Project group 142: Sporting Goods Shop

Eric Kwak, Researcher and Ed discussion liaison

Andrew Adisoemarta, Submission proofreader and deadline enforcer

Actionable Reviews:

Step 3 Reviews

From Joseph Shing:

- The intersection tables were embedded in the entity table pages (Ordering Products in the Order Info tab and Processed Orders Records in the Processed Orders tab). The names of the Product/Supplier/Vendor doesn't match the naming convention of the schema and took a few seconds to match up.
- There are no search/filter features included (I haven't seen this implemented in other assignments too - not sure if this was listed in the rubric). However, this can be easily implemented in various pages in this UI. For example, a search can be implemented in the Processed Orders Records page when a user wants to search for a previous purchase order in the table. The filter feature can easily be implemented in any of the entity table pages as well, such as product info, where the user might want to filter by price, or in the employees page where the user might want to filter by role.
- Based on the UI, if placing an order through the Order Info > Place Order Here! page, then the FK of Purchase ID and Product ID should be linked to the intersection table in the Processed Orders Records page. I am guessing that this is what the team intends on doing as this is what is outlined in the DDL.sql file in which the PurchaseID and ProductID are foreign keys in the PurchaseOrderProducts intersection table.
- Based on the UI and the schema, it would seem like everything that is present on the web page is required information - which isn't a bad thing. In order to make something NULLable, maybe we can look at incorporating a new relationship between entities that might not be ultimately required. For example, maybe the Supplier Entity can be related to Orders in which it can be included but optional if not.
- Overall, great work! I like the site structure and layout so far. I think that everything makes sense if you read the schema first. Some things that could make the page even better are probably just matching up naming conventions with the schema so that everything is as close to 1:1 as possible, and maybe for the sake of this site, breaking out the intersection tables into a separate page, maybe as a tracking or visualization page, so that the user can easily access it instead of having to find it, which was a little confusing to me at first. Otherwise, great work again!

From Taylor Rupe:

- Yes, each table has its own page but it was definitely confusing to find the OrderProducts page/table. Also, I don't know why all the table names are changed in the UI; they don't add any more information that helps us understand them more than the original names. "Product Info" Why not just stick to 'Products"? This was particularly

noticeable in the intersection tables; I had to go back and forth between the PDF schema and the UI to figure out what table corresponded to the UI I was looking at. Especially the OrderProducts, the button 'Place an order here!" why does that bring to the OrderProducts? I suggest keeping the names per your Schema as they're informative enough. The reason I say this and why I believe you did the UI this way is because you're trying to design a customer facing UI which wasn't the objecive. The Project Guide notes that this UI is not for an end user or customer it's for a DB Manager so you're making this for the Manager of this sporting good store not the customers. So while the name changes would be fine for a customer, it wouldn't be ideal if you're designing this for the DB manager of the store.

- No there isn't any search or filter functionality.
- I don't believe so. Currently on orders there's no indicating inserting an Order will create a record in OrderProducts nor is there a way to connect a ProductId to an order. To fix this you can include an input to allow the DB manager to select a productId in the 'Add Order' form. You don't display it on the Orders table page but by doing that you can allow for the record to be created in the OrderProducts table.
- Yes there's a Delete for every table but without the functionality implemented there's no way to test if deleting an Order will also delete a record from OrderProducts. Also, I'd suggest changing the delete form, a delete just needs the unique identifier like the order id in Orders. You shouldn't need to fill in an entire form of its respective attributes to delete it because if you only wanted to delete one attribute but keep the rest, that's an UPDATE not a delete. It should just be a basic button next to the row/record you're deleting to denote which record will be deleted.
- I believe so, the update form doesn't have any 'required' fields denoted by either CSS styling or a description so as of right now it is implied a field can be null but we won't know until the actual functionality is implemented in later steps.
  Just to reiterate what I suggested, I would change then names all back to the original, not only does this make sense for a DB manager of the store but it will also help both reviewers and the TA/graders in the future when they're going over your project.
  Truthfully, I'm afraid the project assignment was misunderstood and I think that's where a lot of the pitfalls of this UI come from. I could be wrong but that's verbatim in the Project Guide which I would trust more than anything else but yeah I think you guys just need to change the names and make it much easier to navigate to the intersection tables. That's all it would take to fix the UI. You already have it as it's own page which is great but don't make a small little text on click to navigate to it, just have it be a big tab like the tables 'Customers' or "Employees". Or at the very least include both, if you want to allow the OrderProducts to be navigated from the Orders page have the button although definitely make it easier to understand what happens when you click that button, and then also include big navigation tab to get to that table's page. Also I'd suggest changing the text color you're using with the blue background, it doesn't provide enough contrast. I tested a bunch of different colors like red, black, even just 'color: blue', is easier to read than the current text color.

From Kevin Liu:

- No, there was no search/filter function with dynamically populated list of properties, the forms are mainly in text box format, not dropdown options,.
- There is no insert function for the intersection table. For insertion made to other entities, the FK attributes are included.
- Based on the project and database outline, there doesn't appear to be any NULLable relationship. The ERD and Schema do not show optional participation either. However, many of the one to many relationships can be changed to optional participation according to the entity description. The group just has to decide which one to change to NULLable relationship
- For the HTML UI portion of your draft, I would suggest removing the DELETE forms on each entities. This way, the end user can just click the delete button next to the row instead of having to fill out all the details of the particular entry to make the delete action. Other than that, I would suggest to simplify it to meet the requirements of the Project guidelines, for instance, you do not need UPDATE and DELETE on all the entities. This way you can have more time and resources to focus on getting the required parts of the project correct and functional.

From Alex Wong :

- I did not see a select that searches or produces a list of properties. The forms are all text input fields.
- The provided DML.sql file with example sql queries did not show inserting into an intersection table. The other tables do not have FKs as attributes.
- The tables do have delete buttons implemented into the UI. The DML.sql file does have a delete example in it that deletes and Orders and the related OrderProducts, but it may be worth considering if the delete methods should use an ON DELETE CASCADE in the creation of the intersection tables.
- Looking at the DDL file, it appears that there are not any nullable relationships because the relationships between table FKs are not null when creating the table, other than the intersection tables. However I do not think it makes sense for intersection table relationships to be nullable as they need both FKs.
- I personally think the green background is a bit harsh to read and that it would be easier to get to the intersection table tabs if they were linked to on the top bar instead of having to click on Order Info to get to Place an Order.

Actions taken from reviews :

Database side : Based on the reviews, there are a few things we need to add such as CASCADE operations for both UPDATEs and DELETEs in order to correctly manage having the data kept the same. Another thing is adding the search query based on a filter. We also need to add a NULLable relationship which was missed before, we decided to move forward with having employees being optional for orders, therefore the value of

employeeID can be null in the orders. With this in mind we need to make sure we correctly attribute FKs.

For the UI side : Reviewers mentioned lack of consistency with naming which we can fix easily. We are going to fix the UI to correctly work as only a DB manager and not a front facing customer product. Remove the Delete forms since they are not needed. One reviewer mentioned the background color was to harsh so we will use a different color.

Step 2 Reviews

From Martin Nguyen :

Schema does present a model that follows the outline and ERD., besides some naming errors. OrderProducts is backwards in the ERD with it being ProductsOrder instead. Also, the ERD for ProductsOrder has a priceAtSale instead of priceAtOrder. ERD for Orders seem to be missing the supplierID, customerID, and employeeID based off the other entities which do have the FK's listed. I noticed was for Employees and Customers, they were listed to h ave 1:M relationships with "Sales." I am assuming that this should be orders based off the ERD and Schema, or perhaps orders should have been sales instead.
Small typo in the relationship for orders, with OrderProducts being misspelled.
ERD for ProductPurchaseOrders has a unitPrice attribute, while outline has a unitCost.
ERD for PurchaseOrders seems to not have the FK productID.
Supplier outline has purchaseOrders instead of PurchaseOrders.
In the SQL, is all example data shown in the PDF INSERTED?
Customers emails and phone number examples in the PDF do not match the SQL.
Employees emails do not match, and Sara Miller's phone number do not match.
Orders SQL do not match, missing product ID, "total" columns does not print and there is an extra quantity table
Orders Product and ProductsPurchaseOrders and purchaseOrders returned an empty result for me

From Kevin Liu :

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

- The intersection table between Orders and Products has different names.
- Products entity name does not match between ERD and Schema.
- ProductID is in PurchaseOrders entity in database outline but not included in schema or ERD.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- Suppliers entity is singular in ERD and plural in Schema.
- Primary Key in Suppliers is in plural, it should be singular as a attribute.
- The FK is missing in the Customers entity part of database outline.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

- It may be a good idea to expand the indexes in the Schema. That way the grader can check if the attributes within the schema matches exactly what was outlined in the overview section.

The arrows in the ERD for PurchaseOrders and ProductPurchaseOrders need to be fixed to one to many relationships.

- The phone number data type should be text or VARCHAR and not int.

Unable to insert data into OrdersProducts entity. (Typo in the code)
Extra quantity attribute in the Orders entity.

From Quin Hoffman :

- Plural vs Singular for some table names (ProductPurchaseOrders is the primary culprit)
- I would decide on plural or non plural for words within the table title, OrdersProducts vs OrderProducts. I would suggest going with the latter not the former.
- OrderProducts may have issues if multiple of the same order at different dates are ordered, since it's primary key will be cloned.
- Database outline does not go into detail on what specific attributes within tables are used for, though most are self explanatory. VARCHAR variables could benefit from an max number (i.e VARCHAR(250)) to more obviously define that they are in fact for strings rather than single characters.
- Above also applies to uses of DECIMAL, specify (10,2) or whatever else you may need.
- Code is unable to insert data into OrdersProducts table, code attempts to insert data into OrderProducts.
- Code is unable to insert data into ProductsPurchaseOrders table, looks for variable unitPrice instead of unitCost.
- No quantity data inserted into Orders table

From Oliver Mendez Lopez:
The table linking Orders and Products has inconsistent naming across documents. Additionally, there's a mismatch in the naming convention for the Products entity between the Entity-Relationship Diagram (ERD) and the Schema. Moreover, the ProductID field is present in the PurchaseOrders entity within the database outline, but it is absent in both the schema and the ERD.
The Suppliers entity is referred to in singular form in the ERD and Schema. The primary key attribute within Suppliers is mistakenly should be singular. Additionally, the foreign key is missing from the Customers entity in the database outline.
I'm can't to insert data into the OrdersProducts entity due to a typo in the code. Additionally, there's an extra quantity attribute in the Orders entity that shouldn't be there.

Actions taken from reviews:

From the reviews a lot of the mentioned the lack of consistency in plural cases, so we decided with having one plural per attribute instead of ProductsPurchaseOrders now it is PurchaseOrderProducts. We fixed the ERD and Schema to be consistent with the outline. We fixed the insert issues and also took into account adding limit to variables such a DECIMAL(10,2). The sample data inserted now correctly follows the example data below as

some of the values were incorrect as mentioned by reviewers. Also fixed several typos in them so they are now inserted correctly.
Did not understand some of the things mentioned such as Customers not having a foreign key that exists in the outline.

Step 1 Reviews :
Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- "I noticed that the naming isn't consistent when it comes to using camel case or snake case. For example, for the employees entity, there is camelCase used for employeeID while there is snake_case used for first_name and last_name. You guys could maybe do employee_ID and keep everything snake_case or do firstName and lastName and keep everything camelCase. I also noticed that Product and Supplier are singular but, aside from that, the rest of the entities are plural and the rest of the attributes are singular"
- "There are some minor inconsistencies with the attributes naming conversions as it contains both camel case and snake case. Beyond that however, the naming is consistent."

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

  - Yes, all of the entities except for Sales, which has a 1:1 relationship, have a 1:M relationship. One example being, employees have a 1:M relationship with sales, as they explained that multiple sales can be associated with one employee.
  - I'm not sure if the ERD represents the relationship correctly between sales and orders, since it isn't mentioned in the description for the relationships, but the ERD shows that both are required to have one and another relationship.

Actions taken from reviews:

In our reviews a lot of them mentioned inconsistency in entity names which were a mix of snake case and camel case. They also mentioned how the database outline and ERD did not match up as there was a M:N in the ERD while in the outline it was 1:M. Finally one reviewer mentioned adding an additional table to reduce complexity.
Based on these reviews we decided that we would use camelCase for all names, matched up the ERD and the outline, and added an additional PurchaseOrders table to assist with tabling orders from suppliers. Before we had just connected suppliers with Orders and had Orders cover both sales from customers and purchases from suppliers with differentiating through orderType, we thought that keeping it simple would be helpful but after further thoughts it would be more complex and adding this table simplifies this. We really appreciate the amount of feedback we got, we had 6 reviewers who helped us change it into this form it is in now.

For the step 2 review, I started out by adding two additional tables to help facilitate the many to many relationship between products and orders, and products and purchases. I then removed the sales table too since the two added tables already helped simplify the design and removed the need for the sales table. Afterwards, I looked into the relationship between my supplier table and PurchaseOrders tables and used TA feedback to fix the 1:M relationship between the two tables by adding in an additional foreign key to PurchaseOrders and removed purchaseID from the supplier table.
Team names:

Project Outline:
A certain sporting goods store sells over 40 million dollars in various sporting goods each year, a huge increase from previous years. To handle this high volume in sales, a database driven website will record the sales of products to customers, as well as the employee who conducted the sale and inventory of the item. Sales will record the details of the order to the customer, such as the order name, status, and date it was one. Sales will also record the customer's name and the employee who placed the order. The orderID in sales will give the order name, quantity, and the product details. The employees will record first and last name, contact info, and position. Inventory will take productID and the quantity of the product. The product will have a distinct category based on the sport, name of item, cost, and price.

Database Outline
Employees : record of the employees at the store

- employeeID: int, auto_increment, unique, not NULL PK
- firstName VARCHAR, not NULL

- lastName VARCHAR, not NULL
- email: VARCHAR
- phone: int
- role: VARCHAR, not NULL

Employees have a one to many relationship with sales as multiple sales can be associated with one employee.

Customers: records detail of each customer

- customerID: int, auto_increment_unique, not NULL PK
- firstName: VARCHAR, not NULL
- lastName : VARCHAR, not NULL
- email : VARCHAR
- phone : int
- address : VARCHAR

Customers have a one to many relationship with Orders as multiple sales can be associated with one customer.

Orders : record details of sale orders

- orderID : int, auto_increment_unique, not NULL PK
- supplierID: int, not NULL FK
- productID : int, not NULL FK
- customerID int,, not NULL FK
- employeeID: int, not NULL FK
- orderType : VARCHAR, not NULL
  Determines the delivery method, retail store purchase, online delivery, and online pickup.
- orderStatus : VARCHAR not NULL
- orderDate: DATE, not NULL
- total : DECIMAL, not NULL

Orders has a many to many relationship with OrderProducts as multiple products can be in multiple orders.

OrderProducts : Facilitates the many to many relationship which will contain Foreign Keys productID and orderID. Will also include time of sale and quantity.

OrdersProducts:

- productID : int, not NULL FK
- orderID : int, not NULL FK
- priceAtOrder : int, not NULL
- quantity : int, not NULL

ProductPurchaseOrders : Second intersection table that contains foreign key productID and purchaseID

ProductsPurchaseOrders :

- productID with reference to Product
- purchaseID with reference to PurchaseOrders
- unitPricet : decimal, not NULL
- quantity: int, not NULL

Product : records product that are sold, its category, price, and quantity

- productID : int, not NULL PK
- category: VARCHAR, not NULL
  5 sports categories, Basketball, Tennis, Golf, Soccer, and Football.
- name: VARCHAR, not NULL
- price: decimal, not NULL
- quantity: int, not NULL

Product has a one to many relationship with ProductPurchaseOrders and also ProductOrders as one product can be in multiple product orders (sales orders) and also purchase orders

PurchaseOrders : records purchase orders from suppliers

- purchaseID: int, auto_increment_unique, not NULL PK
- productID: int, not NULL FK
- purchaseStatus: VARCHAR not NULL
- purchaseDate: DATE, not NULL
- supplierID: Foreign Key with reference to Supplier

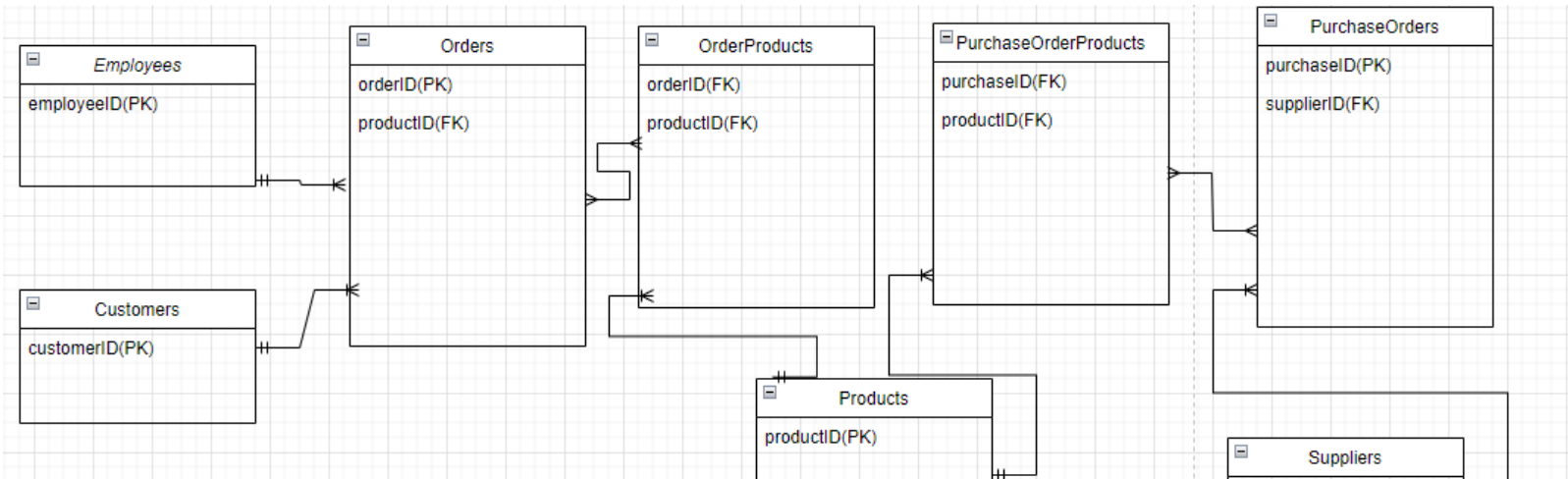
Order has a many to many relationship with ProductOrders as multiple products can be in multiple orders.

Supplier : records supplier of products to be sold

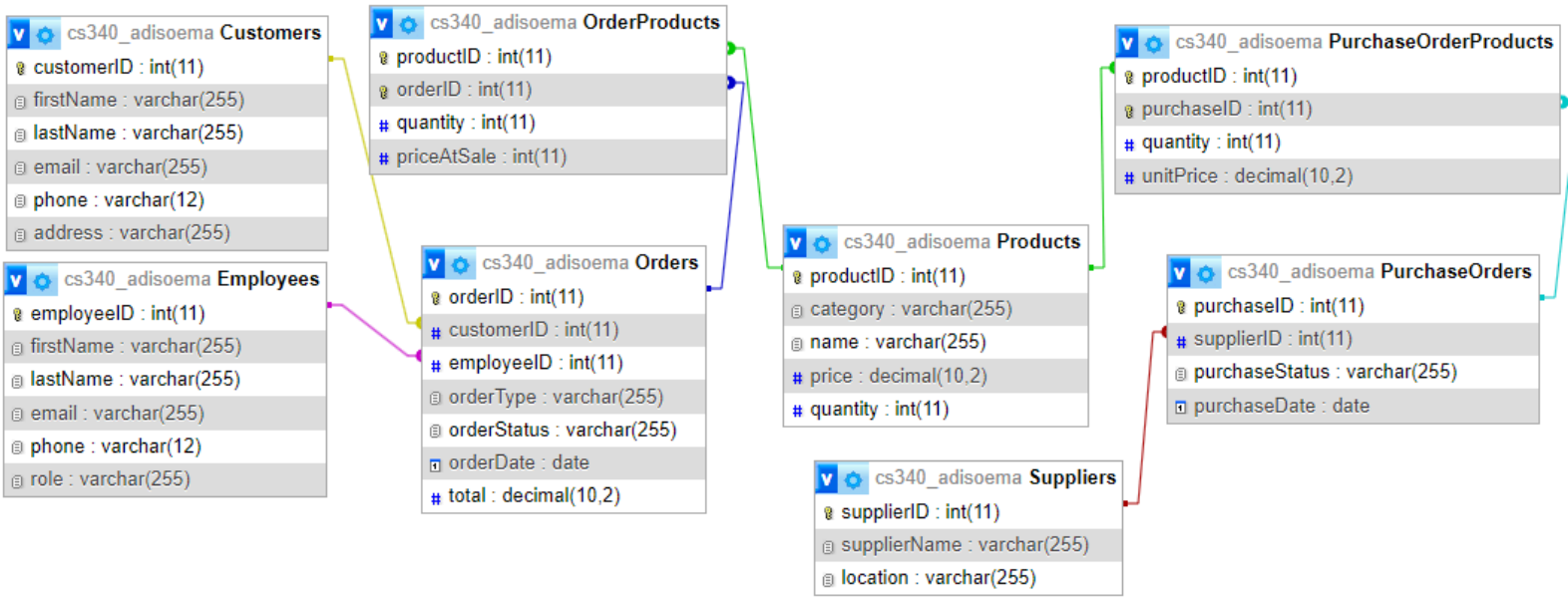
- supplierID: int, auto_increment_unique, not NULL PK
- supplierName: VARCHAR not NULL
- location: VARCHAR not NULL
  Location is where the supplier is based, a local supplier vs international.

Supplier has a one to many relationship with purchaseOrders as suppliers can have multiple orders.

ERD Diagram



d. Schema



e. Example Data

Employees	EmployeeID (PK)	firstname	lastname	email	phone	role				
	1	John	Wick	johnwick@example.com	2062062062	Sales Associate				
	2	Matt	Russo	mattrusso@e	2092092092	Manager				

				xample.com						
	3	Sarah	Miller	sarahmiller@example.com	3513513513	Sales Associate				
Customers	Customer ID (PK)	firstname	lastname	email	phone	address				
	1	Michael	Wilson	michaelwilson@example.com	1234567891	221 Jane St				
	2	James	Turner	jamesturner@example.com	1234567892	341 River Ave				
	3	Emily	Johnson	emilyjohnson@example.com	1234567893	777 Mission				
Orders	orderID (PK)	productID (FK)	customerID (FK)	employeeID (FK)	orderType	orderStatus	orderDate	total		
	1	1	1	1	Store Pickup	Completed	2024-02-01	120.00		
	2	4	3	3	Retail	Completed	2024-02-03	300.00		
	3	3	2	3	Online	Shipped	2024-02-05	239.00		
OrderProducts	orderID (FK)	productID (FK)	priceAtSale	quantity						
	1	1	59.99	2						
	2	4	39.99	8						

	3	3	49.99	5						
ProductsPurchaseOrders	purchaseID (FK)	productID(FK)	unitPrice	quantity						
	1	1	5.99	40						
	1	3	19.99	30						
	2	3	18.59	60						
	3	2	23.00	20						
product	productID (PK)	category	name	price	quantity					
	1	Basketball	Size 7 Performance Basketball	59.99	30					
	2	Tennis	Winners Tennis Racket	119.99	30					
	3	Golf	Golf Gloves	49.99	50					
	4	Soccer	In door Soccer Ball	36.99	40					
	5	Football	Grit Football Helmet	79.99	25					
purchaseOrders	purchaseID (FK)	productID (FK)	supplierID (FK)	purchaseStatus	purchaseDate					
	1	1,3	50	Processed	2024-01-25					
	2	3	40	Shipped	2024-02-03					
	3	2	80	Completed	2024-02-08					
supplier	supplierID (PK)	supplierName	location							
	1	Best Sports	Chicago							

		Goods								
	2	All Sports Inc	Beijing							
	3	Athletic Supply co.	New Jersey							