PROJECT DESCRIPTION

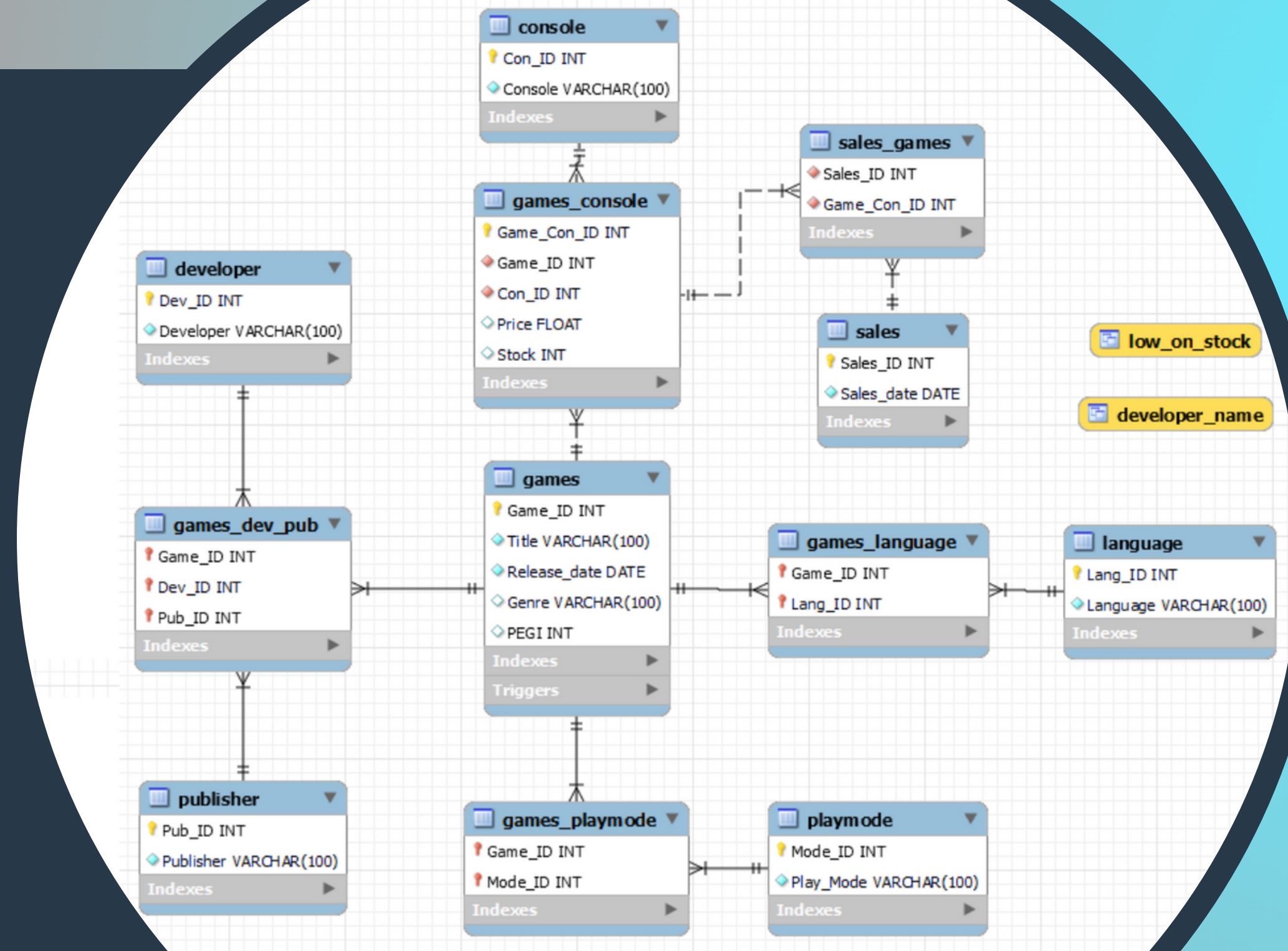## THE DB CONTAINS BASIC INFORMATION ABOUT GAMES FOR PLAYSTATION, XBOX AND NINTENDO SWITCH

Title, Release date, Genre, PEGI, Language, Playmode, Developer & Publisher

THE DATABASE ALSO INCLUDES INFORMATION ON PRICES, SALES AND STOCK.

EER DIAGRAM

The main database, games, has several relationships with other child databases through the foreign key – Game_ID.

There is also table with basic informations about sales – related with game_console table.
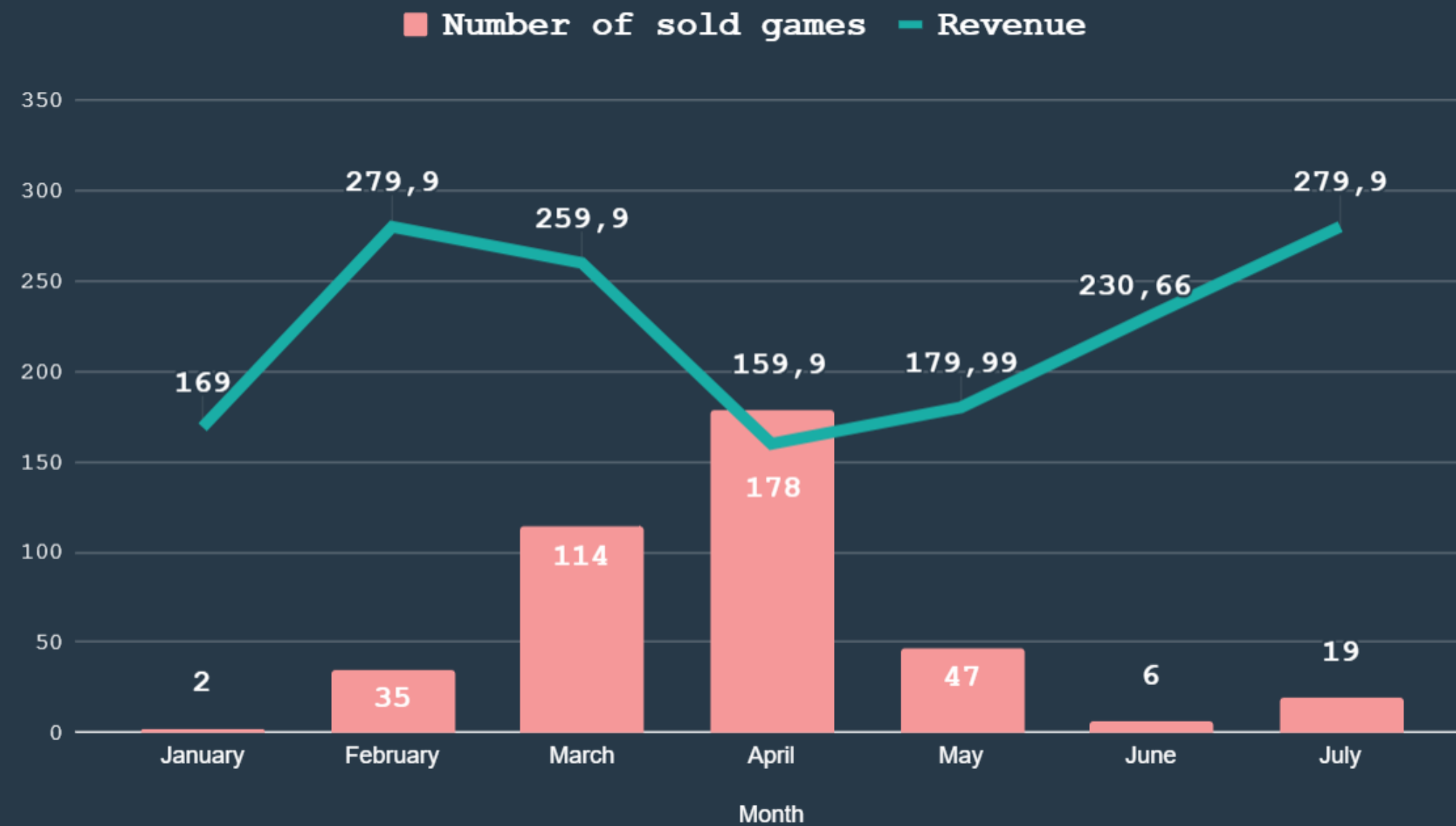
Game_ID	Title	Release_date	Genre	PEGI
1	A Plague Tale: Innocence	2021-07-06	Adventure	18
2	Apex Legends	2022-03-29	Action	16
3	Arcadegeddon	2022-07-05	Action	12
4	Assassin's Creed: Valhalla	2020-11-12	Action	18
5	Battlefield 2042	2021-11-19	Action	16
6	Car Mechanic Simulator 2021	2022-05-02	Simulation	3
7	Control	2021-02-02	Action	16
8	Crash Bandicoot 4: It's About Time	2021-03-12	Arcade	12
9	Crusader Kings III	2022-03-29	Strategy	12
10	Cyberpunk 2077	2022-02-15	RPG	18
...	...	...	...	...
88	Tropico 6: Next Gen Edition	2022-03-31	Strategy	16
89	Uncharted: Legacy of Thieves Collection	2022-01-28	Action	16
90	Worms: Armageddon	2022-06-23	Strategy	12

SAMPLE OF 'GAMES' TABLE

HOW MANY GAMES WERE SOLD EACH MONTH AND WHAT WAS THE REVENUE?

- LEFT JOIN
  - sales_games & sales
  - sales_games & games_console
- GROUP BY month
- ORDER BY month

```sql
SELECT
    DATE_FORMAT(sales.Sales_date, '%M') AS 'Month'
    , sales_games.Game_Con_ID AS 'Number of sold games'
    , games_console.Price AS 'Revenue'
FROM
    sales_games
        LEFT JOIN sales
            ON sales.Sales_ID = sales_games.Sales_ID
        LEFT JOIN games_console
            ON games_console.Game_Con_ID = sales_games.Game_Con_ID
GROUP BY DATE_FORMAT(sales.Sales_date, '%M')
ORDER BY MONTH(sales.Sales_date);
```

Month	Number of sold games	Revenue
January	2	169
February	35	279.9
March	114	259.9
April	178	159.9
May	47	179.99
June	6	230.66
July	19	279.9

SHOW GAMES THAT ARE LOW ON STOCK
VIEW

- CREATE VIEW Low_on_Stock
- LEFT JOIN
  - games & games_console
  - console & games_console
- GROUP BY Stock
- HAVING Stock less than 10

```sql
CREATE VIEW Low_on_Stock AS
    SELECT
        games.Title
        , console.Console
        , games_console.Stock
    FROM games_console
        LEFT JOIN games
            ON games_console.Game_ID = games.Game_ID
        LEFT JOIN console
            ON games_console.Con_ID = console.Con_ID
    GROUP BY games_console.Stock
    HAVING games_console.Stock < 10;


SELECT * FROM Low_on_Stock;
```

Title	Console	Stock
A Plague Tale: Innocence	XSX	3
F1 22	PS4	6
FIFA 22	PS4	2
FIFA 22	Nintendo Switch	1
Kona	Xbox One	4
LEGO Star Wars: The Skywalker Saga	Xbox One	7
Uncharted: Legacy of Thieves Collection	PS5	5

SHOW ALL GAMES DEVELOPED BY UBISOFT
VIEW

- CREATE VIEW Developer_name

- SUBQUERY WITH VIEW USAGE

```sql
CREATE VIEW Developer_name AS
    SELECT
        games_dev_pub.Dev_ID
        , developer.Developer
        , games_dev_pub.Game_ID
    FROM games_dev_pub
        LEFT JOIN developer
            ON  games_dev_pub.Dev_ID = developer. Dev_ID;
```

```sql
SELECT
    games.Title AS 'Game'
FROM games
WHERE Game_ID IN (
    SELECT
        Game_ID
    FROM Developer_name
    WHERE Developer_name.Developer = 'Ubisoft'
);
```

Game
Assassin's Creed: Valhalla
Far Cry 6
Just Dance 2022
Let's Sing 2022
Riders Republic
LEGO Star Wars: The Skywalker Saga
Uncharted: Legacy of Thieves Collection

CHECK IF THE GAME IS APPROPRIATE FOR CHILDREN, TEENAGERS OR FOR ADULTS ONLY

FUNCTION

- CREATE FUNCTION gamer_age
- IFS
  - IF PEGI is less or equal 7 SET 'Kids'
  - IF PEGI greater or equal 12 AND less or equal 16 SET 'Teenagers'
  - IF PEGI equals SET 'Adults'

```sql
DELIMITER //
CREATE FUNCTION gamer_age(PEGI INT)
    RETURNS VARCHAR(20)
    DETERMINISTIC
    BEGIN
        DECLARE gamer_age VARCHAR(20);
        IF PEGI <= 7 THEN
            SET gamer_age = 'Kids';
        ELSEIF (PEGI >= 12 AND PEGI <= 16) THEN
            SET gamer_age = 'Teenagers';
        ELSEIF PEGI = 18 THEN
            SET gamer_age = 'Adults';
        END IF;
        RETURN (gamer_age);
    END//
DELIMITER ;

SELECT
    Title
    , gamer_age(PEGI) AS 'Appropriate for'
FROM games
ORDER BY PEGI;
```

Title	Appropriate for
FIFA 21	Kids
SnowRunner	Kids
Fishing: North Atlantic	Kids
test title2	Kids
Car Mechanic Simulator 2021	Kids
Forza Horizon 5	Kids
Just Dance 2022	Kids
Pokemon Legends: Arceus	Kids
NBA 2K22	Kids
...	...
Outward: Definitive Edition	Teenagers
Riders Republic	Teenagers
Arcadegeddon	Teenagers
Crash Bandicoot 4: It's About Time	Teenagers
It Takes Two	Teenagers
Sea of Thieves	Teenagers
Crusader Kings III	Teenagers
...	...
Grand Theft Auto V	Adults
GreedFall	Adults
Hitman 3	Adults
Martha Is Dead	Adults
Shadow of the Tomb Raider	Adults
Assassin's Creed: Valhalla	Adults
Cyberpunk 2077	Adults
...	...

ENSURING THE CONSISTENCY OF INSERT PEGI DATA
TRIGGER

- CREATE TRIGGER PEGIcheck
- IFS if new instert is:
  - less or equal 3 SET PEGI 3
  - greater than 3 AND less or equal 7 SET PEGI 7
  - greater than 12 AND less than 18 SET PEGI 16
  - greater or quals 18 SET PEGI 18

```
DELIMITER //
CREATE TRIGGER PEGIcheck
    BEFORE INSERT ON games
    FOR EACH ROW
    IF NEW.PEGI <= 3 THEN SET NEW.PEGI = 3;
        ELSEIF (NEW.PEGI > 3 AND NEW.PEGI <= 7) THEN SET NEW.PEGI = 7;
        ELSEIF (NEW.PEGI > 7 AND NEW.PEGI <= 12) THEN SET NEW.PEGI = 12;
        ELSEIF (NEW.PEGI > 12 AND NEW. PEGI < 18) THEN SET NEW.PEGI = 16;
        ELSEIF NEW.PEGI >= 18 THEN SET NEW.PEGI = 18;
    END IF;
    //
DELIMITER ;
```

ADDING NEW GAME TO DB
PROCEDURE

- CREATE PROCEDURE NewGame
  - INSERT INTO 'games' table new values

TRIGGER PEGIcheck ensures consistency of data

```sql
DELIMITER //
CREATE PROCEDURE NewGame(
    IN Title VARCHAR(100)
    , IN Release_date DATE
    , IN Genre VARCHAR(100)
    , IN PEGI INT
    )
    BEGIN
        INSERT INTO games(Title, Release_date, Genre, PEGI)
            VALUES (Title, Release_date, Genre, PEGI);

    END //
DELIMITER ;


CALL NewGame('test title', '2022-07-21', 'Action', 5);
```
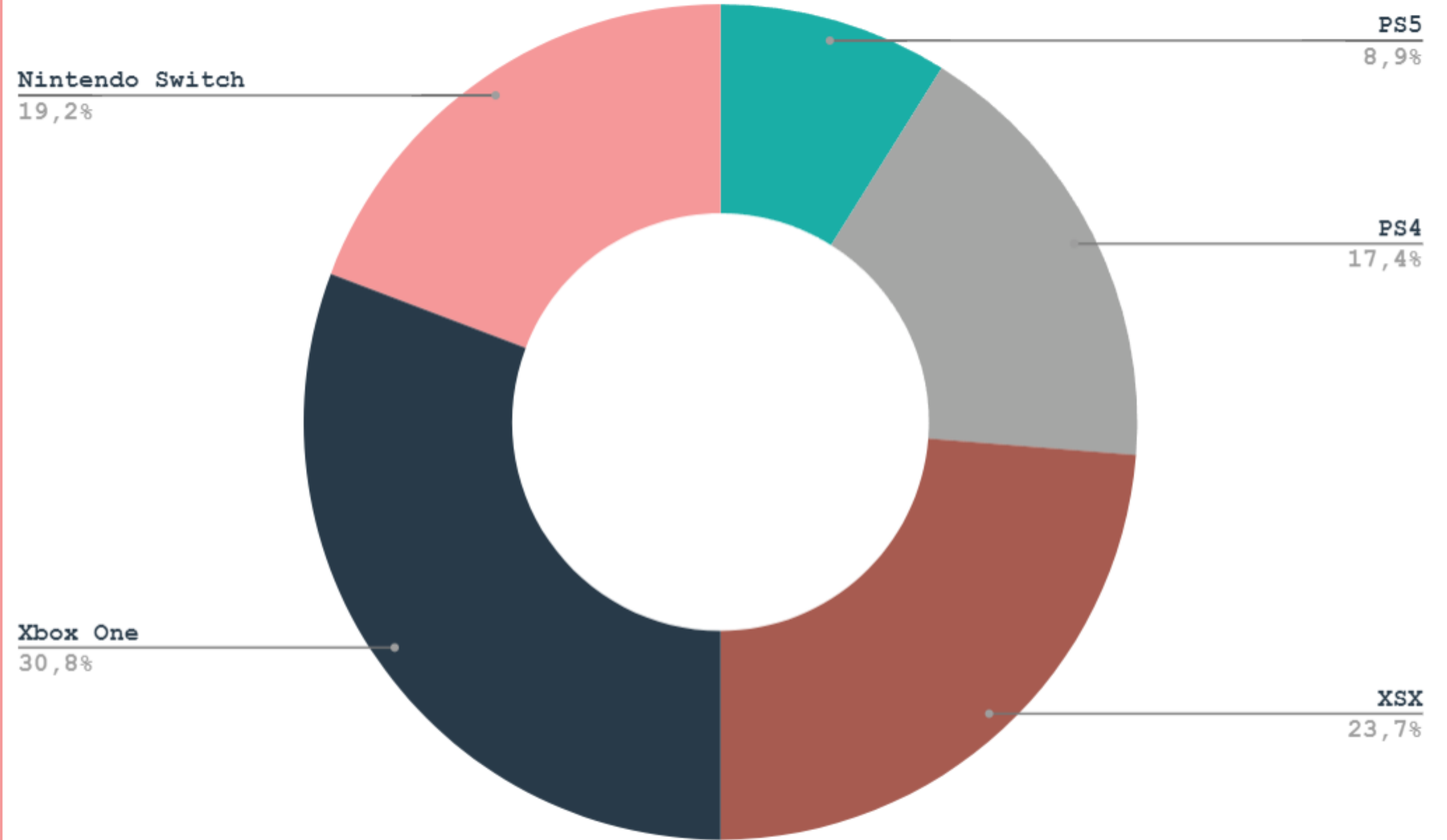
Game_ID	Title	PEGI
1	A Plague Tale: Innocence	18
2	Apex Legends	16
3	Arcadegeddon	12
...	...	...
90	Worms: Armageddon	12
91	WRC 10	3
92	**test title**	**7**

THE NUMBER OF GAMES FOR EACH CONSOLE

PS5
8,9%

PS4
17,4%

XSX
23,7%

Nintendo Switch
19,2%

Xbox One
30,8%

```sql
SELECT
    console.Console
    , SUM(games_console.Con_ID) AS 'Number of games'
FROM
    games_console
        LEFT JOIN console
            ON games_console.Con_ID = console.Con_ID
GROUP BY games_console.Con_ID;
```

THANK YOU

AGNIESZKA KOZŁOWSKA

LINKED IN

www.linkedin.com/in/agnieszka-paulina-kozłowska