

James Dempsey
Theodore Janney Jr
Group 63
RPGTracks

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.

Feedback received:

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 inconsistencies I noticed. Just address these for step 2. 1) Countries.sizeInKm and Countries.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.

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 backend?

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.*

Great job, guys! Not too much to change, in my opinion. Hope the quarter goes well!

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:

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.

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.

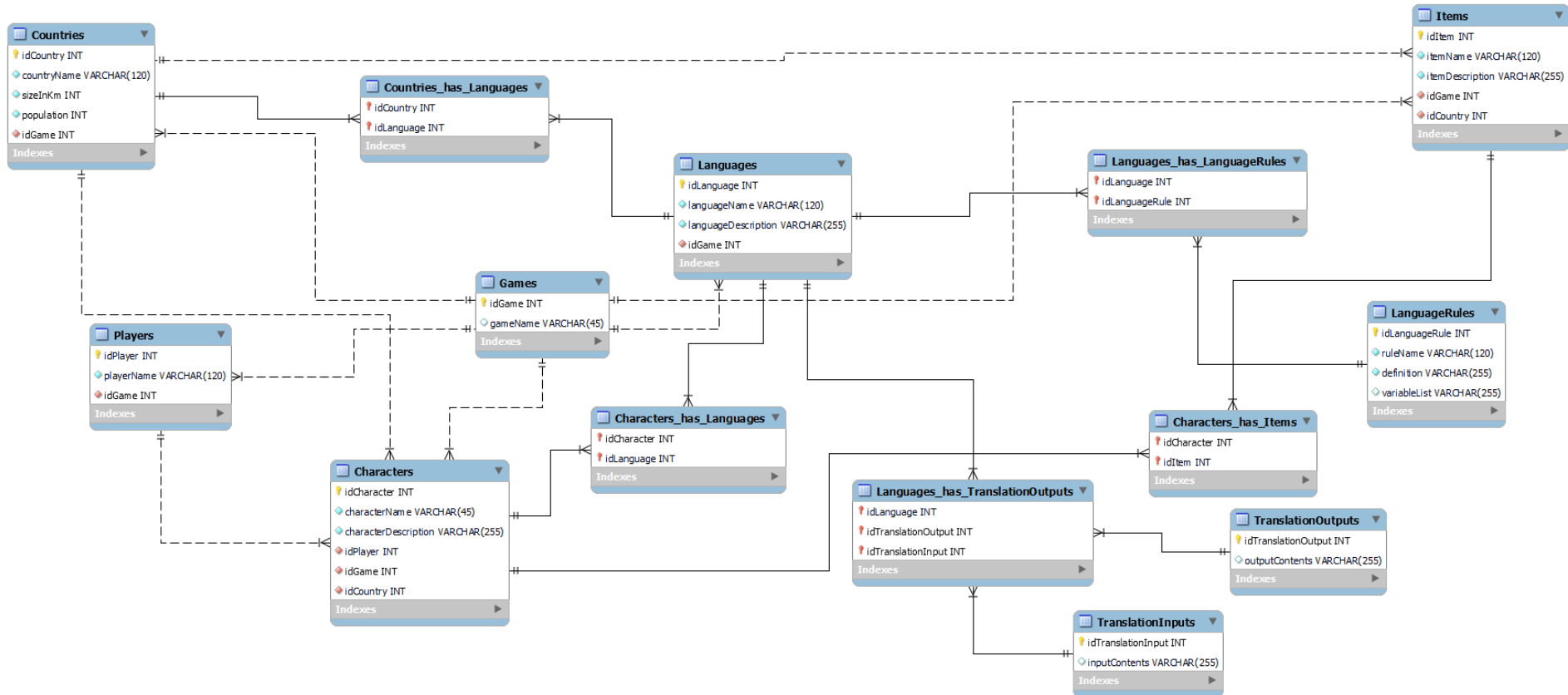
Translation Inputs: records the inputs, or untranslated text to be translated for the program.

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

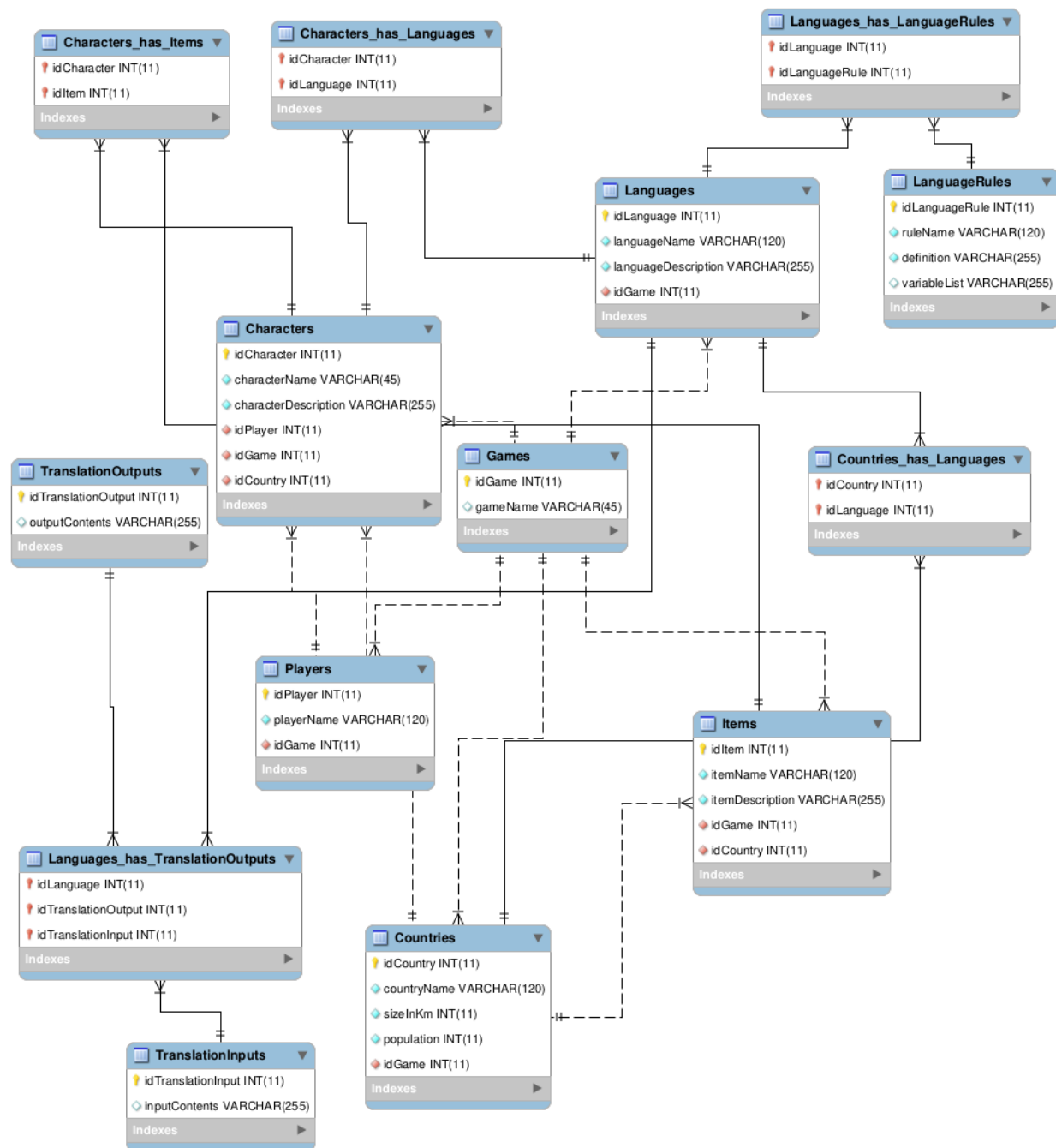
Translation Outputs: records the inputs, or the translated text, after language rules have been applied.

- 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:



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