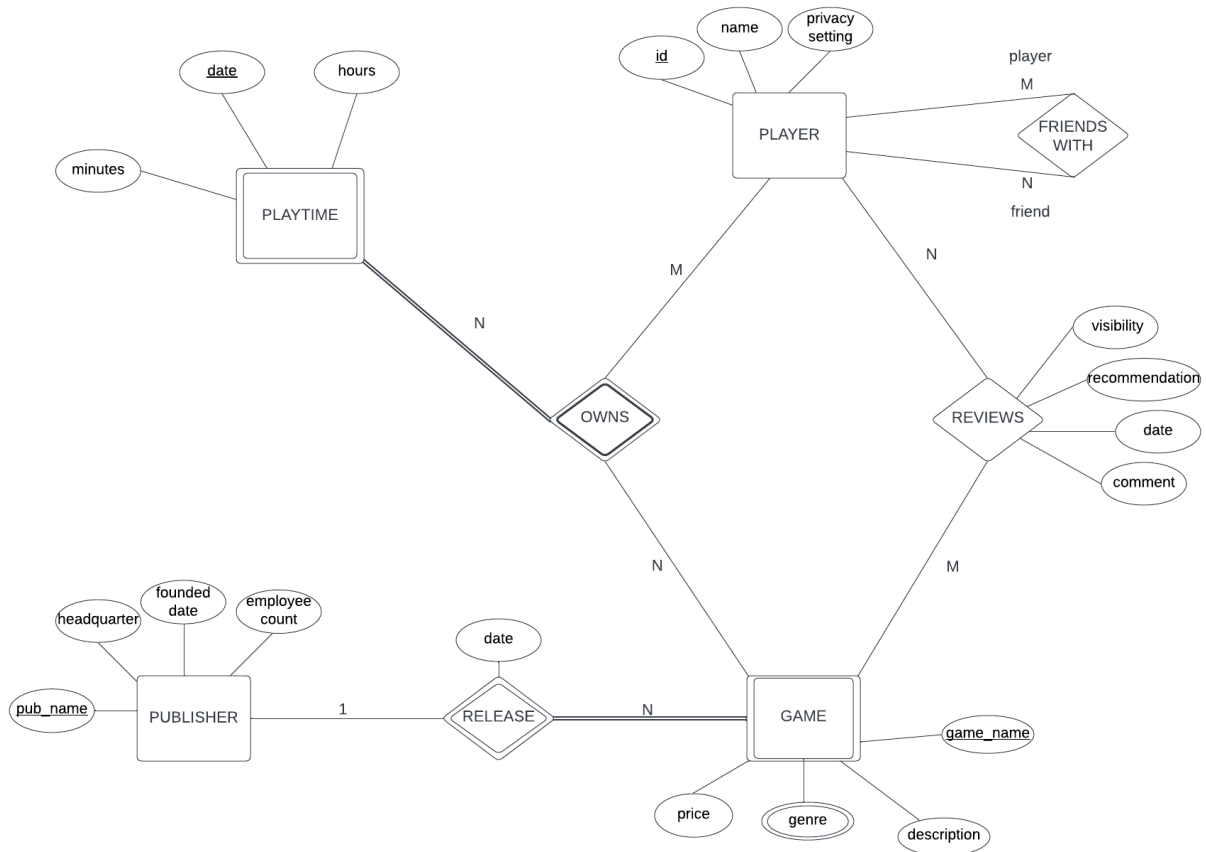


## Final Report

### ER Diagram:



### Relations:

Publisher(pub\_name, headquarter, founded\_date, employee\_count)

Game(game\_name, pub\_name, description, price, release\_date)

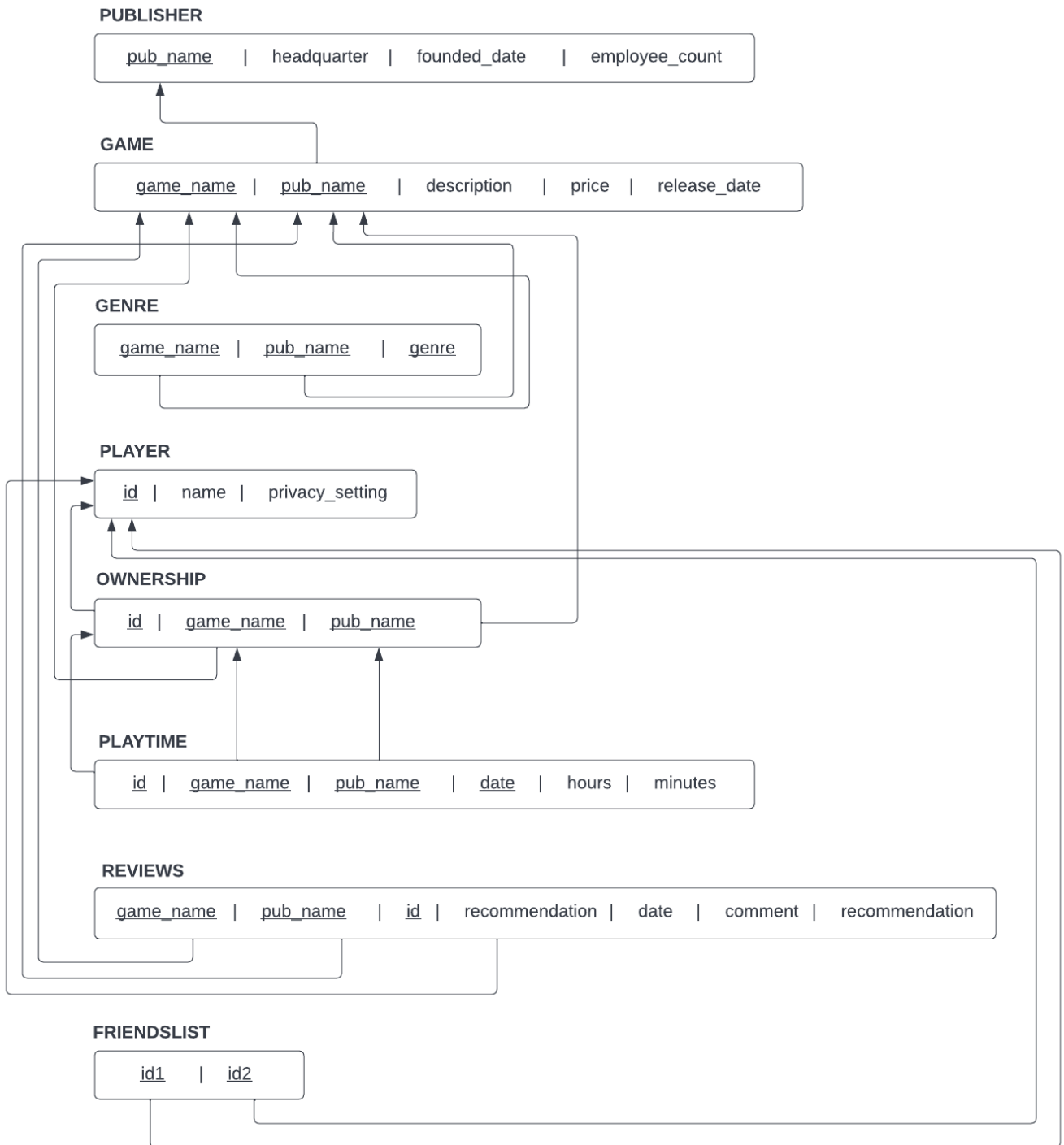
Genre(game\_name, pub\_name, genre)

Player(id, name, privacy\_setting)

Ownership(id, game\_name, pub\_name)

Playtime(id, game\_name, pub\_name, date, hours, minutes)

Review(game\_name, pub\_name, id, recommendation, date, comment)



SQL Statement	Purpose
<pre> CREATE VIEW Ratings AS SELECT y.game_name, y.pub_name, SUM(yes_count) + SUM(no_count) AS num_reviews,     CASE         WHEN SUM(yes_count) / ((SUM(yes_count) + SUM(no_count)) * 1.0) &gt;= 0.8 THEN 'very positive'         WHEN SUM(yes_count) / ((SUM(yes_count) + SUM(no_count)) * 1.0) &gt;= 0.6 THEN 'mostly positive'         WHEN SUM(yes_count) / ((SUM(yes_count) + SUM(no_count)) * 1.0) &gt;= 0.4 THEN 'mixed'         WHEN SUM(yes_count) / ((SUM(yes_count) + SUM(no_count)) * 1.0) &gt;= 0.2 THEN 'mostly negative'         ELSE 'very negative' END AS rating FROM     (         SELECT game_name, pub_name, COUNT(1) as yes_count, 0 AS no_count         FROM Review         WHERE recommendation = 'yes'         GROUP BY game_name, pub_name         UNION         SELECT game_name, pub_name, 0 AS yes_count, COUNT(1) as no_count         FROM Review         WHERE recommendation = 'no'         GROUP BY game_name, pub_name     ) y GROUP BY y.game_name, y.pub_name; </pre>	<p>A view that shows the overall rating of a game</p> <p>&gt;=80% 'yes' is 'very positive'</p> <p>&gt;=60% 'yes' is 'mostly positive'</p> <p>&gt;=40% 'yes' is 'mixed'</p> <p>&gt;=20% is 'mostly negative', else 'very negative'</p>
<pre> CREATE VIEW TotalTime AS SELECT t.id AS pid, name, game_name AS game, pub_name AS publisher,     CAST((SUM(hours) * 60 + SUM(minutes)) / 60 AS INT) AS total_hours,     CAST(SUM(hours) * 60 + SUM(minutes) - CAST((SUM(hours) * 60 + SUM(minutes)) / 60 AS INT) * 60 AS INT) AS total_mins FROM Playtime t, Player p WHERE t.id = p.id GROUP BY pid, game, publisher UNION SELECT o.id AS pid, name, o.game_name AS game, o.pub_name AS publisher, 0 AS total_hours, 0 AS total_minutes </pre>	<p>A view that shows the sum of all the playtime for each player, and the playtime is 0 if a player owns the game but never played it</p>

FROM Player p, Ownership o LEFT JOIN Playtime t ON o.game_name = t.game_name and o.pub_name = t.pub_name WHERE t.hours IS NULL and t.minutes IS NULL and p.id = o.id ORDER BY pid, game;	
SELECT game_name, name AS player_name, recommendation, comment FROM Review r, Player p WHERE visibility = 'public' AND r.id = p.id AND r.id != 'EFGH5678' UNION ALL SELECT game_name, name AS player_name, recommendation, comment FROM Review r, Player p, FriendsList f, ( SELECT id1 AS id FROM FriendsList WHERE id2 = 'EFGH5678' UNION SELECT id2 AS id FROM FriendsList WHERE id1 = 'EFGH5678' )i WHERE visibility = 'private' AND r.id = p.id AND r.id IN (i.id) GROUP BY game_name, pub_name;	Select all the reviews that is visible to player id 'EFGH5678' but not reviewed by himself ( a review is visible to a player if the visibility is public or a player is friend with the reviewer)
SELECT p.name, p.id FROM Ownership o, Player p WHERE o.id = p.id AND o.pub_name = 'Innersloth' AND o.game_name = 'Among Us';	Select all of the players who owns the game 'Among Us'
WITH Recent AS ( SELECT date, id, hours, minutes FROM Playtime t WHERE JULIANDAY('2022-12-03') - JULIANDAY(t.date) <= 14 ) SELECT r.id, p.name, CAST((SUM(hours) * 60 + SUM(minutes)) / 60 AS INT) AS two_weeks_hours, CAST(SUM(hours) * 60 + SUM(minutes) - CAST((SUM(hours) * 60 + SUM(minutes)) / 60 AS INT) * 60 AS INT) AS two_week_mins	Select the sum of playtime in the recent 2 weeks from 22/12/03 for each player

FROM Recent r, Player p WHERE p.id = r.id GROUP BY r.id ORDER BY r.id;	
SELECT id1 AS PlayerId, p1.name AS PlayerName, id2 AS FriendId, p2.name AS FriendsName FROM FriendsList f, Player p1, Player p2 WHERE p1.id = f.id1 AND p2.id = f.id2 UNION SELECT id2 AS PlayerId, p2.name AS PlayerName, id1 AS FriendId, p1.name AS FriendsName FROM FriendsList f, Player p1, Player p2 WHERE p2.id = f.id2 AND p1.id = f.id1 GROUP BY PlayerId;	Select all the friends of each player
SELECT game, publisher, total_hours, total_mins FROM TotalTime WHERE pid = 'EFGH5678' ORDER BY total_hours DESC, total_mins DESC;	Select the games that player with the id 'EFGH5678' owns and order by the number of hours played
INSERT INTO Ownership VALUES ('ABCD1234', 'COD:MW2', 'Activision');	Player id 'ABCD1234' gets 'COD:MW2'
DELETE FROM Ownership WHERE id = 'ABCD1234' AND game_name = 'COD:MW2' AND pub_name = 'Activision';	Delete 'COD:MW2' from player with the id 'ABCD1234'
SELECT game, publisher, CAST((AVG(total_hours) * 60 + AVG(total_mins)) / 60 AS INT) AS Avg_hours, CAST(AVG(total_hours) * 60 + AVG(total_mins) - (CAST((AVG(total_hours) * 60 + AVG(total_mins)) / 60 AS INT) * 60) AS INT) AS Avg_mins FROM TotalTime WHERE game = 'Among Us' AND publisher = 'Innersloth';	Select the average playtime for all players that own 'Among Us'

SELECT * FROM GAME WHERE price = (SELECT MAX(price) FROM Game);	Select the game with the max price
UPDATE Playtime SET minutes = 30 WHERE id = '0112358' AND game_name = 'Among Us' AND date = '2021-12-26';	Update the time played for the player with id '0112358' who played 'Among Us' on '2021-12-26'
SELECT * FROM Ratings WHERE rating = 'very positive' OR rating = 'mostly positive';	Select all the games that are rated 'very positive' or 'mostly positive'
SELECT name, COUNT(1) AS GamesOwned FROM Player p, Ownership o WHERE p.id = o.id AND p.privacy_setting = 'public' GROUP BY p.id UNION SELECT name, COUNT(1) AS GamesOwned FROM Player p, Ownership o, ( SELECT id1 AS id FROM FriendsList WHERE id2 = 'EFGH5678' UNION SELECT id2 AS id FROM FriendsList WHERE id1 = 'EFGH5678' )i WHERE p.id = o.id AND p.privacy_setting = 'private' AND o.id in (i.id) GROUP BY p.id;	Show all the players and number of games they owned who have their profiles set to public or is a friend of player id 'EFGH5678'

## Table Normalization:

### Publisher:

{pub\_name, headquarter, founded\_date, employee\_count}

FD:

Pub\_name -> {headquarter, founded\_date, employee\_count}

The publisher table is in BCNF form

### Game:

{game\_name, pub\_name, description, price, release\_date}

FD:

game\_name, pub\_name -> {description, price, release\_date}

The game table is in BCNF form

### Genre:

{game\_name, pub\_name, genre}

FD:

game\_name, pub\_name -> {genre}

The genre table is in BCNF form

### Player:

{id, name, privacy\_setting}

FD:

Id -> {name, privacy\_setting}

The player table is in BCNF form

### Ownership:

{id, game\_name, pub\_name}

FD:

id, game\_name, pub\_name -> {}

The ownership table is in BCNF form

### Playtime:

{id, game\_name, pub\_name, date, hours, mins}

FD:

id, game\_name, pub\_name, date -> {hours, mins}

The playtime table is in BCNF form

**Reviews:**

{id, game\_name, pub\_name, date, recommendation, comment }

FD:

id, game\_name, pub\_name -> { date, recommendation, comment }

The reviews table is in BCNF form

**Friendslist:**

{id1, id2}

FD:

id1, id2 -> {}

The FriendsList table is in BCNF form