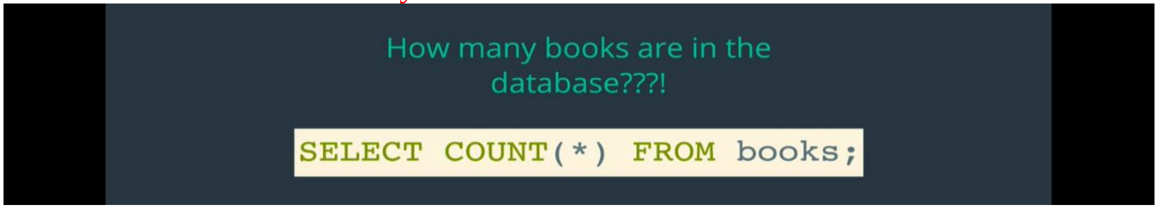
Lesson 9- the magic of aggregate function

1- The count functions
2- The joys of group by
3- Min and max basics
4- Subqueries: min and max problem
5- Using min and max with group by
6- The sum functions
7- The average function
8- Aggregate function challenges
9- Aggregate function challenges solution

Count

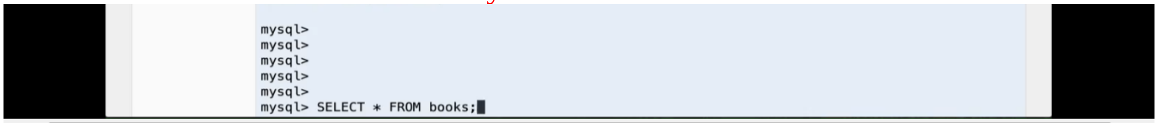
Automated count then manually



How many books are in the database???!

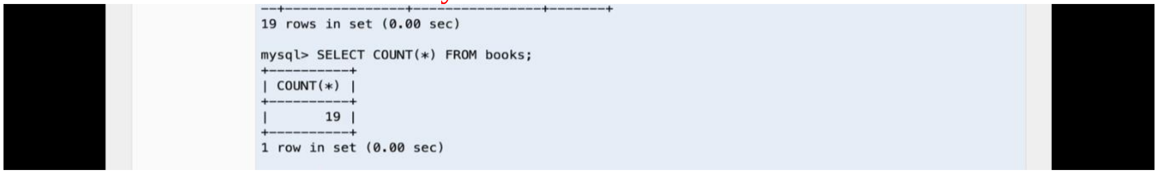
```
SELECT COUNT(*) FROM books;
```

This command need to count manually



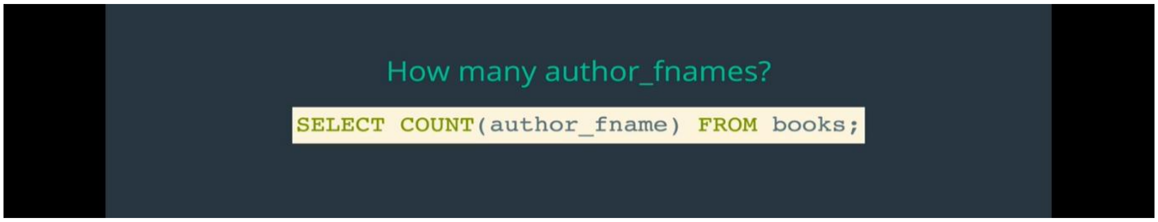
```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT * FROM books;
```

This command count automatically



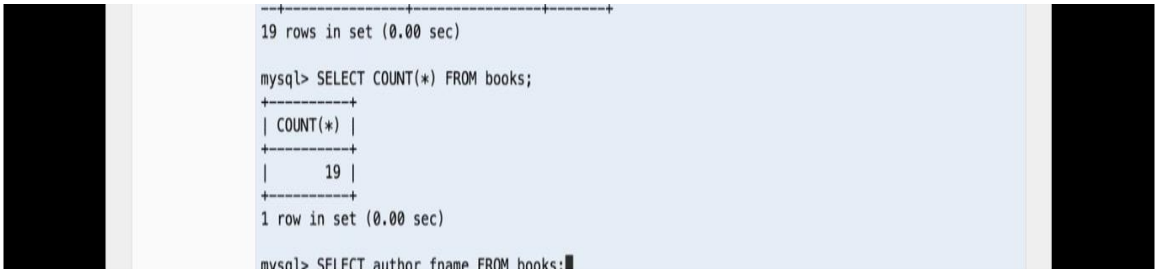
```
--+----------------+----------------+--------+
19 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM books;
+----------+
| COUNT(*) |
+----------+
|       19 |
+----------+
1 row in set (0.00 sec)
```

How many author_fnames?

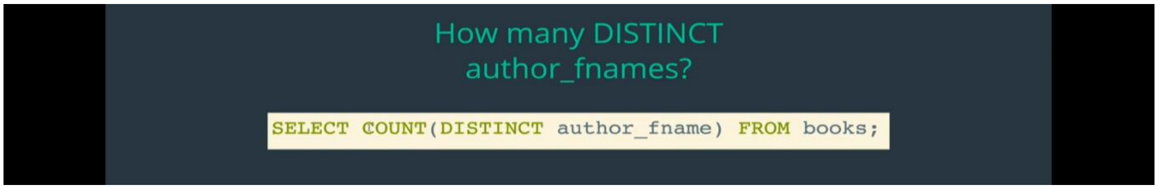


How many author_fnames?

```
SELECT COUNT(author_fname) FROM books;
```



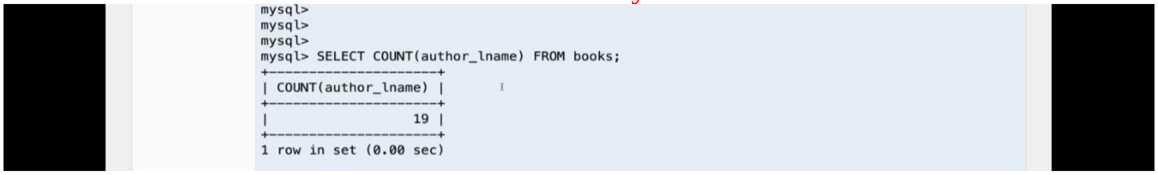
```
--+----------------+----------------+--------+
19 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM books;
+----------+
| COUNT(*) |
+----------+
|       19 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT author_fname FROM books;
```



How many DISTINCT author_fnames?

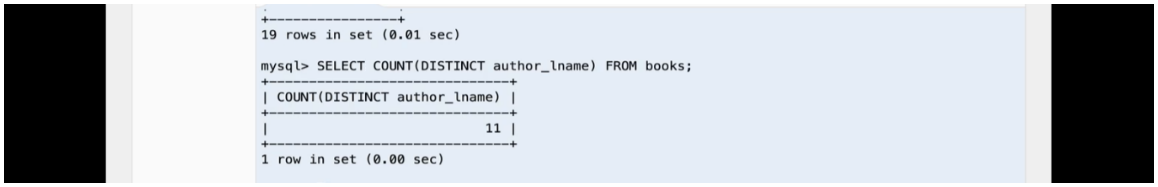
```
SELECT COUNT(DISTINCT author_fname) FROM books;
```

This is not correct amount because there are many last name



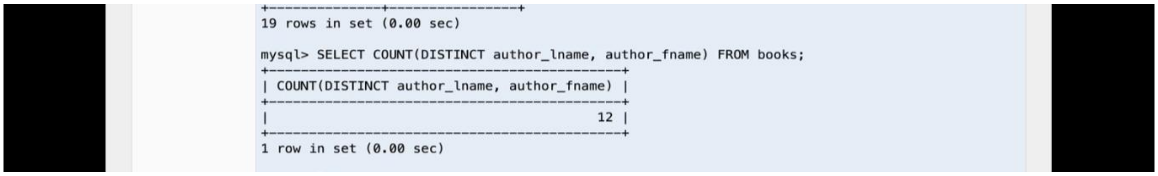
```
mysql>
mysql>
mysql>
mysql> SELECT COUNT(author_lname) FROM books;
+---------------------+
| COUNT(author_lname) |
+---------------------+
|                  19 |
+---------------------+
1 row in set (0.00 sec)
```

This not correct because it will reduce different author but same lastname



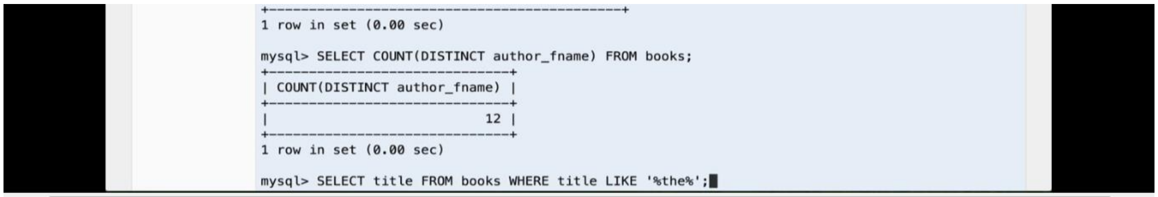
```
+----------------+
19 rows in set (0.01 sec)

mysql> SELECT COUNT(DISTINCT author_lname) FROM books;
+------------------------------+
| COUNT(DISTINCT author_lname) |
+------------------------------+
|                           11 |
+------------------------------+
1 row in set (0.00 sec)
```

This is correct



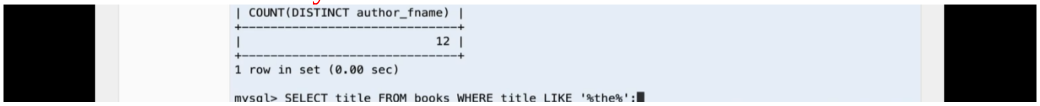
```
+----------------+----------------+
19 rows in set (0.00 sec)

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

This is not correct because different author could have same first name



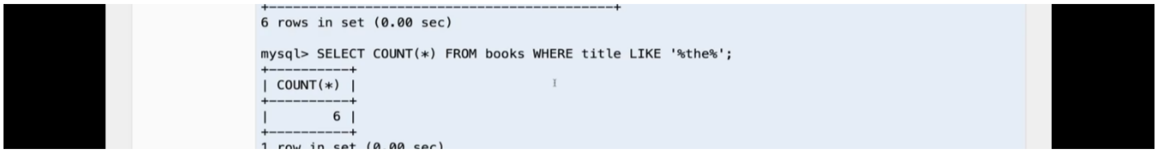
```
+--------------------------------------------+
1 row in set (0.00 sec)

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

mysql> SELECT title FROM books WHERE title LIKE '%the%';
```

Select title from books
With this count manually



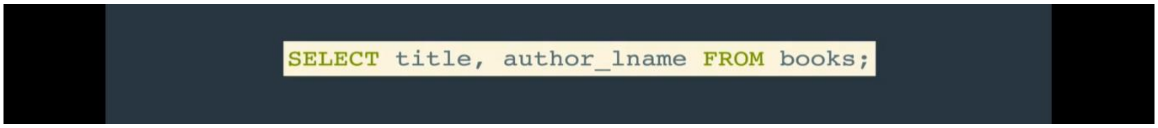
```
| COUNT(DISTINCT author_fname) |
+------------------------------+
|                           12 |
+------------------------------+
1 row in set (0.00 sec)

mysql> SELECT title FROM books WHERE title LIKE '%the%';
```

This is automated

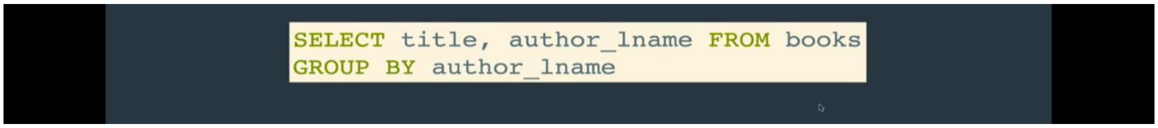


```
+--------------------------------------------+
6 rows in set (0.00 sec)

mysql> SELECT COUNT(*) FROM books WHERE title LIKE '%the%';
+----------+
| COUNT(*) |
+----------+
|        6 |
+----------+
1 row in set (0.00 sec)
```

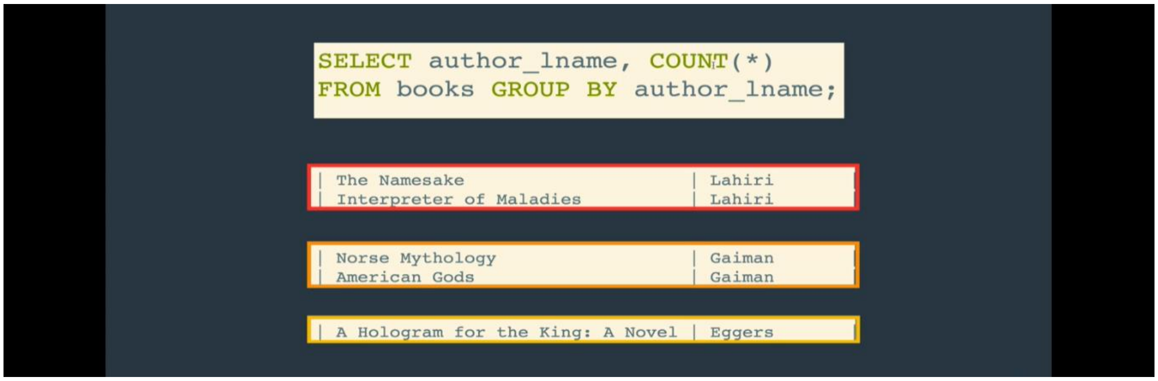
Group by command



```
SELECT title, author_lname FROM books;
```



```
SELECT title, author_lname FROM books
GROUP BY author_lname
```

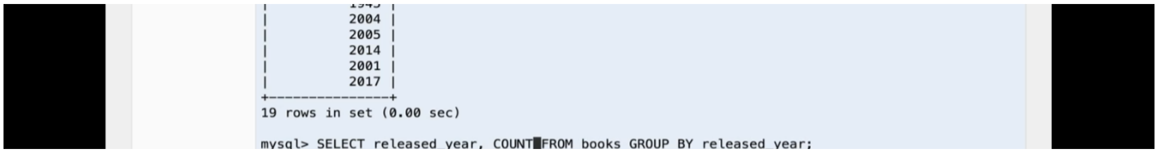


```
SELECT author_lname, COUNT(*)
FROM books GROUP BY author_lname;
```

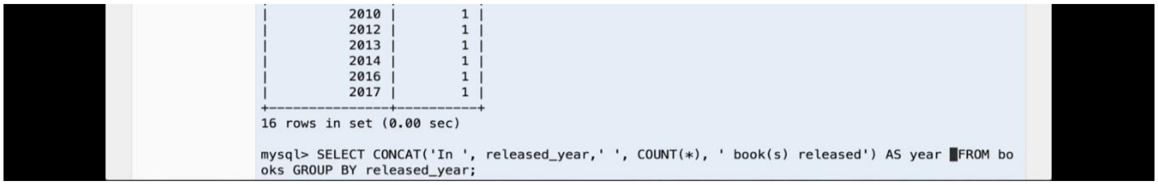
```
| The Namesake                      | Lahiri
| Interpreter of Maladies           | Lahiri
```

```
| Norse Mythology                   | Gaiman
| American Gods                     | Gaiman
```

```
| A Hologram for the King: A Novel | Eggers
```

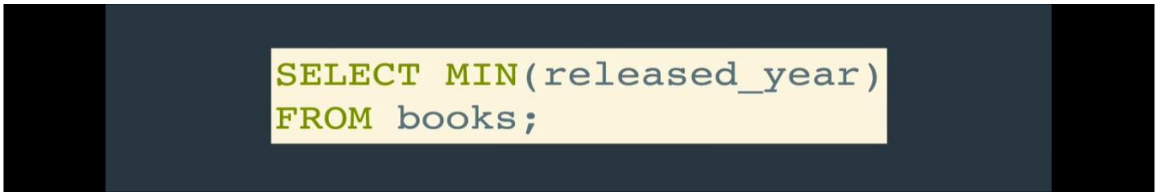


```
|          1945 |
|          2004 |
|          2005 |
|          2014 |
|          2001 |
|          2017 |
+---------------+
19 rows in set (0.00 sec)

mysql> SELECT released_year, COUNT█FROM books GROUP BY released_year;
```

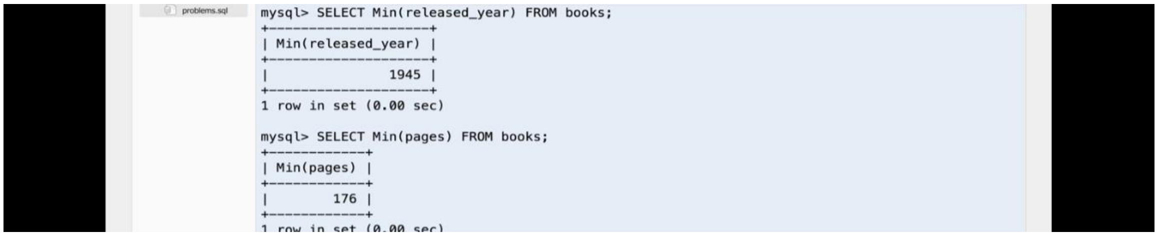


```
|          2010 |        1 |
|          2012 |        1 |
|          2013 |        1 |
|          2014 |        1 |
|          2016 |        1 |
|          2017 |        1 |
+---------------+----------+
16 rows in set (0.00 sec)

mysql> SELECT CONCAT('In ', released_year,' ', COUNT(*), ' book(s) released') AS year █FROM bo
oks GROUP BY released_year;
```

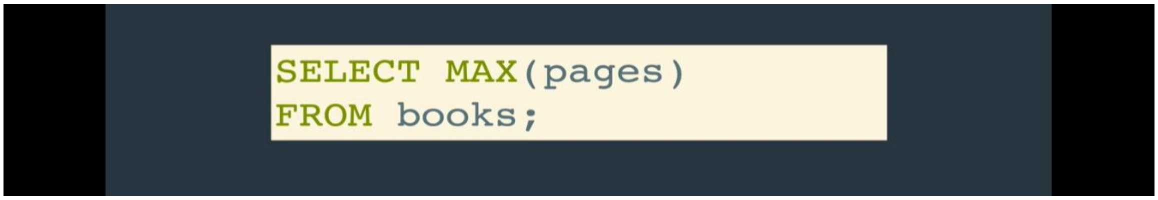
Minimum and maximum



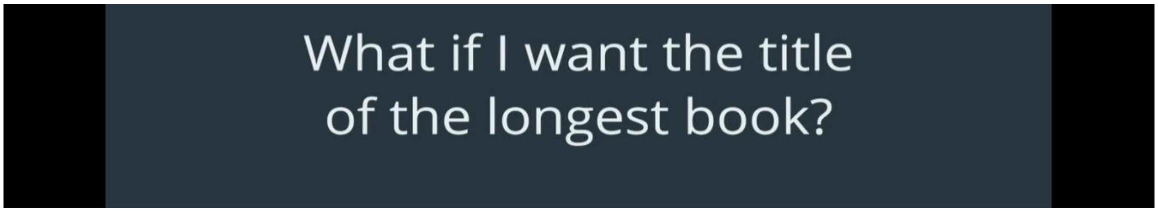
```
SELECT MIN(released_year)
FROM books;
```



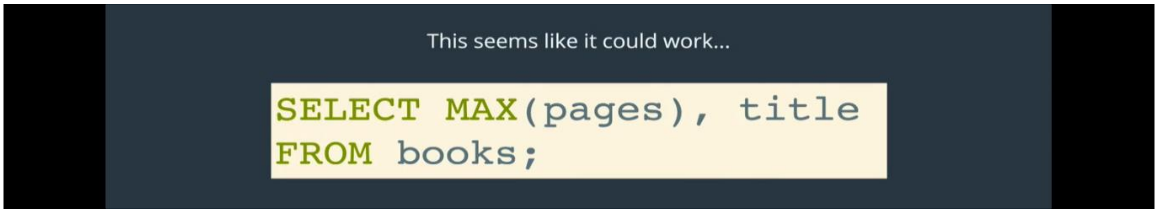
```
mysql> SELECT Min(released_year) FROM books;
+--------------------+
| Min(released_year) |
+--------------------+
|               1945 |
+--------------------+
1 row in set (0.00 sec)

mysql> SELECT Min(pages) FROM books;
+------------+
| Min(pages) |
+------------+
|        176 |
+------------+
1 row in set (0.00 sec)
```



```
SELECT MAX(pages)
FROM books;
```

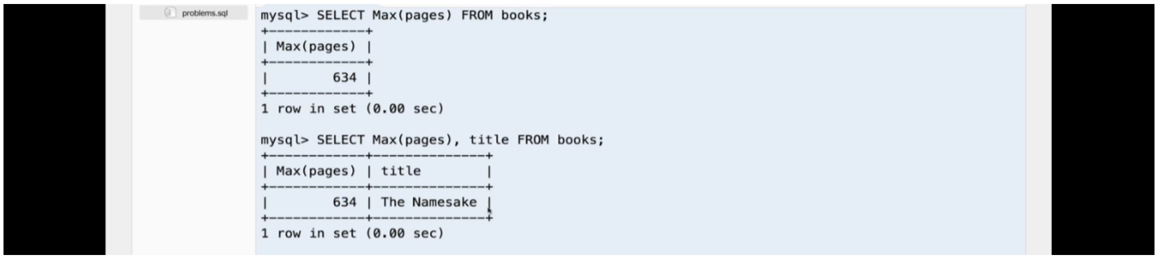


What if I want the title of the longest book?

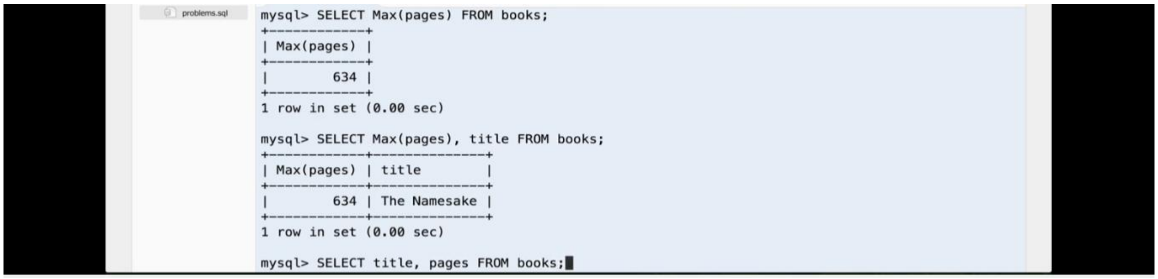


This seems like it could work...

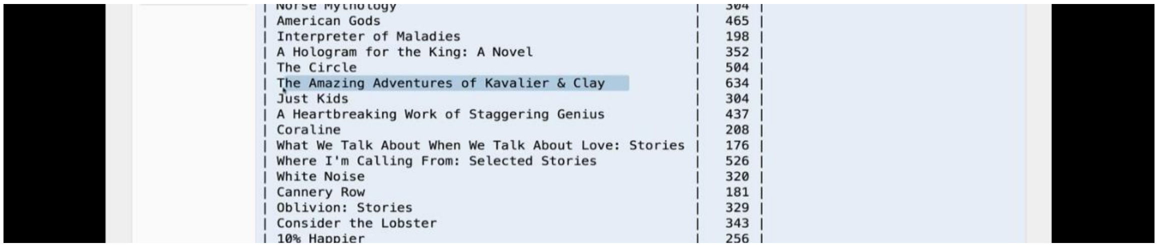
```
SELECT MAX(pages), title
FROM books;
```



```
mysql> SELECT Max(pages) FROM books;
+------------+
| Max(pages) |
+------------+
|        634 |
+------------+
1 row in set (0.00 sec)

mysql> SELECT Max(pages), title FROM books;
+------------+--------------+
| Max(pages) | title        |
+------------+--------------+
|        634 | The Namesake |
+------------+--------------+
1 row in set (0.00 sec)
```

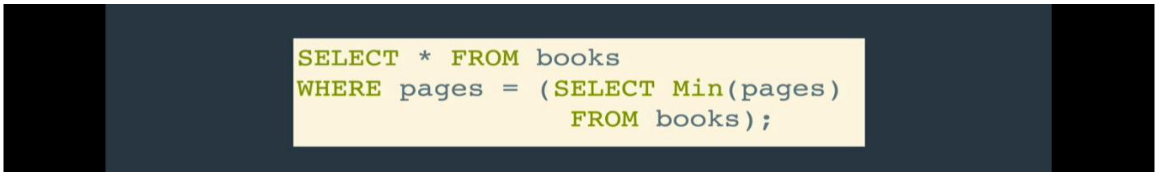


```
mysql> SELECT Max(pages) FROM books;
+------------+
| Max(pages) |
+------------+
|        634 |
+------------+
1 row in set (0.00 sec)

mysql> SELECT Max(pages), title FROM books;
+------------+--------------+
| Max(pages) | title        |
+------------+--------------+
|        634 | The Namesake |
+------------+--------------+
1 row in set (0.00 sec)

mysql> SELECT title, pages FROM books;
```

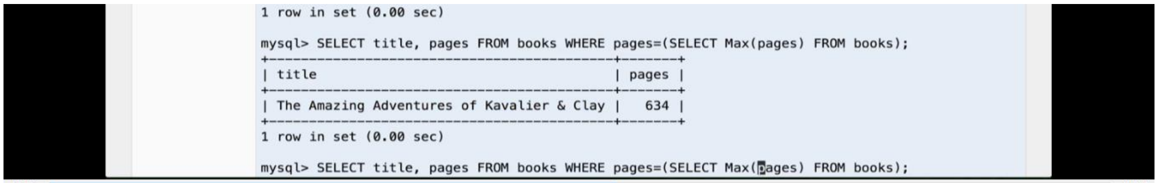


```
| Norse Mythology                                     |   304 |
| American Gods                                       |   465 |
| Interpreter of Maladies                             |   198 |
| A Hologram for the King: A Novel                    |   352 |
| The Circle                                          |   504 |
| The Amazing Adventures of Kavalier & Clay           |   634 |
| Just Kids                                           |   304 |
| A Heartbreaking Work of Staggering Genius           |   437 |
| Coraline                                            |   208 |
| What We Talk About When We Talk About Love: Stories |   176 |
| Where I'm Calling From: Selected Stories            |   526 |
| White Noise                                         |   320 |
| Cannery Row                                         |   181 |
| Oblivion: Stories                                   |   329 |
| Consider the Lobster                                |   343 |
| 10% Happier                                         |   256 |
```

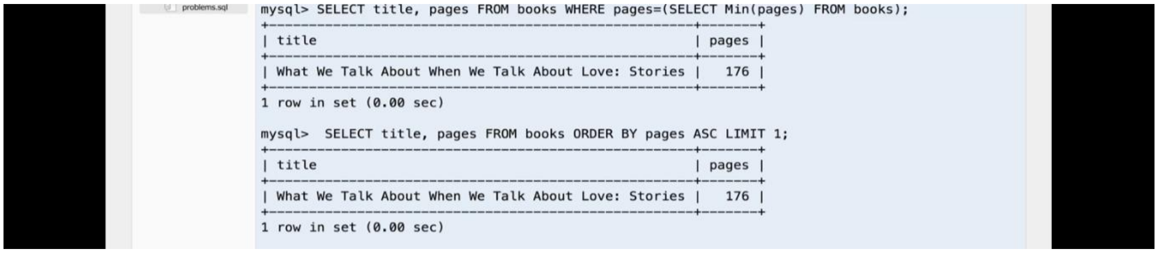
Sub quarries (on quarries into another)



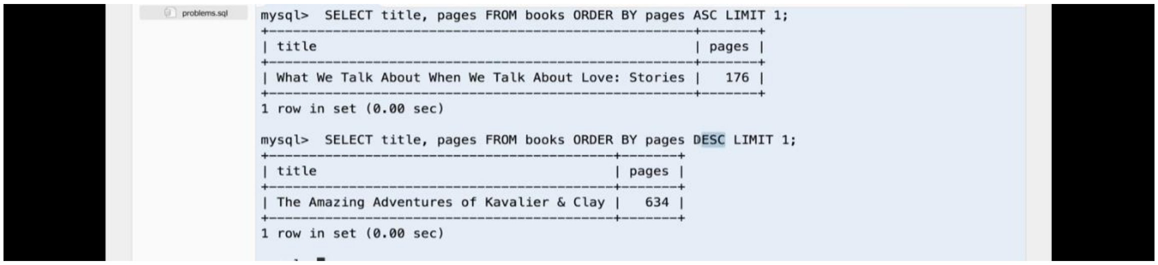
```
SELECT * FROM books
WHERE pages = (SELECT Min(pages)
               FROM books);
```



```
1 row in set (0.00 sec)

mysql> SELECT title, pages FROM books WHERE pages=(SELECT Max(pages) FROM books);
+-------------------------------------------+-------+
| title                                     | pages |
+-------------------------------------------+-------+
| The Amazing Adventures of Kavalier & Clay |   634 |
+-------------------------------------------+-------+
1 row in set (0.00 sec)

mysql> SELECT title, pages FROM books WHERE pages=(SELECT Max(pages) FROM books);
```

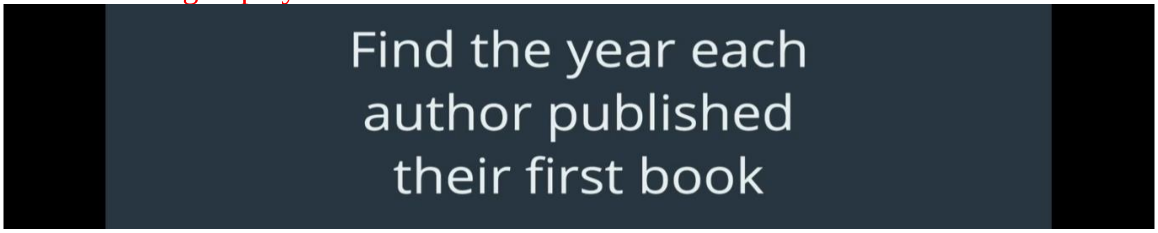


```
mysql> SELECT title, pages FROM books WHERE pages=(SELECT Min(pages) FROM books);
+-----------------------------------------------------+-------+
| title                                               | pages |
+-----------------------------------------------------+-------+
| What We Talk About When We Talk About Love: Stories |   176 |
+-----------------------------------------------------+-------+
1 row in set (0.00 sec)

mysql>  SELECT title, pages FROM books ORDER BY pages ASC LIMIT 1;
+-----------------------------------------------------+-------+
| title                                               | pages |
+-----------------------------------------------------+-------+
| What We Talk About When We Talk About Love: Stories |   176 |
+-----------------------------------------------------+-------+
1 row in set (0.00 sec)
```

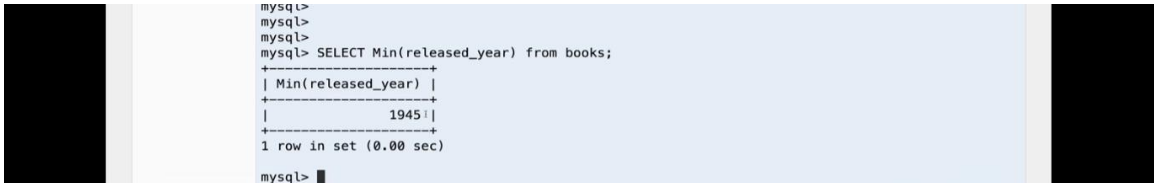


```
mysql>  SELECT title, pages FROM books ORDER BY pages ASC LIMIT 1;
+-----------------------------------------------------+-------+
| title                                               | pages |
+-----------------------------------------------------+-------+
| What We Talk About When We Talk About Love: Stories |   176 |
+-----------------------------------------------------+-------+
1 row in set (0.00 sec)

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)
```

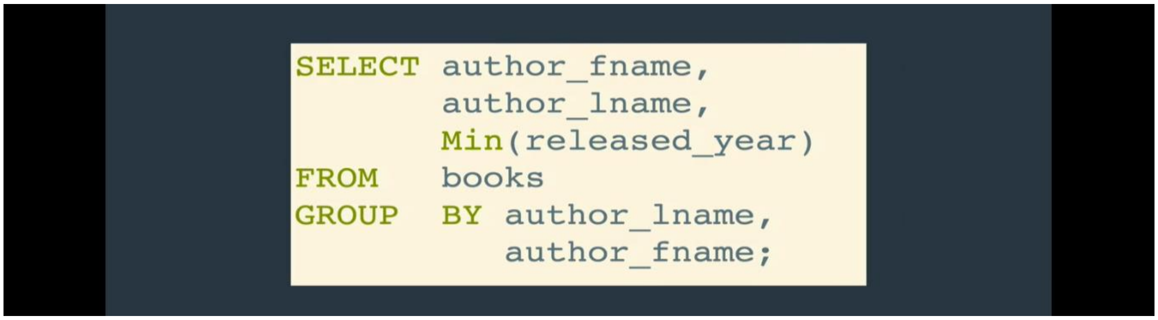
Min and max group by



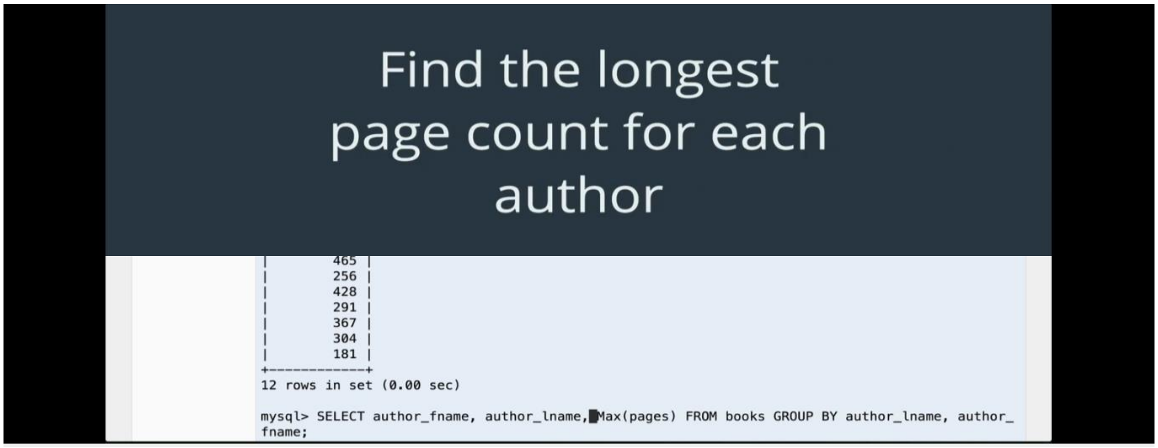
Find the year each author published their first book



```
mysql>
mysql>
mysql>
mysql> SELECT Min(released_year) from books;
+--------------------+
| Min(released_year) |
+--------------------+
|               1945 |
+--------------------+
1 row in set (0.00 sec)

mysql> 
```

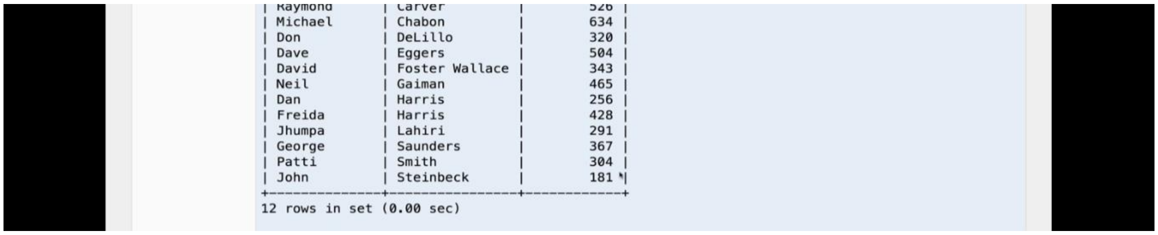


```
SELECT author_fname,
       author_lname,
       Min(released_year)
FROM   books
GROUP  BY author_lname,
          author_fname;
```

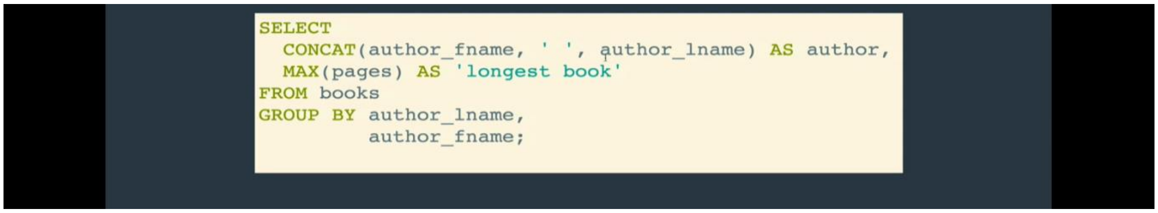


Find the longest page count for each author

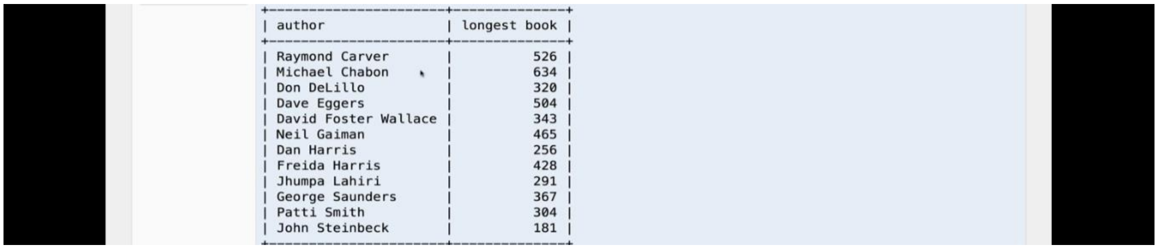
```
|       465 |
|       256 |
|       428 |
|       291 |
|       367 |
|       304 |
|       181 |
+-----------+
12 rows in set (0.00 sec)

mysql> SELECT author_fname, author_lname, Max(pages) FROM books GROUP BY author_lname, author_
fname;
```



```
| Raymond      | Carver         |        526 |
| Michael      | Chabon         |        634 |
| Don          | DeLillo        |        320 |
| Dave         | Eggers         |        504 |
| David        | Foster Wallace |        343 |
| Neil         | Gaiman         |        465 |
| Dan          | Harris         |        256 |
| Freida       | Harris         |        428 |
| Jhumpa       | Lahiri         |        291 |
| George       | Saunders       |        367 |
| Patti        | Smith          |        304 |
| John         | Steinbeck      |        181 |
+--------------+----------------+------------+
12 rows in set (0.00 sec)
```

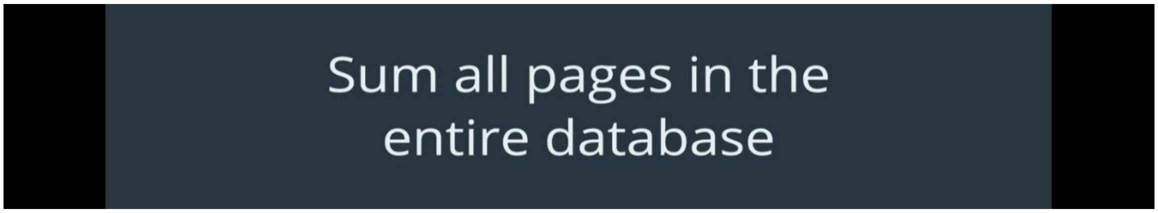


```
SELECT
  CONCAT(author_fname, ' ', author_lname) AS author,
  MAX(pages) AS 'longest book'
FROM books
GROUP BY author_lname,
         author_fname;
```

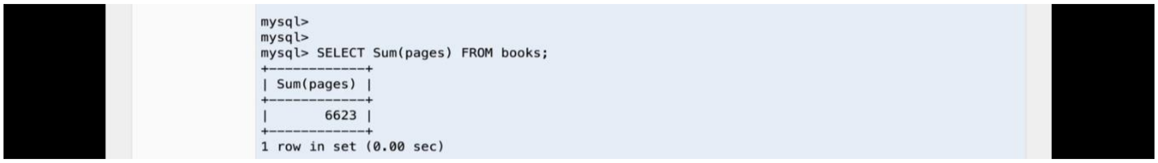


```
+----------------------+--------------+
| author               | longest book |
+----------------------+--------------+
| Raymond Carver       |          526 |
| Michael Chabon       |          634 |
| Don DeLillo          |          320 |
| Dave Eggers          |          504 |
| David Foster Wallace |          343 |
| Neil Gaiman          |          465 |
| Dan Harris           |          256 |
| Freida Harris        |          428 |
| Jhumpa Lahiri        |          291 |
| George Saunders      |          367 |
| Patti Smith          |          304 |
| John Steinbeck       |          181 |
+----------------------+--------------+
```

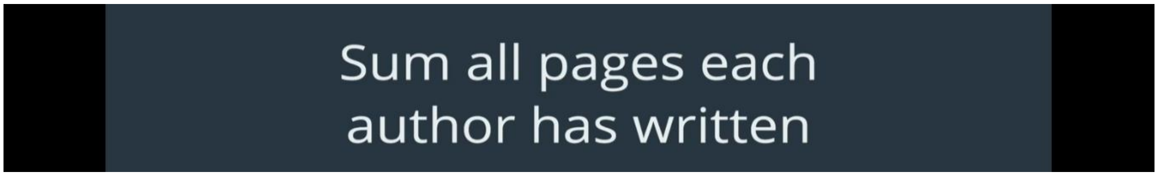
Sum



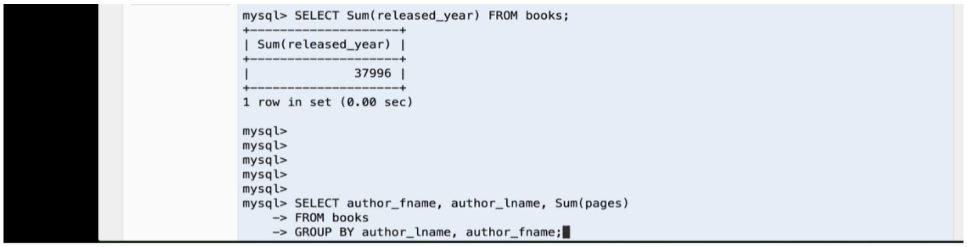
Sum all pages in the entire database



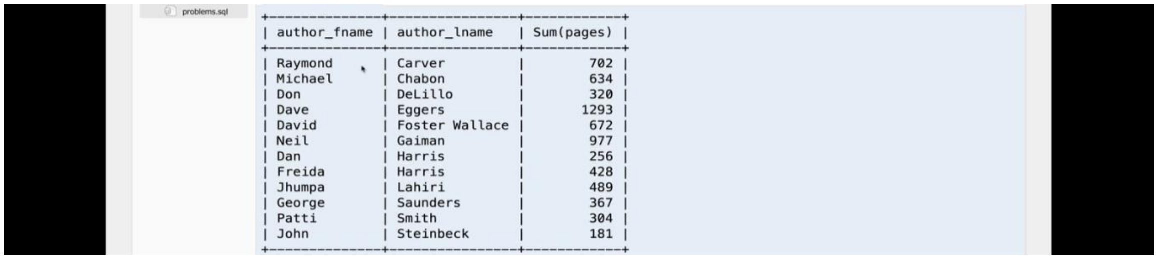
```
mysql>
mysql>
mysql> SELECT Sum(pages) FROM books;
+------------+
| Sum(pages) |
+------------+
|       6623 |
+------------+
1 row in set (0.00 sec)
```



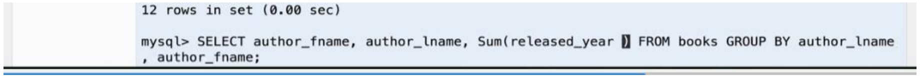
Sum all pages each author has written



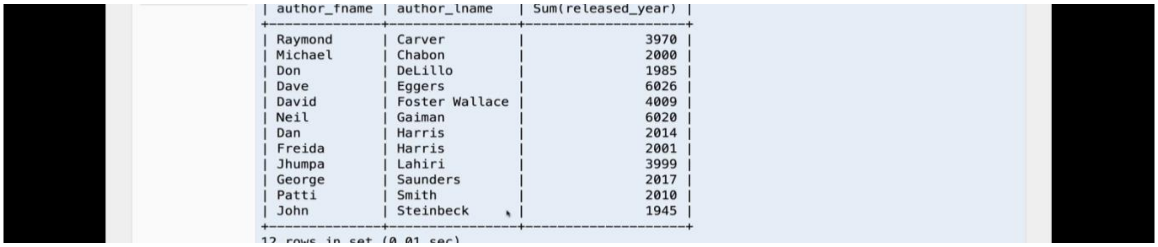
```
mysql> SELECT Sum(released_year) FROM books;
+--------------------+
| Sum(released_year) |
+--------------------+
|              37996 |
+--------------------+
1 row in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT author_fname, author_lname, Sum(pages)
    -> FROM books
    -> GROUP BY author_lname, author_fname;
```



```
+--------------+----------------+------------+
| author_fname | author_lname   | Sum(pages) |
+--------------+----------------+------------+
| Raymond      | Carver         |        702 |
| Michael      | Chabon         |        634 |
| Don          | DeLillo        |        320 |
| Dave         | Eggers         |       1293 |
| David        | Foster Wallace |        672 |
| Neil         | Gaiman         |        977 |
| Dan          | Harris         |        256 |
| Freida       | Harris         |        428 |
| Jhumpa       | Lahiri         |        489 |
| George       | Saunders       |        367 |
| Patti        | Smith          |        304 |
| John         | Steinbeck      |        181 |
+--------------+----------------+------------+
```

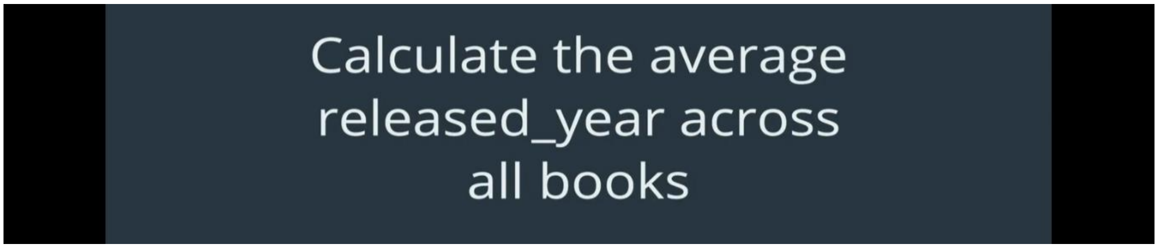


```
12 rows in set (0.00 sec)

mysql> SELECT author_fname, author_lname, Sum(released_year) FROM books GROUP BY author_lname
, author_fname;
```

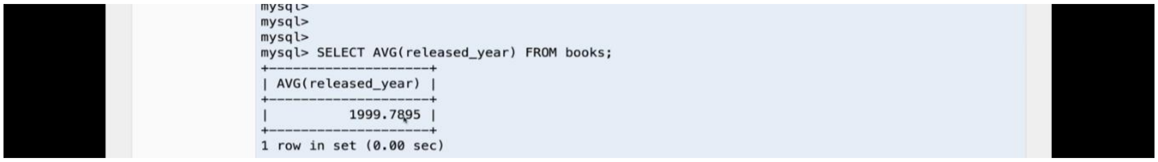


```
| author_fname | author_lname   | Sum(released_year) |
+--------------+----------------+--------------------+
| Raymond      | Carver         |               3970 |
| Michael      | Chabon         |               2000 |
| Don          | DeLillo        |               1985 |
| Dave         | Eggers         |               6026 |
| David        | Foster Wallace |               4009 |
| Neil         | Gaiman         |               6020 |
| Dan          | Harris         |               2014 |
| Freida       | Harris         |               2001 |
| Jhumpa       | Lahiri         |               3999 |
| George       | Saunders       |               2017 |
| Patti        | Smith          |               2010 |
| John         | Steinbeck      |               1945 |
+--------------+----------------+--------------------+
12 rows in set (0.01 sec)
```

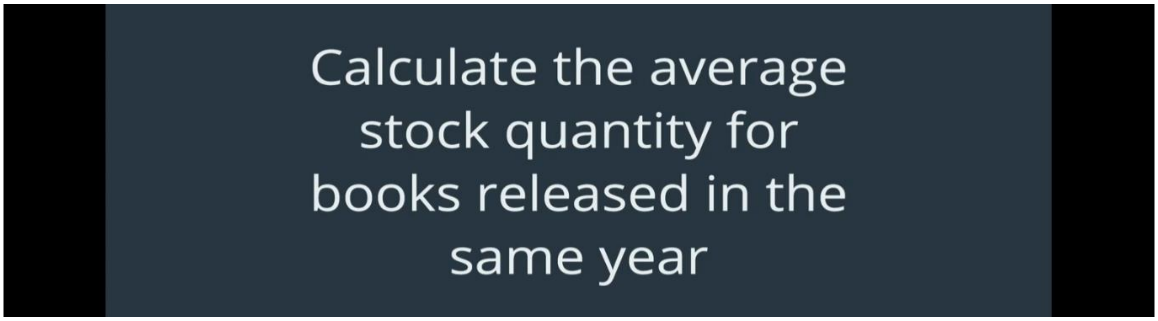
Average



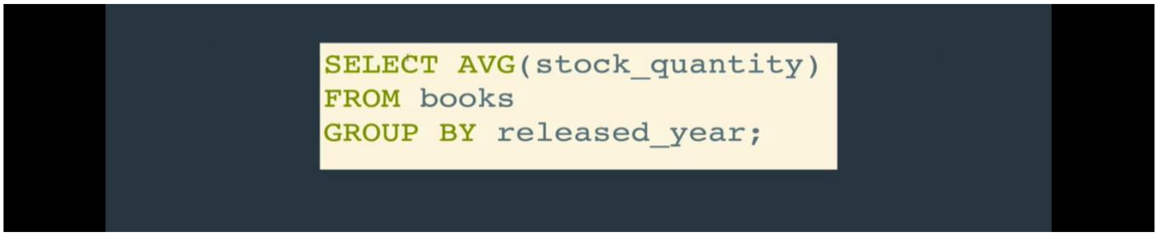
Calculate the average released_year across all books



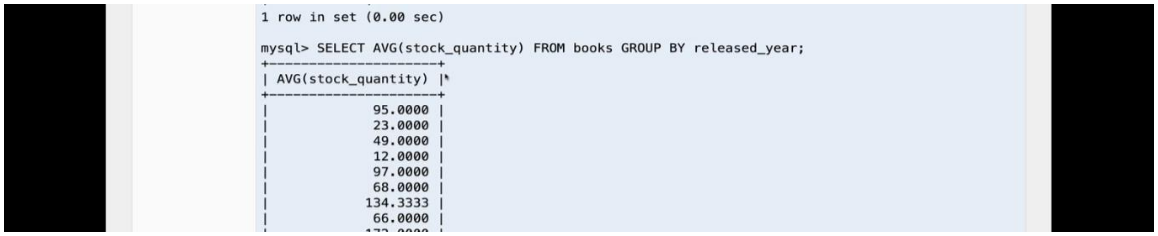
```
mysql>
mysql>
mysql>
mysql> SELECT AVG(released_year) FROM books;
+--------------------+
| AVG(released_year) |
+--------------------+
|          1999.7895 |
+--------------------+
1 row in set (0.00 sec)
```



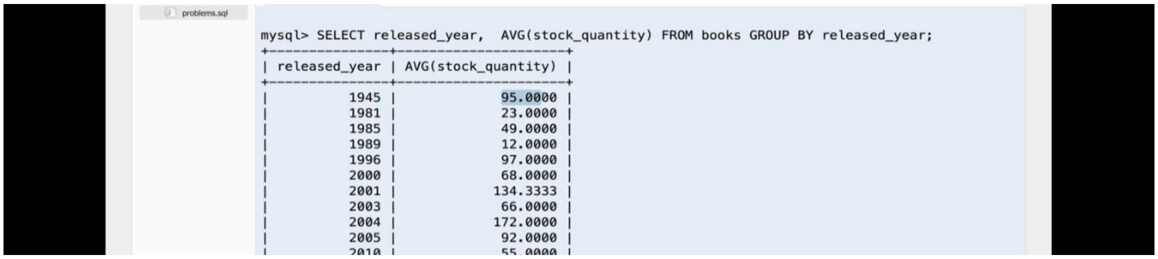
Calculate the average stock quantity for books released in the same year



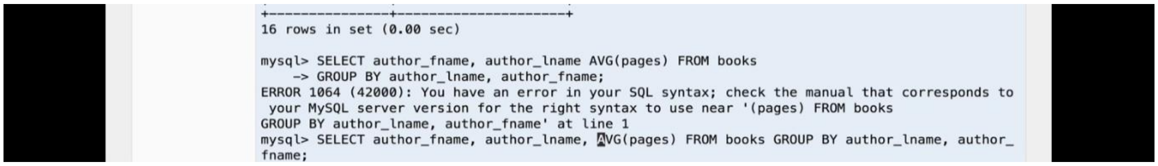
```
SELECT AVG(stock_quantity)
FROM books
GROUP BY released_year;
```



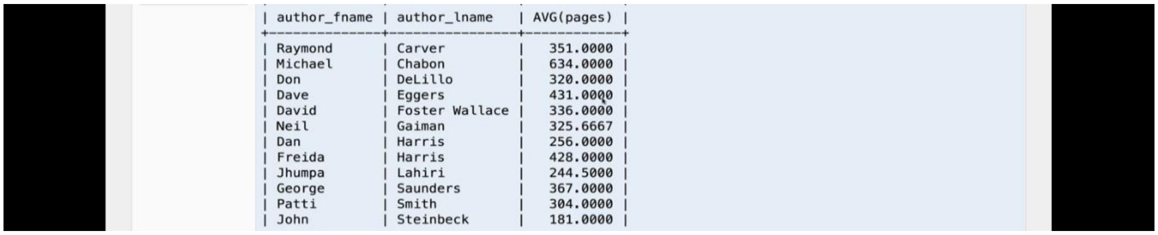
```
1 row in set (0.00 sec)

mysql> SELECT AVG(stock_quantity) FROM books GROUP BY released_year;
+---------------------+
| AVG(stock_quantity) |
+---------------------+
|             95.0000 |
|             23.0000 |
|             49.0000 |
|             12.0000 |
|             97.0000 |
|             68.0000 |
|            134.3333 |
|             66.0000 |
|            172.0000 |
```



```
mysql> SELECT released_year,  AVG(stock_quantity) FROM books GROUP BY released_year;
+---------------+---------------------+
| released_year | AVG(stock_quantity) |
+---------------+---------------------+
|          1945 |             95.0000 |
|          1981 |             23.0000 |
|          1985 |             49.0000 |
|          1989 |             12.0000 |
|          1996 |             97.0000 |
|          2000 |             68.0000 |
|          2001 |            134.3333 |
|          2003 |             66.0000 |
|          2004 |            172.0000 |
|          2005 |             92.0000 |
|          2010 |             55.0000 |
```

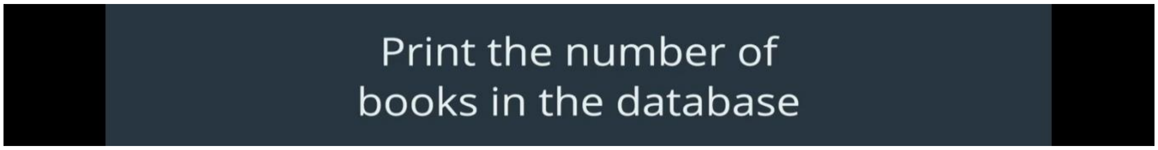


```
+----------------+----------------------+
16 rows in set (0.00 sec)

mysql> SELECT author_fname, author_lname AVG(pages) FROM books
    -> GROUP BY author_lname, author_fname;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
 your MySQL server version for the right syntax to use near '(pages) FROM books
GROUP BY author_lname, author_fname' at line 1
mysql> SELECT author_fname, author_lname, AVG(pages) FROM books GROUP BY author_lname, author_
fname;
```

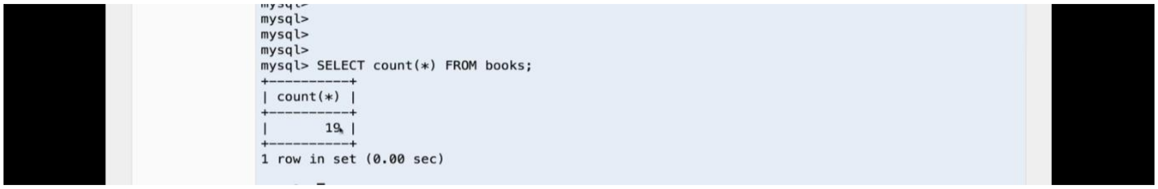


```
| author_fname | author_lname   | AVG(pages) |
+--------------+----------------+------------+
| Raymond      | Carver         |   351.0000 |
| Michael      | Chabon         |   634.0000 |
| Don          | DeLillo        |   320.0000 |
| Dave         | Eggers         |   431.0000 |
| David        | Foster Wallace |   336.0000 |
| Neil         | Gaiman         |   325.6667 |
| Dan          | Harris         |   256.0000 |
| Freida       | Harris         |   428.0000 |
| Jhumpa       | Lahiri         |   244.5000 |
| George       | Saunders       |   367.0000 |
| Patti        | Smith          |   304.0000 |
| John         | Steinbeck      |   181.0000 |
```

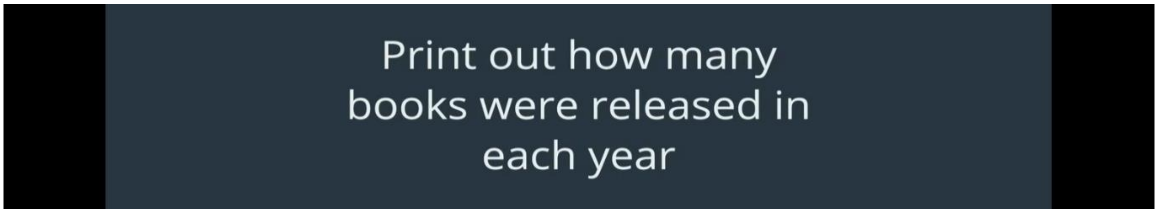
Exercises



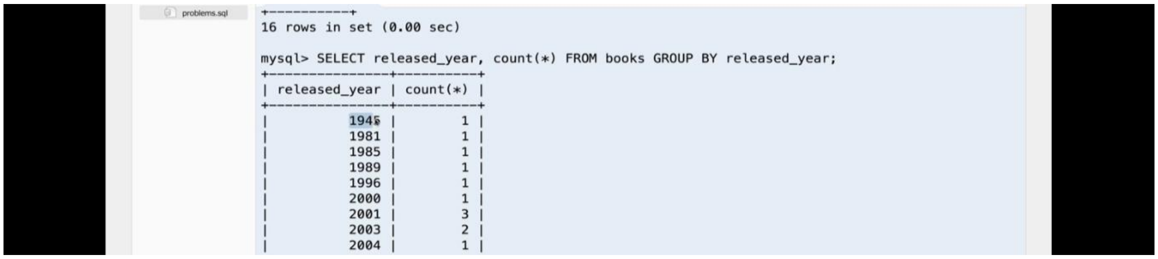
Print the number of books in the database



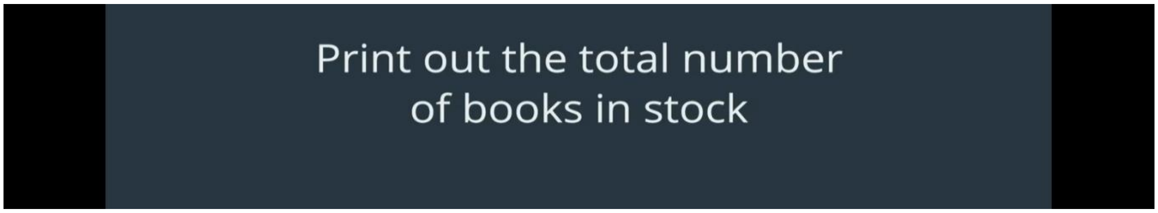
```
mysql>
mysql>
mysql>
mysql> SELECT count(*) FROM books;
+----------+
| count(*) |
+----------+
|       19 |
+----------+
1 row in set (0.00 sec)
```



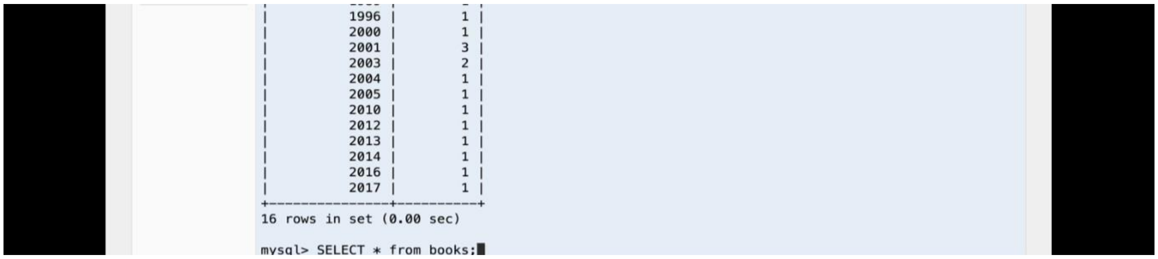
Print out how many books were released in each year



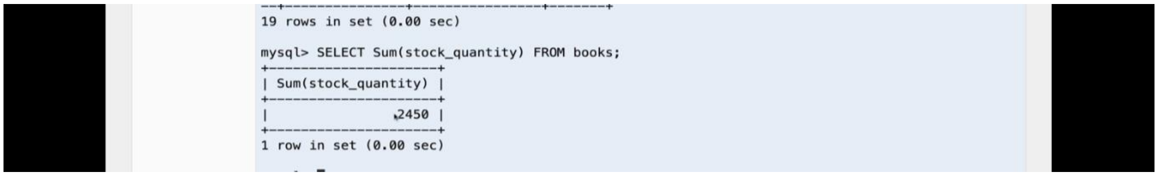
```
+----------+
16 rows in set (0.00 sec)

mysql> SELECT released_year, count(*) FROM books GROUP BY released_year;
+---------------+----------+
| released_year | count(*) |
+---------------+----------+
|          1945 |        1 |
|          1981 |        1 |
|          1985 |        1 |
|          1989 |        1 |
|          1996 |        1 |
|          2000 |        1 |
|          2001 |        3 |
|          2003 |        2 |
|          2004 |        1 |
```



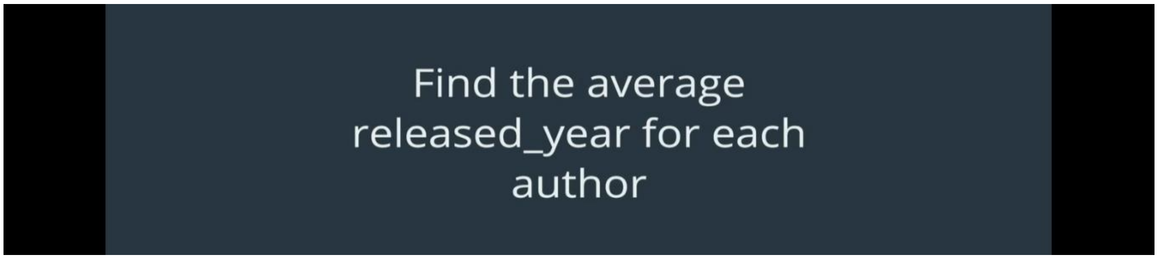
Print out the total number of books in stock



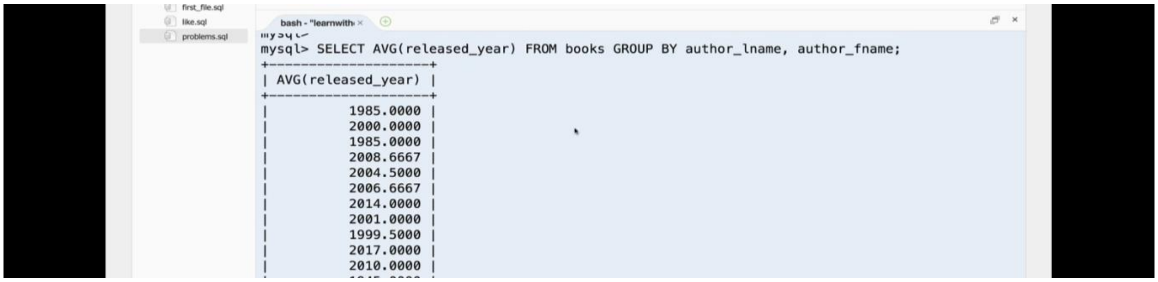
```
|         1996 |         1 |
|         2000 |         1 |
|         2001 |         3 |
|         2003 |         2 |
|         2004 |         1 |
|         2005 |         1 |
|         2010 |         1 |
|         2012 |         1 |
|         2013 |         1 |
|         2014 |         1 |
|         2016 |         1 |
|         2017 |         1 |
+--------------+-----------+
16 rows in set (0.00 sec)

mysql> SELECT * from books;
```



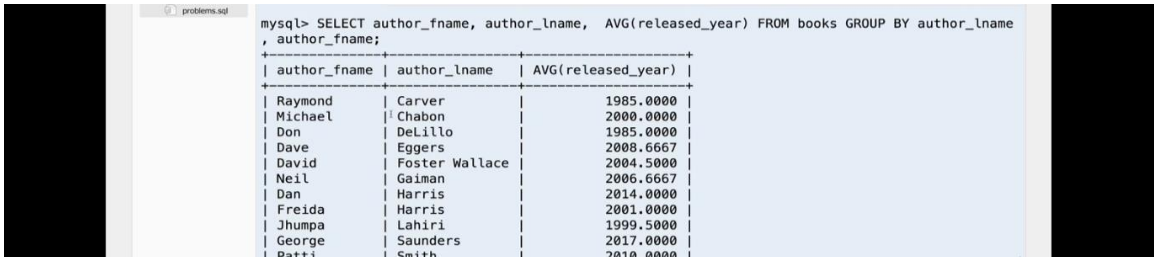
```
--+----------------+----------------+--------+
19 rows in set (0.00 sec)

mysql> SELECT Sum(stock_quantity) FROM books;
+---------------------+
| Sum(stock_quantity) |
+---------------------+
|                2450 |
+---------------------+
1 row in set (0.00 sec)
```



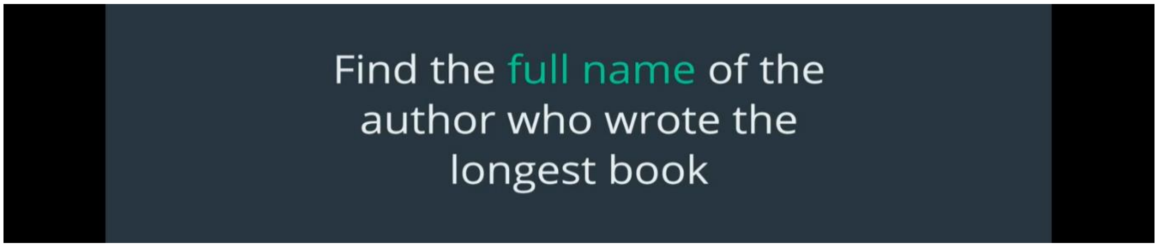
Find the average released_year for each author



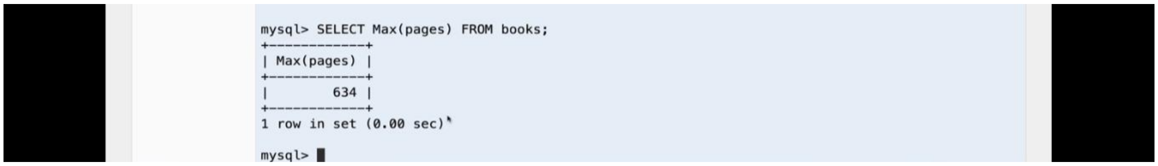
```
mysql> SELECT AVG(released_year) FROM books GROUP BY author_lname, author_fname;
+--------------------+
| AVG(released_year) |
+--------------------+
|          1985.0000 |
|          2000.0000 |
|          1985.0000 |
|          2008.6667 |
|          2004.5000 |
|          2006.6667 |
|          2014.0000 |
|          2001.0000 |
|          1999.5000 |
|          2017.0000 |
|          2010.0000 |
```



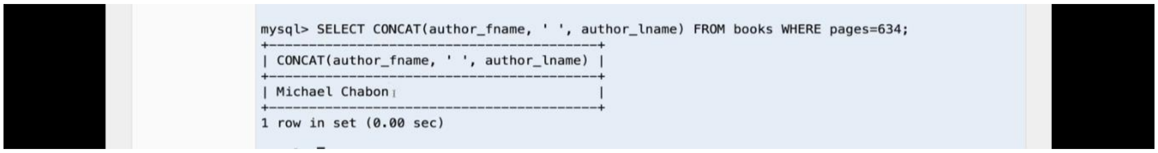
```
mysql> SELECT author_fname, author_lname,  AVG(released_year) FROM books GROUP BY author_lname
, author_fname;
+--------------+----------------+--------------------+
| author_fname | author_lname   | AVG(released_year) |
+--------------+----------------+--------------------+
| Raymond      | Carver         |          1985.0000 |
| Michael      | Chabon         |          2000.0000 |
| Don          | DeLillo        |          1985.0000 |
| Dave         | Eggers         |          2008.6667 |
| David        | Foster Wallace |          2004.5000 |
| Neil         | Gaiman         |          2006.6667 |
| Dan          | Harris         |          2014.0000 |
| Freida       | Harris         |          2001.0000 |
| Jhumpa       | Lahiri         |          1999.5000 |
| George       | Saunders       |          2017.0000 |
| Patti        | Smith          |          2010.0000 |
```



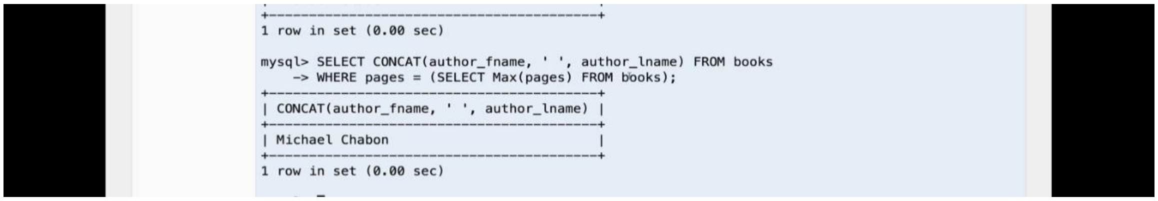
Find the full name of the author who wrote the longest book



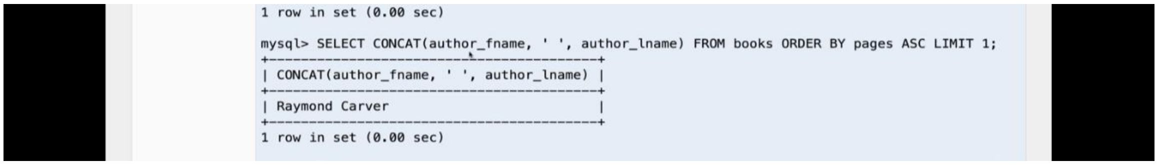
```
mysql> SELECT Max(pages) FROM books;
+------------+
| Max(pages) |
+------------+
|        634 |
+------------+
1 row in set (0.00 sec)

mysql> █
```



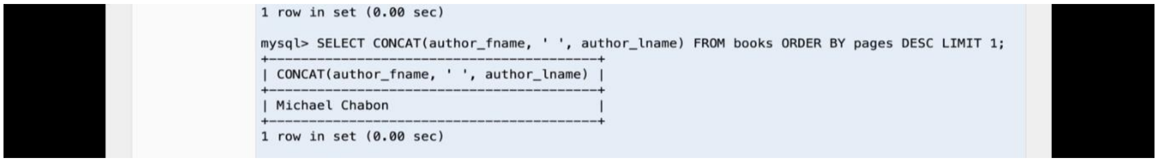
```
mysql> SELECT CONCAT(author_fname, ' ', author_lname) FROM books WHERE pages=634;
+-----------------------------------------+
| CONCAT(author_fname, ' ', author_lname) |
+-----------------------------------------+
| Michael Chabon                          |
+-----------------------------------------+
1 row in set (0.00 sec)
```



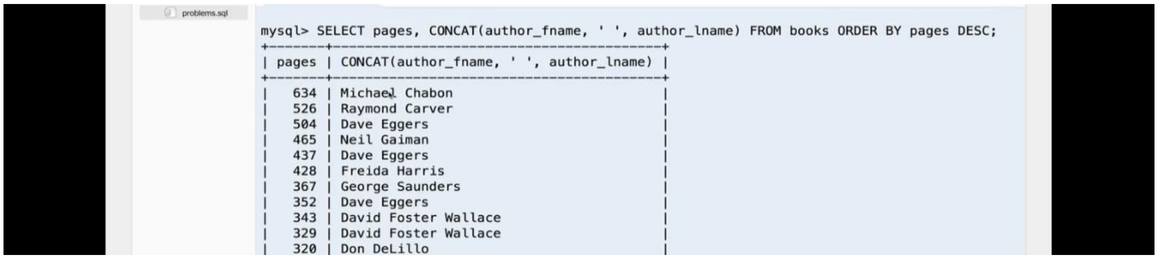
```
+-----------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT CONCAT(author_fname, ' ', author_lname) FROM books
    -> WHERE pages = (SELECT Max(pages) FROM books);
+-----------------------------------------+
| CONCAT(author_fname, ' ', author_lname) |
+-----------------------------------------+
| Michael Chabon                          |
+-----------------------------------------+
1 row in set (0.00 sec)
```



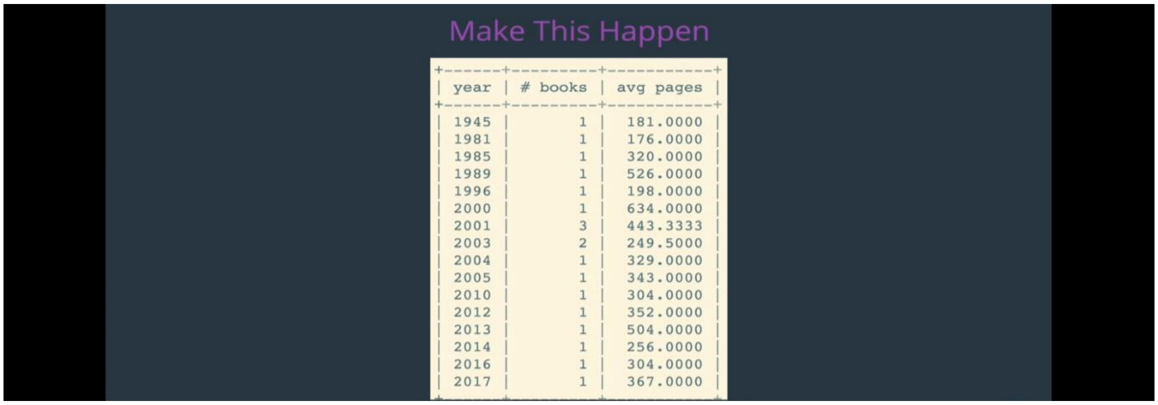
```
1 row in set (0.00 sec)

mysql> SELECT CONCAT(author_fname, ' ', author_lname) FROM books ORDER BY pages ASC LIMIT 1;
+-----------------------------------------+
| CONCAT(author_fname, ' ', author_lname) |
+-----------------------------------------+
| Raymond Carver                          |
+-----------------------------------------+
1 row in set (0.00 sec)
```



```
1 row in set (0.00 sec)

mysql> SELECT CONCAT(author_fname, ' ', author_lname) FROM books ORDER BY pages DESC LIMIT 1;
+-----------------------------------------+
| CONCAT(author_fname, ' ', author_lname) |
+-----------------------------------------+
| Michael Chabon                          |
+-----------------------------------------+
1 row in set (0.00 sec)
```

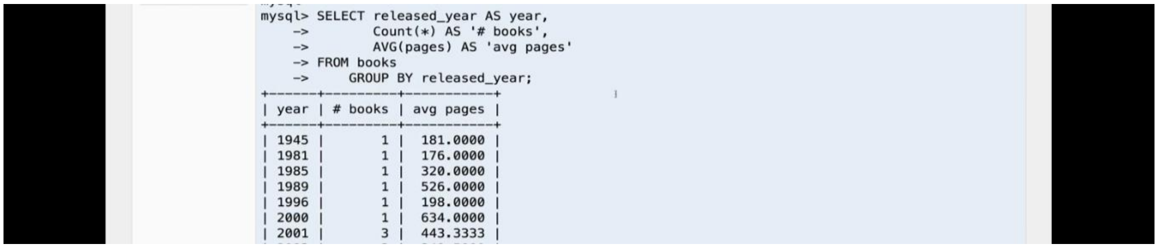


```
mysql> SELECT pages, CONCAT(author_fname, ' ', author_lname) FROM books ORDER BY pages DESC;
+-------+-----------------------------------------+
| pages | CONCAT(author_fname, ' ', author_lname) |
+-------+-----------------------------------------+
|   634 | Michael Chabon                          |
|   526 | Raymond Carver                          |
|   504 | Dave Eggers                             |
|   465 | Neil Gaiman                             |
|   437 | Dave Eggers                             |
|   428 | Freida Harris                           |
|   367 | George Saunders                         |
|   352 | Dave Eggers                             |
|   343 | David Foster Wallace                    |
|   329 | David Foster Wallace                    |
|   320 | Don DeLillo                             |
```



Make This Happen

```
+------+---------+-----------+
| year | # books | avg pages |
+------+---------+-----------+
| 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 |
```



```
mysql> SELECT released_year AS year,
    ->        Count(*) AS '# books',
    ->        AVG(pages) AS 'avg pages'
    -> FROM books
    ->     GROUP BY released_year;
+------+---------+-----------+
| year | # books | avg pages |
+------+---------+-----------+
| 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 |
```