CMPE 2400 Databases

Grouping Data

Recap

- SELECT
  - So far, we have learned how to retrieve selected data from a chosen set of columns from a single table.
- WHERE
  - Once we have a set of data, we have looked at applying filters to the data set so that we can eliminate rows that are outside the scope of a given problem.
- ORDER BY
  - Once the data has been filtered, it is often beneficial to order the data so that it is more usable.
- FUNCTIONS
  - Perform calculations and other manipulations on the data that we retrieve

Aggregate Functions

- Aggregate functions perform a calculation on a set of values, and return a single summarizing value.
  - All aggregate functions are *deterministic* in nature.
  - Aggregate functions may be used:
    - In the select list of a select statement.
    - In a having clause (more on this soon).
    - In the order by clause.

Aggregate Functions

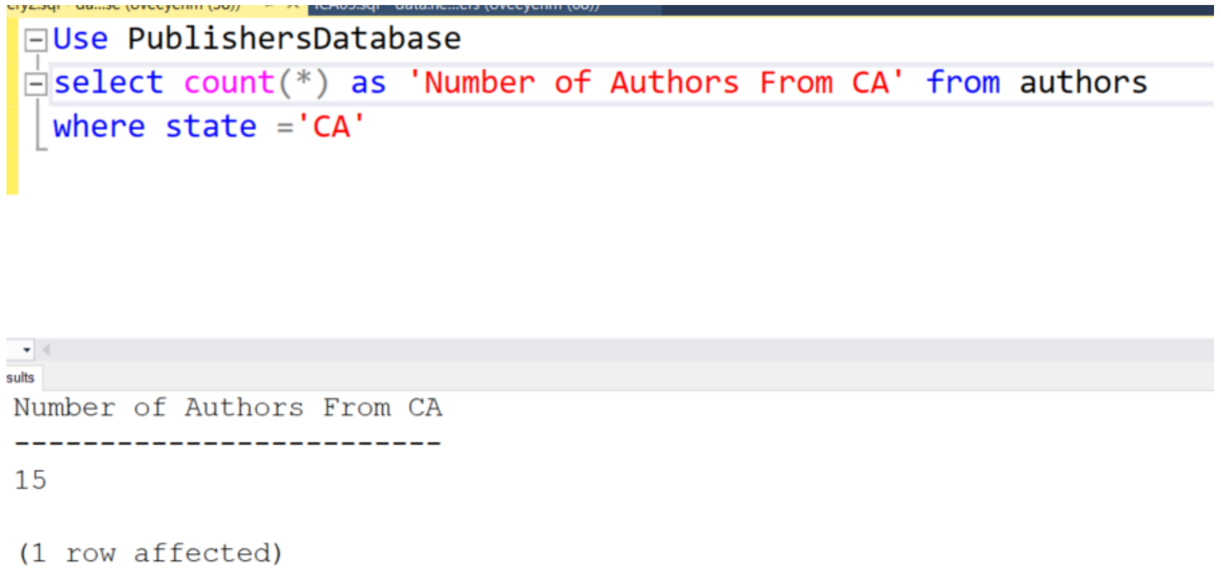
- APPROX_COUNT_DISTINCT
- **AVG**
- CHECKSUM_AGG
- **COUNT**
- COUNT_BIG
- GROUPING
- GROUPING_ID
- **MAX**
- **MIN**
- STDEV
- STDEVP
- STRING_AGG
- **SUM**
- VAR
- VARP

Aggregate Functions … example

- sum (*[all | distinct] expression*)
  - Returns the summation of all *expression* values in the most precise *expression* data type.
    - *all* will cause all non-null values to be considered. This is default.
    - *distinct* will cause only one of each discreet non-null value to be considered.
    - *expression* is any numeric expression except the bit type.
      - *No sub-queries or nested aggregate functions permitted.*
    - Return type is determined by the most precise *expression* type.

Aggregate Example

- How many authors from the Publishers database are from California?



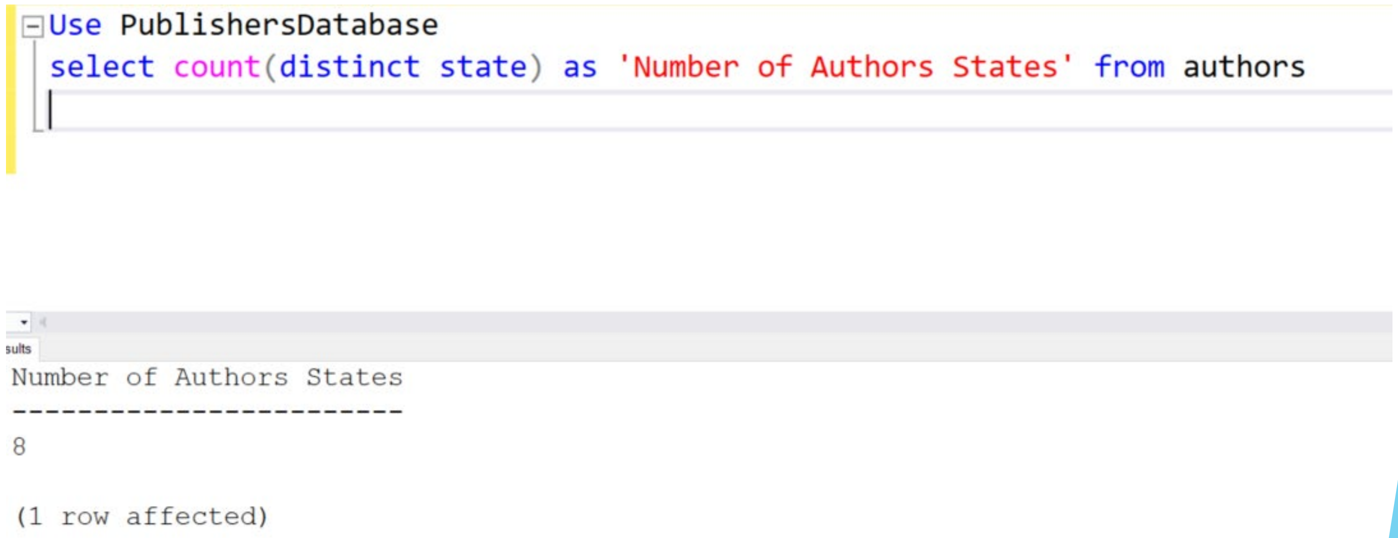
```
Use PublishersDatabase
select count(*) as 'Number of Authors From CA' from authors
where state ='CA'
```

```
Number of Authors From CA
-------------------------
15

(1 row affected)
```

Aggregate Example-using distinct

- Now let's count the number of states from which authors come



```
Use PublishersDatabase
select count(distinct state) as 'Number of Authors States' from authors
```

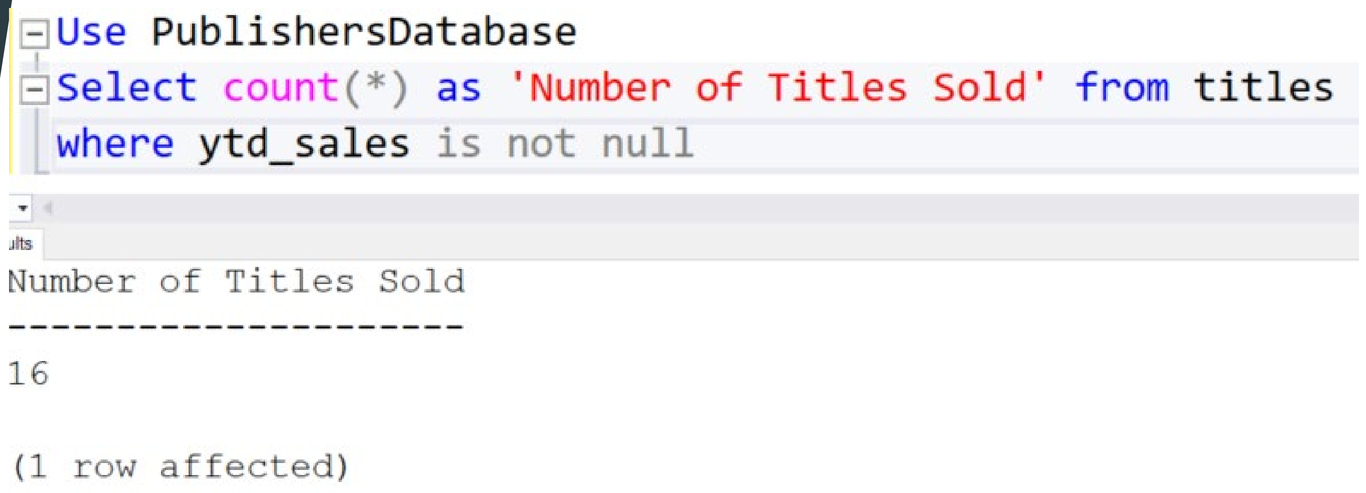
sults

```
Number of Authors States
------------------------
8

(1 row affected)
```

Aggregate Example- Counting Non-null values

- Let’s count all titles for which there has been some sale during the year.
- There are 2 ways of achieving that

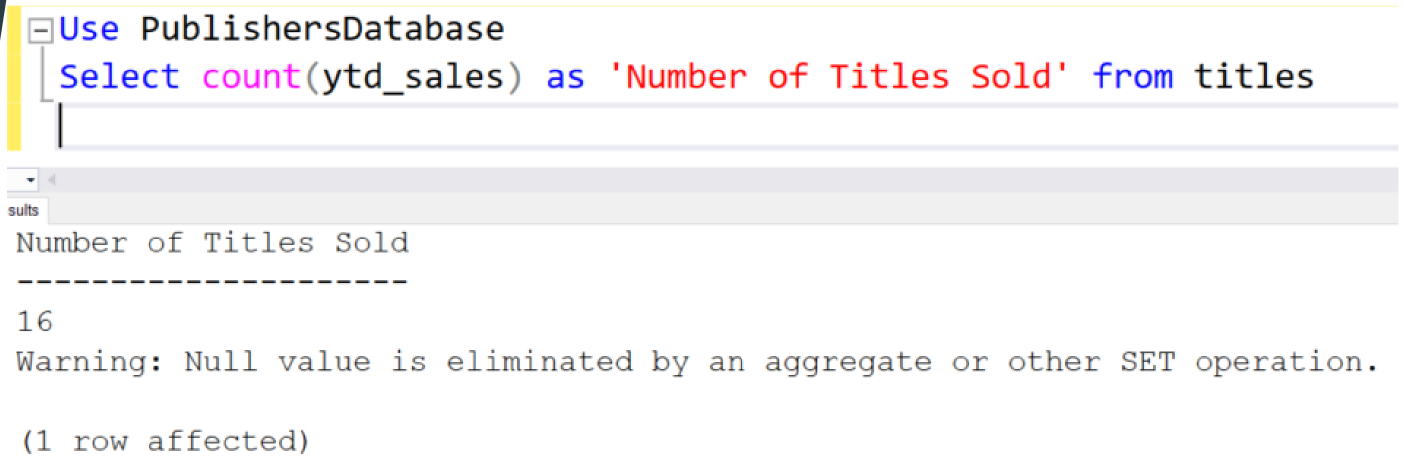


```
Use PublishersDatabase
Select count(*) as 'Number of Titles Sold' from titles
where ytd_sales is not null
```

```
Number of Titles Sold
---------------------
16

(1 row affected)
```

Aggregate Example-Counting Non-null values



```
Use PublishersDatabase
Select count(ytd_sales) as 'Number of Titles Sold' from titles
```

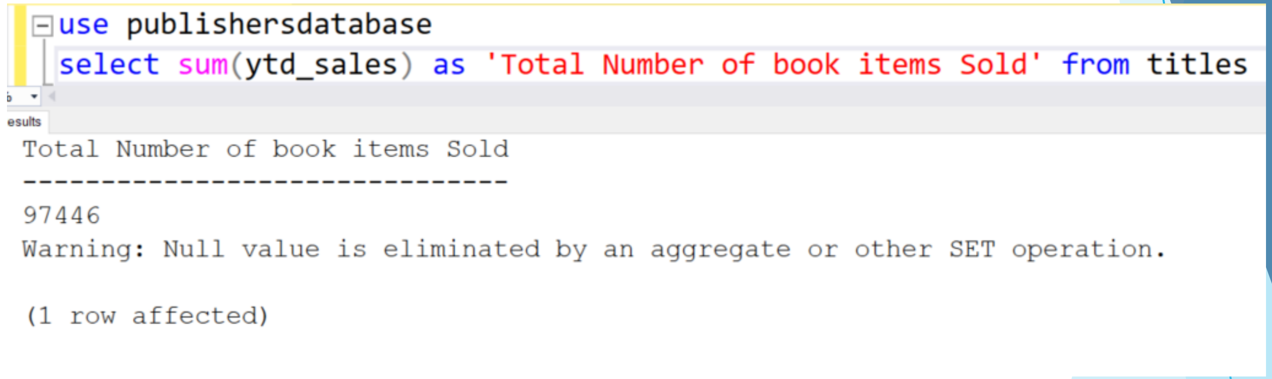
sults

```
Number of Titles Sold
---------------------
16
Warning: Null value is eliminated by an aggregate or other SET operation.

(1 row affected)
```

Aggregate Example- Using Sum

Let’s find the total copies of all books sold during the year



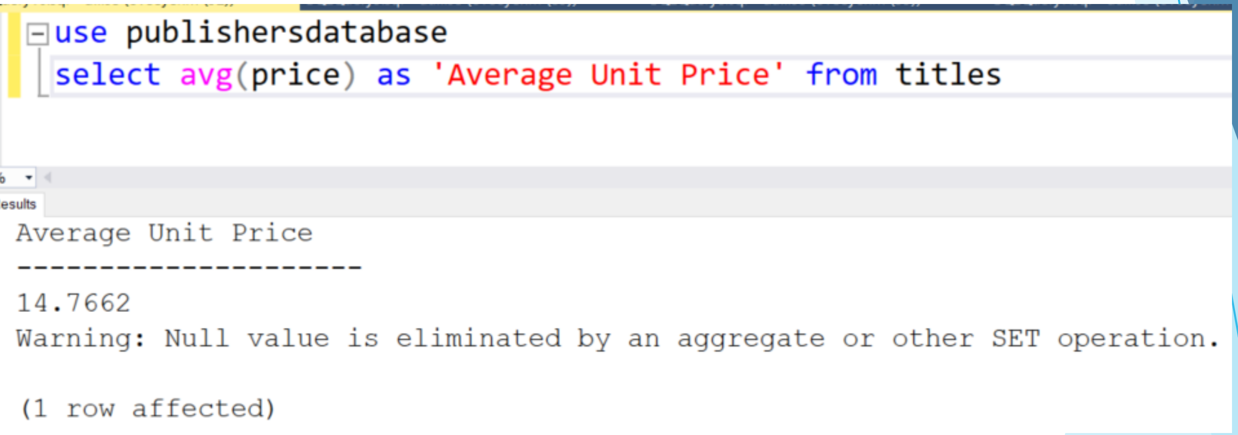
```
use publishersdatabase
select sum(ytd_sales) as 'Total Number of book items Sold' from titles
```

```
Total Number of book items Sold
-------------------------------
97446
Warning: Null value is eliminated by an aggregate or other SET operation.

(1 row affected)
```

Aggregate Example- Using Average

Let’s find the average price of available books



```
use publishersdatabase
select avg(price) as 'Average Unit Price' from titles
```

```
Average Unit Price
---------------------
14.7662
Warning: Null value is eliminated by an aggregate or other SET operation.

(1 row affected)
```

Exercise 1

- Write a query to find the total revenues we have obtained for all the sales during the year

Exercise 2

- Write a query to find the total revenues we have obtained for the sales of all business books during the year

Exercise 3

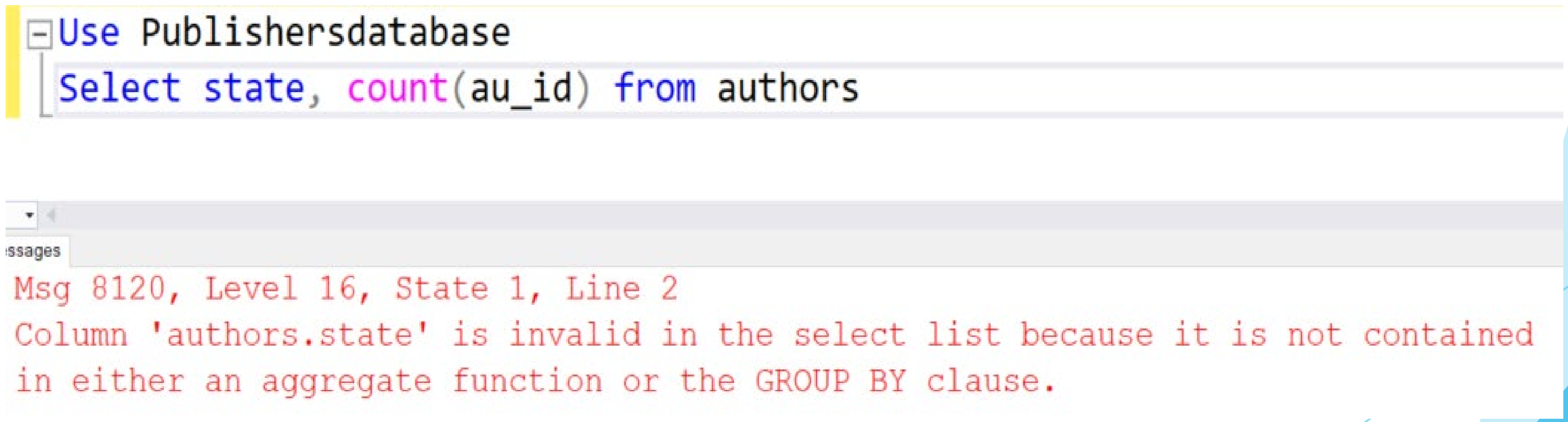
- Write a query to find the total revenues we have obtained during the year for the sales of all books that are either business books or have been published during the year 1991

Exercise 4

- Find the total number of authors who have a title in the publishers database

Using the **group by** Clause

- Consider that we want to obtain the total number of authors from each state.
- We want to obtain the name of the state and the number of authors.
- If we try the query below, we see that we obtain an error



```
Use Publishersdatabase
Select state, count(au_id) from authors
```

```
Msg 8120, Level 16, State 1, Line 2
Column 'authors.state' is invalid in the select list because it is not contained
in either an aggregate function or the GROUP BY clause.
```

group by

- The group by clause allows us to split up a set of data retrieved with a select statement into subsets of rows, but there are limitations
  - Once we have specified the column(s) on which to group our data, all other columns ***must*** be contained in aggregate functions.
    - This is required as the DBMS does not know which of the discrete values you wish to include in your result set.
    - As already shown, the reverse is also true. Once an aggregate function is used, a group by clause must also be employed if there are fields not using the aggregate function

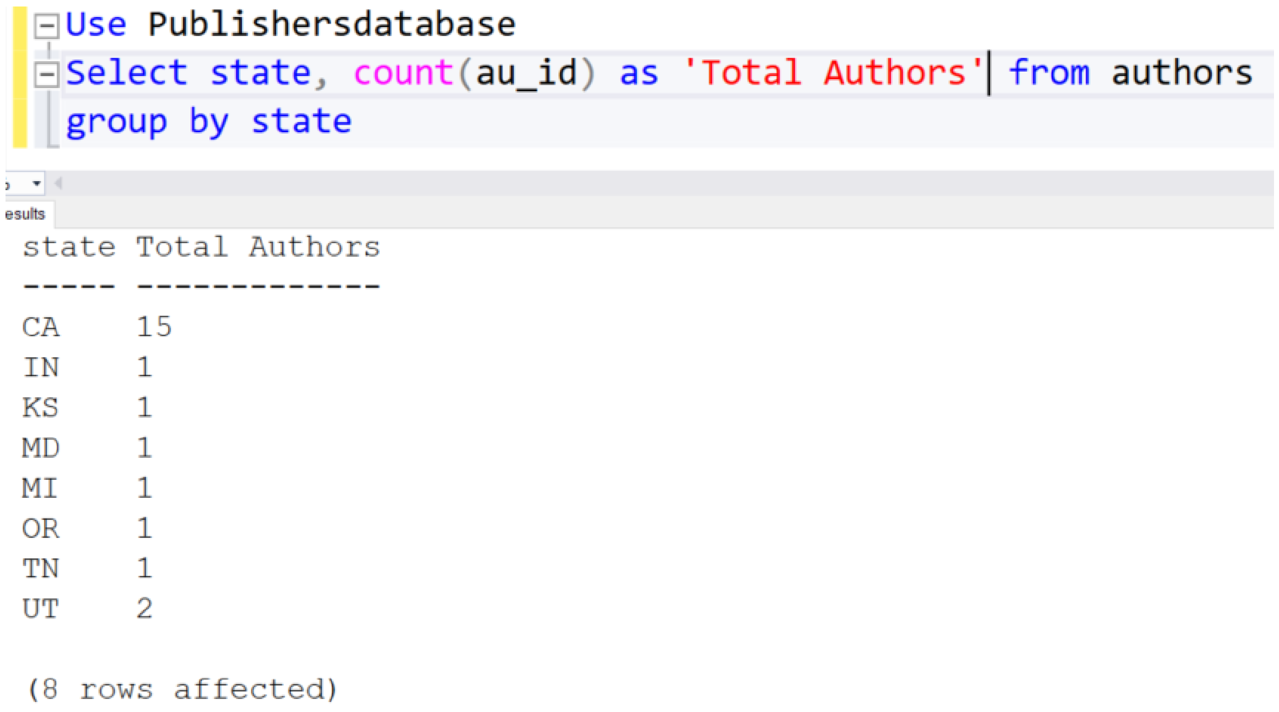
group by

- syntax

```
select      Column1, … , ColumnN,
            AggregateFunction1 (), … , AggregateFunctionN ()
from        Table
where       RowFilter1 [and/or] RowFilter2 [and/or] … RowFilterN
group by    Column1, … , ColumnN
```

group by Example

- Now we'll try our previous example again to obtain the count of authors from each state



```
Use Publishersdatabase
Select state, count(au_id) as 'Total Authors' from authors
group by state
```

```
state Total Authors
----- -------------
CA    15
IN    1
KS    1
MD    1
MI    1
OR    1
TN    1
UT    2

(8 rows affected)
```

Exercise 5

- Write a query that gives us the total number of publishers per country, ordered by country name in reverse alphabetical order

Filtering Grouped Data

- Next, we may wish to narrow our list down to a particular subset of our grouped data.
  - Consider: We want to have to total number of book copies sale per type of book where the total number of copies for that type is more that 15000

having

- The having clause is to grouped data what the where clause is to ungrouped data.
  - if data can be filtered at the ungrouped level, or row level, it should be for the sake of efficiency.
    - Why group a bunch of row data that does not satisfy a row level filter just to eliminate the group in which it appears?
    - Some database engines will be somewhat forgiving with mixing row and group filters, but not all of them. Best to learn the proper way.

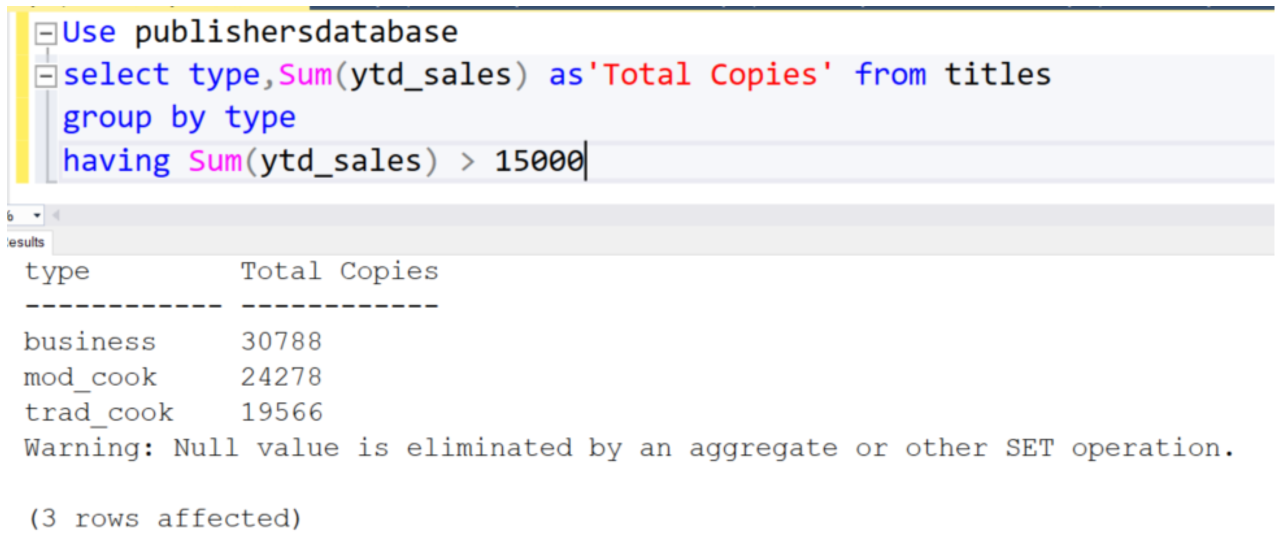
having

- syntax

```
select      Column1, … , ColumnN,
            Function1 (), … , FunctionN ()
from        Table
where       RowFilter1 [and/or] RowFilter2 [and/or] … RowFilterN
group by    Column1, … , ColumnN
having      GroupFilter1 [and/or] … [and/or] GroupFilterN
```

having Example

- Back to the example…



```
Use publishersdatabase
select type,Sum(ytd_sales) as'Total Copies' from titles
group by type
having Sum(ytd_sales) > 15000
```

```
type         Total Copies
------------ ------------
business     30788
mod_cook     24278
trad_cook    19566
Warning: Null value is eliminated by an aggregate or other SET operation.

(3 rows affected)
```

Order by

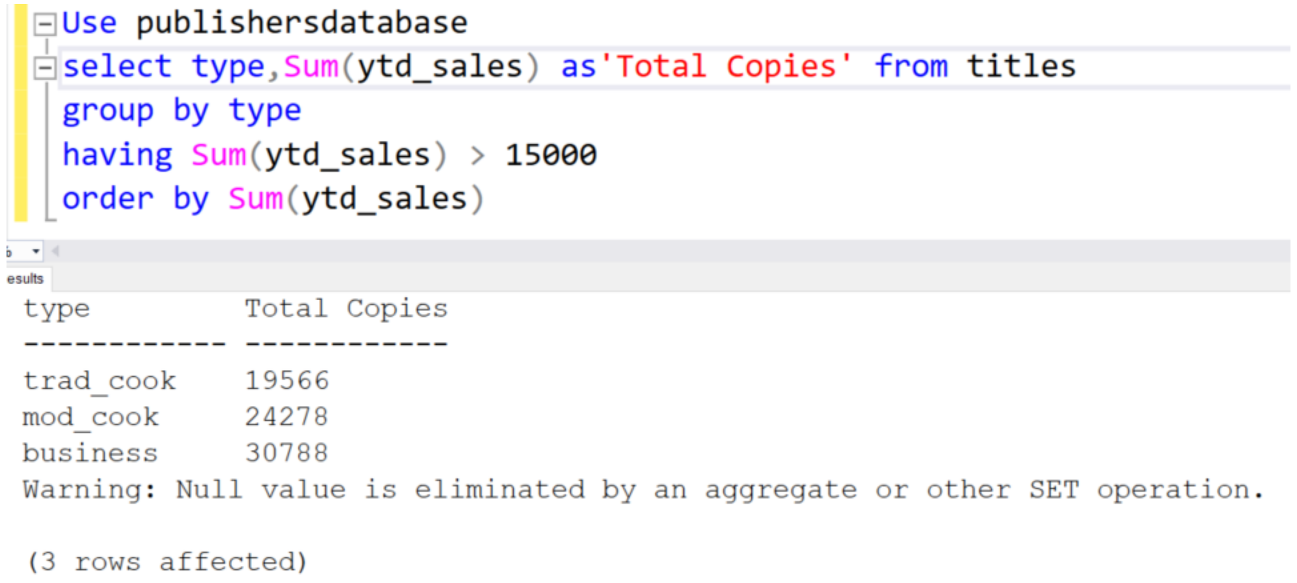
- Finally, we may still order our data using the order by clause as before.
- If a specific order is desired, the order by clause must be used – **do not rely on the group by to order data!**

order by

- syntax

```
select      Column1, … , ColumnN,
            Function1 (), … , FunctionN ()
from        Table
group by    Column1, … , ColumnN
having      GroupFilter1 [and/or] … [and/or] GroupFilterN
order by    ColumnA [asc/desc] , …, ColumnN [asc/desc]
```

order by



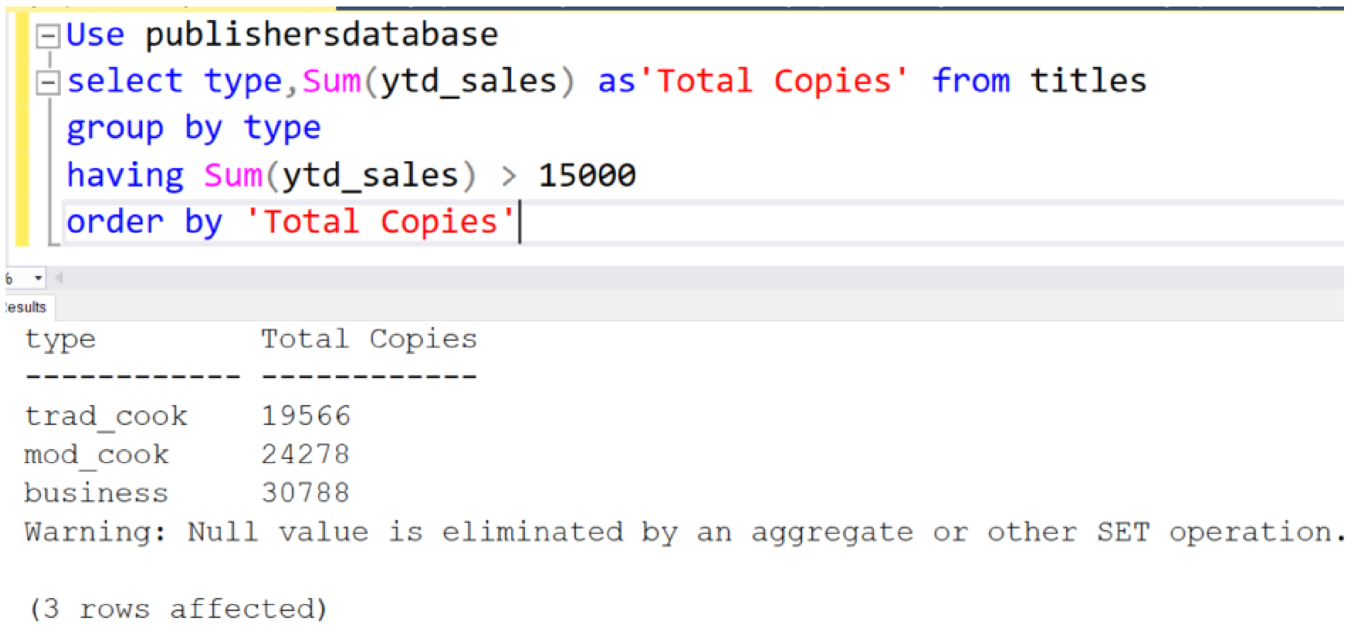
```
Use publishersdatabase
select type,Sum(ytd_sales) as'Total Copies' from titles
group by type
having Sum(ytd_sales) > 15000
order by Sum(ytd_sales)
```

```
type         Total Copies
------------ ------------
trad_cook    19566
mod_cook     24278
business     30788
Warning: Null value is eliminated by an aggregate or other SET operation.

(3 rows affected)
```

order by

- Remember that the order by clause can use an alias



```
Use publishersdatabase
select type,Sum(ytd_sales) as'Total Copies' from titles
group by type
having Sum(ytd_sales) > 15000
order by 'Total Copies'
```

```
type         Total Copies
------------ ------------
trad_cook    19566
mod_cook     24278
business     30788
Warning: Null value is eliminated by an aggregate or other SET operation.

(3 rows affected)
```