

SubSale

saleID
subSaleID
productID
price
amount

Employee

employeeID
firstName
lastName
gender
address
phoneNumber
groupID
salary

Product

productID
Name
retailPrice
Category
membershipDiscount

Supplier

supplierID
name
address
contactPerson

ProductShipment

incomingDate
purchasePrice
Amount
supplierID
productID

3. Normalization analysis: Show the FDs of all your tables and justify why your design adheres to 3NF.

Member	none
Sale	memberID
SubSale	saleID,productID,price
Employee	none
Supplier	none
Product	none
ProductShipment	supplierID,productID

1NF requires that per column of a table, we cannot hold multiple values in a single attribute. Our tables all follow this rule, so all of our tables are therefore 1NF.

2NF requires that the table first be in 1NF, which we have already established is true. It also requires that no non-prime attribute is dependent on the proper subset of any candidate key. All of our tables adhere to that rule, which makes our tables 2NF.

3NF, finally, requires that the table first be 2NF, which of course implies 1NF as well. The final requirement of 3NF is that any transitive functional dependency of non-prime attributes on super keys have to be removed. An example of this could be an address stored in a zip, state and city field. The state and city would be transitively depending on the zip code. To be 3NF, we had to remove all instances of this kind of dependency in our tables.

4. Query description: Describe your created query. Specifically, what question is it answering? What is the significance of including such a query in the system?

Our created query is to display all members that have a given id to the user. This is very useful for a user to locate a members information given only the id. There are many situations where one might want to locate a members info. For example, a scenario where a member is looking to verify that we have the correct address or phone number in our data. This query solves the question of "which member does this ID belong to, and what is all of their personal information?". This makes our system more accessible to users, increasing its functionality and use cases.