**IS 6503 Term Project Deliverable 3**

Your Name: Joseph Mendez

Your email: [joseph.mendez@my.utsa.edu](mailto:joseph.mendez@my.utsa.edu)

1. **Description of the business context and related data management problem(s)**

Vidia, a US-based tech company, has been in the business of movie rentals for many decades. After successfully overcoming the challenge of transitioning to subscription-based streaming, they are seeking to branch out towards including a subscription service for games; after seeing the success of other services such as the Xbox Game Pass. Unlike other competitors on the market, Vidia will be able to host games regardless of platform exclusivity. This will lead to there being many games stored on their servers, and a well as a need for various search queries for subscribers to use.

After an investors meeting, it was decided that if the service does well, they’ll work towards signing more deals with other developers to keep adding to their catalog. During said meeting, a couple of questions were asked by investors as to how Vidia’s services will run. One of these issues was how Vidia was planning on managing their new game catalog along with their pre-existing movie catalog. Another investor worried about the large amount of data and of possible discrepancies that could arise due to mismanagement of Vidia’s database.

The purpose for this Term Project proposal is to create a database application that will help Vidia keep track of their increasingly large catalog to avoid discrepancies. The proposed database should also be flexible enough to be able to integrate any new titles to both their catalogs that could possibly be added in the future.

1. **The entities and the attributes**

**ENTITY 1: Game**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The GAME entity records all the games available on the service. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Game\_ID | 6-digit numeric ID for each game | PK 1 | CHAR(6) |
| Game\_Title | The game’s title | Not Null | VARCHAR(20) |
| Game\_ReleaseYear | Year in which the game was released |  | CHAR(4) |
| Game\_Genre | The primary genre the games falls under |  | VARCHAR(7) |
| Rating | Age rating for the game | Not Null | VARCHAR(10) |
| DeveloperID | 6-digit numeric ID for each developer | FK 1, Not Null | CHAR(6) |

**ENTITY 2: Developer**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The DEVELOPER entity keeps track of all the developers who have games available on the service. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Developer\_ID | 6-digit numeric ID for each developer | PK 1 | CHAR(6) |
| Developer\_Name | The name for the developer’s company | Not Null | VARCHAR(20) |
| Parent\_Company | 6-digit numeric ID for any potential parent company tied to the developers | FK 1 | CHAR(6) |
| Country | The country where the developers are based in |  | CHAR(20) |

**ENTITY 3: Tier**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The TIER entity records all the available tiers on the service. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Tier\_ID | 6-digit numeric ID for each tier | PK 1 | CHAR(6) |
| Tier\_Name | Title for the tier | Not Null | VARCHAR(20) |
| Movie\_Tier | One of the two subtype discriminators; denotes when a tier has movies | Not Null, Default = No | BIT |
| Game\_Tier | One of the two subtype discriminators; denotes when a tier has games | Not Null, Default = No | BIT |
| Tier\_Price | The price that is tied to each tier | Not Null | NUMERIC(3,1) |
| Tier\_Lead | 6-digit numeric ID for each tier lead | Not Null | CHAR(6) |

**ENTITY 4: Game\_Tier (Associative entity between GAME and TIER)**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The GAME\_TIER entity keeps of what tiers the games in GAME are available in. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Game\_ID | 6-digit numeric ID for each game | PK 1, FK 1 | CHAR(6) |
| Tier\_ID | 6-digit numeric ID for each tier | PK 2, FK 2 | CHAR(6) |
| Expiration\_Date | Date when the game should become unavailable | Not Null | VARCHAR(11) |

**ENTITY 5: Movie\_Tier (Associative entity between MOVIE and TIER)**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The MOVIE\_TIER entity keeps track of what movies are available depending on the tier. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Movie\_ID | 6-digit numeric ID for each movie | PK 1, FK 1 | CHAR(6) |
| Tier\_ID | 6-digit numeric ID for each tier | PK 2, FK 2 | CHAR(6) |
| Expiration\_Date | Date when the movie should become unavailable | Not Null | VARCHAR(11) |

**ENTITY 6: Movie**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The MOVIE entity tracks what movies are available as part of the subscription. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Movie\_ID | 6-digit numeric ID for each movie | PK 1 | CHAR(6) |
| Movie\_Title | Theatrical title for the movie | Not Null | VARCHAR(20) |
| Full\_Title | Full title for a movie (if necessary) |  | VARCHAR(40) |
| Movie\_ReleaseYear | Year of release for a movie |  | CHAR(4) |
| Movie\_Genre | Movie genre |  | VARCHAR(10) |
| Movie\_Duration | Movie’s duration |  | VARCHAR(20) |
| Director\_ID | 6-digit numeric ID for each director | FK 1 | CHAR(6) |
| MPA\_Rating | Rating for the movie set by the MPA | Not Null | VARCHAR(6) |
| Country | Movie’s country of origin |  | VARCHAR(20) |

**ENTITY 7: Director**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The DIRECTOR entity keeps track of all the directors who have a movie hosted by the streaming service. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Director\_ID | 6-digit numeric ID for each director | PK 1 | CHAR(6) |
| F\_Name | First name of the director | Not Null | VARCHAR(15) |
| L\_Name | Last name of the director |  | VARCHAR(15) |
| Nationality | A director’s nation of birth |  | VARCHAR(20) |

**ENTITY 8: Parent\_Company**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The COMPANY entity keeps track of any parent companies with developers working under them. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Company\_ID | 6-digit numeric ID for each parent company | PK 1 | CHAR(6) |
| Company\_Name | Name for the parent company | Not Null | VARCHAR(12) |
| Country | Country where the company is based in |  | VARCHAR(20) |

**ENTITY 9: DLC**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The DLC entity keeps track of all the DLC available on the service. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| DLC\_ID | 6-digit numeric ID for each individual dlc | PK 1 | CHAR(6) |
| DLC\_Name | Name for the dlc | Not Null | VARCHAR(20) |
| Game\_ID | 6-digit numeric ID for each game who has DLC | FK 1, Not Null | CHAR(6) |

**ENTITY 10: Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The EMPLOYEE entity keeps track of all the employees at Vidia. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Employee ID | 6-digitl numeric ID for each individual employee | PK 1 | CHAR(6) |
| Emp\_FName | An employee’s first name | Not Null | VARCHAR(15) |
| Emp\_LName | An employee’s last name |  | VARCHAR(15) |
| Supervisor | 6-digit numeric ID for each individual supervisor |  | CHAR(6) |
| Department | Department the employee is a part of | Not Null | VARCHAR(30) |
| Age | Employee’s age |  | CHAR(2) |
| Gender | Employee’s gender |  | CHAR(7) |

**ENTITY 11: Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| Description: The CUSTOMER entity keeps track of all Vidia’s customers. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| Customer\_ID | 6-digit numeric ID for each individual customer | PK 1 | CHAR(6) |
| Cust\_FName | A customer’s first name | Not Null | VARCHAR(15) |
| Cust\_LName | A customer’s last name |  | VARCHAR(15) |
| Gender | Customer’s gender |  | VARCHAR(12) |
| Age | Customer’s age | Not Null | CHAR(2) |
| Country | A customer’s country of residence | Not Null | VARCHAR(20) |
| Email | Email customer uses for their account | Not Null | VARCHAR(25) |
| Tier\_ID | 6-digit numeric ID for each tier | FK 1 | CHAR(6) |

1. **ERD**

**A screenshot of a computer program

Description automatically generated with medium confidence**

1. **Business rules that define all the relationships among entities, the constraints (if any).**

***Relationships:***

**1:1 relationship:**

* **Developer 🡨🡪 Parent\_Company**: A developer may or may not work for a parent company. However, if true, they can work for one company only (0, 1). On the other hand, a parent company must have one and only one developer (1, 1).

**M:N relationship:**

* **Game 🡨🡪 Game\_Tier 🡨🡪Tier**: A game must appear in one or more tiers (1, M) while a tier can contain one or more games (1, M). To break this relationship, we use the Game\_Tier entity. In this case, the Game entity and Game\_Tier entity share a 1:M relationship while the Tier and Game\_Tier entities share a 1:M relationship.
* **Movie 🡨🡪 Movie\_Tier 🡨🡪 Tier:** On a similar note, there can be multiple movies in a tier (1, M) while a movie can appear in multiple tiers (1, M). To break this, the Movie\_Tier is introduced. The Movie entity shares a 1:M relationship with Movie\_Tier while the Tier entity shares a 1:M relationship with Movie\_Tier.

**1:M relationship:**

* **Game 🡨🡪 Developer**: A game must have one, and only one, developer (1, 1). In turn, a developer can develop multiple games; and must develop at least one game (1, M).
* **Game 🡨🡪 DLC**: A game doesn’t have to have DLC, but if it does it can have more than one piece of DLC (0, M). Any individual piece of DLC, however, can only belong to one game (1, 1).
* **Tier 🡨🡪 Employee**: A tier must be overseen by one lead employee (1, 1). An employee may oversee multiple tiers, or may not oversee any at all (0, M).
* **Tier 🡨🡪 Customer**: A tier can have multiple subscribers/customers (1, M). A customer, however, can only be subscribed to one tier (1, 1).
* **Movie 🡨🡪 Director:** A movie can be directed by one and only one director (1, 1). On the other hand, one director can direct one or more movies (1, M).

**1:M Unary relationship:**

* **Employee (Employee\_ID) 🡨🡪 Employee (Supervisor)**: An employee can either be supervised by another employee or not be supervised at all (0, 1). On the other hand, a supervisor may oversee one or more employees (1, M).

**Supertype:**

* Tiers with two subtypes – Game\_Tier and Movie\_Tier. It is differentiated by two subtype discriminators – Game\_Tier and Movie\_Tier.

**Disjointed/Overlapping constraint:**

* The subtypes have an overlapping relationship since they can belong to either subtype.

**Specialization constraint:**

* The subtypes have a total specialization since the tiers must belong to at least one of the two subtypes.

**Strong Relationship(s)**:

* Tier 🡨🡪 Game\_Tier
* Game 🡨🡪 Game\_Tier
* Tier 🡨🡪 Movie\_Tier
* Movie 🡨🡪 Movie\_Tier

**Weak Relationship(s)**:

* Game 🡨🡪 DLC
* Game 🡨🡪 Developer
* Developer 🡨🡪 Parent\_Company
* Employee (Employee\_ID) 🡨🡪 Employee (Supervisor)
* Tier 🡨🡪 Employee
* Movie 🡨🡪 Director
* Customer 🡨🡪 Tier

**Existence Dependency**:

* Parent\_Company depends on Developer
* Developer depends on Game
* Game depends on Developer
* DLC depends on Game
* Director depends on Movie
  + M:N relationship broken by Movie\_Director
* Employee (Supervisor) depends on Employee (Employee\_ID)

**Mandatory relationships** (required a NOT NULL constraint on the FK):

* Tier 🡨🡪 Employee
* Tier 🡨🡪 Game\_Tier
* Game 🡨🡪 Game\_Tier
* Game 🡨🡪 DLC
* Developer 🡨🡪 Game
* Tier 🡨🡪 Movie\_Tier
* Tier 🡨🡪 Employee
* Movie 🡨🡪 Movie\_Tier
* Movie 🡨🡪 Movie\_Director
* Director 🡨🡪 Movie\_Director
* Customer 🡨🡪 Tier

***Constraints:***

1. All tiers must have a title
2. Tier entries must have a listed price
3. Each tier must be overseen by one employee
4. An employee can oversee multiple tiers
5. Since not all cultures use a surname, every employee entry should at least have a first name associated with it
6. An employee entry must have its respective department listed as well
7. An employee may or may not be supervised by one other employee
8. A supervisor must oversee one or more employees
9. A customer must only be subscribed to one tier
10. One tier can be the subscription of many customers
11. Since not all cultures use a surname, every customer entry should at least have a first name associated with it
12. Each customer entry should have an age listed
13. Each customer should have their country of origin tied to their entry
14. Each customer should supply an email
15. Each customer should be subscribed to a tier
16. All tiers must either be a movie tier, game tier, or both
17. Multiple tiers can list the same game, and multiple games can appear in the same tier
18. All games are only available for as long as permitted in the licensing contract
19. Game entries must have a title
20. Any listed game on the service must have a rating assigned to it
21. A game can have none, one, or more pieces or DLC tied to it
22. A single piece of DLC can and must only belong to one game
23. DLC entries must have a name attached to them
24. DLC entries must list the game they’re tied to
25. No more than one developer must be listed per Game entry
26. A single developer can develop multiple different games
27. Every developer entry must list the name of the developer
28. Developers can either belong to one parent company or be independent
29. A company can only the parent of (own/employ) one developer
30. Every parent company entry must have their name listed
31. Multiple tiers can list the same movie
32. Multiple movies can appear in the same tier
33. All movies are only available for as long as permitted in the licensing contract
34. Every movie must have a title
35. All movie entries must have their respective rating associated with them
36. A movie can have multiple directors
37. A director can direct multiple movies
38. Since not all cultures use a surname, every director entry should at least have a first name associated with their entry
39. **A summary of the functionality of the application.**

This new database layout allows for the connection between the newly established GAME tables and previously established MOVIE tables. Thanks to this novel connection, both back-end and front-end users can search for and sort individual parts of each table. For example, back-end users can run queries to find the total amounts of movies, games, or both available to each tier. These can also run queries to check which movies are missing data, whether required data or not, to add it to fill out the entry further if needed. Specific details, such as a developer’s country or directors nationality, can be used to hold special events where certain creators from different countries are highlighted. Front-end users will mainly be able to run various queries to search for games and movies based on certain parameters.

Now that we have completed the third deliverable, let’s discuss what was changed as well as the queries were included in the final Visual Studio application. In terms of edits, we changed the CUSTOMER date of birth column to an age column. This was done to fit some functionalities. The DEVELOPER table’s Developer\_Name attribure was updated to VARCHAR(20) to fit longer developer names. To start off, below are the ten main requirements that we were given to consider when designing the queries for the application. The goal here being to have the queries meet all the functionalities; allowing some to meet more than one functionality. Below we list the said requirements:

1. The 10 functionalities should cover at least four entities in your ERD plus at least one of the subtypes
2. At least 3 functionalities should involve more than one entity
3. At least 3 functionalities should involve the use of mathematical functions such as Sum, AVG, Min, Max, or the ANY and ALL keywords
4. At least 1 functionality should involve the use of the Count feature
5. At least 2 functionalities should involve the "Group By" feature
6. Among the above 2 Group By queries, at least one of them should have the "Having" clause
7. At least one functionality that involves a query with the "LIKE" keyword
8. At least 1 correlated query
9. At least 1 functionality should involve the entity that exhibits the unary relationship
10. At least 1 functionality should involve both the supertype and one of the subtypes

Next, we list the features in the Visual Studio application along with what functionalities they meet. Note that not all queries will be listed, as some do not meet any requirements. As such, only queries that meet any of the ten functionalities will be listed. The features below include data from the EMPLOYEE, CUSTOMER, TIER, GAME, and GAME\_TIER tables, and thus meet the 1st functionality. These will be grouped by the form they’re located in:

* Employee Form
  + Show employees with no supervisor (#9)
  + Show employees with a supervisor (#9)
  + List all employees from \_\_\_\_\_\_ department with \_\_\_\_\_\_ as their first name initial (#7, 9)
* Customer/Tier Form
  + Show customers older than the average (#2, 3, 8)
  + Show customers younger than the average (#2, 3, 8)
  + Oldest Customer Age by Tier - Table Adapter (#2, 3, 5)
  + Youngest Customer Age by Tier - Table Adapter (#2, 3, 5)
  + Number of Customers per Tier – Table Adapter (#2, 4, 5)
  + Show Tiers with more than one customer (#2, 4, 5, 6, 8)
* Game\_Tier/Game/Tier Form
  + Show games from \_\_\_\_\_ tier (#2, 10)