http://flip1.engr.oregonstate.edu:9533/

Andrew Eppinger, Joshua Luo
Project Step 5 Draft Version: Implement UPDATE and DELETE operations
CS 340 Winter 2021

Project Outline and Database Outline, ERD and Schema Updated Version

Overview

Problem: Thousands of new video game titles are released every year, adding to the already countless number of past titles. It can be difficult for a person to keep track of all the releases to make informed decisions regarding what titles to play in their limited amount of time.

Solution: A video game database that can sort through a catalog of several hundred top video game titles to determine likely candidates that the user would enjoy playing. The video game titles can be sorted based on its title, platform availability, genre, ESRB rating, franchise, and development studio.

Outline

- VideoGameTitles: records details of Video Game Titles
    - Implementation lead team member: Andrew Eppinger
    - titleID: int, auto_increment, unique, not NULL, PK
    - titleName: varchar, not NULL
    - titleESRB: char, DEFAULT NULL
        - Uses a simplified one letter symbol (the big one on the ESRB icon)
    - titleGenre: varchar, DEFAULT NULL
        - Titles may be multiple genres, but the value will be only the main genre
    - titleRelease: date, not NULL
        - Titles may release on different dates in different regions, all dates provided are the North American region release date
    - titleDeveloperID: int, not NULL, FK
        - Titles may have collaborating developers, but the value will be only the main developer
    - titleFranchiseID: int, FK
        - Titles may have characters from multiple franchises, but the value will be only the franchise that the title itself belongs to

- DevelopmentStudios: records Development Studios of video games
    - Implementation lead team member: Joshua Luo

    - developerID: int, auto_increment, unique, not NULL, PK
    - developerName: varchar, not NULL
    - developerCountry: varchar, not NULL
        - Developers may be located in multiple countries, but the value will be the original and main developer location
    - developerFounded: date, not NULL

    - Relationship: 1:M relationship between DevelopmentStudios and VideoGameTitles, with developerID as a FK inside of VideoGameTitles

- Platforms: records Platforms video games are released on
    - Implementation lead team member: Andrew Eppinger
    - platformID: int, auto_increment, unique, not NULL, PK
    - platformName: varchar, not NULL
    - platformRelease: date, not NULL
        - Platforms can release on different dates in different regions, all dates provided are the North American region release date
    - platformDeveloper: varchar, not NULL
    - platformInProduction: bool, not NULL

- TitlesPlatforms: intersection table for VideoGameTitles and Platforms
    - Implementation lead team member: Andrew Eppinger
    - titleID: int, not NULL, FK
    - platformID: int, not NULL, FK
    - titleID and platformID form a primary key when used together.

    - Relationship: M:M relationship between VideoGameTitles and Platforms, with titleID as a FK and platformID as a FK

- Franchises: records the franchise that the game is associated with
    - Implementation lead team member: Joshua Luo
    - franchiseID: int, auto_increment, unique, not NULL, PK
    - franchiseName: varchar, not NULL
    - franchiseDeveloper: varchar, not NULL

    - Relationships: 1:M relationship between Franchises and VideoGameTitles, with franchiseID as a FK inside of VideoGameTitles

- PlatformsFranchises: intersection table for Platforms and Franchises (optional M:M implementation)

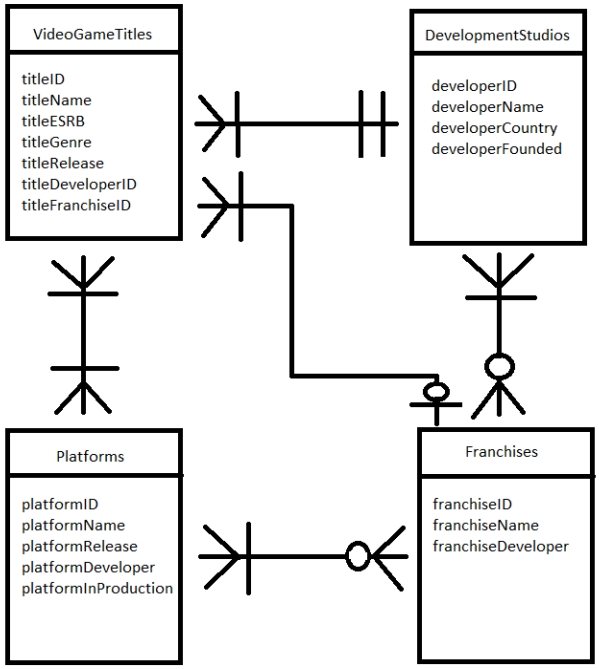
    - platformID: int, not NULL, FK
    - franchiseID: int, not NULL, FK
    - platformID and franchiseID form a PK when used together

    - Relationship: M:M relationship between Platforms and Franchises, with platformID as a FK and franchiseID as a FK

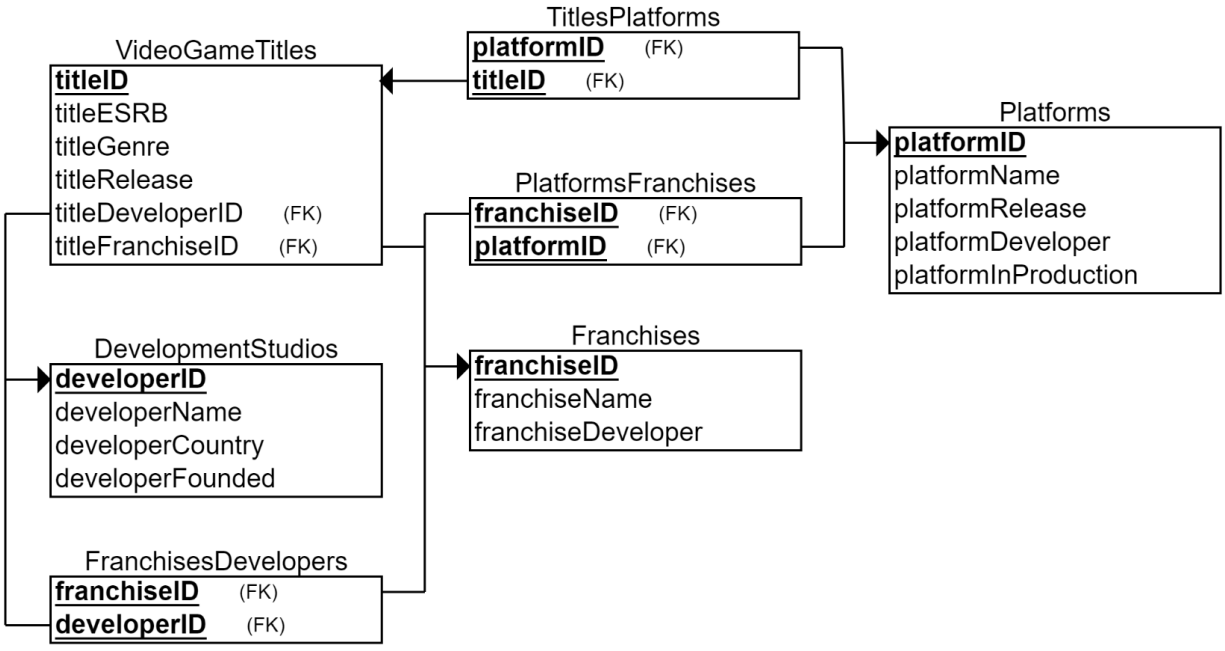
- FranchisesDevelopers: intersection table for Franchises and DeveloperStudios (optional M:M implementation)
    - franchiseID: int, not NULL, FK
    - developerID: int, not NULL, FK
    - franchiseID and developer ID form a PK when used together

    - Relationship: M:M relationship between Franchises and DeveloperStudios, with franchiseID as a FK and developerID as a FK

Entity-Relationship Diagram:



Schema:



b) Fixes based on Feedback from Previous Steps:

- Step 1 Grader Feedback
    - Include "FK" next to attributes, similar to how IDs are listed as "PK". "FK" is now included on entity VideoGameTitles's titleDeveloperID and titleFranchiseID attributes; entity Titles_Platforms's titleID and platformID attributes; and entity Platforms_Franchises's platformID and franchiseID attributes
    - Make a relationship between Franchises and DeveloperStudios entities, because Franchises has a franchiseDeveloper attribute. We added a Franchises_Developers intersection table in the outline with an M:M relationship, where franchiseID and developerID are FKs to show the potential implementation.
    - Indicate that the two FKs in the intersection tables form a PK. We added a note in each intersection table that the two FKs form a PK

- Step 2 Draft: Peer Reviewer Feedback
    - All of the reviewers mentioned that we were missing an overview section. We had thought the instructions were to only provide the outline portion of the draft, due to no explicit reference to "overview". The Overview section has been provided.
    - Nicholas Matsumoto suggested making the relationship between VideoGameTitles and Developers M:M, however we are maintaining a 1:M relationship as it is often the case that collaborating developers are only a supporting role or may be merely licensing intellectual property (e.g. playable characters from other video games).
    - Rhea Sublett mentioned that we do not have team members assigned to two of the M:M tables that are listed as optional. These will be assigned if, during the course of the project, we decide that they should be implemented.
    - We applied the feedback from Kuljot Biring regarding the names of the intersection tables being consistent with other tables, and made them camel case.

- Step 2 Final: Grader Feedback
    - We did not receive any additional feedback for our final version.

- Step 3 Draft: Peer Reviewer Feedback
    - We added a developer filter option on the Catalog page, as per Anna K. Ceneviz's feedback. This makes the filter options better match the attributes of the Video Game Titles table.
    - Christopher Perdriau noted an overlay bug on the "Delete Something" and "Update/Edit Something" pages. Christopher did not describe the bug, but we

saw that the release date range of these pages were missing the "From:" text, which is present in the "Add Something" page. They have correctly added. Christopher also suggested combining the "Add Something" and "Delete Something" to the "Update/Edit Something" Page, as adding and deleting are updates/edits to the database. While this is technically true, these actions are distinct operations to the database, and are therefore implemented separately so that a user does not accidentally perform an unintended operation if they were all on the same page.
- Hyun Kim and Ahmad Al-Qadi recommended additional spacing and css elements. As this is the first implementation of the UI, we were focused more on making sure all of the database operations are present, with only small consideration for presentation style. But, of course, ease-of-use is important and we will take their suggestions into consideration as we continue to develop our project.

- Step 3 Final: Grader Feedback
  - We did not receive any additional feedback for our final version.

- Step 4 Draft: Peer Reviewer Feedback
  - David Kaff noted that the DMQ queries do not explicitly show some kind of default values for empty search parameters, which would make sense to search through all values of the attribute. When we build the queries, we will check for empty parameters (""), in which case the backend will add or not add the appropriate query parts.
  - David Kaff also did not see a clear nullable relationship and removal of an M:M relationship without removing relevant rows. He suggested making the "UPDATE DevelopmentStudios' query to pass NULL to a parameter which will cascade to VideoGameTitles, and had no suggestion for removing an M:M relationship. The 1:M relationship between franchises and video games is nullable, as video games do not have to belong to a franchise, and thus may have null as their franchise attribute.
  - Christina Wilson, Logan Cope, and Cassandra Olsen all noted that there is no apparent way to remove our M:M relationship of TitlesPlatforms. While we have not directly looked at this, it is intended that deleting either a Title or Platform will cascade the delete to TitlesPlatforms, removing the relationship.

- Step 4: Grader Feedback
  - Gabrielle explained how that when using the DATE_FORMAT function in Flask, the % symbol is also the escape symbol. So it needed double % symbols in the query string.

- Step 5 Draft: Peer Reviewer Feedback
    - Arjay Lawrence Hamilton found that when filteringTitles with multiple fields filled in, the resulting table would not update appropriately. We found that when building the search query, we were adding the platformID beginning with the "WHERE" clause, rather than checking for if it should be an "AND". So when filtering by platform and name together, the query would have to "WHERE" clauses.
    - Zhixiang Ye had a suggestion on formatting the date, but this issue was solved by the Step 4 Grader.
    - Joseph Stumpf noticed some display bugs when updating elements. However this was simply an unintentional bug for Step 5 from our continuing work on updating for Step 6. He also noticed that the date strings were changing.
    - Kat Kime also found the Title filter bug from above, which looks like happened when filtering by name and platform, which we have hopefully resolved.

- Changes based on our own design decisions:
    - The Overview section has been reorganized to clarify the problem and solution, and some details regarding the scope of the proposed project.
    - Updated the names of the intersection tables to be in camel case, matching the convention used for the rest of the tables.
    - We added a text input filter option on the Catalog page for the Title attribute, as it is probably the most practical filter option a user would use when finding a video game title.
    - We added a function to the add/delete/update successful message to disappear after a time so that it can reappear to notify of successive actions. Just staying visible on the page with another action may be confusing to the user regarding whether the action was actually successful or not.
    - We made the search results tables on the Delete and Update/Edit pages hidden until the search button appears, indicating a search query is actually made.
    - We added a confirmation alert to delete elements from a table to ensure elements won't be easily deleted on accident.
    - We adjusted the values given from the page's input fields to be the ID values instead of the name where applicable so that we don't need to write subqueries to get the ID from the name.
    - We added an ORDER BY keyword at the end of each search query so that the webpage displays search results in alphabetical order by name.
    - When updating VideoGameTitles, some attributes may be emptied. We now have checks for empty fields, which will update the attribute to NULL, rather than

an empty string. This is particularly important for the FranchiseID field, because it is an FK, which must have the ID of a franchise or NULL, and can't be empty ("").