

Hayden Cole
 Virginia Link
 CS 340
 25 February 2021

Database Project Outline - Step 4

Project Link:

<http://flip1.engr.oregonstate.edu:8888>

We have included peer reviews at the beginning of this document. Peer reviews from Step 2 have been attached at the end of this document.

Peer Reviews from Step 3:

Christopher Eckerson:

- Yes, there is a select for each of the five entities from design.
- Actually it seems that all tables include a search option.
- Yes, each insert is included with each table and has all the attributes.
- Yes, there appears to be this between tables. For example, the payments include the foreign key from customer.
- There are delete options on each table. Because the interaction tables are shown (M:M), it is harder to know if the UI will show this happening. It can be assumed from the design that there would be updating in these tables.
- Yes, there is option to update on nearly every table.
- There is examples of NULL entry option in both the payments and gear tables.
- Everything looks covered well but since there class is interested in seeing that the database correctly updated when M:M relationship are deleted, it may be good to include the interaction tables as well.

Jon Frosch:

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.*

Most of the entities appear to be covered by selects. It is not clear to me whether data from Customers_Employees and Employees_Services is used in a select.

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

Employees, Customers, Payments, Services, and Gear all have at least 1 text search/filter. It is not clear from the mockup whether any of these will be dynamically populated, but I think the requirement is satisfied with a text search.

3. 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.

Employees, Customers, Payments, Services, and Gear all have forms that allow adding new instances to their respective tables. The Customers_Employees and Employees_Services tables both do not appear to have interfaces that allow insertion of new records.

4. 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).

Payments and Gear both have fields to input appropriate FKs in their insertion interfaces. Customers_Employees and Employees_Services do not seem to have interfaces that allow manipulating the required FKs.

5. 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.

Employees has an interface to delete. I do not see a way to delete from Customers_Employees or Employees_Services, which implement the m:m relationships.

6. 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?

Gear can be updated.

7. 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 is possible to null out the relationship between Gear and Employees by setting the FK to null via the update interface.

8. Do you have any other suggestions for the team to help with their HTML UI?

The overall feel of the interface is very clean. I'd suggest making the interfaces to the m:m relationships more clear if they are already implemented, since I could not find them.

Amelia Walsh:

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. Each of this groups tables is clearly linked to a webpage where a SELECT query will be implemented as a means of retrieving data from it's corresponding table.

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

Yes. There is a search bar attached to each individual table within the webpage. It implies that when the webpage is fully functional the user will be able to filter the data by certain properties within the table.

3. *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 webpages that represent each of the tables each contain an "add" form that will allow users to add data to a table. This will require an INSERT on the backend.

4. *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 linetotal).*

Not going to lie I am not super clear on what this question is getting at. However, according to the group's ERD they are implementing a M:M relationship between employees and services. The service page has a location for adding services, but it does not include a space to add the employees who will be following through on those services. I am not sure I can tell just by looking at their front end how or whether or not they plan on populating the employee table when a service is added. So, I'll note here that the group should have a plan for populating their employee/intersection table when a service is added.

5. *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. They will be using a DELETE query on their employee's page in order to remove employees. It will be deleting things from one of the tables involved in a M:M relationship. As such will have to be able to cascade changes from the employees table to the services table.

6. *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?*

I believe so. Their gears page has an 'edit' form that I assume would require an UPDATE query on the backend.

7. *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.*

Yes. I would expect this to be the case in more than one location on their website. However, I think this particular constraint would be met by the gear and employee relationship in that there should be able to be an employee that doesn't have any gear checked out or gear that is not checked out by any employees.

8. *Do you have any other suggestions for the team to help with their HTML UI?*

Looks like you all have a great start! No additional suggestions, the pages are clean and easy to read. They clearly meet the standards of this assignment.

Actions based on Peer Feedback:

The main criticisms we received in peer feedback are as follows:

- Can't see whether M:M relationships correctly update as these interactions aren't visible on the website, or whether the data from the M:M relationships are used in a SELECT.
- "So, I'll note here that the group should have a plan for populating their employee/intersection table when a service is added." (From Amelia's review).
- "The Customers_Employees and Employees_Services tables both do not appear to have interfaces that allow insertion of new records." (From Jon's review).

As such, we've made the following modifications:

- Added two new pages, Certifications and Assignments, that respectively address all points of feedback for the Services:Employees and Employees:Customers relationships. Although these pages don't have perfect functionality yet, they should be able to better represent the interactions a user could have with these tables.

Overview:

The database we have chosen to construct is a central database for a pest control company, charged with the important business of exterminating bugs in the physical realm. The pest control company in question has an annual revenue of approximately \$1 million, a fleet of 5 vehicles, and less than a dozen employees. It will store customer information, services billed, equipment/vehicle allocations, and employee information. It will be an internal database that can be interacted with via a web-based UI.

Database Outline:

(If an attribute is not explicitly marked as can be NULL, it cannot be NULL.)

Customers:

Purpose: Customers that the company is currently serving

Attributes:

- *customerID int(10) (PK)
- firstName varchar(50)
- lastName varchar(50)
- phone varchar(50)
- addressLine1 varchar(50)
- addressLine2 varchar(50)
- city varchar(50)
- state varchar(50)
- zip varchar(50)
- currentBill int(10)

Relationships:

- 1:M with services. A customer can only have one service at a time, but a service could have multiple customers that need it at the same time.
- M:M with employees. One customer could get help from multiple employees, and one employee could be assigned to help multiple customers. It is possible to have 0 customers assigned to an employee, but a customer must always have at least one employee assigned to them.
- 1:M with payments. A customer can make multiple payments on their bill, but those payments cannot be shared between multiple customers.

Customer_Employee:

Purpose: Relate the tables Customers and Employees

Attributes:

- *customerID int(10) (FK)
- *employeeID int(10) (FK)

Payments:

Purpose: Keeps track of the payments a customer has paid.

Attributes:

- *customerID int(10) (PK)
- *paymentNum int(10)
- date varchar(50)
- amount int(10) (can be NULL or negative (e.g. if a refund is performed))

Relationships:

- 1:M with customers. Customers could have made multiple payments, but payments aren't shared between customers.

Services:

Purpose: Describes available services offered by the company i.e. inspection, outdoor extermination, indoor extermination, etc.

Attributes:

- *serviceType varchar(50) (PK)
- gearNeeded varchar(50)
- cost int(10)

Relationships:

- M:M with Employees. Tracks which employees are certified to perform the specified service. This is a 1 or many relationship on both ends. If the company offers a service, at least one or more employees must be certified to perform it, and an employee must be certified to perform at least one or more service.
- M:M with customers. A customer could need one or multiple services, and a service could have multiple customers that need it at the same time.

Employee_Service:

Purpose: Relate the tables Employees and Services

Attributes:

- *certifiedEmployees int(10) (FK) (Multiple - can store IDs of certified employees)
- *serviceType varchar(50) (FK)

Employees:

Purpose: Keeps track of current employees on roster.

Attributes:

- *employeeID varchar(50) (PK)
- firstName varchar(50)

- lastName varchar(50)
- phone varchar(50)
- paymentInfo (If the employee uses direct deposit, etc.) varchar(50)
- hoursWorked int(10)

Relationships:

- M:M with customers. A single employee could be helping multiple customers, and a single customer could have more than one employee assigned to help them. It is possible to have 0 customers assigned to an employee, but a customer must always have at least one employee assigned to them.
- M:M with services. An employee can be qualified to perform multiple services, and a service could have multiple employees that are trained to perform it. Must be at least 1 connection with a service.
- 1:M with gear. An employee can check out more than one type of gear at a time for a given service. Can be a 0 or many relationship.

Gear:

Purpose: Keeps track of gear inventory and which employee has what items checked out at any given time.

Attributes:

- *gearID int(10) (PK)
- gearType varchar(50)
- lastServiceDate varchar(50)
- employeeIDCheckedOut int(10) (can be NULL) (FK)

Relationships:

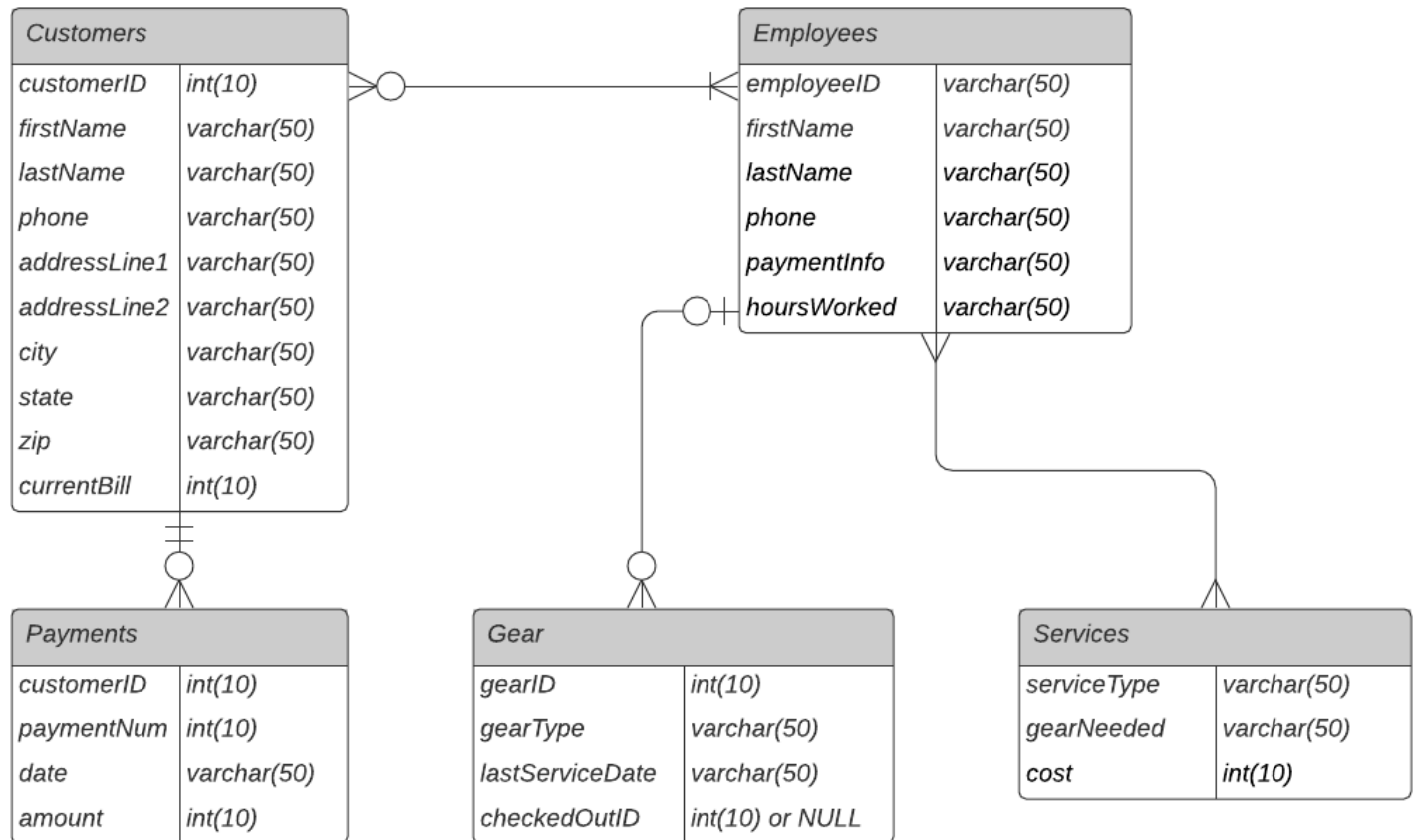
- 1:M with employee. Only one employee can have a given item checked out at once, but that employee could have multiple items checked out. This is a 0 or many relationship.

Division of Responsibilities:

Hayden will be responsible for the gear and employees sections.

Virginia will be responsible for customers and payments.

As services are the logical intersection of these two areas, we will divide that work or redistribute as we see fit as the project moves forward.

Entity-Relationship Diagram:**ERD notes and justifications:**

-A customer can have either zero or many payments. Perhaps they didn't pay yet. This is tracked by the Customer currentBill property.

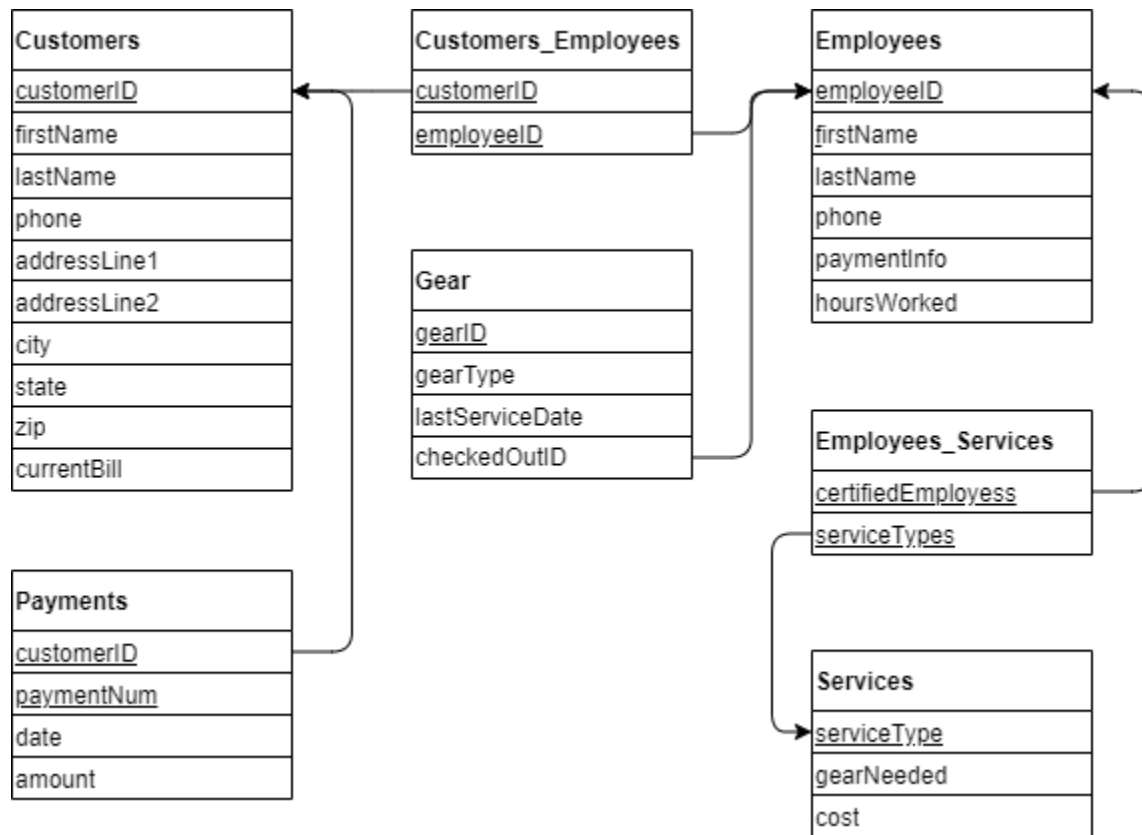
-Gear can be checked out by zero or one employee. Feasibly, multiple employees could be using the same gear, e.g. a van, but for the sake of simplicity, we'll check it out to just one.

-The primary M:M relationship is as follows:

Employees : (are qualified to perform many) : Services and

Services : (can be performed by many) : Employees.

-An employee can be assigned to zero or multiple customers at a time, but a customer must always have at least one (possibly multiple) employee(s) assigned to them at any time.

Schema:

Team Evaluation Form:**Rate your team's performance using the scale below.****1 = Strongly Disagree 2 = Disagree 3 = Agree 4 = Strongly Agree**

<u>Group number</u>	75	
<u>Name of Group TEAM Members:</u>	75 Bugs Detected	
<u>SCALE AND COMMENTS</u>	<u>RATING</u>	<u>ADDITIONAL COMMENTS</u>
<u>How Prepared was your team?</u> Research, reading, and assignment complete	4	We were prepared for the assignment and had no major issues. We were able to work on the assignment asynchronously and combine our work when we were ready.
<u>How responsive & communicative were you both as a team?</u> Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	We stuck with our chosen communication medium (Discord). All responses were reasonably timely and we had no issues with communication.
<u>Did both group members participate equally</u> Contributed best academic ability	4	Yes. We have been working to balance both how much each group member contributes and how to adjust contributions in an equitable manner, based on workload from other classes and obligations.
<u>Did you both follow the initial team contract?</u> Were both team members both positive and productive?	4	We have had no issues with the team contract, and both team members were positive and productive.

Are there any suggestions for improvement for your team and what are your goals moving forward?

We're happy with how this assignment turned out. Hayden did most of the revision work while Virginia laid the groundwork for the website. We continue to work to balance the load whenever possible. :)

Previous feedback by peer reviewers:

Review 1: Amelia Walsh

Does the overview describe what problem is to be solved by a website with DB back end?

The overview provides sufficient detail on the structure of the business. It includes the expected revenue of the company, number of vehicles, and number employees. This is useful information to determine the scope of the database. It also gives a brief description of how the database will be used by the company. This group met the requirements for this section as described in the assignment.

Does the overview list specific facts?

As I stated above it includes revenue, number of cars, and employees. The overview was clear and simple. It was easy to understand what the group intends to do with the database.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, this group lists employees, customers, payments, gear, and services. They also list entities that represent intersection tables that they will use to implement M:M relationships in their database. It seems like they chose appropriate names for each entity, and they have split up the data such that each entity represents a single idea.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

It includes clearly stated purposes for each entity and in some cases provides examples of some of the data that would be included in a table. Each entity includes a relationships section that accurately describes how its data relates to the data held within other entities.

The following things are still missing from the outline:

Needs constraints other than the data type of each attribute (ex. Primary keys must be unique & not NULL)

The outline didn't list which group member would be responsible for implementing each entity. The missing elements are pretty quick fixes and can be easily added!

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, the relationships are very clearly described. There are two M:M relationships. They even include an additional section below the ERD that clarifies why they have chosen particular types

relationships between entities. The ERD does not include the intersection tables that are described in the outline to show how these tables would be implemented, but honestly I am unclear if we are supposed to include them. The schema does include intersection tables which clarifies how these will be implemented for the M:M relationships in the database.

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

For the most part they have followed naming conventions. They have capitalized entity names and use camel casing for attribute names which makes for easy reading. The only thing I would note here is that the intersection tables have singular as opposed to plural names.

Great job overall! Your proposal was easy to read & understand.

Review 2: Patrick Stumbaugh

Does the overview describe what problem is to be solved by a website with DB back end?

Yes, the overview is descriptive and provides clarity of what the problem is and how the DB will be helping solve their problem. It describes how the DB will be efficiently used by the company. This group clearly intends to help (at least what seems like) a very real-world sounding company DB problem!

Does the overview list specific facts?

Yes, the overview gives specific facts about revenue, employees, and vehicles, which is helpful when visualizing what the DB will be working with. It is clear to me that these list items will be benefiting from a DB.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, this group listed more than four entities, which works well for the problem they are looking to solve. They include: Customers, Employees, Payments, Gear, and Services.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Yes, I believe it does describe the purpose of each entity. I would perhaps give slightly more detail on the purpose of each entity. Such as for Customers, you could say "To save contact information about our customers" instead of just saying "Customers that the company is

currently serving". I can infer the Customers entity will do something with customers already. Whereas the purpose will state slightly more about what is in the Customers entity table. I would also suggest adding in more constraints, such as not NULL and showing items as either a PK or a FK. (I see you used * to indicate this, but in some tables there are more than one *, so it would be nice to know which is the PK and which is simply a FK.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, there are clearly stated 1:M relationships throughout this DB. They are clearly written out in the DB outline under each entity table.

Yes, there are multiple M:M relationships in this DB. The group goes on to say, in their ERD notes, that the primary two they will probably be working with are:

Employees : (are qualified to perform many) : Services and

Services : (can be performed by many) : Employees.

The notes section is very helpful describing different things in the tables, btw!

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

Yes, there is consistency among the entity/attributes in the outline, ERD and schema. This group went with the convention of entity tables starting with a capital letter and plural. Then, their attributes are starting lower-case and camelCase.

Lastly, I would perhaps remove the intersection tables and list them below, instead of being part of the ERD and schema (though I'm not sure if this is what we are supposed to do??). The naming convention slightly changes for those intersection tables compared to the entity tables, which is why I note this. But still, it is overall very easy to read and understand the purpose and goal of this DB! Good work!! (also - the description of "... a long night of debugging" was a great add! :)

Review 3: Jaelyn Litzinger

Does the overview describe what problem is to be solved by a website with DB back end?

Yes - to keep track of customers, employees, services, gear, and payments of a pest control business.

Does the overview list specific facts?

Yes - for example revenue and approximate number of employees.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes - customers, employees, services, gear, and payments

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Entity details are clear and thorough. Missing which teammates are in charge of which entities.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, the M:M relationship works between employees and customers. In customers it is noted to have a 1:M relationship with services, but in services it says it has a M:M relationship with customers. I think M:M makes more sense in this case.

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

Yes

Looks great Hayden and Virginia! Just found the two small issues :)

Prior actions based on feedback:

- **From Amelia's review:**
 - Added constraints for data types.
- **From Patrick's review:**
 - Added more detail to entity descriptions; PK/FK designations.
- **From Jaelyn's review:**
 - Assigned implementation responsibilities.
 - Updated Customers:Services relationship to indicate M:M relationship from both sides.

We note that the question of whether intersection tables should be included in the schema or listed separately came up in both Amelia and Patrick's reviews. Canvas feedback:

Your document looks really good. Your ERD and schema are a model for others to follow. Easy on the eyes, easy to follow. Well done.