Team Member Names: Alexandra Fren, Raymond Zhang

Project Title: Stardew Valley Item Guide

Feedback by the Peer Reviewer:
Step One

Review by Damian Russ:

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

The overview does an excellent job of introducing the basic goal of the game and the challenge that its many items offers – knowing which items are needed, and when. The overview lists specific numbers for characters and items, which are the core entities around which the database will be based. It also lists the number of bundles.

I would also suggest mentioning how the time / seasons can limit access to shop merchandise and other items, because this is an additional reason for why a database tracking items and their uses can be so helpful for a new player. My personal story is that I missed catching a pufferfish, and then had to wait a whole year because I didn't know about the Traveling Cart. I also didn't realize that it took so long for a rabbit to give you its foot, or that truffles can't be found in winter and I definitely would've timed my rabbit and pig purchases better had I known.

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

The group has five entities: Shops, Items, Characters, Bundles, and Regions. The group also created three tables to support the three M:M relationships: ShopItems, BundleItems, and CharacterItems. Each entity represents a single idea.

*The Entity Relationship Diagram will need to be updated, because it is still using "Bundles_has_Items instead of BundleItems.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

The group clearly listed the purpose of each entity, listed a number of attributes and their datatypes and constraints for each entity, and described the relationships between entities. The group described the

exact nature of the relationships, such as a shop having many items and an item being sold in many shops for the M:M relationship for Shops and Items.

One suggestion I would make is to include a “season” attribute for the Items entity. It can be as simple as a varchar, and would be very helpful in determining when a player can acquire certain items. From a player’s perspective, it may be even better to make Seasons an entity for easy searching by items available in a particular season, but since you already have five entities, a string is likely best.

My second suggestion is to add a “description” attribute for each entity, for writing out important details about each character / item / store / region / bundle.

The second question was not required in the rubric, but the team does outline which tables will be implemented and which team member is assigned to each table.

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 are correctly formulated. The group has a 1:M relationship between Regions and Characters, as well as between Regions and Shops. There are also 1:M relationships supporting the M:M relationships between Shops and Items, Bundles and Items, and Characters and Items.

The ERD presents a logical view of the database. It models the relationships between the entities, and contains the specified attributes with further details, such as the number of characters allowed by varchar.

For the tables like ShopItems, it’s my understanding that two FKs are sufficient for establishing a unique row, so I think it would be possible to leave out PKs like shop_items_id.

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

In general, there is naming consistency between the overview and entity/attributes, though shops and regions are not mentioned in the overview. These could both be mentioned, to add to overall consistency.

There is a small mistake in the ERD, where BundleItems was left as Bundles_has_Items.

Entities are capitalized and plural, while attributes are lower case, snake case, and singular (with a reasonable exception being “operating_hours”).

Review by Hayden Meek-Avedovech

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

Yes, it clearly states the problem it's trying to solve and how the purpose of the website will solve this problem. It also makes the fact that the game is complex and has a lot of items used for multiple purposes can be stored in a DB backend.

Does the overview list specific facts?

Yes, it gives multiple specific numbers and clearly states that the multitude of numbers makes the game initially difficult due to certain information not being available or confusing.

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

There are 5 entities in total and each one represents a single idea that can be stored as a list. The entities are Items, Bundles, Characters, Shops, and Regions. There are other tables that are used as intersection tables to facilitate M:M relationships. Not necessary, I would change Characters to NPCS because it is a bit more clear this way.

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

For the most part, the entity details describe their purpose of themselves. The Characters details I think should include a sentence or two describing that they are the entity that the playable character is trying to befriend. List attributes are mostly good as well, but there were some issues and confusion. The first one was that birthday in characters should be changed to DATE rather than varchar. The second was that the intersection tables used to facilitate the M:M relationships I don't believe need their own PKs and should just have two FKs. Relationships are clearly described.

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 ERD presents a logical view of the DB and all relationships. The two 1:M relationships are correctly formulated. There are three M:M relationships in the project which are presented correctly although it might cause some confusion or difficulty down the road.

Is there consistency in

a) naming between overview and entity/attributes

It is consistent between both overview and entity/attributes

b) entities plural, attributes singular

Excluding operating_ours in Shops, yes everything is correct.

c) use of capitalization for naming?

Capitalization is consistent between Entities and Entities, and attributes and attributes. They are not consistent between entities and attributes.

Review by Lance Cargill

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

Yes, it describes the problem of the large number of items in the game, which are used to establish relationships with NPCs and to create bundles to restore the community center. A website powered by a database back end would allow users to quickly figure out which items are needed complete both tasks.

Does the overview list specific facts?

Yes, it lists the 34 NPC's, the 30 Bundles, and the 500+ unique game items.

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

Yes, at least 4 entities are described including: Characters, Items, Bundles, Shops.

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

Yes, the outline clearly lists the entities, attributes, data types, constraints and relationships between entities. I also appreciated how each M:N etc. relationship listed an explanation for how it worked, like "Each character can have many favorite items, and each item can be the favorite item of multiple characters"

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, the one-to-many relationships look ok, only seems to apply to the Regions entity (and related Shops/Characters). All the other 1:M are for intersection tables.

It looks like there is a 1:1 between Characters and Shops based on the shop_character_id foreign key.

One M:M relationship applies to characters and items, with an insertion table CharacterItems bridging the two.

Another M:M relationship applies to Items and Bundles, with an insertion table Bundles_has_Items bridging the two.

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

Looks like the attributes are all lowercase and snake_case where needed. The Entities are mostly Camel Case, but one intersection table is snake case.

Yes all entities appear plural, operating_hours is the only plural attribute.

All entities are capitalized.

Review by Benjamin Ling

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

Yes. The overview clearly describes the game world, its game mechanics, and the entities that exist in the game. The proposed database will be used to help Stardew players in management and tracking of items that can be used towards accomplishing two goals in the game: (1) befriending NPC's with gifts and (2) assembling 30 bundles needed to restore the community.

·**Does the overview list specific facts?**

Yes. The overview states the specific number of unique items, NPC characters, and bundles that need to be tracked. The specific mechanics and goals of the game are also clearly defined.

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

Yes. There are five items described (Characters, Shops, Regions, Items, Bundles). Each of these entities clearly represents a single idea to be stored as a list.

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

The outline clearly describes the purpose of each entity. For each entity, the datatype and constraints are listed. The relationships in each entity are also clearly defined. For all M:M relationships, the

intersection table is clearly stated. For example, the relationship between Characters and Items clearly explains that Items can be the favorite of many Characters while a Character can have multiple favorite Items.

·**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 of the 1:M relationships are clearly formulated. There are three M:M relationships(Items and Bundles, Items and Characters, and Items and Shops). The ERD clearly presents a logical view of the database and it is very easy to see the relationships between each table.

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

a) Yes there is consistency in naming between the overview and the entity/attributes. I suppose one thing I did notice was that the overview says “NPC Characters” instead of just “Characters” as it’s called in the outline. I suppose the entity name “Characters” is ok as long as there are no other types of characters (since no data on “player characters” needs to be tracked).

b) Yes, the names of entities are plural (using upper camel case) and the attributes are singular (using snake case).

c) Capitalization for naming is used with entity names.

Step Two

Review by Gary Lutwen

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 model the same database, and they follow the outline.

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

Yes. There is consistency in naming between the overview, outline, ER diagram and schema (for the entities and attributes). The entities are plural, and the attributes are singular. The entities are all capitalized. The attributes use snake_case and lowercase appropriately.

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

crossed)?

Yes, the schema is relatively easy to read. You may want to move the CharacterItems block a bit, so it shows the arrows for character_id and item_id a bit cleaner.

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

Yes, intersection tables are properly formed. BundleItems, ShopItems, and CharacterItems each have two FKs and facilitate a M:N relationship.

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

No. The sample data does not suggest any non-normalized issues (neither partial nor transitive dependencies).

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 is syntactically correct.

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

Yes, all data types are appropriate given their respective descriptions of the attributes in the database outline. VARCHAR is uses for all strings, INT is used for all id values, DATE is used for the birthday attribute in NonPlayableCharacters, and TINYINT is used for a Boolean value of is_romanceable.

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

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

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

Yes, relationship tables are present in the SQL when compared to the ERD/Schema.

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

Yes, all example data is INSERTed in the SQL and matches what is given in the .pdf.

Review by Eric Meyer

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

Yes, the ERD matches the Schema and the Schema matches what is in the DB.

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?

Entities use capital letters to differentiate between words and attributes use underscores when appropriate. It is very easy to read.

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

Some relationships are crossed but I do not think it affects readability. Lines are color coded so they are easy to follow.

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

BundleItems and CharacterItems are both formed properly with two FKs each.

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

No non-normalized issues.

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

Yes, the sql imported successfully with no issues.

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

All attributes are associated with correct data types. Date variables have Dates and ids have INTs.

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

Cascades are covered in their sql file.

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

Yes the relationship tables are well defined and present.

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

Yes, all data is present and accounted for.

Review by Abdifatah Mohamed

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

diagram exactly?

Yes, pretty much! The schema is well presented, and the ERD matches the database outline.

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

Entities are plural and attributes are all lowercase & snake case. Yes, the correct use of capitalization of the entities and attributes.

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

They have an easy-to-read schema and well presented (good job) though I would agree with Hayden's point that there are some crossed lines but it won't make it difficult to read the diagram.

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

Yes, the BundleItems entity is an intersection table facilitating an M:N relationship between Bundles and Items, and it has been properly formed and Lineups that have two foreign key values.

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

No, the sample data does not suggest any non-normalized issues such as partial dependencies to transitive dependencies. I believe their sample data fulfills 1NF and 2NF.

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, .sql file imported successfully with no issues using phpMyAdmin.

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

Yes, the data types are appropriate considering the description of the attribute 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?

Yes, the primary and foreign keys are correctly defined in the SQL compared to the Schema. Yes, ON UPDATE/DELETE CASCADE operations declared in their sql file.

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

Yes, the tables present in the SQL match the ERD/Schema.

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

Yes, all the example data shown in PDF matches the data inserted in the SQL document.

Review by Hayden Meek-Avedovech

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

Yes, the schema correctly represents the ER. However, in table responsibility and in the outline, the NonPlayableCharacters table is still represented by Characters in other table relationship sections. And although the description attribute is added in the schema and ERD in all tables excluding the intersection tables, it is not present in the outline. Finally, I would suggest that the CharacterItems table be changed to NPCItems to match the new name of the NonPlayableCharacters table.

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?

Excluding what I said in the previous section, everything seems to be consistent.

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

The schema is easy to read and understand, there are a few crossed lines but they don't make this diagram any harder to read.

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

All intersection tables are correctly formatted. Additionally, each one has its own primary key which although necessary works just fine.

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

As far as I can see there are no partial or transitive dependencies in 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 a backup of your own database before you do this!)

It imported perfectly into my database so I see no issues.

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

Yes, all the data types are appropriate given the description of the attributes.

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

I believe that the primary and foreign keys are correctly defined compared to the schema, and cascade operations are declared. One personal problem, it looks like it was forward engineered from SQL workshop and is a little hard to read at a glance, but after thorough reading it makes sense.

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

Yes, the relationship tables are present when compared.

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

I believe so, yes.

Actions Based on the Feedback:

1. Include time attribute in the Shops entity - this is already reflected in the Shops table with the operating_hours attribute.
2. Include seasons attribute on Items
3. Include a description attribute on all Entity
4. Change Bundles_has_items to BundleItems to have uniform naming
5. Change Characters to NPCs to have uniformity between overview and database outline and improve clarity on entity use
6. Change attribute type from varchar to DATE on Character birthdays

Upgrades to the Draft Version:

We changed the ERD and the database outline to reflect the feedback received. This is, added seasons attribute to Items, and updated ERD to reflect. Added description attributes on entities Characters, Shops, Items, Regions, and Bundles and updated ERD to reflect changes. Changed Bundles_has_items to BundleItems in ERD. Changed Characters to NonPlayableCharacters and updated this in the ERD.

During the normalization process, we also identified a few necessary changes and updated the NPC occupation and Item seasons attributes to accept NULL values, and changed birthday on NPC to a string since they aren't real dates.

Overview: Stardew Valley is a popular farming/country life RPG. The game is decently complex and can be intimidating to new players who are unfamiliar with the game. At its core, the game is a farming and life simulation RPG, but two of the main priorities of any player is to befriend the 34 NPC characters by talking to them and giving them gifts of items available in the game, and restoring the community center with 30 "bundles" of items available in the game. There are over 500 unique items available in the game that can comprise these bundles or serve as favorite or preferred items for the NPCs. This database and web front end will allow players to quickly see which items are needed for bundles and which items are best to use as gifts in building NPC relationships.

Database Outline:

NonPlayableCharacters:
Entity record, showing details for each non-playable character in the game.

- character_id: int, auto_increment, unique, not NULL, PK
- name: varchar, unique, not NULL
- description: varchar, not NULL
- occupation: varchar, not NULL
- birthday: DATE, not NULL
- region_id: int, not NULL, FK
- is_romanceable: tinyint (boolean), not NULL
- Relationships:
    - Many-to-one relationship with Regions, implemented with the region_id, which is implemented as a FK. A character can have one region, but a region may have many characters.
    - One-to-one relationship with Shops, implemented with shop_charecter_id holding a FK. A shop can have one character as its owner, and a character can own one shop.
    - Many-to-many relationship with Items, which will be implemented with a table called CharacterItems. Each character can have many favorite items, and each item can be the favorite item of multiple characters.

Shops:
Entity record, showing details for each shop that is accessible by the player.

- shop_id: int, auto_increment, unique, not NULL, PK

- name: varchar, not NULL
- description: varchar, not NULL
- shop_character_id: int, not NULL, FK
- region_id: INT, not NULL, FK
- operating_hours: varchar, not NULL
- Relationships:
    - Many-to-one relationship with Regions, implemented with the region_id, which is implemented as a FK. A character can have one region, but a region may have many characters.
    - One-to-one relationship with Characters, implemented with shop_charecter_id holding a FK. A shop can have one character as its owner, and a character can own one shop.
    - Many-to-many relationship with Items, which will be implemented with a table called ShopItems. Each shop may carry many items, and each item may be in many shops.

Items:
Entity record, showing details for individual items in game, which can be bought and sold by the player, and also serve as “favorite items” for NPCs, which the player can gift to boost relationships, and are present in Bundles.

- item_id: int, auto_increment, not NULL, PK
- name: varchar, not NULL
- description: varchar, not NULL
- seasons: varchar, not NULL
- Relationships:
    - Many-to-many relationship with Characters, which will be implemented with a table called CharacterItems. Each character can have many favorite items, and each item can be the favorite item of multiple characters.
    - Many-to-many relationship with Shops, which will be implemented with a table called ShopItems. Each shop may carry many items, and each item may be in many shops.
    - Many-to-many relationship with Bundles, which will be implemented with a table called BundleItems. Each Bundle has multiple items, and each item may be in multiple Bundles.

Regions:
Entity record, showing details for each region within Stardew Valley and the surrounding areas that the player can access.

- region_id: int, auto_increment, unique, not NULL, PK

- name: varchar, not NULL
- description: varchar, not NULL
- Relationships:
    - One-to-many relationship with Characters, implemented by a FK in the Characters table. Each character has one region, but a region may have many characters.
    - One-to-many relationships with Shops, implemented by a FK in the Shops table. Each shop has one region, but a region may have many shops.

Bundles:
Entity record, showing details for each bundle needed to restore the in-game community center.

- bundle_id: int, auto_increment, unique, not NULL, PK
- name: varchar, not NULL
- description: varchar, not NULL
- Relationships:
    - Many-to-many relationship with Items, which will be implemented with a table called BundleItems. Each Bundle has multiple items, and each item may be in multiple Bundles.

CharacterItems:
Serves as the connection for the M:N relationship between Characters and Items.

- character_items_id: int, auto_increment, unique, not NULL, PK
- character_id: int, not NULL, FK
- item_id: int, not NULL, FK
- Relationships:
    - This table is the intersection between the many-to-many relationship of Characters and Items.

ShopItems:
Serves as the connection for the M:N relationship between Shops and Items.

- shop_items_id: int, auto_increment, unique, not NULL, PK
- shop_id: int, not NULL, FK
- item_id: int, not NULL, FK
- Relationships:
    - This table is the intersection between the many-to-many relationship of Shops and Items.

BundleItems:
Serves as the connection for the M:N relationship between Bundles and Items.

- bundle_items_id: int, auto_increment, unique, not NULL, PK
- bundle_id: int, not NULL, FK
- item_id: int, not NULL, FK
- Relationships:
    - This table is the intersection between the many-to-many relationship of Bundles and Items.

Table Responsibility:

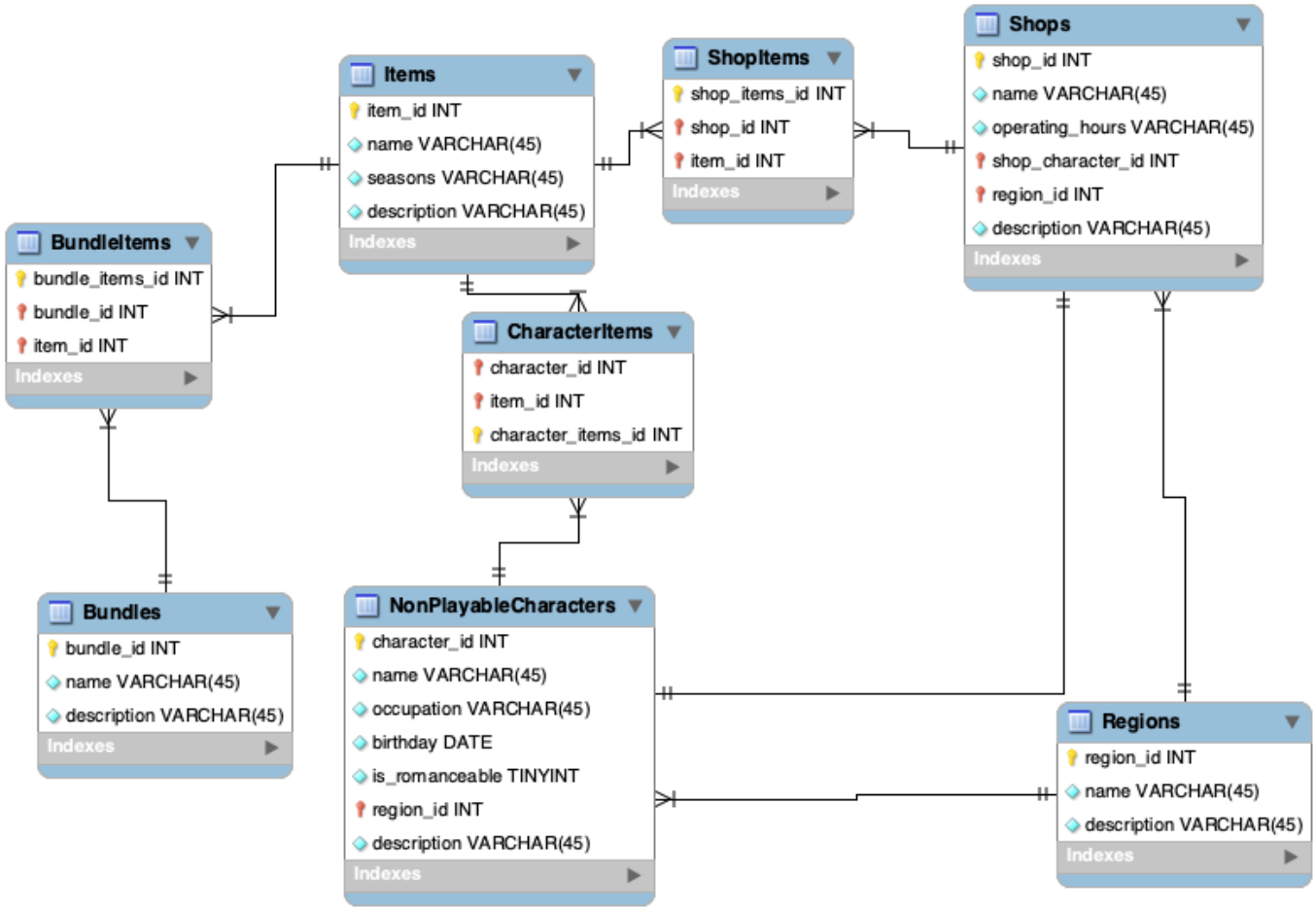
Alex:

- Characters
- Items
- ShopItems
- CharacterItems

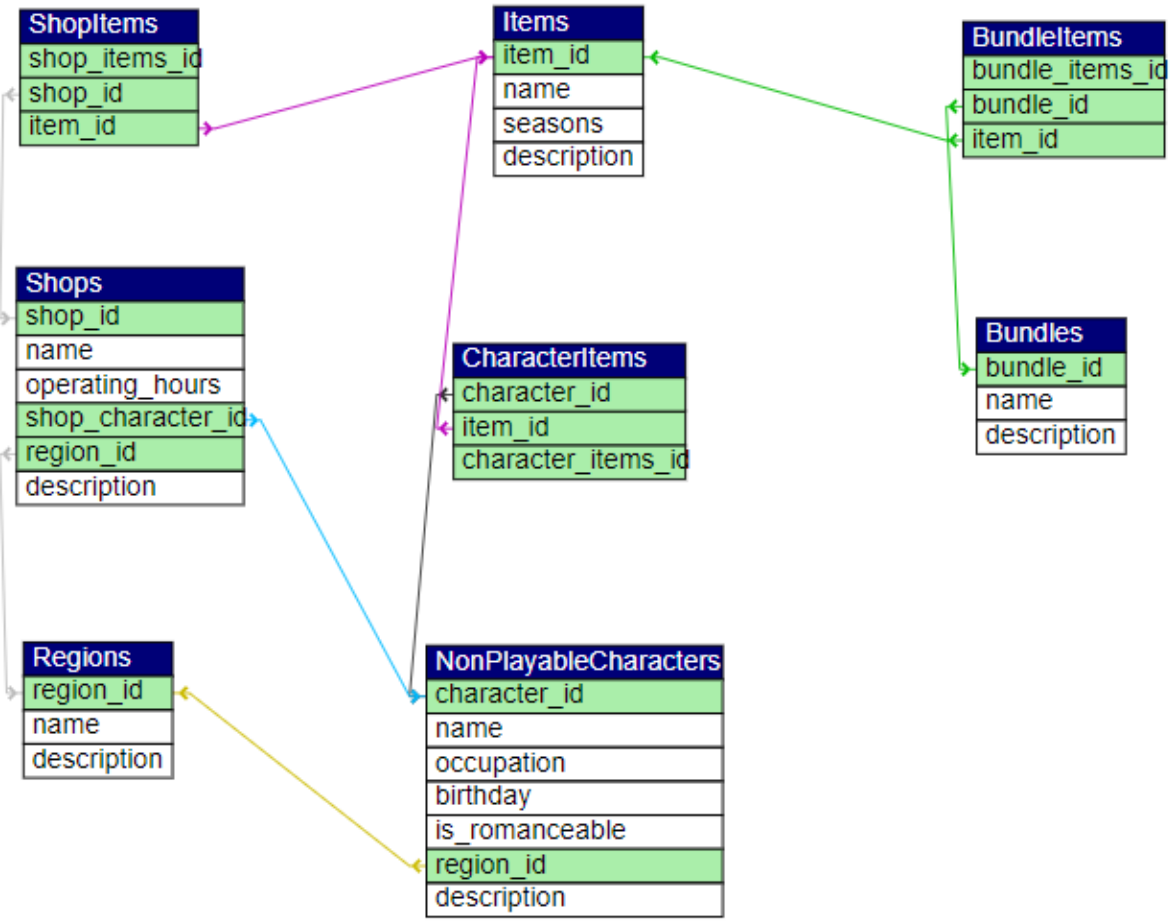
Raymond:

- Shops
- Regions
- Bundles
- BundleItems

Entity Relationship Diagram:



Schema


ShopItems
shop_items_id
shop_id
item_id
Items
item_id
name
seasons
description
BundleItems
bundle_items_id
bundle_id
item_id
Shops
shop_id
name
operating_hours
shop_character_id
region_id
description
CharacterItems
character_id
item_id
character_items_id
Bundles
bundle_id
name
description
Regions
region_id
name
description
NonPlayableCharacters
character_id
name
occupation
birthday
is_romanceable
region_id
description

Example Data

Bundles		
bundle_id	name	description
1	Spring Foraging Bundle	One of six bundles in the Craft Room.
2	Summer Foraging Bundle	One of six bundles in the Craft Room.
3	Fall Foraging Bundle	One of six bundles in the Craft Room.
4	Winter Foraging Bundle	One of six bundles in the Craft Room.

BundleItems		
bundle_items_id	bundle_id	item_id
1	1	1
2	2	2
3	3	3
4	4	4

Items			
item_id	name	seasons	description
1	Wild Horseradish	Spring	A spicy root found in the spring.
2	Grape	Summer, Fall	A sweet cluster of fruit.
3	Common Mushroom	Fall	Slightly nutty, with a good texture.
4	Crystal Fruit	Winter	A delicate fruit that pops up from the snow.
5	Kale Seeds	Spring	Plant these in the spring. Take 6 days to mature. Harvest with a Scythe.
6	Wheat Seeds	Summer, Fall	Plant these in the summer or fall. Take 4 days to mature. Harvest with a Scythe.
7	Blueberry Seeds	Summer	Plant these in the summer. Takes 13 days to mature, and continues

			to produce after first harvest.

NonPlayableCharacters						
character_id	name	occupation	birthday	is_romanceable	region_id	description
1	Krobus	Krobus' Shopkeeper	Winter 1	FALSE	1	Krobus is the only friendly monster that players encounter. He can become a roommate.
2	Vincent	NULL	Spring 10	FALSE	2	The youngest son of Jodi and Kent, brother to Sam. Is best friends with Jas.
3	Pierre	General Store Shopkeeper	Spring 26	FALSE	2	Runs the General Store in town. Is married to Caroline, and the father of Abigail.
4	Willy	Fisherman	Summer 24	FALSE	3	Willy runs the Fish Shop and spends most of his time fishing.

ShopItems		
shop_items_id	shop_id	item_id
1	1	5
2	1	6
3	1	7

Regions		
region_id	name	description
1	The Sewers	A location unlocked by obtaining the Rusty Key after donating 60 items. Home to Krobus and Krobus's Shop.
2	Pelican Town	The main playable area of the game, where the majority of the characters live, most shops are, and the players farm exists.
3	The Beach	The beach is an area south of Pelican Town and is valuable for fishing.

Shops					
shop_id	name	operating_hours	shop_character_id	region_id	description
1	Pierre's General Store	9am to 5pm	3	2	The store sells various seeds, saplings, and fertilizer. It is home to Pierre, Caroline, and Abigail.
2	Fish Shop	9am to 5pm	4	3	The fish shop sells various fishing equipment. It is home to Willy.
3	Krobus' Shop	12am to 12am	1	1	Sells various rare items as well as rotating stock.

CharacterItems		
character_items_id	character_id	item_id
1	1	1
2	2	2
3	3	3