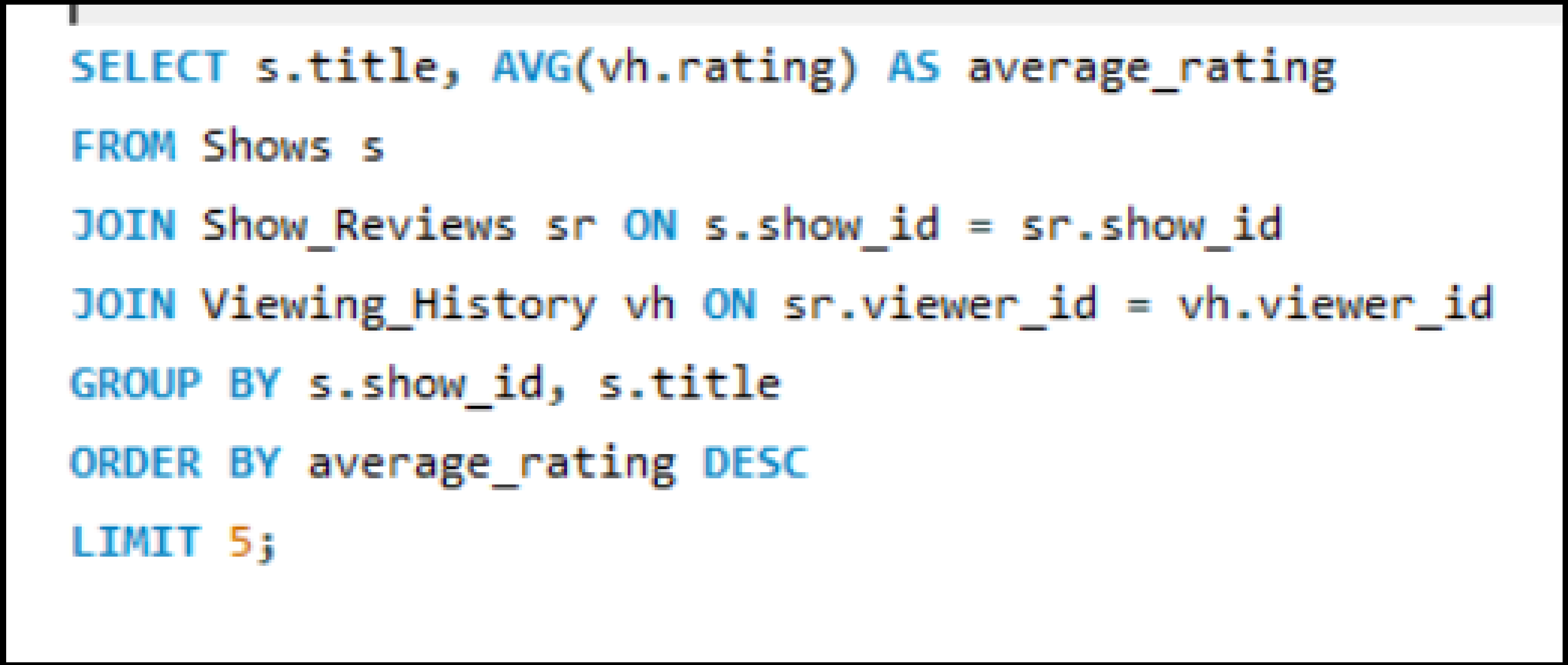
NETFLIX

Data Insights

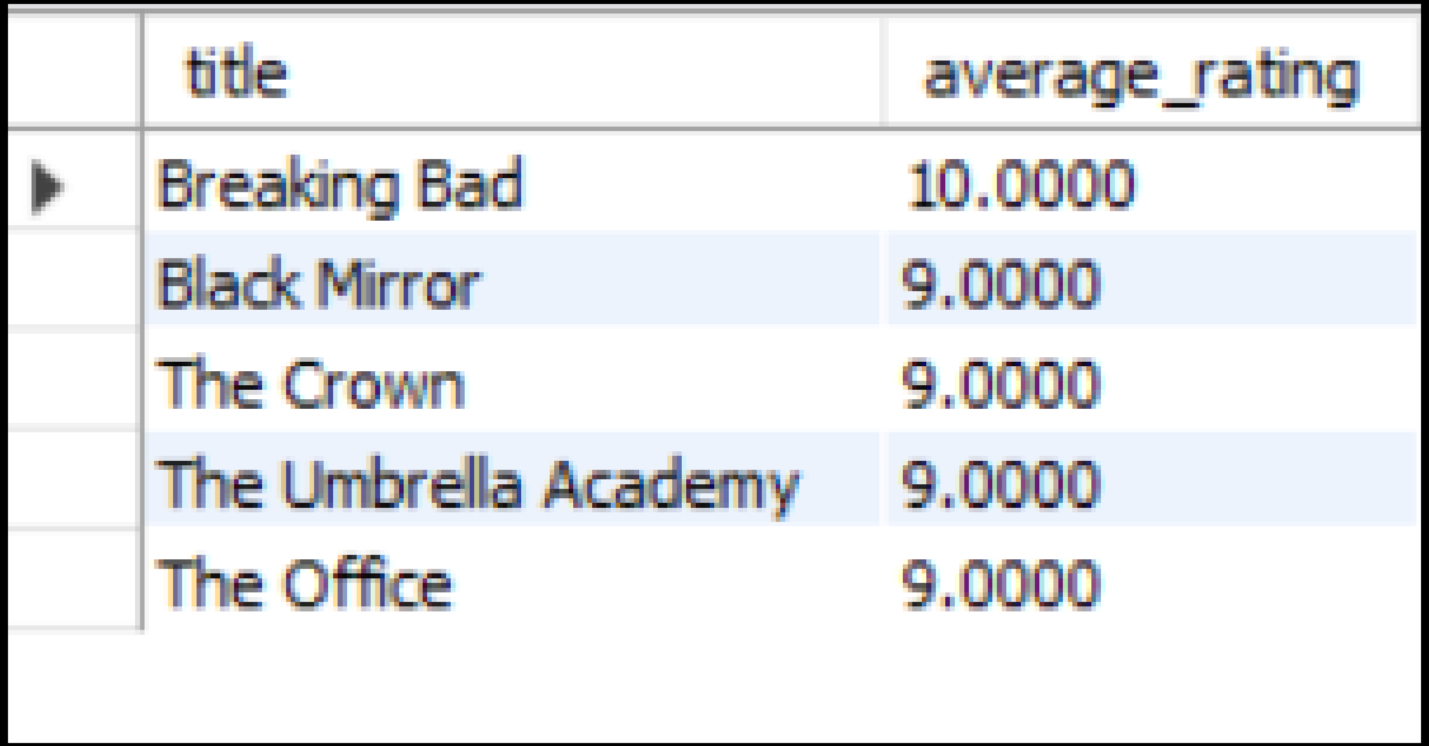
INTRODUCTON

Hello Everyone!
Explore the depths of streaming data with this advanced Netflix analysis project. Utilizing sophisticated SQL queries and advanced data manipulation techniques, this project provides a detailed examination of viewer behavior, content performance, and engagement metrics. Dive into insights from complex SELECT statements and advanced queries to uncover trends and patterns in Netflix's streaming landscape.

Identify the top 5 shows based on the highest average rating given by viewers

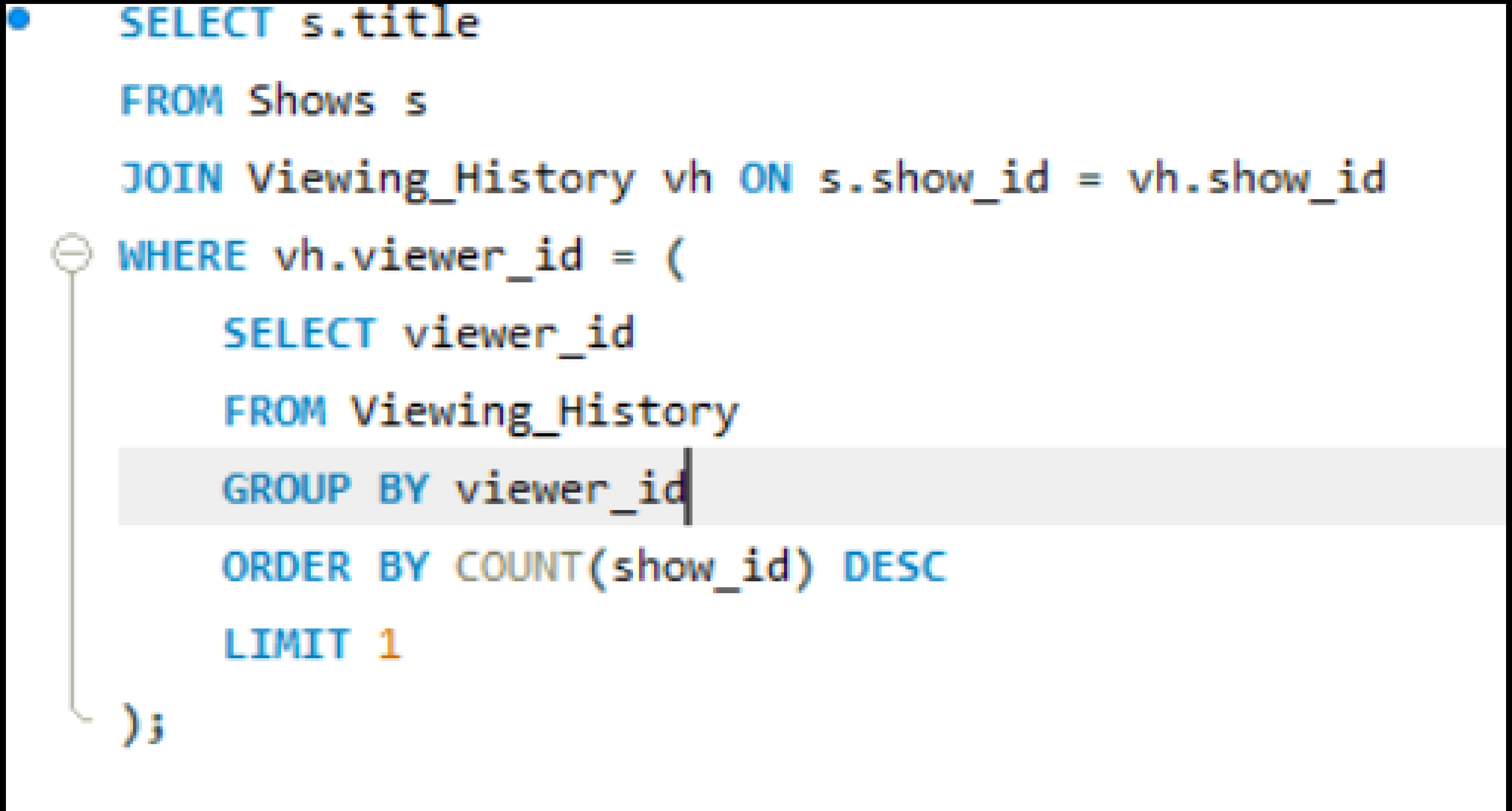


```sql
SELECT s.title, AVG(vh.rating) AS average_rating
FROM Shows s
JOIN Show_Reviews sr ON s.show_id = sr.show_id
JOIN Viewing_History vh ON sr.viewer_id = vh.viewer_id
GROUP BY s.show_id, s.title
ORDER BY average_rating DESC
LIMIT 5;
```

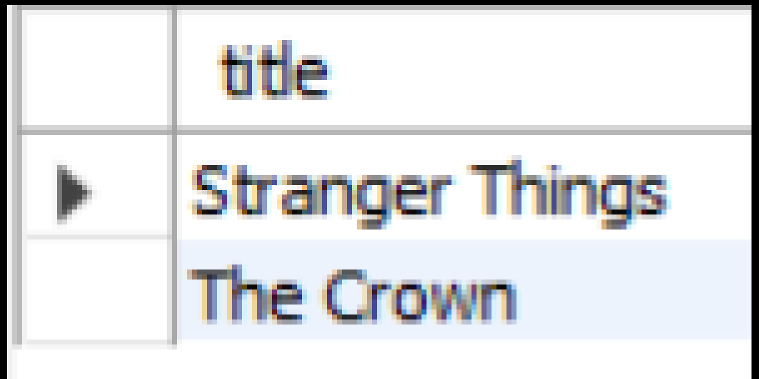


title	average_rating
Breaking Bad	10.0000
Black Mirror	9.0000
The Crown	9.0000
The Umbrella Academy	9.0000
The Office	9.0000

LIst Shows Watched by the Most Active Viewer

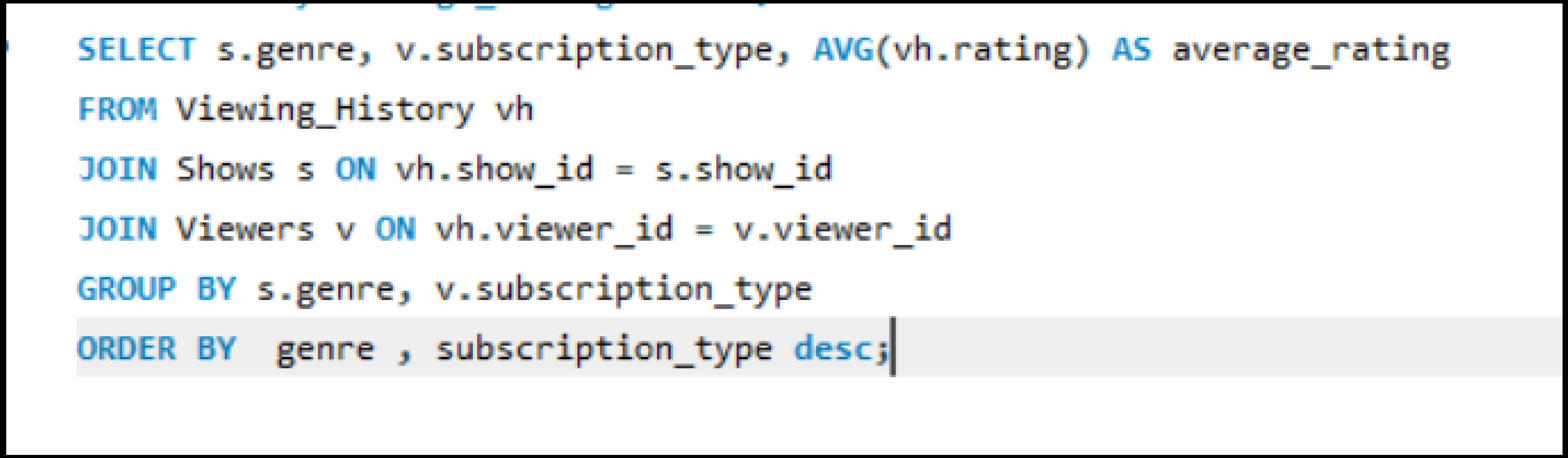


```
SELECT s.title
FROM Shows s
JOIN Viewing_History vh ON s.show_id = vh.show_id
WHERE vh.viewer_id = (
    SELECT viewer_id
    FROM Viewing_History
    GROUP BY viewer_id
    ORDER BY COUNT(show_id) DESC
    LIMIT 1
);
```

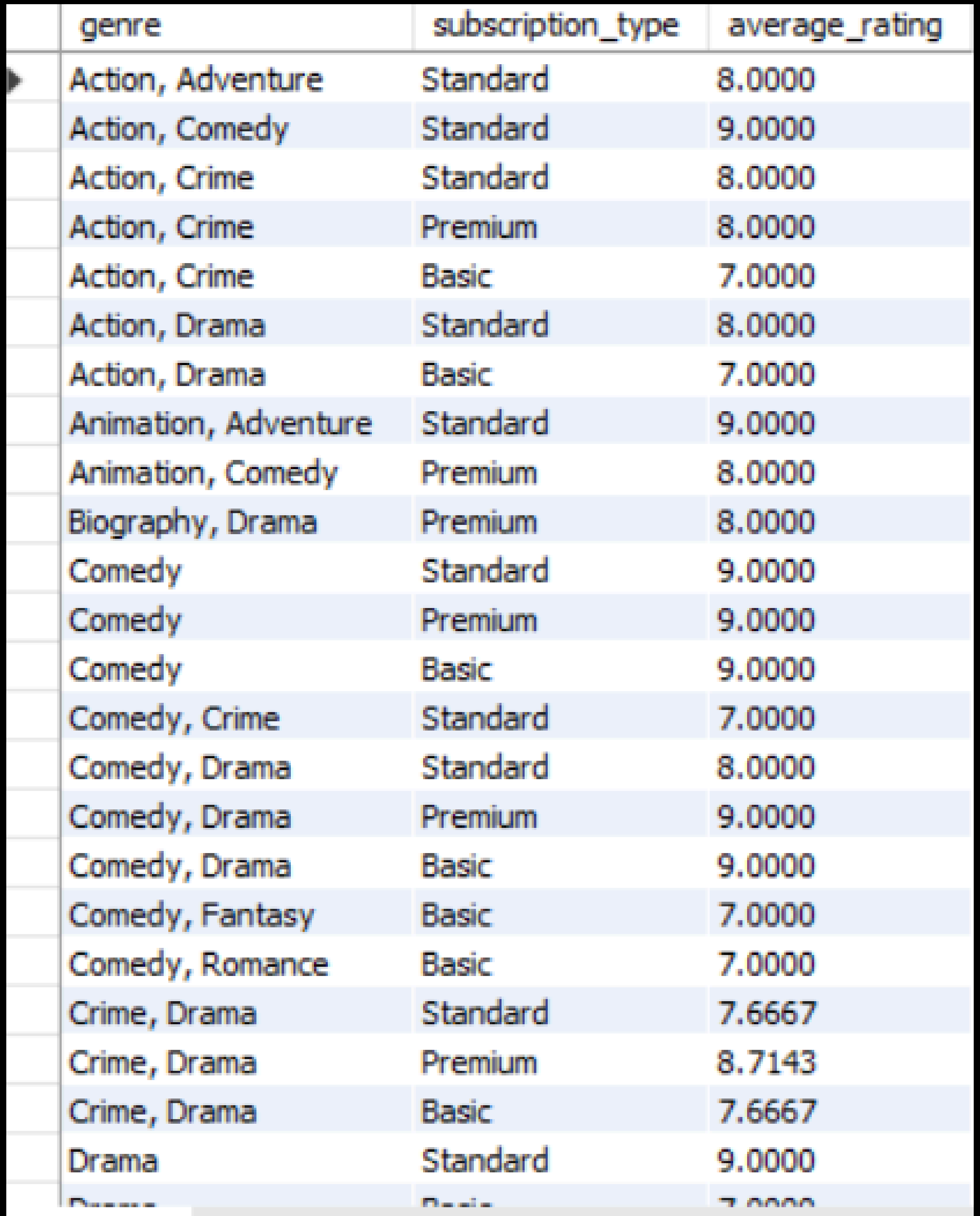


title
Stranger Things
The Crown

Calculate the average rating of shows grouped by genre and subscription type.

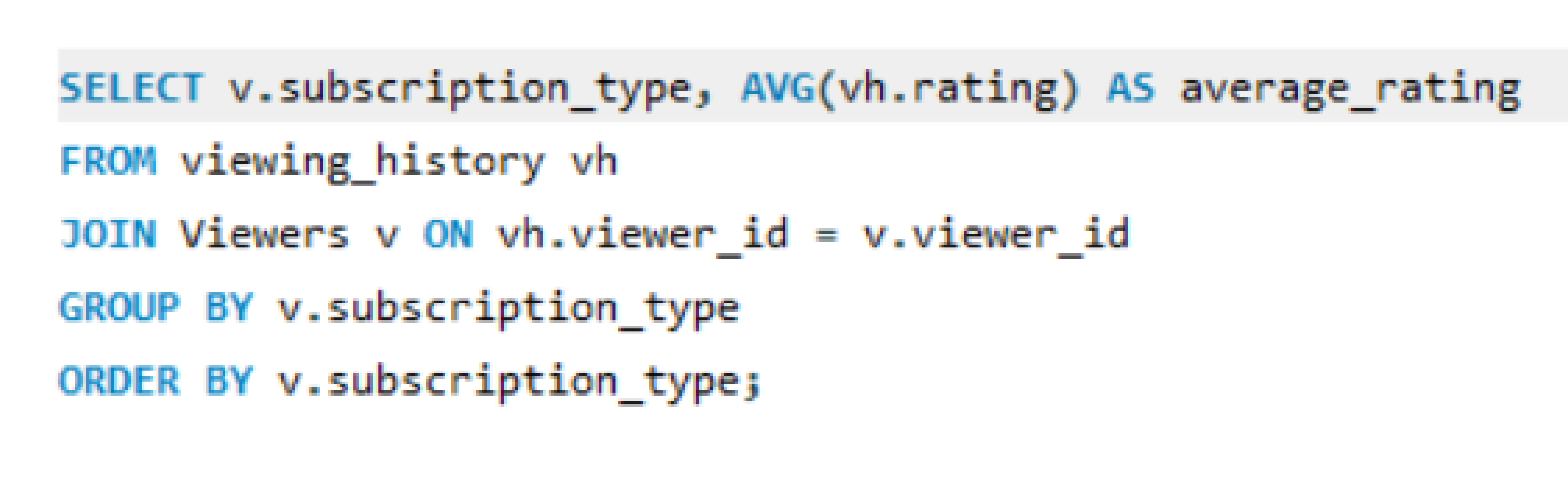


```sql
SELECT s.genre, v.subscription_type, AVG(vh.rating) AS average_rating
FROM Viewing_History vh
JOIN Shows s ON vh.show_id = s.show_id
JOIN Viewers v ON vh.viewer_id = v.viewer_id
GROUP BY s.genre, v.subscription_type
ORDER BY  genre , subscription_type desc;
```

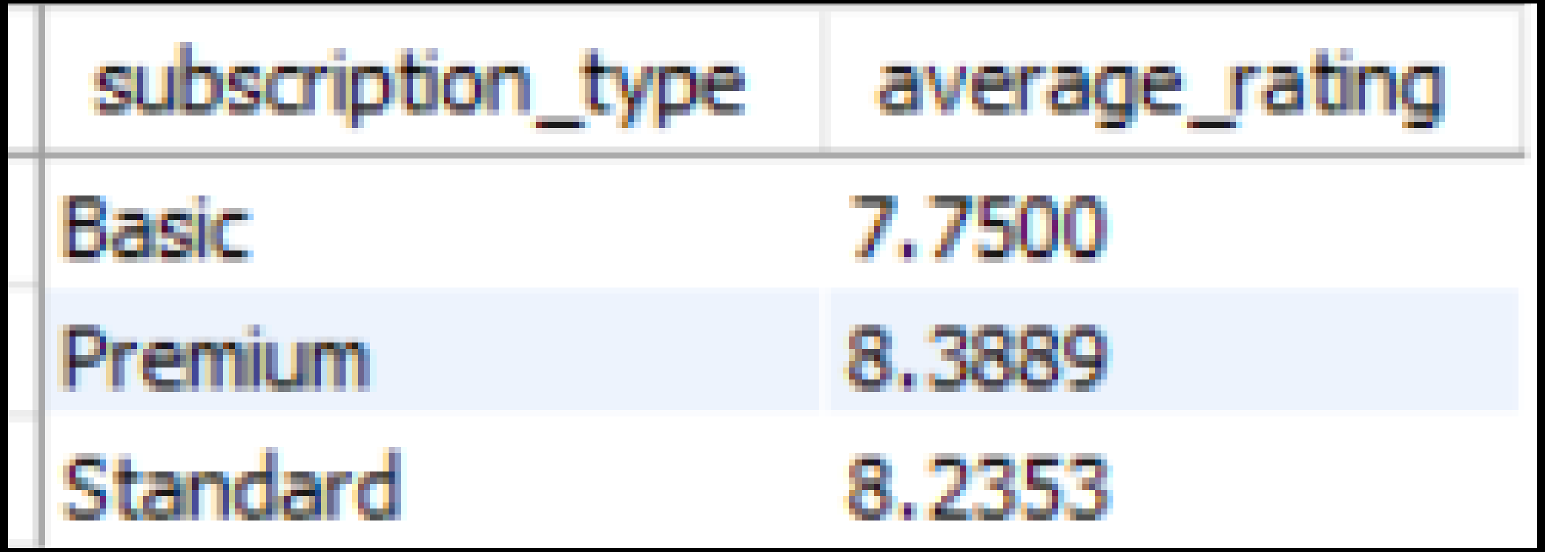


genre	subscription_type	average_rating
Action, Adventure	Standard	8.0000
Action, Comedy	Standard	9.0000
Action, Crime	Standard	8.0000
Action, Crime	Premium	8.0000
Action, Crime	Basic	7.0000
Action, Drama	Standard	8.0000
Action, Drama	Basic	7.0000
Animation, Adventure	Standard	9.0000
Animation, Comedy	Premium	8.0000
Biography, Drama	Premium	8.0000
Comedy	Standard	9.0000
Comedy	Premium	9.0000
Comedy	Basic	9.0000
Comedy, Crime	Standard	7.0000
Comedy, Drama	Standard	8.0000
Comedy, Drama	Premium	9.0000
Comedy, Drama	Basic	9.0000
Comedy, Fantasy	Basic	7.0000
Comedy, Romance	Basic	7.0000
Crime, Drama	Standard	7.6667
Crime, Drama	Premium	8.7143
Crime, Drama	Basic	7.6667
Drama	Standard	9.0000
Drama	Basic	7.0000

Analyze if there is any correlation between the average ratings given by viewers and their subscription type.

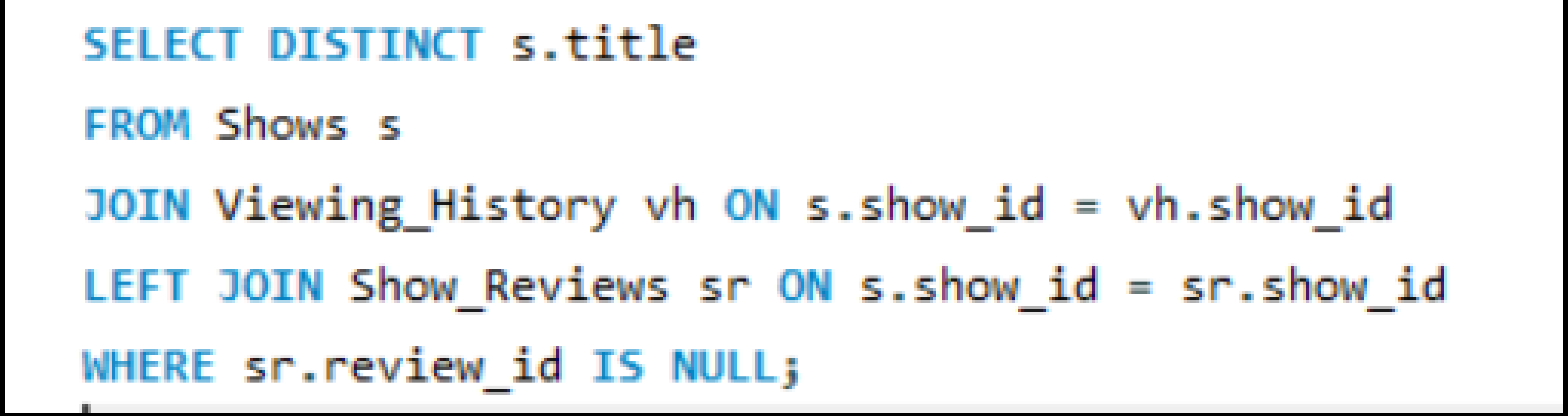


```sql
SELECT v.subscription_type, AVG(vh.rating) AS average_rating
FROM viewing_history vh
JOIN Viewers v ON vh.viewer_id = v.viewer_id
GROUP BY v.subscription_type
ORDER BY v.subscription_type;
```

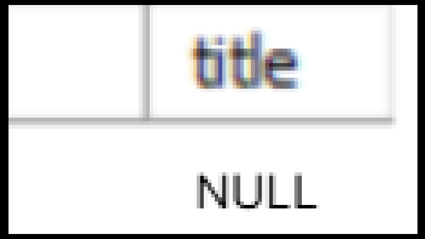


subscription_type	average_rating
Basic	7.7500
Premium	8.3889
Standard	8.2353

Find shows that have been watched but have no reviews.

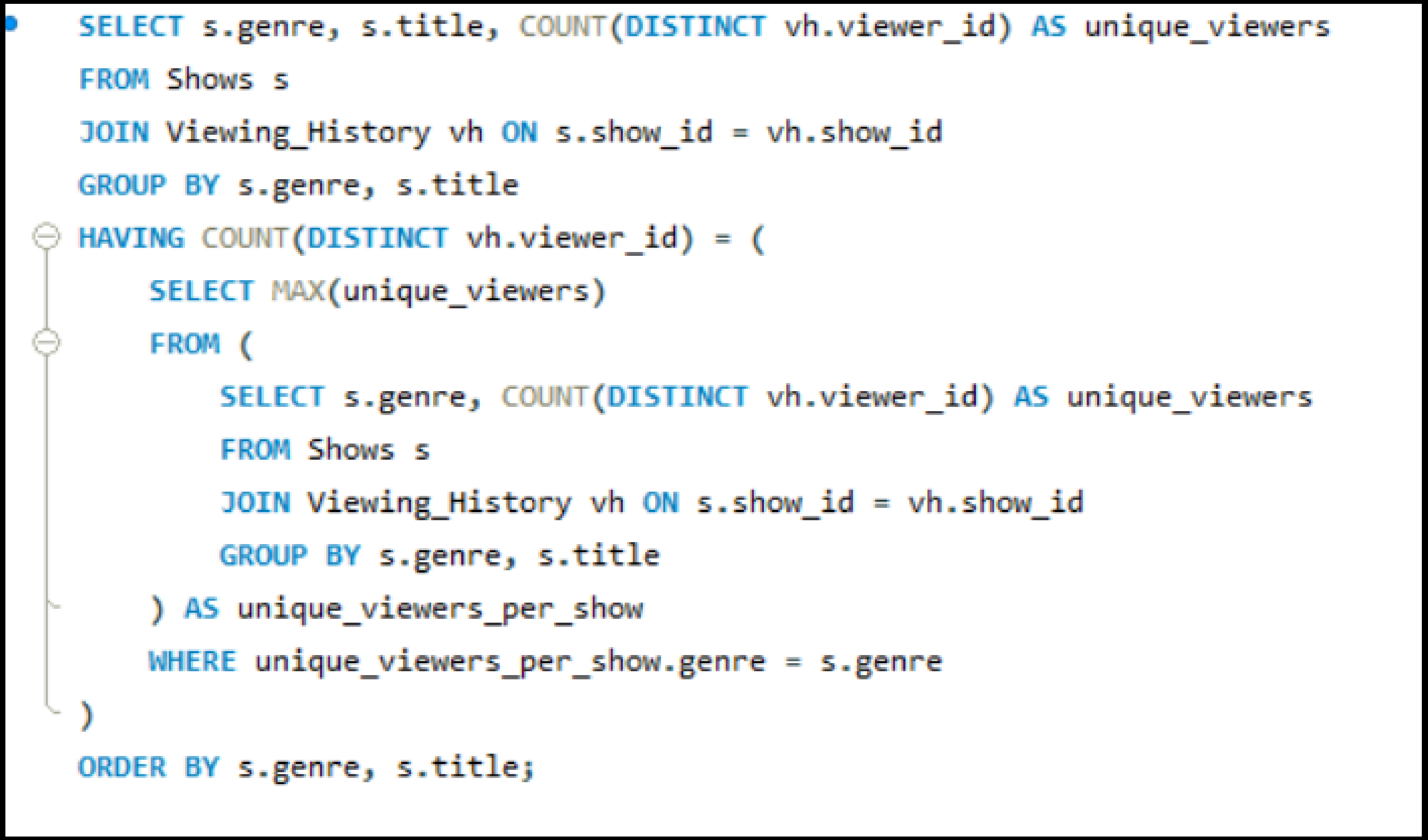


```sql
SELECT DISTINCT s.title
FROM Shows s
JOIN Viewing_History vh ON s.show_id = vh.show_id
LEFT JOIN Show_Reviews sr ON s.show_id = sr.show_id
WHERE sr.review_id IS NULL;
```

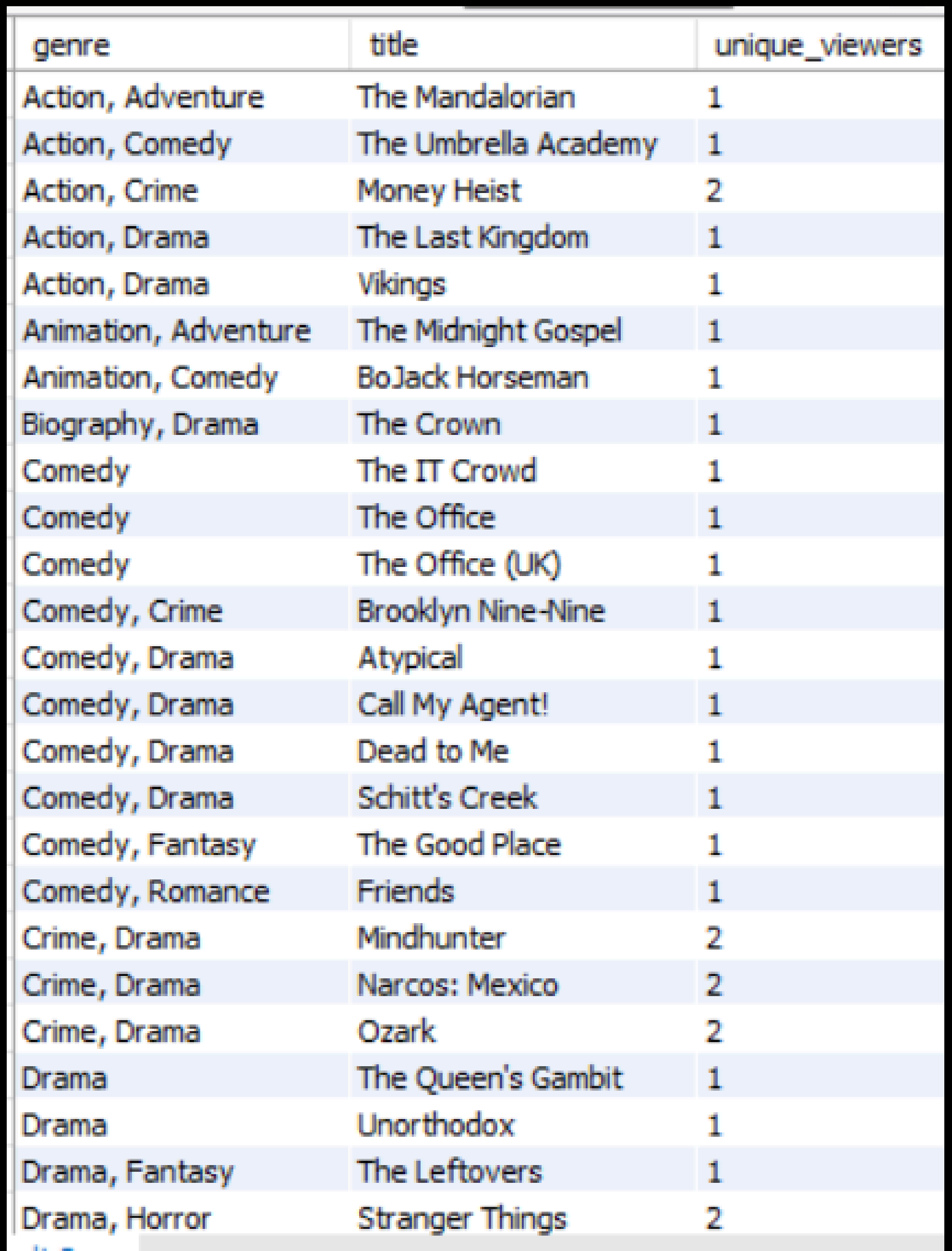


	title
	NULL

Identify the show with the highest number of unique viewers for each genre.

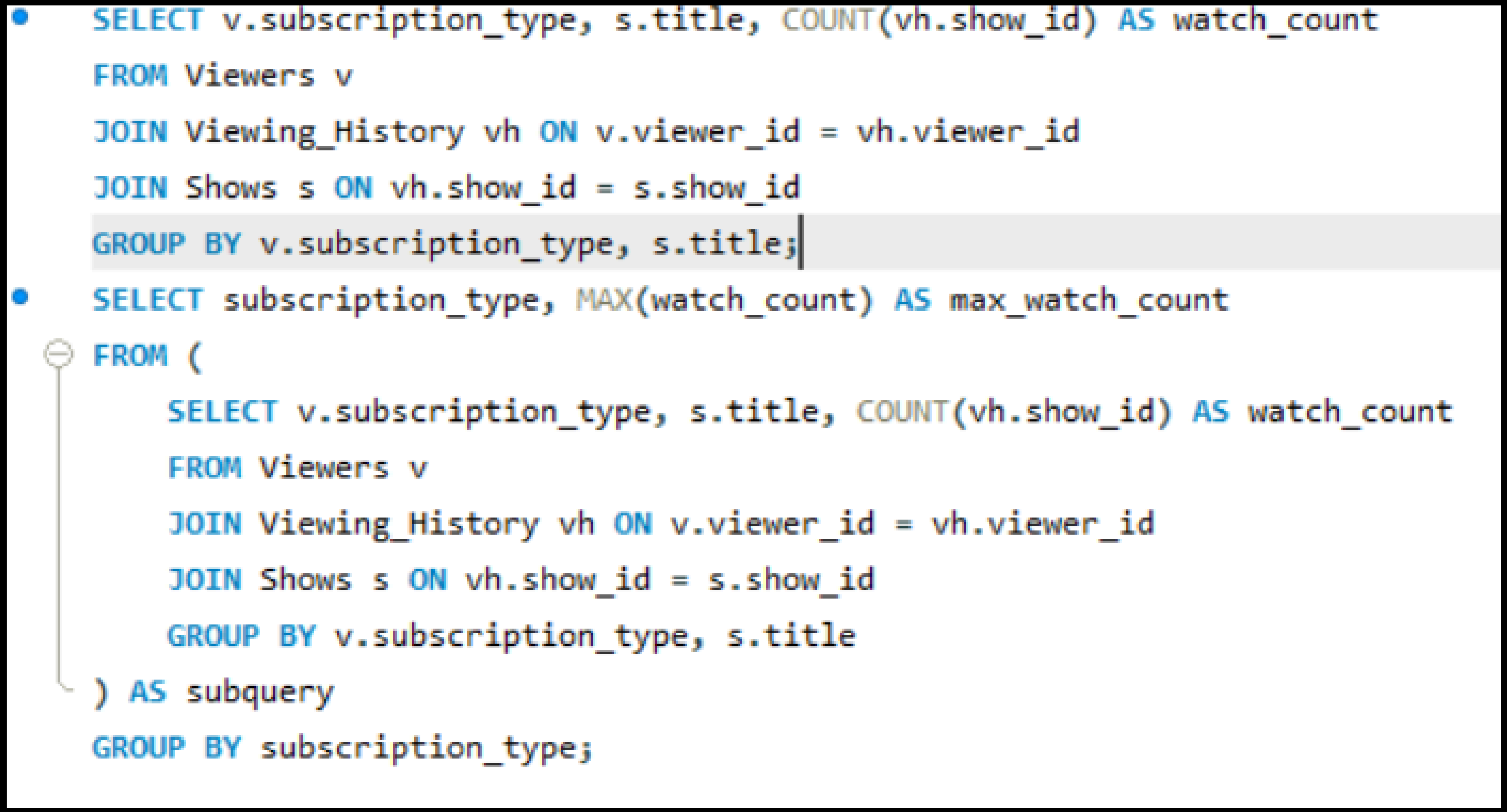


```
SELECT s.genre, s.title, COUNT(DISTINCT vh.viewer_id) AS unique_viewers
FROM Shows s
JOIN Viewing_History vh ON s.show_id = vh.show_id
GROUP BY s.genre, s.title
HAVING COUNT(DISTINCT vh.viewer_id) = (
    SELECT MAX(unique_viewers)
    FROM (
        SELECT s.genre, COUNT(DISTINCT vh.viewer_id) AS unique_viewers
        FROM Shows s
        JOIN Viewing_History vh ON s.show_id = vh.show_id
        GROUP BY s.genre, s.title
    ) AS unique_viewers_per_show
    WHERE unique_viewers_per_show.genre = s.genre
)
ORDER BY s.genre, s.title;
```

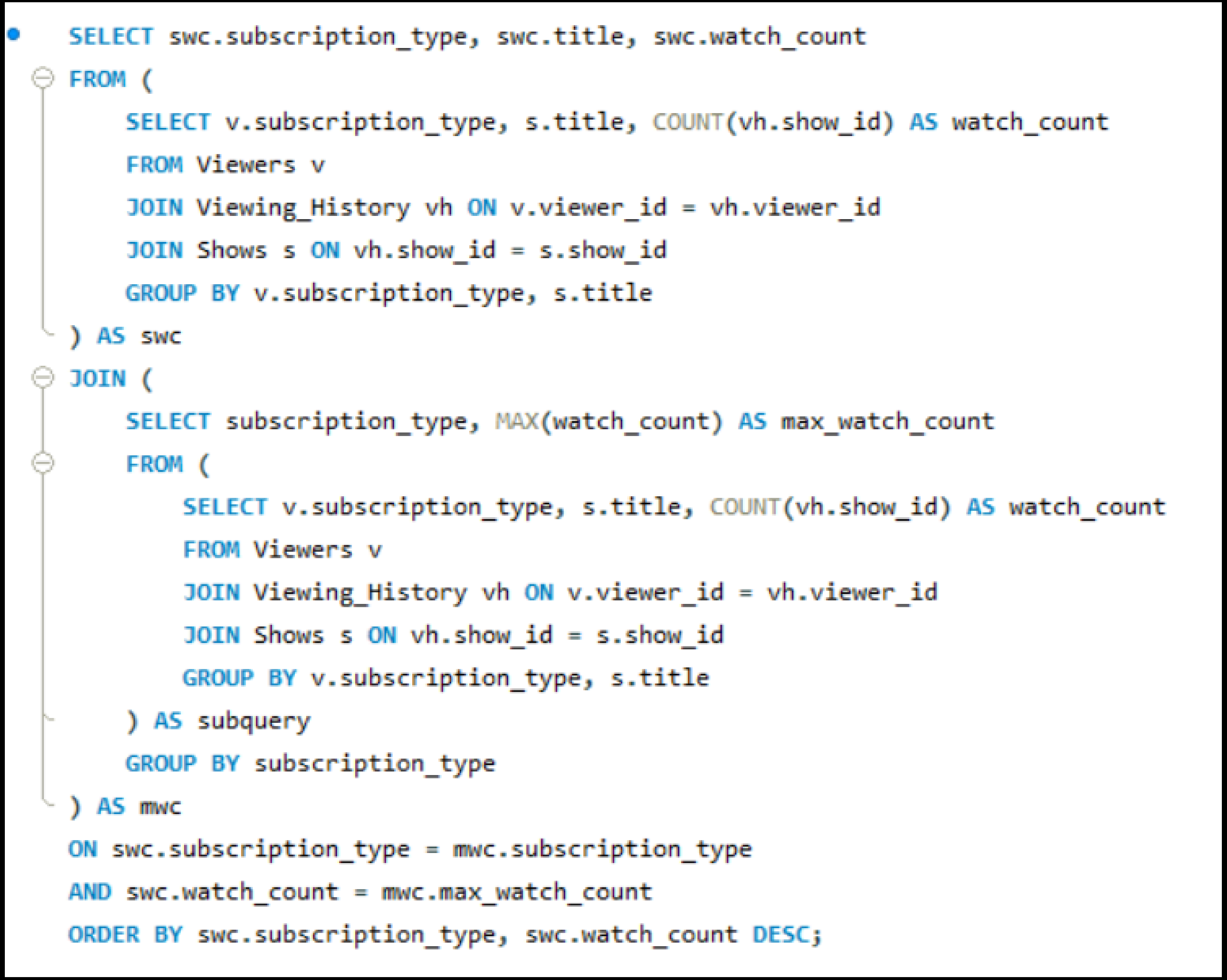


genre	title	unique_viewers
Action, Adventure	The Mandalorian	1
Action, Comedy	The Umbrella Academy	1
Action, Crime	Money Heist	2
Action, Drama	The Last Kingdom	1
Action, Drama	Vikings	1
Animation, Adventure	The Midnight Gospel	1
Animation, Comedy	BoJack Horseman	1
Biography, Drama	The Crown	1
Comedy	The IT Crowd	1
Comedy	The Office	1
Comedy	The Office (UK)	1
Comedy, Crime	Brooklyn Nine-Nine	1
Comedy, Drama	Atypical	1
Comedy, Drama	Call My Agent!	1
Comedy, Drama	Dead to Me	1
Comedy, Drama	Schitt's Creek	1
Comedy, Fantasy	The Good Place	1
Comedy, Romance	Friends	1
Crime, Drama	Mindhunter	2
Crime, Drama	Narcos: Mexico	2
Crime, Drama	Ozark	2
Drama	The Queen's Gambit	1
Drama	Unorthodox	1
Drama, Fantasy	The Leftovers	1
Drama, Horror	Stranger Things	2

Find viewers who have watched shows with all available parental guidelines (e.g., 'PG', 'R')



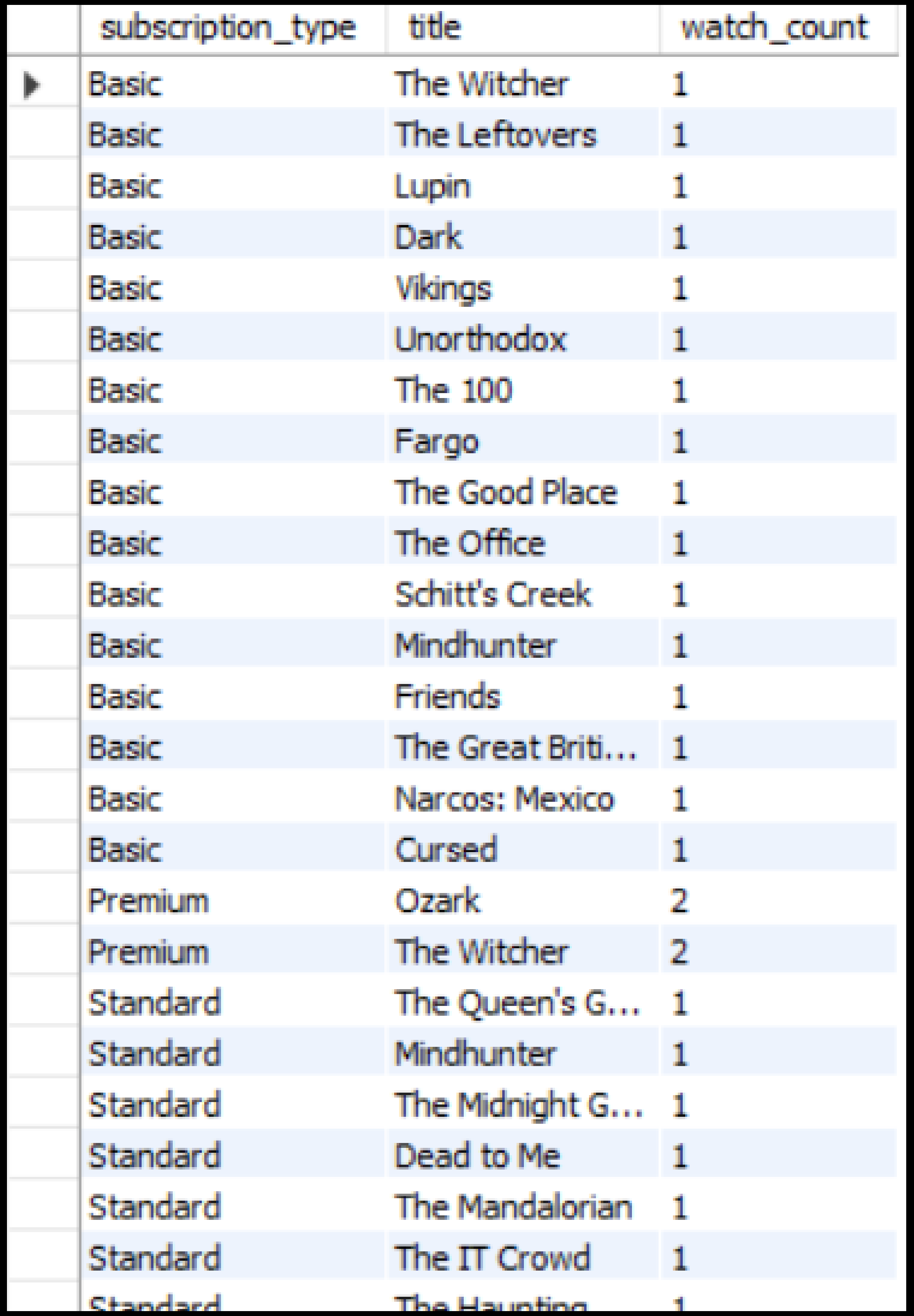
```sql
SELECT v.subscription_type, s.title, COUNT(vh.show_id) AS watch_count
FROM Viewers v
JOIN Viewing_History vh ON v.viewer_id = vh.viewer_id
JOIN Shows s ON vh.show_id = s.show_id
GROUP BY v.subscription_type, s.title;
SELECT subscription_type, MAX(watch_count) AS max_watch_count
FROM (
    SELECT v.subscription_type, s.title, COUNT(vh.show_id) AS watch_count
    FROM Viewers v
    JOIN Viewing_History vh ON v.viewer_id = vh.viewer_id
    JOIN Shows s ON vh.show_id = s.show_id
    GROUP BY v.subscription_type, s.title
) AS subquery
GROUP BY subscription_type;
```



```sql
SELECT swc.subscription_type, swc.title, swc.watch_count
FROM (
    SELECT v.subscription_type, s.title, COUNT(vh.show_id) AS watch_count
    FROM Viewers v
    JOIN Viewing_History vh ON v.viewer_id = vh.viewer_id
    JOIN Shows s ON vh.show_id = s.show_id
    GROUP BY v.subscription_type, s.title
) AS swc
JOIN (
    SELECT subscription_type, MAX(watch_count) AS max_watch_count
    FROM (
        SELECT v.subscription_type, s.title, COUNT(vh.show_id) AS watch_count
        FROM Viewers v
        JOIN Viewing_History vh ON v.viewer_id = vh.viewer_id
        JOIN Shows s ON vh.show_id = s.show_id
        GROUP BY v.subscription_type, s.title
    ) AS subquery
    GROUP BY subscription_type
) AS mwc
ON swc.subscription_type = mwc.subscription_type
AND swc.watch_count = mwc.max_watch_count
ORDER BY swc.subscription_type, swc.watch_count DESC;
```

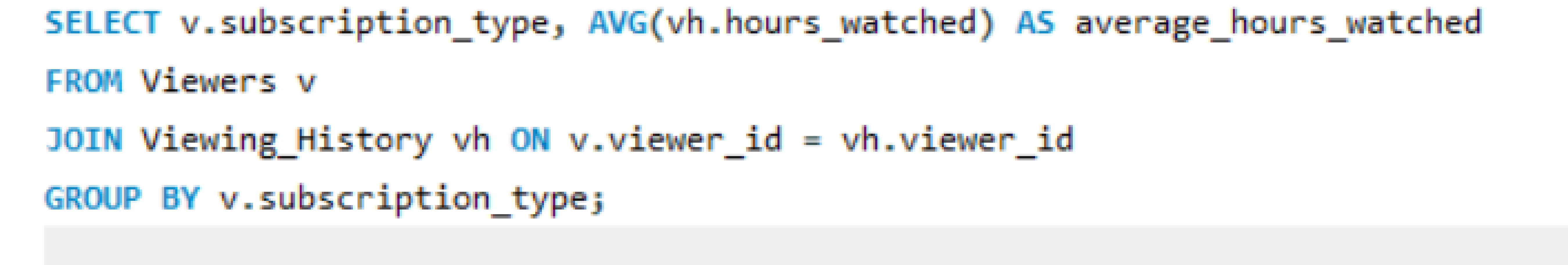
Find viewers who have watched shows with all available parental guidelines (e.g., 'PG', 'R')

output:

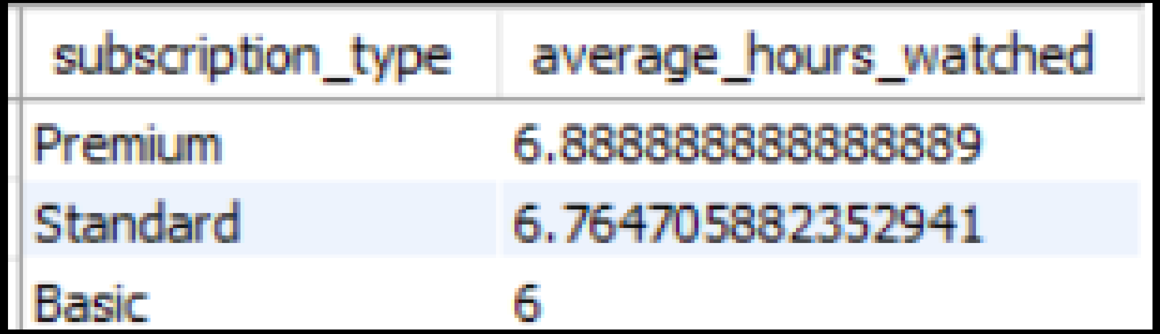


subscription_type	title	watch_count
Basic	The Witcher	1
Basic	The Leftovers	1
Basic	Lupin	1
Basic	Dark	1
Basic	Vikings	1
Basic	Unorthodox	1
Basic	The 100	1
Basic	Fargo	1
Basic	The Good Place	1
Basic	The Office	1
Basic	Schitt's Creek	1
Basic	Mindhunter	1
Basic	Friends	1
Basic	The Great Briti...	1
Basic	Narcos: Mexico	1
Basic	Cursed	1
Premium	Ozark	2
Premium	The Witcher	2
Standard	The Queen's G...	1
Standard	Mindhunter	1
Standard	The Midnight G...	1
Standard	Dead to Me	1
Standard	The Mandalorian	1
Standard	The IT Crowd	1
Standard	The Haunting	1

Calculate the average number of hours watched per viewer, broken down by subscription type.



```
SELECT v.subscription_type, AVG(vh.hours_watched) AS average_hours_watched
FROM Viewers v
JOIN Viewing_History vh ON v.viewer_id = vh.viewer_id
GROUP BY v.subscription_type;
```



subscription_type	average_hours_watched
Premium	6.888888888888889
Standard	6.764705882352941
Basic	6

THANKYOU!