James Dempsey
Theodore Janney Jr
Group 63
RPGTracks

Work in Progress URL: http://flip3.engr.oregonstate.edu:60644/

Step 3 Feedback

Feedback 1:

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

-Yes, each table in the schema uses SELECT and this is accurately reflected on the website's UI.

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

-There is no search/filter implemented at this point that I can see.

- **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 all tables in the schema. However, when using it you do not see the text for what is being inserted.

- **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 demonstrates that the INSERT was used properly with all FKs and attributes as well as their relationships.

- **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 is a functional DELETE button and it does not remove from the M:M 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?**

-Multiple tables contain update buttons which accurately describe their effect.

- **Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

-I believe all relationships are currently needed based on the HTML UI

- **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 guys did a great job and overall and just need to make a few small corrections. The vast majority of the prompts were addressed based on what I could see, you are just missing a few things!

Feedback 2:

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

Yes, there is a SELECT for every table in the schema. Data is populated with a few entries so far. The schema is very detailed and has thorough connections for various entities to display their relations.

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

From the website, there are no search functions for any of the entities, but there is a populated list. From taking a look at the DDL file, this is evident to be hard coded for now. This could be an opportunity for improvement for the next round of drafting refinement.

- *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 are implementations for INSERT for every table on the schema. Currently, the website's functionality to perform the insertion is stagnant. It allows typing for a new entry, but it does not add anything to the tables.

- *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).

The UI does indeed demonstrate that INSERTS with FKs appropriately present the characteristics related to those FKs. Taking a look at the DML file, these relationships are consistent across the platforms.

- *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 multiple DELETE functionalities displayed on the website. When trying to use them, they allow the user to click the buttons successfully, although the deletion does not occur yet, so it seems to be placeholders that are hard coded for future functionality. Additionally,

these delete functions are utilized as DELETE CASCADE for the many to many relationships, for example the entity languages and countries, which holds true within the DDL file.

- *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 at least one UPDATE present for several of the entities. For example, when updating countries, the user is prompted to type in the country name, size, and population to appropriately handle the update based on all attributes required in the DML file.

- *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.

From reading the entity relations, my understanding would be that the countries can exist with or without characters being from them. There also could be more optional relationships that exist, but they are not outright visible upon looking at the information provided, as most relationships seem to be required.

- *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.*

Overall, the website design is easy to read, and understand. The colors are minimalistic and can be an opportunity to express creativity in the near future to match the theme of the gaming interest this project is built on. Additionally, I would suggest to work on the syntax errors aforementioned by my peers in their reviews above as well as sharpening the inconsistencies between the SQL files and website functionalities. Lastly, I would suggest to add information on optional relationships, if they are intended to exist, in the descriptions of entities that have them.

Feedback 3:

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

Yes each table represented in the schema has its own page with its own field for each attribute in the table.

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

No, I'm not sure if it was an issue just on my side but I did not see a dynamic 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.**

All pages that represent a table did have a field. However, you should double check your table view in the page because some fields that were included in the add/updates were not included in the table view.

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 shows inserts for FKs are being included in the tables. This is also consistent with the DML 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 is DELETE option for most of the pages/tables and based on the DDL, the CASCADES would consistently delete from the intersection tables but not the other tables.

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 is an update option for all the tables but the DML does not display consistency for what values can be updated.

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.**

This is a little subjective because its not exactly clear which values are required when entering into the UI/theres no validations in place. However, based on your DML there is a NULLable relationship.

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.

It seems like maybe one of yall took on the UI and the other took on the DML which may have caused some inconsistencies. I would just double check that everything is following your schema consistently

Feedback 4:

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

Yes, there is a SELECT for every table in the schema, as there is a separate page to display data for each table. The tables currently don't have much sample data and doesn't match the sample data in the PDF. It would be a nice addition to have more data displayed so the user can get a better idea of what each table contains.

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

No, I do not see any search/filter functionality on any of the pages.

- ***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 every table present in the schema. Based on the UI, I'm not 100% sure if all attributes are taken into account when inserting to some tables. There may be some issues present. For example, when adding an Item, it gives you the option to insert a player name, however the display for the Items does not mention the player name anywhere. I don't believe there is a relationship between Items and Players based on the Schema. Based on the schema, items should have an associated country as well, which is not mentioned in the add page. In the DML file, insert into 'Games' has attributes for size and population, when those aren't attributes of Games.

- ***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 does show that INSERTS that include FKs do correctly display the attributes associated with those FKs. This is also reflected in the DML file, however the syntax is incorrect for some of the insert statements as they are missing the VALUES keyword.

- ***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 on most of the pages. As for deleting from M:M relationships, based on the DDL, ON DELETE CASCADE was implemented for some of the M:M tables, therefore I believe it would correctly delete the corresponding rows for some tables.

- ***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 are edit pages for several tables. The update statements are correctly formed in the DML file, however they don't necessarily correspond to what is displayed in the UI. For example, the Items page allows for editing of the name, description, game name and player name, whereas the DML only shows the item name being updated.

- ***Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

Based on the UI, I cannot confirm whether there is a NULLable relationship, as there is no example of one and the edit pages to not allow for setting anything to NULL. Judging from the ERD, all relationships are represented as mandatory.

- ***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.***

Overall, I suggest you to make sure that the UI accurately matches the Schema in terms of what attributes are present for each entity. Furthermore, I'd suggest adding more sample data to each table in your UI.

Actions based on Step 3 Feedback:

We decided to implement a search feature for Games so you can retrieve the game you were recently playing. We opted to allow searching by game name or player name for easier retrieval and wrote a select statement. The game name will have to be saved in state since it also controls what players, items, etc. are available. Thus, we didn't implement an update or delete sql for the search feature, only a SELECT.

Much of our feedback on step 3 revolved around whether DELETEs or UPDATEs really deleted or updated. This was hard to judge as we used placeholders, waiting for the time when we have connected the UI to the database. There were concerns about deleting an M:M functionality as required in the project guide. The delete feature for our M:M functionality is part of the table showing each individual entry. So, for instance, if a Country and a Language should have a relationship, they are displayed on the Country_has_Language page and on that page in turn there is a button that will delete the relationship. However, in our schema, it should not in turn delete both the language and the country, only delete the connection between the two. Therefore, there's a page that warns about deleting and then allows you to delete just the relationship. If, however, you were to follow the delete button on the separate Country or Language pages, these would delete the Country or Language entirely, CASCADing to all attached relationships in the Country_has_Languages table. We believe the peer feedback we received was stuck on the Country or Language table, not navigating to the Country_has_Languages table.

We did not have a nullable field in our original design. Although it is possible to have no relationship in an intersection table (and thus the result of a query on an intersection table will be NULL), we did not have a table with optional inputs. Most of the entities require each other in an interlocking fashion. However, the Country table does not really require the sizeInKm or Population attribute to be filled, so we changed it to nullable and added a null example on the page.

We also had several typos in our CRUD sqls where the order of table names in intersection tables was reversed. We had a formula for producing the table names but since the relationship could go either way, we had a few alignment issues. Thus, we synchronized the DDL and CRUD sql files. This included a grammatical change from has to have when the tables were pluralized.

Upgrades to the Draft Version:

- We re-designed the frontend to work with React instead of templates because it added more functionality for the future, and will be easier to integrate with data. In addition, it allowed us to populate the edit pages easily to give an idea of what kind of data will be edited.

- Added a search bar to the games page to search
- In terms of data, React allows us to more easily pull data into the forms. So instead of completely hardcoding everything, the data is pulled and builds the forms and view tables as we need them.

Step 2 Feedback

Fixes based on Feedback from Step 2:

The first major feedback shared by almost everyone was the schema diagram being a little difficult to read. While our project is fairly large with a lot of tables we have attempted to alleviate some of the confusing parts of the Schema and remade it to be a little more organized with less overlap.

The other comment that we got from most people was to look over our cascading definitions. This included double checking where we wanted cascades and in what conditions we wanted them. The current DDL_2.sql file has these changes. In addition, while we reviewed the SQL we found some typos we had missed the first time, which have been updated.

In addition we fixed a typo in the DDL that was causing issues when importing the SQL database, losing one of the connections to the Translation Output Tables.

Peer Feedback - Step 2

Feedback 1:

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

Yes, the Schema does follow the original ER logical diagram and database outline exactly as they have stated in the document.

Is there consistency in

a)naming between overview, outline, ER and schema entity/attributes

While the outline does not have the relation tables stated in the document, the ER and schema do follow do have all of the same entities/tables and attributes/rows.

b)entities plural, attributes singular

All tables in the database have plural entities, and all attributes are singular.

c)use of capitalization for naming?

Capitalization is used when necessary for the naming structure of the tables. The attributes are only capitalized when they have two parts to their names. (idCountry, itemName, itemDescription, etc.)

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

The schema is a bit confusing to follow, where all of the different tables are placed all over the place and the relationship lines cross over each other often. I have read the comments made by the developers, and I understand they have tired to make it as readable as possible.

One potential solution would be to have one of the major entities, like Characters, be placed around the top and then have other entities that are tied around them be placed around or below them.

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

Yes, the intersection tables are formed properly, where they share two foreign keys and have a M:N relationship between the two tables.

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

From what I have read of the sample data there doesn't seem to be any non-normalized issues from what has been submitted.

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!)

Yes the SQL file is syntactically correct, where all the syntax works for the different commands in creating the tables for each entity in the database.

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

Yes, the data types are appropriate and fit their descriptions of the attributes in the database outline. Though the relation tables are missing in the database outline.

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

While the primary and foreign and primary keys are correctly defined when they are compared to the rest of the Schema, almost all the foreign keys have the same ON DELETE NO ACTION, and ON UPDATE NO ACTION for the CASCADE operations. The CASCADE operations may want to be changed in the case one of the entities might want to be saved, such as in the case a Language gets deleted the Country it is linked to might not want to be deleted as well.

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

Yes the relationship tables from the Schema are present in the SQL file, though the ERD does not contain any relationship tables.

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

The example data is in the inserted within the SQL file, however due to the operation called **Use 'cs340_dempsjam'** before the Insert statements, I wasn't able to insert the data into a database to test to make sure that the example data shown would insert into the database. The operation appears before all the insert statements so unless it is removed it cannot be tested by other students.

Good work so far! A lot of work ahead for your project, but it's pretty cool so far!

Feedback 2:

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

Yes, there are a total of 7 entities in the schema and 7 on 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?

Yes, there are capitalized entity names in plural form and attributes are all camelcase with underscores between each word.

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

Yes, there is a very straightforward and easy to follow schema.

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

Yes, the Games_Purchases_Details and Games_Genres_Details are the properly formed intersection tables.

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

No, it looks formalized to the 3rd Normal Form.

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!)

Yes, I was able to run it on the flip server without issues.

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

Yes, the data types all appear appropriate for their described attributes.

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

FK and PK are good, I do not see a CASCADE operation declared.

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

Yes, the relationship tables are present.

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

Yes, the data appears correctly inserted.

Feedback 3:

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

From what I can see in the PDF the schema follows the outline correctly. There does not seem to be any errors in the schema.

Is there consistency in

a)naming between overview, outline, ER and schema entity/attributes

All the tables in the schema have the same names and datatypes as the outline.

b)entities plural, attributes singular

All the tables in the database follow the same naming convention. The names are plural, and tables with multiple words in the title are separated with underscores.

c)use of capitalization for naming?

The attribute names use camelCase for things like idName. The names in the schema are all consistent.

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

The schema is spaced out well, however, it is a bit difficult to see due to the complexity of the database. The number of tables and the complex relationships makes it a bit difficult to understand. The relationship lines crossing over each other makes it more confusing. If the tables were spread out over a larger area, it might make it a bit easier to understand the relationship between tables.

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

The intersection tables are properly laid out. The tables Languages_has_languageRules and Countries_has_languages for example are correctly constructed with two foreign keys.

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

There does not seem to be any issues with the data. All keys in the tables seem to correspond correctly with the other tables. There are no missing attributes that seem to be an issue.

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

The SQL file export seems to be correct. It does not appear that there are any syntax errors in the file. As it is an export it would not really make sense for MyAdmin to export an incorrect file.

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

The tables are correctly formed in the sql file, and the attributes match the datatypes in the outline and 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 in the sql file. They match the schema and the outline. The cascade options are difficult to determine if they are correct without knowing the intended use of the database. Some of the cascade options might not be correct as they might have to delete other entries. However, it could be by design. Without more information, it's difficult to determine.

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

The relationship tables are present in the schema and the sql file. The tables are properly defined in the sql file.

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

In my testing of the database, the inserted data is the same as it appears in the pdf.

Feedback 4:

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

> The schema does present a physical model that follows the database outline and ERD, as I was not able to find any inconsistencies

Is t**here 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 in naming between overview, outline, ER, and schema
>
> Entities are plural attributes singular
>
> For capitalization, the naming is consistent

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

> Although it is already addressed on the schema page, the diagram does have crossing relationships, but that is most likely due to the complexity of the database and the number of relations/entities

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

> The intersection tables seem to be properly formed, but the intersection table "Countries_has_languages" seems to have foreign keys that are not linked to the proper tables. Countries_has_languages has keys idCountry and idLanguage, but it is linked to entity Items instead of Countries.

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

> The data does not suggest any non-normalized issues that I can see. It seems to pass the 1NF, 2NF, and 3NF normalization levels

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

> The SQL file does not seem to have any syntax errors

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

All the data types make sense for the labels given. For definition in LanguageRules, you might consider changing the datatype from varchar(255) to TEXT just in case a the definition is longer than 255 characters.

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 in the SQL file

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

The relationship tables are all present when compared to the ERD and Schema

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

The Insert values are present in the SQL file

Step 1 Feedback Section

Fixes based on Feedback from Step 1:

We received feedback from our TA that made us immediately rewrite our database. Then, when that was graded again, we received a second round of feedback. The first round of feedback had 8 specific action items. We repeat the TA feedback in the Feedback received section, but since this was a rather substantial rewrite we will give a quick recap here. The first feedback item was about a naming convention, and we switched all primary keys to be singular. The second feedback item was about consistent naming between primary keys and foreign keys and we implemented that as well. Feedback items 3, 4, 5, and 7 were about making sure the ERD and the outline in this document matched and we made sure they matched. Feedback 6 was the most conceptual, since it required us to match 1:M relationships between the ERD and the outline.

We then received a second round of feedback. This feedback was implemented not in our resubmit but in the Project Step 2 Draft. The largest, most conceptual piece of feedback was that we could delete our intersection tables. This was because they were all tables meant to assist M:M relationships and only contained the primary keys of the two tables they were intersecting. However, modeling this became confusing in MySQL Workbench and did not significantly reduce our workload when generating our DDL script. Therefore, we retained the intersection tables as a reminder that these were, in fact, M:M relationships. In future designs, however, we would probably take the shortcut and not show the intersection tables. The other three action items were typo or spelling related and are implemented in the current version of the outline and ERD.

Another aspect of our initial feedback was a general confusion about why we had a bit about language generation tacked onto what looked like a generic RPG style game. In reality, we had always seen it almost the other way around. The language generator was the kernel of the idea, but languages take place in the middle of a context - people, places, specific eras of time. The general material for the language is pulled from the other tables as well as input from the UI. The goal of the project overall, however, is to generate artificial languages in a way that is easy to scale and can adapt to new algorithms for the language generation itself.

TA Feedback 1:

There are some pretty serious logical gaps in this proposal. Please read my suggestions below and reach out if you have any questions. 1) Change your PKs to be singular (for example idLanguage instead of idLanguages). 2) It's a good idea to name a FK the same as the PK it is referring to, many of yours don't match so I would go through all your FKs and check them. 3) The attributes in your outline do not match what is in your ERD. For example in the ERD you have gameID as an attribute for Countries but that is not in the outline. 4) Your entities are capitalized in your outline but lower case in your ERD. I recommend capitalizing them in the ERD. 5) You have not included the relationships for Games in the outline. 6) Your relationships described in the outline do not match the ERD. For example in the outline you say Countries has a M:M relationship with Languages, a 1:M with Items, and a 1:M with Players but none of these relationships are shown in the ERD. 7) The entities in your outline and ERD do not match. The players entity is not in the ERD at all, and there are many intersection tables in the ERD that are not in the outline. 8) In future drafts please highlight the parts of reviews that include suggestions 9) The relationships between Languages, Translations, and Rules don't quite make sense to me. Based on your description in the overview, it seems like each language would have one rule used to create translations. Or maybe each translation would be associated with one rule that was used to make that translation. I think this needs to be fleshed out a little bit more.

TA feedback 2:

Great work team! I can tell you put a lot of work into this. I also got some feedback from Professor Curry regarding intersection tables, as there was some confusion in regards to them. If an intersection table exists only to assist a M:M relationship (meaning it has only the two FKs and no other attributes) then it does not need to be included in the ERD or Outline. If it stores other information, then it is considered it's own entity and should be included. Looks like your intersection tables fall into this first category, so you can remove them from the ERD and outline. This should make things look a bit less complicated! Let me know if you have any questions about this. Lastly, there are just a few inconstancies I noticed. Just address these for step 2. 1) Countries.sizeInKm and Countriess.population is type VARCHAR in the outline but type INT in the ERD 2) Characters.characterName is length 120 in the ERD but length 45 in the outline 3) I think there is a typo in the Translations section in the outline. It says M:1 relationship with language, but in the ERD it shows a M:M relationship.

Peer Feedback - Step 1

Feedback 1:

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

Yes, all the relationships are clearly laid out as well as the needs for the overall project. The overview also explains how the different tables will communicate with each other.

Does the overview list specific facts?

Yes, it declares minimum and maximum amounts, the number of games per week, and all the relationships.

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

Yes, all entries are a single idea and are easy to understand based off of title alone.

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

Yes, the outlines are clear and have consistent naming conventions. Each one has the details to show constraints as well as data types. Each one also begins with a summary for a full explanation and ends with a statement to describe the relationships with other 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?

I believe all the relationships are correctly formulated. There are multiple M:M relationships, and from what I can tell the ERD is well done. I am having a little trouble reading it, the picture was either compressed too far or it's just too small.

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

Everything seems consistent except for some attributes are named idName and then others are Name_id (as an example.) I would make sure these are all using either an underscore or camel case.

Feedback 2:

Hi all! I really like your concept; great work on this.

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

Yes, the overview goes over how many players, games and entries are needed to keep track of by a game master. It also describes the types of relationships these entities have, highlighting the need for a website with a database backend. Could potentially include the difficulty in tracking this information by hand, but I think you indirectly addressed that.

Does the overview list specific facts?

Yes, the overview lists the specific amount of games in one week, lower and upper bound on players in a given game, and the amount of entries for a specific game. Could potentially outline how many languages, characters, etc. there are individually instead of just the total of all of those, but I think what you have works too.

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

Yes - games, players, countries, characters, items, languages, language rules, and translations are all described. This may be a bit complicated with your language entities, I'm not sure I fully understand the need for language rules and translations, it may be simpler to omit these two entities.

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

Yes, each entity outlines its purpose and attribute datatypes. Relationships are all listed as well.

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, all the relationships make contextual sense and 1:M relationships are correct. There are a few M:M relationships that all make sense as well. The ERD is a bit blurry but looks about right.

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 and attributes have consistent naming.

Feedback 3:

Hi Team 63!

Cool idea for a project, I am getting some D&D/RISK vibes, super cool! Excited to see the final product and what you guys come up with over the quarter. Please see my comments below:

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

The over describes in detail the expectations of the database, however, it does not specifically state a problem that it is solving. What issues is the Game Master having that he needs a DB? How will the DB solve these issues?

Does the overview list specific facts?

Yes, the overview is highly specific and gives examples of how the entities will interact with each other. It also demonstrates how the DB will be utilized and helpful to the organization of the significant amounts of data.

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

Yes, at least four entities are described. Each entity appears to represent a single idea. I also like how you guys separated the language entity into parts that will make the data easier to gather and analyze.

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 each entity is well described and clearly states the attributes and their datatypes. Relationships are well described and were clearly well thought out.

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

All the 1:M relationships appear to be correctly formed. There is at least one M:M relationship between items and characters. I do think there should be relationships described with the entity games, even though it has a relationship with most, if not all of the other entities.

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

There appears to be consistency in naming between overview, outline and ERD. The ERD is a little hard to visualize due to its size but from what I can see it seems to be consistent with former descriptions.

Feedback 4:

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

Yes. The database is being used by a game master trying to keep track of multiple campaigns, including players, characters, and worldbuilding information.

Does the overview list specific facts?

The overview details the number of campaigns, an estimate of how many players are in each campaign, and a range of how many worldbuilding elements (i.e. items, countries) can be associated with a campaign.

One concern I have is that the purpose of the Translations and LanguageRules tables are not covered thoroughly in the overview. Reading the database outline, it seems as though the language generation part of worldbuilding is a much bigger focus for the game master than the overview suggests. I would add a few sentences to the overview to cover the roles of these tables and the importance of tracking them.

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

Yes - more than four entities are described, and each one represents a single idea.

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

The outline is solid and gives a good idea of what the project will look like. I had some confusion regarding the relationships between the entities, but that is discussed more below.

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

There are several M:M relationships, i.e. between Items and Characters, represented by the intersect table "Characters_has_Items" (showing that a character can own many items, and that items of the same type can be owned by many characters), and between the "Languages" and "Language Rules" tables (a language can have many rules, and rules can apply to more than one language).

The 1:M relationships generally make sense. However, unless I'm misreading the document, the database outline seems to state in different places that Characters have a M:M relationship with Languages (in the relationship portion of the "Characters" section), and a 1:M relationship with Languages (in the relationship portion of the "Languages" section).

Additionally, the ERD is unclear in places - i.e. a M:1 relationship between the "Items" table and the "Countries" table is indicated in the write-up, but I could not see this represented in the ERD.

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

Both entities and attributes are named by capitalizing the first letter of the word and using underscores. I would suggest switching to a different naming convention for attributes, and making the primary keys singular to match other attributes.

Summary of Changes from Step 1 Feedback:

Made attribute naming camelCase and entity naming underscore case.

Added a paragraph to the overview stating the relationship between the language generator and the larger database design.

Added two sentences in the existing overview to make the implicit relationship between concepts more explicit.

Added intersection tables to keep up with modules.

Rewrote attributes to have consistent naming scheme.

Made sure ERD and outline were aligned.

Made sure ERD and outline had the same 1:M relationships.

Made sure M:M relationships had intersection tables.

Upgrades on Draft Version:

Our project idea has been evolving very quickly. What started as a pattern matching exercise to create a fake language has emerged into creating a more modular system that allows a plug-and-play for language generation algorithms. We decided to implement the feedback we received in our initial submission for this final version. We debated very seriously the size of our project since we have a lot of planned tables. We added the intersection tables to show how our many-to-many relationships worked. We also revised both the outline and the ERD to make sure they aligned with our continued development of the project idea. This is the finalized version of our project proposal.

***Ensure you are logged in to Oregon State University VPN to view the index page information.**

Overview

A poor game master runs 10 games in a week with at least 5 players, but up to 20 players in a single game. The website must record *Languages*, *Characters*, *Items*, and *Countries*, sometimes totaling over 2000 entries each, for the game master's *Players*. Keeping track of this via pen and paper is both time consuming and the records can be easily lost. This is further complicated if you want to create an otherworldly language, but trying to track it in the real world. Each character has a one-to-many relationship with languages and items, and a one-to-one relationship with countries. A player has a one-to-many relationship with a character. This allows a game master to see at a glance the state and background of every player's characters.

One of the major focuses will be the Language Translation module that would allow a game master to easily create a language and translate text from English to that created language. The game master will be able to create LanguageRules that are used to create Languages. The program will save any translations from English to a specific language in the Translations table. The general overview will be as such:

1. Game master assigns one or more rules to a language (where certain parts are modified as variables). As an example, if the stored definition will shift the x letter y spots to the right, the game master can change x and y.
2. A language is created using one or more rules. Or more specifically, a language is given a name, a description, and a set of rules that determine how it's created.
3. Any time a game master uses a language to translate from English to the specified language it will be saved in the Translations table.

Database Outline

Games: records a list of games the game master has created. Most other tables relate to this table as described in their sections.

- idGame, INT, auto_increment, not NULL, PK
- gameName, VARCHAR(120), not NULL
- Relationship: 1:M relationship with Items, Countries, Translations, Languages, Characters, and Players. A central hub connected to all other main forms of data.

Players: records of players participating in the game master's game.

- idPlayer, INT, auto_increment, not NULL, PK
- playerName, VARCHAR(120), not NULL
- idGame, FK with Games table
- Relationship: 1:M relationship with the Characters table. M:1 relationship with Games table.

Countries: records of fictional countries *residing in* or *claiming allegiance to* by records in the Characters table.

- idCountry, int, auto_increment, not NULL, PK
- countryName, VARCHAR(120), not NULL
- sizeInKm, int, not NULL
- population, int, not NULL
- idGame, INT, FK with the Games table
- Relationship: M:M relationship with the Languages table, 1:M relationship with the items table, 1:M relationship with the Characters table. M:1 relationship with Games table.

Characters: records of the fictional character who speaks at least one language, belongs to at most one country, and is created by at most one player.

- idCharacter, int, auto_increment, not NULL, PK
- characterName, VARCHAR(120), not NULL
- characterDescription, VARCHAR(255), not NULL
- idGame, INT, FK with Games table
- idPlayer, INT, FK with Players table
- idCountry, INT, FK with Countries table
- Relationship: M:M relationship with the Items and Languages tables, M:1 relationship with the Countries, Players, and Games tables.

Items: records of fictional items *owned by* records in the Characters table.

- idItem, int, auto_increment, not NULL, PK
- itemName, VARCHAR(120), not NULL
- itemDescription, VARCHAR(255), not NULL
- idGame, INT, NOT NULL, FK with Games table
- idCountry, INT, FK with Countries table
- Relationship: M:M relationship with the Characters table, M:1 relationship with the Countries and Games tables.

Tables related to Language Generation

Languages: records of languages based on rules for in-game worldbuilding. English or Elvish, for instance.

- idLanguage, int, auto_increment, not NULL, PK
- languageName, VARCHAR(120), not NULL
- languageDescription, VARCHAR(255), not NULL
- IdGame, INT, FK with Games table
- Relationship: M:M relationship between Languages and Countries,LanguageRules, Characters tables. M:1 relationship with Games table. M:M between Translation Outputs, Translation Inputs, and Languages.

LanguageRules: records of rules that apply to languages in order to create, and be able to recreate, translations. So, for instance, the rules governing how Latin turned into Spanish would be stored in this table. This will be simplistic at first (turn the letter a into the letter e) but can be made more sophisticated outside the course.

- idLanguageRule, int, auto_increment, not NULL, PK
- ruleName, VARCHAR(120), not NULL
- definition, VARCHAR(255), not NULL
- variableList, VARCHAR(255)
- Relationship: M:M relationship between Language Rules and Languages.

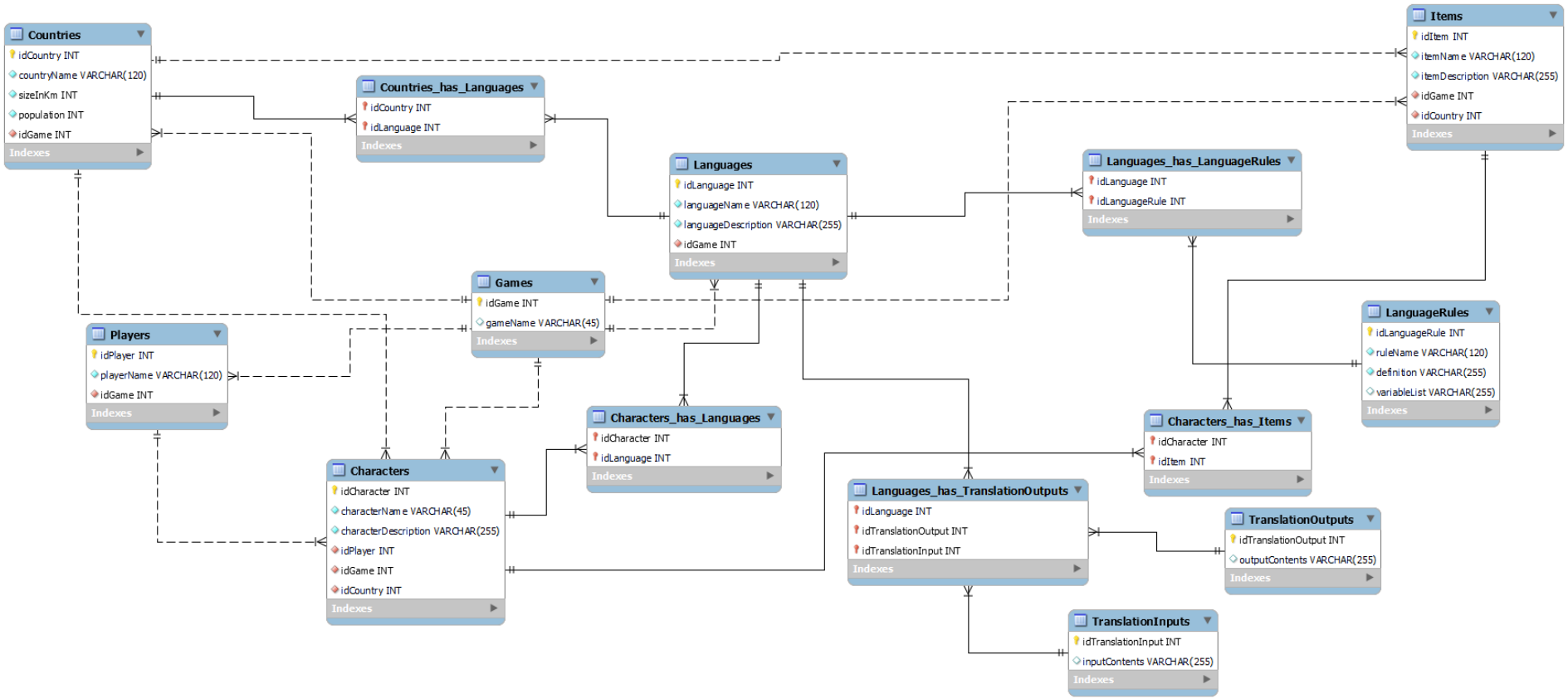
Translation Inputs: records the inputs, or untranslated text to be translated for the program.

- idTranslationInput, INT, auto_increment, not NULL, PK
- inputContents, VARCHAR(255), not NULL
- Relationship: M:M between Translation Outputs, Translation Inputs, and Languages

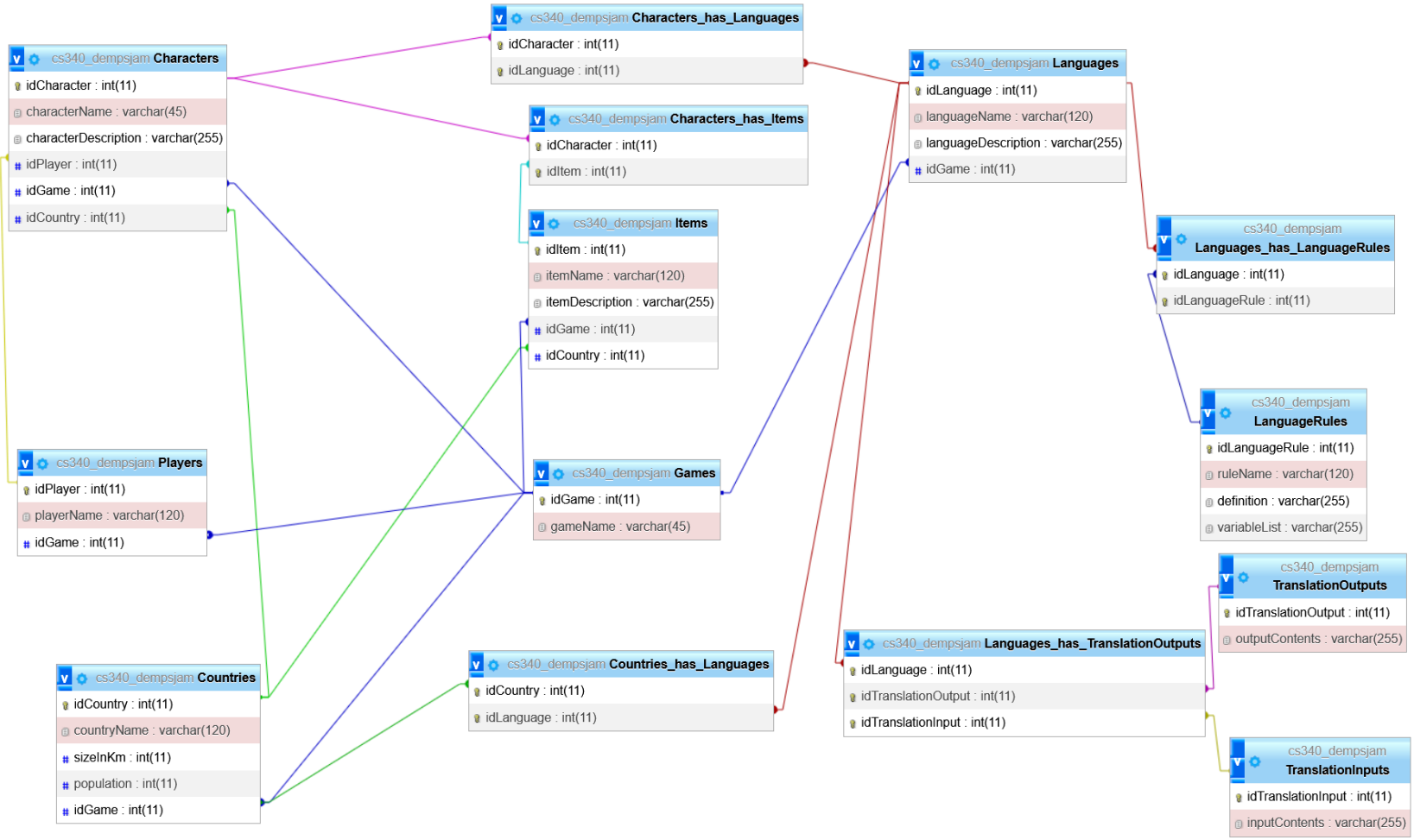
Translation Outputs:records the inputs, or the translated text, after language rules have been applied.

- idTranslationOutput, INT, auto_increment, not NULL, PK
- outputContents, VARCHAR(255), not NULL
- Relationship: M:M between Translation Outputs, Translation Inputs, and Languages

Entity-Relationship Diagram:



Schema:


cs340_dempsjam Characters
idCharacter : int(11)
characterName : varchar(45)
characterDescription : varchar(255)
idPlayer : int(11)
idGame : int(11)
idCountry : int(11)
cs340_dempsjam Characters_has_Languages
idCharacter : int(11)
idLanguage : int(11)
cs340_dempsjam Characters_has_Items
idCharacter : int(11)
idItem : int(11)
cs340_dempsjam Items
idItem : int(11)
itemName : varchar(120)
itemDescription : varchar(255)
idGame : int(11)
idCountry : int(11)
cs340_dempsjam Languages
idLanguage : int(11)
languageName : varchar(120)
languageDescription : varchar(255)
idGame : int(11)
cs340_dempsjam Languages_has_LanguageRules
idLanguage : int(11)
idLanguageRule : int(11)
cs340_dempsjam LanguageRules
idLanguageRule : int(11)
ruleName : varchar(120)
definition : varchar(255)
variableList : varchar(255)
cs340_dempsjam Players
idPlayer : int(11)
playerName : varchar(120)
idGame : int(11)
cs340_dempsjam Games
idGame : int(11)
gameName : varchar(45)
cs340_dempsjam Countries
idCountry : int(11)
countryName : varchar(120)
sizeInKm : int(11)
population : int(11)
idGame : int(11)
cs340_dempsjam Countries_has_Languages
idCountry : int(11)
idLanguage : int(11)
cs340_dempsjam Languages_has_TranslationOutputs
idLanguage : int(11)
idTranslationOutput : int(11)
idTranslationInput : int(11)
cs340_dempsjam TranslationOutputs
idTranslationOutput : int(11)
outputContents : varchar(255)
cs340_dempsjam TranslationInputs
idTranslationInput : int(11)
inputContents : varchar(255)

A friendly note on our schema as generated by MySQL Workbench. We tried optimizing the layout to look as easy to read as possible. Some lines cross and as can be seen, the Games table is connected to everything else and is at the center of this web. We would be happy to clarify any confusion on the few remaining line crossings.

Example Data

The following tables contain the data also included in the SQL with INSERT statements. The data for an entity/table will follow the bold name of the table in question.

Games

idGame	gameName
1	First Game
2	Second Game
3	Third Game
4	Fourth Game

Players

idPlayer	playerName	idGame
1	Donald Duck	1
2	Mickey Mouse	1
3	Goofy	1
4	Minnie Mouse	2
5	Scrooge McDuck	2

Characters

idCharacter	characterName	characterDescription	idGame	idPlayer	idCountry
1	Bilbo Baggins	Ringbearer	1	1	1
2	Frodo Baggins	Hereditary Ringbearer	2	2	3
3	Meriadoc Brandybuck	Buff hobbit	1	3	2
4	Sam Gamgee	Loyal hobbit	2	4	4
5	Peregrin Took	Buff hobbit Supreme	1	5	1

Items

idItem	itemName	itemDescription	idGame	idCountry
1	sword	A bladed weapon	1	1
2	knife	A short bladed weapon	1	1
3	Dull Sword	A dull bladed weapon	2	3
4	Sharp Knife	A sharpened knife	2	4
5	axe	The biggest axe ever	1	2

Countries

idCountry	countryName	sizeInKm	population	idGame
1	USA	96000000	330000000	1
2	China	94000000	1400000000	1
3	England	70000000	70000000	2
4	New Zealand	50000000	5000000	2

Languages

idLanguage	languageName	languageDescription	idGame
1	English	Language of the teamakers	1
2	Ignok	Language of the fisher people	2
3	Chinese	Language of the unified Han	1
4	Welsh	Language of the sheep gazers	2

Countries_has_Languages

idCountry	idLanguage
1	1
2	3
3	2
4	4

LanguageRules

idLanguageRule	ruleName	definition	variableList
1	reverse word	reverse	NULL
2	Random Vowel	Insert random English vowel into $1	2
3	delete last 2 and add e	delete last $1 if length of word > $2 add $3 at the end	2, 3, d
4	add a at beginning and x at end	add $1 at the beginning of a word and adds $2 at the end	a, x
5	change the letter b to the letters vh	Select letter $1, changes to $2, except at the end of the word	b, vh

Languages_has_LanguageRules

idLanguage	idLanguageRule
1	1
2	2
3	3
4	4

TranslationOutputs

idTranslationOutput	textOutput
1	eno
2	tewo
3	thrd
4	aredx

TranslationInputs

idTranslationInput	textInput
1	one
2	two
3	three
4	red

Languages_has_TranslationOutputs

idLanguage	idTranslationOutput	idTranslationInput
1	1	1
2	2	2
3	3	3
4	4	4

Characters_has_Languages

idCharacter	idLanguage
1	1
2	2
3	3
4	4

Characters_has_Items

idCharacter	idItem
1	2
2	4
3	5
4	3
5	1