The Final Project

Design a Normalized Relational Database for a Cosmetic Company.

A local cosmetic company has been very successful in recent years. They decided to expand their business outside the city. A database system for inner Management is crucial for the expansion of the business.

Entity Relationship Model

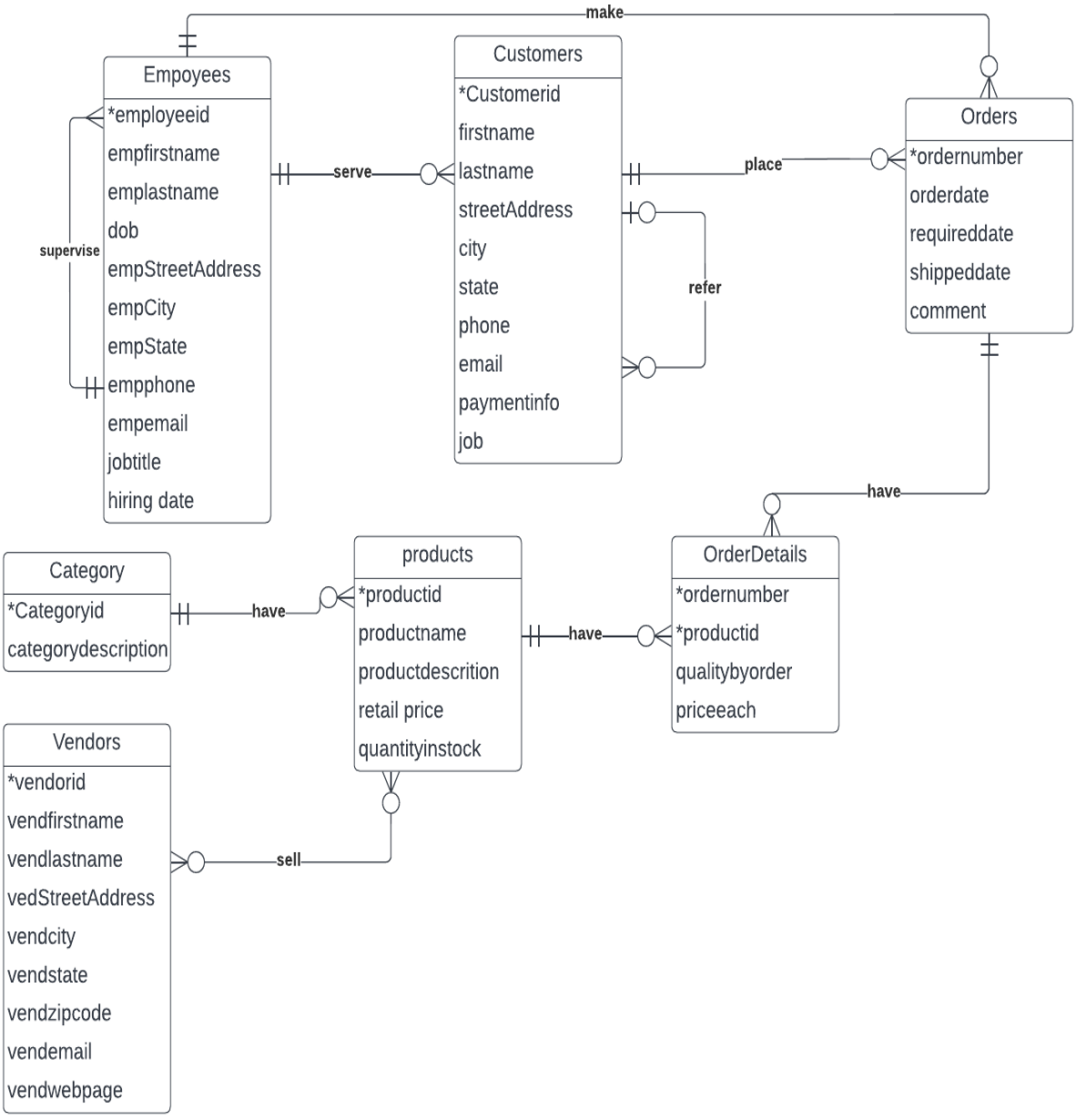
The information needed is: the Entities, attributes, identifiers and relationship(cardinality(how many), participation (optional or mandatory)).

- **Customers**: customerid, firstname, lastname, streetAddress, city, state, phone, email, job, paymentinfo. Customerid is the identifier.
- **Employees**: employeeid, empfirstname, emplastname, dob, empphone, empemail, empstreetAddress, empCity, empState jobtitle, hiringdate. Employeeid is the identifier.
- **Orders:** ordernumber, orderdate, requireddate, shippeddate, comments. Ordernumber is the identifier.
- **Orderdetails:** ordernumber, productid, quntitybyorder, priceeach. (Ordernumber+ productid) is the identifier.
- **Products:** productid, productname, productdescription, retailprice, quantityinstock. Productid is the identifier.
- **Categories:** categoryid, categorydescription. Categoryid is the identifier.

- **Vendors:** Vendid**,** VendFirstname, VendLastname, VenStreetAddress, VendCity, VendState, Vendzipcode, VendWebpage. Vendeid is the identifier.

  The relationships among entities based on my assumptions:
- An employee may serve one or more customers. A customer must be served by one and only one employee.
- An employee may supervise one or more employees. An employee can be supervised by one and only one employee.
- An order must be made by one and only one employee. An employee may make one or more orders.
- A customer may place one or more orders. An order must be placed by one and only one customer.
- A customer may be referred by one (zero or one) customer; and a customer may refer one or more (one or more) customers.
- An order might have one or more orderdetail. An orderdetail must have one and only one order.
- One category may have one or more products. A product belongs to one and only one category.
- A vendor can sell one or more products. A product can be sold by one or more vendors.

Creating the Entity Relationship Diagram (ERD) from Entity Relationship Model Using Crow's Foot Notation.



Converting the ERD to a Relational Model:

Converting an ERD to a Relational Model is by converting:

- Entities to Relations (each relation must satisfy the properties of relation to be considered as a relation)
- Attributes to Attributes
- Identifier to Primary key
- Relationship to Foreign Key.

The relational model converted from the ERD is:

- **Customers** (CustomerID, Firstname, Lastname, StreetAddress, City, State, Phone, Email, Job, Paymentinfo, Employeeid(fk), ReferByCustomer(fk)).
- **Employees** (EmployeeID, Empfirstname, Emplastname, DoB, EmpPhone, EmpEmail, EmpStreetAddress, EmpCity, EmpState, Jobtitle, HiringDate, SupervisorID(fk)).
- **Orders (**Ordernumber, Orderdate, Requireddate, Shippeddate, Comments, EmployeeID(fk), CustomerID(fk)).
- **Orderdetails (**Ordernumber(fk), ProductID(fk), Quntitybyorder, Priceeach, Comment). In this relation we have a composite primary key (Ordernumber+ ProductID).
- **Products (**ProductID, productname, productdescription, RetailPrice, QuantityinStock, CategoryID(fk)).
- **Category** (CategoryID, Categorydescription)
- **Vendors (**Vendid**,** VendFirstname, VendLastname, VenStreetAddress, VendCity, VendState, Vendzipcode, VendWebpage).
- **Vendor_Product (**VendorID(fk), ProductID(fk))

The Functional Dependencies

The Functional Dependencies of the relations based on my assumptions:

- **Customers** (CustomerID, Firstname, Lastname, StreetAddress, City, State, Phone, Email, Job, Paymentinfo, Employeeid(fk), ReferByCustomer(fk)).
    - **FD1**: CustomerID ⟶ Firstname, Lastname, StreetAddress, City, State, Phone, Email, Job, Paymentinfo, Employeeid(fk), ReferByCustomer(fk)
- **Employees** (EmployeeID, Empfirstname, Emplastname, DoB, EmpPhone, EmpEmail, EmpStreetAddress, EmpCity, EmpState, Jobtitle, HiringDate, SupervisorID(fk)).
    - **FD1:** EmployeeID⟶Empfirstname, Emplastname, DoB, EmpPhone, EmpEmail, EmpStreetAddress, EmpCity, EmpState, Jobtitle, HiringDate, SupervisorID(fk)
- **Orders (**Ordernumber, Orderdate, Requireddate, Shippeddate, Comments, EmployeeID(fk), CustomerID(fk)).

  FD1: Ordernumber⟶Orderdate, Requireddate, Shippeddate, Comments, EmployeeID(fk), CustomerID(fk)
- **Orderdetails (**Ordernumber(fk),ProductID(fk), Quntitybyorder, Priceeach, Comment)

  FD1: Ordernumber(fk) + ProductID(fk)⟶Quantitybyorder, Priceeach, Comment.

  FD2: ProductID(fk) ⟶ Priceeach.

  FD3: Ordernumber(fk) ⟶Quantitybyorder

- **Products (**ProductID, productname, productdescription, RetailPrice, QuantityinStock, CategoryID(fk)).

  FD1: ProductID ⟶ productname, productdescription, RetailPrice, QuantityinStock, CategoryID(fk).

  FD2: productname ⟶productdescription

- **Category** (CategoryID, Categorydescription)

FD1: CategoryID ⟶ Categorydescription

- **Vendors (**VendID**,** VendFirstname, VendLastname, VenStreetAddress, VendCity, VendState, Vendzipcode, VendWebpage).
  FD1: Vendid ⟶ VendFirstname, VendLastname, VenStreetAddress, VendCity, VendState, Vendzipcode, VendWebpage.
  FD2: VendFirstname+VendLastname ⟶ VedWebPage.
- **Vendor_Product (**VendorID(fk), ProductID(fk))

Normalize the Relational Model to 3NF

Customers, Employees, Orders, and Category relations are in 1NF, because they are relations; they have no partial functional dependencies, so they are in 2NF; and they have no transitive functional dependencies, so they are in 3NF.

Orderdetails is a relation; so it is in 1NF; priceeach and Quantitybyorder are partially functionally dependent on the composite key(Ordernumber(fk) + productid). ProductID and Ordernumber are subsets of the composite key. So, it is not in 2NF.

The Orderdetails after normalization is:

Orderdetails (Ordernumber(fk),ProductID(fk))

Productprice (ProductID, priceeach)

FD1: productid ⟶ priceeach

quantity(Ordernumber, quantitybyorder)

Ordernumber ⟶ quantitybyorder

Products is a relation. So, it is in 1NF. Every non-primary key attribute is fully functionally dependent on the primary key productid; so, it is in 2NF. It is not in 3NF because there is a transitive FD.

(productid ⟶ productname ⟶ productdescription).

The products after normalization is:

Products (ProductID, productname(fk), RetailPrice, QuantityinStock, CategoryID(fk)).

FD1: ProductID ⟶ RetailPrice, QuantityinStock, CategoryID(fk).

Description (productname, productdescription)

FD1: productname ⟶ productdescription

Vendors is a relation. So, it is the first form. It is in the 2NF because there are no partially FDs. It is not in the 3NF because there is a transitive FD.

(vendid ⟶ VendFirstname, VendLastname ⟶ VendWebpage).

Vendors (VendID, VendFirstname(fk), VendLastname(fk), VenStreetAddress, VendCity, VendState, Vendzipcode).

FD1: Vendid ⟶ VenStreetAddress, VendCity, VendState, Vendzipcode.

Webpage (VendFirstname, VendLastname, VendWebPage)

FD2: VendFirstname+VendLastname ⟶ VendWebPage.

Final Model:

All the relations now are in the 3NF and ready to be implemented.

Customers (CustomerID, Firstname, Lastname, StreetAddress, City, State, Phone, Email, Job, Paymentinfo, Employeeid(fk), ReferByCustomer(fk)).

FD1: CustomerID ⟶ Firstname, Lastname, StreetAddress, City, State, Phone, Email, Job, Paymentinfo, Employeeid(fk), ReferByCustomer(fk).

Employees (EmployeeID, Empfirstname, Emplastname, DoB, EmpPhone, EmpEmail, EmpStreetAddress, EmpCity, EmpState, Jobtitle, HiringDate, SupervisorID(fk)).

FD1: EmployeeID ⟶ Empfirstname, Emplastname, DoB, EmpPhone, EmpEmail, EmpStreetAddress, EmpCity, EmpState, Jobtitle, HiringDate, SupervisorID(fk)

Orders (Ordernumber, Orderdate, Requireddate, Shippeddate, Comments, EmployeeID(fk), CustomerID(fk)).

FD1: Ordernumber → Orderdate, Requireddate, Shippeddate, Comments, EmployeeID(fk), CustomerID(fk)

Category (CategoryID, Categorydescription)

FD1: CategoryID → Categorydescription

Orderdetails (Ordernumber(fk),ProductID(fk))

Productprice (ProductID, priceeach)

FD1: productid → priceeach

quantity (Ordernumber, quantitybyorder)

FD1: Ordernumber → quantitybyorder

Products (ProductID, productname(fk), RetailPrice, QuantityinStock, CategoryID(fk)).

FD1: ProductID → RetailPrice, QuantityinStock, CategoryID(fk).

Description (productname, productdescription)

FD1: productname → productdescription

Vendors (VendID, VendFirstname(fk), VendLastname(fk), VenStreetAddress, VendCity, VendState, Vendzipcode).

FD1: Vendid → VenStreetAddress, VendCity, VendState, Vendzipcode.

Webpage (VendFirstname, VendLastname, VendWebPage)

FD2: VendFirstname+VendLastname → VendWebPage.

Vendor_Product (VendorID(fk), ProductID(fk))