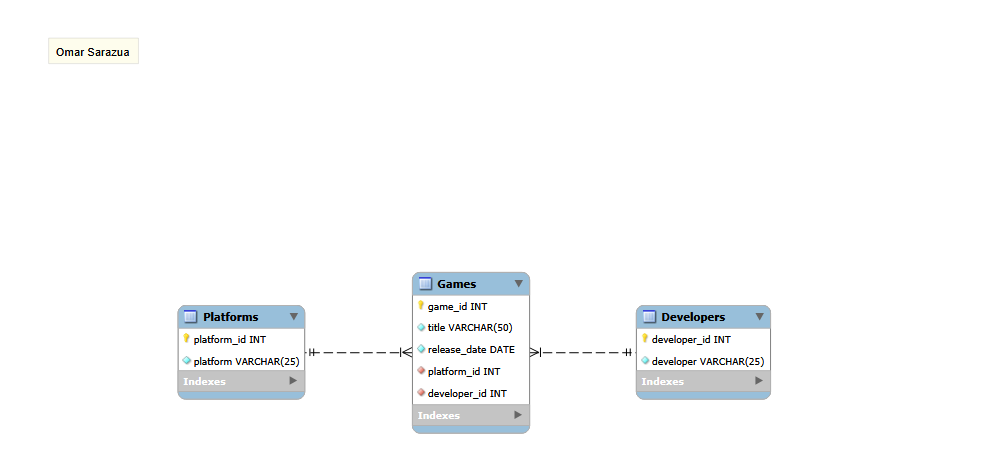
## Game Database Project

Omar Sarazua

My database is a game database that stores the bestselling video games along with their developers and available platforms. This is done through the Games, Platforms, and Developers tables. Platforms and Developers have a PlatformID or DeveloperID as their primary keys while only the Games table uses any foreign keys. Along with the GameID primary key the Games table has an ID key to link either Platforms or Developers respectively. This creates two one to many relationships where a platform can have many game’s but a game can have only platform, and where a developer can have many games, but a game can only have one developer. The Games table functions as a junction table between Platforms and Developers. A platform can have many games released on it and developers can release many games, and often games are released on more than one platform. As such, breaking up the otherwise many to many relationships this way (and providing data for multiple platforms) is necessary to normalize the tables and data.

## The EER Diagram



## Database Project Data Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Name** | **column Name** | **DataType(length)** | **Required (Y/N)** | **Primary Key (Y/N)** | **Foreign Key (Y/N)** | **References** | **Attribute** |
| platforms | platform\_id | int | yes | yes | no |  | auto\_increment |
| platforms | platform | varchar(25) | yes | no | no |  |  |
| Games | game\_id | int | yes | yes | no |  | auto\_increment |
| Games | title | varchar(50) | yes | no | no |  |  |
| games | release\_date | Date | yes | no | no |  |  |
| games | developer\_id | int | yes | no | yes | developers |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Name** | **column Name** | **DataType(length)** | **Required (Y/N)** | **Primary Key (Y/N)** | **Foreign Key (Y/N)** | **References** | **Attribute** |
| games | platform\_id | int | yes | no | yes | platforms |  |
| develeopers | developer\_id | int | yes | yes | no |  | auto\_increment |
| developers | developer | varchar(25) | yes | no | no |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

## MySQL DDL Script

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Created by Omar Sarazua on 10/28/2024 using MySQL

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

USE sys;

DROP DATABASE IF EXISTS game\_os;

CREATE DATABASE game\_os;

USE game\_os;

CREATE TABLE platforms (

platform\_id INT NOT NULL AUTO\_INCREMENT,

platform VARCHAR(25) NOT NULL,

CONSTRAINT platforms\_id\_PK PRIMARY KEY (platform\_id)

);

CREATE TABLE developers (

developer\_id INT NOT NULL AUTO\_INCREMENT,

developer VARCHAR(25) NOT NULL,

CONSTRAINT developers\_id\_PK PRIMARY KEY (developer\_id)

);

CREATE TABLE games (

game\_id INT NOT NULL AUTO\_INCREMENT,

title VARCHAR(50) NOT NULL,

release\_date DATE NOT NULL,

developer\_id INT NOT NULL,

platform\_id INT NOT NULL,

CONSTRAINT games\_id\_PK PRIMARY KEY (game\_id),

CONSTRAINT games\_developer\_id\_FK FOREIGN KEY (developer\_id) REFERENCES developers (developer\_id),

CONSTRAINT games\_platform\_id\_FK FOREIGN KEY (platform\_id) REFERENCES platforms (platform\_id)

);

INSERT INTO platforms

(platform)

VALUES

("Multi-platform"),

("Wii"),

("Wii U"),

("Switch"),

("NES"),

("SNES"),

("Game Boy"),

("Game Boy Color"),

("Game Boy Advance"),

("DS"),

("3DS"),

("Xbox 360"),

("PlayStation 3"),

("PlayStation 4"),

("Windows");

INSERT INTO developers

(developer)

VALUES

("Nintendo"),

("Xbox Game Studios"),

("Mojang Studios"),

("Rockstar Games"),

("PUBG Studios"),

("Re-Logic"),

("505 Games"),

("Blizzard Entertainment"),

("No Breaks Games"),

("CD Projekt"),

("Game Freak"),

("ConcernedApe"),

("Bandai Namco"),

("Sora Ltd."),

("Infinity Ward"),

("Telltale Games"),

("Bethesda Game Studios");

INSERT INTO games

(title, release\_date, developer\_id, platform\_id)

VALUES

("Minecraft", "2011-11-18", 3, 1),

("Grand Theft Auto V", "2013-09-17", 4, 1),

("Wii Sports", "2006-11-19", 1, 2),

("PUBG: Battlegrounds", "2017-12-20", 5, 1),

("Mario Kart 8", "2014-05-29", 1, 4),

("Red Dead Redemption 2", "2018-10-26", 4, 1),

("Terraria", "2011-05-16", 6, 1),

("Super Mario Bros.", "1985-09-13", 1, 1),

("Overwatch", "2016-05-24", 8, 1),

("Human: Fall Flat", "2016-07-22", 9, 1),

("The Witcher 3: Wild Hunt", "2015-05-19", 10, 1),

("Animal Crossing: New Horizons", "2020-03-20", 1, 4),

("Wii Fit", "2007-12-01", 1, 2),

("Tetris", "1989-06-14", 1, 1),

("The Elder Scrolls V: Skyrim", "2011-11-11", 17, 1);

## SQL Server DDL Script

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Created by Omar Sarazua on 11/18/24 using Microsoft SQL Server

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

USE master;

GO --Go required to run more than once

DROP DATABASE IF EXISTS GameOS;

CREATE DATABASE GameOS;

GO

USE GameOS;

CREATE TABLE Platforms (

PlatformID INT NOT NULL IDENTITY,

Platform VARCHAR(25) NOT NULL,

CONSTRAINT PK\_Platforms\_PlatformID PRIMARY KEY (PlatformID)

);

CREATE TABLE Developers (

DeveloperID INT NOT NULL IDENTITY,

Developer VARCHAR(25) NOT NULL,

CONSTRAINT PK\_Developers\_DeveloperID PRIMARY KEY (DeveloperID)

);

CREATE TABLE Games (

GameID INT NOT NULL IDENTITY,

Title VARCHAR(50) NOT NULL,

ReleaseDate DATE NOT NULL,

Price DECIMAL(4,2)NOT NULL,

DeveloperID INT NOT NULL,

PlatformID INT NOT NULL,

CONSTRAINT PK\_Games\_GameID PRIMARY KEY (GameID),

CONSTRAINT FK\_Developers\_Games FOREIGN KEY (DeveloperID) REFERENCES Developers (DeveloperID),

CONSTRAINT FK\_Platforms\_Games FOREIGN KEY (PlatformID) REFERENCES Platforms (PlatformID)

);

INSERT INTO Platforms

(Platform)

VALUES

('Multi-Platform'),

('Wii'),

('Wii U'),

('Switch'),

('NES'),

('SNES'),

('Game Boy'),

('Game Boy Color'),

('Game Boy Advance'),

('DS'),

('3DS'),

('Xbox 360'),

('PlayStation 3'),

('PlayStation 4'),

('Windows');

INSERT INTO Developers

(Developer)

VALUES

('Nintendo'),

('Xbox Game Studios'),

('Mojang Studios'),

('Rockstar Games'),

('PUBG Studios'),

('Re-Logic'),

('505 Games'),

('Blizzard Entertainment'),

('No Breaks Games'),

('CD Projekt'),

('Game Freak'),

('ConcernedApe'),

('Bandai Namco'),

('Sora Ltd.'),

('Infinity Ward'),

('Telltale Games'),

('Bethesda Game Studios');

INSERT INTO Games

(Title, ReleaseDate, Price, DeveloperID, PlatformID)

VALUES

('Minecraft', '2011-11-18', 26.95, 3, 1),

('Grand Theft Auto V', '2013-09-17', 60, 4, 1),

('Wii Sports', '2006-11-19', 20, 1, 2),

('PUBG: Battlegrounds', '2017-12-20', 0, 5, 1),

('Mario Kart 8', '2014-05-29', 59.99, 1, 4),

('Red Dead Redemption 2', '2018-10-26', 59.99, 4, 1),

('Terraria', '2011-05-16', 10, 6, 1),

('Super Mario Bros.', '1985-09-13', 25, 1, 1),

('Overwatch', '2016-05-24', 40, 8, 1),

('Human: Fall Flat', '2016-07-22', 20, 9, 1),

('The Witcher 3: Wild Hunt', '2015-05-19', 40, 10, 1),

('Animal Crossing: New Horizons', '2020-03-20', 59.99, 1, 4),

('Wii Fit', '2007-12-01', 89.99, 1, 2),

('Tetris', '1989-06-14', 50, 1, 1),

('The Elder Scrolls V: Skyrim', '2011-11-11', 59.99, 17, 1);

## Select Queries

/\* Created by Omar Sarazua on 11/25/2024 using SQL Server \*/

USE GameOS;

/\* OUTER join \*/

/\* Who are the developers without any games? \*/

SELECT Developer

FROM Developers AS d LEFT JOIN Games AS g

ON d.DeveloperID = g.DeveloperID

WHERE GameID IS NULL;

/\* Results \*/

|  |
| --- |
| **Developer** |
| **Xbox Game Studios** |
| **505 Games** |
| **Game Freak** |
| **ConcernedApe** |
| **Bandai Namco** |
| **Sora Ltd.** |
| **Infinity Ward** |
| **Telltale Games** |

/\* Subquery \*/

/\* What platforms have games? \*/

SELECT Platform

FROM Platforms

WHERE PlatformID IN

(SELECT DISTINCT PlatformID

FROM Games);

/\* Results \*/

|  |
| --- |
| Platform |
| Multi-Platform |
| Wii |
| Switch |

/\* TOP values \*/

/\* What are the three most recent games? \*/

SELECT TOP 3 Title

FROM Games

ORDER BY ReleaseDate DESC;

/\* Results \*/

|  |
| --- |
| Title |
| Animal Crossing: New Horizons |
| Red Dead Redemption 2 |
| PUBG: Battlegrounds |

/\* Aggregate Function with GROUP BY INNER Join \*/

/\* How many games does each developer in Games have? \*/

SELECT d.Developer, COUNT(g.DeveloperID) AS Games

FROM Developers AS d INNER JOIN Games AS g

ON d.DeveloperID = g.DeveloperID

GROUP BY d.Developer;

/\* Results \*/

|  |  |
| --- | --- |
| Developer | Games |
| Bethesda Game Studios | 1 |
| Blizzard Entertainment | 1 |
| CD Projekt | 1 |
| Mojang Studios | 1 |
| Nintendo | 6 |
| No Breaks Games | 1 |
| PUBG Studios | 1 |
| Re-Logic | 1 |
| Rockstar Games | 2 |

/\* Aggregate Function with GROUP BY and HAVING \*/

/\* What platforms have at least 3 games? \*/

SELECT Platform

FROM Platforms

WHERE PlatformID IN

(SELECT PlatformID

FROM Games

GROUP BY PlatformID

HAVING COUNT(PlatformID) >= 3 );

/\* Results \*/

|  |
| --- |
| Platform |
| Multi-Platform |

/\* Arithmetic OR String Expression \*/

/\* Who made each game for what platform? \*/

SELECT Title, 'By ' + Developer + ', for ' + Platform AS Information

FROM Developers AS d JOIN Games AS g

ON d.DeveloperID = g.DeveloperID JOIN Platforms AS p

ON p.PlatformID = g.PlatformID;

/\* Results \*/

|  |  |
| --- | --- |
| Title | Information |
| Minecraft | By Mojang Studios, for Multi-Platform |
| Grand Theft Auto V | By Rockstar Games, for Multi-Platform |
| Wii Sports | By Nintendo, for Wii |
| PUBG: Battlegrounds | By PUBG Studios, for Multi-Platform |
| Mario Kart 8 | By Nintendo, for Switch |
| Red Dead Redemption 2 | By Rockstar Games, for Multi-Platform |
| Terraria | By Re-Logic, for Multi-Platform |
| Super Mario Bros. | By Nintendo, for Multi-Platform |
| Overwatch | By Blizzard Entertainment, for Multi-Platform |
| Human: Fall Flat | By No Breaks Games, for Multi-Platform |
| The Witcher 3: Wild Hunt | By CD Projekt, for Multi-Platform |
| Animal Crossing: New Horizons | By Nintendo, for Switch |
| Wii Fit | By Nintendo, for Wii |
| Tetris | By Nintendo, for Multi-Platform |
| The Elder Scrolls V: Skyrim | By Bethesda Game Studios, for Multi-Platform |

/\* DATEDIFF Function \*/

/\* How old is each game in years? \*/

SELECT Title, DATEDIFF(year, ReleaseDate, GETDATE()) AS Age

FROM Games;

/\* Results \*/

|  |  |
| --- | --- |
| Title | Age |
| Minecraft | 13 |
| Grand Theft Auto V | 11 |
| Wii Sports | 18 |
| PUBG: Battlegrounds | 7 |
| Mario Kart 8 | 10 |
| Red Dead Redemption 2 | 6 |
| Terraria | 13 |
| Super Mario Bros. | 39 |
| Overwatch | 8 |
| Human: Fall Flat | 8 |
| The Witcher 3: Wild Hunt | 9 |
| Animal Crossing: New Horizons | 4 |
| Wii Fit | 17 |
| Tetris | 35 |
| The Elder Scrolls V: Skyrim | 13 |

## Views

/\* Created by Omar Sarazua on 12/4/2024 using SQL Server \*/

USE GameOS;

GO

/\* CAST or CONVERT function \*/

/\* The purpose of this view is to present the release date for games in a format Americans are used to seeing \*/

CREATE VIEW VW\_GamesReleaseDate AS

SELECT Title, CONVERT(VARCHAR, ReleaseDate, 101) AS AmericanReleaseDate

FROM Games;

GO

/\* Two String Functions \*/

/\* The purpose of this view is to seperate the release year and day \*/

CREATE VIEW VW\_GamesYearDay AS

SELECT Title, LEFT(ReleaseDate, 4) AS Year, RIGHT(ReleaseDate, 2) AS Day

FROM Games;

GO

/\* SUBSTRING Function OUTER \*/

/\* The purpose of this view is to view all platforms without any games \*/

CREATE VIEW VW\_PlatformsWithoutGames AS

SELECT Platform, SUBSTRING(Platform, 1, 4) AS GeneralPlatform

FROM Platforms AS p LEFT JOIN Games AS g

ON p.PlatformID = g.PlatformID

WHERE GameID IS NULL;

GO

/\* ROUND Function with Calculated Value and/or Aggregate Function INNER \*/

/\* The purpose of this view is to provide the average price of a game per developer \*/

CREATE VIEW VW\_GamesPrice (Developer, Games, AveragePrice) AS

SELECT Developer, COUNT(g.DeveloperID), ROUND(AVG(Price), 0)

FROM Developers AS d JOIN Games AS g

ON d.DeveloperID = g.DeveloperID

GROUP BY d.Developer;

GO

/\* DATEDIFF Function SUBQUERY \*/

/\* The purpose of this view is to show how old each game is where there is at least 3 games in platforms \*/

CREATE VIEW VW\_GamesAgeWith AS

SELECT Title, DATEDIFF(year, ReleaseDate, GETDATE()) AS Age

FROM Games AS G JOIN Platforms AS P

ON G.PlatformID = P.PlatformID

WHERE P.PlatformID IN

(SELECT PlatformID

FROM Games

GROUP BY PlatformID

HAVING COUNT(PlatformID) >= 3 );

GO

SELECT CAST(AmericanReleaseDate AS DATE) AS ReleaseDate

FROM VW\_GamesReleaseDate

ORDER BY ReleaseDate DESC;

|  |
| --- |
| ReleaseDate |
| 2020-03-20 |
| 2018-10-26 |
| 2017-12-20 |
| 2016-07-22 |
| 2016-05-24 |
| 2015-05-19 |
| 2014-05-29 |
| 2013-09-17 |
| 2011-11-18 |
| 2011-11-11 |
| 2011-05-16 |
| 2007-12-01 |
| 2006-11-19 |
| 1989-06-14 |
| 1985-09-13 |

SELECT Title

FROM VW\_GamesYearDay

WHERE Year = '2011';

|  |
| --- |
| Title |
| Minecraft |
| Terraria |
| The Elder Scrolls V: Skyrim |

SELECT Platform

FROM VW\_PlatformsWithoutGames

WHERE Platform > ‘W’;

|  |
| --- |
| Platform |
| Wii U |
| Xbox 360 |
| Windows |

SELECT Developer, AveragePrice

FROM VW\_GamesPrice

WHERE AveragePrice > 30;

|  |  |
| --- | --- |
| Developer | AveragePrice |
| Bethesda Game Studios | 60.000000 |
| Blizzard Entertainment | 40.000000 |
| CD Projekt | 40.000000 |
| Nintendo | 51.000000 |
| Rockstar Games | 60.000000 |

SELECT Title, Age

FROM VW\_GamesAgeWith

ORDER BY Age;

|  |  |
| --- | --- |
| Title | Age |
| Red Dead Redemption 2 | 6 |
| PUBG: Battlegrounds | 7 |
| Overwatch | 8 |
| Human: Fall Flat | 8 |
| The Witcher 3: Wild Hunt | 9 |
| Grand Theft Auto V | 11 |
| Minecraft | 13 |
| Terraria | 13 |
| The Elder Scrolls V: Skyrim | 13 |
| Tetris | 35 |
| Super Mario Bros. | 39 |

## Stored Procedures

/\* Created by Omar Sarazua on 12/9/2024 using Microsoft SQL Server \*/

USE GameOS;

DROP PROC IF EXISTS spGamesAffordability;

DROP PROC IF EXISTS spGamesTotalByDate;

GO

/\* Stored Procedure 1 - CASE function \*/

/\* The purpose of this sproc is to show what games can and can't be bought with the money supplied \*/

CREATE PROC spGamesAffordability

@Budget DECIMAL(4,2)

AS

SELECT Title, ReleaseDate, Price,

CASE

WHEN Price - @Budget <= 0 THEN 'Can Afford'

ELSE 'Cant Afford'

END AS Affordable

FROM Games;

GO

/\* Stored Procedure 2 – Input and Output Parameters \*/

/\* The purpose of this sproc is to display the total price of games released after the date supplied \*/

CREATE PROC spGamesTotalByDate

@InputDate DATE,

@PriceTotal MONEY OUTPUT

AS

SELECT @PriceTotal = SUM(Price)

FROM Games

WHERE ReleaseDate > @InputDate;

GO

EXEC spGamesAffordability 20;

|  |  |  |  |
| --- | --- | --- | --- |
| Title | ReleaseDate | Price | Affordable |
| Minecraft | 2011-11-18 | 26.95 | Cant Afford |
| Grand Theft Auto V | 2013-09-17 | 60.00 | Cant Afford |
| Wii Sports | 2006-11-19 | 20.00 | Can Afford |
| PUBG: Battlegrounds | 2017-12-20 | 0.00 | Can Afford |
| Mario Kart 8 | 2014-05-29 | 59.99 | Cant Afford |
| Red Dead Redemption 2 | 2018-10-26 | 59.99 | Cant Afford |
| Terraria | 2011-05-16 | 10.00 | Can Afford |
| Super Mario Bros. | 1985-09-13 | 25.00 | Cant Afford |
| Overwatch | 2016-05-24 | 40.00 | Cant Afford |
| Human: Fall Flat | 2016-07-22 | 20.00 | Can Afford |
| The Witcher 3: Wild Hunt | 2015-05-19 | 40.00 | Cant Afford |
| Animal Crossing: New Horizons | 2020-03-20 | 59.99 | Cant Afford |
| Wii Fit | 2007-12-01 | 89.99 | Cant Afford |
| Tetris | 1989-06-14 | 50.00 | Cant Afford |
| The Elder Scrolls V: Skyrim | 2011-11-11 | 59.99 | Cant Afford |

DECLARE @MyTotal MONEY;

EXEC spGamesTotalByDate @InputDate = '2000-12-03', @PriceTotal = @MyTotal OUTPUT;

PRINT 'The total price of games released after that date is ' + CONVERT(VARCHAR, @MyTotal);

The total price of games released after that date is 546.90