Team Members: Zane Miller, Eemi Juntunen
Project Title: Stars Wars Encyclopedia

http://flip3.engr.oregonstate.edu:31225

1. Feedback Received After Step 1

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

    - *Yes the overview states that it will collect data about the star wars universe to create relationships as it can be difficult to navigate those relationships.*
    - *The document specifies the purpose of the database. The database will store information about Star Wars characters, ships, planets, etc., and their relationships with one another.*
    - *Yes, the overview describes using the database to record and relate characters, ships, planets, species, and media in the Star Wars universe. So the problem being solved is the problem of storing an encyclopedia of Star Wars information. However, the overview does not mention the affiliations table, so I would add that.*
    - *Yes, it describes the need for an encyclopedia of different relational entities in the Star Wars Universe.*

2. Does the overview list specific facts?

    - *No, It talks about relating data between characters, species, ships, and planets but that's as specific as it gets. If they were able to add some data about how many ships, planets, characters, etc that would be good.*
    - *The document does not list the amount of data that would be stored in the database. It is unclear how large the database is expected to be. I would recommend adding some presumed statistics to the description to make it more clear.*
    - *No, the overview does not list any specific facts. It states the Star Wars universe is "large" but does not list approximate figures for how many characters, species, ships, planets, affiliations, or media sources there are to be stored in the database. I would recommend looking at existing online wikis and encyclopedias to get an idea of the scale for each table.*
    - *Yes, it mentions specific things in the Star Wars Universe.*
    - *This could be augmented by giving some ballpark numbers of the different categories to understand what the scale of the database will be.*

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

    - *Yes there are more than four entities, and they each represent a single idea. The entities include: characters, species, planets, ships.*

- *Yes, there are seven entities in total, and each represents a single idea. For example, the planets entity represents a single planet and the characters entity represents a single character.*
- *Yes, there are 6 entities named characters, species, planets, ships, affiliations, and media, each of which represents a single idea in the Star Wars universe.*
- *Yes, the entities listed are: characters, species, planets, ships, media, affiliations*

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

    - *The group did a good job of going through and highlighting all of the relationships that each entity has, as well as their data types, and constraints. They also did a good job of explaining the relationships between entities making it very easy to follow.*
    - *Yes, the document explains the various connections between the entities and the attributes are descriptive and unique to each entity. Each relationship is well explained, and the limitations are also laid out. The attributes are descriptive and easy to understand.*

    - *Yes, each entity lists its purpose, the attributes of the entity with datatype and constraints, and the relationships between entities. However, there are some errors present between the outline and the ERD listed below.*
        a. *The outline is missing size information for varchar attributes in the characters, species, and planets tables.*
        b. *The home_planet attribute in characters is listed as NOT NULL in the ERD, but this is not specified in the outline.*
        c. *The affiliations outline has an attribute called affiliation, but it is named organization in the ERD.*
        d. *The outline does not mention the names of the foreign keys in the intersection tables for M:M relationships.*
    - *In addition to these errors I have suggested changes.*
        a. *The affiliations table only contains a primary key and one attribute at the moment, its name. This makes the entity pretty weak, easily just replaceable by another attribute in the characters table called affiliation. I suggest adding more information, such as affiliation goal, years active, major events it was involved in, leaders, etc. so that this becomes a full entity.*
    - *Yes, each entity has datatypes, constraints and relationships described.*

5. 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 appear to be correct, there is a M:M relationship between characters and affiliations, and also ships and characters. They have an ERD that shows a good logical view of the database and matches what they have written in the outline.*
    - *It appears the 1:M relationships are correctly represented in the document as well as the diagram. The keys are connected between each of the entities. The diagram does not contain an M:M relationship. The ERD does present a logical view of the*

database. I would add a M:M relationship between planets and ships perhaps. I.e. multiple planets multiple ships.

- *Yes, the 1:M relationships are correctly formulated for the interconnections between media, characters, planets, and species. There is at least one M:M relationship, with a relationship between characters and ships and a relationship between characters and affiliations. The ERD shows a logical view, but there is one minor conflict with the outline, as specified in the last section. Overall, my biggest point of suggestion would be consider whether the species/characters relationship could be M:M instead of 1:M. For example Jacen Syndulla from Star Wars Rebels is a Human/Twi'lek hybrid and there are also Human/Twi'lek hybrid characters in Star Wars: The Clone Wars named Shaeeah and Jek Lawquane. There are fewer cases of this though than being one species, so it could be kept the same. Just a suggestion.*
- *I noticed that organizations are listed as an attribute under affiliations. I think that might be a good candidate as an entity since organizations have attributes like governor, member, charter, etc. Affiliation could then just be the connection between the two entities.*

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

- *Yes there is consistency in the naming, the group uses snake case for their naming, and has lowercase for all entities.*
- *There is a unified naming scheme in the database design. However, some of the names of the entities are plural. Not sure if that is intentional. All the keys follow the same naming convention, and the names are all lowercase. I would recommend changing some of the entity names.*
- *The plurality and capitalization of entities and attributes in the outline and ERD are consistent. There is one small naming inconsistency for the affiliations table listed in one of my previous review sections, but otherwise capitalization is consistent.*
- *Yes, the entities are plural, and the attributes are singular. Underscores and lowercase letters are used consistently.*

2. Feedback Received After Step 2

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

- *Yes, it does. I was looking at Media table and thought about the example in the book where they had actors for shows. It might be worth considering changing the Media relationships so that Media is an intersection table of first shows that table could have many characters, many planets and many types.*
- *I see that you are choosing to omit all the other attributes for all the tables except the keys. I think you should modify the Character_Affiliations and Character_Ships tables to have the PKs be FKs to make it more clear that they are FKs. There cannot be two PKs in one table so the ER Diagram is not representing the tables accurately. Otherwise I think that the diagram is fine.*

- *Looks good but you might be missing the "optional" symbol for characters to ships. You note in your DB outline that a character can fly on 0 or many ships. Same note for affiliations*
- *The ER diagram has 2 PKs in the intersection tables, when it should be 2 FKs. Otherwise, everything looks good.*

2. 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?

    - *The Characters table is not consistent between the three different versions. On the ER it is Character (looks like a typo).*
    - *There are what appears to be foreign keys that could be labeled better. Example: "species" in table Characters. This looks like a foreign key reference so should be labeled as "character_species_id" or something similar.*
    - *Characters table suggestions:*
        1. *The ER Diagram should have the Character table's name be updated to Characters to make it plural.*
        2. *The species FK in the Characters table should be named species_id rather than species to one, make it singular, and to have it match attribute name in the Species table.*
        3. *The same should be done for the character_home_planet and character_first_appearance attributes as well. So, character_home_planet_id and character_first_appearance_id.*
    - *Species table suggestions:*
        1. *species_home_planet should be updated to species_home_planet_id to make it more accurate.*
    - *"Character" entity in ERD should be "Characters" based on DB outline and schema. All attributes look correct with lowercase snakecase.*
    - *In the species table, species_home_planet should be species_home_planet_id. In the characters table, character_home_planet should be character_home_planet_id. The Character entity in the ER diagram should be Characters.*

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

    - *The diagrams are laid out very well with clear lines and are easy to read.*
    - *The diagrams are somewhat clear. It would be better to use a diagram tool which shows the lines between PK and FK constraints. That way its clear which PK each FK is referencing.*

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

- *Yes - also have my suggestion (supra) for using Media as one.*
- *Yes the intersection tables are properly formed, however again, I would update the ER diagrams to say FK instead of PK.*
- *Yes the two intersection tables have to FKs. might consider adding a PK attribute.*

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

    - *If there was a normalization process it was not outlined, the current form looks pretty good to me. There's actually a lot of tables and very few attributes. I think it could be possible to intentionally de-normalize some of the tables to save space.*
    - *Looks fine to me. But I personally would combine some of the tables like ship_classification and ships.*

6. 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!)

    - *To import I had to remove references to the schema. Recommend removing these references since we are mostly using flip servers and only have access to our own schema*.

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

    - *Yes they look appropriate.*

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

    - *There are no CASCADE operations declared but there are ON DELETE actions specified which make sense to me. I think a CASCADE might be appropriate unless some of the FKs are better identified. It's a little hard to tell which are keys vs attributes currently.*

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

    - *Yes.*

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

    - *Yes.*

3. Feedback Received After Step 3

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. Every object entity and intersection table is displayed on its own page with appropriate attribute fields.*

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

    - *I'm not sure this requirement has been met. Most tables include update forms which could conceivably use SELECT criteria, depending on how the backend is implemented. However, no page includes a search function that displays a filtered subset of existing entries. I believe this is what's intended by this criterion and, if I'm correct, this needs to be added. In addition, the DML should have SELECT criteria that filter on specific attributes in the "SELECT Statements for Browse Pages" section.*

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

    - *No. Most tables do include a form for adding new entries but new relationships cannot be created as the intersection tables (Character Affiliations and Character Ships) do not have INSERT functionality. This also needs to be added to the DML.*

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

    - *No, it doesn't appear this can be done. There is no functionality for adding intersection relationships on any page and thus no way for such an update to occur.*
    - *That said, I'm not entirely clear on what this requirement means. The only way INSERTing and Order could also INSERT OrderDetails entries is if the INSERT form on the Orders page also included the ability to add the data for an arbitrary number of OrderDetails. If this is what is desired, I think the easiest way to implement it would be to add OrderDetails fields for the INSERT Order form that creates a relationship with a new*

OrderDetails entry and then have a separate INSERT OrderDetails form on the same page that allows for selecting existing Orders. I suppose this would also satisfy the previous requirement.

- *So, for your project, an example of this would be to have an optional Affiliation pulldown menu in Update Character and then a separate Add Affiliation form with pulldown menus for both Character and Affiliation. That said, It's entirely possible I'm misreading this requirement. I think it's worth getting instructor clarification.*

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

    - *This requirement is partially fulfilled. Every object entity has a delete but none of them would obviously remove other items in an M:M relationship. I also don't think you would want this to happen for the existing M:M relationships. Deleting an affiliation shouldn't delete any characters or vice versa. Likewise for ships and characters. I think the "best" way to fulfil this is to make a new intersection table where it would make sense and implement it there. However, in the interst of time, you might just want to have deleting characters delete ships or something, even if that's not what you'd really want.*

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

    - *Yes. Every object entity has an update form.*

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

    - *Yes. The sample data does show NULL fields. Previously filled fields may or may not be NULLable via the update forms. Several fields are text entry and could be submitted empty. If this would update an attribute to NULL then this requirement is met.*

8. Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

- *As of now, the update and delete forms don't actually let the user choose which entries to delete. In fact, some of the update forms' text seems to indicate the only entry that can be updated is the one with PK ID 1 (e.g., Media Type). I assume this is just placeholder stuff but it needs to be updated.*
- *I think the biggest issues are with the ability to add relationships and to add some sort of ON DELETE CASCADE functionality. Otherwise, I think it's a well designed project. Great work!*

4. Feedback Received After Step 4

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

    - *The search function appears to be working as long as you search for the attribute exactly ("Star Wars: The Clone Wars", but "Clone Wars" returns nothing)*

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

    - *Overall looks great! I would suggest adding the navbar to all pages, as it is a little jarring going to the update/delete pages and the table is suddenly at the top of the screen. It would also be cool to have the nav bar design and picture implemented on the home page rather than just the bulleted list. Also, if you have time it could be nice to add a message that appears when no search results are found. Right now you just see the Add New Planet link and a period.*

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

    - *I think my only suggestion would be to make the home page mirror the look and feel of the other pages and using your navigation you created for the other pages on the home page.*

1. Summary of Changes from Review 1

a. Updated the Overview to include numerous facts about the Star Wars Universe to better support the motivations for a Star Wars Universe database.
b. Added size constraints for varchar() data types for attributes that were missing a fixed value.
c. Removed the NOT NULL designation from Characters and Species in the ERD and Outline to reflect the fact that the home_planet of a Species or Character, may not be known.

d. Updated the naming of organization attribute in the Affiliations entity to be 'affiliation' everywhere.
e. Added to the descriptions, within the M:N relationships, clarifying information to about the PK/FK references for the intersection tables.
f. The Affiliations table is intentionally vague at the moment. As of now, it is just a category table but may be expanded in future iterations to include more attributes. The Affiliations table is represented as a M:N table since multiple characters can be affiliated with multiple affiliations. We are still in the process of deciding what more attributes to include.
g. At the moment, we are not concerned with the relationship between Planets and Ships and may not choose to include any relationship between these two entities.
h. Our M:M relationships are represented with 2 x 1:M relationships represented by the intersection tables "Character_Affiliations" and "Character_Ships" and mentioned in the outline.
i. For the moment, in the interest of simplicity, we are going to leave the Characters/Species relationship as is and limit character species to one choice or perhaps include a record for hybrid classification.
j. Replaced entity names with capitalized entity names.
k. We left Media as is because, while "Medias" is a pluralized form of "Media" so too is "Media" and we like "Media" more.
l. Updated home_planet in Characters, and Species to read character_home_planet and species_home_planet for clarity.
m. Updated first_appearance in Planets and Characters to read planet_first_appearance and character_first_appearance respectively.
n. Added planet_name to Planets.
o. Updated name in Characters, Ships, and Media to read character_name, ship_name, and media_name.
p. Added category tables Media_Types, Regions, and Ship_Types.
    i. Changed Planets.region to Planets.region_id and referenced it as a FK INT from Regions.region_id.
    ii. Changed Ships.ship_type to a INT FK reference to Ship_Classifications.ship_classification_id.
    iii. Changed Media.media_type to an INT FK that references Media_Types.media_type_id.
q. Added release_year YEAR(4) attribute to Media.

2. Response to Changes from Review 2

a. We're leaving the Media table as is for the sake of simplicity.
b. Updated the ER Diagram to correct type in Character -> Characters and changed PKs in intersection tables to FKs.
c. Updated the Schema to reflect the optional relationship between Characters and Ships and Affiliations.
d. Foreign key names are kept as is. This is a team design choice and we like species as the attribute name in Characters vs character_species_id. Same goes for all other references to this name convention.

e. No changes were made to diagrams in reference to making the key relations more clear. For example, the FK species in Characters table is marked red which already denotes it is a foreign key and obviously comes from the Species table.
f. Ships and Ship_Classifications won't be combined into one which it was originally. By normalizing our database, we realized having a category table for Ship_Classifications would reduce the chance for input and delete anomalies since names can be misspelled. Same goes for the regions and Media_Types tables.
g. Updated on DELETE clauses to reflect expected referential integrity aligned with database outline.

3. Response to Changes from Review 3

a. We have not implemented the search feature yet. The plan was to implement this in step 4 once we've narrowed down the CRUD operations for specific tables. We didn't find that implementing the search feature with no backend implementation to be a valuable use of time at this step.
b. Added insert() functionalities to the tables missing them.
c. Updated the ON DELETE and ON UPDATE clauses to delete records from intersection tables when their associated PK records are deleted. We have restriction clauses on Regions, Media_Types, and Ship_Classifications as these are going to be immutable tables.
d. All of the form functionality is simple placeholder text for now. We did not find it a valuable use of time to implement these functionalities at this time with no implemented backend.

4. Response to Changes from Review 4

a. Corrected the error with the connection dropping – we believe the db connections were timing out. We changed the app code to create and drop new connections around each query.
b. We choose not to search for partial strings as in a very large database, this can be a very time consuming operation.
c. Comments regarding UI style and interactions will be updated in the final step.
d. A fix is in the works for defaulting the planets display to show all planets if the user enters a search criterion that yields 0 results or would simply like to exit the sorted records.

5. Overview

The Star Wars Encyclopedia is a database driven website that will collect and relate data between the many characters, species, ships, planets, and canon media throughout the Star Wars Universe. The Star Wars Universe is large and finding the relationships between the different lore can be as difficult as navigating the universe.

Within the Star Wars Universe there are, by some estimates, more than 20,000 characters, more than 100 of which are important enough to be named across the different Star War's media, spread over more than 10 species, affiliated with any number of the more than 25 different gang organizations. There are over 1,100 Star Wars titles spanning a time of 36,000 years. The Star Wars universe consists of more than 1,000,000 (an arbitrarily large number because the exact number is impossible to measure) planets and moons, accessible by more than 50 space craft.

This database will start with the cataloging of named objects in Star Wars canon media with the option to expand its use to include legends media in the future. This database will also be a great companion at your next trivia night!

6. Database Outline

a. **Characters**: named characters in the Star Wars Universe
   i. character_id: INT, auto_increment, unique, NOT NULL, PK
   ii. character_name: VARCHAR(145), NOT NULL
   iii. character_home_planet: INT, FK
   iv. character_first_appearance: INT, FK
   v. species: INT, NOT NULL, FK
   vi. birth_year: VARCHAR(145)
   vii. **Relationship**(s):
      1. There is a M:N. relationship between ships and characters represented by the intersection table "CharacterShips" referencing character_ID and ship_ID PKs as foreign keys. Each ship may have one or many characters, and each character may fly on 0 or many ships.
      2. There is a M:N relationship between characters and affiliations represented by the intersection table "Character_Affiliations" referencing character_id and affiliation_id PKs as foreign keys. Each character may be affiliated with 0 or more organizations, and each organization may have many affiliates (characters).
      3. There is a M:1 relationship between characters and media. Each character first appears in one media (film, show, comic, etc.) and each media may have multiple characters in it making their first appearance in the Star Wars Universe. This relationship is implemented with *character_first_appearance* as a FK in characters which is the PK media_id in media.
      4. There is a M:1 relationship between characters and species implemented using species as a FK in characters which is the species_id in species. Each character can belong to one species, but a species might have many different characters.
      5. There is a 1:M relationship between planets and characters implemented using character_home_planet as a FK in characters which is the PK planet_id in planets. Each character

has exactly one home planet, but a planet could be home to many characters.

b. **Species:** A category table of the different species in the Star Wars Universe
    i. species_id: INT AUTO_INCREMENT NOT NULL, PK
    ii. classification: VARCHAR(145) NOT NULL
    iii. species_home_planet: INT, FK
    iv. **Relationship**(s):
        1. There is a 1:M relationship between species and characters implemented using species as a FK in characters which is the species_id in species. Each character can belong to one species, but a species might have many different characters.
        2. There is a M:1 relationship between species and planets implemented using species_home_planet as a FK in species which is the PK planet_id in planets. Each species originates from one planet, but a planet could be the planet of origin for many species.

c. **Planets:** Records the characteristics about a planet in the Star Wars Universe
    i. planet_id: INT, auto_increment, unique, NOT NULL, PK
    ii. planet_name VARCHAR(145) NOT NULL
    iii. region_id: INT, NOT NULL FK
    iv. description: TEXT(65535)
    v. planet_first_appearance: INT, FK
    vi. **Relationship**(s):
        1. There is a 1:M relationship between planets and species implemented using species_home_planet as a FK in species which is the PK planet_id in planets. Each species originates from one planet, but a planet could be the planet of origin for many species.
        2. There is a 1:M relationship between planets and characters implemented using character_home_planet as a FK in characters which is the PK planet_id in planets. Each planet can be home to multiple characters, but a character can have only one home planet.
        3. There is a 1:M relationship between media and planets using planet_first_appearance as a FK in planets which is the PK, media_id, in media. Each planet first appears in one media, but a single media may first introduce many planets.
        4. There is a 1:M relationship between Regions and Planets. Each region can be home to many planets, but each planet is from one region.
        5. There is a M:1 relationship between Planets and Regions with region_id as a foreign key in Planets. Each Region is home to many planets

d. **Regions**: Records the different regions in the Galaxy – Category Table
    i. region_id: INT, AUTO_INCREMENT, NOT NULL, PK
    ii. region_name: VARCHAR(145) NOT NULL
    iii. **Relationship(s):**
        1. There is a 1:M relationship between Regions and Planets. Each region can be home to many planets, but each planet is from one region.

e. **Ships** : Records the details of Ships in the Star Wars Universe
    i. ship_id: int, auto_increment, NOT NULL, PK
    ii. ship_name: varchar(145), NOT NULL
    iii. ship_type: INT, NOT NULL FK
    iv. **Relationship(s):**
        1. There is a M:N relationship between ships and characters represented by the intersection table "Character_Ships" referencing character_ID and ship_ID PKs as foreign keys. Each ship may have more than one character associated with it, and each character can have more than one ship associated with it.
        2. There is a 1:M relationship between Ship_Classifications and Ships with ship_type as a Foreign Key in Ships that references the ship_classification_id PK in Ship_Classifications. Each ship has one classification, but a ship classification can describe many different ships.

f. **Ship_Classifications**: Category table that describes the type of ship
    i. ship_classification_id: INT, NOT NULL, PK
    ii. ship_type: VARCHAR(145), NOT NULL
    iii. **Relationship(s)**
        1. There is a 1:M relationship between Ship_Classifications and Ships with ship_type as a Foreign Key in Ships that references the ship_classification_id PK in Ship_Classifications. Each ship has one classification, but a ship classification can describe many different ships.

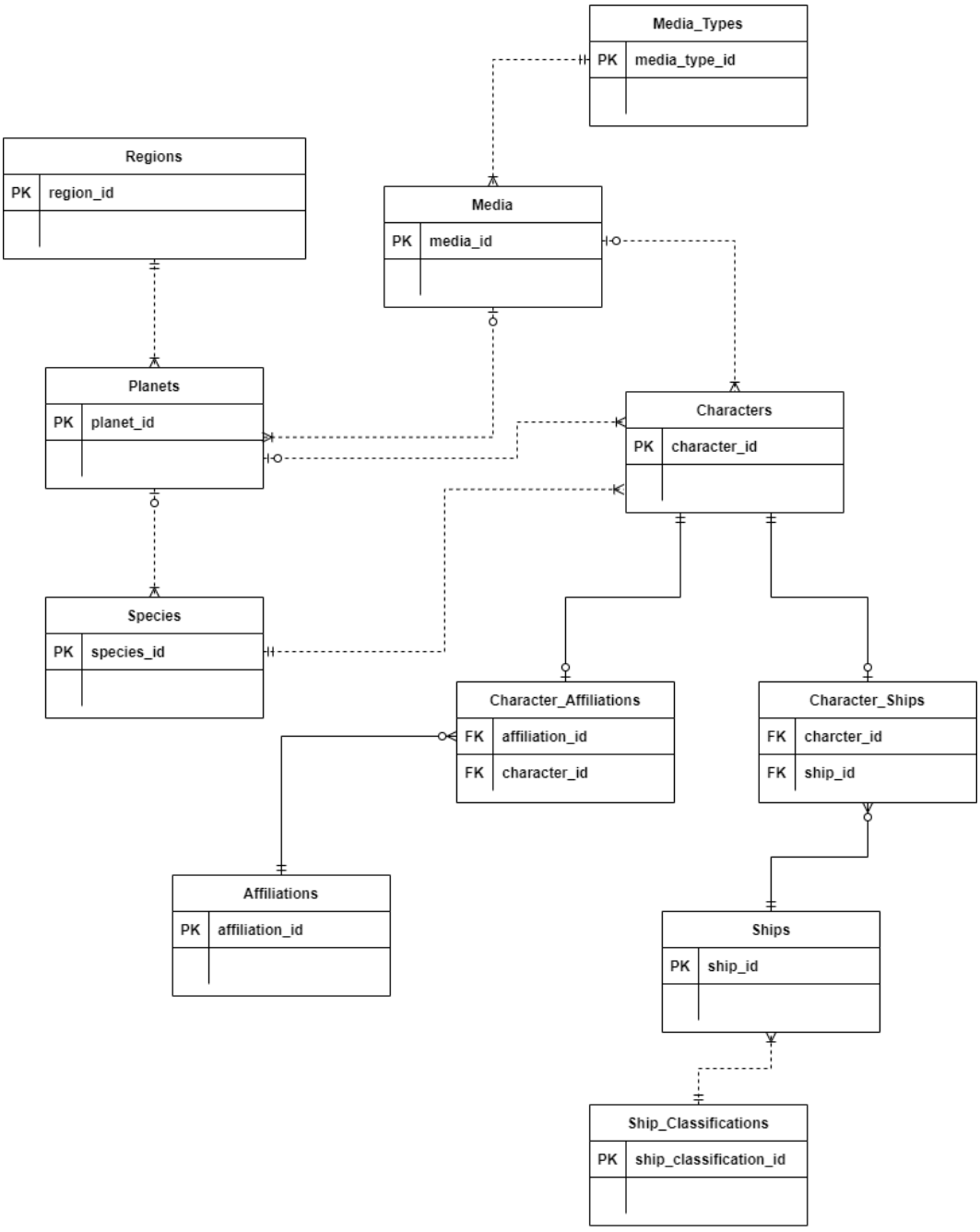
g. **Media**: Media titles in the canon Star Wars universe
    i. media_id: int, auto_increment, NOT NULL, PK
    ii. media_name: varchar(145), NOT NULL
    iii. media_type_id: INT NOT NULL FK
    iv. release_year YEAR(4), NOT NULL
    v. **Relationship(s):**
        1. There is a 1:M relationship between characters and media, with the PK media_id being a FK first_apperance in characters. A character has 1 media as their first appearance, but a media can have many characters as their first appearance media.
        2. There is a 1:M relationship between planets and media, with the PK media_id being a FK first_apperance in characters. A

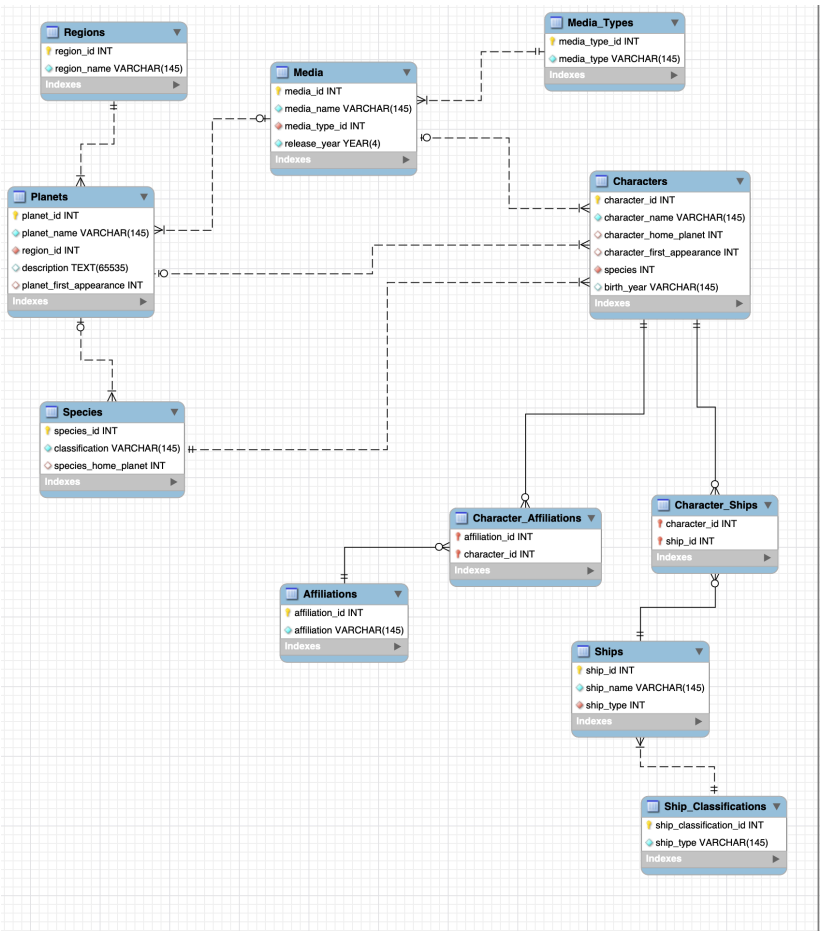
planet has 1 media where they first appeared, but a media can support many planets first appearance.
3. There is a 1:M relationship between Media_Types and Media where each Media has one type but a type might describe multiple media using FK media_type_id in Media that references the PK by the same name in Media_Types.

h. **Media_Types**: Classifies Media by type (Movie, book, comic, etc)
   i. media_type_id: INT, NOT NULL, AUTO_INCREMENT, PK
   ii. media_type: VARCHAR(145) NOT NULL
   iii. **Relationship(s):**
      1. There is a 1:M relationship between Media_Types and Media where each Media has one type but a type might describe multiple media using FK media_type_id in Media that references the PK by the same name in Media_Types.

i. **Affiliations**: affiliations a character is associated with
   i. affiliation_id: int, auto_increment, unique, NOT NULL, PK
   ii. affiliation: varchar(145), NOT NULL
   iii. **Relationship(s):**
      1. There is a M:N relationship between characters and affiliations represented by the intersection table "Character_Affiliations" referencing character_ID and affiliation_ID PKs as foreign keys. Each organization may have more than one affiliation, and each character can be affiliated with 0 or more organizations.

7. ER Diagram and Schema


Media_Types
PK
media_type_id
Regions
PK
region_id
Media
PK
media_id
Planets
PK
planet_id
Characters
PK
character_id
Species
PK
species_id
Character_Affiliations
FK
affiliation_id
FK
character_id
Character_Ships
FK
charcter_id
FK
ship_id
Affiliations
PK
affiliation_id
Ships
PK
ship_id
Ship_Classifications
PK
ship_classification_id



8. Example Data

a. Affiliations

```
MariaDB [star_wars_database_cs340]> select * from Affiliations;
+----------------+-----------------+
| affiliation_id | affiliation     |
+----------------+-----------------+
|              1 | Rebel Alliance  |
|              2 | Galatic Empire  |
|              3 | Jedi Order      |
|              4 | Hutt Cartel     |
+----------------+-----------------+
4 rows in set (0.000 sec)
```

b. Character_Affiliations

```
MariaDB [star_wars_database_cs340]> select * from Character_Affiliations;
+----------------+--------------+
| affiliation_id | character_id |
+----------------+--------------+
|              1 |            1 |
|              1 |            3 |
|              3 |            1 |
+----------------+--------------+
3 rows in set (0.000 sec)
```

c. Character_Ships

```
MariaDB [star_wars_database_cs340]> select * from Character_Ships;
+--------------+---------+
| character_id | ship_id |
+--------------+---------+
|            1 |       1 |
|            1 |       2 |
|            1 |       3 |
|            3 |       1 |
+--------------+---------+
4 rows in set (0.000 sec)
```

d. Characters

```
MariaDB [star_wars_database_cs340]> select * from Characters;
+--------------+----------------+----------------------+-----------------------------+---------+------------+
| character_id | character_name | character_home_planet | character_first_appearance | species | birth_year |
+--------------+----------------+----------------------+-----------------------------+---------+------------+
|            1 | Luke Skywalker |                    1 |                           1 |       1 | 19BBY      |
|            2 | Owen Lars      |                    1 |                           1 |       1 | NULL       |
|            3 | Chewbacca      |                 NULL |                        NULL |       2 | 200BBY     |
+--------------+----------------+----------------------+-----------------------------+---------+------------+
3 rows in set (0.000 sec)
```

e. Media

```
MariaDB [star_wars_database_cs340]> select * from Media;
+----------+-----------------------------------------------------+---------------+--------------+
| media_id | media_name                                          | media_type_id | release_year |
+----------+-----------------------------------------------------+---------------+--------------+
|        1 | Star Wars Episode IV: A New Hope                    |             1 |         1977 |
|        2 | The Mandalorian                                     |             2 |         2019 |
|        3 | Darth Vader #3                                      |             4 |         2015 |
|        4 | Star Wars: From the Adventures of Luke Skywalker    |             3 |         1976 |
|        5 | Star Wars: The Clone Wars                           |             2 |         2008 |
|        6 | Star Wars Episode VI: Return of the Jedi, Special Edition |       1 |         1997 |
+----------+-----------------------------------------------------+---------------+--------------+
6 rows in set (0.000 sec)
```

f. Media_Types

```
MariaDB [star_wars_database_cs340]> select * from Media_Types;
+---------------+-------------+
| media_type_id | media_type  |
+---------------+-------------+
|             1 | Movie       |
|             2 | TV Series   |
|             3 | Novel       |
|             4 | Comic       |
|             5 | Video Game  |
+---------------+-------------+
5 rows in set (0.000 sec)
```

g. Planets

Planets | Enter a SQL expression to filter results (use Ctrl+Space)

	planet_id	planet_name	region_id	description	planet_first_appearar
1	1	Tatooine	6	Tatooine was a sparsely inhabited circumbinary desert planet located in the	4
2	2	Mandalore	6	Outer Rim planet that is the homeworld of the Mandalorians torn by wars be	5
3	3	Nevarro	6	[NULL]	2
4	4	Coruscant	6	Coruscant (pronounced /ˈkɔɹəsənt/), also known as Imperial Center during t	6

h. Regions

```
MariaDB [star_wars_database_cs340]> select * from Regions;
+-------------+-------------------------+
| region_id | region_name             |
+-------------+-------------------------+
|         1 | Deep Core               |
|         2 | Core Worlds             |
|         3 | Colonies                |
|         4 | Expasion Region         |
|         5 | Mid Rim                 |
|         6 | Outer Rim Territories   |
|         7 | Wild Space              |
|         8 | Unknown Regions         |
+-------------+-------------------------+
8 rows in set (0.000 sec)
```

i. Ship_Classifications

```
MariaDB [star_wars_database_cs340]> select * from Ship_classifications;
+-------------------------+-------------------------------+
| ship_classification_id | ship_type                     |
+-------------------------+-------------------------------+
|                      1 | Corellian Light Freighter     |
|                      2 | T-65B X-Wing Starfighter      |
|                      3 | J-Type 327 Nubian Starship    |
+-------------------------+-------------------------------+
3 rows in set (0.000 sec)
```

j. Ships

```
MariaDB [star_wars_database_cs340]> select * from Ships;
+---------+--------------------+------------+
| ship_id | ship_name          | ship_type  |
+---------+--------------------+------------+
|       1 | Millenium Falcon   |          1 |
|       2 | Red Five           |          2 |
|       3 | Breha              |          3 |
|       4 | Red Three          |          2 |
+---------+--------------------+------------+
4 rows in set (0.000 sec)
```

k. Species

```
MariaDB [star_wars_database_cs340]> select * from Species;
+------------+----------------+---------------------+
| species_id | classification | species_home_planet |
+------------+----------------+---------------------+
|          1 | Human          |                   4 |
|          2 | Wookie         |                NULL |
|          3 | Droid          |                   4 |
+------------+----------------+---------------------+
3 rows in set (0.000 sec)
```