

# Report

## Key points

- There is a space in the name of the **Video Views variable**, so we have mentioned it in commas( ` ` ) because it is a variable with a space in its name.

```
FROM youtube_statistics  
GROUP BY continents  
ORDER BY total_earnings DESC;
```

- In this code segment we can code the “**continent variable**” as 1. Because “Continent” is the first column in our new result.
- When we use **CASE** and **WHEN** keywords the rest of the code must end using the **END** keyword.
- **WITH** keyword to **CTE** (COMMON TABLE EXPRESSION)—

## CTE-1 COMMON TABLE EXPRESSION

- CTE is a temporary named result set that you can reference within a SELECT, INSERT, UPDATE, or DELETE statement.
- SYNTAX: **WITH** CTE\_NAME **AS** (CTE query) ...;
- The **WITH** clause can include one or more **CTEs** separated by commas.
- Enables user to more easily write and maintain complex queries via increased readability and simplification.
- **CTEs** can be a useful tool when you need to generate temporary result sets that can be accessed in a **SELECT, INSERT, UPDATE, DELETE**, or **MERGE** statement.
- A CTE must be followed by a single **SELECT, INSERT, UPDATE** or **DELETE** statement that references some or all the CTE columns.

```
ROUND(1.0 * SUM(CASE WHEN continents = 'Asia' THEN total_earnings ELSE 0 END)/  
NULLIF(SUM(total_earnings),0) * 100, 2) AS pct_earnings_Asia;
```

In this SQL code segment:

```
ROUND(1.0 * SUM(CASE WHEN continents = 'Asia' then total_earnings ELSE 0 END)/  
NULLIF(SUM(total_earnings),0) * 100, 2) AS pct_earnings_Asia;
```

It calculates the percentage of earnings from the 'Asia' continent out of the **total earnings**, and rounds the result to two decimal places. Let's break down this segment step by step:

1. **SUM(CASE WHEN continents = 'Asia' THEN total\_earnings ELSE 0 END):**
  - This part calculates the sum of **total\_earnings** for all rows where the '**continents**' column is equal to '**Asia**'. It effectively adds up the earnings for countries in the Asian continent. If the country is not in Asia, it contributes 0 to the sum.
2. **NULLIF(SUM(total\_earnings), 0):**
  - The **NULLIF** function is used here to avoid division by zero errors. It calculates the sum of all '**total\_earnings**'. If the total earnings sum is 0, it returns null to prevent division by zero.
3. **1.0 \* SUM(...) / NULLIF(...) \* 100:**
  - This part first multiplies the sum of earnings from Asia by 1.0 to ensure that the division will result in a floating-point number. Then, it divides the sum of earnings from Asia by the total earnings sum. This gives you the proportion of earnings from Asia out of the total earnings.
4. **ROUND(..., 2):**
  - Finally, it rounds the calculated percentage to two decimal places, making it more readable and presentable in the output.

So, the result of this code segment, **pct\_earnings\_Asia**, will be the percentage of earnings from the '**Asia**' continent out of the total earnings, rounded to two decimal places. This code segment is part of a larger SQL query that calculates similar percentages for different continents.