Goodreads Book Ratings

Covers ratings for books on Goodreads

Single table, 13525 rows, 20 columns

Downloaded from https://mavenanalytics.io/data-playground

Sourced from University of California San Diego (UCSD)

Public domain license

Overall objective: Generate lists of books that people might be interested in reading

Identify the set of works with the highest average ratings (rounded to two decimal places).

```sql
SELECT TOP 10 original_title, author, original_publication_year, CONVERT(DECIMAL(3, 2),
avg_rating) AS average_rating
FROM dbo.goodreads_works
ORDER BY average_rating DESC, author ASC, original_publication_year ASC
;
```

original_title	author	original_publication_year	average_rating
The Way of Kings, Part 2	Brandon Sanderson	2010	4.8
Words of Radiance	Brandon Sanderson	2014	4.8
Words of Radiance, Part 2	Brandon Sanderson	2014	4.8
Oathbringer	Brandon Sanderson	2017	4.7
Saga: Book One	Brian K. Vaughan	2014	4.7
A Court of Mist and Fury	Sarah J. Maas	2016	4.7
Neverseen	Shannon Messenger	2015	4.7
The Long and Winding Road	T.J. Klune	2017	4.7
Stars & Stripes	Abigail Roux	2012	4.6
Gemina	Amie Kaufman	2016	4.6

Focus on the set of works with the highest average rating, and then list some similar works.

```sql
SELECT top_works_by_highest_avg_rating_with_similar_book.author,
       top_works_by_highest_avg_rating_with_similar_book.original_title,
       top_works_by_highest_avg_rating_with_similar_book.original_publication_year,
       top_works_by_highest_avg_rating_with_similar_book.average_rating AS
avg_rating_of_original_work,
       -- top_works_by_highest_avg_rating_with_similar_book.avg_rating_rank,
       -- top_works_by_highest_avg_rating_with_similar_book.similar_book,
       goodreads_works.author AS author_of_similar_work,
       goodreads_works.original_title AS original_title_of_similar_work,
       goodreads_works.original_publication_year AS
original_publication_year_of_similar_work,
       CONVERT(DECIMAL(3, 2), goodreads_works.avg_rating) AS avg_rating_of_similar_work
FROM (
```

```sql
        SELECT author, original_title, original_publication_year, average_rating,
avg_rating_rank, TRIM(value) AS similar_book
        FROM (
                SELECT TOP 10 author, original_title, original_publication_year,
CONVERT(DECIMAL(3, 2), avg_rating) AS average_rating,
                    RANK() OVER (ORDER BY avg_rating DESC) AS avg_rating_rank,
similar_books
                FROM dbo.goodreads_works
                WHERE similar_books IS NOT NULL
                ORDER BY average_rating DESC
        ) AS top_ten_works_by_avg_rating_with_rank
        CROSS APPLY STRING_SPLIT (similar_books, ',')
        WHERE avg_rating_rank = 1
) AS top_works_by_highest_avg_rating_with_similar_book
INNER JOIN
dbo.goodreads_works
ON top_works_by_highest_avg_rating_with_similar_book.similar_book =
goodreads_works.work_id
ORDER BY top_works_by_highest_avg_rating_with_similar_book.author ASC,
        top_works_by_highest_avg_rating_with_similar_book.original_publication_year ASC,
        avg_rating_of_similar_work DESC,
        author_of_similar_work ASC,
        goodreads_works.original_publication_year ASC
;
```

author	original_title	original_publication_year	avg_rating_of_original_work	author_of_similar_work	original_title_of_similar_work	original_publication_year_of_similar_work	avg_rating_of_similar_work
Brandon Sanderson	The Way of Kings, Part 2	2010	4.8	George R.R. Martin	A Dance With Dragons: Part 2 After The Feast	2011	4.4
Brandon Sanderson	Words of Radiance	2014	4.8	Brent Weeks	The Broken Eye	2014	4.5
Brandon Sanderson	Words of Radiance	2014	4.8	Michael J. Sullivan	Heir of Novron	2012	4.5
Brandon Sanderson	Words of Radiance	2014	4.8	Brian McClellan	The Crimson Campaign	2014	4.4
Brandon Sanderson	Words of Radiance	2014	4.8	Anthony Ryan	Tower Lord	2014	4.2
Brandon Sanderson	Words of Radiance	2014	4.8	Brian Staveley	The Providence of Fire	2015	4.2
Brandon Sanderson	Words of Radiance	2014	4.8	Peter V. Brett	The Daylight War	2013	4.2
Brandon Sanderson	Words of Radiance	2014	4.8	Scott Lynch	The Republic of Thieves	2013	4.2

Note: Not all works in the top ten list of works by average rating will have any similar works listed in the dataset.

Identify some other highly rated works and some of their highly rated similar works.

```sql
SELECT similar_works_with_ranks.author,
       similar_works_with_ranks.original_title,
       similar_works_with_ranks.original_publication_year,
       similar_works_with_ranks.avg_rating_of_original_work,
       similar_works_with_ranks.author_of_similar_work,
       similar_works_with_ranks.original_title_of_similar_work,
       similar_works_with_ranks.original_publication_year_of_similar_work,
       similar_works_with_ranks.avg_rating_of_similar_work
FROM (
      SELECT
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed_for_jo
in_with_ranks_for_similar_books.work_id,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.author,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.original_title,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.original_publication_year,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.avg_rating_of_original_work,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.author_of_similar_work,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.original_title_of_similar_work,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.original_publication_year_of_similar_work,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join_with_ranks_for_similar_books.avg_rating_of_similar_work,
            RANK() OVER (PARTITION BY work_id ORDER BY avg_rating_of_similar_work DESC)
AS rank_of_avg_rating_of_similar_work
      FROM (
            SELECT
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed_for_jo
in.work_id,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join.author,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join.original_title,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join.original_publication_year,

      top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
_for_join.average_rating AS avg_rating_of_original_work,
                  dbo.goodreads_works.author AS author_of_similar_work,
```

```sql
                    dbo.goodreads_works.original_title AS
original_title_of_similar_work,
                    dbo.goodreads_works.original_publication_year AS
original_publication_year_of_similar_work,
                    CONVERT(DECIMAL(3,2), dbo.goodreads_works.avg_rating) AS
avg_rating_of_similar_work
            FROM (
                    SELECT
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed.work_i
d,
        top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
.author,
        top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
.original_title,
        top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
.original_publication_year,
        top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
.average_rating,
                        TRIM(value) AS similar_book
                    FROM (
                        SELECT *
                        FROM (
                            SELECT
top_ten_works_by_average_rating_with_similar_books.work_id,
        top_ten_works_by_average_rating_with_similar_books.author,
        top_ten_works_by_average_rating_with_similar_books.original_title,
        top_ten_works_by_average_rating_with_similar_books.original_publication_year,
        top_ten_works_by_average_rating_with_similar_books.average_rating,
                                RANK() OVER (ORDER BY
top_ten_works_by_average_rating_with_similar_books.average_rating DESC) AS
average_rating_rank,
        top_ten_works_by_average_rating_with_similar_books.similar_books
                            FROM (
                                SELECT TOP 10 work_id, author, original_title,
original_publication_year, CONVERT(DECIMAL(3, 2), avg_rating) AS average_rating,
similar_books
                                FROM dbo.goodreads_works
                                ORDER BY average_rating DESC
                            ) AS top_ten_works_by_average_rating_with_similar_books
                            WHERE similar_books IS NOT NULL
                        ) AS
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank
                        CROSS APPLY STRING_SPLIT(similar_books, ',')
                        WHERE 1 < average_rating_rank
                    ) AS
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed
            ) AS
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed_for_jo
in
```

```sql
                INNER JOIN dbo.goodreads_works
                    ON
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed_for_jo
in.similar_book =
                        dbo.goodreads_works.work_id
            ) AS
top_ten_works_by_average_rating_with_similar_books_and_average_rating_rank_trimmed_for_jo
in_with_ranks_for_similar_books
        -- ORDER BY avg_rating_of_original_work DESC, author ASC,
rank_of_avg_rating_of_similar_work ASC, author_of_similar_work ASC
) AS similar_works_with_ranks
WHERE rank_of_avg_rating_of_similar_work = 1
ORDER BY avg_rating_of_original_work DESC, author ASC, rank_of_avg_rating_of_similar_work
ASC, author_of_similar_work ASC
;
```

author	original_title	original_publication_year	avg_rating_of_original_work	author_of_similar_work	original_title_of_similar_work	original_publication_year_of_similar_work	avg_rating_of_similar_work
Brandon Sanderson	Oathbringer	2017	4.7	Brian McClellan	Sins of Empire	2017	4.5
Brandon Sanderson	Oathbringer	2017	4.7	Robert Jackson Bennett	City of Miracles	2017	4.5
Brian K. Vaughan	Saga: Book One	2014	4.7	Jay Faerber	Copperhead, Vol. 2	2015	3.9
Brian K. Vaughan	Saga: Book One	2014	4.7	Joshua Dysart	Harbinger, Volume 1: Omega Rising	2013	3.9
Brian K. Vaughan	Saga: Book One	2014	4.7	Rick Remender	Low, Volume 2: Before the Dawn Burns Us	2015	3.9
Sarah J. Maas	A Court of Mist and Fury	2016	4.7	Jennifer Lynn Barnes	The Long Game	2016	4.4
Sarah J. Maas	A Court of Mist and Fury	2016	4.7	Morgan Rhodes	Frozen Tides	2015	4.4
Sarah J. Maas	A Court of Mist and Fury	2016	4.7	Paula Weston	Burn (The Rephaim #4)	2015	4.4
Shannon Messenger	Neverseen	2015	4.7	Jay Kristoff	Endsinger	2014	4.3
Hiromu Arakawa	Gang noLian Jin Shu Shi 3	2002	4.6	Tsugumi Ohba	desunoto #4 (Desu Noto) Koigokoro (Lian Xin)	2004	4.4

Note: Not all works in the top ten list of works by average rating will have any similar works listed in the dataset.

Identify the set of works with the highest rating count.

```sql
SELECT TOP 10 original_title, author, original_publication_year, ratings_count
FROM dbo.goodreads_works
ORDER BY ratings_count DESC, author ASC, original_publication_year ASC
;
```

original_title	author	original_publication_year	ratings_count
The Hunger Games	Suzanne Collins	2008	5066596
Harry Potter and the Philosopher's Stone	J.K. Rowling	1997	4972886
Twilight	Stephenie Meyer	2005	3992661
To Kill a Mockingbird	Harper Lee	1960	3402363
The Great Gatsby	F. Scott Fitzgerald	1925	2852789
The Fault in Our Stars	John Green	2012	2564656
Divergent	Veronica Roth	2011	2277881
Pride and Prejudice	Jane Austen	1813	2239951
The Hobbit : or There and Back Again	J.R.R. Tolkien	1937	2228361
The Catcher in the Rye	J.D. Salinger	1951	2166748

Focus on the set of works with the highest rating count, and then list some similar works.

```sql
SELECT top_works_by_rating_count_with_rank_trimmed.author,
       top_works_by_rating_count_with_rank_trimmed.original_title,
       top_works_by_rating_count_with_rank_trimmed.original_publication_year,
       top_works_by_rating_count_with_rank_trimmed.ratings_count AS
ratings_count_of_original_work,
       dbo.goodreads_works.author AS author_of_similar_work,
       dbo.goodreads_works.original_title AS original_title_of_similar_work,
       goodreads_works.original_publication_year AS
original_publication_year_of_similar_work,
       dbo.goodreads_works.ratings_count AS ratings_count_of_simiarl_work
FROM (
       SELECT top_works_by_rating_count_with_rank.author,
              top_works_by_rating_count_with_rank.original_title,
              top_works_by_rating_count_with_rank.original_publication_year,
              top_works_by_rating_count_with_rank.ratings_count,
              TRIM(value) AS similar_book
       FROM (
              SELECT TOP 10 author, original_title, original_publication_year,
CONVERT(DECIMAL(3, 2), avg_rating) AS average_rating, ratings_count,
                     RANK() OVER (ORDER BY ratings_count DESC) AS ratings_count_rank,
                     similar_books
              FROM dbo.goodreads_works
              WHERE similar_books IS NOT NULL
              ORDER BY ratings_count DESC
       ) as top_works_by_rating_count_with_rank
       CROSS APPLY STRING_SPLIT (similar_books, ',')
       WHERE ratings_count_rank = 1
) AS top_works_by_rating_count_with_rank_trimmed
INNER JOIN dbo.goodreads_works
ON top_works_by_rating_count_with_rank_trimmed.similar_book =
       dbo.goodreads_works.work_id
ORDER BY ratings_count_of_simiarl_work DESC
;
```

author	original_title	original_publication_year	ratings_count_of_original_work	author_of_similar_work	original_title_of_similar_work	original_publication_year_of_similar_work	ratings_count_of_simiarl_work
Suzanne Collins	The Hunger Games	2008	5066596	Cassandra Clare	Clockwork Prince	2011	339508
Suzanne Collins	The Hunger Games	2008	5066596	Richelle Mead	Last Sacrifice	2010	222434
Suzanne Collins	The Hunger Games	2008	5066596	Becca Fitzpatrick	Silence	2011	202917
Suzanne Collins	The Hunger Games	2008	5066596	Karen Marie Moning	Shadowfever	2011	81459
Suzanne Collins	The Hunger Games	2008	5066596	Moira Young	Blood Red Road	2011	53083
Suzanne Collins	The Hunger Games	2008	5066596	Mary E. Pearson	The Adoration of Jenna Fox	2008	40908
Suzanne Collins	The Hunger Games	2008	5066596	Cory Doctorow	Little Brother	2008	40518
Suzanne Collins	The Hunger Games	2008	5066596	Scott Westerfeld	Behemoth	2010	32571
Suzanne Collins	The Hunger Games	2008	5066596	Meg Cabot	Ninth Key	2001	32557
Suzanne Collins	The Hunger Games	2008	5066596	Colleen Houck	Tiger's Voyage	2011	27354
Suzanne Collins	The Hunger Games	2008	5066596	Caragh M. O'Brien	Prized	2011	21054
Suzanne Collins	The Hunger Games	2008	5066596	Melina Marchetta	Froi of the Exiles	2011	14532
Suzanne Collins	The Hunger Games	2008	5066596	Maria V. Snyder	Outside In	2011	12438
Suzanne Collins	The Hunger Games	2008	5066596	Cate Tiernan	Dark Magick	2001	10862
Suzanne Collins	The Hunger Games	2008	5066596	Hugh Howey	Wool 5: The Stranded	2012	8654
Suzanne Collins	The Hunger Games	2008	5066596	L.A. Meyer	In the Belly of the Bloodhound: Being an Account of a Particularly Peculiar Adventure in the Life of Jacky Faber	2006	8319
Suzanne Collins	The Hunger Games	2008	5066596	Monica Hughes	Invitation to The Game	1991	3924

Identify some other works with high rating counts along with similar works with high rating counts.

```sql
SELECT
top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filtered.au
thor,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.original_title,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.original_publication_year,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.ratings_count AS ratings_count_of_original_work,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.author_of_similar_work,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.original_title_of_similar_work,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.original_publication_year_of_similar_work,
        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filt
ered.ratings_count_for_similar_work
FROM (
        SELECT
top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.work_id,

        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.auth
or,

        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.orig
inal_title,

        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.orig
inal_publication_year,

        top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.rati
ngs_count,
                dbo.goodreads_works.author AS author_of_similar_work,
                dbo.goodreads_works.original_title AS original_title_of_similar_work,
                dbo.goodreads_works.original_publication_year AS
original_publication_year_of_similar_work,
                dbo.goodreads_works.ratings_count AS ratings_count_for_similar_work,
                DENSE_RANK() OVER (PARTITION BY
top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.work_id
ORDER BY goodreads_works.ratings_count DESC)
                        AS ratings_count_rank_for_similar_work
        FROM (
                SELECT top_ten_works_with_similar_books_and_rank_filtered.work_id,
                        top_ten_works_with_similar_books_and_rank_filtered.author,
                        top_ten_works_with_similar_books_and_rank_filtered.original_title,

        top_ten_works_with_similar_books_and_rank_filtered.original_publication_year,
                        top_ten_works_with_similar_books_and_rank_filtered.ratings_count,
                        top_ten_works_with_similar_books_and_rank_filtered.similar_book
                FROM (
                        SELECT top_ten_works_with_similar_books_and_rank.work_id,
                                top_ten_works_with_similar_books_and_rank.author,
```

```sql
                        top_ten_works_with_similar_books_and_rank.original_title,

        top_ten_works_with_similar_books_and_rank.original_publication_year,
                        top_ten_works_with_similar_books_and_rank.ratings_count,
                        DENSE_RANK() OVER (ORDER BY
    top_ten_works_with_similar_books_and_rank.ratings_count DESC) AS ratings_count_rank,
                        top_ten_works_with_similar_books_and_rank.similar_books,
                        TRIM(value) as similar_book
                    FROM (
                        SELECT TOP 10 work_id, author, original_title,
    original_publication_year, ratings_count, similar_books
                        FROM dbo.goodreads_works
                        ORDER BY ratings_count DESC
                    ) AS top_ten_works_with_similar_books_and_rank
                    CROSS APPLY STRING_SPLIT(similar_books, ',')
                ) AS top_ten_works_with_similar_books_and_rank_filtered
                WHERE 1 < ratings_count_rank
        ) AS top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join
        INNER JOIN dbO.goodreads_works
        ON
top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join.similar_boo
k =
                dbo.goodreads_works.work_id
) AS
top_ten_works_with_similar_books_and_rank_filtered_and_trimmed_for_inner_join_filtered
WHERE ratings_count_rank_for_similar_work = 1
ORDER BY ratings_count DESC
;
```

author	original_title	original_publication_year	ratings_count_of_original_work	author_of_similar_work	original_title_of_similar_work	original_publication_year_of_similar_work	ratings_count_for_similar_work
J.K. Rowling	Harry Potter and the Philosopher's Stone	1997	4972886	Rick Riordan	The Mark of Athena	2012	314931
Stephenie Meyer	Twilight	2005	3992661	Richelle Mead	Shadow Kiss	2008	284377
Harper Lee	To Kill a Mockingbird	1960	3402363	Louisa May Alcott	Little Women	1868	1334710
F. Scott Fitzgerald	The Great Gatsby	1925	2852789	Oscar Wilde	The Picture of Dorian Gray	1890	694769
John Green	The Fault in Our Stars	2012	2564656	Marie Lu	Legend	2011	328100
Veronica Roth	Divergent	2011	2277881	Lauren Oliver	Delirium	2011	358789
Jane Austen	Pride and Prejudice	1813	2239951	Frances Hodgson Burnett	A Little Princess	1905	222187
J.R.R. Tolkien	The Hobbit : or There and Back Again	1937	2228361	C.S. Lewis	The Voyage of the Dawn Treader	1952	329310
J.D. Salinger	The Catcher in the Rye	1951	2166748	John Steinbeck	The Grapes of Wrath	1939	566015

Identify the authors with the highest weighted mean of the average ratings among their works.

```sql
SELECT TOP 10 author,
        CONVERT(DECIMAL(3,2), SUM(ratings_count_times_avg_rating)/SUM(ratings_count)) AS
weighted_mean_of_avg_ratings_for_author
FROM (
        SELECT work_id, author, original_title, ratings_count, avg_rating,
(ratings_count*avg_rating) AS ratings_count_times_avg_rating
FROM dbo.goodreads_works
) AS authors_their_works_and_ratings_count_times_avg_rating
GROUP BY author
ORDER BY weighted_mean_of_avg_ratings_for_author DESC,
        author ASC
;
```

author	weighted_mean_of_avg_ratings_for_author
Angie Thomas	4.6
Bryan Stevenson	4.6
Haruichi Furudate	4.6
Sarah Noffke	4.6
T. Kingfisher	4.6
T.M. Frazier	4.6
Francine Rivers	4.54
Jorge Luis Borges	4.54
Patrick Rothfuss	4.54
Hiromu Arakawa	4.51

Identify the set of authors with the highest weighted mean of average ratings among their works, and then showcase their works.

```sql
SELECT authors_with_top_weighted_means.author,
        authors_with_top_weighted_means.weighted_mean_of_avg_ratings AS
weighted_mean_of_avg_ratings_for_author,
        dbo.goodreads_works.original_title,
        dbo.goodreads_works.original_publication_year,
        CONVERT(DECIMAL(3,2), dbo.goodreads_works.avg_rating) AS average_rating
FROM (
        SELECT *
        FROM (
                SELECT *,
                        RANK() OVER (ORDER BY weighted_mean_of_avg_ratings DESC) AS
weighted_mean_rank
                FROM (
                        SELECT TOP 10 author,
                                CONVERT(DECIMAL(3,2),
SUM(ratings_count_times_avg_rating)/SUM(ratings_count)) AS weighted_mean_of_avg_ratings
                        FROM (
                                SELECT work_id, author, original_title, ratings_count,
avg_rating, (ratings_count*avg_rating) AS ratings_count_times_avg_rating
                                FROM dbo.goodreads_works
                        ) AS authors_their_works_and_ratings_count_times_avg_rating
```

```sql
                    GROUP BY author
                    ORDER BY weighted_mean_of_avg_ratings DESC
            ) AS authors_with_weighted_means
        ) AS authors_with_weighted_means_and_ranks
        WHERE weighted_mean_rank = 1
) AS authors_with_top_weighted_means
INNER JOIN dbo.goodreads_works
ON authors_with_top_weighted_means.author = dbo.goodreads_works.author
ORDER BY weighted_mean_of_avg_ratings DESC,
        author ASC
;
```

author	weighted_mean_of_avg_ratings_for_author	original_title	original_publication_year	average_rating
Angie Thomas	4.6	The Hate U Give	2017	4.6
Bryan Stevenson	4.6	Just Mercy: A Story of Justice and Redemption	2014	4.6
Haruichi Furudate	4.6	haikiyu!! 1	2012	4.6
Sarah Noffke	4.6	Paralyzed	2016	4.6
T. Kingfisher	4.6	Summer in Orcus	2017	4.6
T.M. Frazier	4.6	Preppy: The Life and Death of Samuel Clearwater, Part One	2016	4.6

Identify some other authors with high weighted means of average ratings among their works, and then showcase a significant work by each of them.

```sql
SELECT other_authors_sorted_with_ranks.author,
       other_authors_sorted_with_ranks.weighted_mean_of_avg_ratings AS
weighted_mean_of_avg_ratings_for_author,
       other_authors_sorted_with_ranks.original_title,
       other_authors_sorted_with_ranks.original_publication_year,
       other_authors_sorted_with_ranks.average_rating_of_work
FROM (
      SELECT *,
             RANK() OVER (PARTITION BY author ORDER BY
ratings_count_times_average_rating DESC) as ratings_count_times_average_rating_rank
      FROM (
             SELECT other_authors_with_high_weighted_means.author,
                    other_authors_with_high_weighted_means.weighted_mean_of_avg_ratings,
                    dbo.goodreads_works.original_title,
                    dbo.goodreads_works.original_publication_year,
                    dbo.goodreads_works.ratings_count AS ratings_count_of_work,
                    CONVERT(DECIMAL(3,2), dbo.goodreads_works.avg_rating) AS
average_rating_of_work,
                    CONVERT(DECIMAL(10, 2), (dbo.goodreads_works.ratings_count *
dbo.goodreads_works.avg_rating)) AS ratings_count_times_average_rating
             FROM (
                   SELECT *
                   FROM (
                         SELECT *,
                                RANK() OVER (ORDER BY weighted_mean_of_avg_ratings
DESC) AS weighted_mean_rank
                         FROM (
                               SELECT TOP 10 author,
                                      CONVERT(DECIMAL(3,2),
SUM(ratings_count_times_avg_rating)/SUM(ratings_count)) AS weighted_mean_of_avg_ratings
                               FROM (
                                     SELECT work_id, author, original_title,
ratings_count, avg_rating, (ratings_count*avg_rating) AS ratings_count_times_avg_rating
                                     FROM dbo.goodreads_works
                               ) AS
authors_their_works_and_ratings_count_times_avg_rating
                               GROUP BY author
                               ORDER BY weighted_mean_of_avg_ratings DESC
                         ) AS authors_with_weighted_means
                   ) AS authors_with_weighted_means_and_ranks
                   WHERE 1 < weighted_mean_rank
             ) AS other_authors_with_high_weighted_means
             INNER JOIN dbo.goodreads_works
             ON other_authors_with_high_weighted_means.author =
dbo.goodreads_works.author
             -- ORDER BY weighted_mean_of_avg_ratings, author,
original_publication_year, original_title
      ) AS other_authors_with_ratings_count_times_average_rating_ranks
      -- ORDER BY weighted_mean_of_avg_ratings DESC, author ASC,
ratings_count_times_average_rating_rank ASC
) as other_authors_sorted_with_ranks
WHERE ratings_count_times_average_rating_rank = 1
ORDER BY weighted_mean_of_avg_ratings DESC, author ASC
;
```

author	weighted_mean_of_avg_ratings_for_author	original_title	original_publication_year	average_rating_of_work
Francine Rivers	4.54	Redeeming Love	1991	4.5
Jorge Luis Borges	4.54	Labyrinths: Selected Stories and Other Writings	1962	4.5
Patrick Rothfuss	4.54	The Name of the Wind	2007	4.6
Hiromu Arakawa	4.51	Gang noLian Jin Shu Shi 1	2002	4.5

Identify the authors with the highest total ratings count across all of their works in the dataset.

```sql
SELECT TOP 10 author, SUM(ratings_count) AS total_ratings_count_for_author
FROM dbo.goodreads_works
GROUP BY author
ORDER BY total_ratings_count_for_author DESC
;
```

author	total_ratings_count_for_author
J.K. Rowling	16814800
Suzanne Collins	9310785
Stephenie Meyer	8502945
Stephen King	7797932
Rick Riordan	5874480
J.R.R. Tolkien	5676766
Dan Brown	4960481
Cassandra Clare	4891687
John Green	4743107
William Shakespeare	4452598

Focus on the author with the highest count of ratings, and then showcase their works.

```sql
SELECT most_rated_authors_with_rank_and_work_details.author,
       most_rated_authors_with_rank_and_work_details.total_ratings_count_for_author,
       most_rated_authors_with_rank_and_work_details.original_title,
       most_rated_authors_with_rank_and_work_details.original_publication_year,
       most_rated_authors_with_rank_and_work_details.ratings_count AS
ratings_count_for_work
FROM (
       SELECT most_rated_authors_with_rank.author,
              most_rated_authors_with_rank.total_ratings_count_for_author,
              RANK() OVER (ORDER BY total_ratings_count_for_author DESC) AS
total_ratings_count_rank,
              dbo.goodreads_works.original_title,
              dbo.goodreads_works.original_publication_year,
              dbo.goodreads_works.ratings_count
       FROM (
              SELECT TOP 10 author, SUM(ratings_count) AS total_ratings_count_for_author
              FROM dbo.goodreads_works
              GROUP BY author
              ORDER BY total_ratings_count_for_author DESC
       ) AS most_rated_authors_with_rank
       INNER JOIN dbo.goodreads_works
       ON most_rated_authors_with_rank.author = goodreads_works.author
       -- ORDER BY total_ratings_count_for_author DESC, author ASC,
original_publication_year ASC
) AS most_rated_authors_with_rank_and_work_details
WHERE total_ratings_count_rank = 1
ORDER BY total_ratings_count_for_author DESC, author ASC, original_publication_year ASC
;
```

author	total_ratings_count_for_author	original_title	original_publication_year	ratings_count_for_work
J.K. Rowling	16814800	Harry Potter and the Philosopher's Stone	1997	4972886
J.K. Rowling	16814800	Harry Potter and the Chamber of Secrets	1998	1955144
J.K. Rowling	16814800	Harry Potter and the Prisoner of Azkaban	1999	2019176
J.K. Rowling	16814800	Harry Potter and the Goblet of Fire	2000	1912948
J.K. Rowling	16814800	Harry Potter and the Order of the Phoenix	2003	1875594
J.K. Rowling	16814800	Harry Potter and the Half-Blood Prince	2005	1824878
J.K. Rowling	16814800	Harry Potter and the Deathly Hallows	2007	1889600
J.K. Rowling	16814800	The Casual Vacancy	2012	258101
J.K. Rowling	16814800	Fantastic Beasts and Where to Find Them: The Original Screenplay	2016	46686
J.K. Rowling	16814800	Short Stories from Hogwarts of Power, Politics and Pesky Poltergeists	2016	16921
J.K. Rowling	16814800	Hogwarts: An Incomplete and Unreliable Guide	2016	16887
J.K. Rowling	16814800	Short Stories from Hogwarts of Heroism, Hardship and Dangerous Hobbies	2016	25979

Examine other authors with high rating counts, and then examine each one's most rated work.

```sql
SELECT other_highly_rated_authors_with_work_details_and_ratings_count_rank.author,
       other_highly_rated_authors_with_work_details_and_ratings_count_rank.total_ratings_count_for_author,
       other_highly_rated_authors_with_work_details_and_ratings_count_rank.original_title,
       other_highly_rated_authors_with_work_details_and_ratings_count_rank.original_publication_year,
       other_highly_rated_authors_with_work_details_and_ratings_count_rank.ratings_count_for_work
FROM (
       SELECT *,
               RANK() OVER (PARTITION BY author ORDER BY ratings_count_for_work DESC) AS ratings_count_for_work_rank
       FROM (
               SELECT most_rated_authors_with_rank_and_work_details.author,
       most_rated_authors_with_rank_and_work_details.total_ratings_count_for_author,
                       most_rated_authors_with_rank_and_work_details.original_title,
       most_rated_authors_with_rank_and_work_details.original_publication_year,
                       most_rated_authors_with_rank_and_work_details.ratings_count AS ratings_count_for_work
               FROM (
                       SELECT most_rated_authors_with_rank.author,
                               most_rated_authors_with_rank.total_ratings_count_for_author,
                               RANK() OVER (ORDER BY total_ratings_count_for_author DESC) AS total_ratings_count_rank,
                               dbo.goodreads_works.original_title,
                               dbo.goodreads_works.original_publication_year,
                               dbo.goodreads_works.ratings_count
                       FROM (
                               SELECT TOP 10 author, SUM(ratings_count) AS total_ratings_count_for_author
                               FROM dbo.goodreads_works
                               GROUP BY author
                               ORDER BY total_ratings_count_for_author DESC
                       ) AS most_rated_authors_with_rank
                       INNER JOIN dbo.goodreads_works
                       ON most_rated_authors_with_rank.author = goodreads_works.author
                       -- ORDER BY total_ratings_count_for_author DESC, author ASC, original_publication_year ASC
               ) AS most_rated_authors_with_rank_and_work_details
               WHERE 1 < total_ratings_count_rank
```

```
            -- ORDER BY total_ratings_count_for_author DESC, author ASC,
original_publication_year ASC
        ) AS other_highly_rated_authors_with_work_details
      -- ORDER BY total_ratings_count_for_author DESC, author ASC,
ratings_count_for_work DESC
) AS other_highly_rated_authors_with_work_details_and_ratings_count_rank
WHERE ratings_count_for_work_rank = 1
ORDER BY total_ratings_count_for_author DESC, author ASC
;
```

author	total_ratings_count_for_author	original_title	original_publication_year	ratings_count_for_work
Suzanne Collins	9310785	The Hunger Games	2008	5066596
Stephenie Meyer	8502945	Twilight	2005	3992661
Stephen King	7797932	The Shining	1977	845784
Rick Riordan	5874480	The Lightning Thief	2005	1458186
J.R.R. Tolkien	5676766	The Hobbit : or There and Back Again	1937	2228361
Dan Brown	4960481	Angels & Demons	2000	2126047
Cassandra Clare	4891687	City of Bones	2007	1271375
John Green	4743107	The Fault in Our Stars	2012	2564656
William Shakespeare	4452598	An Excellent conceited Tragedie of Romeo and Juliet	1597	1702565

Comments Regarding Listed Works

After generating the lists for top works, it seemed worthwhile to generate lists of works similar to those top works as well. However, because the lists of similar works for the top works would've been very long, it made more sense to limit those lists to a more manageable length. One factor for filtering similar works would be whether or not a work is at the top of a list, with a top work having more similar works showcased than other works in the same list. Another factor for filtering similar works would be whether a list is organized by average rating or ratings count, with similar works being based off of average rating if the list of top works is based off of average rating, and with similar works being based off of total ratings count if the list of top works is based off of total ratings count.

With respect to top works by either average rating or total rating count, all the similar works for that one top work would be listed. Therefore, *The Hunger Games* by Suzanne Collins, which has the highest total ratings count among all works in this dataset, had all similar works listed. Now, if there's a tie between top works by either average rating or total rating count, like in the case of Brandon Sanderson with three works (*The Way of Kings, Part 2*, *Words of Radiance*, and *Words of Radiance, Part 2*) all having the same highest average rating, then each of those top works would have their similar works listed if available.

Other top works that are not the number one could then have just one similar work listed based on whether the initial work had a high average rating or a high rating count. For example, *Harry Potter and the Philosopher's Stone* by J.K. Rowling has a high total rating count, but not high enough to be the number one, so that work has only one similar work listed, *The Mark of Athena* by Rick Riordan, and that similar work has the highest rating account among all works similar to *Harry Potter and the Philosopher's Stone*. Now, if there's a tie between similar works, then all of those similar works would be listed. Take for example *Oathbringer* by Brandon

Sanderson. That work has a high average rating, but not high enough to be number one, so it should have just one similar work listed, and that similar work would have the highest average rating among all works similar to *Oathbringer*. However, there are two works, *Sins of Empire* by Brian McClellan and *City of Miracles* by Robert Jackson Bennet, that have the highest average rating among similar works. Therefore, both of these similar works are listed with *Oathbringer*.

A similar approach had been taken with top authors. Lists of their works would've been very long had all works by all top authors been listed. Therefore, the lists were cut down to a manageable size by showcasing all the works for just the number one author of their list, and then limiting the list of works of the other top authors to just one work for each of them. That one work would then be the top work of that author based off of the list for that author. If the list was for the top authors by weighted mean of average ratings, then that one work, referred to as a "significant work," would have the "heaviest average rating," or the largest result found when multiplying the average rating for a work by the rating count for that work, among all of that author's works. (To simplify the list, only the average rating of this significant work was shown.) If the list was for the top authors by total ratings count, then that one work would have the highest rating count among all of that author's works.

Take for example Suzanne Collins. This author has a high total rating count, but not high enough to be number one. Therefore, only one work by that author, *The Hunger Games*, is showcased, and that work has the highest rating count among all of their works.

Now, if there's a tie among authors, then all of those authors were listed along with all of their works. For example, in the list of top authors by weighted mean of average ratings for their works, six authors all had the same top weighted mean - Angie Thomas, Bryan Stevenson, Haruichi Furudate, Sara Noffke, T. Kingfisher, and T.M. Frazier. Therefore, all of them were listed with all of their works. Note that each of these authors has just one of their works listed in this dataset, so their combined list of their works turned out to be conveniently manageable.

Also, it's worth noting that when listing a significant work by an author with a high weighted mean of the average ratings of all of their works, the work listed won't necessarily have the highest average rating among all the works for a particular author. For example, *Redeeming Love* by Francine Rivers has an average rating of 4.5 while the author has a weighted mean of 4.54. This is because, as mentioned before, the work listed will have the "heaviest average rating," which once again is the largest result found when multiplying the average rating of a work with the number of ratings for that work, among all of that author's works. (Also once again, to simplify the list, only the average rating of this significant work was shown.) This decision was made in order to align with how the weighted mean of the average ratings was calculated for each author, which involves the following steps:

1.) Calculate the product of the average rating of a work by an author and the rating count for that work by that author
2.) Repeat that calculation for all other works by that author if that author has more than one work listed in the dataset
3.) Add up those products
4.) Divide that sum by the sum of all the rating counts for all works by that author

In the event that the weighted mean of average ratings among works by a particular author is equal to the average rating of the work listed for that author, it's possibly because that's the only work by that author listed in the dataset. This is the case with the six authors who tied with the highest weighted mean of 4.6; they each have one work listed in the dataset, and those works each had an average score of 4.6. However, it's also possible that two or more works by the same author can have the same "heaviest average rating" if the math and rounding work out that way.

At any rate, this approach to showcasing works was an attempt to place emphasis on authors and their associates who had written, published, and promoted works that rated well among readers or garnered a lot of reviews.

Comments Regarding Author Names

Because of the way the name of each author had been recorded in the dataset, there was no easy way to parse their names in such a way that the authors could all be sorted by family name. Take for example the following authors:

1. Aaron Becker
2. Abhijit V. Banerjee
3. Alain de Botton
4. Alexander McCall Smith
5. Andre Dubus III
6. Kurt Vonnegut Jr.

In the case of Aaron Becker, the first word is the author's personal name, and the second word is their family name. In the case of Abhijit V. Banerjee, the first word is the author's personal name, the second "word" is their middle initial, and the third word is their family name. In the case of Alain de Botton, the first word is their personal name, and both the second and third words combine to form their family name. In the case of Alexander McCall Smith, the first word is one of their personal names, and both the second and third words combine to form their family name. In the case of Andre Dubus III, the first word is the author's personal name, the second word is their family name, and the third "word" is a generational suffix. In the case of

Kurt Vonnegut Jr., the first word is the author's personal name, the second word is their family name, and the third "word" is another generational suffix.

Simply writing a script that takes the last "word" in an author's recorded name as their last name runs the risk of either missing a part of the family name (like in the case of de Botton and McCall Smith) or outright missing the family name altogether (like in the case of Dubus and Vonnegut). Therefore, authors had been sorted by the first letter of the name recorded in the dataset for the sake of simplicity.

Comments Regarding Associated Genres

Attempts have been made to identify top books for each genre by either average rating or total ratings counts. However, some genres assigned to some works seemed dubious.

Take for example the work *The Hunger Games* by Suzanne Collins:

```sql
SELECT author, original_title, genres
FROM dbo.goodreads_works
WHERE original_title = 'The Hunger Games'
;
```

author	original_title	genres
Suzanne Collins	The Hunger Games	young-adult, fiction, fantasy, paranormal, romance, mystery, thriller, crime

According to the *Wikipedia* entry for that work (https://en.wikipedia.org/wiki/The_Hunger_Games_(novel)), *The Hungers Games* is an adventure and science-fiction young-adult novel, so the "young-adult," "fiction," and "thriller" genres make sense. According to the "Plot" section of that *Wikipedia* entry, there is some romantic interest between two of the characters; however, that relationship doesn't appear to be the main focus of the novel, so the "romance" genre doesn't feel like a good fit for this work. Moreover, there doesn't appear to be any hint of fantasy, the paranormal, mystery, or crime in the plot of the work, so those genres don't feel like a good fit for this work as well.

Next, take into consideration the works by George R.R. Martin:

```sql
-- Investigate the works by George R.R. Martin and their associated genres.
SELECT author, original_title, genres
FROM dbo.goodreads_works
WHERE author = 'George R.R. Martin'
;
```

author	original_title	genres
George R.R. Martin	A Feast for Crows	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	Songs of Love and Death: All-Original Tales of Star-Crossed Love	fantasy, paranormal, romance, fiction, history, historical fiction, biography, mystery, thriller, crime
George R.R. Martin	A Storm of Swords	fiction, fantasy, paranormal, history, historical fiction, biography
George R.R. Martin	The Armageddon Rag	fantasy, paranormal, fiction, mystery, thriller, crime, history, historical fiction, biography
George R.R. Martin	A Game of Thrones	fantasy, paranormal, history, historical fiction, biography, romance, fiction, mystery, thriller, crime
George R.R. Martin	Down These Strange Streets	fantasy, paranormal, fiction, history, historical fiction, biography, romance, mystery, thriller, crime
George R.R. Martin	The Mystery Knight	fantasy, paranormal, fiction, comics, graphic, mystery, thriller, crime
George R.R. Martin	A Storm of Swords: Part 1 Steel and Snow	fantasy, paranormal, fiction, history, historical fiction, biography, romance, mystery, thriller, crime
George R.R. Martin	Sandkings	fiction, fantasy, paranormal, mystery, thriller, crime, comics, graphic
George R.R. Martin	A Dance With Dragons: Part 2 After The Feast	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	Dangerous Women	history, historical fiction, biography, fantasy, paranormal, mystery, thriller, crime, romance, fiction
George R.R. Martin	A Dance With Dragons: Part 1 Dreams and Dust	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	The World of Ice and Fire	fantasy, paranormal, fiction, history, historical fiction, biography, non-fiction
George R.R. Martin	A Knight of the Seven Kingdoms	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	Fevre Dream	fantasy, paranormal, fiction, history, historical fiction, biography, mystery, thriller, crime, romance
George R.R. Martin	Rogues	fiction, fantasy, paranormal, mystery, thriller, crime, history, historical fiction, biography
George R.R. Martin	A Dance with Dragons	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	A Clash of Kings	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	The Ice Dragon	fantasy, paranormal, children, young-adult, fiction, comics, graphic
George R.R. Martin	The Hedge Knight	fantasy, paranormal, fiction, comics, graphic
George R.R. Martin	A Dream of Spring	fantasy, paranormal, fiction, history, historical fiction, biography
George R.R. Martin	A Storm of Swords: Part 2 Blood and Gold	fantasy, paranormal, fiction, history, historical fiction, biography, romance
George R.R. Martin	Warriors	fantasy, paranormal, fiction, history, historical fiction, biography, mystery, thriller, crime

According to the *Wikipedia* article for this author (https://en.wikipedia.org/wiki/George_R._R._Martin), he focuses on fantasy, horror, and science fiction. Many of his works in this dataset have the "history" or "biography" genres associated with them. It is my understanding that many of his works had been inspired by medieval Europe, including the War of the Roses in Britain (source = https://medium.com/hooked-on-books/historical-inspirations-behind-george-rr-martins-series-a8756fbd88e1); however, his works are fiction while history is non-fiction, so the "history" genre doesn't feel like a good fit for his works. Moreover, while it is possible to use fiction to portray a person's life, Martin mixes and matches different people in history to create his characters, at least for the series *A Song of Ice and Fire* which includes many of his works listed above (source = https://rozk.livejournal.com/93995.html#cutid1). Therefore, the "biography" genre doesn't feel like a good fit for some of his works.

Finally, take into consideration the work of Anne Frank:

```sql
-- Look into the work by Anne Frank.
SELECT author, original_title, genres
FROM dbo.goodreads_works
WHERE author = 'Anne Frank'
;
```

author	original_title	genres
Anne Frank	Het Achterhuis: Dagboekbrieven 14 juni 1942 - 1 augustus 1944	non-fiction, history, historical fiction, biography, young-adult, fiction, comics, graphic, children

Her work is listed in this dataset under its original Dutch title, but the work is commonly referred to as *The Diary of Anne Frank*. (Source = https://en.wikipedia.org/wiki/The_Diary_of_a_Young_Girl). Now, Anne Frank did exist, and her work is a first-hand account of her experiences in Nazi Germany. Therefore, the "fiction" and "historical fiction" genres don't feel appropriate. There had been a graphic novelization of her work; however, this entry in the dataset is for the original work by Anne Frank, which to my

knowlege has only text, so the "comics" and "graphic novel" genres don't make sense here. Moreover, considering the content of Anne Frank's work, it might not be appropriate for young children, so the "children" genre feels questionable.

The idea to "clean" the genre tags for these works had been considered, but doing so for every work listed in the dataset would be very time-consuming. Without having read each and every work listed, there would also be no guarantee that the data-cleaning process would result in accurate genres for each work. Moreover, even limiting the cleaning to just the works that show up in the outputs of queries would still be both time-consuming and potentially inaccurate. Therefore, a list of top works by average rating or total ratings count for each genre had to be foregone, at least with this dataset.