



✓ **Congratulations! You passed!**

TO PASS 80% or higher

Keep Learning

GRADE  
87.50%

## Weekly challenge 4

LATEST SUBMISSION GRADE

87.5%

1. A data analyst is working with a spreadsheet from a furniture company. To use the template for this spreadsheet, click the link below and select "Use Template."

1 / 1 point

Link to template: [Sample Transaction Table](#)

Or, if you don't have a Google account, download the file directly from the attachment below.

Sample Transaction Table - transactional-data-format-csv.csv

The analyst inputs a function to find the number of product prices that are less than \$150.00. Which formula will return that result?

- ☐ =SUMIF(G2:G30, "<150")
- ☐ =SUMIF(G2:G30, ">150")
- ☒ =COUNTIF(G2:G30, "<150")
- ☐ =COUNTIF(G2:G30, ">=150")

✓ **Correct**

The COUNTIF formula =COUNTIF(G2:G30, "<150") will allow the analyst to count all product price values in Column G that are less than \$150.

2. You are working in a spreadsheet and uses the SUMIF function in the formula below as part of your analysis.

1 / 1 point

=SUMIF(A1:A25, "<10", C1:C25)

Which part of this formula is the criteria or condition?

- ☐ =SUMIF
- ☐ C1:C25
- ☒ "<10"
- ☐ A1:A25

✓ **Correct**

The criteria or condition for this SUMIF formula is "<10". This means that if any values in the range A1 through A25 are less than 10, their corresponding values in the range C1 through C25 will be added together.

3. The following is a formula with the SUMPRODUCT function:

1 / 1 point

=SUMPRODUCT(A2:A10,B2:B10).

It will add the values from the first range (A2:A10) to the values from the second range (B2:B10). Then, the sums will be multiplied.

- ☐ True
- ☒ False

✓ **Correct**

The formula will multiply the first range of values (A2:A10) by the second range of values (B2:B10). Then, the products will be added together.

4. You create a pivot table in a spreadsheet containing movie data. To use the template for this spreadsheet, click the link below and select "Use Template."

0 / 1 point

Link to template: [Movie Data Project](#).

Or, if you don't have a Google account, download the file directly from the attachment below.

Movie Data Starter Project.xlsx

If you want to find out how many movies there are in each genre, which function in the Values menu would you use to summarize the data?

- ☐ SUM
- ☐ MAX
- ☐ COUNTA
- ☒ COUNT

✖ Incorrect

Review the spreadsheet and [the section on pivot tables](#) for a refresher.

5. A data analyst uses the following query to perform basic calculations on their data. The variables in the query have the following values: yes\_responses = 10, no\_responses = 12, total\_surveys = 22. What is the value of the Responses\_Per\_Survey variable?

1 / 1 point

```
SELECT
Yes_Responses,
No_Responses,
Total_Surveys,
(Yes_Responses + No_Responses) / Total_Surveys AS Responses_Per_Survey
FROM
Survey_1
```

- ☐ 11
- ☐ 22
- ☒ 1
- ☐ 44

✔ Correct

The value of the Responses\_Per\_Survey variable is 1. In this query, the analyst first calculates the sum of the "yes" and "no" responses, then divides the sum by the total surveys.

6. You are working with a database table that contains data about music. The table includes columns for *track\_id*, *track\_name*, *composer*, and *album\_id*. You are only interested in data about the classical musician Johann Sebastian Bach. You want to create new album IDs. You decide to multiply the current album IDs by 10 to create new album IDs, and use the AS command to store them in a new column called *new\_album\_id*.

1 / 1 point

Add a statement to your SQL query that calculates a new album Id for each track and stores it in a new column as *new\_album\_id*.

NOTE: The three dots (...) indicate where to add the statement.

```
1  SELECT
2  track_id,
3  track_name,
4  composer,
5  album_id,
6  album_id * 10 AS new_album_id
7  FROM
8  track
9  WHERE
10 composer = "Johann Sebastian Bach"
```

Run

Reset

| track_id | track_name   | composer              | album_id | new_album_id |
|----------|--|-----------------------|----------|--------------|
| 3407     | Concerto for 2 Violins in D Minor, BWV 1043: I. Vivace         | Johann Sebastian Bach | 276      | 2760         |
| 3408     | Aria Mit 30 Veränderungen, BWV 988 "Goldberg Variations": Aria | Johann Sebastian Bach | 277      | 2770         |
| 3409     | Suite for Solo Cello No. 1 in G Major, BWV 1007: I. Prélude    | Johann Sebastian Bach | 278      | 2780         |
| 3430     | Tocatta and Fugue in D Minor, BWV 565: I. Toccata              | Johann Sebastian Bach | 297      | 2970         |
| 3433     | Concerto No.2 in F Major, BWV1047, I. Allegro                  | Johann Sebastian Bach | 300      | 3000         |
| 3482     | Suite No. 3 in D, BWV 1068: III. Gavotte I & II                | Johann Sebastian Bach | 327      | 3270         |
| 3490     | Partita in E Major, BWV 1006A: I. Prelude                      | Johann Sebastian Bach | 335      | 3350         |

What is the new album ID for the track with Id number 3490?

- ☐ 2760
- ☐ 2970
- ☐ 3000
- ☒ 3350

✔ Correct

You add the statement `album_id * 10 AS new_album_id` to calculate a new album ID for each track and store it in a new column as *new\_album\_id*. The complete query is `SELECT track_id, track_name,`

```
composer, album_id, album_id * 10 AS new_album_id FROM track WHERE composer = "Johann Sebastian Bach". The AS command gives a temporary name to the new column.
```

The new Album Id for the track with Id number 3490 is 3350.

7. You are working with a database table that contains data about music. The table includes columns for *album\_id* and *milliseconds* (duration of the music tracks on each album). You want to find out the total duration for each album in milliseconds, and store the result in a new column named *total\_duration*.

1 / 1 point

You write the SQL query below. Add a GROUP BY clause that will group the data by album Id number.

```
1 SELECT
2 album_id,
3 SUM(milliseconds) AS total_duration
4 FROM
5 track
6 GROUP BY album_id ;
```

Run

Reset

| album_id | total_duration |
|----------|----------------|
| 1        | 2400415        |
| 2        | 342562         |
| 3        | 858088         |
| 4        | 2453259        |
| 5        | 4411709        |
| 6        | 3450925        |
| 7        | 3249365        |
| 8        | 2906926        |
| 9        | 2671407        |
| 10       | 3927713        |
| 11       | 3224237        |
| 12       | 1615722        |
| 13       | 2680524        |
| 14       | 4059919        |
| 15       | 1447755        |
| 16       | 2294801        |
| 17       | 2601921        |
| 18       | 3192389        |
| 19       | 3694022        |
| 20       | 2636849        |
| 21       | 3819382        |
| 22       | 959711         |
| 23       | 7875643        |
| 24       | 4238776        |
| 25       | 3016667        |

(Output limit exceeded, 25 of 347 total rows shown)

What is the total duration of the album with Id number 2?

- ☐ 959711
- ☐ 257252
- ☐ 858088
- ☒ 342562

✓ Correct

You add the clause **GROUP BY album\_id** to group the data by album Id number. The complete query is **SELECT album\_id, SUM(milliseconds) AS total\_duration FROM tracks GROUP BY album\_id**. The GROUP BY command groups rows that have the same values from a table into summary rows. GROUP BY is always placed as the last command in a SELECT-FROM-WHERE query.

The total duration of the album with ID number 2 is 342562 milliseconds.

8. You are working with a database table that contains invoice data. The table includes columns for *billing\_country* and *total*. You want to know the average total price for the invoices billed to the country of India. You decide to use the AVG function to find the average total, and use the AS command to store the result in a new column called *average\_total*.

1 / 1 point

Add a statement to your SQL query that calculates the average total and stores it in a new column as *average\_total*.

NOTE: The three dots (...) indicate where to add the statement.

```
1 SELECT
2 billing_country,
3 AVG(total) AS average_total
4 FROM
5 invoice
6 WHERE
7 billing_country = "India"
```

Run

Reset

| billing_country | average_total |
|-----------------|---------------|
| India           | 5.78          |

What is the average total for India?

- ☐ 5.64

☐ 6.02

☐ 5.37

☒ 5.78

✓ Correct

You add the statement `AVG(total) AS average_total` to calculate the average total and store it in a new column as `average_total`. The complete query is `SELECT billing_country, AVG(total) AS average_total FROM invoice WHERE billing_country = "India"`. The `AVG` function is an aggregate function that returns the average value of a group of values. The `AS` command gives a temporary name to the new column.

The average total for India is 5.78.