Background: A shoe store has been collecting records of customer, shoe, employee, and transaction information by hand for the 5 years it has been in business. The owner would like to move all the data and future information into a relational database management system as it is getting too time consuming to search and write down all the information. He comes to you for help in building and implementing a relational model for his business.

After talking to the owner, you have created an entity relationship model.

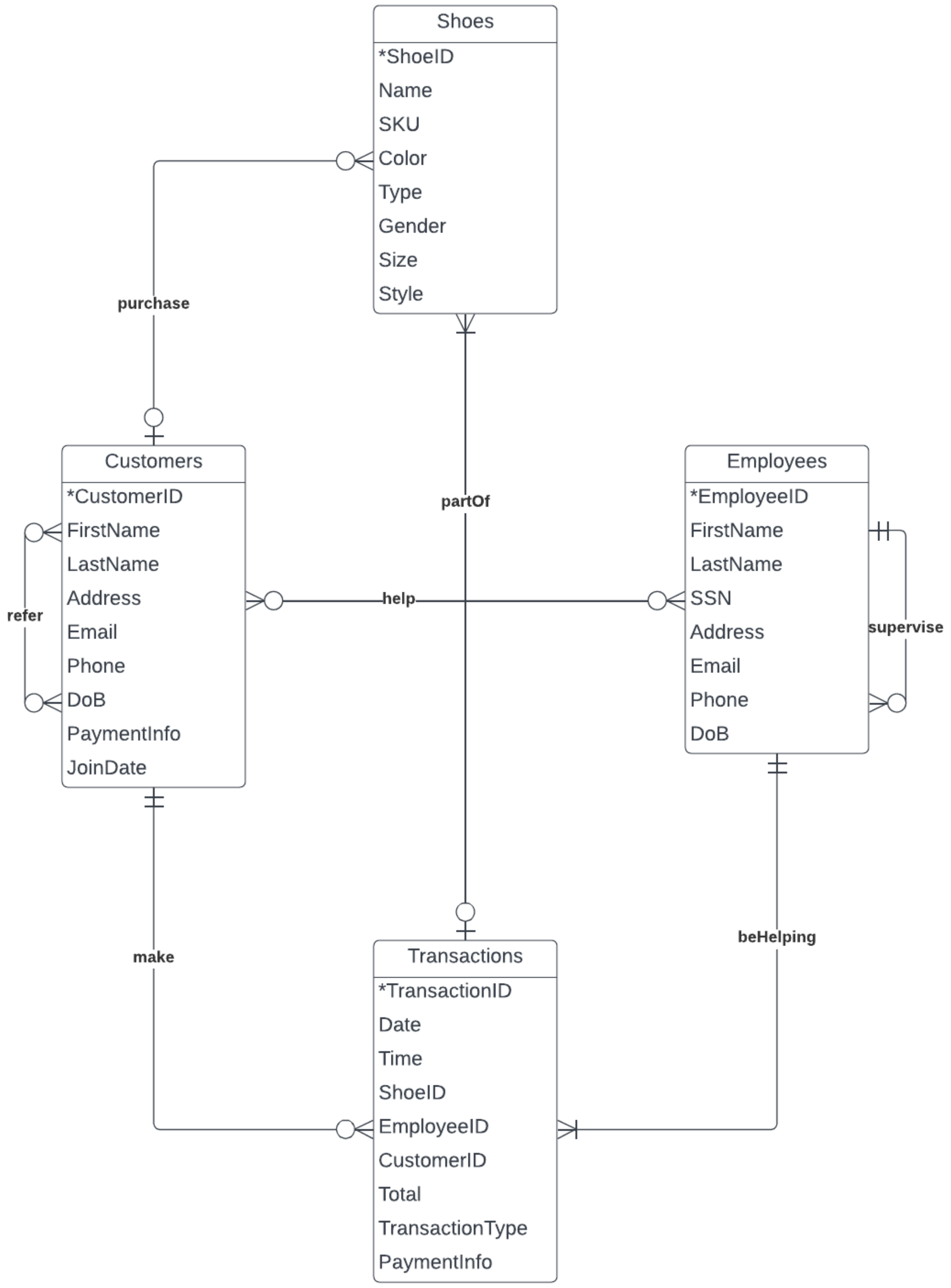
The entities are:

- Customers: CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate. CustomerID is the identifier.
- Shoes: ShoeID, Name, SKU, Color, Type, Gender, Size, Style. ShoeID is the identifier.
- Employees: EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB. EmployeeID is the identifier.
- Transactions: TransactionID, Date, Time, ShoeID, EmployeeID, CustomerID, Total, TransactionType, PaymentInfo. TransactionID is the identifier.

The relationships are:

- A customer may purchase one or more shoes; A shoe may belong to one and only one customer.
- An employee may help one or more customers; A customer may be helped by one or more employees.
- A customer may make one or more transactions; A transaction must be taken by one and only one customer.
- A shoe may be a part of one and only one transaction; A transaction must include one or more shoes.
- An employee must be helping with one or more transactions; A transaction must be helped by one and only one employee.
- A customer may be referred by one or more customers; A customer may refer one or more customers.
- An employee must have one and only one supervisor. A supervisor may supervise one or more employees.

This is the entity relationship diagram:



Here is the relational model:

- Shoes (ShoeID, Name, SKU, Color, Type, Gender, Size, Style, CustomerID(fk), TransactionID(fk))
- Customers (CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate)
- Customers_Customers (CustomerID(fk), CustomerIDA(fk))
- Employees (EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, EmployeeIDA(fk))
- Customers_Employees (CustomerID(fk), EmployeeID(fk))
- Transactions (TransactionID, Date, Time, ShoeID(fk), EmployeeID(fk), CustomerID(fk), Total, TransactionType, PaymentInfo)

After talking to the owner, you received the following functional dependencies:

- Shoes (ShoeID, Name, SKU, Color, Type, Gender, Size, Style, CustomerID(fk), TransactionID(fk))
    - FD1: ShoeID -> Name, SKU, Color, Type, Gender, Size, Style, CustomerID, TransactionID
- Customers (CustomerID, FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate)
    - FD1: CustomerID -> FirstName, LastName, Address, Email, Phone, DoB, PaymentInfo, JoinDate
    - FD2: FirstName, LastName, Address -> Email, Phone, DoB, PaymentInfo, JoinDate
- Customers_Customers (CustomerID(fk), CustomerIDA(fk))
    - There is no non-primary-key attribute
- Employees (EmployeeID, FirstName, LastName, SSN, Address, Email, Phone, DoB, EmployeeIDA(fk))
    - FD1: EmployeeID -> FirstName, LastName, SSN, Address, Email, Phone, DoB, EmployeeIDA
    - FD2: SSN -> FirstName, LastName
- Customers_Employees (CustomerID(fk), EmployeeID(fk))
    - There is no non-primary-key attribute
- Transactions (TransactionID, Date, Time, ShoeID(fk), EmployeeID(fk), CustomerID(fk), Total, TransactionType, PaymentInfo)
    - FD1: TransactionID -> Date, Time, ShoeID, EmployeeID, CustomerID, Total, TransactionType, PaymentInfo

Time to normalize all relations to 3NF:

- Shoes is already in 3NF since it is a relation and only has one functional dependency.
- Transactions is in 3NF for the same reason.
- Customers_Customers is in 3NF since it is a relation and there are no functional dependencies.
- Customers_Employees is in 3NF for the same reason.

- Customers is in 2NF since it has no partial functional dependencies. However, it is not in 3NF because it has a transitive functional dependency. We will have to remove FD2, make FirstName, LastName, and Address as foreign keys in the original relation and remove Email, Phone, DoB, PaymentInfo, and JoinDate from Customers. We will create a new relation called CustomersA, making FirstName, LastName, Address as the primary key and bring FD2 from the original relation to this new relation as FD1.
- Employees is in 2NF since there are no partial functional dependencies. However, it has a transitive functional dependency so it cannot be in 3NF. To normalize it to 3NF, we remove FD2, which causes the transitive functional dependency. Then, we make SSN a foreign key in the original relation. We will remove FirstName and LastName from the relation. We will create a new relation called EmployeesA which will have SSN as the primary key. We will have FD2 from the original relation, which we removed, as FD1 in this new relation.

Here is the final relational model in 3NF:

- Shoes (<u>ShoeID</u>, Name, SKU, Color, Type, Gender, Size, Style, CustomerID(fk), TransactionID(fk))
    - FD1: ShoeID -> Name, SKU, Color, Type, Gender, Size, Style, CustomerID, TransactionID
- Customers (<u>CustomerID</u>, FirstName(fk), LastName(fk), Address(fk))
    - FD1: CustomerID -> FirstName, LastName, Address
- CustomersA (<u>FirstName</u>, <u>LastName</u>, <u>Address</u>, Email, Phone, DoB, PaymentInfo, JoinDate)
    - FD1: FirstName, LastName, Address -> Email, Phone, DoB, PaymentInfo, JoinDate
- Customers_Customers (<u>CustomerID(fk)</u>, <u>CustomerIDA(fk))</u>
    - There is no non-primary-key attribute
- Employees (<u>EmployeeID</u>, SSN(fk), Address, Email, Phone, DoB, EmployeeIDA(fk))
    - FD1: EmployeeID -> SSN, Address, Email, Phone, DoB, EmployeeIDA
- EmployeesA (<u>SSN</u>, FirstName, LastName)
    - FD1: SSN -> FirstName, LastName
- Customers_Employees (<u>CustomerID(fk)</u>, <u>EmployeeID(fk)</u>)
    - There is no non-primary-key attribute
- Transactions (<u>TransactionID</u>, Date, Time, ShoeID(fk), EmployeeID(fk), CustomerID(fk), Total, TransactionType, PaymentInfo)
    - FD1: TransactionID -> Date, Time, ShoeID, EmployeeID, CustomerID, Total, TransactionType, PaymentInfo