

▼ 1. The ten best-selling video games



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, we'll explore the top 400 best-selling video games created between 1977 and 2020. We'll compare a dataset on game sales with critic and user reviews to determine whether or not video games have improved as the gaming market has grown.

Our database contains two tables.

game_sales

column	type	meaning
game	varchar	Name of the video game
platform	varchar	Gaming platform
publisher	varchar	Game publisher
developer	varchar	Game developer
games sold	float	Number of copies sold (millions)

column	type	meaning
year	int	Release year

reviews

column	type	meaning
game	varchar	Name of the video game
critic_score	float	Critic score according to Metacritic
user_score	float	User score according to Metacritic

Let's begin by looking at some of the top selling video games of all time!

```
%%sql
```

```
postgresql:///games
```

```
--Top selling Video Games of all Time
```

```
SELECT * FROM game_sales
```

```
ORDER BY games_sold DESC
```

```
LIMIT 10;
```

10 rows affected.

game	platform	publisher	developer	games_sold	year
Wii Sports for Wii	Wii	Nintendo	Nintendo EAD	82.90	2006
Super Mario Bros. for NES	NES	Nintendo	Nintendo EAD	40.24	1985
Counter-Strike: Global Offensive for PC	PC	Valve	Valve Corporation	40.00	2012
Mario Kart Wii for Wii	Wii	Nintendo	Nintendo EAD	37.32	2008
PLAYERUNKNOWN'S BATTLEGROUNDS for PC	PC	PUBG Corporation	PUBG Corporation	36.60	2017
Minecraft for PC	PC	Mojang	Mojang AB	33.15	2010
Wii Sports Resort for Wii	Wii	Nintendo	Nintendo EAD	33.13	2009
Pokemon Red / Green / Blue Version for GB	GB	Nintendo	Game Freak	31.38	1998

```
%%sql
```

```
--Total number of video games
```

```
SELECT COUNT(*) FROM game_sales;
```

```
* postgresql:///games
```

1 rows affected.

count

400

▼ 2. Missing review scores

Wow, the best-selling video games were released between 1985 to 2017! That's quite a range; we'll have to use data from the `reviews` table to gain more insight on the best years for video games.

First, it's important to explore the limitations of our database. One big shortcoming is that there is not any reviews data for some of the games on the `game_sales` table.

```
%%sql
```

```
--Video games with no reviews
```

```
SELECT * FROM game_sales g
```

```
LEFT JOIN reviews r
```

```
ON g.game = r.game
```

```
WHERE critic_score IS NULL AND user_score IS NULL;
```

```
* postgresql:///games
```

```
31 rows affected.
```

game	platform	publisher	developer	games_sold	year	game_1	critic_score	user_score
7 Days to Die for PC	PC	The Fun Pimps	The Fun Pimps	4.18	2013	None	None	None
Arma 2: Operation Arrowhead for PC	PC	Meridian4	Bohemia Interactive	4.51	2010	None	None	None
Arma III for PC	PC	Bohemia Interactive	Bohemia Interactive	4.00	2013	None	None	None
Assassin's Creed: Unity for PS4	PS4	Ubisoft	Ubisoft Montreal	4.14	2014	None	None	None
Call of Duty: Black Ops 3 for XOne	XOne	Activision	Treyarch	7.37	2015	None	None	None
Call of Duty: Black Ops III for XOne	XOne	Activision	Treyarch	4.85	2018	None	None	None
Disney's Aladdin for GEN	GEN	Sega	Virgin Interactive	4.00	1993	None	None	None
Dr. Mario for GB	GB	Nintendo	Nintendo R&D1	5.34	1990	None	None	None
Dr. Mario for NES	NES	Nintendo	Nintendo R&D1	4.85	1990	None	None	None
FIFA 13 for PS3	PS3	EA Sports	EA Canada	8.01	2012	None	None	None
FIFA 13 for X360	X360	EA Sports	EA Canada	5.11	2012	None	None	None
Garry's Mod for PC	PC	Valve	Facepunch Studios	16.90	2006	None	None	None
Golf for NES	NES	Nintendo	Nintendo	4.01	1985	None	None	None
Gran Turismo Sport for PS4	PS4	Sony Computer Entertainment America	Polyphony Digital	5.22	2013	None	None	None

▼ 3. Years that video game critics loved

It looks like a little less than ten percent of the games on the `game_sales` table don't have any reviews data. That's a small enough percentage that we can continue our exploration, but the missing reviews data is a good thing to keep in mind as we move on to evaluating results from more sophisticated queries.

There are lots of ways to measure the best years for video games! Let's start with what the critics think.

```
%%sql
```

```
-- Year with highest Average Critics Rating
WITH CTE AS (SELECT * FROM game_sales g
INNER JOIN reviews r
ON g.game = r.game)
SELECT year , ROUND(AVG(critic_score),2) as avg_critic_score
FROM CTE
GROUP BY year
ORDER BY avg_critic_score DESC
LIMIT 10;
```

```
* postgresql:///games
10 rows affected.
```

```
year avg_critic_score
```

```
1990 9.80
```

```
1992 9.67
```

```
1998 9.32
```

```
2020 9.20
```

```
1993 9.10
```

```
1995 9.07
```

```
2004 9.03
```

```
1982 9.00
```

```
2002 8.99
```

```
1999 8.93
```

▼ 4. Was 1982 really that great?

The range of great years according to critic reviews goes from 1982 until 2020:

Maybe there weren't a lot of video games in our dataset that were released in certain years.

Let's update our query and find out whether 1982 really was such a great year for video games.

```
%%sql
```

```
-- Update the query to see the number of Video Games released in each Year
WITH CTE AS (SELECT g.game, g.games_sold, g.year, r.critic_score, r.user_score FROM game_sales
INNER JOIN reviews r
ON g.game = r.game)
SELECT year ,COUNT(game) as num_games, ROUND(AVG(critic_score),2) as avg_critic_score
FROM CTE
GROUP BY year
HAVING COUNT(game) > 4
ORDER BY avg_critic_score DESC
LIMIT 10;
```

```
* postgresql:///games
```

```
10 rows affected.
```

```
year num_games avg_critic_score
```

```
1998 10          9.32
```

```
2004 11          9.03
```

```
2002 9           8.99
```

```
1999 11          8.93
```

```
2001 13          8.82
```

```
2011 26          8.76
```

```
2016 13          8.67
```

```
2013 18          8.66
```

```
2008 20          8.63
```

```
2017 13          8.62
```

▼ 5. Years that dropped off the critics' favorites list

That looks better! The `num_games` column convinces us that our new list of the critics' top games reflects years that had quite a few well-reviewed games rather than just one or two hits. But which years dropped off the list due to having four or fewer reviewed games? Let's identify them so that someday we can track down more game reviews for those years and determine whether they might rightfully be considered as excellent years for video game releases!

We are creating two TEMPORARY TABLE TO STORE OUR RESULTS OF PREVIOUS QUERIES

`top_critic_years`

column	type	meaning
year	int	Year of video game release
avg_critic_score	float	Average of all critic scores for games released in that year

`top_critic_years_more_than_four_games`

column	type	meaning
year	int	Year of video game release

column	type	meaning
num_games	int	Count of the number of video games released in that year
avg_critic_score	float	Average of all critic scores for games released in that year

```
%%sql
```

```
postgresql:///games
```

```
CREATE TEMP TABLE top_critic_years AS
```

```
SELECT * FROM (WITH CTE AS (SELECT * FROM game_sales g
INNER JOIN reviews r
ON g.game = r.game)
SELECT year , ROUND(AVG(critic_score),2) as avg_critic_score
FROM CTE
GROUP BY year
ORDER BY avg_critic_score DESC
LIMIT 10) AS F
```

(psycopg2.errors.DuplicateTable) relation "top_critic_years" already exists

```
[SQL: CREATE TEMP TABLE top_critic_years AS
```

```
SELECT * FROM (WITH CTE AS (SELECT * FROM game_sales g
INNER JOIN reviews r
ON g.game = r.game)
SELECT year , ROUND(AVG(critic_score),2) as avg_critic_score
FROM CTE
GROUP BY year
ORDER BY avg_critic_score DESC
LIMIT 10) AS F]
```

(Background on this error at: <https://sqlalche.me/e/14/f405>)

```
%%sql
```

```
postgresql:///games
```

```
CREATE TEMP TABLE top_critic_years_more_than_FOUR_games AS
```

```
SELECT * FROM (
WITH CTE AS (SELECT g.game, g.games_sold, g.year,r.critic_score, r.user_score FROM game_sales
INNER JOIN reviews r
ON g.game = r.game)
SELECT year ,COUNT(game) as num_games, ROUND(AVG(critic_score),2) as avg_critic_score
FROM CTE
GROUP BY year
HAVING COUNT(game) > 4
ORDER BY avg_critic_score DESC
LIMIT 10) AS T
```

(psycopg2.errors.DuplicateTable) relation "top_critic_years_more_than_four_games" already exists

```
[SQL: CREATE TEMP TABLE top_critic_years_more_than_FOUR_games AS
```

```

SELECT * FROM (
  WITH CTE AS (SELECT g.game, g.games_sold, g.year,r.critic_score, r.user_score FROM game_
  INNER JOIN reviews r
  ON g.game = r.game)
  SELECT year ,COUNT(game) as num_games, ROUND(AVG(critic_score),2) as avg_critic_score
  FROM CTE
  GROUP BY year
  HAVING COUNT(game) > 4
  ORDER BY avg_critic_score DESC
  LIMIT 10) AS T]
(Background on this error at: https://sqlalche.me/e/14/f405)

```

```
%%sql
```

```

-- We will select the year which has number of video games more than 4 and and years in Top 10
SELECT year, avg_critic_score
FROM top_critic_years
EXCEPT
SELECT year, avg_critic_score
FROM top_critic_years_more_than_FOUR_games
ORDER BY avg_critic_score DESC;

```

```

* postgresql:///games
6 rows affected.

```

```
year avg_critic_score
```

```

1990 9.80
1992 9.67
2020 9.20
1993 9.10
1995 9.07
1982 9.00

```

▼ 6. Years video game players loved

Based on our work in the task above, it looks like the early 1990s might merit consideration as the golden age of video games based on `critic_score` alone, but we'd need to gather more games and reviews data to do further analysis.

Let's move on to looking at the opinions of another important group of people: players! To begin, let's create a query very similar to the one we used in Task Four, except this one will look at `user_score` averages by year rather than `critic_score` averages.

```
%%sql
```

```

-- SELECTING THE YEARS WITH HIGH USER SCORE
WITH CTE AS (SELECT g.game, g.games_sold, g.year,r.critic_score, r.user_score FROM game_sales

```

```

INNER JOIN reviews r
ON g.game = r.game)
SELECT year ,COUNT(game) as num_games, ROUND(AVG(user_score),2) as avg_user_score
FROM CTE
GROUP BY year
HAVING COUNT(game) > 4
ORDER BY AVG_user_score DESC
LIMIT 10;

```

```

* postgresql:///games
10 rows affected.

```

```

year num_games avg_user_score

```

```

1997 8          9.50
1998 10         9.40
2010 23         9.24
2009 20         9.18
2008 20         9.03
1996 5          9.00
2005 13         8.95
2006 16         8.95
2000 8          8.80
2002 9          8.80

```

```

%%sql

```

```

postgresql:///games

```

```

CREATE TEMP TABLE top_user_years_more_than_FOUR_games AS
SELECT * FROM (
WITH CTE AS (SELECT g.game, g.games_sold, g.year,r.critic_score, r.user_score FROM game_sales
INNER JOIN reviews r
ON g.game = r.game)
SELECT year ,COUNT(game) as num_games, ROUND(AVG(user_score),2) as avg_user_score
FROM CTE
GROUP BY year
HAVING COUNT(game) > 4
ORDER BY AVG_user_score DESC
LIMIT 10) AS U

```

(psycopg2.errors.DuplicateTable) relation "top_user_years_more_than_four_games" already

```

[SQL: CREATE TEMP TABLE top_user_years_more_than_FOUR_games AS
SELECT * FROM (
WITH CTE AS (SELECT g.game, g.games_sold, g.year,r.critic_score, r.user_score FROM game_
INNER JOIN reviews r
ON g.game = r.game)
SELECT year ,COUNT(game) as num_games, ROUND(AVG(user_score),2) as avg_user_score
FROM CTE
GROUP BY year
HAVING COUNT(game) > 4
ORDER BY AVG_user_score DESC

```



```
LIMIT 10) AS U]
```

```
(Background on this error at: https://sqlalche.me/e/14/f405)
```

▼ 7. Years that both players and critics loved

Alright, we've got a list of the top ten years according to both critic reviews and user reviews. Are there any years that showed up on both tables? If so, those years would certainly be excellent ones!

:

`top_critic_years_more_than_four_games`

column	type	meaning
year	int	Year of video game release
num_games	int	Count of the number of video games released in that year
avg_critic_score	float	Average of all critic scores for games released in that year

We've also saved the results of our top user years query from the previous task into a table:

`top_user_years_more_than_four_games`

column	type	meaning
year	int	Year of video game release
num_games	int	Count of the number of video games released in that year
avg_user_score	float	Average of all user scores for games released in that year

```
%%sql
```

```
-- Select the year results that appear on both tables
SELECT year FROM top_critic_years_more_than_FOUR_games
INTERSECT
SELECT year FROM top_user_years_more_than_FOUR_games
```

```
* postgresql:///games
3 rows affected.
```

```
year
2008
2002
1998
```

▼ 8. Sales in the best video game years

Looks like we've got three years that both users and critics agreed were in the top ten!

We will see the number of games_sold in those 3 years

```
%%sql
```

```
postgresql:///games
```

```
SELECT year, sum(games_sold) as total_games_sold FROM game_sales  
WHERE year IN (SELECT year FROM top_critic_years_more_than_FOUR_games  
INTERSECT  
SELECT year FROM top_user_years_more_than_FOUR_games)  
GROUP BY year  
ORDER BY total_games_sold DESC
```

3 rows affected.

year	total_games_sold
------	------------------

2008	175.07
------	--------

1998	101.52
------	--------

2002	58.67
------	-------