CS340 Final Project
Nintendo Game Database
John Moyer & Darlene Zouras
Fall 2016

Overview

This database focuses on the products of Nintendo, a Japanese gaming company originally known for producing playing cards. As the company expanded further into the toy and game industry, it began delving into electronic home games in the 1970s, embarking on a quest for the cutting edge of entertainment that would cement its legacy as one of the most durable, prolific, and beloved video game companies.

The database represents the various Nintendo products, both video game systems and games, as well as some thematic concepts related to the games themselves. Nintendo released a number of video game consoles and mobile systems dating back to the late 70s, each with a wide variety of games available. The creativity of game developers spawned all sorts of games, games where you jump around squashing enemies to games where you shoot at aliens or create music. Many games prominently feature characters or settings that the player can play as or interact with in other ways.

Whether it's for general reference, getting a leg up in a game, or searching for the next game to play, there are all manner of databases devoted to video games. The Minecraft Wiki features a plethora of information about how to play the games and what one might expect to encounter, gamefaqs hosts a huge collection of games with cheats, hints, and walkthroughs to help gamers play, and the Entertainment Software Rating Board maintains resources on video game content ratings to help consumers make good purchasing decisions. All of these video game resources are supported by databases behind the scenes. This Nintendo database would function as the precursor to a more full archive on Nintendo products, whether for reference or nostalgia, and would serve as a resource for people interested in video games, or specifically Nintendo games.

Database Outline

This database features four entities: systems, games, characters, and locations. It will also have four relationships, those between characters and locations, between games and systems, between characters and games, and between locations and games.

Location Entity

The location entity represents a location in a Nintendo game. The location entity features a locationID, a locationName, and an environment. The locationID, an integer, is not null, auto increments and is the primary key. The location name is a varchar that cannot be null and is also a unique key. The environment is a varchar that represents the type of place the location is, which could be something like a forest or outer space.

Character Entity

The character entity represents a specific character, or in some cases a group of characters, in a Nintendo Game. The character entity features a characterId, name, and raceOrSpecies. The characterID is an int key attribute that is auto incremented and cannot be null. It is also the primary key. The name is a varchar key attribute with a unique key that cannot be null. The raceOrSpecies attribute is necessarily vague varchar attribute, as some characters may have no stated race or species or may be somewhat ambiguous. As such, it can be null. The character entity also has a relationship with a location, represented by the homeland foreign key referencing a location's locationID. Even if a character's homeland is unknown, they have a clear origin within the game, and hence, their homeland cannot be NULL.

System Entity

The system entity represents an individual Nintendo video game console or mobile system. The system entity features a systemID, systemName, releaseMonth, releaseDay, releaseYear, unitsSold, and introPriceUS. The systemID is an int that cannot be null, is auto incremented, and is the primary key. systemName is a varchar that cannot be null. systemName is not a unique key, as some systems have released multiple variants that do not have unique names, which may allow for future expandability of the database. releaseMonth, releaseDay, and releaseYear are ints that represent the release date of the system. Individual integers were chosen to decrease the difficulty of parsing date variables in a situation where the year is generally the only important value needed. unitsSold is an int that represents the total units sold, this cannot be NULL, as even an unreleased system should have 0 units sold. Lastly, introPriceUS is a float for the initial sell price in USD of the system, this can be null as a system may never be released.

Game Entity

The game entity represents Nintendo games. The game features a gameID, gameTitle, setting, releaseDay, releaseMonth, releaseYear, and releaseSystem. gameID is an auto-incrementing int that is also the primary key. gameTitle is a varchar that is a unique key that cannot be null. Setting is an int that is a foreign key referencing a locationID. As not all games have settings, the foreign key requirements necessitate them having a setting, so an 'undefined' or 'abstract' setting entry may be recommended. releaseDay, releaseMonth, and releaseYear are ints that represent the U.S. release date of the game - they can be null as some games are never released. releaseSystem is an int that is a foreign key referencing a systemID, it cannot be NULL as even if it was unreleased it had an intended releaseSystem.

Homeland Relationship

A character originally comes from a single location

Every character has at most one location where they are from, their homeland. This is the location most closely identified with the character's origin. In some circumstances this may be unknown, allowing for the presence of a 'null' value for a character's homeland. While each character has at most one homeland, a location may serve as the homeland for many characters.

Release System Relationship

A game originally released on a single system

A game has at most one game system where it is initially released. There may be games that never make it out of development, and so do not have a release system. A game can have one release system, but a system can have many games.

Setting Relationship

A game has a location as its primary setting

A game has at most one location that is the primary setting for the game. A game like tetris has no setting, it is merely a box that shapes enter and stack up. However, a game like the Legend of Zelda: A Link to the Past has Hyrule as it's setting, even if Hyrule itself is made up of many different locations. A game has one location as its primary setting, but a location may serve as the setting for many different games.

Characters in a Game and Games a Character is in

Many characters can be in a game, and a character can be in many games

A character may appear in many different games, and a game may feature many different characters. This serves as a many-to-many relationship. A character must be in at least one game, as this entity encompasses video game characters, and a character isn't a video game character without appearing in a game. This is represented on the ER diagram with total participation, though is actually a semantic integrity constraint and not enforced in the database. The combination of gameID and characterID serves as the primary key for this table, ensuring no duplicate game/character relationships.

MYSQL QUERIES

Create Tables

```
SET FOREIGN_KEY_CHECKS=0;
DROP TABLE IF EXISTS locations;
DROP TABLE IF EXISTS systems;
DROP TABLE IF EXISTS characters;
DROP TABLE IF EXISTS games;
DROP TABLE IF EXISTS gameChars;
SET FOREIGN_KEY_CHECKS=1;

CREATE TABLE locations (
        locationID int(11) NOT NULL AUTO_INCREMENT,
        locationName varchar(50) NOT NULL,
        environment varchar(50),
        UNIQUE KEY (locationName),
        PRIMARY KEY (locationID)
) ENGINE=innoDB DEFAULT CHARSET=latin1;


CREATE TABLE systems (
        systemID int NOT NULL AUTO_INCREMENT,
        systemName varchar(50) NOT NULL,
        releaseMonth int(2),
        releaseDay int(2),
        releaseYear int(4),
        unitsSold int(11) NOT NULL,
        introPriceUS float,
        UNIQUE KEY (systemName),
        PRIMARY KEY (systemID)
) ENGINE=innoDB DEFAULT CHARSET=latin1;


CREATE TABLE characters (
        characterID int(11) NOT NULL AUTO_INCREMENT,
        name varchar(50) NOT NULL,
        raceOrSpecies varchar(50),
        homeland int(11) NOT NULL,
        UNIQUE KEY (name),
        FOREIGN KEY (homeland) REFERENCES locations(locationID) ON DELETE
CASCADE,
        PRIMARY KEY (characterID)
) ENGINE=innoDB DEFAULT CHARSET=latin1;


CREATE TABLE games (
        gameID int(11) NOT NULL AUTO_INCREMENT,
        gameTitle varchar(50) NOT NULL,
        setting int(11) NOT NULL,
        releaseDay int(2),
      releaseMonth int(2),
      releaseYear int(4),
        releaseSystem int(11) NOT NULL,
        UNIQUE KEY (gameTitle),
        FOREIGN KEY (setting) REFERENCES locations(locationID) ON DELETE CASCADE,
        FOREIGN KEY (releaseSystem) REFERENCES systems(systemID) ON DELETE
CASCADE,
        PRIMARY KEY (gameID)
) ENGINE=innoDB DEFAULT CHARSET=latin1;


CREATE TABLE gameChars (
        cid int(11) NOT NULL,
        gid int(11) NOT NULL,
        FOREIGN KEY (cid) REFERENCES characters(characterID) ON DELETE CASCADE,
        FOREIGN KEY (gid) REFERENCES games(gameID) ON DELETE CASCADE,
        PRIMARY KEY (cid, gid)
) ENGINE=innoDB DEFAULT CHARSET=latin1;
```

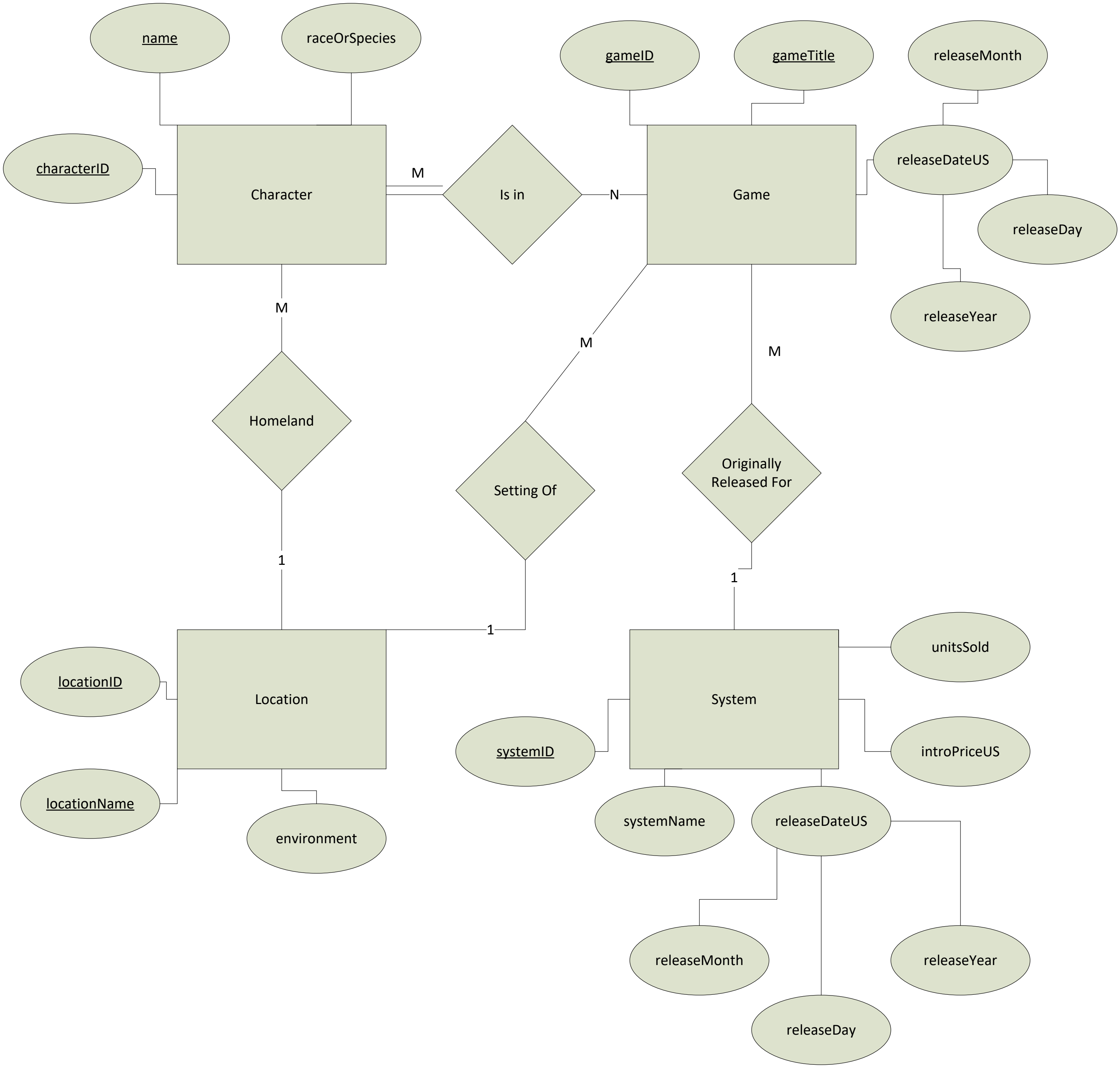
GENERAL QUERIES

```
--Inserts a new location into the locations table
INSERT INTO locations(locationName, environment) VALUES
([nameInput],[environmentInput]);

--Inserts a new system into the systems table
INSERT INTO systems(systemName, releaseMonth, releaseDay, releaseYear, unitsSold,
introPriceUS) VALUES ([nameInput], [monthInput],[dayInput],[yearInput], [unitsSoldInput],
[introPriceInput]);

--Inserts a new character into the characters table - references foreign key of homeland location
INSERT INTO characters(name, raceOrSpecies, homeland) VALUES ([nameInput],
[raceOrSpeciesInput], (SELECT locationID FROM locations WHERE
locationName=[homelandInput]));
```

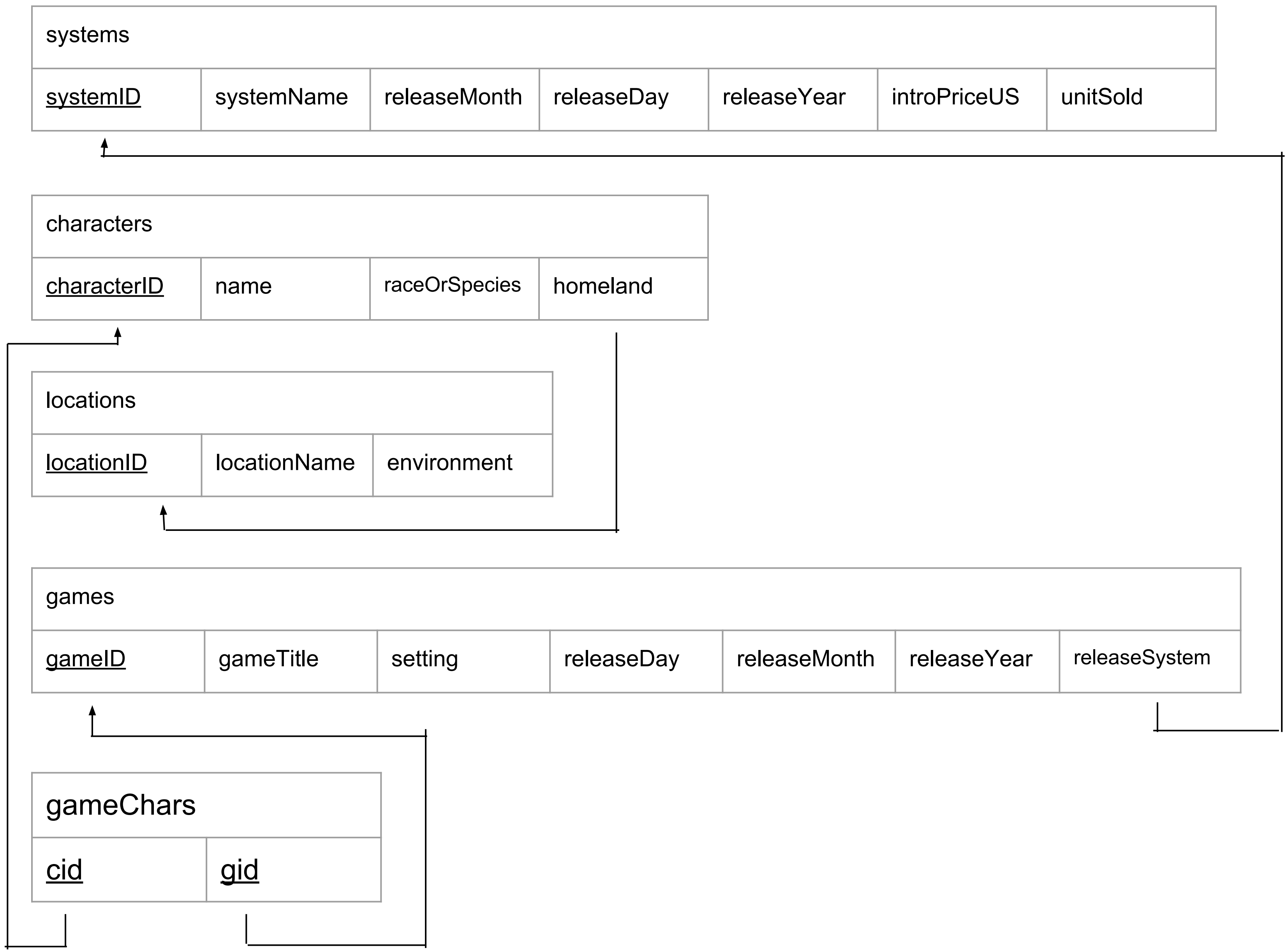
```
--Inserts a new game into the game table - references foreign key of the setting location and the release system
INSERT INTO games(gameTitle, setting, releaseDay, releaseMonth, releaseYear, releaseSystem) VALUES ([nameInput], (SELECT locationID FROM locations WHERE locationName=[settingInput]),
        dayInput, monthInput, yearInput, (SELECT systemID FROM systems WHERE systemName=[systemInput]));

--Inserts new game/character relationship
INSERT INTO gameChars (cid, gid) VALUES ((SELECT characterID FROM characters WHERE name=[nameInput]), (SELECT gameID FROM games WHERE gameTitle=[gameInput]));

--Delete location from the locations table based on the name
DELETE FROM locations WHERE locationName=[nameDeleteInput];

--Delete system from systems table based on the system name
DELETE FROM systems WHERE systemName=[systemDeleteInput];

--Deletes a character from the characters table based on the name
DELETE FROM characters WHERE name=[nameDeleteInput];

--Deletes a game from the games table based on the game title
DELETE FROM games WHERE gameTitle=[gameDeleteInput];

--Deletes an entry from the gameChars
--It seems far more likely that for the gameChars table a character would be entered for a game incorrectly
DELETE FROM gameChars WHERE cid=(SELECT characterID FROM characters WHERE name=[characterDeleteInput]);

--Updates number of units sold for the matching system
UPDATE systems SET unitsSold=[unitsSoldInput] WHERE systemName=[systemNameInput];

--Updates the release date of a game for the matching title
UPDATE games SET releaseDay=[dayInput], releaseMonth=[monthInput], releaseYear=[yearInput] WHERE gameTitle=[gameInput];

--Updates the name of an environment with a new name
UPDATE locations SET environment=[newEnvironmentInput] WHERE environment=[oldEnvironmentInput];
```

```
--Updates the race or species info of the matching character
UPDATE characters SET raceOrSpecies=[raceOrSpeciesInput] WHERE name=[nameInput];

--Select and display location names of locations with a certain environment type
SELECT locationName FROM locations WHERE environment=[environmentInput] ORDER BY
locationName;

--Display system name, release year, and units sold for systems exceeding input number sold
SELECT systemName, releaseYear, unitsSold FROM systems WHERE
unitsSold>[unitSoldInput] ORDER BY unitsSold DESC;

--Display name of characters with a specific homeland
SELECT name FROM characters WHERE homeland=(SELECT locationID FROM locations
WHERE locationName=[homelandInput]) ORDER BY name;

--Display game and release year of all games released for a system ordered by release date
SELECT game, releaseYear FROM games WHERE releaseSystem=(SELECT systemID FROM
systems WHERE systemName=[systemInput]) ORDER BY releaseYear, releaseMonth,
releaseDay;

--Display characters appearing in the selected game
SELECT name FROM characters INNER JOIN gameChars ON gameChars.gid =
games.gameID INNER JOIN characters ON characters.characterID = gameChars.cid WHERE
games.gameTitle=[gameInput] ORDER BY name;

--Display games that a character appears in
SELECT gameTitle FROM games INNER JOIN gameChars ON gameChars.cid =
characters.characterID INNER JOIN games ON games.gameID = gameChars.gid WHERE
characters.name=[nameInput] ORDER BY releaseYear, gameTitle;

--Display character table
SELECT name, raceOrSpecies, locations.locationName FROM characters INNER JOIN
locations ON locations.locationID = characters.homeland ORDER BY name

--Display systems table
SELECT systemName, releaseYear, introPriceUS, unitsSold FROM systems ORDER BY
releaseYear

--Display locations table
SELECT locationName, environment FROM locations ORDER BY locationName
```

```
--Display games table
SELECT gameTitle, location.locationName, releaseYear, releaseSYSTEM FROM games LEFT
JOIN locations ON locations.locationID = games.setting LEFT JOIN systems ON
systems.systemID = games.releaseSystem ORDER BY gameTitle

--Display gamechars table
SELECT games.gameTitle, characters.name FROM gameChars INNER JOIN characters ON
characters.id = gameChars.cid INNERJOIN games ON games.gameID = gameChars.gid
ORDER BY gameTitle
```


name
raceOrSpecies
characterID
Character
M
Is in
N
gameID
gameTitle
releaseMonth
Game
releaseDateUS
releaseDay
releaseYear
M
M
M
Homeland
Setting Of
Originally Released For
1
1
1
Location
locationID
locationName
environment
System
unitsSold
systemID
introPriceUS
systemName
releaseDateUS
releaseMonth
releaseYear
releaseDay

Schema



systems

systemID	systemName	releaseMonth	releaseDay	releaseYear	introPriceUS	unitSold

characters

characterID	name	raceOrSpecies	homeland

locations

locationID	locationName	environment

games

gameID	gameTitle	setting	releaseDay	releaseMonth	releaseYear	releaseSystem

gameChars

cid	gid