✅ **Congratulations! You passed!**

Grade received 100%   **To pass** 80% or higher

[Go to next item]

Weekly challenge 4

Latest Submission Grade 100%

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 File

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 use 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?

- ○ A1:A25
- ○ C1:C25
- ○ =SUMIF
- ◉ "<10"

✅ **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."   **1 / 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 File

If you want to summarize the data using the AVERAGE function in the Values menu, which spreadsheet columns could you add data from? Select all that apply.

- ☑ Box Office Revenue

✅ **Correct**
To summarize the data using the AVERAGE function, you could use the Budget column or the Box Office Revenue column. Both have numeric values that the AVERAGE function could calculate.

- ☑ Budget

✅ **Correct**
To summarize the data using the AVERAGE function, you could use the Budget column or the Box Office Revenue column. Both have numeric values that the AVERAGE function could calculate.

- ☐ Movie Title

☐ Movie Title

☐ Genre

5. A data analyst uses the following SQL query to perform basic calculations on their data. Which types of operators is the analyst using in this SQL query? Select all that apply.

1 / 1 point

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

☐ Multiplication

☑ Division

> ⊘ **Correct**
> The analyst is using the division operator (/) in this SQL query to divide the sum of "yes" and "no" responses by the total number of surveys.

☐ Subtraction

☑ Addition

> ⊘ **Correct**
> The analyst is using the addition operator (+) in this SQL query to calculate the sum of "yes" and "no" responses.

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.

```sql
1    SELECT
2    album_id*10 AS new_album_id
3
4    FROM
5    track
6    WHERE
7    composer = "Johann Sebastian Bach" AND track_id = 3490
```

[Run] [Reset]

```
+--------------+
| new_album_id |
+--------------+
|         3350 |
+--------------+
```

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

○ 2760

○ 3000

◉ 3350

○ 2970

> ⊘ **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 invoice data. The table includes columns for *customer_id* and *total* (total amount billed for each invoice). Some customers have multiple invoices. You want to find out the total amount billed to each customer, and store the result in a new column as *total_amount*.

1 / 1 point

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

```sql
1    SELECT
2    customer_id,
3    SUM(total) AS total_amount
4    FROM
5    invoice
6    WHERE
7    customer_id = 5
```

[Run] [Reset]

```
+-------------+--------------+
| customer_id | total_amount |
+-------------+--------------+
|           5 |        40.62 |
+-------------+--------------+
```

What is the total amount billed to the customer with Id number 5?

○ 49.62

○ 37.62

◉ 40.62

○ 39.62

> ⊘ **Correct**
> You add the clause `GROUP BY customer_id` to group the data by customer Id number. The complete query is `SELECT customer_id, SUM(total) AS total_amount FROM invoice GROUP BY`
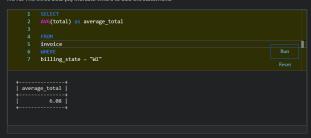
customer_id. The GROUP BY command groups rows that have the same values from a table into summary rows. GROUP BY is always placed as the last command in a SELECT-FROM-WHERE query.

The total amount billed to the customer with Id number 5 is 40.62.

8. You are working with a database table that contains invoice data. The table includes columns for *billing_state*, *billing_country*, and *total*. You want to know the average total price for the invoices billed to the state of Wisconsin. 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   AVG(total) as average_total
3
4   FROM
5   invoice
6   WHERE
7   billing_state = "WI"
```

Run

Reset

```
+---------------+
| average_total |
+---------------+
|          6.08 |
+---------------+
```

What is the average total for Wisconsin?

○ 5.37

◉ 6.08

○ 5.78

○ 5.54

✓ **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_state, billing_country, AVG(total) AS average_total FROM invoice WHERE billing_state = "WI"`. 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 Wisconsin is 6.08.