

Selection Refining & Aggregate Function - SQL Session 6

06 July 2024 14:31

Selection Refining & Aggregate Function - SQL Session 6

QUICK CHALLENGE :

Select All Story Collections

Titles That contain 'stories'

```
+-----+  
| title |  
+-----+  
| What We Talk About When We Talk About Love: Stories |  
| Where I'm Calling From: Selected Stories |  
| Oblivion: Stories |  
+-----+
```

```
mysql> SELECT title FROM Books where title like "%stories%";  
+-----+  
| title |  
+-----+  
| What We Talk About When We Talk About Love: Stories |  
| Where I'm Calling From: Selected Stories |  
| Oblivion: Stories |  
+-----+  
3 rows in set (0.00 sec)
```

Find The Longest Book

Print Out the Title and Page Count

```
+-----+-----+  
| title | pages |  
+-----+-----+  
| The Amazing Adventures of Kavalier & Clay | 634 |  
+-----+-----+
```

```
mysql> SELECT title , pages FROM books ORDER BY pages DESC LIMIT 1;  
+-----+-----+  
| title | pages |  
+-----+-----+  
| The Amazing Adventures of Kavalier & Clay | 634 |  
+-----+-----+  
1 row in set (0.00 sec)
```

Print a summary containing the title and year, for the 3 most recent books

```
+-----+  
| summary |  
+-----+  
| Lincoln In The Bardo - 2017 |  
| Norse Mythology - 2016 |  
| 10% Happier - 2014 |  
+-----+
```

```
mysql> SELECT concat(title , " - " , released_year) AS Summary
```

```

mysql> SELECT concat(title , " - " , released_year) AS Summary
-> FROM Books ORDER BY released_year DESC Limit 3;
+-----+
| Summary |
+-----+
| Lincoln In The Bardo - 2017 |
| Norse Mythology - 2016 |
| 10% Happier - 2014 |
+-----+
3 rows in set (0.01 sec)

```

Find all books with an author_lname
that contains a space(" ")

title	author_lname
Oblivion: Stories	Foster Wallace
Consider the Lobster	Foster Wallace

```

mysql> SELECT title , author_lname FROM Books where author_lname LIKE "% %";
+-----+-----+
| title | author_lname |
+-----+-----+
| Oblivion: Stories | Foster Wallace |
| Consider the Lobster | Foster Wallace |
+-----+-----+
2 rows in set (0.00 sec)

```

Find The 3 Books With The
Lowest Stock

Select title, year, and stock

title	released_year	stock_quantity
American Gods	2001	12
Where I'm Calling From: Selected Stories	1989	12
What We Talk About When We Talk About Love: Stories	1981	23

```

mysql> SELECT
->   title,
->   released_year,
->   stock_quantity
->   from Books
->   ORDER BY 3,1
->   LIMIT 3;
+-----+-----+-----+
| title | released_year | stock_quantity |
+-----+-----+-----+
| American Gods | 2001 | 12 |
| Where I'm Calling From: Selected Stories | 1989 | 12 |
| What We Talk About When We Talk About Love: Stories | 1981 | 23 |
+-----+-----+-----+
3 rows in set (0.01 sec)

```

Print title and author_lname, sorted first
by author_lname and then by title

title	author_lname
What We Talk About When We Talk About Love: Stories	Carver
Where I'm Calling From: Selected Stories	Carver
The Amazing Adventures of Kavalier & Clay	Chabon
White Noise	DeLillo
A Heartbreaking Work of Staggering Genius	Eggers
A Hologram for the King: A Novel	Eggers
The Circle	Eggers
Consider the Lobster	Foster Wallace
Oblivion: Stories	Foster Wallace
American Gods	Gaiman
Coraline	Gaiman
Norse Mythology	Gaiman
10% Happier	Harris

Oblivion: Stories	Foster Wallace
American Gods	Gaiman
Coraline	Gaiman
Norse Mythology	Gaiman
10% Happier	Harris
fake_book	Harris
Interpreter of Maladies	Lahiri
The Namesake	Lahiri
Lincoln In The Bardo	Saunders
Just Kids	Smith
Cannery Row	Steinbeck

```
mysql> SELECT title , author_lname from books
-> order by 2,1;
+-----+-----+
| title | author_lname |
+-----+-----+
| What We Talk About When We Talk About Love: Stories | Carver
| Where I'm Calling From: Selected Stories | Carver
| The Amazing Adventures of Kavalier & Clay | Chabon
| White Noise | DeLillo
| A Heartbreaking Work of Staggering Genius | Eggers
| A Hologram for the King: A Novel | Eggers
| The Circle | Eggers
| Consider the Lobster | Foster Wallace
| Oblivion: Stories | Foster Wallace
| American Gods | Gaiman
| Coraline | Gaiman
| Norse Mythology | Gaiman
| 10% Happier | Harris
| fake_book | Harris
| Interpreter of Maladies | Lahiri
| The Namesake | Lahiri
| Lincoln In The Bardo | Saunders
| Just Kids | Smith
| Cannery Row | Steinbeck
+-----+-----+
19 rows in set (0.00 sec)
```

Make This Happen...

Sorted Alphabetically By Last Name

```
+-----+
| yell
+-----+
| MY FAVORITE AUTHOR IS RAYMOND CARVER!
| MY FAVORITE AUTHOR IS RAYMOND CARVER!
| MY FAVORITE AUTHOR IS MICHAEL CHABON!
| MY FAVORITE AUTHOR IS DON DELILLO!
| MY FAVORITE AUTHOR IS DAVE EGGERS!
| MY FAVORITE AUTHOR IS DAVE EGGERS!
| MY FAVORITE AUTHOR IS DAVE EGGERS!
| MY FAVORITE AUTHOR IS DAVID FOSTER WALLACE!
| MY FAVORITE AUTHOR IS DAVID FOSTER WALLACE!
| MY FAVORITE AUTHOR IS NEIL GAIMAN!
| MY FAVORITE AUTHOR IS NEIL GAIMAN!
| MY FAVORITE AUTHOR IS NEIL GAIMAN!
| MY FAVORITE AUTHOR IS FREIDA HARRIS!
| MY FAVORITE AUTHOR IS DAN HARRIS!
| MY FAVORITE AUTHOR IS JHUMPA LAHIRI!
| MY FAVORITE AUTHOR IS JHUMPA LAHIRI!
| MY FAVORITE AUTHOR IS GEORGE SAUNDERS!
| MY FAVORITE AUTHOR IS PATTI SMITH!
| MY FAVORITE AUTHOR IS JOHN STEINBECK!
```

```
mysql> SELECT CONCAT("MY FAVORITE AUTHOR IS " , author_fname , " " , author_lname , "!") AS yell
-> FROM Books ORDER BY author_lname;
+-----+
| yell
+-----+
| MY FAVORITE AUTHOR IS Raymond Carver!
| MY FAVORITE AUTHOR IS Raymond Carver!
| MY FAVORITE AUTHOR IS Michael Chabon!
| MY FAVORITE AUTHOR IS Don DeLillo!
| MY FAVORITE AUTHOR IS Dave Eggers!
| MY FAVORITE AUTHOR IS Dave Eggers!
| MY FAVORITE AUTHOR IS Dave Eggers!
| MY FAVORITE AUTHOR IS David Foster Wallace!
| MY FAVORITE AUTHOR IS David Foster Wallace!
| MY FAVORITE AUTHOR IS Neil Gaiman!
| MY FAVORITE AUTHOR IS Neil Gaiman!
| MY FAVORITE AUTHOR IS Neil Gaiman!
| MY FAVORITE AUTHOR IS Dan Harris!
| MY FAVORITE AUTHOR IS Freida Harris!
| MY FAVORITE AUTHOR IS Jhumpa Lahiri!
| MY FAVORITE AUTHOR IS Jhumpa Lahiri!
| MY FAVORITE AUTHOR IS George Saunders!
| MY FAVORITE AUTHOR IS Patti Smith!
```

```

| MY FAVORITE AUTHOR IS Dan Harris!
| MY FAVORITE AUTHOR IS Freida Harris!
| MY FAVORITE AUTHOR IS Jhumpa Lahiri!
| MY FAVORITE AUTHOR IS Jhumpa Lahiri!
| MY FAVORITE AUTHOR IS George Saunders!
| MY FAVORITE AUTHOR IS Patti Smith!
| MY FAVORITE AUTHOR IS John Steinbeck!
+-----+
19 rows in set (0.00 sec)

```

Aggregation Function:

1. Count :

```

mysql> SELECT COUNT(*) FROM books;
+-----+
| COUNT(*) |
+-----+
|      19 |
+-----+
1 row in set (0.04 sec)

mysql> SELECT COUNT(author_lname) FROM books;
+-----+
| COUNT(author_lname) |
+-----+
|          19 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT COUNT(DISTINCT author_lname) FROM books;
+-----+
| COUNT(DISTINCT author_lname) |
+-----+
|           11 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT COUNT(DISTINCT author_fname) FROM books;
+-----+
| COUNT(DISTINCT author_fname) |
+-----+
|           12 |
+-----+
1 row in set (0.00 sec)

```

```

mysql> SELECT COUNT(*) FROM books
    -> WHERE title LIKE '%the%';
+-----+
| COUNT(*) |
+-----+
|       6 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT title FROM books
    -> WHERE title LIKE '%the%';
+-----+
| title |
+-----+
| The Namesake
| A Hologram for the King: A Novel
| The Circle
| The Amazing Adventures of Kavalier & Clay
| Consider the Lobster
| Lincoln In The Bardo
+-----+
6 rows in set (0.00 sec)

```

- GROUP BY :

```

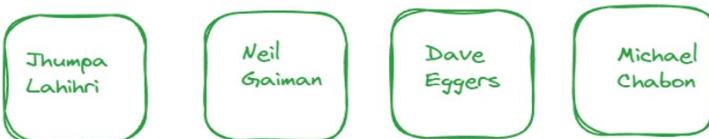
mysql> SELECT author_lname from books GROUP BY author_lname;
+-----+
| author_lname |
+-----+

```

```
mysql> SELECT author_lname from books GROUP BY author_lname;
+-----+
| author_lname |
+-----+
| Lahiri      |
| Gaiman      |
| Eggers      |
| Chabon      |
| Smith       |
| Carver      |
| DeLillo     |
| Steinbeck   |
| Foster Wallace|
| Harris      |
| Saunders    |
+-----+
11 rows in set (0.00 sec)
```

COUNT how many
books each author has
written

```
mysql> SELECT CONCAT(author_fname , " " , author_lname) as author , Count(*)  
      -> FROM books GROUP BY author;  
+-----+-----+  
| author | Count(*) |  
+-----+-----+  
| Jhumpa Lahiri | 2 |  
| Neil Gaiman | 3 |  
| Dave Eggers | 3 |  
| Michael Chabon | 1 |  
| Patti Smith | 1 |  
| Raymond Carver | 2 |  
| Don DeLillo | 1 |  
| John Steinbeck | 1 |  
| David Foster Wallace | 2 |  
| Dan Harris | 1 |  
| Freida Harris | 1 |  
| George Saunders | 1 |  
+-----+-----+  
12 rows in set (0.01 sec)
```



MIN & MAX with GROUP BY :

Find the minimum
released_year

```
mysql> SELECT MIN(released_year) FROM Books;
+-----+
| MIN(released_year) |
+-----+
|          1945 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT MAX(released_year) FROM Books;
+-----+
```

```
mysql> SELECT MAX(released_year) FROM Books;
+-----+
| MAX(released_year) |
+-----+
|          2017 |
+-----+
1 row in set (0.00 sec)
```

GROUP
THE
DATE
+ USE OF
AGGREGATE
FUNCTION
TO GET MOST
MEANINGFUL
INSIGHT.

12 , 11 , 99 , 33 , 45 , 123 , 129

12 , 14 , 9 , 133 , 145 , 23 , 29

121 , 111 , 991 , 331 , 451 , 1231 , 29

112 , 111 , 199 , 133 , 145 , 123 , 9

SUB-QUERIES

```
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM books
-> WHERE pages = (SELECT Min(pages)
->                   FROM books);
+-----+-----+-----+-----+-----+-----+
| book_id | title           | author_fname | author_lname | released_year | stock_quantity | pages |
+-----+-----+-----+-----+-----+-----+
|    11 | What We Talk About When We Talk About Love: Stories | Raymond     | Carver       |        1981 |          23 |    176 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql>
mysql> SELECT MIN(PAGES) FROM Books;
+-----+
| MIN(PAGES) |
+-----+
|    176 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM BOOKS WHERE PAGES = 176;
+-----+-----+-----+-----+-----+-----+
| book_id | title           | author_fname | author_lname | released_year | stock_quantity | pages |
+-----+-----+-----+-----+-----+-----+
|    11 | What We Talk About When We Talk About Love: Stories | Raymond     | Carver       |        1981 |          23 |    176 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

```
> SELECT * FROM books
> WHERE pages = (SELECT Min(pages))
```

```
> SELECT * FROM books  
> WHERE pages = (SELECT Min(pages)  
                  FROM books);
```

```
mysql>  
mysql> SELECT MIN(PAGES) FROM Books;  
+-----+  
| MIN(PAGES) |  
+-----+  
|      176   |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT * FROM BOOKS WHERE PAGES = 176;
```

QUICK CHALLENGE :

Find the year each author published their first book → GROUP BY

```
mysql> SELECT CONCAT(author_fname , " " , author_lname) as author , MIN(released_year) as first_book  
      -> FROM BOOKS GROUP BY author;  
+-----+-----+  
| author          | first_book |  
+-----+-----+  
| Jhumpa Lahiri  |    1996 |  
| Neil Gaiman     |    2001 |  
| Dave Eggers     |    2001 |  
| Michael Chabon |    2000 |  
| Patti Smith     |    2010 |  
| Raymond Carver  |    1981 |  
| Don DeLillo     |    1985 |  
| John Steinbeck  |    1945 |  
| David Foster Wallace | 2004 |  
| Dan Harris      |    2014 |  
| Freida Harris   |    2001 |  
| George Saunders |    2017 |  
+-----+-----+  
12 rows in set (0.00 sec)
```

Find the longest page count for each author

```
mysql> SELECT CONCAT(author_fname , " " , author_lname) as author , MAX(pages) as 'longest page'  
      -> FROM books GROUP BY author;  
+-----+-----+  
| author          | longest page |  
+-----+-----+  
| Jhumpa Lahiri  |      291 |  
| Neil Gaiman     |      465 |  
| Dave Eggers     |      504 |  
| Michael Chabon |      634 |  
| Patti Smith     |      304 |  
| Raymond Carver  |      526 |  
| Don DeLillo     |      320 |  
| John Steinbeck  |      181 |  
| David Foster Wallace | 343 |
```

```

| Patti Smith          |      304 |
| Raymond Carver     |      526 |
| Don DeLillo          |      320 |
| John Steinbeck      |      181 |
| David Foster Wallace |      343 |
| Dan Harris           |      256 |
| Freida Harris        |      428 |
| George Saunders      |      367 |
+-----+
12 rows in set (0.00 sec)

```

Sum all pages in the entire database

```

mysql> SELECT SUM(pages) FROM Books;
+-----+
| SUM(pages) |
+-----+
|      6623 |
+-----+
1 row in set (0.00 sec)

```

Sum all pages each author has written

```

mysql> SELECT CONCAT(author_fname , " " , author_lname) as author , SUM(pages) as page_count FROM BOOKS
-> GROUP BY author;
+-----+-----+
| author          | page_count |
+-----+-----+
| Jhumpa Lahiri   |      489 |
| Neil Gaiman     |      977 |
| Dave Eggers     |     1293 |
| Michael Chabon  |      634 |
| Patti Smith      |      304 |
| Raymond Carver  |      782 |
| Don DeLillo      |      320 |
| John Steinbeck  |      181 |
| David Foster Wallace | 672 |
| Dan Harris       |      256 |
| Freida Harris    |      428 |
| George Saunders  |      367 |
+-----+
12 rows in set (0.00 sec)

mysql> SELECT CONCAT(author_fname , " " , author_lname) as author , SUM(pages) as page_count , COUNT(*) FROM BOOKS
-> GROUP BY author;
+-----+-----+-----+
| author          | page_count | COUNT(*) |
+-----+-----+-----+
| Jhumpa Lahiri   |      489 |      2 |
| Neil Gaiman     |      977 |      3 |
| Dave Eggers     |     1293 |      3 |
| Michael Chabon  |      634 |      1 |
| Patti Smith      |      304 |      1 |
| Raymond Carver  |      782 |      2 |
| Don DeLillo      |      320 |      1 |
| John Steinbeck  |      181 |      1 |
| David Foster Wallace | 672 |      2 |
| Dan Harris       |      256 |      1 |
| Freida Harris    |      428 |      1 |
| George Saunders  |      367 |      1 |
+-----+

```

Calculate the average released_year across all books

```

mysql> SELECT AVG(released_year) from books;
+-----+
| AVG(released_year) |
+-----+
|      1999.7895 |
+-----+
1 row in set (0.01 sec)

```

```
+-----+  
| 1999.7895 |  
+-----+  
1 row in set (0.01 sec)
```

Calculate the average stock quantity for books released in the same year

```
mysql> SELECT released_year , ROUND(avg(stock_quantity),1) FROM Books GROUP BY released_year;  
+-----+-----+  
| released_year | ROUND(avg(stock_quantity),1) |  
+-----+-----+  
| 2003 | 66.0 |  
| 2016 | 43.0 |  
| 2001 | 134.3 |  
| 1996 | 97.0 |  
| 2012 | 154.0 |  
| 2013 | 26.0 |  
| 2000 | 68.0 |  
| 2010 | 55.0 |  
| 1981 | 23.0 |  
| 1989 | 12.0 |  
| 1985 | 49.0 |  
| 1945 | 95.0 |  
| 2004 | 172.0 |  
| 2005 | 92.0 |  
| 2014 | 29.0 |  
| 2017 | 1000.0 |  
+-----+  
16 rows in set (0.01 sec)
```

Make This Happen

year	# books	avg pages	COPY
1945	1	181.0000	
1981	1	176.0000	
1985	1	320.0000	
1989	1	526.0000	
1996	1	198.0000	
2000	1	634.0000	
2001	3	443.3333	
2003	2	249.5000	
2004	1	329.0000	
2005	1	343.0000	
2010	1	304.0000	
2012	1	352.0000	
2013	1	504.0000	
2014	1	256.0000	
2016	1	304.0000	
2017	1	367.0000	