

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

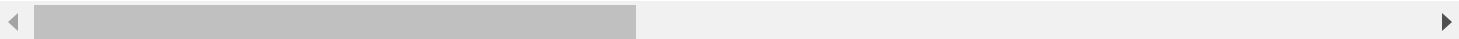
Question 1

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`.

 Projects Data DataFrame as `best_selling_games`

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

	▼ name	▼ platform
0	Wii Sports for Wii	Wii
1	Super Mario Bros. for NES	NES
2	Counter-Strike: Global Offensive for PC	PC
3	Mario Kart Wii for Wii	Wii
4	PLAYERUNKNOWN'S BATTLEGROUNDS for PC	PC
5	Minecraft for PC	PC
6	Wii Sports Resort for Wii	Wii
7	Pokemon Red / Green / Blue Version for GB	GB
8	New Super Mario Bros. for DS	DS
9	New Super Mario Bros. Wii for Wii	Wii



10 rows 

Question 2

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.

 Projects Data DataFrame as `critics_top_ten_years`

```
-- critics_top_ten_years
SELECT
    year,
    COUNT(1) AS num_games,
    ROUND(AVG(critic_score), 2) AS avg_critic_score
FROM game_sales gs
INNER JOIN reviews r ON gs.name = r.name
GROUP BY year
HAVING COUNT(1) >= 4
ORDER BY avg_critic_score DESC
LIMIT 10;
```

▼	year	▼	num_games
0	1998		
1	2004		
2	2002		
3	1999		
4	2001		
5	2011		
6	2016		
7	2013		
8	2008		
9	2017		

10 rows [↓](#)

Question 3

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`.

 Projects Data DataFrame as gold

```
-- golden_years
SELECT
    ur.year,
    ur.num_games,
    ur.avg_user_score,
    cr.avg_critic_score,
    (cr.avg_critic_score - ur.avg_user_score) AS diff
FROM users_avg_year_rating AS ur
```

```
JOIN critics_avg_year_rating AS cr ON ur.year = cr.year
WHERE ur.avg_user_score > 9 OR cr.avg_critic_score > 9
ORDER BY year;
```

▼	year	▼	num_games	▼	avg_u
0	1997		8		
1	1998		10		
2	2004		11		
3	2008		20		
4	2009		20		
5	2010		23		

6 rows [↓](#)

Note: Please also ensure that you do not change the names of the DataFrames that the three query results will be saved as - creating new cells in the workbook will rename the DataFrame. Make sure that your final solutions use the names provided: `best_selling_games` , `critics_top_ten_years` , and `golden_years` .