Project Step 3
Behrad Noorani
Matthew Gragg

https://web.engr.oregonstate.edu/~nooranib/restaurantcs340/

Feedback Peer Reviews Draft Step 3:

Derrick Macaranas:

Does the UI utilize a SELECT for every table in the schema?

Yes, the UI utilizes a SELECT for every table in their schema. In your SELECT section in your DML your comments should be updated to correctly describe the operations. For instance,

"-- Select customer by name" is written in places where it should be "--Select employee by name".

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

Yes, there are multiple SELECT operations that utilize a search/filter to dynamically populate a property list.

Does the UI implement an INSERT for every table in the schema?

It appears the UI has an INSERT operation for Customers, Orders, and Employees. However, the UI for INSERT OrderItems and INSERT Items have not been implemented in the UI. An admin does not have room in the UI to manually enter in a new suborder (OrderItems) to a specific Order or to add a new Item the restaurant wants to sell.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

Yes, I was able to find this in their DDL file. However, in their DML file their INSERT operations were missing.

What should be added to their Final submission is adding how their INSERT INTO operations would be incorporated to their frontend UI. For example, In their Orders page, they have two dropdowns for Assign Employee and Assign Customer.

What needs to be shown in the DML is similar to what was on the Exploration - Database Application Design

-- get all Planet IDs and Names to populate the Homeworld dropdown SELECT planet_id, name FROM bsg_planets -- add a new character INSERT INTO bsg_people (fname, lname, homeworld, age) VALUES (:fnameInput, :lnameInput, :homeworld_id_from_dropdown_Input, :ageInput)

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

While there are many DELETE operations in the DML, the frontend does not show it has delete buttons to execute the DELETE operations to the backend.

Is there at least one UPDATE for any one entity?

Yes, there are UPDATE operations for multiple entities.

Is at least one relationship NULLable?

I do not think so. In their Orders page, their dropdown for Assign Customers is pre-set to Customers 1 through 4. In their ERD, an order does not require it to be associated with a customer, this specific field should have a NULL option so the form can submit a NULL value.

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

I would recommend a navbar to be able to navigate to different pages of your database. Currently I have to use the browser's back button to get back to the main page.

Reiterating adding INSERT operations for OrderItems and Items.

The buttons on the homepage can be more descriptive so the admin understands the purpose of the page. "Customer" --> "Add a Customer".

I prefer seeing data at-a-glance without having to search beforehand. So if I click on Customers, I can see all the Customers, then if I want to add a customer, I can do so on the same page.

Hang Liu:

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.

The UI does not utilize any SELECT for tables. They should show the list of their SELECT command. Besides, they are lacking a separate tab for the Items table.

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

They have a tab called Search in the website, which enables search for a certain customer, employee, order and item.

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.

The UI implement an INSERT for all tables except Item 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).

They do not have INSERT command in their DM file.

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 are DELETE commands in their DM file. However, they do not have a special character to mark that values are receiving from frontend.

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?

They have Update commands for Customers and Employees tables, however, like above, they do not mark receiving values from frontend.

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.

A customer may not necessarily have an order.

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.

They should first of all show a list for every table. Besides, add a tab for Items table. Thirdly, they should implement Update and Delete button. At last, it's better to show the tabs all the time, so that user does not need to click back button to navigate back to the home page.

Kaelan Trowbridge:

Hey team! I like the restaurant idea, I actually worked at a restaurant with similar employee numbers and probably somewhere similar in sales per day so it was nice to read about something I have some experience with.

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.

The DM file has a select for all tables, however we cannot see the tables being selected within the UI. Additionally the selects all contain filters, so I don't think any of them will be used to

display the tables in their entirety. Fortunately these are really easy to add with a select * from table_name!

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

Yes, all selects have a where clause attached to them. I'm assuming these are primarily for your search/prefilling in boxes.

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.

All of the tables that the UI includes (missing OrderItems and Items) have the ability to insert on the UI. There is however no insert statements in the DM file, I do see an "Inserts are already in DDL.sql(?)", to my understanding the insert statements that will be added into the DM file will be to add the values that users input into the UI, contrasted with the values that we inserted as starter values in the DDL.sql file.

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).

Insert statements are absent, please read my above review. The UI does include the FK attributes when inserting, so this is good, just needs corresponding sql (:

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.

Delete functionality is included in the SQL, but corresponding options to actually delete the values in the UI is absent. The deletes from orders and items will remove items from a M:M relationship, so be sure to take care when doing so here.

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?

No updating is possible on the UI, but there is a lot of example update sql statements in your DM.sql file. There are no update statements for general variables such as "UPDATE Customers SET phone = :userPhoneInput," these will need to be added for update functionality in the next step.

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 diagram states that customerID in Orders is optional, so this can be nullable, or the relationship diagram should get rid of the optional (the circle next to the three prongs) for the 1:M relationship here.

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.

The best thing to do would be to add tables with sample data to your UI, as well as finish adding access to the final entities on your main page of the website. I think it would also be helpful to have access to the homepage or other entities while inside of each entity, currently we have to use the backarrow to navigate between entities.

like how the blue bubbles to click into each entity, definitely keep a similar style moving on !

If any of my comments don't make sense don't hesitate to reply and I'll try to get back to you and clarify!

Adam Hampel:

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 display them.

The UI utilizes a SELECT for every table in the schema. Curiously, your SELECTs in your sql file won't display the tables in their entirety. I don't know why you wouldn't just want to see the whole thing using SELECT *.

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

Yes, most of the SELECTs have the ability to filter with their WHERE clauses.

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.

While it does appear that the UI has an insert for every table in the schema, it isn't made very clear to the user that this is what's being done. Looking at the SQL, there doesn't appear to be any INSERT statements.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes, the UI includes the corresponding FK attributes, but as I said before, there is no SQL that reflects this functionality.

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 are no DELETE capabilities shown in the FrontEnd UI. There are, however, DELETE capabilities shown in the SQL. Be cautious when deleting from orders AS WELL AS items, because this will remove certain items from the relationship.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

The UI provides no UPDATE Functionality, but the SQL does indeed provide this functionality for at least one 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.

customerID in the table Orders appears to be optional & therefore nullable. This, however, is only according to the symbol provided on the ERD. If this was not intentional, then you should change your symbols on the ERD to reflect what your schema dictates.

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.

Show the data. It's nice to be able to just see the inserts that I'm doing, the updates that I make, as well as any deletions in real time. If I don't have a table to see the data, then I'm constantly wondering if the DB actually processed my queries.

Actions based on the feedback and Updrages to Draft Step 3:

- Added inserts to DML queries.
- Update styling for better clarity.
- Added all pages for each table in the schema.
- Implemented changes summarized above as an overall improvement of the clartiy of the webpage.
- Added id tags and internal organization so that added CRUD to the database will be easier.
- Thanks for the feedback!

Feedback by the peer reviewer Step 1:

Audrey Minch:

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

yes, the overview describes that it will store data of customers at a restaurant and their orders, email, phone, and wage in order to snapshot day to day interactions with customers.

Does the overview list specific facts?

yes it describes the number of employees and how many orders are taken at the restaurant.

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

Yes, the entities are customers, items, employees, orders, and orderitems and each represents a single idea to be stored as a list.

Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities?

Yes, the outline describes the entity details and purpose of each.

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

1:M relationships are correctly formulated and there is at least one M:M relationship. The ERD does present a logical view of the database.

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, entity names are capitalized.

Jose Reino Reloj:

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

Yes. The database will track day to day interactions a restaurant has with customers which in turn will probably help restaurant operations.

Does the overview list specific facts?

Yes. The restaurant encounters 700 orders on a daily basis and has 40 employees.

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

There are 5 entities in their database. Customers, Employees, orders, order items, and items. Order items and items seem to be redundant? Neither of which seem to be described as a transaction table and could possibly be combined.

Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities?

Not every entity seems to have a description of its purpose. It would be helpful to know so someone can tell the difference between the Order, Order items, and items entities.

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

The 1:M relationships seem to be correct. There is a 1:M relationship between customers and orders as well as a 1:M relationship between orders and employees. There seems to be a M:M relationship between order items and items. The ERD is getting to the point of having a logical view of the database, I am just trying to understand the purpose of certain entities due to the lack of description.

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

Yes. All entities are plural. All entities names are capitalized except for employees. All attributes are singular. My only issue would be to have a more unique and specific name scheme for some entities as one could consider some of the entities to be very similar without the proper description.

Minh Thao Le:

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

Yes. The proposal overview did a good job of presenting the problem and how the database comes in to solve that problem, recording day-to-day orders from customers. The restaurant can make use of acquired data to provide better customer service.

Does the overview list specific facts?

Yes. Information about the restaurant is clearly presented: 40 employees, 700 orders per day.

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

There are 5 entities: customers, orders, orderitems, items, employees. To my understanding, each of the entities represents a single idea. Customers, orders and employees are self-explanatory, orderitems record all items from each order per order and items record information about the item (dish). I also saw other restaurants display halal food as well, just in case you would like to add it as a boolean data type.

Does the outline of entity details describe the purpose of each, list attribute data types and constraints, and describe relationships between entities?

Definitions for employees, orders, orderItems are not provided. This may cause confusion for understanding in implementation.

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

1:M: Customers and Orders is presented correctly as 1 customer can have many orders.

M:M: Orders and OrderItems is presented correctly as one item/dish can be ordered from many order, and one orders can have many items(orderitems)

ERD presents a logical view of the database as well as the relationship between tables.

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

"employees" is not capitalized in the diagram otherwise entities are plural, all attributes are singular

Jesse Curran:

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

Yes. Websites with DB backend will help track store level customer-restaurant interaction into usable data. The overview was a good overlay to presenting the problem and how the database and website will add value. I liked it.

Does the overview list specific facts?

Yes. Name of restaurant, daily order amount, number of employees. Specific data to be extrapolated. Facts seem to be present and in order. Another bit of feedback, I do like the detail to which attributes and their scopes are defined. Varchar(20) and having some attributes not incorporating NOT NULL shows me some thought was put into the implementation of this website and database.

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

Yes. Customers, items, employees, orders, order items. All of these represent a single idea that could be stored as a list.

Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities?

Yes and no. All attribute data types and constraints appear clear. However, Items does not show a relationship in entity details but shows so in ERD model. Nothing huge though.

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

Yes. 1:many relationships appear to be correct, and there is at least one M:M. The ERD also appears logical. Yes, looks like all relationships are accurate at first glance.

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 in naming, entities, and capitalization. Nothing seems to be an issue here.

Actions based on the feedback

Based on the feedback we have received:

- Update order items attributes so that its purpose is more clear.
- Update all the spelling and naming consistency.
- Make the descriptions of the entities more different to ease understanding.
- Describe entities in more detail.
- Add food type/allergies/religion field to items.
- Changed the M:M relationship between order items and items to be between orders and items instead and orderitems is a intersection table
- Changed relationship between employees and orders to 1:1

Upgrades to the Draft version

The peer reviews encapsulated the core of the changes we want to make so, there are no changes we plan on making that cannot be traced to some feedback we received. Additional changes were made after the feedback on the rubric.

Feedback by the peer reviewer Step 2:

Matt Trimner:

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

- I don't think I found the Schema. The file I had only showed the outline and ERD.

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

- No, the outline has some attributes capitalized and some not. The ones capitalized on the outline are not capitalized on the ERD. This might be on purpose though.

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

- I only found the ERD and Outline. The ERD might be meant to double as the Schema though. In that case, yes.

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

- Yes. Orderitems was properly formatted. The PK was a combination of orderID and itemID which might make it a little complicated to implement. But I don't see a problem with it at all!

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

- I don't think there are any problems with the data.

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!)

- I got errors when I tried to run it locally. However, this is probably on my end as visually, everything seems fine.

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

- Yes, the data types make sense.

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

- Yes, but there weren't any CASCADE operations. But that probably isn't needed.

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

- Yes, all tables in the SQL are represented in the ERD.

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

- Yes, the data is there, but the formatting seems off in the SQL. Some stuff like the phone, is in quotes in some instances and not in others. It also doesn't match the PDF. Not sure if that was necessary though.

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

- It looks to be structured fine, but there aren't any comments. However, I didn't really need comments to understand what was going on.

Derrick Macaranas

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

- It appears that their ERD is formatted like a schema but doesn't contain all of the information that they put in their database outline. For your final draft for step 2, you can refer to this Ed post to write your schema: https://edstem.org/us/courses/32532/discussion/2419121

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?

- In their database outline some attributes are all lowercase, some follow camelCase, and some are capitalized. Since a good number of your attributes are in camelCase, I suggest that you stay consistent with this naming convention. For example change firstname to firstName and lastname to lastName.

In their ERD/Schema, in the Items table, they have customerID as a foreign key, but their Database Outline does not list it as foreign key.

- In their ERD/Schema, Employees (title of entity) should be capitalized.

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

- Based on what they put in their ERD, their diagram is clear and readable with relationships not crossed. However, the only part I am concerned about is the point of having itemID as a foreign key in the Customers table when there is no relationship between the Customers table and the Items table? If there is a link between these two tables, there should be a line that shows their relationship.

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

- Yes, foreign keys orderID and itemID, refer to the primary keys of the "Orders" and "Items" tables so it allows this table to facilitate an M:N relationship between orders and items. Each order-item combination is unique since they are using a composite primary key, the combination of orderID and itemID.

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

- Since you included customerID in your Items table as a foreign key, there could be a partial dependency between customerID and restriction. When a non-primary key

attribute like restriction depends only on the customerID (or the customer's restrictions), but not the entirely on itemID, then you might be facing a 2nd normal form issue. I suggest that you create a separate restrictions table with a foreign key referencing this table within the Items table. So if you need to add or modify your dietary restrictions in the future, you do not need to change the structure of your Items table.

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!)

- Line 42 References should be Customers instead of Customer.
- Line 94, `orderID`int NOT NULL. I believe you need a space before int.
  `orderID` int NOT NULL,

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

- The data types in the SQL file appear to be appropriate. However, one inconsistency between the ERD/database outline and the SQL file is in the Employees table. In the ERD & database outline, they have 'pay' as an int and float in their SQL. It appears the intention should be a float in this case so it should be changed in their ERD/outline too.

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

- In the SQL, I believe their primary and foreign keys are correctly defined.
- However, according to the project requirement 'You should be able to DELETE a record from an M:M relationship without creating a data anomaly in the related tables.' This requirement is currently not being met. You can add the following keywords ON DELETE CASCADE to the ends of lines 86, 88, 102, and 104 to handle this.

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

- Yes, the relationship tables are consistent in the SQL as with their ERD/Schema.

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

- Yes, the example data from the PDF is inserted into the SQL file. However, one thing to note here is when inserting dates, they should be written as strings in the format 'YYYY-MM-DD'. The Orders data and total cost value do not add up with what is being ordered, but I am not sure if the data must be consistent for this assignment.

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

- Yes, the SQL file is clear in its structure. It might help to add some comment headers between distinct portions of your code.

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

- You don't have a schema. Only an ERD. Derrick Macaranas's comment above already gave a link on how they differentiate from each other.Besides, I would like to point out that in the DB Outline section, you wrote in Employees table that a 1:1 relationship exists between employees and orders, however, in your ERD, you have a 1:M, which makes sense. I guess you have a typo in your DB Outline.

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?

- In the Database Outline, the attribute names have different formats, e.g. firstname and lastname in Customers table, but Name and Price in Items table. Each word in an attribute name should be differentiated. So the correct form should be firstName and lastName in Customers table, and name and price in Items table.

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

- You don't have a schema.

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

- Yes. You have an OrderItem intersection table. Two FKs are orderID and itemID. OrderItem tables have a M:1 relationship with Orders table and Items table, which facilitates a M:N relationship between Orders table and Items table.

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

- In your ERD and SQL, you have customerID in Items table, which have non-normalized issues. However, you don't have it in your DB outline, so maybe it is a typo. Other than this, I don't find non-normalized issues.

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!)

- 1, You have a commit statement at last, but you don't have a start transaction statement at the beginning. You should wrap your SQL in one transaction to prevent partial execution.
- 2, the phone in the Customers table is varchar(20), however, you forgot the quotation marks at the insert statements.
- 3, You also forgot the quotation marks for the date column in the Orders table.

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

- Float is not a valid data type. Your price and total columns should be decimal. Please refer to the link below for valid data types for mysql. https://canvas.oregonstate.edu/courses/1901738/pages/sql-datatypes-and-examples?module_item_id=22733676

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

- You declare every PK as Unique, which I find unnecessary. PK is already unique, and Unique declaration is used only to declare unique keys.
- You don't have CASCADE operations. Without such, you may have a problem if you delete a row in one table that works as a FK in another table.

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

- In your ERD, you have itemID as FK in Customers table, but you didn't declare it a FK in your SQL. ItemID should not exist as a column in the Customer table anyway.

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

- Yes. But you insert only one row each time. You can actually insert all values using one insert statement.

Cammille Mitchell

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

- It is well structured. It does not have comments, but I don't find comments necessary though.
- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - There is no schema. However, an ERD is present.
    - The Items table includes a FK reference using customerID which is the PK of the Customers table. Given that the Items table isn't shown connecting to the Customers table in the ERD and instead is connected to the OrderItems I'm not sure if this is intentional. If it is intentional you may need to redefine your relationships because the outline states Customers has a 1:M relationship with *Orders*.
- 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, yes. However I'd recommend clearly labeling the FKs in the outline like you have for the PKs. For example, in the Customers table section of the outline, it's not immediately apparent that itemID is the FK here:
        - *"itemID: int //customer's most ordered item*
          1:M relationship between customers and orders,"
    - In the employees table there are a couple mistakes:
        - 1. The table name should be capitalized.
        - 2. Your primary key has a duplicative 'PK' designation.
    - Inconsistency observed comparing the ERD which shows payrate has a data type INT, while the SQL shows float.
    - There are a few examples of inconsistent attribute naming conventions in the outline. Some attributes are capitalized while others are not. For example, Employees table has attributes: employeeID, firstname, lastname, phone, email, **Address, Role, and Pay**
    - I noticed a small redundancy in the overview section: "The database's purpose is to store data of each customer **and their orders** **and their orders** and to..." Recommend trimming that part.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - There is no schema.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes, the intersection table (OrderItems) has 2 FKs and 1 PK. It facilitates the M:N relationship between Orders and Items.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - I think there could be non-normalized issues stemming from the use of customerID as a FK in Items and Orders
- 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!)
    - no autocommit=0 statement at the start of the file
    - missing quotes around phone numbers in the insert statements
    - i think dates should also have quotes
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - need to use int or decimal data types instead of float
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - No CASCADE operations declared.
    - PKs are correctly defined.
    - itemID is declared as a FK in the ERD and referenced in the outline, however it is not declared as a FK in the SQL file.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Yes they are present, but there are inconsistencies as mentioned previously

- In the SQL, is all example data shown in the PDF INSERTED?
    - Yes
- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
    - It appears well structured, although it does not have any comments.

Actions based on the feedback:

Based on the feedback we have received:

- Removed ItemID from customers
- Removed CustomerID from items in the ERD
- Added schema and Updated ERD
- Added FKs to outline
- Updated names to camel case
- Changed sample data to not include the auto increment IDs
- Changed DDL.sql to match ERD and schema and added CASCADE operations
-

Overview

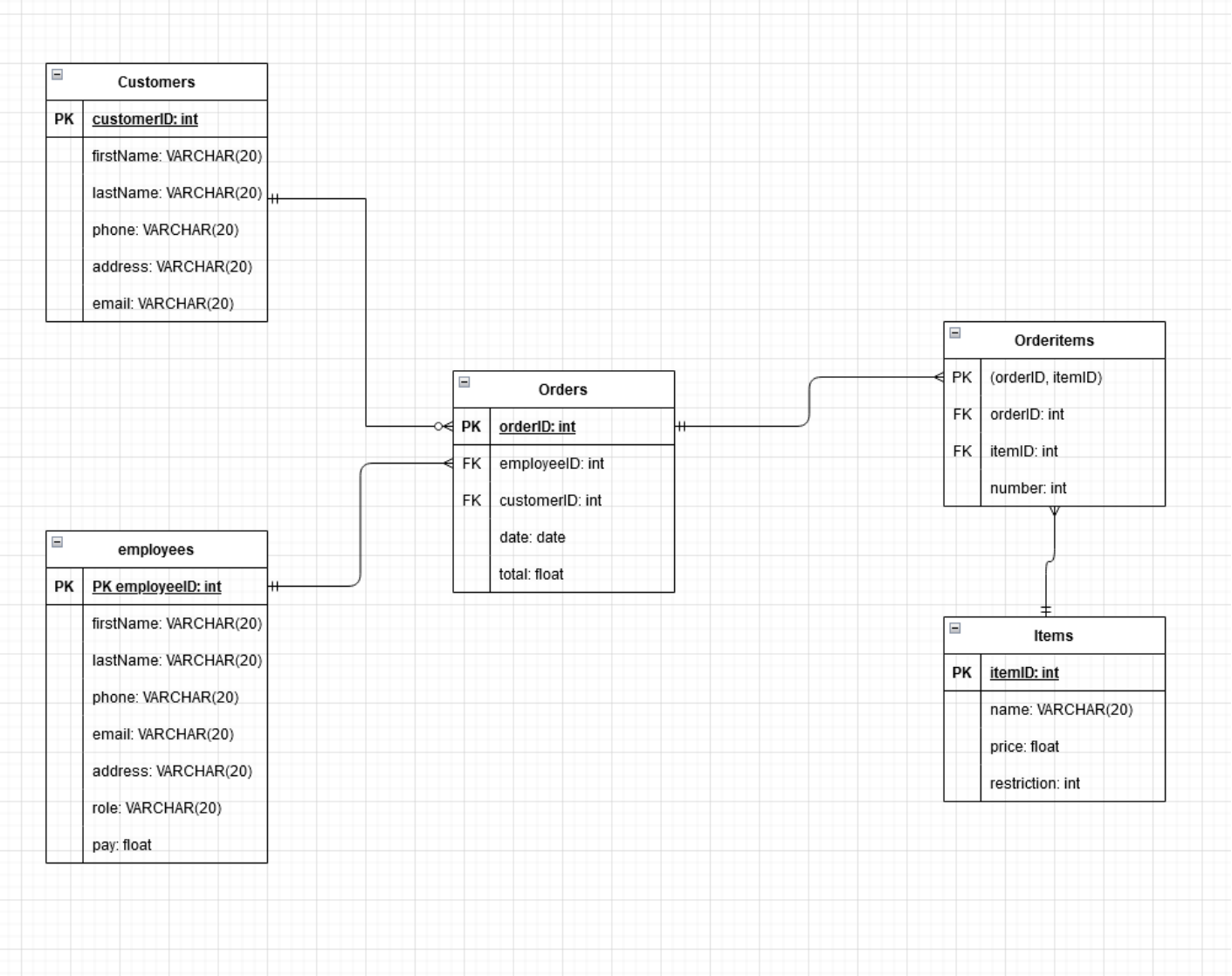
Radish is a popular local restaurant, it takes 700 orders on a daily basis. The restaurant features various home style dishes and drinks and also has a bakery for desserts and employs 40 hard working employees. The database's purpose is to store data of each customer and their orders and their orders and to store the data of employees such as email, phone and wage. The basis for this system will allow for customer interaction to be traced at the store level and offer restaurants a way to snapshot the day-day interactions with customers in a dataset. This system will also facilitate easy abstraction of the menu and allow for easy tracing of orders and fulfillment.

Database Outline:

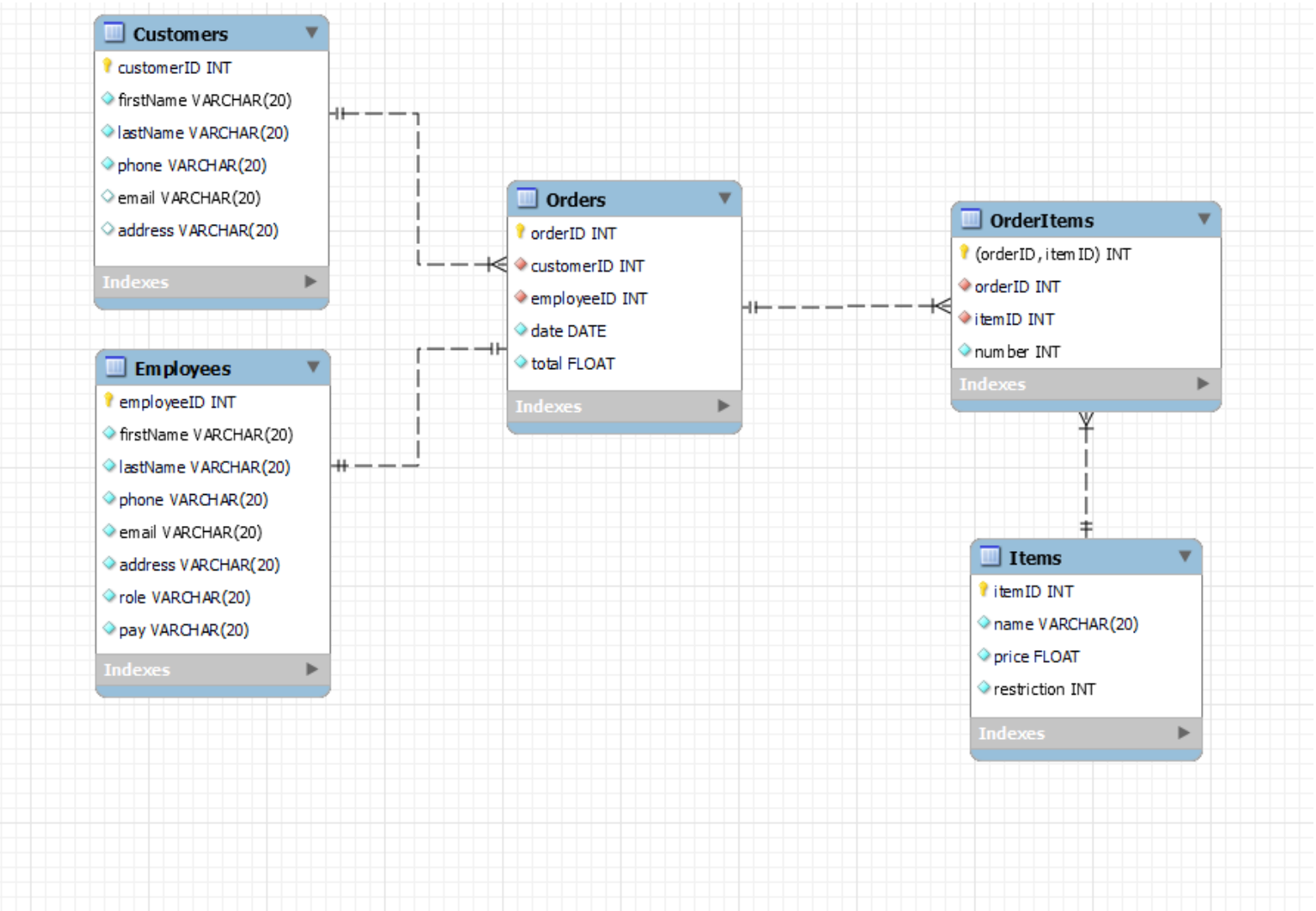
- ★ Customers: Customer details that dine in or order online
  - customerID: int, auto_increment, unique, not NULL, PK //customer system ID
  - firstName: VARCHAR(20), not NULL //first name of customer
  - lastName: VARCHAR(20), not NULL //last name of customer
  - phone: VARCHAR(20), not NULL //phone number of customer
  - email: VARCHAR(20) //email of customer
  - address: VARCHAR(20) //address of customer
    - 1:M relationship between customers and orders,
- ★ Items: items on the menu and prices
  - itemID: int, auto_increment, unique, not NULL, PK //menu item ID
  - name: VARCHAR(20), not NULL //name of item on menu
  - price: float, not NULL //item price
  - restriction: int, not NULL //dietary restrictions as ints (0 = none, 1 = vegi, 2= vegan, 3 = halal… etc)
- ★ Employees:
  - employeeID: int, auto_increment, unique, not NULL, PK//employee system ID
  - firstName: VARCHAR(20), not NULL //firstname
  - lastName: VARCHAR(20), not NULL //lastname
  - phone: VARCHAR(20), not NULL //phone#
  - email: VARCHAR(20), not NULL //email
  - address: VARCHAR(45), not NULL //address line
  - role: VARCHAR(20), not NULL //job title/role
  - pay: int, not NULL //wage (if salaried will be salary divided by contract term)
    - 1:1 relationship between employees and orders,
- ★ Orders:
  - orderID: int, auto_increment, unique, not NULL, PK//orders system ID
  - customerID: int, not NULL //customer attributed to order FK
  - employeeID: int, not NULL //employee attributed to order FK
  - date: date, not NULL // date and time of the order
  - total: float, not NULL // total price of the order

        - M:M relationship between orders and items, CustomerID and employeeID exist as FK in orders
- ★ OrderItems:
    - orderID: int, not NULL// FK from order, orders system ID, FK
    - itemID: int, not NULL // FK from items, menu item ID, FK
    - number: int, not NULL // Number of an item in order
    - (orderID, itemID) PK
        - Intersection table for orders and items

Entity-Relationship Diagram:



Schema:


Customers
customerID INT
firstName VARCHAR(20)
lastName VARCHAR(20)
phone VARCHAR(20)
email VARCHAR(20)
address VARCHAR(20)
Indexes
Orders
orderID INT
customerID INT
employeeID INT
date DATE
total FLOAT
Indexes
OrderItems
(orderID, itemID) INT
orderID INT
itemID INT
number INT
Indexes
Employees
employeeID INT
firstName VARCHAR(20)
lastName VARCHAR(20)
phone VARCHAR(20)
email VARCHAR(20)
address VARCHAR(20)
role VARCHAR(20)
pay VARCHAR(20)
Indexes
Items
itemID INT
name VARCHAR(20)
price FLOAT
restriction INT
Indexes

Example Data:
Customers:
(John, Wick, 5489879874, 3 Baba Yaga Dr: houston: TX, Johnwick12@gmail.com)
(Luke, Skywalker, 4879842321, 6 Sabre St: suite 12: Houston: TX, mydadisntvader@yahoo.com)
(Peter, Parker, 5568875235, 56 Spider Dr: houston: TX, fakewebs@gmail.com)

Employees:
(Bob Belcher, 5477779874, bobby24@gmail.com, 5 Bobs Dr: houston: TX,, Manager, 30)
(Linda Belcher, 5477779875, linda24@gmail.com , 5 Bobs Dr: houston: TX, Waiter, 25)
(Gene Belcher, 5477779876, extracharacter@gmail.com , 5 Bobs Dr: houston: TX, Waiter, 15)

Orders:
(2, 1, 2023-2-7, 130)
(2, 1, 2023-2-7, 15.99)
(3, 3, 2023-2-8, 15.77)

Orderitems
((1,1), 1, 1, 20)
((2,2), 2, 2, 1)
((3,2), 3, 2, 1)

Items:
(Fries, 6.5, 0)
(Pepperoni Pizza, 15.99, 2)
(Cheese Pizza, 12.99, 0)