**Chapter 1 – NoSQL Data Model**

**Epic Game Store**

# Table of Contents

[Table of Contents 1](#_Toc179150101)

[Presentation of the dataset 2](#_Toc179150102)

[A) Selection of the dataset 2](#_Toc179150103)

[B) Presentation of the tables and organisation 2](#_Toc179150104)

[C) Schema 4](#_Toc179150105)

[D) Volume 4](#_Toc179150106)

[Use cases 5](#_Toc179150107)

[A) End-User view 5](#_Toc179150108)

[B) Data Analyst view 5](#_Toc179150109)

[Relate queries of the dataset 5](#_Toc179150110)

[A) End-User view 5](#_Toc179150111)

[1. View reviews for a specific game 5](#_Toc179150112)

[2. Find hardware requirements for a game 5](#_Toc179150113)

[3. Search for social networks for a game 6](#_Toc179150114)

[4. View tweets related to a game 6](#_Toc179150115)

[B) Data Analyst view 6](#_Toc179150116)

[1. Average rating for each game over time 6](#_Toc179150117)

[2. Identify Most Active Games on Twitter 7](#_Toc179150118)

[3. Top 5 Publishers by Average Game Rating 7](#_Toc179150119)

[4. Trending Genres Based on Twitter Activity and Reviews 7](#_Toc179150120)

[Statistics 8](#_Toc179150121)

[A) Number of documents in each table 8](#_Toc179150122)

[B) Cardinality of attributes 8](#_Toc179150123)

[C) Cardinality of joins 9](#_Toc179150124)

**CDOF3 - LE QUERNEC Loévan / MARTIN Santiago / RINGEISSEN Léo / SALEH Julien**

# Presentation of the dataset

## Selection of the dataset

To find our dataset, we began by searching various online dataset repositories, as outlined in the project guidelines. We needed to find a dataset that met the criteria for volume and complexity (over 400 MB with multiple interconnected tables). However, after extensive research, we couldn't find a dataset that fully satisfied these requirements. After consulting our professor, we were allowed to choose a dataset larger than 100 MB.

We then reviewed several options and ultimately selected a dataset from Kaggle. This dataset was the most complete in terms of structure and data volume, while also meeting our analysis and data infrastructure needs. It contained multiple interconnected tables, which allows us to meet the requirement of at least three joins for the project.

Link of the dataset: <https://www.kaggle.com/datasets/mexwell/epic-games-store-dataset>.

It contains 6 tables, that will be presented just after.

## Presentation of the tables and organisation

The schema above represents a relational database organized around six main tables. Each table stores specific information related to video games, reviews, social networks, and Twitter accounts. Below is a brief description of the various tables and their columns.

1. **Table game**

This table contains the main information about the video games:

* **id** (INT): Unique identifier for the game.
* **name** (VARCHAR(45)): Name of the game.
* **game\_slug** (VARCHAR(45)): Slug of the game (simplified version of the name for URLs).
* **price** (VARCHAR(45)): Price of the game.
* **release\_date** (DATE): Release date of the game.
* **platform** (VARCHAR(45)): Platforms on which the game is available (e.g., PC, Console).
* **description** (VARCHAR(45)): Description of the game.
* **developer** (VARCHAR(45)): Developer of the game.
* **publisher** (VARCHAR(45)): Publisher of the game.
* **genres** (VARCHAR(45)): Genres of the game (e.g., Adventure, Action).

1. **Table critic**

This table groups critical reviews about games:

* **id** (INT): Unique identifier for the review.
* **company** (VARCHAR(45)): Name of the company that published the review.
* **author** (VARCHAR(45)): Author of the review.
* **rating** (VARCHAR(45)): Rating given to the game.
* **comment** (VARCHAR(45)): Comment from the author.
* **date** (DATE): Publication date of the review.
* **top\_critic** (TINYINT): Indicates whether the author is a top critic.
* **fk\_game\_id** (INT): Foreign key referring to the reviewed game.

1. **Table necessary\_hardware**

This table lists the necessary hardware requirements to play the games:

* **id** (INT): Unique identifier.
* **operational\_system** (VARCHAR(45)): Required operating system.
* **processor** (VARCHAR(45)): Required processor.
* **memory** (VARCHAR(45)): Required memory (RAM).
* **graphics** (VARCHAR(45)): Required graphics card.
* **fk\_game\_id** (INT): Foreign key linked to a game.

1. **Table tweets**

This table contains tweets related to games or reviews:

* **id** (INT): Unique identifier for the tweet.
* **text** (VARCHAR(45)): Content of the tweet.
* **url\_media** (VARCHAR(45)): URL of any media attached to the tweet.
* **quantity\_likes** (INT): Number of likes on the tweet.
* **quantity\_retweets** (INT): Number of retweets.
* **quantity\_quotes** (VARCHAR(45)): Number of tweet quotes.
* **quantity\_replys** (VARCHAR(45)): Number of replies to the tweet.
* **timestamp** (DATETIME): Date and time of the tweet.
* **tweet\_ids\_in\_reply** (VARCHAR(45)): IDs of the tweets the user is replying to.
* **fk\_twitter\_account\_id** (INT): Foreign key linked to the Twitter account.

1. **Table twitter\_accounts**

This table contains information about Twitter accounts:

* **id** (INT): Unique identifier for the Twitter account.
* **name** (VARCHAR(45)): Full name of the Twitter account holder.
* **username** (VARCHAR(45)): Twitter username.
* **bio** (VARCHAR(45)): User’s biography.
* **location** (VARCHAR(45)): User’s location.
* **website** (VARCHAR(45)): User’s website.
* **join\_date** (DATE): Date when the account was created.
* **following** (INT): Number of accounts the user is following.
* **followers** (INT): Number of followers of the user.
* **fk\_game\_id** (INT): Foreign key linked to a game.

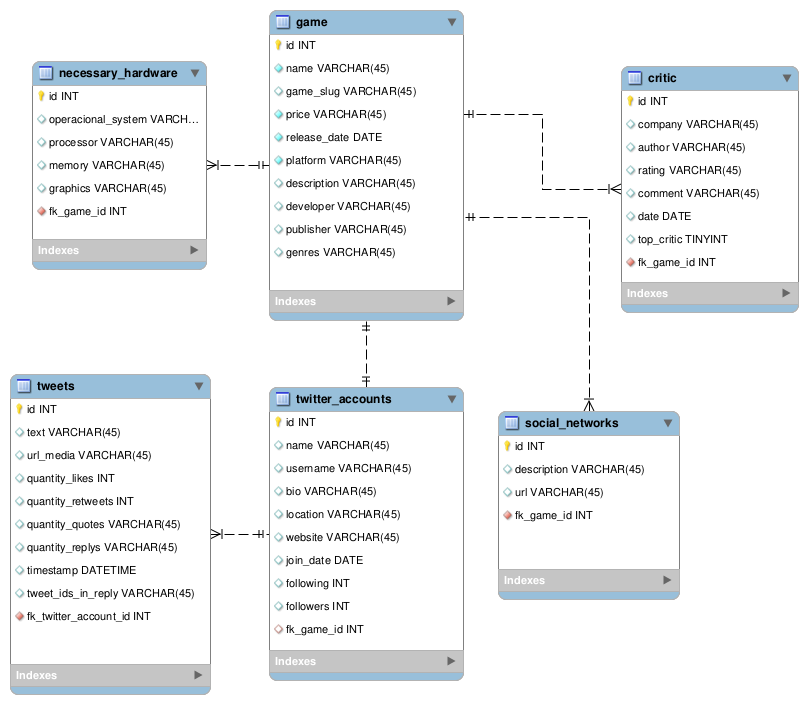
1. **Table social\_networks**

This table stores information about social networks related to games.

* **id** (INT): Unique identifier for the social network.
* **description** (VARCHAR(45)): Description of the social network.
* **url** (VARCHAR(45)): URL of the social network.
* **fk\_game\_id** (INT): Foreign key linked to a game.

## Schema

Here is the schema resuming the structure of our dataset. It is important to note the cardinality between each table: two lines mean 1, an arrow ending with a line means N. In this case, 1 game can have 0 to N reviews, for example.



## Volume

The size of the combined files is 194.62 MB. Our biggest file is the one concerning the tweets, with a single size of 186.24 MB (almost the total volume). The second one is the one containing all the critics, “critic”, with a size of 7.32 MB.

# Use cases

## End-User view

1. View reviews for a specific game
2. Find hardware requirements for a game
3. Search for social networks for a game
4. View tweets related to a game

## Data Analyst view

1. Average rating for each game over time
2. Identify most active games on Twitter
3. Top 5 publishers by average game rating
4. Trending Genres Based on Twitter Activity and Reviews

# Relate queries of the dataset

## End-User view

### 1. View reviews for a specific game

* **Targeted Tables and Attributes**:
  + game: id, name, game\_slug
  + critic: author, rating, comment, date
* **Joins to be Performed**:
  + Join game on critic using game.id = critic.fk\_game\_id
* **Attribute Filters**:
  + Filter on the specific game via game\_slug = 'some\_game\_slug'
* **Projections**:
  + game.name, critic.author, critic.rating, critic.comment, critic.date
* **Aggregates**:
  + None, as we are simply fetching individual reviews for a particular game.

### 2. Find hardware requirements for a game

* **Targeted Tables and Attributes**:
  + game: id, name, game\_slug
  + necessary\_hardware: operational\_system, processor, memory, graphics
* **Joins to be Performed**:
  + Join game on necessary\_hardware using game.id = necessary\_hardware.fk\_game\_id
* **Attribute Filters**:
  + Filter on the specific game via game\_slug = 'some\_game\_slug'
* **Projections**:
  + game.name, necessary\_hardware.operational\_system, necessary\_hardware.processor, necessary\_hardware.memory, necessary\_hardware.graphics
* **Aggregates**:
  + None, as the goal is to retrieve hardware requirements directly.

### 3. Search for social networks for a game

* **Targeted Tables and Attributes**:
  + game: id, name, game\_slug
  + social\_networks: url, description
* **Joins to be Performed**:
  + Join game on social\_networks using game.id = social\_networks.fk\_game\_id
* **Attribute Filters**:
  + Filter on the specific game via game\_slug = 'some\_game\_slug'
* **Projections**:
  + game.name, social\_networks.url, social\_networks.description
* **Aggregates**:
  + None, this query retrieves URLs directly for each game.

### 4. View tweets related to a game

* **Targeted Tables and Attributes**:
  + game: id, name, game\_slug
  + twitter\_accounts: name, username
  + tweets: text, timestamp
* **Joins to be Performed**:
  + Join game on twitter\_accounts using game.id = twitter\_accounts.fk\_game\_id
  + Join twitter\_accounts on tweets using twitter\_accounts.id = tweets.fk\_twitter\_account\_id
* **Attribute Filters**:
  + Filter on the specific game via game\_slug = 'some\_game\_slug'
  + Filter on tweets.timestamp for time-based filtering (e.g., last 24 hours)
* **Projections**:
  + game.name, twitter\_accounts.username, tweets.text, tweets.timestamp
* **Aggregates**:
  + None, as the goal is to retrieve individual tweets related to the game.

## Data Analyst view

### 1. ****Average rating for each game over time****

* **Targeted Tables and Attributes**:
  + game: id, name
  + critic: rating, date
* **Joins to be Performed**:
  + Join game on critic using game.id = critic.fk\_game\_id
* **Attribute Filters**:
  + Filter critic.date to only include reviews from the last 6 months (e.g., critic.date >= 'some\_date')
* **Projections**:
  + game.name, MONTH(critic.date)
* **Aggregates**:
  + AVG(critic.rating) to calculate the average rating per month
  + COUNT(critic.id) to count the number of reviews per month

### 2. ****Identify Most Active Games on Twitter****

* **Targeted Tables and Attributes**:
  + game: id, name
  + twitter\_accounts: id
  + tweets: id, timestamp
* **Joins to be Performed**:
  + Join game on twitter\_accounts using game.id = twitter\_accounts.fk\_game\_id
  + Join twitter\_accounts on tweets using twitter\_accounts.id = tweets.fk\_twitter\_account\_id
* **Attribute Filters**:
  + Filter tweets.timestamp for recent activity (e.g., last month tweets.timestamp >= 'some\_date')
* **Projections**:
  + game.name
* **Aggregates**:
  + COUNT(tweets.id) to calculate the total number of tweets per game

### 3. ****Top 5 Publishers by Average Game Rating****

* **Targeted Tables and Attributes**:
  + game: id, name, publisher
  + critic: rating
* **Joins to be Performed**:
  + Join game on critic using game.id = critic.fk\_game\_id
* **Attribute Filters**:
  + Filter to include only games with at least 10 reviews (HAVING COUNT(critic.id) >= 10)
* **Projections**:
  + game.publisher
* **Aggregates**:
  + AVG(critic.rating) to calculate the average rating per publisher
  + COUNT(critic.id) to ensure only publishers with enough reviews are considered
  + Use LIMIT 5 to get the top 5 publishers.

### 4. ****Trending Genres Based on Twitter Activity and Reviews****

* **Targeted Tables and Attributes**:
  + game: id, name, genres
  + critic: rating
  + tweets: id
* **Joins to be Performed**:
  + Join game on critic using game.id = critic.fk\_game\_id
  + Join game on tweets using game.id = tweets.fk\_game\_id
* **Attribute Filters**:
  + Filter games with average ratings above 80% (AVG(critic.rating) > 80)
  + Filter recent tweets (tweets.timestamp >= 'some\_date')
* **Projections**:
  + game.genres
* **Aggregates**:
  + COUNT(tweets.id) to count the number of tweets per genre
  + AVG(critic.rating) to ensure that only games with high reviews are included

# Statistics

## Number of documents in each table

* “games”: 915 documents.
* “necessary\_hardware”: 1765 documents.
* “open\_critic”: 17,584 documents.
* “social\_networks”: 3045 documents.
* “tweets”: 989,000 documents.
* “twitter\_accounts”: 529 documents.

## Cardinality of attributes

* **Game Table:**
  + **game\_slug:** Used to uniquely identify games, this attribute has high cardinality (close to 915 unique slugs for 915 games).
  + **developer:** Filters based on game developers are common. The cardinality is medium, as some developers have created multiple games.
  + **publisher:** Same principle as developer, filtering by publisher is practical for users. Cardinality is medium, with several publishers linked to multiple games.
  + **genres:** Same principle as developer, and this attribute has between low and medium cardinality, as many games share the same genres.
  + **price**: We often filter games by cost. The cardinality for this column is medium, as many games share similar price ranges but there can still be many of those.
* **Critic Table:**
  + **rating**: Filtering reviews by rating is common, and this attribute has very low cardinality since there are few scores to choose from.
  + **author:** Filters by author have medium to high cardinality, since there’s a decent number of unique reviewers.
  + **company**: Filtering by the review company adds another dimension for users. Cardinality is medium, as multiple reviews come from a few companies.
* **Necessary Hardware Table:**
  + **operational\_system**: Filtering games by operating system is common. Cardinality is low, as the number of operating systems is quite limited.
  + **processor**: This has low cardinality, as many games share similar processor requirements.
  + **memory**: Filtering by memory requirements also has low cardinality, with several games sharing the same requirements.
  + **graphics:** Just like all the previous haptics-related attributes, filters based on graphics card requirements will also have low cardinality, as multiple games use the same range of hardware.
* **Tweets Table:**
  + **timestamp**: Time-based filtering is common, and the cardinality of these will be quite high, as timestamp can take on a very large number of unique values.
  + **quantity\_likes,** **quantity\_retweets**, **quantity\_replys**: All these attributes could be used to measure the level of interaction with a tweet and are therefore important. These attributes all have medium to high cardinality depending on the popularity of the tweets.
* **Twitter Accounts Table:**
  + **location**: Filters by location have low to medium cardinality, as some accounts share similar locations.
  + **followers** and **following:** Once again, these are important to measure social media engagement. These attributes have medium to high cardinality.
* **Social Networks Table:**
  + **url**: This attribute is often unique for each game and therefore has high cardinality. There aren’t other attributes in this table that would likely serve as filters.

## Cardinality of joins

For context, we’ll be calling one-to-one as 1:1, one-to-many as 1, and many-to-many as N.

* **Game - Critic (game.id = critic.fk\_game\_id):**
  + The relationship between the game table and the critic able is a 1 relationship, where one game can have many associated reviews.
  + There are 915 games and 17,584 critic reviews. This means that on average, each game has approximately 19 reviews. However, some games may have many more reviews, while others may have few or none.
  + Cardinality: High due to the many reviews per game.
* **Game - Necessary Hardware (game.id = necessary\_hardware.fk\_game\_id):**
  + This is a 1 relationship, where one game can have multiple necessary hardware entries.
  + There are 915 games and 1765 hardware records. On average, each game is associated with about two hardware entries, although some games may have more or fewer.
  + Cardinality: Medium, with a relatively limited number of hardware records per game.
* **Game - Twitter Accounts (game.id = twitter\_accounts.fk\_game\_id):**
  + This is a 1 relationship, where one game can have multiple Twitter accounts discussing or linking to it.
  + There are 915 games and 529 Twitter accounts. This means some games will have multiple Twitter accounts linked to them, while others may have fewer.
  + Cardinality: Moderate, with multiple accounts possibly linked to each game, but the number of Twitter accounts is smaller than other relationships like reviews or tweets.
* **Game - Social Networks (game.id = social\_networks.fk\_game\_id):**
  + This is a 1 relationship, where one game can be associated with multiple social networks.
  + There are 915 games and 3045 social network records, meaning that on average, each game is associated with roughly 3-4 social network entries.
  + Cardinality: Moderate, since each game has a handful of social network entries.
* **Tweets - Twitter Accounts (tweets.fk\_twitter\_account\_id = twitter\_accounts.id):**
  + This is a 1 relationship, where each Twitter account can have multiple tweets associated with it.
  + There are 989,000 tweets and 529 Twitter accounts, meaning that each Twitter account is responsible for many tweets, creating a high cardinality in this relationship.
  + Cardinality: Very High, as active Twitter accounts can post numerous tweets, especially about popular games.