Group Members of Planet Express (Group: 65):

Title: Planet Express Database
Benjamin Rifleman - co-lead, submission proofreader, researcher
Ryan Canete - co-lead, researcher, debugger

Site:
http://flip2.engr.oregonstate.edu:1533/index

Project Step 4 Draft Feedback:
Peer Feedback:

- Add required constraints for inputs to avoid blank records
- Modify form so that a planet can be updated to null.
- Make sure inputs follow outlines for required fields. This can be done by adding required keywords in the forms.
- Data validation for email(@ and . required) and planet distance(no negative values)
- Potential changes/ ease of use suggestions:
    - Larger font
    - Update customer interface more centered/ apparent
- Change the column names on pages to be in plain english, but is_active in shipment types if okay with 0 and 1 since it adds to the theme.

Upgrades/Updates:

- Added constraints and required to html to avoid blank entries
- Added inset functionality
    - Customers page
    - Packages page
    - Shipment invoices page
    - Shipment types page
- Customers has a nullable relationship with planets
- Customers now able to be updated
- Added drop down menu queries to DML
- Updated to an innerjoin to allow null values to show for planets when a customer does not have a planet id
- Added data validation for various inputs
- Added error handling in the event of attempting to remove a planet with an active customer (custom error page)

Jesseline Velazquez
4 days ago

1

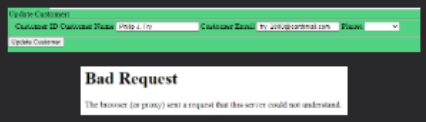
Hi Group 65 👋

talented, brilliant, incredible, amazing, show stopping, spectacular, never the same, k-totally unique, completely not ever been done before, unafraid to reference or not reference

- Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?
  - **Create**: Planets page - works. No issues adding a new Planet "Pandora."
    - I would add required constraints here because I was also able to add a blank record.

11	Pandora	Another planet for humans to colonize	260.0
12			0.0

  - **Read**: All pages - works as expected. No issues viewing records.
  - **Update**: Customers page - works. I was able to update the Philip Fry record.
    - It looks like you can't submit an update without the planet though, so I would update the form so that it's a required field and/or have it default to the value already in there. Right now its default value is NULL.



Bad Request

The browser (or proxy) sent a request that this server could not understand.

    - But I was able to edit the record once I picked something from the drop down menu. You can update the customer to not have a name and email though which contradicts the outline.

customer_id	customer_name	customer_email	home
5		fry_2000@earthmail.com	Earth

customer_id	customer_name	customer_email	home
5	Philip J. Fry		Earth

  - **Delete**: Shipment invoices page - I was able to delete the Zapp Brannigan invoice.
    - I didn't get a message to confirm deletion but it's not a requirement so no biggie.
- Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.
  - Yes, the UI is clear in terms of navigating complete the CRUD functions.
- What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit.
  - I would add the required keyword in your forms and update the types so the front end requests matches the NULL/NOT NULL and type constraints in your outline. It would clear up some of the oddities mentioned above.

T

Taylor Reed

4 days ago

Hi Benjamin & Ryan, nice work on your project so far. I like how you have all the pages working for read so we can view your full website!

- **Do the implemented CRUD steps function as the team expects?**
  Yes, I was able to create a planet, edit a customer, and delete a shipment invoice (you're running low on those by the way!). Read is functioning on all pages.
- **Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.**
  Yes, the UI is easy to navigate. I like how when you edit a customer their name and email show up in the update fields, it would be awesome to see the current planet selected as well.
- **What suggestions do you have for the team in any areas where they are blocked or having difficulty?**
  The team stated they don't have any blocks currently. My only other suggestions are to add data validation for email addresses (currently you submit an email with no @ or .) and planet distance from hq (currently you can enter negative values).

Comment •••

Add comment

H

Harinder Garcha

2 days ago

Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?

- **As described by the group Create->Planets Page, Read->All pages, Update->Customers Page, Delete-> Shipment invoices page.**
- **Clean work. However, restricting the input fields with constraints is needed.**

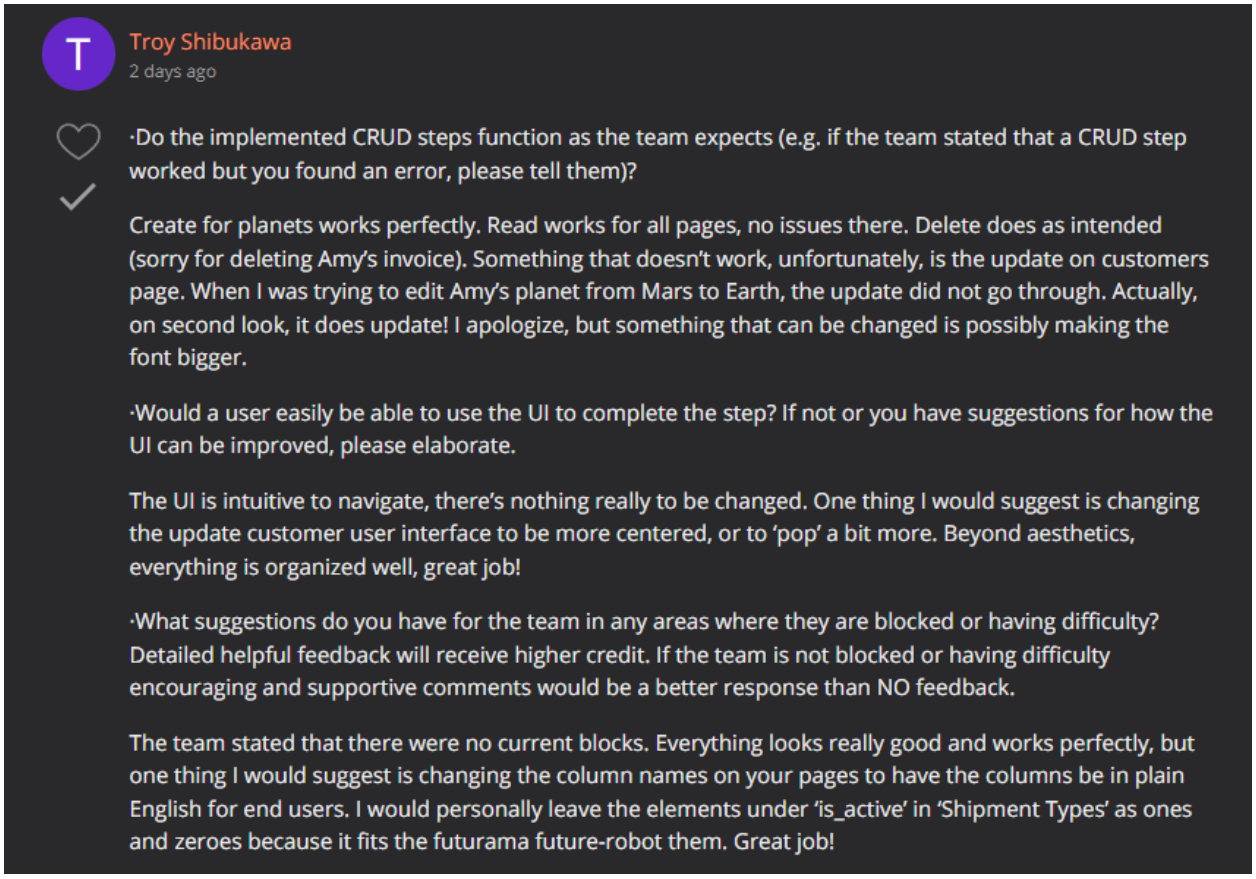
Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.

- **Very impressive here. It was clear and easy to navigate. Great Work.**

What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit.

- **Input fields are auto filled whenever an Update request arrives. It will help matching the database original constraints like NOT NULL etc.**
- **While passing the data from cursor to database. You can validate the data there too and based on the result, pop an error/success on front end. Or you can simply do it on the front end as well.**
- **Very impressive code. Great Work.**

Comment •••



Project 3 Final Draft Feedback:

Peer Feedback:

- Create solution so users don't have to input primary and foreign keys
    - Will become a dynamically populated dropdown menu and button to populate edit forms.
- DML.sql file needs to have drop down queries
- ERM to be updated to show nullable/optional relationship

Upgrades/Updates:

- Formatted static html pages into dynamically loaded pages with CRUD operations for different entities
    - Create: Planets page
    - Read: All pages will be populated with data from the database
    - Update: Customers Page
    - Delete: Shipment Invoices page
- Created a backend in Flask which will handle the routes and work with the front end to make working pages
- Dynamically populated dropdown menu in the Customers page
- Still need to add dropdown menu queries to DML

4 Answers

Travis Dulyeaparker
2 weeks ago

Looks interesting and the futurma references were fun. I know it is the first draft and its not part of the question but I would not leave the blue text on the off blue table header background color in the later versions its rather hard to read.

Nice work overall.

- *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 every table has a select in the UI that displays the values held within the table and correctly shows their attributes. They also make good use of FK calls to populate the invoices table with useful information such as the sender's name instead of their id.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
    - I do not see a select which utilizes a search or filter in the sql or UI. It might be worth having customers sorted by name instead of ID as it's faster to look by name alphabetically than to scroll through every number.
- *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, All inserts are implemented in the UI and there is a matching insert in the DML.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).
    - There are dropdowns in the insertion section of the UI which would add FK attributes to the insertion. There is an insertion for customer on the shipment invoice insertion but the insertion sql asks for id and the ui is asking for name. It might be the intended goal to have if name not found open tab to insert to new customer?
- *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 3 deletes in the SQL for Customers, shipment invoices and Planets. The Shipment invoices delete feature does remove entities that have FKs from other tables and does not delete any of the information from them.
- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
    - There are updates for the Customers table and the Shipment invoices table. Both tables make use of a dropdown list referencing their planets table which is a nice touch.
- *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.
    - Negative. It appears that all relationship values are listed as NOT NULL so there is no relationship that can be left as a null value.
- *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.*
    - There is some confusion on the Shipment Types page. I assume that Shipment time might be shipping duration i.e. 1 day, 2 day shipping, There might be a better way to label it. Also Shipment ID might be good to have as 'shipping type ID' or 'Type ID' so that there is less overlap with the Shipment table. I had to double check to make sure I was on the right page when looking at the table names. The planet ID is also labeled as package ID

Comment •••

Ahmed Abbas
Last week

HI Benjamin and Ryan,

Overall, great job in implementing the HTML UI for the database schema. The UI allows for inserting, deleting, and updating records. It includes all the tables in the schema and considered various aspects such as M:M relationships and search/filter functionalities. I provided suggestions for improvements that could further enhance the UI. Keep up the good work!

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

- Yes, the UI displays data from all 5 tables in the schema. However, there is a discrepancy between the Sender attribute in Shipment Invoices and the Customer Name attribute on the Customers page.

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

- No, there is no SELECT query that utilizes search or filter to dynamically populate the list of properties. However, there is a drop-down filter to update records.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

- Yes, the UI provides an input field for each table and attribute in the schema when performing an INSERT.

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

- Yes, the INSERT operation adds the corresponding foreign key attributes including at least one M:M relationship. For example, inserting a new customer also inserts a row in the intersection table for the M:M relationship between Customers and Planets.

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 3 DELETE operations for deleting Customer, Planet, and Invoice records that also remove the corresponding rows in their respective tables. But, the Shipment Type records are not deleted.

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.

- No, there are no relationships in the UI or database schema that allow for optional and NULL values. One possible approach could be to create a separate table to represent the relationship with foreign key columns that allow NULL values.

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.

- Some suggestions for improving the UI include using more readable values for boolean attributes, such as "yes" or "no" or "active" or "inactive" instead of 1 and 0.
- I recommend using design elements and different colors that help distinguish the tables from the background. A light color such as white or light gray can help make the text and data stand out more clearly.
- Provide filtering\sorting options to help users quickly find the data they need. For example, this can include drop-down menus, search bars, or sortable column headers.

Comment •••

Long To Lotto Tang
Last week

Hello!

1) Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

- Yes, there are 5 SELECT statements to display the data on the UI.

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

- I am not sure whether the INNER JOIN part should be considered as a filter or not (as it filters some of the data out from other table). I agree with the above comments, some data can be sorted by names (more readable for human) than ID for a better searching.

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, every table equips with INSERT.

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).

- Yes (e.g. in customers, planet can be chosen in dropdown list and thus the planetID will also be 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.

- There are DELETE UI to remove the unwanted records. In addition, in the DDL file, I can see there are lot of ON CASCADE DELETE to cater the needs for the M:M DELETE operations.

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?

- Definitely, each table will equip with update and fields for entering the new data.

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.

- In the Schema, there is no optional relationship stated. In the intersection table Shipment Invoices, all the attributes are NOT NULL. It seems all data is important and not able to make one empty for modification.

8) 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.

- Since it is a draft, the background color makes the text a little bit difficult to read. Other than that, the UI is good and easy to use. In Shipment Types, maybe you can add some descriptive text to explain "0" or "1" (as everyone will have different interpretation of that)

Alec Sudtelgte
Last week

·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, every table has a select that pulls the values for the tables.

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

No, there doesn't appear to be any search/filters.

·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, all tables have an insert.

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

Yes, they all do.

·Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Yes, there is a delete for customers, planets, and shipment invoices.

·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, there is an update for customers and shipment invoices.

·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.

No, there is no optional relationship from what I can see. You could possibly allow for the sender to be anonymous(Null) to satisfy this requirement.

·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.

You could possibly add a short description to the home page to let the user know that they are on the home page. Also the grey for the top column is a bit hard to read the "new" button.

Comment •••

Add comment

Changes from feedback:

- CSS changed for more clear reading of the tables

- App now launched on flip server using Flask
- Update SQL to sorting in order to sort by name for easier search of find
- Implementation of JavaScript to auto populate data into the update fields for edits for specific page.
-

—--

URL to index.html page:
https://web.engr.oregonstate.edu/~riflemab/customers.html **(now on flip)**

Peer Feedback Project Step 1 Review (3 Total)

1/3

Guyllian Dela Rosa
5 days ago

- **Does the overview describe what problem is to be solved by a website with DB back end?**
  - Yes, I think the overview is fun and the problem is laid out clearly that the business is niche and that they want to focus on tracking their packages. I especially liked the great quote.
- **Does the overview list specific facts?**
  - Yes, the overview lists what they intend to track, how many deliveries they expect in a month, as well as the fact the recipient info is not tracked.
- **Are at least four entities described and does each one represent a single idea to be stored a s a list?**
  - Yes, they are interested in tracking the Customers, Planets, Shipment Invoices, and Packages.
- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?**
  - Yes, the outline of the entity details describes the purpose for Customers, Planets, Shipment Invoices, and Packages.
  - The 'distance_from_planet_express_building' i think could be shorter. Maybe just 'dist_to_PE' ?
  - I'm a little unsure why Packages has a 'contents' attribute? It was not clearly indicated in the overview that the Planet Express shipping company would know the contents of the packages they're delivering.
- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**
  - I think the relationship between Shipment Invoices and Packages would we 1:M ? I think a shipping invoice could have more than one packages.
  - In the outline, Shipment Type is described to have a 1:1 relationship with the packages, but the ERD is showing a 1:M relationship instead.
- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**
  - The 'package_idcustomer' in the Shipment Invoices entity I think should be edited to be 'idcustomer' for ease of reading.
  - ERD is showing 'distance_to_hq', however in the outline this is 'distance_from_planet_express_building', need consistency.

Great job! I'm very excited to see the end product of your database!

Comment •••

R Ryan Kirkpatrick
4 days ago

- Does the overview describe what problem is to be solved by a website with DB back end?
  - Yes, the problem at hand can be solved using a website with a DB back end. The overview described the problem sufficiently.
- Does the overview list specific facts?
  - Yes, the overview lists the business specifications, such as where they deliver and how many estimated customers they will have.
- Are at least four entities described and does each one represent a single idea to be stored as a list?
  - Yes. The company has five entities at hand, all of which represent a single idea.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?
  - Yes. Each entity details its attribute datatypes and constraints. Relationships between entities are also outlined.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
  - Shouldn't shipping types be a 1:1 relationship with packages?
  - No M:M relationship is apparent.
  - Customers should have a 0:M relation with packages so customers can be added without the need for a package to be associated.
- Is there consistency in a) naming between overview and entity/attributes, b) entities plural, attributes singular c) use of capitalization for naming?
  - All entities are plural.
  - Attribute names are consistent but confusing. 'package_idcustomer' is not very descriptive.

Comment •••

B Behrad Noorani
4 days ago

1

- Does the overview describe what problem is to be solved by a website with DB back end?
  - Yes the problem can be solved using a DB back end. I like the futurama reference!
- Does the overview list specific facts?
  - Yes the overview describes what they need to track and what is stored.
- Are at least four entities described and does each one represent a single idea to be stored a s a list?
  - Yes the DB will contain 5 separate entities which represent a single idea.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
  - Yes all entities has details for its attributes datatype and constraints and the relationship between entities is listed.
- 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 is correctly formulated
  - There is no M:M relationship \
- is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
  - Yes the consistency is there but some attributes aren't descriptive

Comment •••

Updates Project Step 1:

- Per feedback:
  - updated shipment_types to be a 1:1 with packages on the ERD

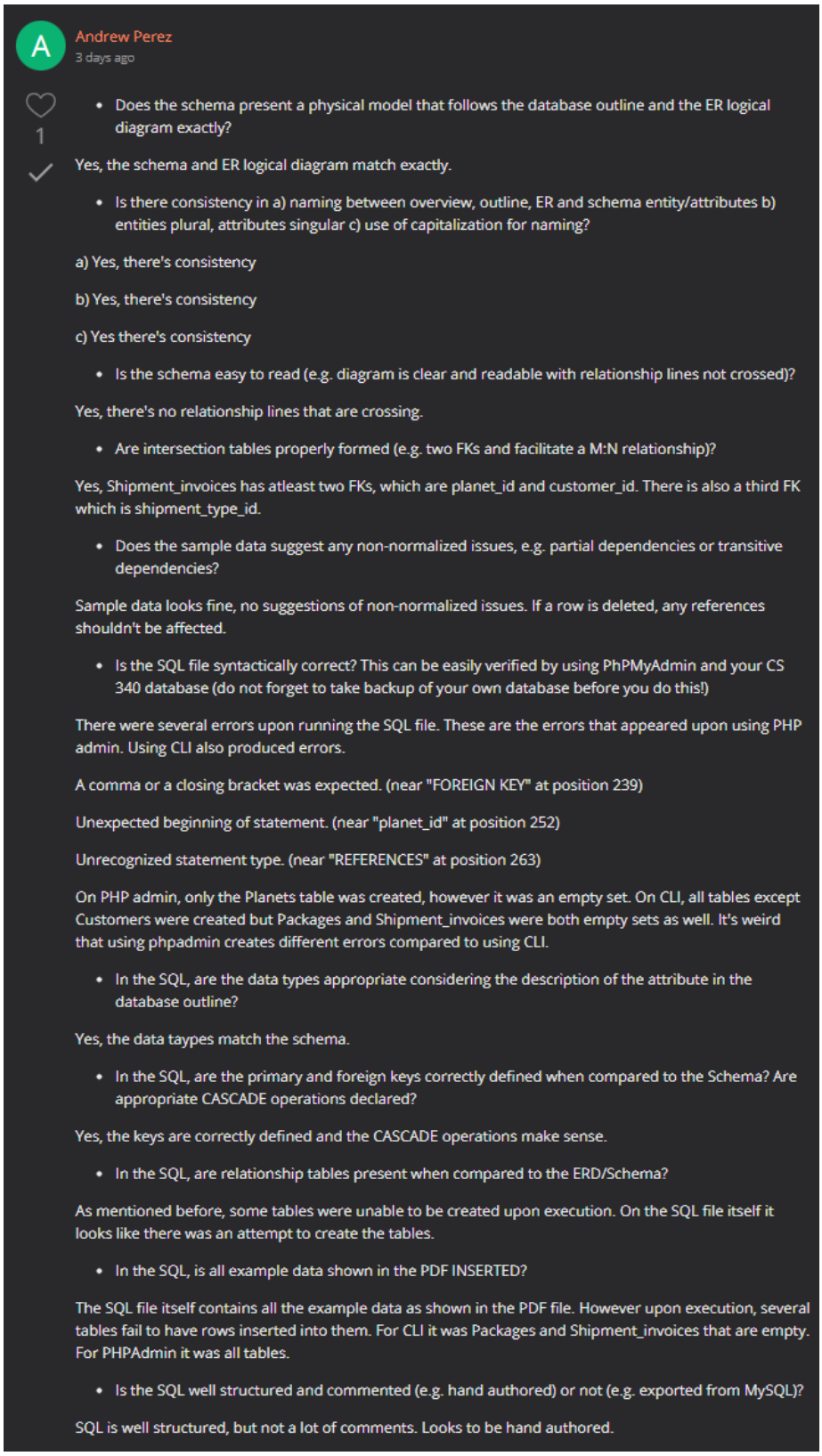
        - changed various attribute names to be more concise
        - removal of description of package contents
        - changed distance_from_planet_express to be distance_from_hq
        - Disagreed with feedback that package_id_customer wasn't descriptive enough, in fact might be too confusing as overly descriptive. changed to idcustomer.
        - Entirely new ERD
- Upgrades:
    - added a secondary address/planet to customers (planet_2 default null) to establish a M:M relationship between plants and customers, as now, customers can have multiple planets associated with them.
        - updated ERD to reflect

Feedback/Updates Project Step 1:

- Per TA feedback:
    - Add Title
    - Hazards in packages → Hazards
    - Ids should be in snake case for consistency
    - Entity tables names should be capitalized, while attributes are not
    - Foreign Key names should match Primary Key
    - Shipment Invoices table should be reconsidered/looked over again
        - The 3 foreign keys do not have a direct relationship between the other 3 tables.
        - If kept these keys need to be arranged in a star schema
    - There was actually a M:M relationship in the original design, so maybe we can remove M:M from planet-customer, and use packages as an interaction table.
        - Packages will also be a entity table because it has more foreign keys than primary keys
        - Will switch the places of shipment_invoices and packages, as it makes more sense.
    - Update varchar length in ER diagram to match outline
    - Update planet_2 and planet_1 to be more descriptive, such as primary_planet and secondary_planet.
- Upgrades:
    - Capitalized entity tables/lower case on attributes
    - Updated all names to use snake case.
    - Switched the position of shipment_invoices and packages in ER diagram
    - Added invoice_id foreign key in packages
    - Added planet_id as primary key for plants
        - Used auto incrementing unique int to use smallest data type
    - Added a simplified er diagram
    - Updated Schema to match updated outline

Peer Feedback Project Step 2 Review (4 Total)

1/4



Andrew Perez
3 days ago

1

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

Yes, the schema and ER logical diagram match exactly.

- 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?

a) Yes, there's consistency

b) Yes, there's consistency

c) Yes there's consistency

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

Yes, there's no relationship lines that are crossing.

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

Yes, Shipment_invoices has atleast two FKs, which are planet_id and customer_id. There is also a third FK which is shipment_type_id.

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

Sample data looks fine, no suggestions of non-normalized issues. If a row is deleted, any references shouldn't be affected.

- 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!)

There were several errors upon running the SQL file. These are the errors that appeared upon using PHP admin. Using CLI also produced errors.

A comma or a closing bracket was expected. (near "FOREIGN KEY" at position 239)

Unexpected beginning of statement. (near "planet_id" at position 252)

Unrecognized statement type. (near "REFERENCES" at position 263)

On PHP admin, only the Planets table was created, however it was an empty set. On CLI, all tables except Customers were created but Packages and Shipment_invoices were both empty sets as well. It's weird that using phpadmin creates different errors compared to using CLI.

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

Yes, the data taypes match the schema.

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

Yes, the keys are correctly defined and the CASCADE operations make sense.

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

As mentioned before, some tables were unable to be created upon execution. On the SQL file itself it looks like there was an attempt to create the tables.

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

The SQL file itself contains all the example data as shown in the PDF file. However upon execution, several tables fail to have rows inserted into them. For CLI it was Packages and Shipment_invoices that are empty. For PHPAdmin it was all tables.

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

SQL is well structured, but not a lot of comments. Looks to be hand authored.

Luke Babcock
2 days ago

Group 65 Project 2 Draft Review

Group 65! Looks like you got some great momentum on your project. Keep going!

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

- Yes, the outline and the ERD match.

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

- There is consistency between the naming of the overview and entities/attributes. All attributes are singular, and all the names of the entities are plural. Only the entities contained a capital letter.

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

- Schema is perfectly legible with no lines crossed.

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

- Yes, Shipment_Invoices contains two Foreign Keys (planet_id, customer_id) and also, a third(shipment_type)

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

- I couldn't find any issues from the sample 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 saw two things when I pulled up your SQL. The first was a comma missing in the Customers table after the Primary Key. The other was another missing comma after one of the values in the INSERT INTO Packages.

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

- Yes, they are all appropriate.

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

- Yes, all primary and foreign keys are correctly defined. Also, contains correct operations for CASCADE.

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

- Yes, the tables were present.

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

- Yes, in a very neat excel file.

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

- Yes, very easy to follow and understand what everything does.

Christopher Sanchez
2 days ago

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

- Yes, the schema follows the database outline and ER logical diagram.

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?

- Yes, there is consistency. Entity names are plural and the various attributes are singular.

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

- Yes, the schema is easy to read and clear in its presentation.

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

- The Shipment_invoices entity is the intersection table is what is facilitating the M:N relationship between Customers and Planets. In your outline it is somewhat confusing when you state under the Customers entity regarding the 1:M relationship between Planets and Customer and also subsequently state that shipment invoices is the intersection table. I do not see how there is a M:N relationship between Customer and Planets, but I do see how it is a 1:M/M:1 relationship

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

- No non-normalized issues noted

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

- SQL file is syntactically correct. You updated sql file you sent in response to Andrew Perez is working

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

- Are you sure you want to have the customer_name attribute under Customers as Unique because you might have two distinct individuals who may have the same name?
- Otherwise all data types are correct and do match the schema.

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

- The primary and foreign keys are correctly defined.
- On delete cascade correctly added

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

- Yes, relationship tables are present

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

- Yes, all the example data is added and matches the content of the PDF

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

    - Yes, the SQL file is well structured and commented.

Good Job!

Stefan Ene
2 days ago

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

Yup, your schema matches the ERD as well as the database outline that are present in your Step 1 doc.

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?

Entity names and data variable names remain consistent between all these four pieces of work, so good job there.

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

Presence of multiple ER diagrams helps getting the point across and does not impede the readability of the initial design doc. Though there are not too many comment lines present in the SQL code, the schema remains well spaced and indented, so it is quite readable.

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

It is quite clear that the Shipment_invoices is an intersection table due to its double FK nature and connection to various entities. Formatting in this case seems solid.

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

No non-normalized issues seem present, and the bullet-pointed explanations that follow the sample data in the doc explaining the different NF requirements help solidify this property.

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

The re-issued version of your coded SQL file appear to be working correctly, as there were no issued compiling this code in our PhPMyAdmin tool.

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

Yes, the coded data types match appropriately the initially drafted variable types for the data storing within entities. I wonder though if there is a way to represent "UNIQUE"-ness in mySQL...

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

Keys are formatted correctly across the entire code, and the appropriate presence of CASCADE operation in your group's code makes me now doubt the correctitude of my own group's submission

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

Yes, relationship tables are present, and the design doc also does a good job of supporting this.

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

The data in the SQL matches the one in the PDF, great job with the Excel representation of this information in the doc.

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

Code is structured very well and clearly hand written and evaluated. Again, some comments are present, but stick to very general divisions between code (table creation and data insertion); readability of SQL file could be improved with more comment divisions between code chunks.

Feedback/Updates Project Step 2 Review:

- Per feedback:
    - Fix SQL file where there were some errors when trying to import.
    - Potentially take out the Unique data type for customer_name, since customers can have the same name
    - Potentially add more comments to SQL file/ comment division between code chunks.
    - Clarify M:N relationship in outline
- Upgrades/Fixes:
    - Updated SQL files to import without problem
    - Added more comments
    - Take out unique data type from customer_name
    - Added all previous feedback and fixes from previous drafts
    - Created SQL file constraining Data Manipulation Queries
    - Created/uploaded 1st draft of html to engineering servers
    - Removed unique data type for customer_name to match outline

Feedback/Updates Project Step 3 Review:

- Per feedback:
    - Need to add a SELECT that utilizes a search/filter for the sql or UI that will dynamically populate a list of properties.
        - Sort customers alphabetically instead of scrolling through numbers
        - Dropdown menus, search bars, sortable column headers
    - There needs to be a relationship that is NULLABLE
        - Potentially have the sender be null, could be like an anonymous package
        - Also potentially have the homeworld/planet for customers be nullable.
    - Shipment time might be confusing, maybe change to shipment duration
    - In the Planet UI, our planet if is labeled package id
    - More user friendly value for the boolean value in shipment types, True or False vs 0 or 1.
        - Can be some descriptive text or a potential ui or sql logic that can apply that.
    - Easier to read text specifically in the background so text and data are easier to read.
        - Grey for the top column is hard to read the “new” button.
    - Add a short description on the homepage so that the user knows they are on the homepage.
- Upgrades/Fixes:
    - Fix the planet page in UI so it displays the correct planet if title.
    - Added a Nullable relationship
        - The customerID/sender in the shipment invoices table is now nullable and can represent a anonymously sent package
    - Updated Data manipulation queries to order customer information alphabetically to make looking for names easier.

- Fixed “NEW” visibility in tables as coloring was a bit dark for some to view (Darkish text on dark background)
- Changed Shipment_Invoice table add to display “Customer ID” as input rather than “Customer” to clarify entry. Also added helper to text to show that value can be empty for “NULL” relationship.
- Did not add a filter or search since the drop down menu that is dynamically populated will account for this.

Group Members of Planet Express (Group: 65):

Benjamin Rifleman - co-lead, submission proofreader, researcher
Ryan Canete - co-lead, researcher, debugger

A) Overview:
   a) Planet Express seeks to be the undisputed leader in planetary delivery systems. Planet Express delivers all 8 planets, The Moon, and Pluto. As a niche delivery company, we understand, *“Our crew is replaceable, your package isn’t.”* ™. In order to stay organized, Planet Express uses the latest DBMS technology in order to keep track of our Customers, Planets/Destinations, Shipping Invoices, and Shipment Types. As a niche business, we expect to do less than 100 deliveries a month but to destinations throughout the solar system. All deliveries will be tracked in Shipment Invoices. For discretion, invoices only contain a senderID and not a recipient. Our site will be able to list all invoices to allow our employees to quickly access data our customers request.

B) Database Outline In Words:
   a) **Customers:** records the details of Customers that we work with and deliver to/from:
      i) Object Table
      ii) customer_id: int, auto_increment, unique, not NULL, Primary Key
      iii) customer_name: varchar(145), NOT NULL
      iv) customer_email: varchar(145), NOTNULL
      v) planet_id: int, NOT NULL, Foreign Key
      vi) Relationship:
         (1) M:M relationship between Planets and Customers with shipment invoices as a intersection table
   b) **Planets**: Records the planets/ destinations that we deliver to
      i) Object Table
      ii) planet_id: int, auto_increment, unique, not NULL, Primary Key
      iii) planet_name: varchar(60), not NULL, Unique
      iv) description: varchar(145), NOT NULL
      v) distance from hq: float, Not NULL
      vi) Relationship:

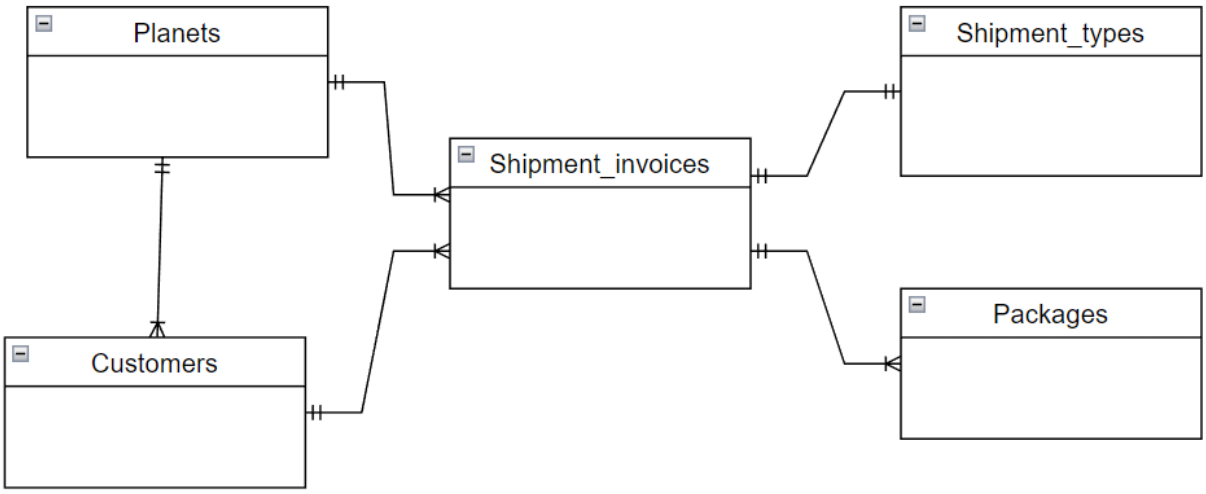
(1) M:M relationship between Planets and Customers with shipment invoices as a intersection table
(2) 1:M relationship with customers where planet_id (destination planet) is the FK in customers.

c) **Shipment_invoices:** Records the shipments
   i) Entity Table & Intersection table
   ii) invoice_id: int, auto_increment, unique, not NULL, Primary Key
   iii) cost: FLOAT, not NULL
   iv) customer_id: int, foreign key(sender of package)
   v) shipment_type_id: int, foreign key, not NULL
   vi) planet_id: int foreign key, not NULL (Will be the destination planet/where the package is going.)
   vii) Relationship: 1:M relationship between shipment invoices and packages where invoice is a foreign key in packages.

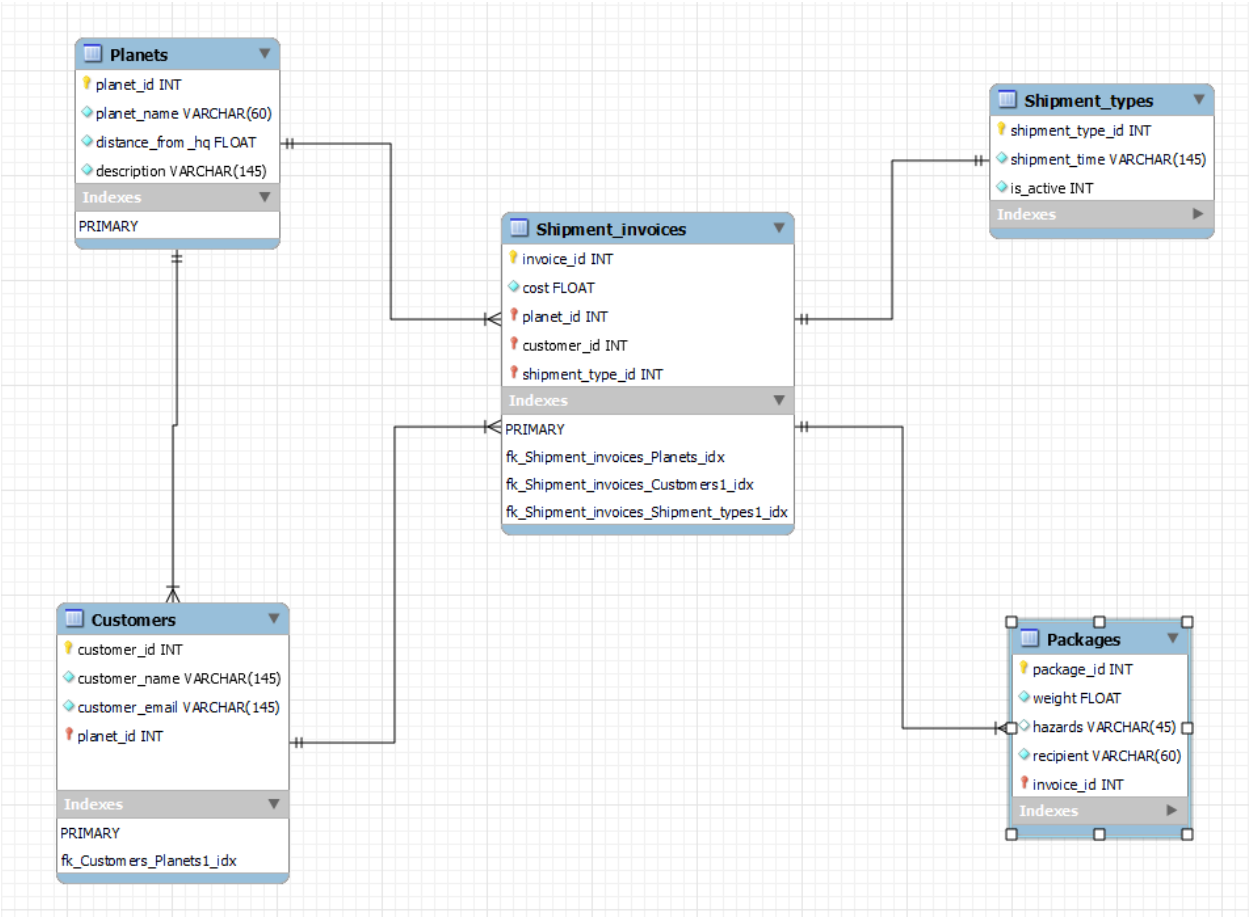
d) **Packages:** Records the contents of the package
   i) Object Table
   ii) package_id: int, auto_increment, Not NULL, Primary Key
   iii) weight: FLOAT, NOT NULL,
   iv) hazards: VARCHAR(45)
   v) recipient: VARCHAR (60)
   vi) invoice_id: INT, FOREIGN KEY

e) **Shipment_types:** Holds the terms for different types of shipping (1 day, 1 year, etc.)
   i) Category table
   ii) shipment_type_id: INT, auto_increment, unique, not NULL, primary key
   iii) shipment_time: VARCHAR(145), not NULL
   iv) is_active: BOOLEAN, NOT NULL, DEFAULT 1 (0/1 for boolean)
   v) Relationship: 1:1 relationship between shipment type and packages

ER Diagram:


Planets
Shipment_types
Shipment_invoices
Packages
Customers

Schema:


Planets
planet_id INT
planet_name VARCHAR(60)
distance_from _hq FLOAT
description VARCHAR(145)
Indexes
PRIMARY
Shipment_types
shipment_type_id INT
shipment_time VARCHAR(145)
is_active INT
Indexes
Shipment_invoices
invoice_id INT
cost FLOAT
planet_id INT
customer_id INT
shipment_type_id INT
Indexes
PRIMARY
fk_Shipment_invoices_Planets_idx
fk_Shipment_invoices_Customers1_idx
fk_Shipment_invoices_Shipment_types1_idx
Customers
customer_id INT
customer_name VARCHAR(145)
customer_email VARCHAR(145)
planet_id INT
Indexes
PRIMARY
fk_Customers_Planets1_idx
Packages
package_id INT
weight FLOAT
hazards VARCHAR(45)
recipient VARCHAR(60)
invoice_id INT
Indexes

Example Data:

Foreign Key | Primary Key

Customers:

customer_id	customer_name	customer_email	planet_id
1	Zapp Brannigan	zapbran@doop.com	1
2	Amy Wong	amy.wong@femputer.com	2
3	Hermes Conrad	hermes.conrad@bureaucrat.com	1
4	Turanga Leela	leela_t@mutantsforjustice.org	1
5	Philip Fry	fry_2000@mail.com	1
6	Bender Bending Rodriguez	bender_moon_base@momsrobots.com	10

Planets:

planet_id	planet_name	description	distance_from_hq
1	Earth	Home planet of the Planet Express crew	0
2	Mars	The red planet	0.38
3	Uranus	The coldes planet in the solar system	19.2
4	Pluto	The smallest planet in the solar system	39.5
5	Saturn	The planet with the most rings	9.5
6	Jupiter	The largest planet in the solar system	5.2
7	Neptune	The planet with the strongest winds	30.1
8	Mercury	The closet planet to the sun	0.39
9	Venus	The hottest plannet in the solar system	0.72
10	Moon	The only natural satellite of Earth	0.00257

Shipment_invoices:

Invoice_id	cost	customer_id	shipment_type_id	planet_id
1	$300	1	1	2
2	$225	2	2	1
3	$100.00	3	3	9
4	50	4	4	1
5	150	5	7	10

Packages

package_id	weight	hazards	recipient	invoice_id
1	10	NILL	Amy Wong	1
2	5	NILL	Hermes Conrad	2
3	2	NILL	Femputer	3
4	1	NILL	Professor Farnsworth	4
5	20	Radiation	Bender Bending Rodriguez	5

Shipment_types

shipment_types_id	shipment_time	is_active
1	1 day	1
2	2 days	1
3	3 days	1
4	4 days	1
5	5 days	1
6	6 days	0
7	7 days	1

- Based on the sample data that we have created, our database meets the requirements of 1NF, 2NF and 3NF.
- 1NF: All key attributes are defined and when all remaining attributes are dependent on the primary key.
- 2NF: There are no partial dependencies
- 3NF: There are no transitive dependencies that we have observed.

URL to index.html page:

https://web.engr.oregonstate.edu/~riflemab/customers.html