Back Weekly challenge 4
Graded Quiz • 40 min

Congratulations! You passed!

Grade received 87.50% | Latest Submission Grade 87.50% | To pass 80% or higher

Go to next item

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." 0 / 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

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." 0 / 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 transactions that include a brass-colored product. The syntax of which of the following formulas would return that result?

- ○ =SUMIF(F2:F30, "=brass")
- ○ =COUNTIF(F2:F30, "brass")
- ◉ =COUNTIF(E2:E30, "=brass")
- ○ =SUMIF(F2:F30, "brass only")

Incorrect
Review the section on IF functions and functions with conditions for a refresher.

Correct answer = option b) countif(F2:F30,"brass")

2. Fill in the blank: When you build a SUMIF or COUNTIF formula, the first part of the formula in parentheses is the _____. 1 / 1 point

- ○ operator
- ○ condition
- ○ criteria
- ◉ range

Correct
When you build a SUMIF or COUNTIF formula, the first part of the formula in parentheses is the range. The range of cells is then evaluated by the criteria or condition that you include in the formula.

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

Link to template: Retail Sales Data

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

Retail Sales Data - transactional-data-format-csv
CSV File

The analyst wants to figure out the value of all of the items in the spreadsheet. Which formula will calculate the total price of all of the items?

- ○ =SUM(C2:C21)
- ◉ =SUMPRODUCT(B2:B21,C2:C21)
- ○ =SUMIFS(C2:C21,B2:B21,"1",A2:A21,"_20")
- ○ =SUMIF(B2:B21, "=1")

Correct
To calculate the value of all items in Column C, use the SUMPRODUCT function and the range C2:C21.

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 figure out how much box office revenue each genre earned, which function in the Values menu would you use to summarize the data?

- ○ PRODUCT
- ◉ SUM
- ○ COUNTA
- ○ AVERAGE

Correct
You would use the SUM function to figure out how much box office revenue each genre earned. In the pivot table, the SUM function would add the total revenue separately for each genre.

5. Which part of the following SQL query enables an analyst to control the order of the calculations? 1 / 1 point

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

- ◉ (Yes_Responses + No_Responses)
- ○ FROM Survey_1
- ○ AS Responses_Per_Survey
- ○ Yes_responses

Correct
In a SQL query with calculations, an analyst includes parentheses to control the order of the calculations. The parentheses tell the server which calculation to complete first.

6. You are working with a database table that contains data about music. The table includes columns for *track_id*, *track_name* (name of the music track), *composer*, and *bytes* (digital storage size of the music track). You are only interested in data about the classical musician Johann Sebastian Bach. You want to know the size of each Bach track in kilobytes. You decide to divide bytes by 1000 to get the size in kilobytes, and use the AS command to store the result in a new column called *kilobytes*.

1 / 1 point

Add a statement to your SQL query that calculates the size in kilobytes for each track and stores it in a new column as *kilobytes*.

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

```
1   SELECT
2   track_id,
3   track_name,
4   composer,
5   bytes,
6   bytes/1000 as kilobytes
7   FROM
8   track
9   WHERE
10  composer = "Johann Sebastian Bach"
```

Run

Reset

track_name	composer	bytes	kilobytes
Concerto for 2 Violins in D Minor, BWV 1043: I. Vivace	Johann Sebastian Bach	3192890	3192
Aria Mit 30 Veränderungen, BWV 988 "Goldberg Variations": Aria	Johann Sebastian Bach	2081895	2081
Suite for Solo Cello No. 1 in G Major, BWV 1007: I. Prélude	Johann Sebastian Bach	2315495	2315
Toccata and Fugue in D Minor, BWV 565: I. Toccata	Johann Sebastian Bach	2649938	2649
Concerto No.2 in F Major, BWV1047, I. Allegro	Johann Sebastian Bach	5064553	5064
Suite No. 3 in D, BWV 1068: III. Gavotte I & II	Johann Sebastian Bach	3847164	3847
Partita in E Major, BWV 1006A: I. Prelude	Johann Sebastian Bach	4744929	4744

What is the size in kilobytes of the track with Id number 3407?

- ○ 5064
- ○ 4744
- ◉ 3192
- ○ 2315

Correct

You add the statement `bytes / 1000 AS kilobytes` to calculate the size in kilobytes for each track and store it in a new column as kilobytes. The complete query is `SELECT track_id, track_name, composer, bytes, bytes / 1000 AS kilobytes FROM track WHERE composer = "Johann Sebastian Bach"`. The AS command gives a temporary name to the new column.

The size of the track with Id number 3407 is 3192 kilobytes.

7. You are working with a database table that contains invoice data. The table includes columns for *invoice_id* and *quantity* (the number of purchases included in each line item of an invoice). Each invoice contains multiple line items. You want to find out the total number of purchases for each invoice, and store the result in a new column as *total_purchases*.

1 / 1 point

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

```
1  SELECT
2  invoice_id,
3  SUM(quantity) AS total_purchases
4  FROM
5  invoice_item
6  group by invoice_id
```

Run

Reset

```
+------------+-----------------+
| invoice_id | total_purchases |
+------------+-----------------+
|          1 |               2 |
|          2 |               4 |
|          3 |               6 |
|          4 |               9 |
|          5 |              14 |
|          6 |               1 |
|          7 |               2 |
|          8 |               2 |
|          9 |               4 |
|         10 |               6 |
|         11 |               9 |
|         12 |              14 |
|         13 |               1 |
|         14 |               2 |
|         15 |               2 |
|         16 |               4 |
```

What is the total number of purchases for the invoice with Id number 4?

- ○ 2
- ◉ 9
- ○ 4
- ○ 14

Correct

You add the clause `GROUP BY invoice_id` to group the data by customer Id number. The complete query is `SELECT invoice_id, SUM(quantity) AS total_purchases FROM invoice_item GROUP BY invoice_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 number of purchases for the invoice with Id number 4 is 9.

8. You are working with a database table that contains invoice data. The table includes columns for *billing_city*, *billing_country*, and *total*. You want to know the average total price for the invoices billed to the city of Vancouver. 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.

```
SELECT
billing_city,
billing_country,
avg(total) as average_total
FROM
invoice
WHERE
billing_city = "Vancouver"
```

Run

Reset

```
+--------------+-----------------+---------------+
| billing_city | billing_country | average_total |
+--------------+-----------------+---------------+
| Vancouver    | Canada          |          5.51 |
+--------------+-----------------+---------------+
```

What is the average total for Vancouver?

What is the average total for Vancouver?

- ○ 5.80
- ◉ 5.51
- ○ 6.23
- ○ 5.43

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_city, billing_country, AVG(total) AS average_total FROM invoice WHERE billing_city = "Vancouver"`. 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 Vancouver is 5.51.