# Create Table: mybooks

CREATE TABLE mybooks

(

author VARCHAR (60),

title TEXT PRIMARY KEY,

score TEXT,

status TEXT

);

Comments:

* I filtered out all the blank spaces my file had on what would become the primary key
* I took out all the links that appeared
* I took out the headers on the excel/csv file ( not necessary)
* I took out the link column because I had not added it to my table
* I saved all these changes as a CSV file

# Create Table: tbr

CREATE TABLE tbr

(

author VARCHAR (60),

title TEXT PRIMARY KEY,

series VARCHAR (10),

s\_name TEXT

);

# Questions

1. How many books have a score of 5/4/3/2/1 stars? (WHERE)

* 5 stars:

A: 91 books

SELECT \*

FROM mybooks

WHERE score = '⭐️⭐️⭐️⭐️⭐️'

* 4 stars

A: 61

SELECT \*

FROM mybooks

WHERE score = '⭐️⭐️⭐️⭐️'

* 3 stars

A: 54

SELECT \*

FROM mybooks

WHERE score = '⭐️⭐️⭐️'

* 2 stars

A: 10

SELECT \*

FROM mybooks

WHERE score = '⭐️⭐️'

* 1 star

A: 4

SELECT \*

FROM mybooks

WHERE score = '⭐️'

1. How many books are currently being read? (WHERE)

SELECT \*

FROM mybooks

WHERE status = 'Reading'

1. Create a list of 10 authors and the number of books they have with a 5-star rating. Order the list in descending order (COUNT/LIMIT/ORDER BY/WHERE)

SELECT author, COUNT(author),score

FROM mybooks AS mb

WHERE score = '⭐️⭐️⭐️⭐️⭐️'

GROUP BY author, score

ORDER BY author DESC

LIMIT 10

1. Create a list of authors that have more than one 5 star book written, then 4 stars, then 4-5 stars. Group by author (WHERE/GROUP BY/ORDER BY)

* 5 stars

A: 14 books

SELECT author, COUNT(title) AS num\_read

FROM mybooks AS mb

WHERE score='⭐️⭐️⭐️⭐️⭐️'

GROUP BY author HAVING COUNT(title) > 1

ORDER BY num\_read DESC

* 4 stars

A: 9 books

SELECT author, COUNT(title) AS num\_read

FROM mybooks AS mb

WHERE score='⭐️⭐️⭐️⭐️'

GROUP BY author HAVING COUNT(title) > 1

ORDER BY num\_read DESC

* 4-5 stars, should give me 23 entries (correct ?)

SELECT author, COUNT(title) AS num\_read

FROM mybooks AS mb

WHERE score='⭐️⭐️⭐️⭐️' OR score='⭐️⭐️⭐️⭐️⭐️'

GROUP BY author HAVING COUNT(title) > 1

ORDER BY num\_read DESC

1. List all the book titles that start with “The” (WHERE/LIKE/ COUNT)

A: 65 books have a title that start with “ The “

SELECT author, title, COUNT(title)

FROM mybooks AS mb

WHERE title LIKE 'The%'

GROUP BY author, title

1. List all the book titles that have “the” somewhere in the title (WHERE/LIKE/ COUNT)

A: there are 44 entries

SELECT author, title, COUNT(title)

FROM mybooks AS mb

WHERE title LIKE '%the%'

GROUP BY author, title

1. How many authors have you read from? (COUNT)

A: 133

SELECT author, count(title)

FROM mybooks

GROUP BY author

ORDER BY COUNT(title) DESC

1. how many authors have you read 5 or more books of? (HAVING)

A: 10 authors

SELECT author, count(title)

FROM mybooks

GROUP BY author

HAVING COUNT(title) >= 5

ORDER BY COUNT(title) DESC

1. Add the book **Almendra** by **Won-Pyung Sohn** to the TBR table (INSERT INTO)

INSERT INTO tbr(author,title,series,s\_name)

VALUES

('Won Pyung Sohn', 'Almendra', 'NO', 'N/A');

SELECT \*

FROM tbr

WHERE title='Almendra'

1. how many series have been read? (WHERE/GROUP BY)

A: 53

SELECT author, s\_name

FROM tbr

WHERE s\_name != 'N/A'

GROUP BY tbr.author, s\_name

1. Create a new column that converts the score into numerical values.

SELECT \*,

CASE

WHEN(score='⭐️') THEN '1'

WHEN(score='⭐️⭐️') THEN '2'

WHEN(score='⭐️⭐️⭐️') THEN '3'

WHEN(score='⭐️⭐️⭐️⭐️') THEN '4'

WHEN(score='⭐️⭐️⭐️⭐️⭐️') THEN '5'

ELSE '0'

END AS num\_score

FROM mybooks

1. Using your solution in question 11, find the Average score of books

SELECT

COALESCE (

AVG(

CAST(

CASE

WHEN(score='⭐️') THEN '1'

WHEN(score='⭐️⭐️') THEN '2'

WHEN(score='⭐️⭐️⭐️') THEN '3'

WHEN(score='⭐️⭐️⭐️⭐️') THEN '4'

WHEN(score='⭐️⭐️⭐️⭐️⭐️') THEN '5'

ELSE '0'

END

as NUMERIC)

)

) AS avg\_scores

FROM mybooks AS m

JOINED TABLES

1. Do you have any books on **tbr** that have the same authors on **mybooks** table?? (INNER JOIN)

A: Yes, 2 authors with one book each

SELECT mb.author, t.title

FROM mybooks AS mb

INNER JOIN tbr AS t

USING (title)

1. Combine the entries of both mybooks table and tbr table (UNION)

mybooks has: 240 entries

mybooks has: 153 entries

Total entries: 391

SELECT author, title

FROM mybooks AS mb

UNION

SELECT author, title

FROM tbr AS t

1. Select **only** entries of the tbr table that **do not** have an author in common with the table “mybooks” (LEFT/RIGHT join)

A: 114

SELECT author

FROM tbr AS t

LEFT OUTER JOIN mybooks AS m

USING (author)

GROUP BY author

ORDER BY author DESC

1. Select **only** entries of the tbr table that **have** an author in common with the table “mybooks” (LEFT/RIGHT join)

A: 21 entries

SELECT t.author

FROM tbr AS t

LEFT OUTER JOIN mybooks AS m

USING (author)

WHERE t.author = m.author

GROUP BY author

ORDER BY author DESC