

Link: <https://luke503.github.io/management-system/>

Project Step #3 Final Version: Design HTML Interface

Project Step 3: Reviews

Verbatim from Piazza:

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.

There is a SELECT for the 4 main tables in the schema.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties? It appears that the selects will return the entire 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.

There is an insert for each table but each one also includes a field to enter the primary key.

Based on the schema the primary key should not be inserted and rather auto incremented.

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).

It does not appear so. Also I think there should be a department_fk in the employees table.

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.

I do not see that option.

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?

Possibly, The submit button on each page could be used to insert or update. For update it would make sense to keep the primary key field.

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 ERD shows that an employee does not need a project.

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

The pages show the table and a submit form, would be nice to have a search feature to find an employee, see who is working on a project, see who has fewer projects when looking for employees to assign to a new project.

The ERD shows a relationship between employees > projects > tasks but the schema shows the relationship as employees > tasks > projects based on how the tables are set up I would assume that the schema is correct.

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.

I see selects for Employee, Departments, Positions, and Projects. From the description I would expect something to show all of the Tasks for a project.

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

I do not see a search or filter.

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.

There is for Employee, Departments, Positions, and Projects but not for Tasks.

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 didn't see anything that links Tasks to Projects, or Positions to Employees, or Employees to Departments.

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.

There's not a delete option.

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's not an update option.

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, employees can be added without a projects

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

There are a few things missing but I think this is a really good start!

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.

It doesn't have any "select" (search) buttons for the tables, but it displays the full table next to the input area.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties? Nothing is dynamic as far as I can tell.

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, you can input into 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, productID, qty, price, and line_total).

I don't see a specific M:M table in here, though a project may allow many employees and vice versa, so perhaps it's being captured that way.

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.

Nope, no delete functionality yet.

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?

As of now, not yet.

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's not super clear to me at this point, but I assume projects could be assigned to no one or there may be no assigned task.

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

Your website is super aesthetically pleasing and the inputs are really good. Maybe just add a navigation menu to the top so you can get to any page from any page and just add the other functionality requirements. Altogether it is really excellently done. Nice job!

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.

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

I can't see anything about the search/filter,

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 is an INSERT for 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, productID, qty, price, and line_total).

I can't see anything about the M:M relationship.

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.

I do not see that option.

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, they use the submit button to realize this.

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.

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

Maybe it's better to add a table in the index page which shows the describe the database system.

Actions Based on the Feedback:

Based upon the four peer reviews we received, as well as the summarized feedback suggestions above. We have decided to:

1. change our primary key to be auto incremented instead of inserted
2. add INSERT functionality in our intersection tables
3. create a department_fk attribute in the employees table
4. add DELETE functionality in our intersection tables
5. create a search/filter option on each page of our website
6. implement the INSERT table for the task entity
7. add SELECT functionality to our intersection table
8. create a navigation bar that is at the top of all pages
9. add a table on the index page to show the database system

Upgrades to the Draft Version:

1. Changed all id's for each table to auto-increment.
2. We already have insert functionality.
3. We didn't find this foreign key necessary.
4. All of our tables have delete functionality.
5. We are currently working on implementing search/filter functionality.
6. We have removed the task entity and added it as an attribute in the projects table.
7. We are currently still working on select functionality.
8. We are currently working on a navigation bar for every page

Project Step 2: Reviews

Verbatim from Piazza:

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

Yes, it does.

Does the overview list specific facts?

I think so, yes. It might do well to add examples of positions though.

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

Yes, there are four entities, and yes, I think they represent single ideas pretty well.

Does the outline of entity details describe the purpose of each, list attribute data types 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, there is quick info on each.

Yes, it does have the attribute data types and constraints

Yes, it does list the relationships

Yes and yes

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

I think the relationships in the outline could have been more clear, but the relationships in the pictures were fine.

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

a. Yes

b. Yes

c. Attributes were not capitalized

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

Yes, although it could be improved by including a bit more detail.

Does the overview list specific facts?

The overview is a bit light on facts. It does list the number of employees and departments, but could be improved by including numerical estimates for the other entities. For example, how many positions exist at the company? How many projects?

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

Yes. There are more than 4 entities described here and each represents a single idea (with the exception of Projects, which is described in more detail below).

Does the outline of entity details describe the purpose of each, list attribute data types 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)?

The outline of entity details does indeed describe the purpose of each, except the description for Projects, "records which employees are assigned to which project", does not seem to be accurate. Projects has only 2 attributes: project_id and project_name. Tasks is the only entity that actually connects employees and projects (and therefore Tasks and Employees should also be connected in the schema).

Attribute datatypes and constraints look like they are listed appropriately for the most part. Don't forget to mention that ids are unique. All attributes in the database are described as not NULL, but maybe consider a couple where NULLs might be acceptable.

The outline has a section that details the division of work within the group, but does not mention the Tasks entity. Will Tasks not be implemented in the final version?

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

The crow's foot notation between Positions, Employees_Positions, and Employees seems to be backwards. Each position_id in Employees_Positions must be represented once and only once in Positions. Likewise, each employee_id in Employees_Positions must be represented once and only once in Employees. Each position_id in Positions may be represented many times in

Employee_Positions. Finally, each employee_id in Employees may be represented many times in Employee_Positions.

Additionally, the 2 M:M relationships between Departments/Employees and Projects/Employees ought to be represented in their own relationship tables.

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

There is some capitalization inconsistency between the outline, ERD, and schema (entities are capitalized in the outline but not in the ERD or schema).

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

Yes, explains what it will keep track of, however doesn't explain why that information should be tracked or what kind of problem it will solve.

Does the overview list specific facts?

Yes, lists 100 employees over 3 departments.

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

Yes, lists Departments, Projects, Employees, Salaries, Positions, and Tasks entities, with an Employees_positions intersection table. Each represents one core 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)?

Clearly indicates which tables will be implemented and by which member. Attributes are mostly well defined, however the attributes for Employees_positions are missing their type. Some attributes that are required for relationships may be missing. Also, the Salaries table doesn't seem to have any information about the actual salary of the employee, it just link some dates to an employee.

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

There are a few M:M relationships, however some attributes may be missing. How do you plan to associate a project with an employee, there is no attribute in the project for the employee_id of the owner. Also, how do you associate employee with being part of a department, you probably need another intersection table, but none is listed. The M:M relationship between employees and positions seems to be correctly formulated however. Most of the 1:M relationships are formulated mostly correctly in the outline, however the crow's foot notation in the Schema is backwards for some, and when explaining the relationships in the outlines, it would be helpful to explicitly list which entities the relationship is between, including correct capitalization of the entities.

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

In the outline, the entities and attributes follow a consistent naming convention, however there is some discrepancy when referring to these when describing relationships and in the ERD and schema.

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

Yes - this project does that

Does the overview list specific facts?

Yes. The overview provides specific numbers in their project.

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

Yes - and those entities are capitalized and well highlighted in the outline at the beginning.

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, each entity has attributes, datatypes, constraints, and relationships between the entities.

Yes, this group has included this - they're slightly hard to read though.

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

This group has 2 1:M relationships and are formatted correctly.

Yes! This group has 3 M:M relationships.

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 consistent naming schemes present.

Plural and capitalized entities present in this group's work.

Yes, this group consistently used caps!

Feedback by the Peer Reviewer:

Our draft submission received feedback from four reviewers, and it is very consistent throughout all reviews of what aspects need to be modified within our project outline. According to our first review by Liam Zimmerman, the feedback given are: "add examples of positions", "the relationships in the outline could have been more clear", and "attributes were not capitalized".

Moving forward to our second review by Casey Cheek, the feedback we received are:

(regarding our overview) "could be improved by including a bit more detail", "improved by including numerical estimated for the other entities", "does not mention the Tasks entity", "crow's foot notation between Positions, Employees_Positions, and Employees seems to be backwards. Each position_id in Employees_Positions must be represented once and only once in Positions. Likewise, each employee_id in Employees_Positions must be represented once and only once in Employees. Each position_id in Positions may be represented many times in

Employee_Positions. Finally, each employee_id in Employees may be represented many times in Employee_Positions. Additionally, the 2 M:M relationships between Departments/Employees and Projects/Employees ought to be represented in their own relationship tables." and "there is some capitalization inconsistency between the outline, ERD, and schema". Next, in our third review provided by Trevor Jones, he stated: "attributes for employees_positions are missing their type", "some attributes that are required for relationships may be missing", "salaries table doesn't seem to have any information about the actual salary of the employee", "helpful to explicitly list which entities the relationship is between, including correct capitalization of the entities", and (referring to capitalization consistency) "discrepancy when referring to these when

describing relationships in the ERD and schema”. Lastly, in our fourth peer review provided by Matthew Hawkins, he stated when referring to our entities, “they’re slightly hard to read”.

Actions Based on the Feedback:

Based upon the four peer reviews we received, as well as the summarized feedback suggestions above. We have decided to:

- provided examples of positions in our outline in order to add more beneficial information
- after meeting with a TA we established our relationships, ERD and Schema correctly
- added more consistency in regard to our capitalization and consistency throughout our outline, ERD, and schema, as there have been a few noticeable outliers
- provided more in depth detail in our overview regarding our database’s purpose
- implemented more numerical details throughout our entities
- ensure that there are no missing entities or relationships in our outline
- organize our document so that it is much easier to comprehend

Upgrades to the Draft Version:

Since our first outline, we have completely redesigned our project and ended up switching our topic all together. The items we were told to fix by our TA were including numerical facts to give an idea of the scope of the database, giving an idea of how entities are connected to each other with the use of primary/foreign keys, and making sure we have enough entities with an appropriate intersection table. We fixed these issues by completing rethinking our project idea and creating an outline with all of the correct components. We have also decided to remove our salaries table as it was not necessary for the scope of our project.

Project Outline:

Project Title:

Employee Management System

Team Members:

- Luke Bray
- Gretel Rajamoney

Database Overview:

Company Incorporated a company in the banking industry wants a database to keep track of their 100 employees and 5 different departments. Our database driven website will keep track of each employees: personal information, salaries, departments within the company, positions, and employees within those departments. It will also look at each project and which tasks are required in that project.

Database Outline:

employees: records basic information about each employee such as their employee id, first/last name, and birth date.

- employee_id, int, not NULL, Auto-increment, PK
- first_name, varchar, not NULL
- last_name, varchar, not NULL
- birth_date, date, not NULL
- Relationship: M:M, this relationship serves all of the other entities. The employee_id acts as a FK in the salaries and positions entities.

projects: records which employees are assigned to which project.

- project_id, int, not NULL, Auto-increment PK
- project_name, varchar, not NULL
- Relationship: M:M, one or many different employees can be working on zero or many different projects.

positions: records the position of each employee

- position_id, int, not NULL, Auto-increment, PK
- position_name, varchar, not NULL (ex. accountant, bank teller, treasurer)
- position_rank, varchar, not NULL
- Relationship: 1:M, one or more employees can have the same type of position.

employees_positions: intersection table between employees and positions table.

- employee_id, not NULL, FK
- positions_id, not NULL, FK
- Relationship: 1:M, this intersection table connects the tables of employees and positions giving them a M:M relationship.

departments: records each department of the company with an id and name, the departments include: retail banking, corporate/commercial banking, global banking, private banking, and investment banking.

- department_id, int, not NULL, Auto-increment, PK
- department_name, varchar, not NULL
- Relationship: M:M, one or many departments can have one or many employees and employees can be in more than one department.

tasks: records which tasks are in the project and which employees are assigned to which tasks.

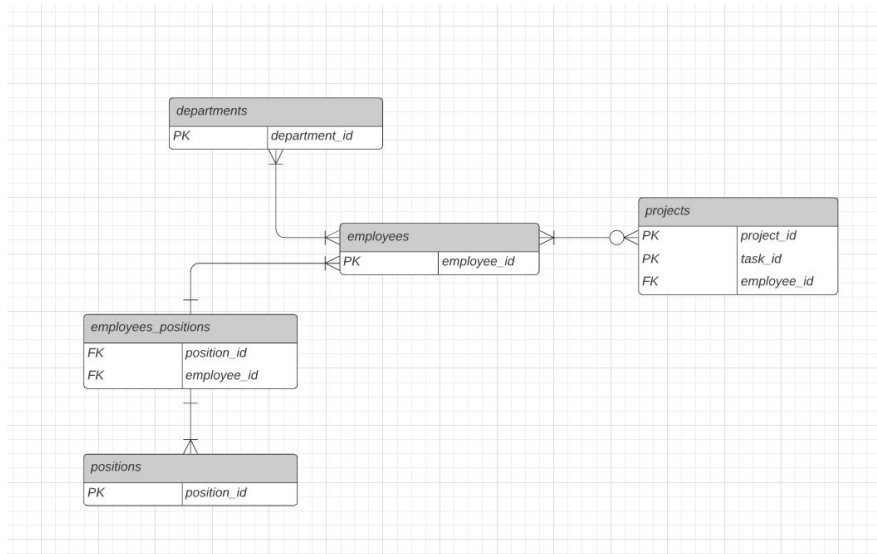
- task_id, int, not NULL, Auto-increment, PK
- project_id, int, not NULL, FK
- employee_id, int, not NULL, FK
- assigned_task, varchar, not NULL
- assigned_to, varchar, not NULL
- Relationship: 1:M, one project can have one or many tasks.

Division of Work

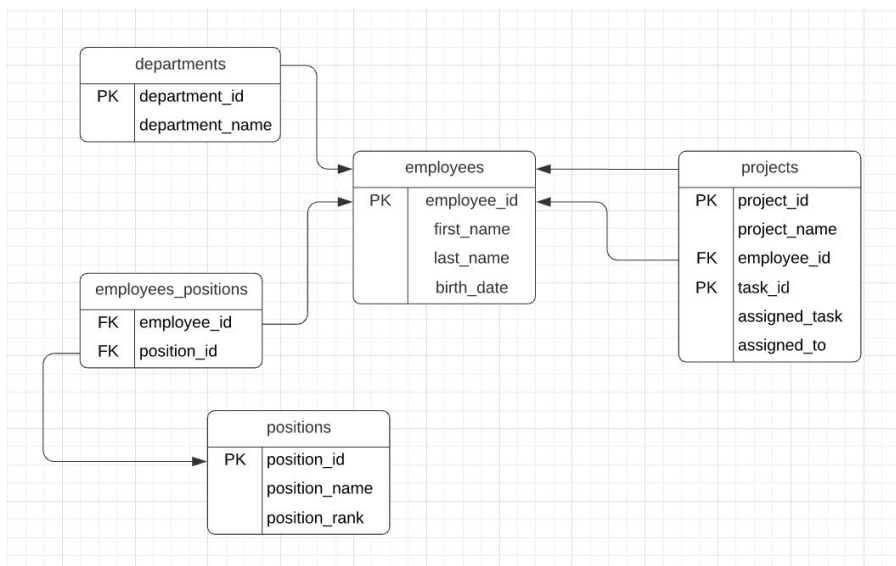
Gretel: Will be in charge of the department, employees, and salaries entities.

Luke: Will be in charge of the M:M relationships, this will involve the employees projects and the employees positions.

ERD



Schema



CS 340 TEAM EVALUATION FORM STEP 2

GROUP NUMBER	50	
NAME OF GROUP TEAM MEMBERS	Luke Bray & Gretel Rajamoney	
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
<p>HOW PREPARED WAS YOUR TEAM?</p> <p>Research, reading, and assignment complete</p>	4	Although we ran into issues in our first outline, we quickly bounced back and created an outline much better.
<p>HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM ?</p> <p>Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.</p>	4	We have strong communication with each other and complete all steps well in advance as well as respond to each other in a timely manner. We always check in with each other after each single grade release as well.
<p>DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY ?</p> <p>Contributed best academic ability</p>	4	Yes, we both carried our weight and collaborated in order to ensure we were submitting our best work and splitting the work equally.
<p>DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT ?</p> <p>Were both team members both positive and productive?</p>	4	Yes, we both have worked together throughout the term as a team both positively as well as productively.

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

We both have been working well together, no changes are necessary in our team contract as of right now. Our goal is to start the workload early and submit our best work for each and every step.

CS 340 TEAM EVALUATION FORM STEP 3

GROUP NUMBER	50	
NAME OF GROUP TEAM MEMBERS	Luke Bray & Gretel Rajamoney	
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
HOW PREPARED WAS YOUR TEAM? Research, reading, and assignment complete	4	We have been pretty prepared regarding the HTML front-end part of this step because we both have strong web development experience.
HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM ? Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	As stated in the prior team evaluation for step 2, we are still maintaining strong communication with each other and working together well.
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY ? Contributed best academic ability	4	Yes, we always meet up virtually to work on the project together and split each task equally.
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT ? Were both team members both positive and productive?	4	Throughout the term, the two of us have been working extremely efficiently and productively, and the environment is very positive.

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

We are still performing well together and collaborating very successfully, there are not many changes that are necessary, but we both definitely need to learn more regarding the back-end details for our website, but we both have no concerns that need to be addressed as we both adapt very quickly.