

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	<code>varchar</code>
platform	Gaming platform	<code>varchar</code>
publisher	Game publisher	<code>varchar</code>
developer	Game developer	<code>varchar</code>
games_sold	Number of copies sold (millions)	<code>float</code>
year	Release year	<code>int</code>

`reviews` table

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

`users_avg_year_rating` table

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

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 all information for the top ten best-selling games
-- Order the results from best-selling game down to tenth best-selling
SELECT *
FROM game_sales
ORDER BY games_sold DESC
LIMIT 10;
```

	▼ name	▼ platform	▼ publisher	▼ dev
0	Wii Sports for Wii	Wii	Nintendo	Nin
1	Super Mario Bros. for NES	NES	Nintendo	Nin
2	Counter-Strike: Global Offensive for PC	PC	Valve	Val
3	Mario Kart Wii for Wii	Wii	Nintendo	Nin
4	PLAYERUNKNOWN'S BATTLEGROUNDS for PC	PC	PUBG Corporation	PU
5	Minecraft for PC	PC	Mojang	Mo
6	Wii Sports Resort for Wii	Wii	Nintendo	Nin
7	Pokemon Red / Green / Blue Version for GB	GB	Nintendo	Ga
8	New Super Mario Bros. for DS	DS	Nintendo	Nin
9	New Super Mario Bros. Wii for Wii	Wii	Nintendo	Nin

10 rows 

 Projects Data DataFrame as critics_top_ten_years

```
-- critics_top_ten_years
-- Select release year and average critic score for each year, rounded and aliased
-- Join the game_sales and reviews tables
-- Group by release year
-- Order the data from highest to lowest avg_critic_score and limit to 10 results

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;
```

	▼	year	▼	num_games	▼	avg_c
	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	

10 rows [↓](#)

 Projects Data DataFrame as gold

```
-- golden_years
-- Select year, num_games, avg_critic_score, avg_user_score and the difference between the
user and critic scores as avg_score.
-- Join critics_top_ten_years and users_top_years on the year column
-- Where the average user or critic rating for the year was over 9.
-- Order data by diff in ascending order
```

```
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
```

	▼	year	▼	num_games	▼	avg_critic_score	▼	a
	0		1997		8		7.93	
	1		1998		10		9.32	
	2		2004		11		9.03	
	3		2008		20		8.63	
	4		2009		20		8.55	
	5		2010		23		8.41	

6 rows [↓](#)