

Video games are big business: the global gaming market is projected to be worth more than \$300 billion by 2027 according to Mordor Intelligence. With so much money at stake, the major game publishers are hugely incentivized to create the next big hit. But are games getting better, or has the golden age of video games already passed?

In this project, you'll analyze video game critic and user scores as well as sales data for the top 400 video games released since 1977. You'll search for a golden age of video games by identifying release years that users and critics liked best, and you'll explore the business side of gaming by looking at game sales data.

Your search will involve joining datasets and comparing results with set theory. You'll also filter, group, and order data. Make sure you brush up on these skills before trying this project! The database contains two tables. Each table has been limited to 400 rows for this project, but you can find the complete dataset with over 13,000 games on Kaggle.

game\_sales table

Column	Definition	Data Type
name	Name of the video game	varchar
platform	Gaming platform	varchar
publisher	Game publisher	varchar
developer	Game developer	varchar
games_sold	Number of copies sold (millions)	float
year	Release year	int

reviews table

Column	Definition	Data Type
name	Name of the video game	varchar
critic_score	Critic score according to Metacritic	float
user_score	User score according to Metacritic	float

users\_avg\_year\_rating table

Column	Definition	Data Type
year	Release year of the games reviewed	int
num_games	Number of games released that year	int
avg_user_score	Average score of all the games ratings for the year	float

critics\_avg\_year\_rating table

Column	Definition	Data Type
year	Release year of the games reviewed	int
num_games	Number of games released that year	int
avg_critic_score	Average score of all the games ratings for the year	float

Projects Data DataFrame as best\_selling\_games

-- Find the ten best-selling games. The output should contain all the columns in the game\_sales table and be sorted by the games\_sold column in descending order. Save the output as best\_selling\_games.

```
SELECT *
FROM game_sales
ORDER BY games_sold DESC
LIMIT 10;
```

i...	...	↑↓	name	...	↑↓	plat...	...	↑↓	publisher	...	↑↓	developer
	0		Wii Sports for Wii			Wii			Nintendo			Nintendo EAD
	1		Super Mario Bros. for NES			NES			Nintendo			Nintendo EAD
	2		Counter-Strike: Global Offensive for PC			PC			Valve			Valve Corporation
	3		Mario Kart Wii for Wii			Wii			Nintendo			Nintendo EAD
	4		PLAYERUNKNOWN'S BATTLEGROUNDS for PC			PC			PUBG Corporation			PUBG Corporation
	5		Minecraft for PC			PC			Mojang			Mojang AB
	6		Wii Sports Resort for Wii			Wii			Nintendo			Nintendo EAD
	7		Pokemon Red / Green / Blue Version for GB			GB			Nintendo			Game Freak
	8		New Super Mario Bros. for DS			DS			Nintendo			Nintendo EAD
	9		New Super Mario Bros. Wii for Wii			Wii			Nintendo			Nintendo EAD

Rows: 10

Expand Table

Projects Data DataFrame as critics\_

-- Find the ten years with the highest average critic score, where at least four games were released (to ensure a good sample size). Return an output with the columns year, num\_games released, and avg\_critic\_score. The avg\_critic\_score should be rounded to 2 decimal places. The table should be ordered by avg\_critic\_score in descending order. Save the output as critics\_top\_ten\_years. Do not use the critics\_avg\_year\_rating table provided; this has been provided for your third query.

```
SELECT g.year, COUNT(g.name) AS num_games, ROUND(AVG(r.critic_score),2) AS avg_critic_score
FROM game_sales g
INNER JOIN reviews r
ON g.name = r.name
GROUP BY g.year
HAVING COUNT(g.name) >= 4
ORDER BY avg_critic_score DESC
LIMIT 10;
```

index	...	↑↓	year	...	↑↓	num_games	...	↑↓	avg_critic_score
		0			1998			10	
		1			2004			11	
		2			2002			9	
		3			1999			11	
		4			2001			13	
		5			2011			26	
		6			2016			13	
		7			2013			18	
		8			2008			20	
		9			2017			13	

Rows: 10

Expand Table

Projects Data

DataFrame as g

-- Find the years where critics and users broadly agreed that the games released were highly rated. Specifically, return the years where the average critic score was over 9 OR the average user score was over 9. The pre-computed average critic and user scores per year are stored in users\_avg\_year\_rating and critics\_avg\_year\_rating tables respectively. The query should return the following columns: year, num\_games, avg\_critic\_score, avg\_user\_score, and diff. The diff column should be the difference between the avg\_critic\_score and avg\_user\_score. The table should be ordered by the year in ascending order, save this as a DataFrame named golden\_years.  
SELECT u.year, u.num\_games, c.avg\_critic\_score, u.avg\_user\_score, c.avg\_critic\_score - u.avg\_user\_score AS diff  
FROM critics\_avg\_year\_rating c  
INNER JOIN users\_avg\_year\_rating u  
ON c.year = u.year  
WHERE c.avg\_critic\_score > 9 OR u.avg\_user\_score > 9  
ORDER BY year ASC

...	↑↓	...	↑↓	n.	...	↑↓	avg_critic_s...	...	↑↓	avg_use...	...	↑↓	...	↑↓
0		1997				8			7.93			9.5		-1.57
1		1998				10			9.32			9.4		-0.08
2		2004				11			9.03			8.55		0.48
3		2008				20			8.63			9.03		-0.4
4		2009				20			8.55			9.18		-0.63
5		2010				23			8.41			9.24		-0.83

Rows: 6

Expand Table