1. **Table creation**

* create table games (

ID char(7),

Name varchar(255) NOT NULL,

Support\_Info varchar(255),

Contains\_dlc Boolean,

Base\_Price Int,

Current\_Price Int,

Developer varchar(255),

Publisher varchar(255),

Genre varchar(255),

Release\_Date varchar(255),

Required\_Age Int,

Controller\_Support Boolean,

Website (nullable) varchar(255),

Short\_Description varchar(255),

Detailed\_Description varchar(255),

Supported\_Languages varchar(255),

Platform varchar(255),

Header\_image varchar(255),

PRIMARY KEY(ID)

FOREIGN KEY (ID) REFERENCES movie(ID)

);

*Note: Support\_Languages will have multi-columns like “English”, “Spanish”, etc.*

*This is the main table for gathering information about games. Information that will be displayed to the user if they search for games.*

* create table music (

ID char(7),

Name varchar(255),

Base\_Price Int,

Current\_Price Int,

Developer varchar(255),

Publisher varchar(255),

Release\_Date varchar(255),

Required\_Age Int,

Controller\_Support Boolean,

Website (nullable) varchar(255),

Short\_Description varchar(255),

Detailed\_Description varchar(255),

Supported\_Languages varchar(255),

Platform varchar(255),

Header\_image varchar(255),

Fullgame\_id char(7)

PRIMARY KEY(ID)

FOREIGN KEY (Fullgame\_id) REFERENCES games(ID)

);

*This is the main table for gathering information about music on Steam. It will display information to the user if they search for music or songs.*

* create table DLC (

ID char(7),

Name varchar(255),

Base\_Price Int,

Current\_Price Int,

Developer varchar(255),

Publisher varchar(255),

Release\_Date varchar(255),

Required\_Age Int,

Controller\_Support Boolean,

Website varchar(255),

Short\_Description varchar(255),

Detailed\_Description varchar(255),

Supported\_Languages varchar(255),

Platform varchar(255),

Header\_image varchar(255),

PRIMARY KEY(ID)

);

*– Note: Since Game(name) and DLC(name) will not match, they will not be foreign keys of each other. May need to add a game\_name for referencing a foreign key from the games table; contains\_DLC may help with this.*

*This is the main table for gathering information about DLC for games. Information that will be displayed to the user if they search for DLC names.*

* create table demo (

ID char(7),

Name varchar(255),

Base\_Price Int,

Current\_Price Int,

Developer varchar(255),

Publisher varchar(255),

Release\_Date varchar(255),

Required\_Age Int,

Controller\_Support Boolean,

Website (nullable) varchar(255),

Short\_Description varchar(255),

Detailed\_Description varchar(255),

Supported\_Languages varchar(255),

Platform varchar(255),

Header\_image varchar(255),

Fullgame\_id char(7)

PRIMARY KEY(ID)

);

*–Note: games(Name) and demo(Name) will not be foreign keys for same reason. The Demo will however contain something along the lines of {Game name} followed by the word “Demo”.*

*This is the main table for gathering information about demo games. Information that will be displayed to the user if they search for demos.*

* create table movie (

ID char(7),

Highlight varchar(255),

Trailer varchar(255),

480p\_video varchar(255),

Max\_video varchar(255),

PRIMARY KEY (ID)

FOREIGN KEY (ID) REFERENCES games(ID)

);

*–All data was obtained from the Steam Web API. Only needed schema have been selected.*

The movie table holds the videos, highlights, and trailers for the games. Its foreign key ID references the games table and will be needed to display these videos on the website.

**2. 2-3 Draft Dummy tables**

**Please see attached excel sheet for tables. They would not visually fit on doc.**

INSERT INTO Games (ID, Name, Support\_Info, Contains\_dlc, Base\_Price, Current\_Price, Developer, Publisher, Genre, Release\_Date , Required\_Age, Controller\_Support, Website, Short\_Description, Detailed\_Description, Supported\_Languages, Platform, Header\_image) VALUES (...);

**3. Write a minimum of 5 select statements.**

1. SELECT \*

FROM games

WHERE Name LIKE search% OR Name = search;

**Description:**

**Selects all columns of information from the games table where user input (“search”) matches the name or where the input starts to reach a game name. This is the main selection of the project, as most users are going to search for a game by name. But it is equally important because this statement will be slightly adjusted in order to search by categories such as Genre, Publisher, Developer, or more.**

1. SELECT \*

FROM music

WHERE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

**Description:**

1. SELECT \*

FROM demo

WHERE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Description:**

1. SELECT \*

FROM DLC

WHERE year LIKE search% OR year = search

**Description:**

**Selects all columns of information from the games table where user input matches the year. Great for finding recently released DLC or finding some old ones depending on user input.**

1. Select \*

FROM games

WHERE Genre LIKE search% or Genre = search

**Select all columns of information from all games where the genre is the same as the user input (“search”). Finding games with a certain genre is perfect for narrowing down what is being looked for.**