Khoa M. Phan
CS340 - Final Project
December 2, 2017

Outline:

Since the first video game was created, it has grown into a worldwide phenomenon that continues to grow exponentially. Since then, there have been many games that made and many companies that have been built upon this premise. Generations of gaming consoles have existed with many games specifically released for that unit. This database allows users to see these games and retrieve information of that game from the year it was released to the studio that creates them.

Database:

Game - gameID is the Primary Key, with the publisherID as the Foreign Key. This database holds the information regarding the specific game.

Genre - genreID is the Primary Key. This contains the genre a game will fall under and a description of what that genre is.

Review- reviewID is the Primary Key with gameID as the Foreign Key to tie the review back to the specific game. This contains the game’s review score by IGN and a brief description of the review.

Publisher - publisherID is the Primary Key with a studioID as the Foreign Key. This contains the information like the name, location and year established of the publisher who distributes the game.

Platform - platformID is the Primary Key. This will hold the make, model and year that the console that the video games can be released under.

Game Platform - contains platformID and gameID which act as Primary Keys and Foreign Keys.

Game Genre - contains genreID and gameID which act as Primary Keys and Foreign Keys..

Table Relationships:

Game Platform - Since a video game can be released under multiple platforms and there can be many platforms for many games, this becomes a many-to-many relationship. Both keys are primary between the game and platform.
Game Genre - Since a video game can have multiple genres and many genres can be have multiple games, this becomes a many-to-many relationship .Both keys are primary between the game and genre.

Publisher - The publisher can publisher many games, but the games can only be published by one publisher. This means this is a one to many relationship.

Review - This review can only be linked to one game, thus creating a one to one relationship.

Table Creation Queries:

```
CREATE TABLE genre (
        `genreID` INT(11) PRIMARY KEY AUTO_INCREMENT,
        `type` VARCHAR(255),
        `description` TEXT
) ENGINE = 'innoDB';

CREATE TABLE review (
        `reviewID` INT(11) PRIMARY KEY AUTO_INCREMENT,
        `gameID` INT(11),
        `score` INT(11),
        `description` TEXT,
        FOREIGN KEY(`gameID`) REFERENCES `game`(`gameID`)
) ENGINE = 'innoDB';

CREATE TABLE platform (
        `platformID` INT(11) PRIMARY KEY AUTO_INCREMENT,
        `make` VARCHAR(255),
        `name` VARCHAR(255),
        `year` INT(11)
) ENGINE = 'innoDB';




CREATE TABLE publisher (
        `publisherID` INT(11) PRIMARY KEY AUTO_INCREMENT,
        `name` VARCHAR(255),
        `location` VARCHAR(255),
        `yearEstab` INT(11)
) ENGINE = 'innoDB';

CREATE TABLE game_platform (
        `platformID` INT(11),
        `gameID` INT(11),
        PRIMARY     KEY (`platformID`, `gameID`),
        FOREIGN KEY(`platformID`) REFERENCES `platform`(`platformID`),
        FOREIGN KEY(`gameID`) REFERENCES `game`(`gameID`)
) ENGINE = 'innoDB';
```

```
CREATE TABLE game_genre (
        `genreID` INT(11),
        `gameID` INT(11),
        PRIMARY     KEY (`genreID`, `gameID`),
        FOREIGN KEY(`genreID`) REFERENCES `genre`(`genreID`),
        FOREIGN KEY(`gameID`) REFERENCES `game`(`gameID`)
) ENGINE = 'innoDB';

CREATE TABLE `game` (
        `gameID` INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
        `name` VARCHAR(255),
        `releasedate` VARCHAR(255),
        `publisherID` INT(11),
        FOREIGN KEY(`publisherID`) REFERENCES `publisher`(`publisherID`),
) ENGINE = 'innoDB';
```

Table Select Queries

```
SELECT * FROM game WHERE name LIKE "%'+req.query.search+'%"
SELECT * FROM game
SELECT * FROM publisher
SELECT * FROM genre
SELECT * FROM review
SELECT * FROM platform
SELECT gameID FROM game
SELECT type FROM genre
SELECT name FROM platform
SELECT * FROM game WHERE gameID = ' "+req.params.gameID+"'
SELECT * FROM publisher WHERE publisherID = ' "+req.params.publisherID+"'
SELECT * FROM genre WHERE genreID = ' "+req.params.genreID+"'
SELECT * FROM review WHERE reviewID = ' "+req.params.reviewID+" '
SELECT * FROM platform WHERE platformID = ' "+req.params.platformID+"'
SELECT * FROM game g INNER JOIN publisher p ON p.publisherID = g.publisherID
SELECT score FROM review r INNER JOIN game g ON g.gameID = r.gameID
SELECT type FROM genre g INNER JOIN game_genre gg
        ON gg.genreID = g.genreID
ORDER BY gameID ASC
SELECT name FROM platform p INNER JOIN game_platform gp
        ON gp.platformID = p.platformID ORDER BY gameID ASC;
SELECT `AUTO_INCREMENT` FROM INFORMATION_SCHEMA.TABLES
        WHERE TABLE_SCHEMA = 'cs340_phankhoa' AND TABLE_NAME = 'game'
```

Table Insert Queries

```
INSERT INTO game (`name`, `releasedate`, `publisherID`)
        VALUES ([game.name], [game.releasedate], [game.publisherID])
INSERT INTO game_genre (`genreID`, `gameID`)
        VALUES (genreID, [game_genre.gameID])
```

```
INSERT INTO game_platform (`platformID`, `gameID`)
        VALUES (platformID, [game_platform.gameID])
INSERT INTO genre (`type`, `description`)
        VALUES ([genre.type], [genre.description])
INSERT INTO review (`gameID`, `score`, `description`)
        VALUES ([review.gameID], [review.score], [review.description])
INSERT INTO publisher (`name`, `location`, `yearEstab`)
        VALUES ([publisher.name], [publisher.location], [publisher.yearEstab])
INSERT INTO platform (`make`, `name`, `year`)
        VALUES ([platform.make], [platform.name], [platform.year])
```

Table Update Queries

```
UPDATE game SET name=[game.name], releasedate=[game.releasedate],
publisherID=[game.publisherID]
        WHERE gameID=' "+req.params.gameID+" '
UPDATE genre SET type=[genre.type], description=[genre.description]
        WHERE genreID=' "+req.params.genreID+" '
UPDATE review SET score=[review.score], description=[review.description]
        WHERE reviewID=' "+req.params.reviewID+" '
UPDATE publisher SET name=[publisher.name], location=[publisher.location],
yearEstab=[publisher.yearEstab]
        WHERE publisherID=' "+req.params.publisherID+" '
UPDATE platform SET make=[platform.make], name=[platform.name],
year=[platform.year]
        WHERE platformID=' "+req.params.platformID+" '
```

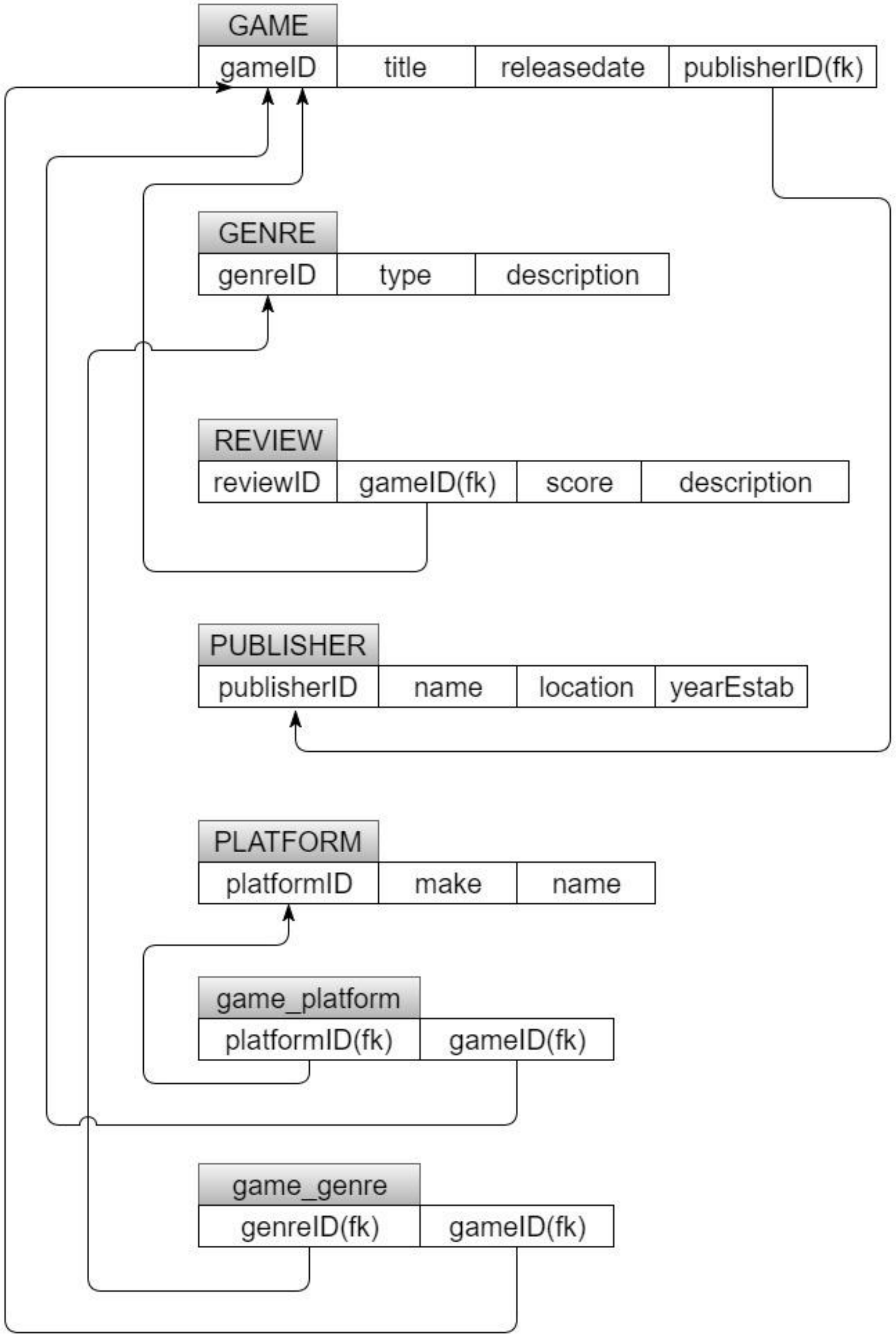
Table Delete Queries

```
DELETE FROM game_genre WHERE gameID=' "+req.params.gameID+" ' AND
genreID
DELETE FROM game_platform WHERE gameID=' "+req.params.gameID+" ' AND
platformID
DELETE FROM game WHERE gameID=' "+req.params.gameID+" '
DELETE FROM review WHERE gameID=' "+req.params.gameID+" '
DELETE FROM genre WHERE genreID=' "+req.params.genreID+" '
DELETE FROM review WHERE reviewID=' "+req.params.reviewID+" '
DELETE FROM publisher WHERE publisherID=' "+req.params.publisherID+" '
DELETE FROM platform WHERE platformID=' "+req.params.platformID+" '
```

Utility Queries

```
SET FOREIGN_KEY_CHECKS=0
SET FOREIGN_KEY_CHECKS=1
```

PROJECT GENERAL ER DIAGRAM



GAME

gameID	title	releasedate	publisherID(fk)

GENRE

genreID	type	description

REVIEW

reviewID	gameID(fk)	score	description

PUBLISHER

publisherID	name	location	yearEstab

PLATFORM

platformID	make	name

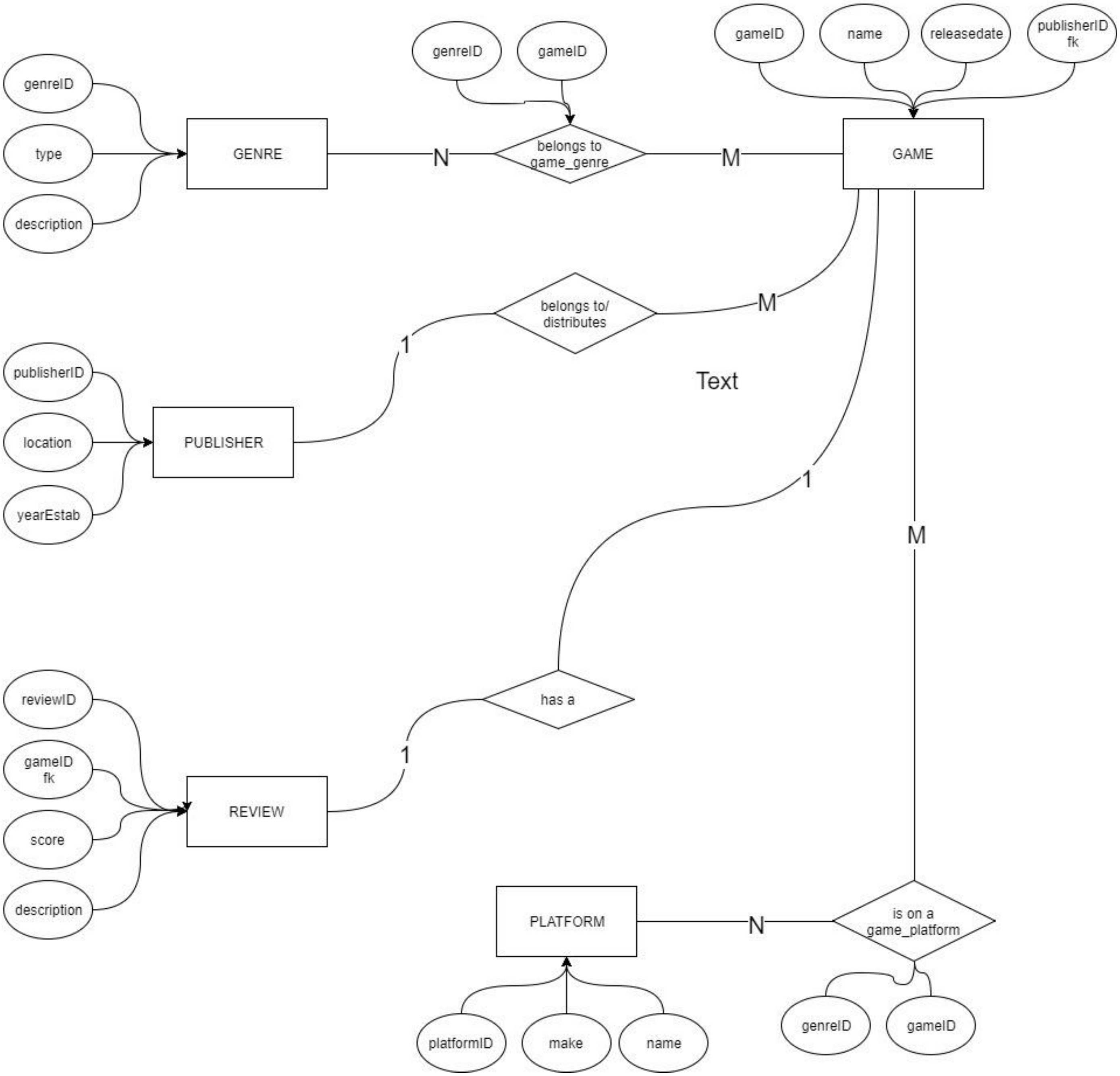
game_platform

platformID(fk)	gameID(fk)

game_genre

genreID(fk)	gameID(fk)

PROJECT DIAGRAM - RELATIONSHIP AND TABLES


genreID
type
description
GENRE
genreID
gameID
belongs to game_genre
N
M
gameID
name
releasedate
publisherID fk
GAME
publisherID
location
yearEstab
PUBLISHER
belongs to/ distributes
1
M
Text
1
M
reviewID
gameID fk
score
description
REVIEW
has a
1
PLATFORM
N
is on a game_platform
platformID
make
name
genreID
gameID