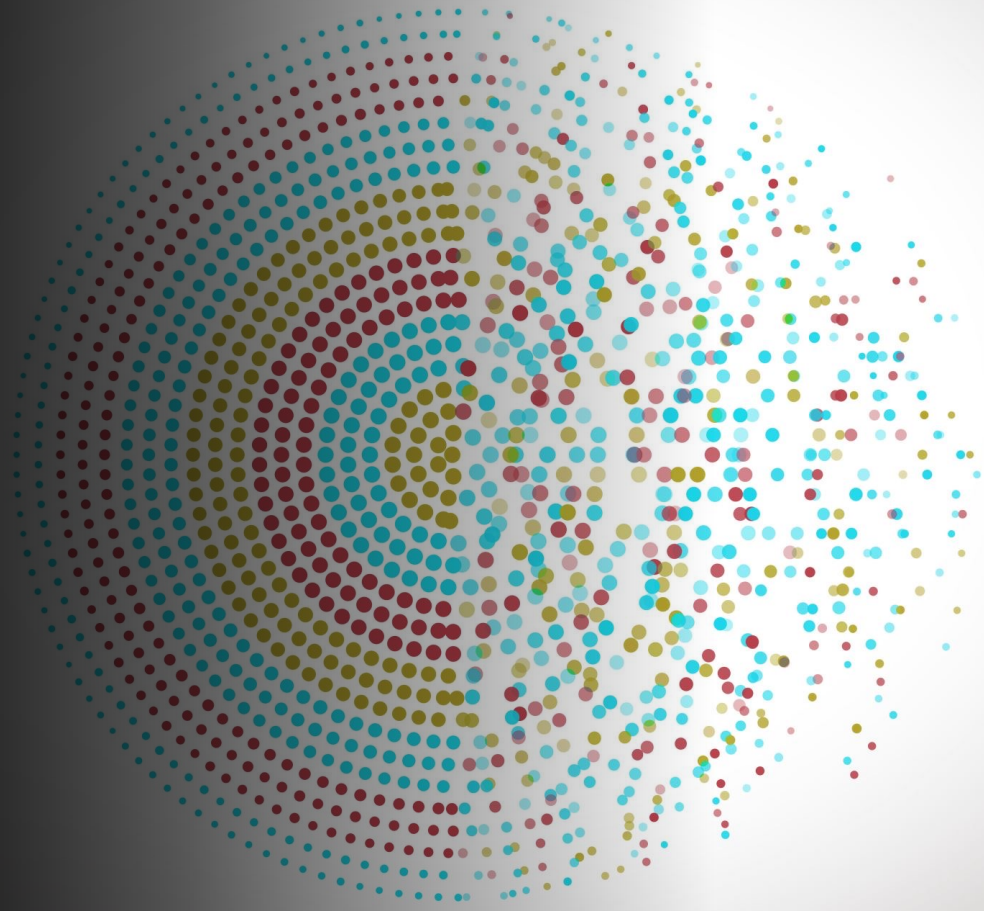
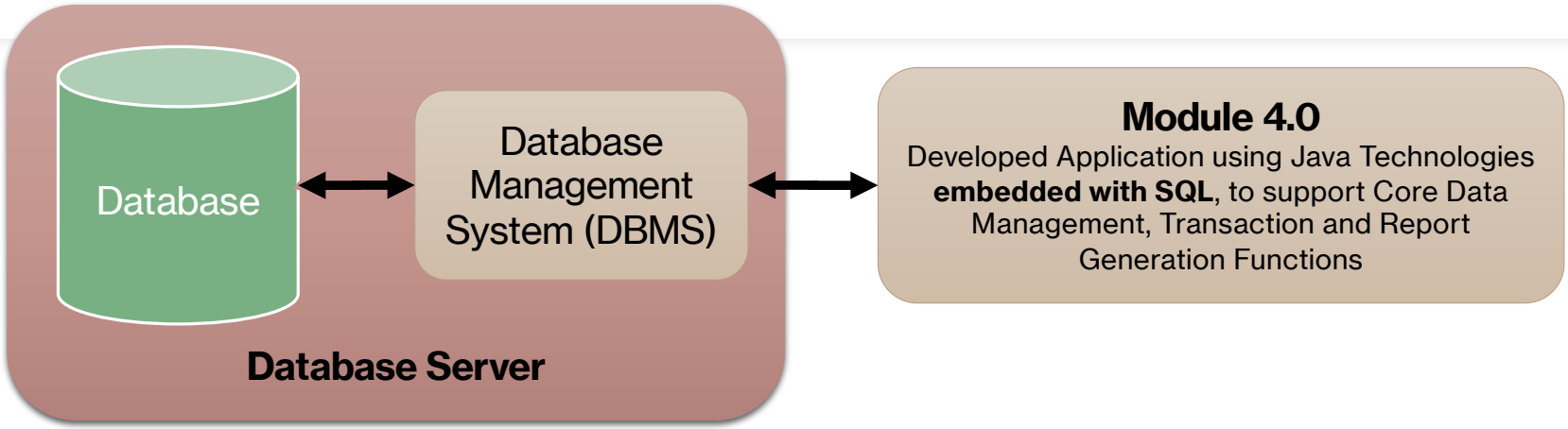
Anatomy of a Basic

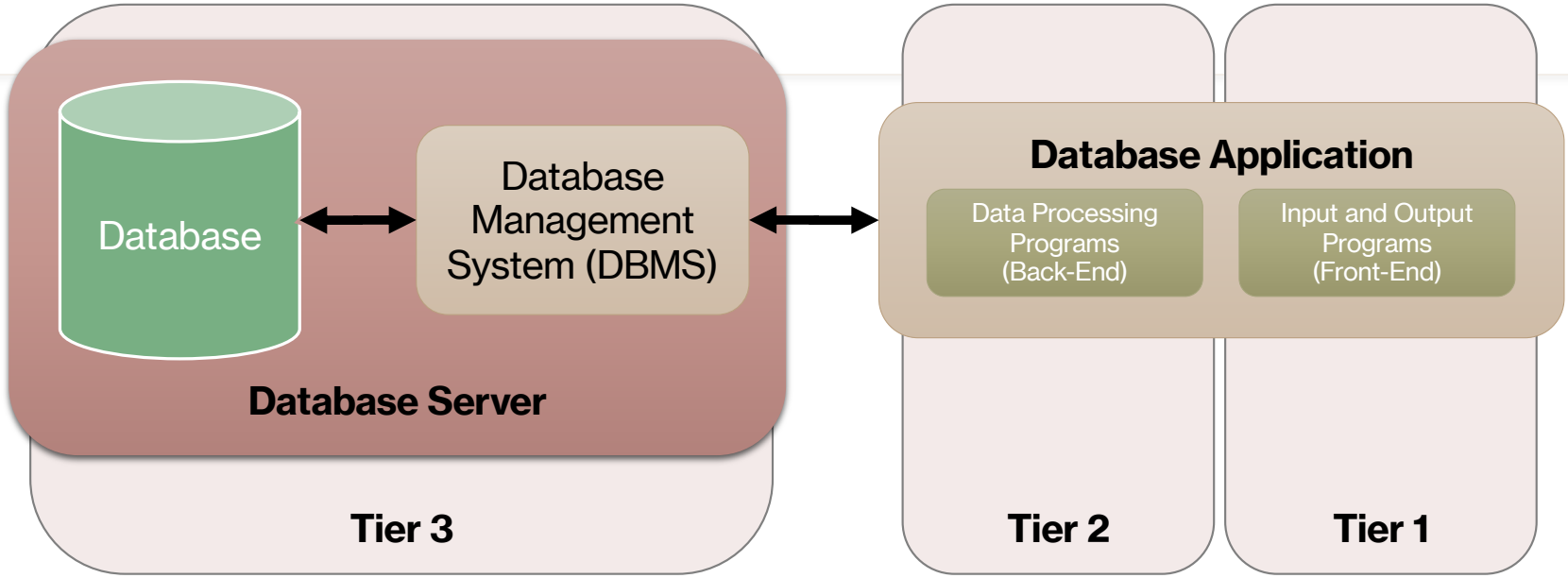
DB Application



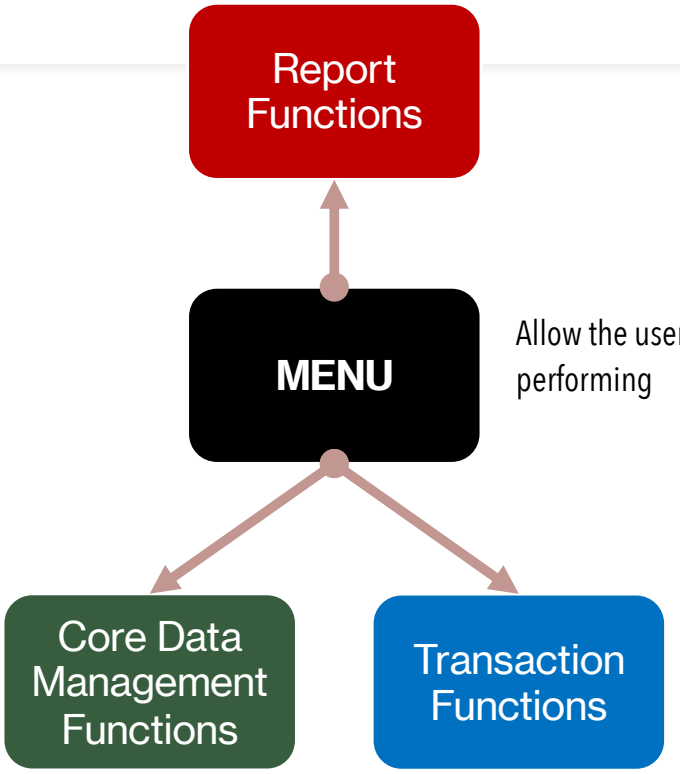
Components of a Database System



Components of a Database System

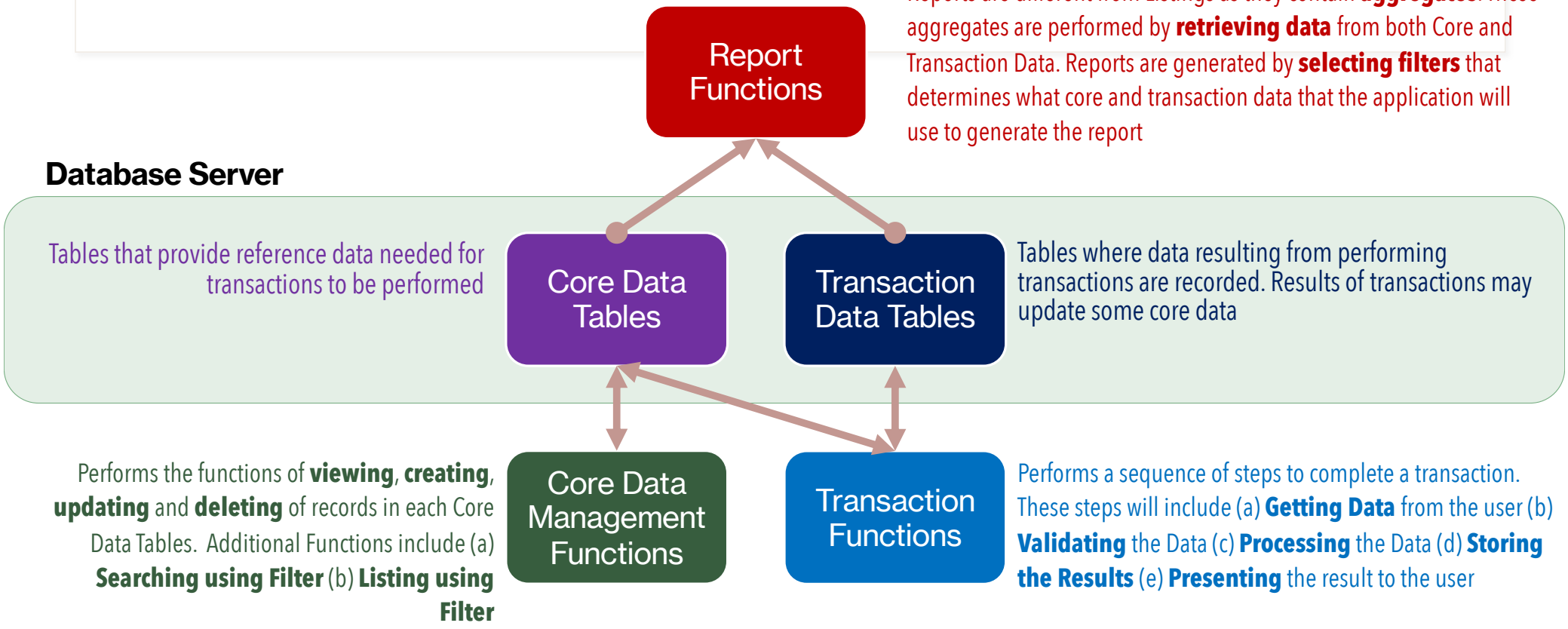


What makes up a Basic Database Application?



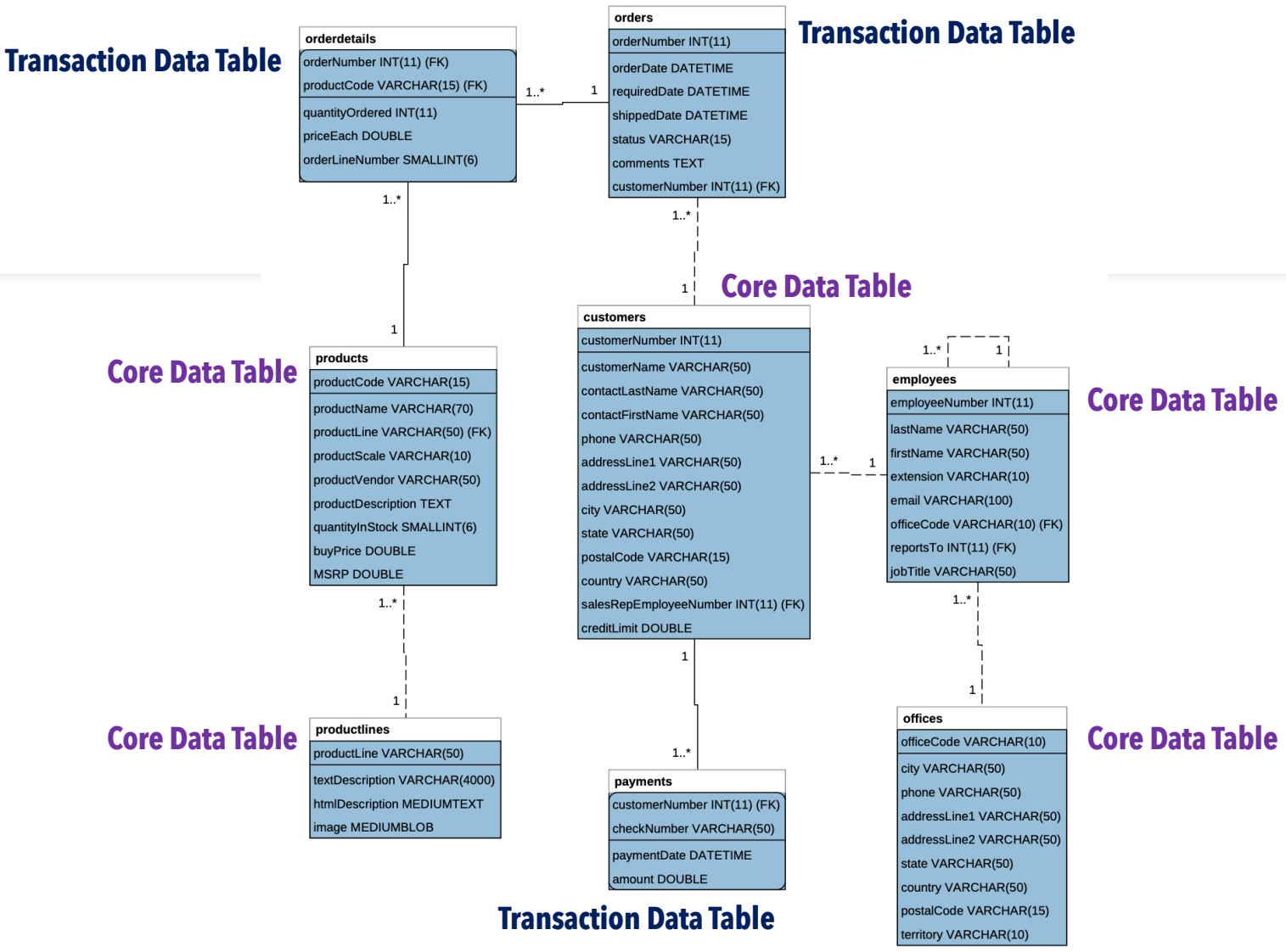
Allow the user to navigate the application and select what it will be performing

What makes up a Basic Database Application?



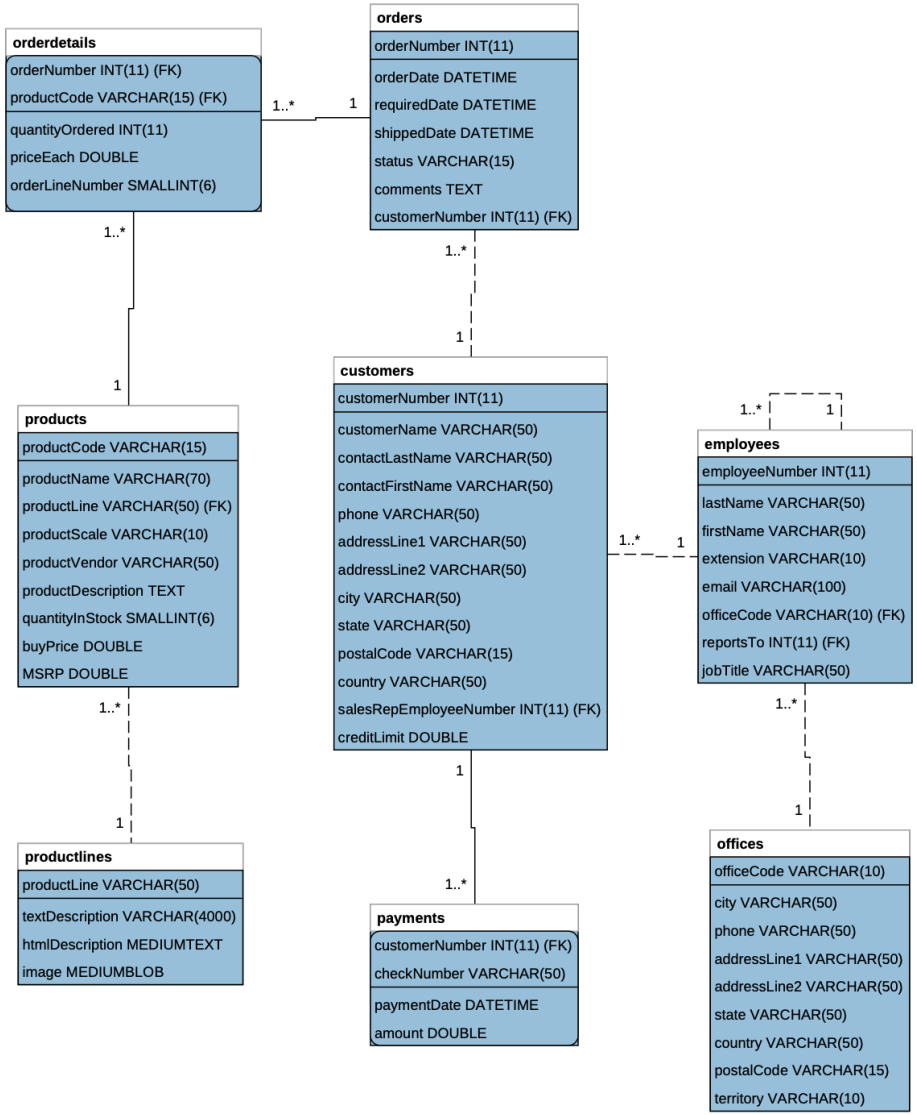
Reality of Application Functions

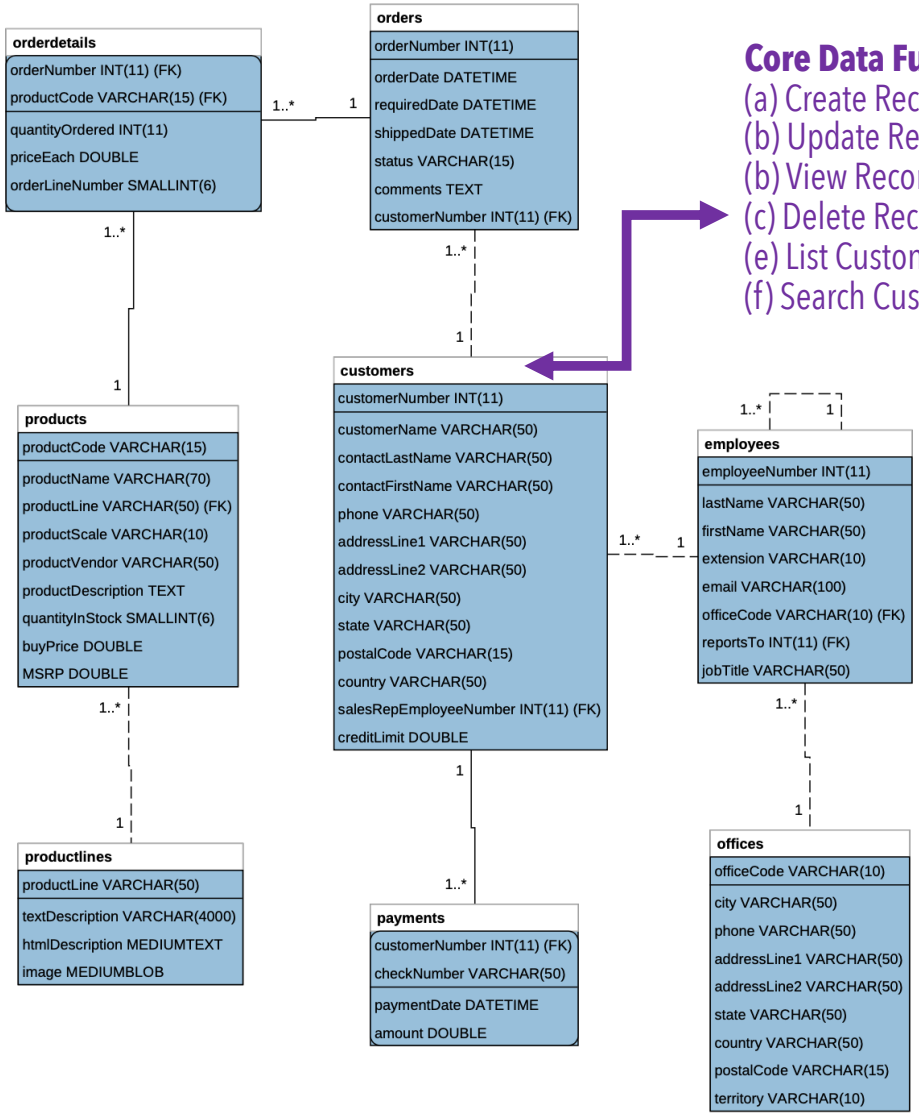
- The steps involve to perform the function may need multiple pages with forms
- At any step, the user should have the option to get out of the transaction (cancel the transaction) and go back to main menu
- The CONSISTENCY and DURABILITY principles of any function (Core Data or Transaction) states that nothing is SAVED in the database unless all the data collection and processing is completed, and it is SAVED, it is saved permanently
  - There are two other principles but these are not covered in CCINFOM. These are Atomity and Isolation

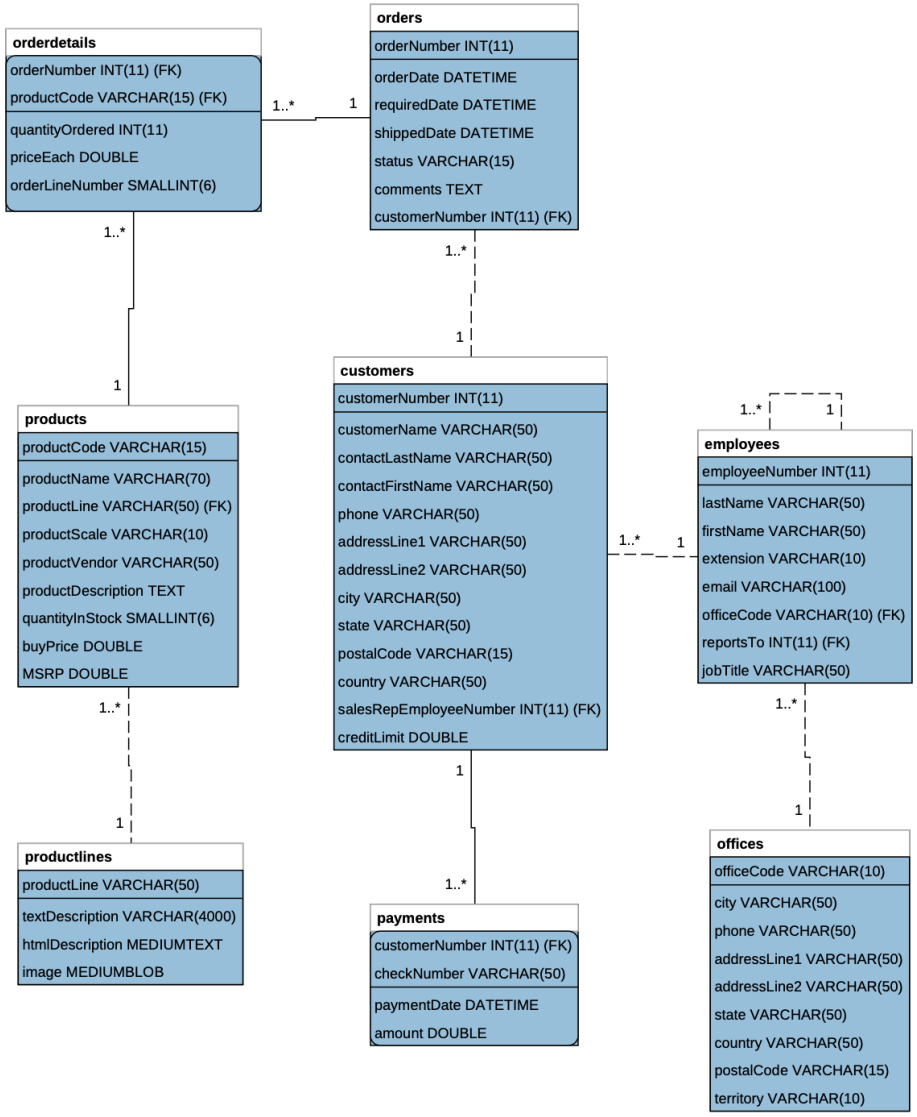

Transaction Data Table
orderdetails
orderNumber INT(11) (FK)
productCode VARCHAR(15) (FK)
quantityOrdered INT(11)
priceEach DOUBLE
orderLineNumber SMALLINT(6)
1..*
1
orders
orderNumber INT(11)
orderDate DATETIME
requiredDate DATETIME
shippedDate DATETIME
status VARCHAR(15)
comments TEXT
customerNumber INT(11) (FK)
Transaction Data Table
1..*
1
Core Data Table
customers
customerNumber INT(11)
customerName VARCHAR(50)
contactLastName VARCHAR(50)
contactFirstName VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(50)
state VARCHAR(50)
postalCode VARCHAR(15)
country VARCHAR(50)
salesRepEmployeeNumber INT(11) (FK)
creditLimit DOUBLE
1..*
1
Core Data Table
products
productCode VARCHAR(15)
productName VARCHAR(70)
productLine VARCHAR(50) (FK)
productScale VARCHAR(10)
productVendor VARCHAR(50)
productDescription TEXT
quantityInStock SMALLINT(6)
buyPrice DOUBLE
MSRP DOUBLE
1..*
1
Core Data Table
productlines
productLine VARCHAR(50)
textDescription VARCHAR(4000)
htmlDescription MEDIUMTEXT
image MEDIUMBLOB
1..*
1
Core Data Table
employees
employeeNumber INT(11)
lastName VARCHAR(50)
firstName VARCHAR(50)
extension VARCHAR(10)
email VARCHAR(100)
officeCode VARCHAR(10) (FK)
reportsTo INT(11) (FK)
jobTitle VARCHAR(50)
1..*
1
Core Data Table
offices
officeCode VARCHAR(10)
city VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
state VARCHAR(50)
country VARCHAR(50)
postalCode VARCHAR(15)
territory VARCHAR(10)
1..*
1
payments
customerNumber INT(11) (FK)
checkNumber VARCHAR(50)
paymentDate DATETIME
amount DOUBLE
Transaction Data Table

Core Data Functions

Create Record of a Products (a)
Update Record of a Product (b)
View Record of a Product (c)
Delete Record of a Product (d)
List Products based on Filter (e)
Search Products based on Filter (f)




orderdetails
orderNumber INT(11) (FK)
productCode VARCHAR(15) (FK)
quantityOrdered INT(11)
priceEach DOUBLE
orderLineNumber SMALLINT(6)
orders
orderNumber INT(11)
orderDate DATETIME
requiredDate DATETIME
shippedDate DATETIME
status VARCHAR(15)
comments TEXT
customerNumber INT(11) (FK)
products
productCode VARCHAR(15)
productName VARCHAR(70)
productLine VARCHAR(50) (FK)
productScale VARCHAR(10)
productVendor VARCHAR(50)
productDescription TEXT
quantityInStock SMALLINT(6)
buyPrice DOUBLE
MSRP DOUBLE
customers
customerNumber INT(11)
customerName VARCHAR(50)
contactLastName VARCHAR(50)
contactFirstName VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(50)
state VARCHAR(50)
postalCode VARCHAR(15)
country VARCHAR(50)
salesRepEmployeeNumber INT(11) (FK)
creditLimit DOUBLE
employees
employeeNumber INT(11)
lastName VARCHAR(50)
firstName VARCHAR(50)
extension VARCHAR(10)
email VARCHAR(100)
officeCode VARCHAR(10) (FK)
reportsTo INT(11) (FK)
jobTitle VARCHAR(50)
productlines
productLine VARCHAR(50)
textDescription VARCHAR(4000)
htmlDescription MEDIUMTEXT
image MEDIUMBLOB
payments
customerNumber INT(11) (FK)
checkNumber VARCHAR(50)
paymentDate DATETIME
amount DOUBLE
offices
officeCode VARCHAR(10)
city VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
state VARCHAR(50)
country VARCHAR(50)
postalCode VARCHAR(15)
territory VARCHAR(10)
1..*
1
Core Data Functions
(a) Create Record of a Customer
(b) Update Record of a Customer
(b) View Record of a Customer
(c) Delete Record of a Customer
(e) List Customers based on Filter
(f) Search Customers based on Filter



Core Data Function on Create of Product

(a) Display in the form, elements where user can enter data about Products (Product Code, Product Name, Product Scale, Product Vendor, Product Description, Quantity In Stock, Buy Price and MSRP.

(b) Retrieve from the database the list of product lines and put then in choices (dropdown), that the user can choose from for the product line of the product being recorded.

(c) In the same form if user clicks on Cancel, go back to main menu

(d) In the same form if user clicks on Proceed, then go to next page.

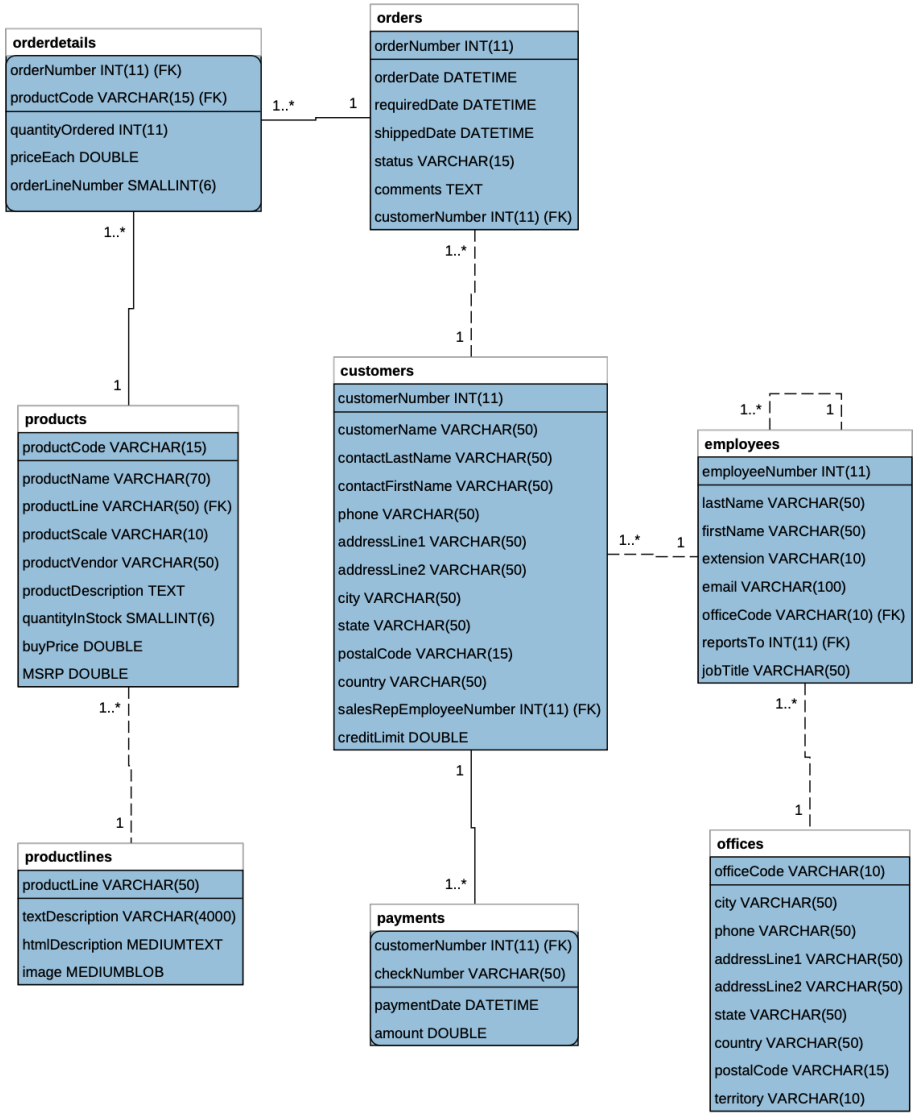
(e) Check if all the fields are provided with values. If one of them does not have a value, display on screen that the values for the fields are required and provide a button to go back to (a). Otherwise, proceed to (f)

(f) Check if all the fields are of correct data type, if one of them does not conform to the correct data type, display on screen that the value must be in correct data type and provide a button to go back to (a). Otherwise, proceed to (g)

(g) Save the new record to the database

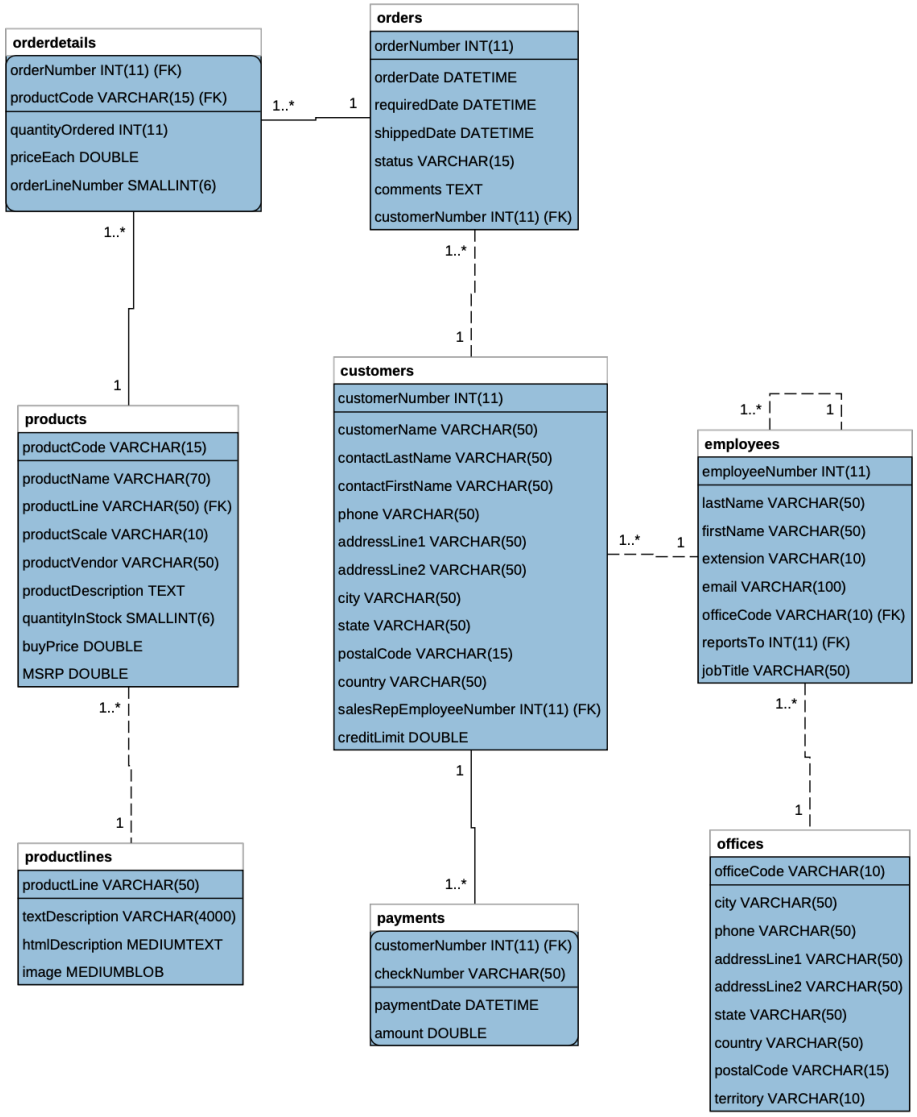
(h) Display on screen the newly saved record

(i) Provide a button to go back to the main menu and a button to go back to (a) to create another product record



Core Data Function on Update of Customer

(a) Display in the form, elements where user can enter his/her Sales Person ID and Customer Number
(b) In the same form if user clicks on Cancel, go back to main menu
(c) In the same form if user clicks on Proceed, then go to next page.
(d) Check if Sales Person ID and Customer Number was provided with values. If one of them does not have a value, display on screen that both values are required and provide a button to go back to (a). Otherwise, proceed to (e)
(e) Check if all the fields are of correct data type, if one of them does not conform to the correct data type, display on screen that the value must be in correct data type and provide a button to go back to (a). Otherwise, proceed to (f)
(f) Retrieve Customer Record
(g) Check if customer belongs to the Sales Person. If not belonging to the Sales Person, display on screen that the customer is not under the user's list of customer and provide a button to go back to (a), Otherwise, proceed to (h)
(h) Display in the form, elements where the old values from the record can be shown and updated, and buttons to cancel and save the update. Clicking on the cancel button will go back to main menu.
(i) User will update the old values as he decides
(j) User clicks on Save to go to the next page
(k) Check if all the fields were provided with values. If one of them does not have a value, display on screen that the values are required and provide a button to go back to (d). Otherwise, proceed to (l)
(l) Check if all the fields are of correct data type, if one of them does not conform to the correct data type, display on screen that the value must be in correct data type and provide a button to go back to (d). Otherwise, proceed to (m)
(m) Save the updated data to the database
(n) Display the old and new values of the record
(o) Provide a button to go back to the main menu and a button to go back to (a) to update another customer record

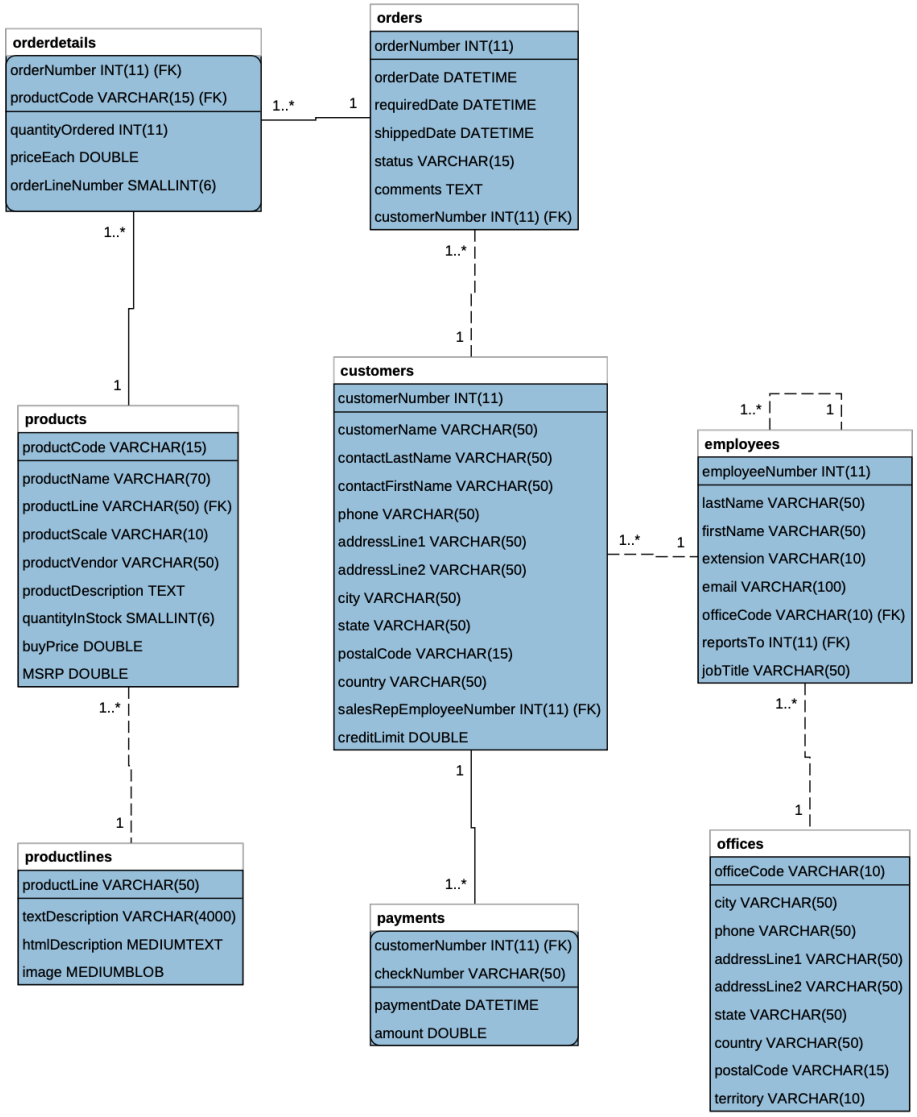


Core Data Function on Search of Employees

(a) Display form, elements representing fields used as criteria to search employees. For this search, the filter criteria are last name, first name, office code and job title.
(b) For job title, the list of job titles are retrieved from the database and placed on choices (dropdown). For the office code, the list of office codes are retrieved from the database and placed on choices (dropdown).
(c) The user puts or selects values for his/her criteria choices. Not all filter criteria may be given a value or selected a value on, this means that the criteria is not used.
(d) In the same form if user clicks on Cancel, go back to main menu
(e) In the same form if user clicks on Search, then go to next page.
(f) Using the filter criteria used in the previous page, retrieve the list of employees (employee number, last name and first name, email and job title) sorted accordingly and display them on screen as a list with proper headers
(g) In the same form, if user clicks on Main menu, go back to main menu
(h) In the same form, if user clicks on Search Again, go back to (a)

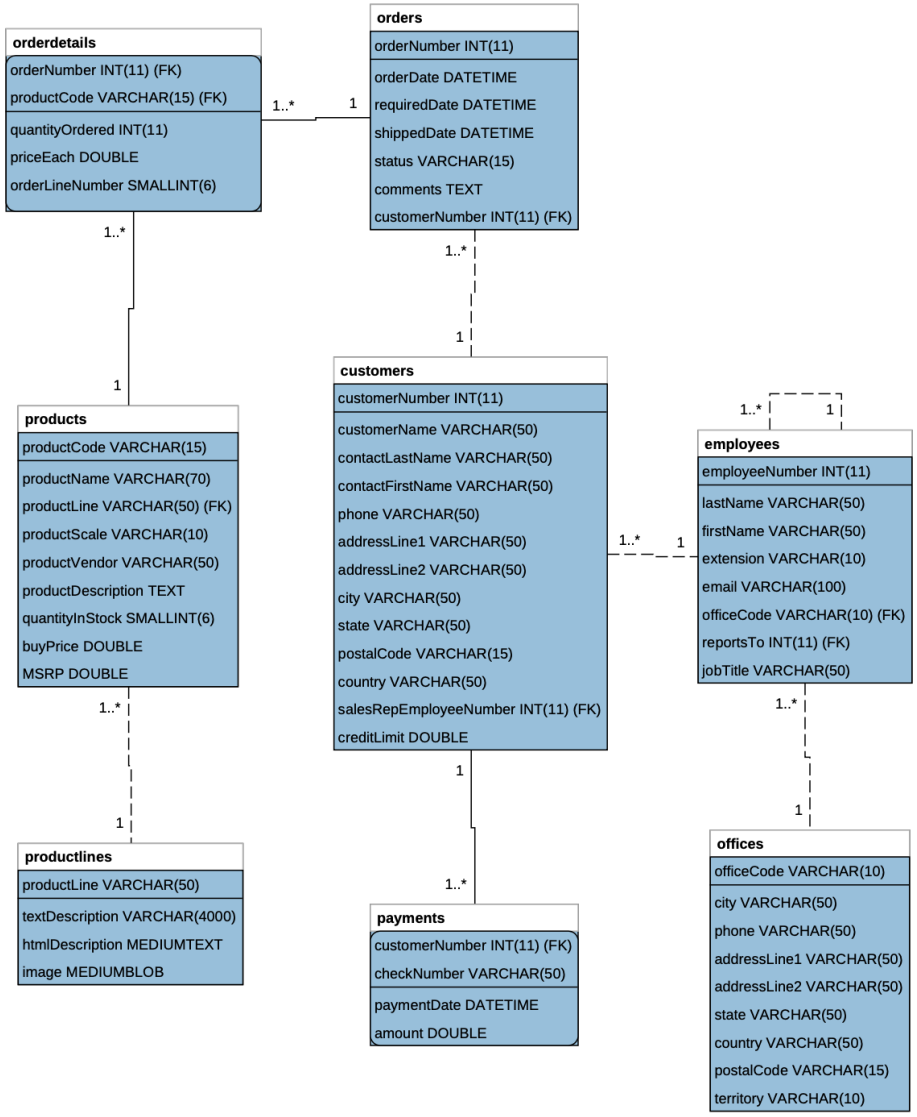
Core Data Function on View of Employees

(a) Display in the form, a field where user can enter the employee number to view the record
(b) In the same form if user clicks on Cancel, go back to main menu
(c) In the same form if user clicks on View, then go to next page.
(d) The record of the employee is retrieved from the database
(e) If the record is not found, display on screen that the employee cannot be found, and provide a button to go back to (a). Otherwise, proceed to (f)
(f) Display the retrieved record of the employee
(g) Provide a button to go back to the main menu and a button to go back to (a)



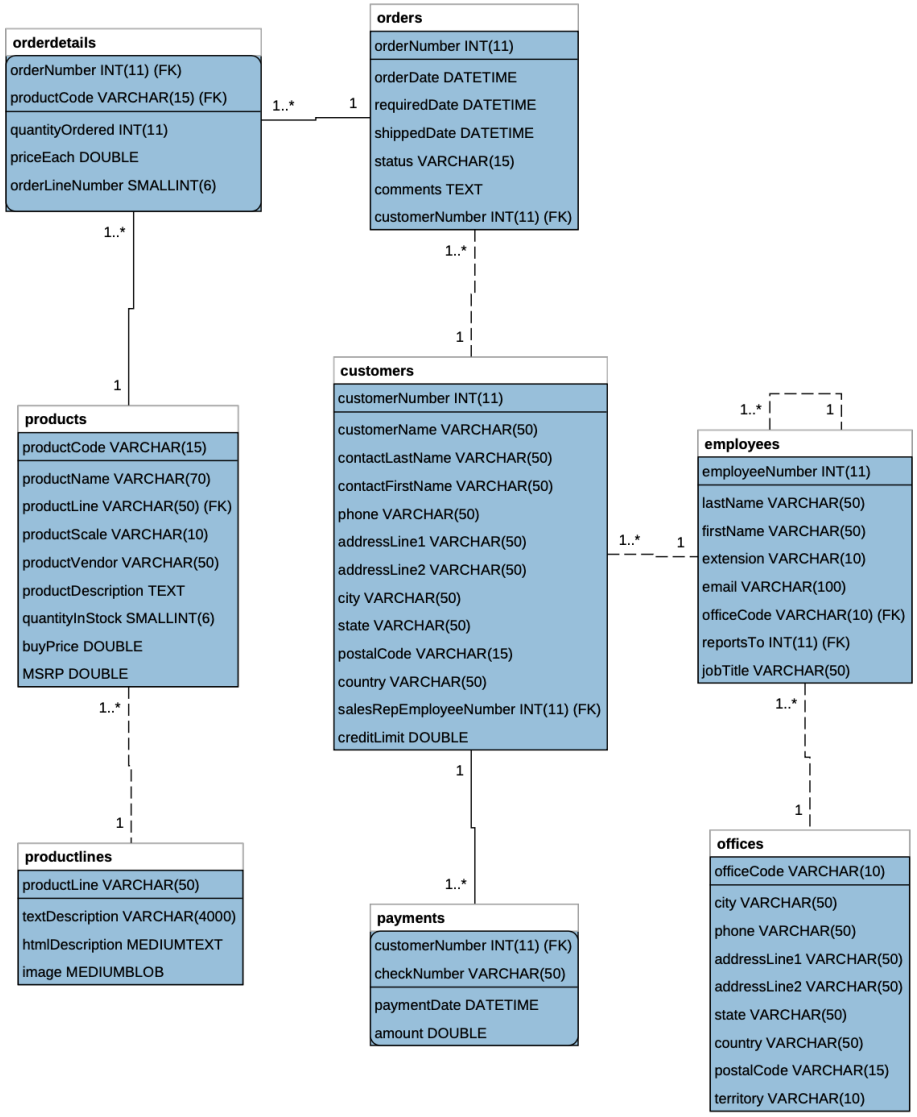
Core Data Function on Delete of Employees

(a) Display in the form, a field where user can enter the employee number to delete the record
(b) In the same form if user clicks on Cancel, go back to main menu
(c) In the same form if user clicks on View, then go to next page.
(d) The record of the employee is retrieved from the database
(e) If the record is not found, display on screen that the employee cannot be found and provide a button to go back to (a). Otherwise proceed to (f)
(f) Display the retrieved record of the employee
(g) Check the database if the employee is being used in Customers.
(h) If being used, display on screen that the employee cannot be deleted and provide a button to go back to (a). Otherwise, proceed to (i).
(i) Provide a button to go back to the main menu and a button to go the next page to delete the record.
(j) Delete the record from the database
(k) Display "Record is Deleted"
(l) Provide a button to go back to the main menu and a button to go back to (a) to delete another customer record



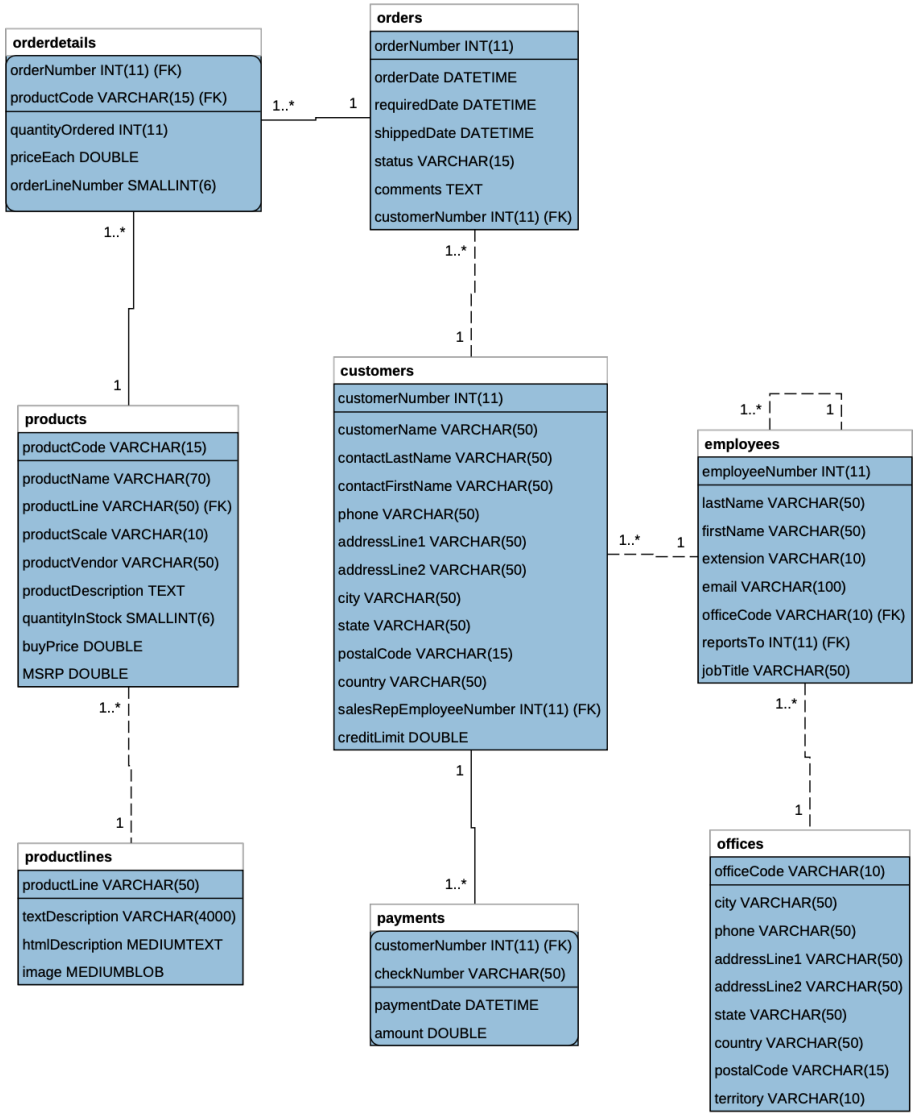
Transaction on Sales

(a) Display in the form, field to enter the Sales Person ID, Customer Number and Required Date
(b) In the same form if user clicks on Cancel, go back to main menu
(c) In the same form if user clicks on Order, then go to next page.
(d) Check if Sales Person ID, Customer Number and Required Date was provided with values. If one of them does not have a value, display on screen that both values are required and provide a button to go back to (a). Otherwise, proceed to (e)
(e) Check if all the fields are of correct data type, if one of them does not conform to the correct data type, display on screen that the value must be in correct data type and provide a button to go back to (a). Otherwise, proceed to (f)
(f) Retrieve Customer Record
(g) Check if customer belongs to the Sales Person. If not belonging to the Sales Person, display on screen that the customer is not under the user's list of customer and provide a button to go back to (a). Otherwise, proceed to (h).
(h) Retrieve all ordered amount and payments made. Compute if there are still amounts needed to be paid.
(i) Check if Customer is eligible to order based on if credit limit has not been exceeded. If customer is not eligible, display on screen that customer is not eligible to order and provide a button to go back to (a). Otherwise, proceed to (i).
(j) Save the order record in memory
(k) In the same form if user clicks on Cancel, go back to main menu
(l) In the same form if user clicks on Order Products, then go to next page.
(m) Display in the form, fields to enter what product is being ordered, quantity to order and price to use for the product
(n) Retrieve **available** products that can be ordered and put in the choices (dropdown)
(o) User puts in data for the ordered products
(p) In the same form if user clicks on Cancel, go back to main menu
(q) In the same form if user clicks on Add to Cart, then go to next page.



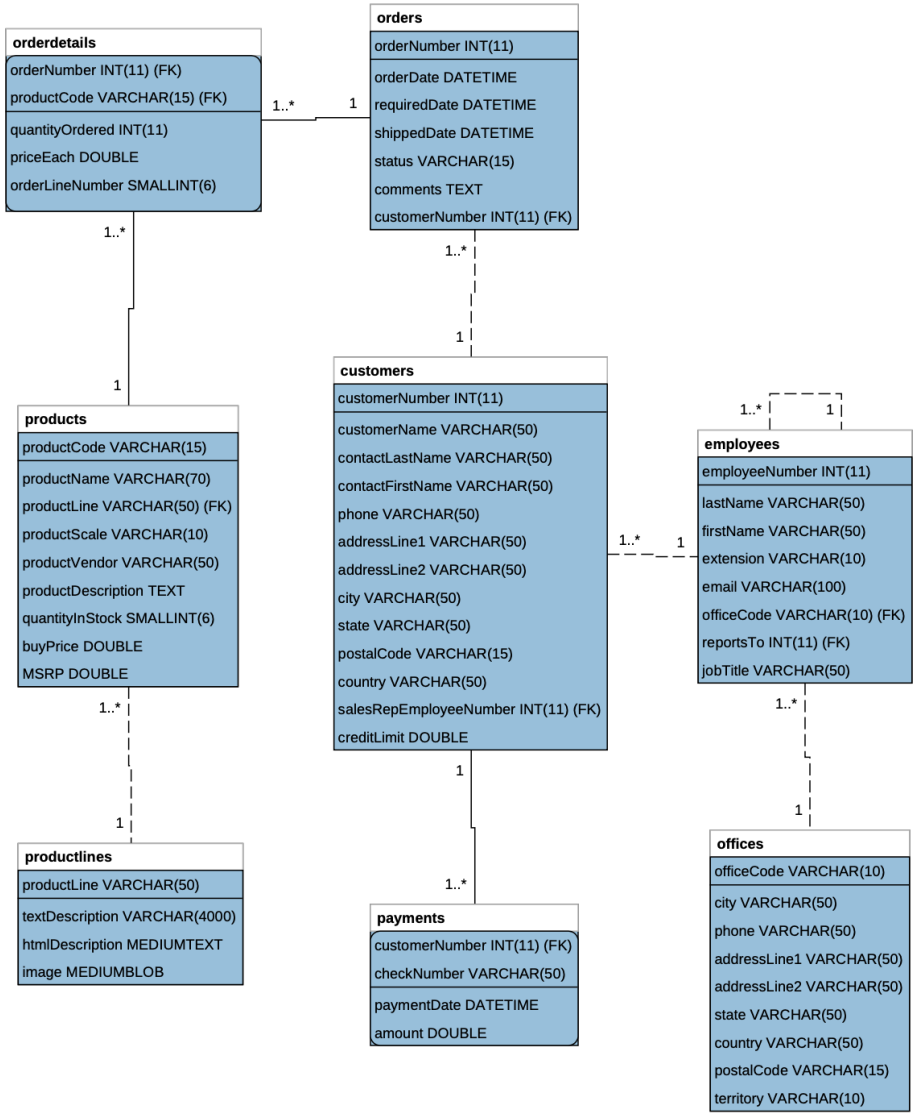
Transaction on Sales

(r) Retrieve data about the Product
(s) Check if quantity ordered is within the quantity of the product, if not then display on screen that there are not enough products and provide a button to go back to (m). Otherwise, proceed to (t).
(t) Check if price is below buyPrice, if it is then display on screen that the price cannot be below the buy Price amount and provide a button to go back to (m). Otherwise, proceed to (u).
(u) Save the order details in memory
(v) In the same form if user clicks on Add another Order, go back to (m)
(w) In the same form if user clicks on Save Order, then go to next page.
(x) Generate a new Order Number by checking the last order number and adding 1
(y) Save the order record in the database with Order Date is the date today, Shipped Date is NULL, Status is In Process, Comments is NULL
(z) Save every product ordered in OrderDetails. Order Line Number starts with 1 and increments for every product ordered.
(aa) Update Products by deducting the quantity ordered from the quantity in stock
(bb) Display the details of the order and order details
(cc) Provide a button to go back to the main menu and a button to go back to (a) to perform another Sales Transaction



Transaction on Payment

(a) Display form, field to enter the customer Number
(b) In the same form if user clicks on Cancel, go back to main menu
(c) In the same form if user clicks on Retrieve, then go to next page.
(d) Retrieve customer record
(e) Retrieve total amount of orders
(f) Retrieve total amount of payments made
(g) Compute for amount still needed to pay
(h) Display amount to be paid on screen
(i) Display in the form, fields for the user to enter the amount to be paid and the check number
(j) In the same form if user clicks on Cancel, go back to main menu
(k) In the same form if user clicks on Pay, then go to next page.
(l) If amount of payment exceeds the amount needed to be paid, display on screen that amount to be paid is less than payment amount, and provide a button to go back to (a). Otherwise, proceed to (m)
(m) Check if check number already exists. If check number already exists, display on screen that check number is already used and and provide a button to go back to (a). Otherwise, proceed to (n)
(n) Save the payment record
(o) Display the result the payment details
(p) Provide a button to go back to the main menu and a button to go back to (a) to make another payment transaction



Report Generation - Sales Report

(a) Display form, elements representing fields used as criteria to generate the report. For this report, the filter criteria are start month and end month, start year and end year, and customer number

(b) The user puts or selects values for his/her criteria choices.

(c) In the same form if user clicks on Cancel, go back to main menu

(d) In the same form if user clicks on Generate, then go to next page.

(e) Using the filter criteria used in the previous page, retrieve the report using SQL, sorted accordingly and display them on screen as a list with proper headers - MONTH, YEAR, CUSTOMERNUMBER, TOTAL SALES

(f) In the same form, if user clicks on Main menu, go back to main menu

(g) In the same form, if user clicks on Generate Report Again, go back to (a)

Important to take note of for the DB Applications Project

- A Database Design is needed
- A Database implemented is needed
- The database must have adequate sample values to demonstrate the feature. For example:
  - Report Generation cannot be adequately demonstrated if there are not enough data already inside the DB.
  - Search cannot be adequately demonstrated if there are not enough data already inside the DB.
  - Update and Delete cannot be adequately demonstrated if there are not enough data to perform the functions
  - If the functions cannot be adequately demonstrated because there is lack of data in the database, the function is deemed **NOT IMPLEMENTED.** How did you adequately test the application, if in the first place, there are not enough data in your database.
- Take note that in the requirements for submission, the DATABASE with adequate data is needed.