

Team 33

Avni Gharpurey, Michelino Gali

URL to index.html:

[https://web.engr.oregonstate.edu/~gharpura/cs\\_340/index.html](https://web.engr.oregonstate.edu/~gharpura/cs_340/index.html)

- a. Project Review Feedback and Changes
  - i. See end of feedback for details on changes.

### Step 3 Feedback

**Reviewer 1:**

- *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 is a SELECT for each table in the schema, including one for every intersection table.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

I don't believe there are any SELECT tables with the integration of a search/filter. It appears each table simply has a list that displays all the records of that entity. An example filter could be to have the ability to filter products by vendor name and only show those with the selected vendor.

- *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, each table has a corresponding INSERT.

- *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).

No, there does not appear to be a mechanism that automatically INSERTs in the intersection table when new values are added into the table. This appears logical, as a new record should not be added in Purchases just because a record was added in Customers. It should only be added when the new customer makes a purchase.

- *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.

Both category/vendors and purchases have corresponding DELETE functions in the html UI. These happen to be the two intersection tables, so it will not affect any other tables' records.

- *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?

Similar to the DELETE function, both category/vendors and purchases have UPDATE functionality. There does not appear to be an actual update form in the html UI, but I assume that will be integrated later. It is up to you guys to implement an editing feature for single entities versus an entire record.

- *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.

Based on the sample data in the html UI, there is no obvious example of a Nullable relationship. It does appear based on the schema in the PDF file that there are numerous examples of this, however.

- *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.*

This website has a clean UI and is easy to navigate. The only suggestion I would make is to make the Home page appear first. It does not make as much sense for that

to be the last page on the navigation bar. Other than that and the suggestions from the earlier questions, it seems this concept would work well as it has been designed!

### **Reviewer 2:**

*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 is a SELECT for each table in the schema, each on a different page. However, none of the pages display the table ID's, which from a design perspective I think is nice, but the example provided in the Step 3 Draft assignment page included them, and I double-checked with a TA who also said they should be included.

*Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

- I see dropdowns in the UI which will all presumably require a SELECT with a filter, but I don't see any queries in the dml file that would enable a dropdown feature by utilizing a filter, i.e.: `SELECT ... WHERE ...`

*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 sections on each page for adding an attribute to the respective 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, productID, qty, price and line\_total).

- No, I don't think so. Based on the schema, I'm seeing two different M:M relationships (Customers:Products with Purchases as intersection, and Vendors:Categories with both Categories\_Vendors and Products as intersection tables)? It looks like Categories and Vendors should each always be associated with the other, but the UI doesn't have a field for creating this

association when adding an element to either table. Although this is a little confusing because the relationships are optional through the Products table and not optional through the Categories\_Vendors table. Maybe you just missed the optional relationship in the schema? This is kind of a separate issue, but I would consider re-thinking the relationships for these tables and double-checking that your schema matches your outline.

- For the Purchases table, the UI does have dropdowns for the customer's first and last name and the product name, so creating a purchase this way should also insert the corresponding foreign keys, but the insert statements in the dml don't reflect this. You need to add some subqueries to inserts to get the appropriate foreign key associated with the product name or customer first/last name. The same is true for the Categories\_Vendors and Products pages and insert statements.

*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 for Categories\_Vendors that removes the relationship between a Category and a Vendor, and a DELETE for a Purchase which would remove the relationship between a Customer and a Product. There are also buttons in the UI for deleting these attributes.

*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?

- There are UPDATES for Purchases and Categories\_Vendors and corresponding "Edit" buttons in 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.

- The relationship between a Customer and Product is nullable through updating a Purchase, and it looks like the same should be true for a Category and Vendor through updating the Categories\_Vendors table although the schema has this relationship as not optional. Again, the update queries are directly referencing the id's, though, so these should be changed to use subqueries.

*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 think the UI looks great, simple and easy on the eyes. I don't see any aliases being used in the dml, but the UI shows the names of columns in a readable aliased form, so you will need to add aliases if you want them to match your hardcoded column titles when everything is connected. The index page is the Home page, so that was the first page that I landed on when clicking the link. I don't think there's anything wrong with having the Home link on the far right of the nav bar.

Your UI design looks good and overall I think your project works well aside from my notes about the M:M relationship between Categories and Vendors, which hopefully is just a misunderstanding on my part. My other overarching advice is that you should have subqueries for the foreign keys when adding new attributes, because the user isn't going to enter in the id number for any given entry in a table when creating or updating a row. The UI already reflects this, so that's good. I wish you the best in finishing up your project!

### **Reviewer 3**

- *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. I do think table ID's need to be included though.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

I don't see anything that would fulfill the requirement of a search/filter feature. A dropdown window is present but I'm assuming the SELECT portion hasn't been implemented yet.

- *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!

- **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).

As of right now I don't see anything that will **INSERT** or interact with this relationship. The UI doesn't seem to have something that will create an appropriate relationship that would meet this requirement.

- **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 multiple **DELETE**'s, from categories\_vendors as well as purchase.

- **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?

There are **UPDATE** cascades present, which would indeed allow for the editing of an entity which would fulfill this requirement!

- **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.

I can't find a relationship that would be **NULLable**. I think an option would be for the the relationship between a customer and product as a customer could potentially not buy anything.

- **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 think the UI looks really good and is very simple and not too confusing either. I think personally I would prefer the Home navigation page to be earlier than appear later if

that makes sense. But I think that'll differ from person to person! Either way great project so far!

#### Reviewer 4

Want to start off by saying I like the color palette, the cyan(?) buttons and the cream background looks nice. Not a huge fan of the blue tables, I thought I had accidentally highlighted everything at first lol.

- *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 is a select for every tables. Everything is very cleaned up so it feels like it is geared towards customers/users. I believe the app is supposed to be an admin menu, so things like being able to see the ID of the data would be helpful.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
  - There are drop downs under some of the sections when adding and I am guessing this will be used to dynamically generate options.
- *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 it does have an Insert for every table. There are also corresponding UI elements to add the necessary data.
- *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).
  - i don't think so. The UI does not let users directly see or interact with this relationship. Looking at the given sql file It doesn't look like the insert would go into both tables.
- *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.
  - Both category/vendors and purchases have corresponding DELETE functions in the UI.

- ***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?
  - Purchases and Category/Vendors has an edit button which can be used to edit an 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.
  - It does not look like there is an obvious nullable relationship. The relationship between customer and product should be nullable as you can go shopping and not buy anything.

## Step 2 Feedback

### Reviewer 1

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

The schema presents the database outline and the ER diagram correctly. For future reference, it is possible to drop regular attributes from ERD and keep only PKs and FKs.

- 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?

There are a few typos in the document. In product table: vendor\_id attribute, there is a double comma ( Vendor ID: vendor\_id: int,, FK) In department table: department\_id attribute is missing a data type. (I guess you meant it to be an int) In various tables, foreign keys like vendor\_id, department\_id are correctly data typed. However, in some tables, the data type is missing. This might create a referential integrity problem.

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

The schema is easy to read

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

The intersection table looks like properly formed. Two FKs facilitate a M:N relationship.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

I have checked all tables and could not find any partial or transitive dependency problem. This design conforms to 3NF rules.

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

SQL file is correct. The tables and example data are created without any error.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

The data types in the SQL script align with the descriptions provided in the database outline.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

Each table defines a primary key appropriately using the PRIMARY KEY constraint. Foreign keys are correctly defined in the Products, Purchases, and Categories\_Vendors tables, referencing the primary keys of related tables. CASCADE operations are declared for the foreign keys.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes, SQL script includes the Categories\_Vendors table for that.

- In the SQL, is all example data shown in the PDF INSERTED?

Yes, they are all presented. There are minor mismatches, but it is not like a major problem.

- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

Comments are present in the script, indicating the purpose of each section, such as creating tables, inserting data, and setting SQL modes.

## Reviewer 2

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

The schema does follow the logical structure of the ER diagram and the database outline. I can see that auto\_increment properties and default Null are present in the SQL file but are not visualized in the schema diagram, which is normal.

- 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?

Overall, there is good consistency in naming conventions across the overview, outline, ER diagram, and schema with entities being plural and attributes singular across the documents.

Entities are capitalized while attributes are in lower case with underscores separating words (snake\_case). All of them follow naming convention. (even though I am not sure if camelCase is preferred over snake\_case when naming attributes).

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

The schema is easy to read, and it is very clear and consistent with the outline structure. I do not see any issues here.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

The intersection table is properly formed as it includes the primary key of its own (`category_vendor_id`) and has two foreign keys (`category_id` and `vendor_id`) that reference the respective primary keys of the `Categories` and `Vendors` tables. This structure allows the table to facilitate a many-to-many relationship between categories and vendors. It makes sense to me.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

All of your tables have achieved 3NF. there are no signs of partial or transitive dependencies within the tables. This suggests that the database is normalized. This is the case because I can see that all tables are at least in 1NF, and they are also in 2NF for having no partial dependency. Finally, no column is dependent on non-primary key columns, which makes the database 3NF.

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

Upon reviewing your SQL file, I did not detect anything out of order. Everything seems correct and the schema was generated without errors. It loaded into php server just fine without any errors.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Great job here. Yes, the data types are appropriate considering the description of the attribute. Names use `varchar`, date uses `date`, IDs use `integers`, and prices use `decimal`.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate `CASCADE` operations declared?

The primary and foreign keys correctly defined and they match the Schema's design. I see that you have correctly used `ON DELETE CASCADE` and `ON UPDATE CASCADE` on multiple tables. Great job.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes, the `Categories_Vendors` intersection table correctly connects `Categories` and `Vendors` in the SQL as described on the schema. Other tables are also connected correctly as intended.

- In the SQL, is all example data shown in the PDF INSERTED?

There is consistency and I can see that example data were used correctly. There is a minor mismatched of Chequita and Chiquita in the Vendors table, other than that, everything else is matched.

- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

The SQL file is generated from mySQL workbench. The author wrote his team's member names on it with date and time. The queries are formatted to show that tables are being created first and the second portion includes data insertion. Often times, I have found that Auto correction and Alter tables that are generated from the SQL engines can cause confusion and difficult to follow, but with the format you presented, I can clearly see ON CASCADE function being used and where it was used exactly. Good job.

### Reviewer 3

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

The schema shows a physical model that matches the ER diagram and outlines exactly. All the relationships that the schema shows match the description within the outline. The names for all the entities and attributes match, along with their data types.

- 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?

There is consistency within the naming conventions for all 4 diagrams/outlines. All entity names are capitalized, and they are plurals. The attributes are named to be singular. There are no typos throughout the document.

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

The schema provides a clear understanding of the flow of the system. There are no crosses between the relationship lines, and the tables are placed with appropriate spacing in between so it does not look crowded. The variable names are clear as they provide enough information for users to understand what they are used for.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

The Categories\_Vendors table is properly formed. It has the 2 FKs it needs (vendor\_id and category\_id), and it correctly shows the M:N relationship since the Vendors and Categories both have a 1:M relationship with it.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

All data is normalized. There is no sign of any dependencies

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

The SQL is syntactically correct. When I enter it into PhPMyAdmin, it does not receive any errors, and the tables are created and data is inserted correctly.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

All data types within the SQL files follows what is described in the database outline

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

The primary keys and foreign keys are correctly defined and they match exactly to the schema.

The CASCADE operations are declared appropriately

- In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes, all the relationships within the SQL match the ERD/schema. Even the intersection table is present

- In the SQL, is all example data shown in the PDF INSERTED?

In the SQL most of the example data from the PDF was inserted. There were a few typos within the SQL. For example for the 'Vendors' table, the last data in the SQL was ('chiquita', '2018-05-15), which the pdf has ('chequita', 05/15/2018). That may have been a typo or an autocorrect. The rest of the data from the SQL match the PDF exactly.

- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

The SQL was exported from MySQL but it was reformatted into a well-structured script that is easy to understand. The authors did add comments like when a table is made and when a table was completed

## Reviewer 4

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

The schema depicts a physical model with precise outline and matching to the ER diagram. Every relationship displayed by the schema corresponds to the outline's description. All of the entities, properties, and data types have identical names.

- 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?

The naming rules for each of the four diagrams/outlines are consistent. The entity names are all plural and capitalized. The names of the qualities are singular. The document is error-free throughout.

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

I thought the scheme was quite organized and easy to follow.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, the Categories\_Vendors table has been formatted correctly. Since the Vendors and Categories have a 1:M relationship with it, it has the two FKs it requires (vendor\_id and category\_id), and it displays the M:N relationship correctly.

- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

Every table you have obtained 3NF. The tables don't appear to have any transitive or partial dependencies.

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

Syntactically, the SQL is accurate. It creates the tables and inserts the data successfully when I enter it into PhPMyAdmin without receiving any issues.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Every data type in the SQL files complies with the database outline's description.

- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

Both the primary and foreign keys are precisely defined and align with the schema. The CASCADE activities have been duly declared.

- In the SQL, are relationship tables present when compared to the ERD/Schema?

Indeed, every relationship in the SQL matches the schema and ERD. There is even an intersection table.

- In the SQL, is all example data shown in the PDF INSERTED?

Consistency exists, and it appears that the example data were handled appropriately. Everything else in the vendors table matches, with the exception of a small mismatch between Chiquita and Chequita.

- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?

Having been extracted from MySQL, the SQL was restructured into a comprehensible script. When a table was created and finished, for example, the writers included remarks.

## Record of Changes

Step	Source	Feedback	Changes
1	Peer Feedback		Removed optional extra line between 'products' and 'categories'
1	Peer Feedback		Added product_id PK in Products (Inserted and connected to Purchases instead of an intersection table)
1	Peer Feedback		Added line between Purchases product_id to products product_id (1:M between products and purchases so the crows feet are on the purchases side)
1	Peer Feedback		Reversed the 1:M relationship between Categories_Vendors and Vendors (so the crows feet are on the Categories_Vendors side)
1	Peer Feedback		Reversed the 1:M relationship between Categories_Vendors and Categories (so the crows feet are on the Categories_Vendors side)

1	Peer Feedback		Corrected inconsistency to show Categories_Vendors as the Table name in the outline
1	Peer Feedback		Added default 0 to product_count in outline and diagram
1	Peer Feedback		Updated vendor_start_date to not NULL in outline and diagram
1	Peer Feedback		Added not NULL to category_name in outline and diagram
1	Peer Feedback		Added PK to Purchases called total_purchases
1	TA Feedback	Clarify what problem is being solved and why a database is useful to solve it.	Clarified the problem being solved and why a database is useful.
1	TA Feedback	Reupload the ER Diagram so it is easier to read	Reuploaded the ER Diagram so it is easier to read
1	Self Review		Added category_vendor_id primary key to Category Vendor table
1	Self Review		Removed total_purchases from Purchase table, added purchase_id as primary key instead
1	Self Review		Made foreign key references NULL instead of NOT NULL, which allows new values to be added even if foreign key references are not available
2	Peer Feedback	Fix typos	Fixed typos
3	Peer Feedback	Include table IDs in UI	Added table IDs in UI
3	Peer Feedback	Add SELECT to DML for dynamically-populated dropdown list	Added select to DML for drop down list
3	Peer Feedback	Add 'where' clause for filtering	Choosing not to add at this time. This would require some UI redesign. This could result in scope creep and filtering is not a requirement based on project specifications. We may come back to this as the project progresses.

3	Peer Feedback	Add operations in the DML that inserts the foreign key ID into Purchases based on the provided customer name	Added operations in the DML that inserted foreign key IDs into Purchases and Categories-Vendors based on the provided name for Updates and Creates
3	Peer Feedback	Ensure at least one relationship is Nullable from the DML.	The relationships are all nullable because NULL is already allowed from the DDL. If a name like “-” is provided that doesn’t match an existing name, a NULL value will be added.
3	Peer Feedback	Add aliases in the DML	Added aliases
3	Peer Feedback	Add data to drop downs for clarity in UI	Added data to drop downs
3	Peer Feedback	Move Home Page navigation to front	Moved Home Page navigation link to the beginning of the navigation links.
3	Self Review	Clarify why Categories-Vendors only has dropdowns in the ‘add’ section in the UI.	Added clarification, also added clarification that Products will do the same thing
3	Self Review	Update Purchases section to have Customer Name as one section, matching schema	Updated purchases section to have customer name as one section matching schema
3	Self Review	Make sure a new categories/vendors relationship is added if a Product is added that adds a new category/vendor relationship. Also add note about it to UI	Added ‘insert where not exists’ under Products to update Categories Vendors table when new category/vendor relationship is created from the Products table.

## b. Project Outline and Database Version

**Overview:**

The local grocery store needs a system to help keep track of the multiple products they sell, as well as the vendors they buy the products from and the customers they sell the products to. They also need to keep track of the orders they complete. Developing a database will allow them to keep track of inventory and revenues on a day-to-day basis, as well as perform long-term analysis to see which products and vendors are most popular.

This grocery store serves 1000 customers and has 30 vendors, with each vendor selling between 1 and 5 products. There are about 5000 product purchases per week. A database driven website will record *Purchases* of *Products* by *Customers*. *Products* are owned and sold by *Vendors*. *Products* are also part of *Categories* and store *Departments*.

## Database Outline: Post-Normalization

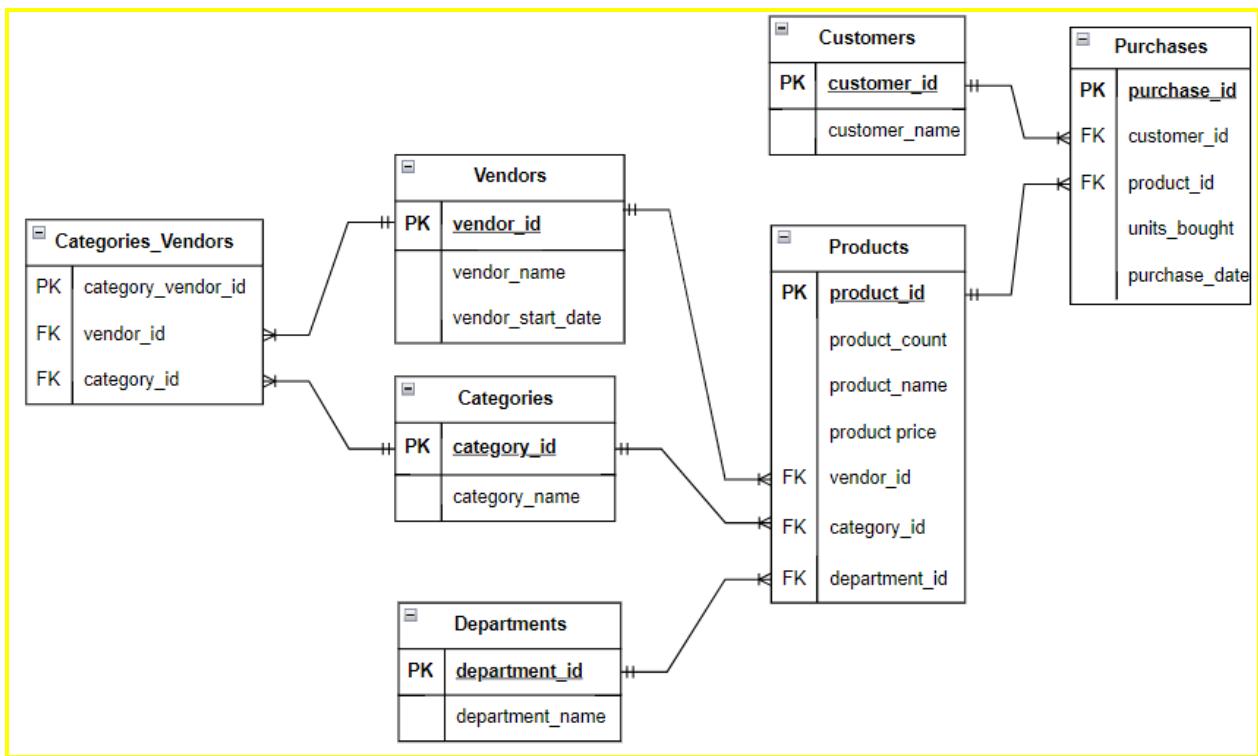
This database has 6 entities: **Vendors**, **Customers**, **Products**, **Departments**, **Categories**, and **Purchases**. There are 5 relationships: **Vendors-Products (1-M)**, **Departments-Products (1-M)**, **Categories-Products (1-M)**, **Customers-Purchases (M-1)**, and **Categories-Vendors (M-M)**.

- **Products:** Includes information about products sold in the grocery store. Includes product name, ID, category, vendor, and department.
  - Product ID: **product\_id**: int, auto\_increment, unique, not NULL, PK
  - Units of product: **product\_count**: int, NULL, default 0
  - What product is: **product\_name**: varchar(100), not NULL
  - Price of product: **product\_price**: decimal(12,2), not NULL
  - Vendor ID: **vendor\_id**: int, FK
  - Department ID: **department\_id**: int, FK
  - Category ID: **category\_id**: int, FK
- **Relationship:** A required (both sides) 1:M relationship between Products and Vendors is implemented with vendor\_id as a FK inside of Products
- **Relationship:** A required (both sides) 1:M relationship between Products and Departments is implemented with department\_id as an FK inside of Products
- **Relationship:** A required (both sides) 1:M relationship between Products and Categories is implemented with category\_id as an FK inside of Products
- **Vendors:** Includes information about vendors of the products sold in the grocery store. Includes vendor name, ID, and the date the vendor began selling their first product in the store.
  - Vendor ID: **vendor\_id**: int, auto\_increment, unique, not NULL, PK
  - Vendor Name: **vendor\_name**: varchar(100), not NULL
  - Vendor Date: **vendor\_start\_date**: date, not NULL
- **Relationship:** A required (both sides) 1:M relationship between Products and Vendors is implemented with vendor\_id as a FK inside of Products

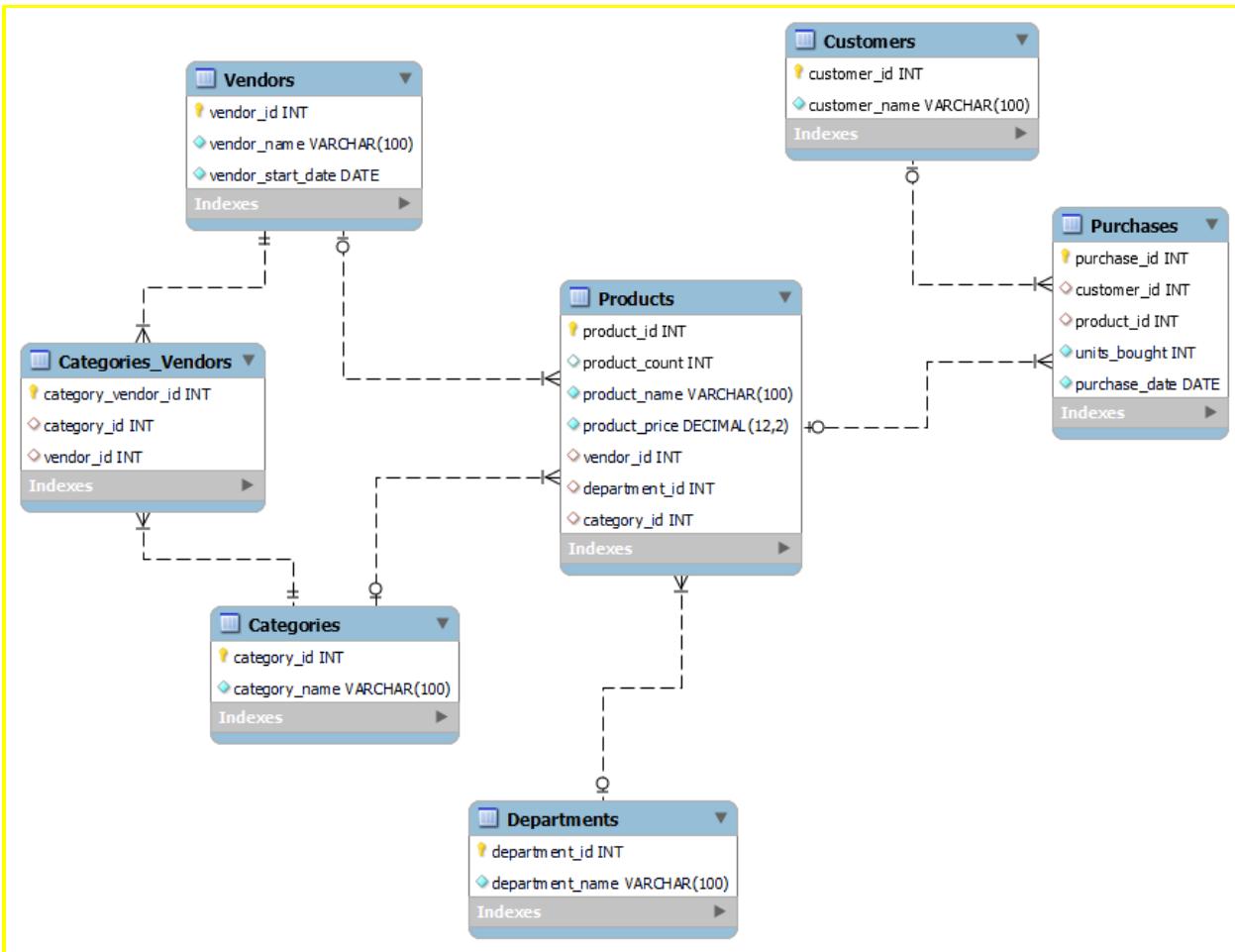
- **Departments:** Includes information about departments in the grocery store. Includes the department name and ID.
  - Department ID: department\_id: int, auto\_increment, unique, not NULL, PK
  - Department Name: department\_name: varchar(100), not NULL
  - **Relationship:** A required (both sides) 1:M relationship between Products and Departments is implemented with department\_id as an FK inside of Products
- **Categories:** Includes information about product categories, includes category ID and category name.
  - Category ID: category\_id, int, auto\_increment, unique, not NULL, PK
  - Category Name: category\_name, varchar(100), not NULL
  - **Relationship:** A required (both sides) 1:M relationship between Products and Categories is implemented with category\_id as an FK inside of Products
  - **Relationship:** An M:M relationship between Categories and Vendors is implemented using an intersection table with foreign keys of vendor\_id and category\_id inside the Categories-Vendors Intersection Table.
- **Categories\_Vendors:** An intersection table that connects the multiple categories to the multiple vendors. This intersection table is used because one vendor may have many product categories, and one product category may have multiple vendors.
  - Category Vendor ID: category\_vendor\_id, int, auto\_increment, not NULL, PK
  - Category ID: category\_id, FK
  - Vendor ID: vendor\_id, FK
  - **Relationship:** A M:M relationship between Categories and Vendors as described in the intersection table with foreign keys of vendor\_id and category\_id.
- **Purchases:** Includes information about purchases of products by customers. Includes customer ID, product ID, units bought and date of purchase.
  - Purchase ID: purchase\_id, int, auto\_increment, not NULL, PK
  - Customer ID: customer\_id: int, not NULL, FK
  - Product ID: product\_id: int, not NULL, FK
  - Units Bought: units\_bought: int, not NULL
  - Date of purchase: purchase\_date: date, not NULL
  - **Relationship:** A required (both sides) 1:M relationship between Products and Purchases is implemented with product\_id as an FK inside of Purchases
  - **Relationship:** A required (both sides) 1:M relationship between Customers and Purchases is implemented with customer\_id as an FK inside of Purchases

- **Customers:** Includes information about customers. Includes customer ID and customer name.
  - Customer ID: customer\_id: int, auto\_increment, unique, not NULL, PK
  - Customer Name: customer\_name: varchar(100), not NULL
- **Relationship:** A required (both sides) 1:M relationship between Customers and Purchases is implemented with customer\_id as an FK inside of Purchases

### c. ERD



## d. Schema



## e. Example data

### Products

- **Products:** Includes information about products sold in the grocery store. Includes product name, ID, category, vendor, and department.
  - Product ID: **product\_id**: int, auto\_increment, unique, not NULL, PK
  - Units of product: **product\_count**: int, NULL, default 0
  - What product is: **product\_name**: varchar(100), not NULL
  - Price of product: **product\_price**: decimal(12,2), not NULL
  - Vendor ID: **vendor\_id**: int, not NULL, FK
  - Department ID: **department\_id**: int, not NULL, FK
  - Category ID: **category\_id**: int, not NULL, FK

Product_id PK	product_count	product_name	product_price	vendor_id	department_id	category_id
1	30	Cheerios	\$3.50	1 (from vendors)	3 (from departments)	2 (from categories)
2	10	Ketchup	\$4.00	7 (from vendors)	6 (from departments)	5 (from categories)
3	30	Fruit Loops	\$3.00	1 (from vendors)	3 (from departments)	2 (from categories)
4	80	Advil (200 tablets)	\$18.49	5 (from vendors)	7 (from departments)	6 (from categories)
5	12	Almond Milk (half gallon)	\$3.38	4 (from vendors)	1 (from departments)	1 (from categories)

## Vendors

- **Vendors:** Includes information about vendors of the products sold in the grocery store. Includes vendor name, ID, and the date the vendor began selling their first product in the store.
  - Vendor ID: vendor\_id: int, auto\_increment, unique, not NULL, PK
  - Vendor Name: vendor\_name: varchar(100), not NULL
  - Vendor Date: vendor\_start\_date: date, not NULL

vendor_id	vendor_name	vendor_start_date
1	Kellogg's	5/1/1941
2	Best Groceries	10/1/2022
3	The Food People	12/14/2021
4	Blue Diamond	1/3/1910
5	Pfizer	3/8/1974
6	Quaker	6/8/1877
7	Kraft/Heinz	3/28/1915
8	Dole	10/07/2021
9	Chiquita	05/15/2018

## Departments

- **Departments:** Includes information about departments in the grocery store. Includes the department name and ID.

- Department ID: department\_id: auto\_increment, unique, not NULL, PK
- Department Name: department\_name: varchar(100), not NULL

department_id	department_name
1	Dairy
2	Produce
3	Dry Goods
4	Meat
5	Fish
6	Food Misc.
7	Pharmacy

## Categories

- **Categories:** Includes information about product categories, includes category ID and category name.
  - Category ID: category\_id, int, auto\_increment, unique, not NULL, PK
  - Category Name: category\_name, varchar(100), not NULL

category_id	category_name
1	Milk
2	Cereal
3	Cheese
4	Fruit
5	Condiments
6	Ibuprofen
7	Snacks
8	Breakfast

## **Categories\_Vendors**

- **Categories\_Vendors:** An intersection table that connects the multiple categories to the multiple vendors. This intersection table is used because one vendor may have many product categories, and one product category may have multiple vendors.
  - Category Vendor ID: cat\_vendor\_id, int, auto-increment, not NULL, PK
  - Category ID: category\_id, FK
  - Vendor ID: vendor\_id, FK

cat_vendor_id	category_id	vendor_id
1	1	4
2	2	1
3	2	6
4	3	7
5	4	8
6	4	9
7	5	7
8	6	5
9	7	1
10	8	1

## **Purchases**

- **Purchases:** Includes information about purchases of products by customers. Includes customer ID, product ID, units bought and date of purchase.
  - Purchase ID: purchase\_id, int, auto\_increment, not NULL, PK
  - Customer ID: customer\_id: int, not NULL, FK
  - Product ID: product\_id: int, not NULL, FK
  - Units Bought: units\_bought: int, not NULL
  - Date of purchase: purchase\_date: date, not NULL

purchase_id	customer_id	product_id	units_bought	purchase_date
1	3	1	2	01/06/2024

2	3	4	3	01/06/2024
3	3	2	2	01/06/2024
4	5	1	1	01/14/2024
5	1	3	5	01/29/2024
6	1	5	4	01/29/2024

- **Customers:** Includes information about customers. Includes customer ID and customer name.
  - Customer ID: customer\_id: int, auto\_increment, unique, not NULL, PK
  - Customer Name: customer\_name: varchar(100), not NULL

customer_id	customer_name
1	Joey Tribbiani
2	Chandler Bing
3	Rachel Green
4	Phoebe Buffay
5	Monica Geller
6	Ross Geller