Analysis

1) Looking at the dataset first

```
%%sql
postgresql:///games
select * from game_sales order by games_sold desc limit 10
-- Order the results from best-selling game down to tenth best-selling
```

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
New Super Mario Bros. for DS	DS	Nintendo	Nintendo EAD	30.80	2006

2) We'll have to use data from the reviews table to gain more insight into the best years for video games. Let's check how many games have reviews.

```
%%sql

select
    count(gs.game)
from
    game_sales gs
left join
    reviews r
on
    gs.game = r.game
where
    r.critic_score is Null and r.user_score is Null
```

count
31

It looks like a little less than ten percent of the games on the game_sales table don't have any review data. That's a small enough percentage that we can continue our exploration

3) We are going to find out which is the best-reviewed game by critics.

```
Select
    gs.year,
    round(avg(critic_score),2) as avg_critic_score
from
    game_sales gs
join
    reviews r
on
    gs.game = r.game
group by
    gs.year
order by
    avg_critic_score desc
limit 10
```

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

Some of those avg_critic_score values look like suspiciously round numbers for averages.

```
%%sql
Select
    gs.year,
    round(avg(critic_score),2) as avg_critic_score,
    count(gs.game) as num_games
from
    game_sales gs
join
    reviews r
on
    gs.game = r.game
group by
    gs.year
having
    count(gs.game) > 4
order by
    avg_critic_score desc
limit 10
```

year	avg_critic_score	num_games
1998	9.32	10
2004	9.03	11
2002	8.99	9
1999	8.93	11
2001	8.82	13
2011	8.76	26
2016	8.67	13
2013	8.66	18
2008	8.63	20
2017	8.62	13

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.

4) We are going to find out which is the best-reviewed game by users.

```
Select
    gs.year,
    round(avg(user_score),2) as avg_user_score,
    count(gs.game) as num_games
from
    game_sales gs
join
    reviews r
on
    gs.game = r.game
group by
    gs.year
having
    count(gs.game) > 4
order by
    avg_user_score desc
limit 10
```

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

5) To find which years are presented in both queries, we can use aliases for the previous two results. We have created two aliases top_critic_years_more_than_four_games and top_user_years_more_than_four_games.

```
%%sql

Select
    tcyu.year
from
    top_user_years_more_than_four_games tcyu
left join
    top_critic_years_more_than_four_games tcyc
on
    tcyu.year = tcyc.year
where
    tcyc.avg_critic_score is not NULL
```

year
1998
2008
2002

6) Looks like we've got three years that both users and critics agreed were in the top ten! Now we can look at the sales for each year.

```
%%sql
Select
    year,
    sum(games_sold) as total_games_sold
from
    game_sales
where
    year
in
    (Select
        tcyu.year
    from
        top_user_years_more_than_four_games tcyu
    left join
        top_critic_years_more_than_four_games tcyc
    on
        tcyu.year = tcyc.year
    where
        tcyc.avg_critic_score is not NULL)
group by
    year
order by
    total_games_sold desc
```

year	total_games_sold
2008	175.07
1998	101.52
2002	58.67