Chris Hall & John McCormick

Instructor: Dr. Danielle Safonte, M.I.S. Ed.D., & Dr. Michael Curry

CS340 Introduction to Databases

18 May, 2023

Project Step 3 Final Version: Design HTML Interface + DML SQL

Team Members' Names and Roles

Chris Hall and John McCormick
Team 86, Schema-Teama

Project Title

Portland Cycling Specialties

Link to HTML

http://flip1.engr.oregonstate.edu:7821/

Feedback from Peer Review

From Vincent Ly

- *Does the UI utilize a SELECT for every table in the schema?* In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
    - Yes. There are 7 entities and 7 UI pages.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
    - No. Seeing that it's all text input, I did not notice anything that would dynamically populate a list.
- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.
    - Yes. There are forms in each UI page for inserting.
- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT

row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
    - The insert does add the FK attributes. Similar to the question, the database has Order, Product, and OrderProducts table. From the UI, I don't expect OrderProduct to automatically insert when inserting a new order.
- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - Yes. In the OrderProducts table.
- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - Yes there are update for each of the entity
- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
    - No. From the DDL, all the foreign key are NO NULL.
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*
    - Overall, looks good. I would've built it the same way. Add some drop down menu and build in some dynamic lookup for creating or updating records.

From Joana Shin

- Does the UI utilize a SELECT for every table in the schema?
    - Yes each table from the schema utilizes a select query!
- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
    - I don't see a drop down feature on the UI nor on the dml sql file.
- Does the UI implement an INSERT for every table in the schema?
    - Yes, the UI implements an INSERT for every table in the schema!
- Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?
    - Yes, the INSERT adds to the corresponding FK attributes as shown between Orders entity and OrderProducts entity which was manually done by adding an INSERT feature in the intersection table. My partner and I executed like this as well.
- Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?
    - There are several DELETE features and there is one DELETE that removes things from the M:M relationship as represented in the OrderProducts entity.
- Is there at least one UPDATE for any one entity?
    - Yes, there are UPDATE features on all entities!
- Is at least one relationship NULLable?

    - I see a couple of ON DELETE CASCADE but not a nullable relationship on the ddl sql file.
- Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.
    - Adding a navigation menu at the top or side of each page might help with browsing through the website more seamlessly instead of clicking the back button.
    - Very minor and more of a personal preference but maybe label insert form (i.e. adding new customers, adding new products)

Great job you two! Can't wait to see the outcome!

From Andrew Park

- Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI.
    - Yes, there are 7 pages and each utilize a SELECT query for every table.
- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
    - No, I did not see a dropdown option.
- Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.
    - Yep. I seen an INSERT for every table in the schema.
- Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
    - Yes, the INSERT function makes the correct update between Order and OrderProducts.
- Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - Yes, I see the that there is a DELETE option for OrderProducts.
- Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - Yes, there is an UPDATE function for each entity in each table.
- Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
- Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

    - One suggestion I have is including a menu option at the top on each HTML page that links to all the other HTML pages. That way, you are able to navigate to other pages or the home page directly instead of having to hit go backwards.
    - Another suggestion is to add a NULLable relationship. I'm not sure which would be fitting of having a nullable value since most of your entities and attributes seem important. I think maybe the product description might be nullable if the product's name is self explanatory and doesn't need a description. For example, a steel bolt might not need any more descriptions.

Overall great job on your HTML pages!

From Paola Cernada

- Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.
    - Yes, the UI utilizes a SELECT for every table in the schema which are all included in the DML.sql file.
- Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
    - There is no SELECT included in the DML.sql file or in the UI which utilizes a search/filter with a dynamically populated list of properties. I think a good place to include this could be in the Products table, the Supplier ID could be a dropdown, and this can display just the Supplier ID or it can be linked by ID to the Suppliers table and display the Supplier Name instead of Supplier ID under the Products table.
- Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.
    - Yes, the UI implements an INSERT for every table in the schema. The UI includes input fields that correspond to each table and the attributes in each table.
- Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words, if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, roductid, qty, price and line_total).
    - Yes, the INSERT operation updates the related foreign key attributes between the Orders entity and OrderProducts entity.
- Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
    - Yes, there is a DELETE statement for each table, and specifically, the DELETE statement for the OrderProducts table handles the deletion in a many-to-many relationship.

- Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g., be updated for a single ProductID record?
    - In the DML.sql file there are UPDATE queries for each table. The UPDATE feature is also included in every page of the UI.
- Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g., having an Employee might be optional for any Order. Thus, it should be feasible to edit an Order and change the value of Employee to be empty.
    - After reviewing the files and UI I was unable to find any NULLable relationships. I do see the importance of every one of the attributes in these tables, I would suggest making the 'email' in the Customers table a NULLable value as not everyone might have an email or want to share their email, therefore, I think this field is not absolutely mandatory.
- Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.
    - I really like the idea, design, and implementation. I think you guys have done a great job thus far. I would suggest reviewing some of the attributes in your tables and determining what is truly essential information. Below a couple additional suggestions:
        - Email and Address in the Customers table don't necessarily have to be mandatory; this is personal information that is nice to have, but not everyone feels safe sharing.
        - representativeName in the Suppliers table can also be an optional field or renamed for clarity as primaryPointOfContact, representativeName makes me think of the person you spoke to while placing an order.
        - In the Inventory table I see inStockQuantity, but where are you guys keeping track of what has been sold? Maybe that is not part of the DB design, but something I wanted to point out.
        - In the Locations table is name absolutely necessary, as you already have the location ID and address, another possibility for NULLable value.

Keep up the great work!

Actions Accepted Based Upon the Feedback

- Need to add dropdown menus for SELECT.
- Make product description a NULLable value.
- Add navigation to all pages.

Actions Rejected Based Upon the Feedback

- Any change to having mandatory entries such as name, email, or address become optional. This is because, in a real-world scenario, you would want to know your

customer's name, email, and address to send promotions to, among other reasons. Small business B2B is often built off of personal relationships that have recurring interactions.

Upgrades to the Draft Version

- CSS has been implemented
- Added top navigation to all pages
- Added dropdown menus to SELECT where needed (Products and Customers).
- Prepped for the dynamic population in step 4.

Project Outline and Database Outline:

Overview

Portland Cycling Specialties was a small bicycle specialties shop. They were a small mom and pop business which has grown in the past decade into multiple locations. Sales are up to more than $1500 in 200 sales on average each day in most of their stores and the old manual system of recordkeeping no longer works well for them. They need a good database system in which to track sales at each location and for orders which come in on their website.

The database will need to track customers and their orders, as well as inventory at store location, and products and suppliers.

Database Outline in Words

1. **Customers:** Records of each customer who makes a purchase.
    - customerID: int, auto_increment, unique, not NULL, PK
    - firstName: varchar(45), not NULL
    - lastName: varchar(45), not NULL
    - email: varchar(45), not NULL
    - address: varchar(45), not NULL
    - phoneNum: varchar(45), not NULL
        i. One-to-many (1:M) relationship with Orders using customerID as a foreign key in Orders.

2. **Products:** Items we sell at Portland Cycling Specialities.
    - productID: int, auto_increment, unique, not NULL, PK
    - name: varchar(45), not NULL
    - description: varchar(45), not NULL
    - price: decimal(10,2), not NULL
    - supplierID: int, not NULL, FK (ref Suppliers)
        i. One-to-many (1:M) relationship with Inventory using productID as a foreign key in Inventory.
        ii. Many-to-many (M:N) relationship with Orders implemented in the OrderProducts table.

3. **Suppliers:** Companies which supply products to us.
    - supplierID: int, auto_increment, unique, not NULL, PK
    - name: varchar(45), not NULL
    - email: varchar(45), not NULL
    - phone: varchar(45), not NULL
    - representativeName: varchar(45), not NULL

    i. One-to-many (1:M) relationship with Products using supplierID as a foreign key in Products.

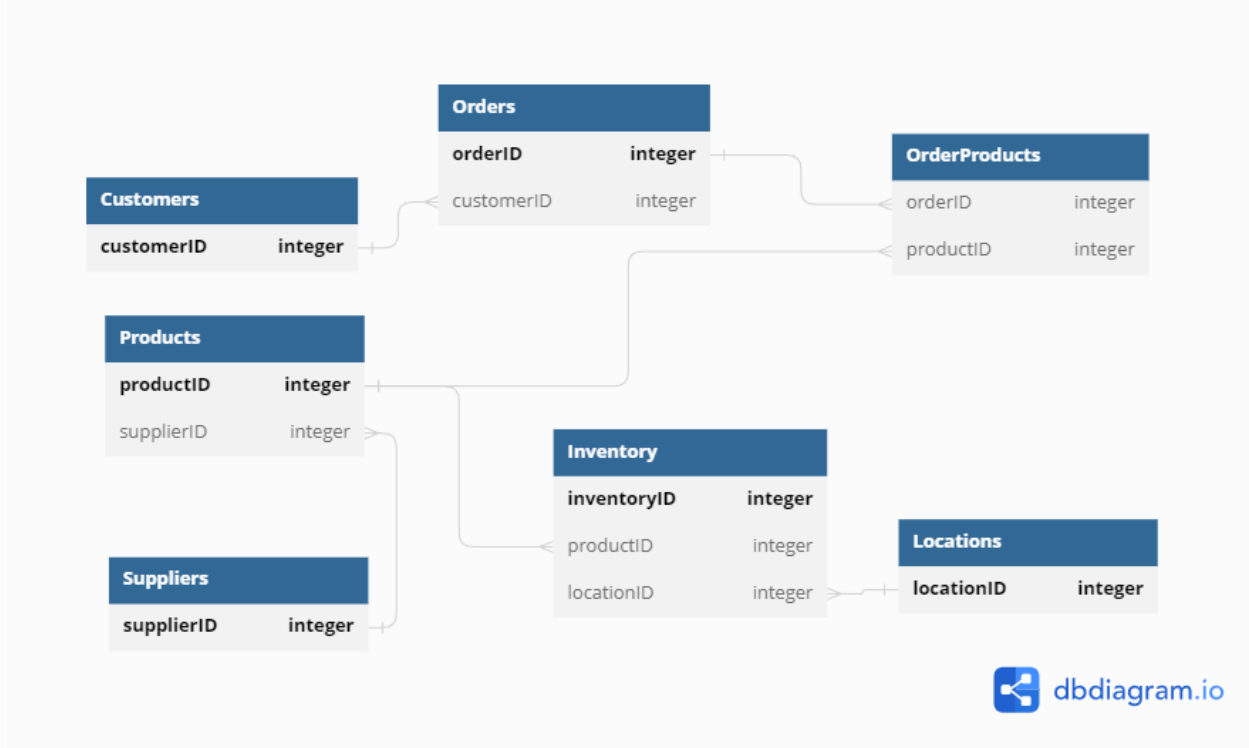
4. **Inventory:** Quantity of each product at each of our locations.
    - inventoryID: int, auto_increment, unique, not NULL, PK
    - productID: int, not NULL, FK (ref Products)
    - locationID: int, not NULL, FK (ref Locations)
    - inStockQuantity: int, not NULL
        i. One-to-many (1:M) relationship with Products using productID as a foreign key in Inventory.
        ii. One-to-many (1:M) relationship with Locations using locationID as a
        iii. foreign key in Inventory.

5. **Orders:** Records of sales to each customer.
    - orderID: int, unique, not NULL, PK
    - customerID: int, not NULL, FK (ref Customers)
    - orderDate: date, not NULL
    - total: decimal(10,2), not NULL
        i. One-to-many (1:M) relationship with OrderProducts using orderID as a foreign key in OrderProducts.
        ii. One-to-many (1:M) relationship with Customers using customerID as a foreign key in Orders.

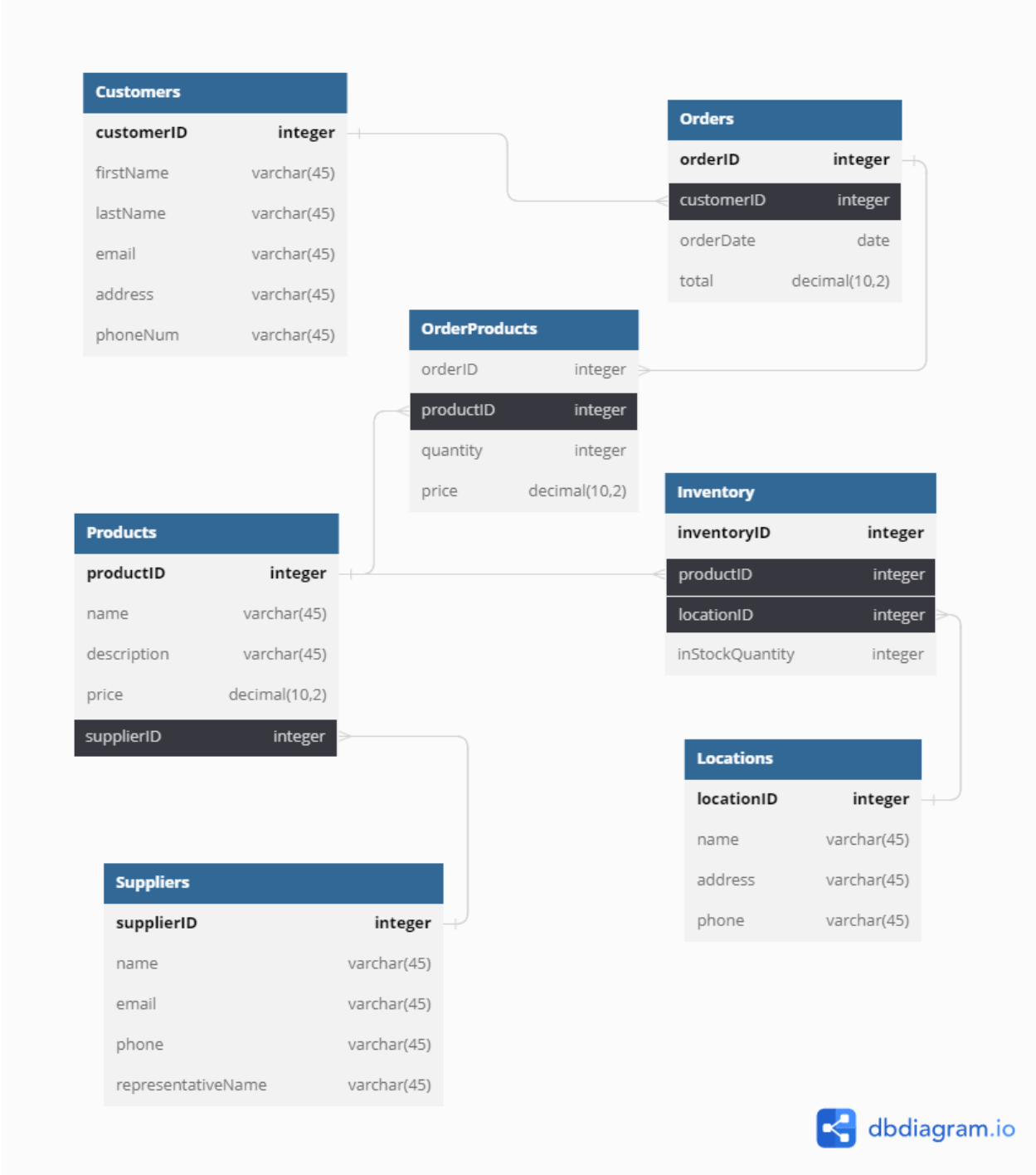
6. **OrderProducts:** Links Orders and Products in a M:N relationship.
    - orderID: int, not NULL, FK (ref Orders)
    - productID: int, not NULL, FK (ref Products)
    - quantity: int, not NULL
    - price: decimal(10,2), not NULL
        i. Many-to-many (M:N) relationship with Products implemented using productID and orderID as foreign keys.

7. **Locations:** Our company stores locations.
    - locationID: int, auto_increment, unique, not NULL, PK
    - name: varchar(45), not NULL
    - address: varchar(45), not NULL
    - phone: varchar(45), not NULL
        i. One-to-many (1:M) relationship with Inventory using locationID as a foreign key in Inventory.

Database Diagram



Schema


Customers
customerID integer
firstName varchar(45)
lastName varchar(45)
email varchar(45)
address varchar(45)
phoneNum varchar(45)
Orders
orderID integer
customerID integer
orderDate date
total decimal(10,2)
OrderProducts
orderID integer
productID integer
quantity integer
price decimal(10,2)
Products
productID integer
name varchar(45)
description varchar(45)
price decimal(10,2)
supplierID integer
Inventory
inventoryID integer
productID integer
locationID integer
inStockQuantity integer
Locations
locationID integer
name varchar(45)
address varchar(45)
phone varchar(45)
Suppliers
supplierID integer
name varchar(45)
email varchar(45)
phone varchar(45)
representativeName varchar(45)
dbdiagram.io

Example Data

Customers:

customerID	firstName	lastName	email	address	phoneNum
1	John	Doe	j.doe@email.com	123 Main St, Portland	5035555555
2	Jane	Doe	jane.d@email.com	456 Elm St, Portland	5035555556
3	Bob	Dole	b.dole@email.com	789 Oak St, Portland	5035555557

Products:

productID	name	description	price	supplierID
1	Mountain Bike	Sturdy all-terrain bike	350.00	1
2	Road Bike	Lightweight and fast road bike	550.00	2
3	Helmet	Protective headgear	50.00	3

Suppliers:

supplierID	name	email	phone	representativeName
1	AllTerrain Co.	allterrain@email.com	5035550001	John
2	SpeedBikes Inc.	speedbikes@email.com	5035550002	Sarah
3	SafeCyclist Ltd.	safecyclist@email.com	5035550003	Robert

Inventory:

inventoryID	productID	locationID	inStockQuantity
1	1	1	10
2	2	1	8
3	3	1	20

Locations:

locationID	name	address	phone
1	Portland Downtown	124 Main St, Portland	5035551122
2	Portland East	456 82nd St, Portland	5035552233
3	Gresham	789 Main St, Gresham	5035553344

Orders:

orderID	customerID	orderDate	total
1	1	2023-05-01	700.00
2	2	2023-05-02	600.00
3	3	2023-05-03	400.00

OrderProducts:

orderID	productID	quantity	price
1	1	1	350.00
1	3	1	50.00
2	2	1	550.00
3	3	2	100.00