Matt Fry

CS340 – Final Project

Outline

This database is meant to represent a collection of video games. It holds information about an individual video game, the platforms the games are played on, the genres they fit in, and the companies that create the games and platforms.

This database can be used to keep track of the games that one person owns. The database would make it easy for one to sift through their games and organize them based on certain criteria. It could be further extended to compare two or more people's personal collections. It could be useful compare similarities in games owned or to compare the ratings that two people give a game.

Outline in Words

This database's primary purpose is to store video games. Each video game has an ID, title, release year, and rating, and a publishing company. The id is the primary key.

Also stored are different companies. Each company has an ID, name, headquarters location, and president. The id is the primary key.

A video game must be published by exactly one company. A company may publish multiple games but doesn't have to publish any.

Platforms are the devices that are used to play video games on. Each platform has an ID, release year, name, and initial price. The id is the primary key.

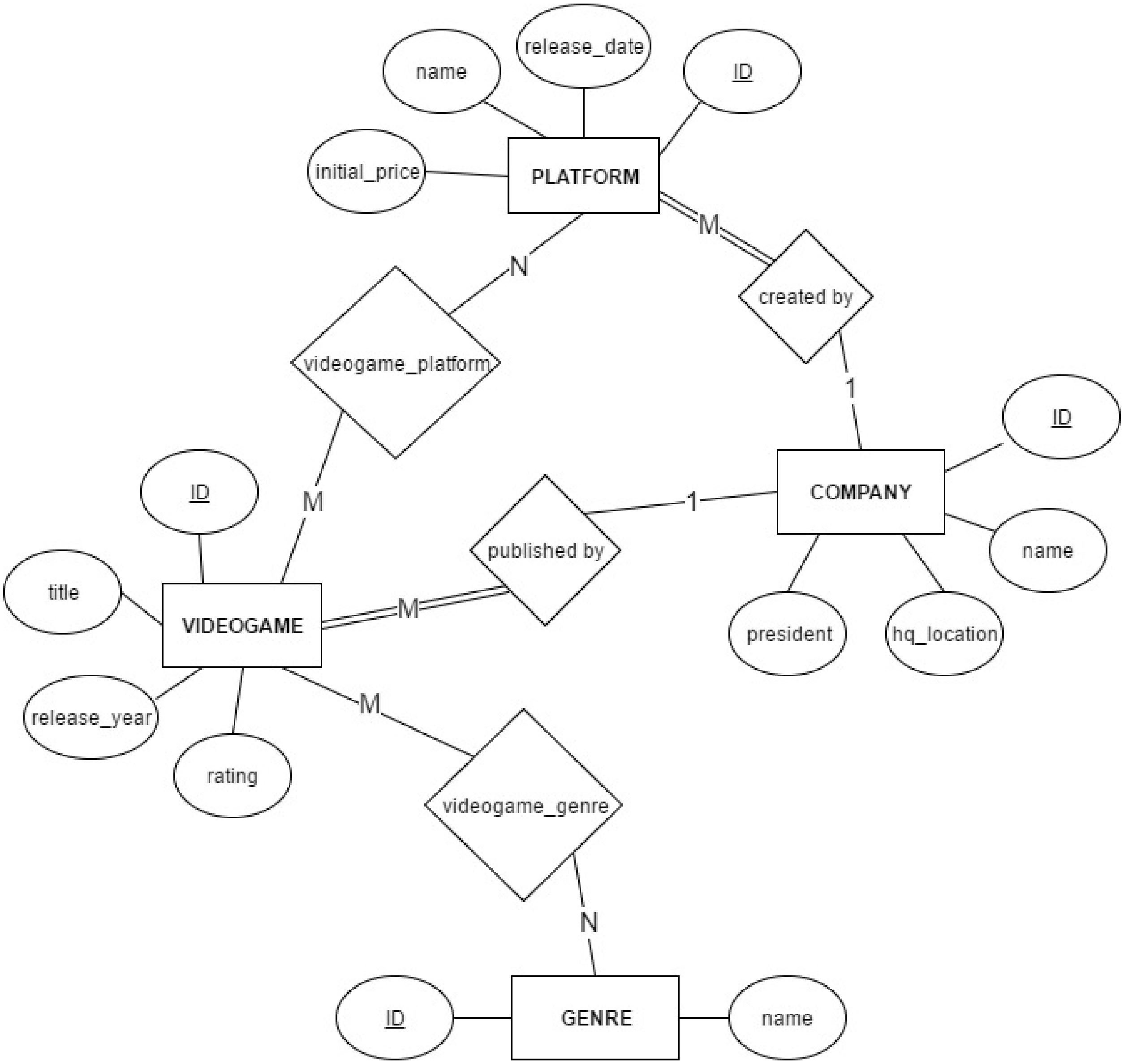
A platform must be created by exactly one company. A company may create multiple platforms but doesn't have to create any.

A video game can belong to multiple platforms and a platform can have many video games. A video game doesn't have to have any platforms and a platform doesn't have to have any video games. The combination of the videogame id and the platform id is the primary key.

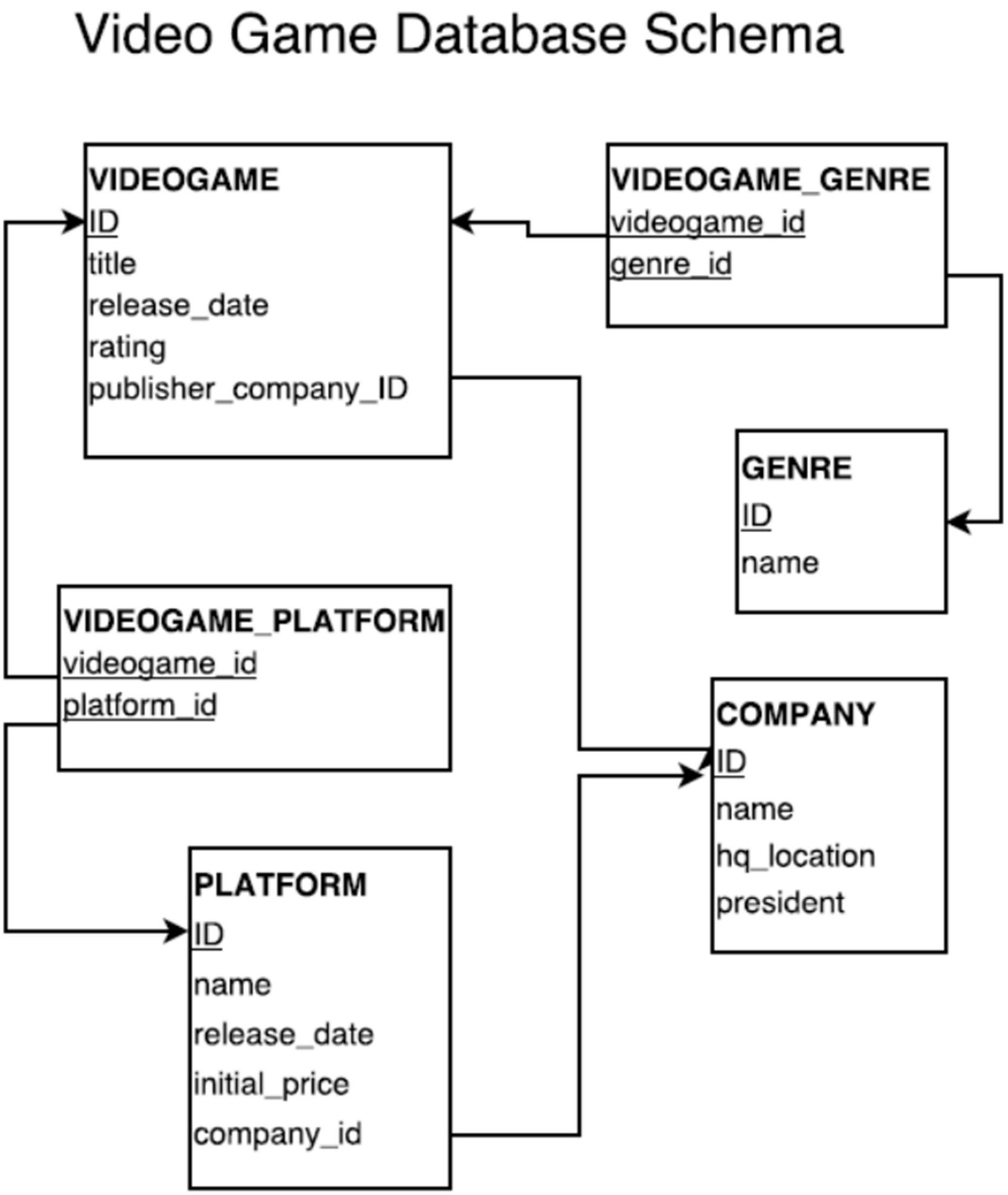
Genres represent the type categories video games are described by. A genre has an id and a name. The id is the primary key.

A video game can have multiple genres and a genre can have multiple video games. A video game doesn't have to have any genres and a genre doesn't have to have any video games. The combination of the videogame id and the genre is the primary key.

ER Diagram


name
release_date
ID
initial_price
PLATFORM
M
N
created by
videogame_platform
1
ID
COMPANY
name
1
published by
M
ID
title
VIDEOGAME
M
release_year
rating
president
hq_location
M
videogame_genre
N
ID
GENRE
name

Schema



Video Game Database Schema

Table Creation Queries

```
CREATE TABLE company(
        id int(11) NOT NULL AUTO_INCREMENT,
        name varchar(255) NOT NULL,
        hq_location varchar(255),
        president varchar(255),
        UNIQUE KEY(name),
        PRIMARY KEY(id)
)ENGINE=InnoDB;

CREATE TABLE videogame (
        id int(11) NOT NULL AUTO_INCREMENT,
        title varchar(255) NOT NULL,
        release_year int(11),
        rating float(11),
        publishing_company_id int(11),
        FOREIGN KEY(publishing_company_id) REFERENCES company(id) ON DELETE SET NULL,
        PRIMARY KEY(id)
)ENGINE=InnoDB;

CREATE TABLE platform(
        id int(11) NOT NULL AUTO_INCREMENT,
        name varchar(255) NOT NULL,
        release_year int(11),
        initial_price int(11),
        company_id int(11),
        PRIMARY KEY(id),
        UNIQUE KEY(name),
        FOREIGN KEY(company_id) REFERENCES company(id) ON DELETE SET NULL
)ENGINE=InnoDB;

CREATE TABLE genre(
        id int(11) NOT NULL AUTO_INCREMENT,
        name varchar(255) NOT NULL,
        UNIQUE KEY(name),
        PRIMARY KEY(id)
)ENGINE=InnoDB;

CREATE TABLE videogame_genre(
        videogame_id int(11) NOT NULL,
        genre_id int(11) NOT NULL,
        PRIMARY KEY(videogame_id,genre_id),
        FOREIGN KEY(videogame_id) REFERENCES videogame(id) ON DELETE CASCADE,
        FOREIGN KEY(genre_id) REFERENCES genre(id)  ON DELETE CASCADE
)ENGINE=InnoDB;

CREATE TABLE videogame_platform(
        videogame_id int(11) NOT NULL,
        platform_id int(11) NOT NULL,
        PRIMARY KEY(videogame_id, platform_id),
        FOREIGN KEY(videogame_id) REFERENCES videogame(id) ON DELETE CASCADE,
        FOREIGN KEY(platform_id) REFERENCES platform(id) ON DELETE CASCADE
)ENGINE=InnoDB;
```

General Use Queries

Get All companies for a drop down menu

```
SELECT id, name FROM company ORDER BY name
```

Get all games for a drop down menu

```
SELECT id, title FROM videogame ORDER BY title
```

Get all platforms for a drop down menu

```
SELECT id, name FROM platform ORDER BY name
```

Get all genres for a drop down menu

```
SELECT id, title FROM videogame ORDER BY title
```

Get data to populate table of games

```
SELECT vg.id, vg.title, vg.release_year, c.name, vg.rating FROM videogame vg LEFT JOIN
company c ON vg.publishing_company_id = c.id ORDER BY vg.title;
```

Get all platforms for a specific video game

```
SELECT name FROM videogame_platform vp INNER JOIN platform p ON p.id = vp.platform_id
WHERE vp.videogame_id=[gameID];
```

Get all genres for a specific game

```
SELECT name FROM videogame_genre vg INNER JOIN genre g ON g.id = vg.genre_id WHERE
vg.videogame_id=[gameID];
```

Get data to populate table of companies

```
SELECT id, name, hq_location, president  FROM company ORDER BY name;
```

Get all platforms owned by a specific company

```
SELECT p.name FROM company c INNER JOIN platform p ON p.company_id = c.id WHERE
p.company_id=[companyId];
```

Get all data to populate a table of platforms

```
SELECT p.id, p.name, p.release_year, p.initial_price, c.name FROM platform p LEFT JOIN
company c ON p.company_id = c.id;
```

Get data to populate table of genres

```
SELECT id, name FROM genre ORDER BY name;
```

Insert row into videogame table

```
INSERT INTO videogame(title, release_year, publishing_company_id, rating) VALUES
([title],[release_year],[publishing_company_id],[rating]);
```

Insert row into videogame_platform table

```
INSERT INTO videogame_platform(videogame_id, platform_id) VALUES
([videogame_id],[platform_id]);
```

Insert row into videogame_genre table

```
INSERT INTO videogame_genre(videogame_id, genre_id) VALUES ([videogame_id], [genre_id]);
```

Insert row into company table

```
INSERT INTO company(name, hq_location, president) VALUES
([name],[hq_location],[president]);
```

Insert row into platform table

```
INSERT INTO platform(name, release_year, company_id, initial_price) VALUES
([name],[release_year],[company_id],[initial_price]);
```

Insert row into genre table

```
INSERT INTO genre(name) VALUES ([name]);
```

Delete row from videogame table

```
DELETE FROM videogame WHERE id = [gameId];
```

Delete row from company table

```
DELETE FROM company WHERE id = [companyId];
```

Delete row from platform table

```
DELETE FROM platform WHERE id = [platformId];
```

Delete row from genre table

```
DELETE FROM genre WHERE id = [genreId];
```

Select videogames that match a search for title

```
SELECT vg.id, vg.title, vg.release_year, c.name, vg.rating FROM videogame vg LEFT JOIN
company c ON vg.publishing_company_id = c.id WHERE vg.title=[searchInput] ORDER BY vg.title
```

Filter video games by a selected publishing company

```
SELECT vg.id, vg.title, vg.release_year, c.name, vg.rating FROM videogame vg LEFT JOIN
company c ON vg.publishing_company_id = c.id WHERE c.id = [filterInput] ORDER BY vg.title
```

Filter video games by a selected platform

```
SELECT p.name, vg.id, vg.title, vg.release_year, c.name, vg.rating FROM videogame vg LEFT JOIN
company c ON vg.publishing_company_id = c.id INNER JOIN videogame_platform vp ON vg.id =
vp.videogame_id INNER JOIN platform p ON vp.platform_id = p.id WHERE vp.platform_id
=[filterInput] ORDER BY vg.title
```

Filter video games by a selected genre

```
SELECT g.name, vg.id, vg.title, vg.release_year, c.name, vg.rating FROM videogame vg LEFT JOIN
company c ON vg.publishing_company_id = c.id INNER JOIN videogame_genre vgg ON vg.id =
vgg.videogame_id INNER JOIN genre g ON vgg.genre_id = g.id WHERE vgg.genre_id =
[filterInput] ORDER BY vg.title
```