Video Game Database

Byte Thugs -N- Harmony



**By: Trey Chambers, Joe Horton, Catherine Khesin, Kaitlyn Yamamoto, Misha Ward**

Table of Contents

[**1. INTRODUCTION**](#_yspy8tt3f0xe) **2**

[**2. Design**](#_75rf4vta81ax) **2**

[ER Diagram](#_k1fkgcr2lfxw) 2

[Relational Model](#_xn87tuaxpd8q) 4

[Assumptions](#_46y7olc9w581) 5

[Normalization](#_rq5yxrl8chgs) 5

[**3. SQL Statements**](#_yl5t0cn6ae4t) **6**

[Create SQL Database](#_ttr6zs7pgmqp) 6

[Select Query Code and Explanation](#_hqktjmt3z1db) 7

[**4. Evaluation of Project**](#_chou9188p6co) **9**

[Effort](#_6v99se49ue3q) 9

[Successes](#_78eu3px8s1ye) 10

[Areas for Improvements](#_r2j1sd85l3m0) 10

[Learning Outcomes](#_65nb7qednvs) 11

[**5. Appendix (additional Insert, Delete, and Update queries)**](#_vub7j3mujoxb) **12**

[Insert Queries](#_ak0phh541oq3) 12

[Update Queries](#_wlj59fugw4ry) 13

[Delete Queries](#_7usyxvyqaeu) 13

# 

# 1. INTRODUCTION

As a group, we have chosen to create a backend for a website that video game enthusiasts can utilize as a resource to review, rate, and learn about video games on all platforms. Our database will have a variety of information which includes; Game information, Company information, Users information, Reviews information, Genre, and Platform information. The future front end of this database will have the option to create an account, which will collect their information and then allow them to create a username. When a user is on our site they can search for any video game, platform, or company. Once a user does this search they are provided with other users reviews, ratings, price of the game and platform, release date of game and platform and the genre.

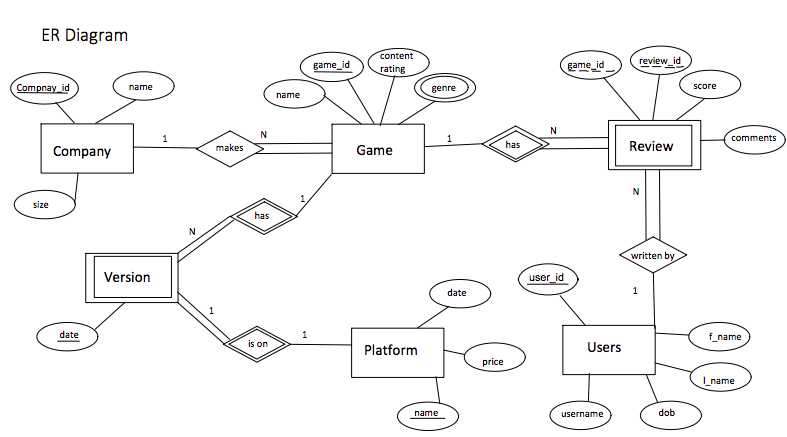
# 2. Design

In order to implement a correct database that meets the needs of the client, we have used design techniques such as the Entity Relationship Diagram and Relational Model. These two techniques have helped us find the optimal way of storing the data for the Video Game Review site. In this section, you will find our ER Diagram, Relational Model, Assumptions, and efforts at Normalization we made during the implementation of this project.

## ER Diagram

To create our database, our team first developed an Entity Relationship Diagram to map out all the entities and the relationships between them. The first entity that we came up with was Game and worked our way from there after brainstorming what pertinent information would be needed by the user. During this process, we discovered other tables would need to be added to meet higher normalization forms that are required by the client. Below is the complete ER diagram under Diagram 1 that shows all the main tables. It should be noted that the Game\_Genre and Genre tables are part of the composite genre attribute in Game.

**Diagram 1:**  ER Diagram Showing Entity Relationships for Game Review Database

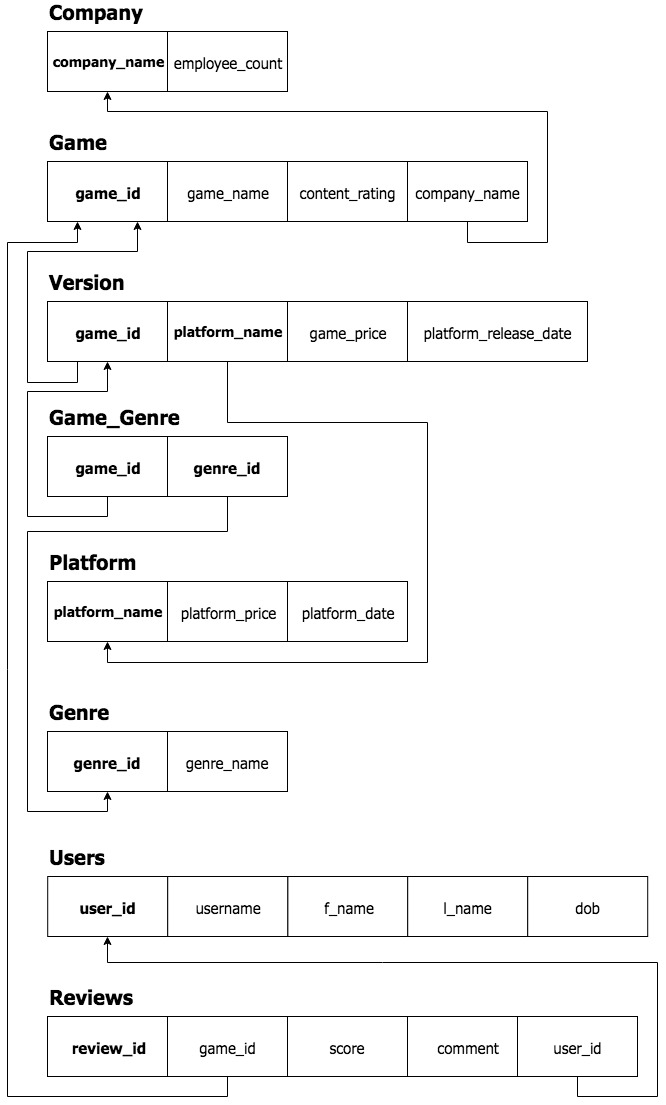


## 

## Relational Model

Building off the ER diagram, our team used the 7 basic steps used by database professionals to create our Relational Model. By walking meticulously through each step, we were able to create Diagram 2 which shows the Relational Model for our complete database.

**Diagram 2:** Relational Model Showing how the Tables Relate to Each Other



## 

## Assumptions

While implementing our ER diagram and Relational Models, we determined a few assumptions that were needed to declare and explain. It should be noted that *none* of these assumptions changes the proposal that we developed but how the tables work together. Below in Table 1 are the assumptions and why we are making them.

**Table 1:** Assumptions for the Video Game Database and Reasoning for them

|  |  |
| --- | --- |
| Assumptions | Reasoning |
| Companies can exist without having made a game. | We decided that it made sense for a company to exist in our database regardless of whether or not they have made a game. This made sense because a company that has not published their first game yet still exists. |
| A game can have more than one genre associated with it. | We made this assumption because many games fall under multiple genre categories, such as a “fantasy” game that is also a “first person shooter”. |
| A game can have different prices and release dates for each platform | Some games come in modified versions for different platforms, such as a game that has a mobile version and a console version. Different versions occasionally have varying prices for games. |

## 

## Normalization

During the early stages of our project, we started brainstorming ideas for what our database tables should consist of, illustrated in Table 2. Before normalization, there were non-key attributes determining other non-key attributes, which meant there were attributes not dependent on the full primary key. The Game and Version tables are examples of this, which were initially in third normal form (3NF) and second normal form (2NF), respectively. After performing normalization, we were able to produce cleaner tables, all in Boyce-Codd normal form (BCNF), displayed in Table 3.

**Table 2:** Database Tables Prior to Normalization

|  |  |
| --- | --- |
| Database Table | Functional Dependencies |
| Company | company\_name -> employee\_count |
| Game | game\_id -> game\_name, content\_rating, company\_name, genre\_id  genre\_id -> genre\_name |
| Version | game\_id, platform\_name -> platform\_release\_date, game\_price  platform\_name -> platform\_price, platform\_date |
| User | user\_id -> username, f\_name, l\_name, Dob |
| Review | review\_id -> game\_id, score, comment, user\_id |

**Table 3:** Database Tables After Normalization

|  |  |
| --- | --- |
| Database Table | Functional Dependencies |
| Company | company\_name -> employee\_count |
| Game | game\_id -> game\_name, content\_rating, company\_name |
| Genre | genre\_id -> genre\_name |
| Game\_Genre | genre\_id, game\_id |
| Platform | platform\_name -> platform\_price, platform\_date |
| Version | game\_id, platform\_name -> platform\_release\_date, game\_price |
| User | user\_id -> username, f\_name, l\_name, Dob |
| Review | review\_id -> game\_id, score, comment, user\_id |

# 3. SQL Statements

## Create SQL Database

The following code is how we created the tables in our database. We have six different entities that have been coded into tables which can be found in Table 2. Our tables are Game, Company, Game\_Genre, Genre, Version, Platform, Users, and Reviews:

**Table 4:** Create SQL Database Queries

|  |
| --- |
| Create Tables |
| CREATE TABLE Game\_Genre (  genre\_id INT NOT NULL,  game\_id INT NOT NULL,  PRIMARY KEY (game\_id, genre\_id)  FOREIGN KEY (game\_id) REFERENCES Game (game\_id) on update cascade on delete cascade,  FOREIGN KEY (genre\_id) REFERENCES Genre (genre\_id) on update cascade on delete cascade  ); |
| CREATE TABLE Genre (  genre\_name VARCHAR(15) NOT NULL,  genre\_id INT NOT NULL,  PRIMARY KEY (genre\_id)  ); |
| CREATE TABLE Version (  game\_id NOT NULL,  platform\_name NOT NULL,  platform\_release\_date DATE NOT NULL,  game\_price DECIMAL(5,2) NOT NULL,  CHECK (game\_price >= 0 AND game\_price <= 999.99),  PRIMARY KEY (platform\_name**,** game\_id)  FOREIGN KEY (game\_id) REFERENCES Game (game\_id) on update cascade on delete cascade,  FOREIGN KEY (platform\_name) REFERENCES Platform (platform\_name) on update cascade on delete cascade  ); |
| CREATE TABLE Game (  game\_id INT NOT NULL,  game\_name VARCHAR(100),  content\_rating VARCHAR(3),  company\_name VARCHAR(100),  PRIMARY KEY (game\_id)  FOREIGN KEY (company\_name) REFERENCES Company (company\_name) on update cascade on delete set null,  CHECK (content\_rating IN ('E', 'E10', 'T', 'M', 'A'))  ); |
| CREATE TABLE Platform (  platform\_name VARCHAR (20) NOT NULL,  platform\_price DECIMAL(9,2),  platform\_date DATE,  PRIMARY KEY (platform\_name)  CHECK (platform\_price > 0)  ); |
| CREATE TABLE Company (  company\_name VARCHAR(100) NOT NULL,  employee\_count INT,  PRIMARY KEY (company\_name)  ); |
| CREATE TABLE Users (  user\_id INT,  username VARCHAR(15) UNIQUE NOT NULL,  f\_name VARCHAR(100) NOT NULL,  l\_name VARCHAR(100) NOT NULL,  Dob DATE NOT NULL,  PRIMARY KEY (user\_id)  ); |
| CREATE TABLE Reviews (  review\_id INT,  game\_id INT,  score INT,  comment VARCHAR(160),  user\_id INT,  FOREIGN KEY (user\_id) REFERENCES Users (user\_id),  FOREIGN KEY (game\_id) REFERENCES Game (game\_id),  PRIMARY KEY(review\_id)  ); |

## 

## Select Query Code and Explanation

The queries we included are designed to organize games based on platform, price, release date, rating, or any combination of the above. Below is a list of the proposal queries and Table 3 which shows the SQL Queries, Purpose of the query, and Result of the query.

Initial Proposal Queries (with footnote if slightly changed in table below):

1. List all horror genre games with a rating of 85% or higher that can be played on a PS4 [1]
2. List the top 5 users that leave the most reviews
3. List reviews with ratings in a given range (e.g, 40-60%) for a specific game [2]
4. List shooting games with ratings of 90% or higher that came out in the last two years [1]
5. List the top 5 games by score
6. Show all related games by genre to a certain game. [3]
7. List all games created by developers from Blizzard Entertainment [4]
8. Show reviews that user (Insert username) has created on website

**Table 5: SQL Queries and Purpose**

|  |  |  |
| --- | --- | --- |
| SQL Query Statements | Purpose | Result |
| SELECT g.game\_name, AVG(rr.score) as Result  FROM Reviews rr  JOIN Game g on g.game\_id = rr.game\_id  JOIN Game\_genre gg on gg.game\_id = rr.game\_id  JOIN Version v on v.game\_id = g.game\_id  GROUP BY g.game\_name  HAVING Result >= 4  AND gg.genre\_id = 6  AND v.platform\_name = 'PS4'; | List all ‘FPS’ genre games with an average rating of 4 or higher that can be played on a PS4 [1] | game\_name Result  ---------- ----------  Battlefield 1 5  Destiny 2 5  Star Wars Battlefront 2 5 |
| SELECT u.username, COUNT(r.comment) as Result  FROM Reviews r  JOIN Users u on u.user\_id = r.user\_id  GROUP BY r.user\_id  ORDER BY Result desc limit 5; | Name the top 5 users with the most reviews | username Result  ---------- ----------  normalAsForm 4  funnyUserName 3  uwMascot 2  DungeonsNBagons 2  littleBear 1 |
| SELECT g.game\_name, r.score, r.comment  FROM Game g  JOIN Reviews r on r.game\_id = g.game\_id  WHERE g.game\_id = 4 AND  r.score >= 2 AND  r.score <= 4  GROUP BY r.review\_id; | List reviews with ratings in a given range (e.g, 2-4) for Fortnite [2][3] | game\_name r.score r.comment  ---------- ------------ ----------------  Fortnite|2|I do not get what all the hype is about. Okay at best!  Fortnite|4|Enjoyable but causes extreme addiction and insomnia. Currently seeking professional help. Thanks, Fortnite. |
| SELECT g.game\_name, AVG(rr.score) as Result, v.platform\_release\_date  FROM Reviews rr  JOIN Game g on g.game\_id = rr.game\_id  JOIN Game\_genre gg on gg.game\_id = rr.game\_id  JOIN Version v on v.game\_id = g.game\_id  GROUP BY g.game\_name  HAVING Result >= 4.5  AND gg.genre\_id = 6  AND v.platform\_release\_date > datetime('now','-2 years'); | List ‘FPS’ games with an average rating of 4.5 or higher that came out in the last 2 years [1] | game\_name AVG( r.score) release\_date  ------------------ ------------------- ------------------  Battlefield 1|5.0|2016-10-21  Destiny 2|5.0|2017-09-06  Star Wars Battlefront 2|5.0|2017-11-17 |
| SELECT AVG(r.score) as Result, g.game\_name  FROM Game g, Reviews r  WHERE g.game\_id = r.game\_id  GROUP BY game\_name  ORDER BY Result desc limit 5; | List the top 5 games by score | Result game\_name  ---------- -----------------  5.0|Battlefield 1  5.0|Destiny 2  5.0|League of Legends  5.0|Star Wars Battlefront 2  4.0|Grand Theft Auto |
| SELECT distinct game\_name FROM ( SELECT genre\_name as Genre FROM Game g, Genre ge, Version v, Game\_Genre gg WHERE g.game\_name = 'Halo' and  v.game\_id = g.game\_id and  gg.game\_id = v.game\_id and   ge.genre\_id = gg.genre\_id ) a , Game g, Genre ge, Version v, Game\_Genre gg WHERE v.game\_id = g.game\_id and gg.game\_id = v.game\_id and  ge.genre\_id = gg.genre\_id and ge.genre\_name = a.Genre; | Show all games related by genre to Halo [3] | Game\_name  -----------------  Fortnite  Grand Theft Auto  Call of Duty  Halo 2  Halo |
| SELECT game\_name, company\_name FROM Game WHERE company\_name = 'Microsoft'; | List all games created by Microsoft [4] | game\_name company\_name  ----------------- -----------------  Halo 2 Microsoft  Halo Microsoft  Minecraft Microsoft |
| SELECT r.comment  FROM Reviews r  JOIN Users u on u.user\_id = r.user\_id  WHERE u.username = 'funnyUserName'; | Show reviews that user ‘funnyUserName’ has created on website | -Very nice! \*in Borat voice.  -Legen--wait for it--dary! Legendary.  -Enjoyable but causes extreme addiction and insomnia. Currently seeking professional help. Thanks, Fortnite. |

Notes:

[1] Modified from project proposal because a specific genre for shooting in the database is called ‘FPS’ for first-person shooter, and we changed our game ratings from a percent scale to a 5-star scale.

[2] Modified from project proposal because we changed our game ratings from a percent scale to a 5-star scale.

[3] Modified from project proposal because we needed to choose a specific game to retrieve games of the same genre.

[4] Modified from project proposal because we decided to not include Blizzard Entertainment in our database.

# 4. Evaluation of Project

Although the implementation of the project is important, it's also critical to review how the project went and provide an evaluation of the overall project. Our evaluation of this project consists of understanding the Effort and Successes that we had while also identifying Areas for Improvement and Learning Outcomes. One can find the complete analysis of these areas below.

## Effort

The effort put forth into this project by the team was significant. Our team believes that we met and surpassed the high standards that we set out to achieve in our Proposal, Report, and Presentation. Below is a summary of each deliverable and an overview of the effort our team put into it.

*Proposal:* Our project proposal was a culmination of thoughtful discourse and collaboration on what we as a team wanted to achieve. Prior to the submission of the proposal, our team met multiple times to discuss what we wanted to do and how we would produce a high quality product. We believe that the acceptance of the proposal and ability to stick to the original proposal shows the effort we put into it.

*Report:* As with the proposal, the report shows the diligent work that our team has approached this project. Prior to starting any SQL, our team met multiple times to plan exactly what the database would need. Starting with a Entity Relationship diagram and moving on to the Relational Model, we systematically built out the database. The report is the culmination of the effort put into the database and shows the thoughtfulness of design and high quality work of the implementation. Finally, the report is retrospective as we wanted to review how the project went and really provide insights on what we could do better.

*Presentation:* Once the database was constructed and tested, our team worked on the PowerPoint presentation to showcase our final product. The presentation covers all aspects of the project from our design, to our analysis of the SQL statements, to evaluation of the project. We hope that the presentation shows our commitment to the database and shows how we achieved our goals set out in the proposal in a clear and concise manner.

## Successes

When reviewing this project, our team believes that overall, our project was very successful. From the effort put into making a great final product, to the written report which highlights the design and implementation of the database, to the presentation that explains what we did and why our team took that approach. Each element of the project was completed with consideration for quality and professionally done.

## Areas for Improvements

Although we believe the project was successful, there are still areas of improvement. The main area that we could work on is being more specific on who would work on what part of the project from the onset of the project. The lack of clear communication caused some confusion on who would work on what. Additionally, in several instances it was not clear who worked on what and making the distribution of more difficult. Finally, the team believes that a team schedule would have helped with all members attending working sessions. To prevent such issues, we believe that having a team leader that helped coordinate the progress of the project and delegate roles would have been beneficial.

## Learning Outcomes

This project helped our team learn more about database design, implementation, teamwork, and collaboration. Specifically, this project helped us understand how to start from just an idea and then figure out ways to implement our vision. In the process, we learned how to overcome the difficulty of making ER diagram from just an idea, to implementing that idea from a Relational Model. Finally, we learned that it can be quite fun creating a powerful database from scratch.

# 

# 

# 5. Appendix (additional Insert, Delete, and Update queries)

Although we originally planned on only using SQL select queries in our proposal, we decided that the client should be able to utilize Insert, Update, and Delete queries as well. The supplemental Insert, Update, and Delete queries are shown below showing a Insert and Update script for each table in our database.

## Insert Queries

INSERT into Company values ('Microsoft', 124000), ('Riot Games', 2500), ('Epic Games', 700),

('Activision', 4000), ('Rockstar Games', 1000), ('Bungie', 750), ('EA', 9300);

INSERT into Game values (1, 'Halo 2', 'M', 'Microsoft'), (2, 'Halo', 'M', 'Microsoft'), (3, 'League of Legends', 'T', 'Riot Games'), (4, 'Fortnite', 'T', 'Epic Games'), (5, 'Grand Theft Auto', 'A', 'Rockstar Games'), (6, 'Call of Duty', 'M', 'Activision'),

(7, 'Minecraft', 'E', 'Microsoft') , (8, 'Battlefield 1', 'M', 'EA'), (9, 'Star Wars Battlefront 2', 'T', 'EA'), (10, 'Destiny 2', 'T', 'Bungie');  
  
INSERT INTO Users values

(1, 'mishsov09', 'Misha', 'Ward', '1990-09-20'), (2, 'littleBear', 'Wheres', 'Waldo', '1995-03-10'),

(3, 'funnyUserName', 'Laurel', 'Yanny', '1990-14-20'), (4, 'uwMascot', 'Harry', 'the Husky', '1891-05-25'),

(5, 'dBMaster', 'Harold', 'Hacks', '1998-10-21'), (6, 'normalAsForm', 'Boyce', 'Codd', '1996-01-25'),

(7, 'DungeonsNBagons', 'Will', 'Byers', '1999-07-19');  
  
INSERT INTO Reviews values  
(1, 2, 4, 'Very nice! \*in Borat voice', 3), (2, 3, 5, 'Legen--wait for it--dary! Legendary.', 3), (3, 4, 2, 'I do not get what all the hype is about. Okay at best!', 4), (4, 5, 4, 'Thrilling. Like Breaking Bad IRL, without the dead bodies.', 4),   
(5, 4, 5, 'Love this game. Hold on while I hop on the bandwagon. \*hop', 2), (6, 4, 4, 'Enjoyable but causes extreme addiction and insomnia. Currently seeking professional help. Thanks, Fortnite.', 3), (7, 8, 5, 'Best game ever!', 6),

(8, 8, 5, 'Highly recommend!', 7), (9, 9, 5, 'This is a must try!', 6), (10, 9, 5, 'Gotta have it!', 7), (11, 10, 5, 'You wont regret it!', 6), (12, 10, 5, 'Get it now!', 6);  
  
INSERT INTO Genre values   
('Action', 1), ('RPG', 2), ('Fantasy', 3), ('Horror', 4), ('Strategy', 5), ('FPS', 6), ('TPS', 7);  
  
INSERT INTO Game\_Genre values(1, 1), (6, 1), (1, 2), (6, 2), (5, 3),   
(3, 3), (5, 4), (1, 4), (7, 4), (2, 5), (1, 5), (6, 6), (1, 6), (2, 7), (6, 8), (6, 9), (6, 10);  
  
INSERT INTO Platform values  
('Xbox', 39.99, '2015-11-15'), ('Xbox One X', 479.97, '2017-11-07'), ('Xbox 360', 109.99, '2013-11-22'),   
('Xbox One', 259.99, '2016-08-01'), ('PS4', 322, '2013-11-15'), ('Switch', 299.99, '2017-03-03'), ('PC', NULL, NULL);  
  
INSERT into Version values (2, 'Xbox', '2001-11-15', 20.00), (1, 'Xbox', '2004-11-04', 30.00), (1, 'Xbox 360', '2004-11-04', 30.00), (3, 'PC', '2009-10-07', 0), (4, 'PC', '2017-07-25', 59.99), (4, 'PS4', '2017-07-25', 59.99), (4, 'Xbox One', '2017-07-25', 59.99), (5, 'Xbox One', '2017-07-25', 59.99), (5, 'Switch', '2013-09-17', 59.99), (5, 'PS4', '2013-09-17', 59.99), (5, 'PC', '2013-09-17', 59.99), (7, 'PC', '2008-04-20', 9.99), (6, 'Xbox One', '2011-11-11', 59.99), (6, 'PS4', '2011-11-11', 59.99), (6, 'Switch', '2011-11-11', 59.99), (8, 'PS4', '2016-10-21', 59.99), (9, 'PS4', '2017-11-17', 59.99), (10, 'PS4', '2017-09-06', 59.99);

## Update Queries

UPDATE Company  
SET employee\_count = 125000  
WHERE company\_name = 'Microsoft';  
  
UPDATE Game  
SET content\_rating = 'M'  
WHERE game\_name = 'Fortnite';  
  
UPDATE Game\_Genre  
SET genre\_id = 3  
WHERE genre\_id = 1;  
  
UPDATE Genre  
SET genre\_name = 'First Person Shooter'  
WHERE genre\_id = 6;  
  
UPDATE Platform  
SET platform\_price = 469.97  
WHERE platform\_name = 'Xbox One X';  
  
UPDATE Reviews  
SET score = 4  
WHERE review\_id = 13;  
  
UPDATE Users  
SET username = 'stripedAssassin'  
WHERE f\_name = 'Wheres' AND l\_name = 'Waldo';  
  
UPDATE Version  
SET game\_price = 45.99  
WHERE game\_id = 6 AND platform\_name = 'Switch

## Delete Queries

DELETE FROM Company  
WHERE company\_name = ‘Microsoft’;

DELETE FROM Game\_Genre  
WHERE genre\_id = 1;

DELETE FROM Genre  
WHERE genre\_id = 5;

DELETE FROM Platform  
WHERE platform\_name = 'Xbox';

DELETE FROM Reviews  
WHERE review\_id = 6;

DELETE FROM Users  
WHERE username = ‘mishsov09’;

DELETE FROM Version  
WHERE game\_price = 0;