



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

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

in...	...	↑↓	name	...	↑↓	plat...	...	↑↓	publisher	...	↑↓	developer	...	↑↓	games_s...	...	↑↓	ye...	...
0	Wii Sports for Wii					Wii			Nintendo			Nintendo EAD			82.9			2	
1	Super Mario Bros. for NES					NES			Nintendo			Nintendo EAD			40.24			1	
2	Counter-Strike: Global Offensive for PC					PC			Valve			Valve Corporation			40			2	
3	Mario Kart Wii for Wii					Wii			Nintendo			Nintendo EAD			37.32			2	
4	PLAYERUNKNOWN'S BATTLEGROUNDS for PC					PC			PUBG Corporation			PUBG Corporation			36.6			2	
5	Minecraft for PC					PC			Mojang			Mojang AB			33.15			2	
6	Wii Sports Resort for Wii					Wii			Nintendo			Nintendo EAD			33.13			2	
7	Pokemon Red / Green / Blue Version for GB					GB			Nintendo			Game Freak			31.38			1	
8	New Super Mario Bros. for DS					DS			Nintendo			Nintendo EAD			30.8			2	
9	New Super Mario Bros. Wii for Wii					Wii			Nintendo			Nintendo EAD			30.3			2	

Rows: 10 ↗ Expand

Projects Data DataFrame as critics.

```
-- critics_top_ten_years
SELECT
    year,
    num_games,
    ROUND(avg_critic_score,2) AS avg_critic_score
FROM critics_avg_year_rating
WHERE num_games >=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

Projects Data DataFrame as gold

```
-- golden_years
SELECT
    u.year,
    u.num_games,
    s.avg_critic_score AS avg_critic_score,
    u.avg_user_score AS avg_user_score,
    (u.avg_user_score - s.avg_critic_score) AS diff
FROM users_avg_year_rating u
JOIN critics_avg_year_rating s
ON u.year = s.year
WHERE s.avg_critic_score > 9 OR u.avg_user_score > 9
ORDER BY u.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