

SQL Week 3

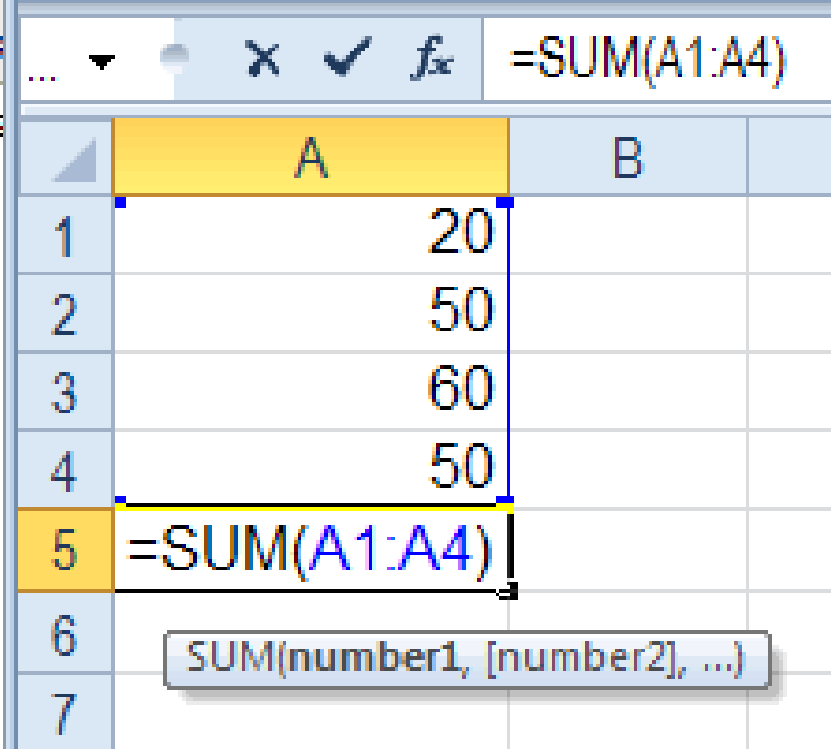
Topics List

- Aggregate Functions
- Group By clause
- Having clause
- Using Round()

SQL

Aggregate Functions

- An aggregate function allows you to perform a calculation on a set of values to return a single scalar value.



The screenshot shows an Excel spreadsheet with a formula bar at the top displaying `=SUM(A1:A4)`. The spreadsheet has two columns, A and B, and rows numbered 1 through 7. Column A contains the values 20, 50, 60, and 50 in rows 1 through 4. Row 5 is highlighted in yellow and contains the formula `=SUM(A1:A4)` in cell A5. A tooltip is visible below row 5, showing the syntax `SUM(number1, [number2], ...)`.

	A	B
1	20	
2	50	
3	60	
4	50	
5	=SUM(A1:A4)	
6		
7		

SQL

Aggregate Functions

- ▶ **AVG** – calculates the average of a set of values.
- ▶ **COUNT** – counts rows in a specified table or view.
- ▶ **MIN** – gets the minimum value in a set of values.
- ▶ **MAX** – gets the maximum value in a set of values.
- ▶ **SUM** – calculates the sum of values.

```
SELECT AVG(price)  
FROM book;
```

```
SELECT AVG(price) as 'Average Book Price'  
FROM book;
```

Topics List

- Aggregate Functions
- Group By clause
- Having clause
- Using Round()

SQL

Group By clause

- The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

```
SELECT county as County,  
       COUNT(*) as 'Number of Students'  
FROM student  
GROUP BY county;
```

County	Number of Students
Kilkenny	7
Tipperary	4
Waterford	5
Wexford	5

Topics List

- Aggregate Functions
- Group By clause
- Having clause
- Using Round()

SQL

Having clause

- In order to restrict rows returned with aggregate functions, you must use a *HAVING* clause.

```
SELECT county as County,  
       COUNT(*) as 'Number of Students'  
FROM student  
GROUP BY county  
HAVING COUNT(*) >=5;
```


Topics List

- Aggregate Functions
- Group By clause
- Having clause
- Using Round()

SQL

Using Round()

- To round a numeric value to a whole number (or to 2 decimal places for example), use *round*.

Round(12.75) returns 13

Round(12.751,2) returns 12.75

Round(12.755,2) returns 12.76