

🔗 Google Discussions



## Exam Professional Data Engineer All Questions

View all questions & answers for the Professional Data Engineer exam

[Go to Exam](#)

### 📄 EXAM PROFESSIONAL DATA ENGINEER TOPIC 1 QUESTION 8 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 8

Topic #: 1

[\[All Professional Data Engineer Questions\]](#)

You are building new real-time data warehouse for your company and will use Google BigQuery streaming inserts. There is no guarantee that data will only be sent in once but you do have a unique ID for each row of data and an event timestamp. You want to ensure that duplicates are not included while interactively querying data. Which query type should you use?

- A. Include ORDER BY DESK on timestamp column and LIMIT to 1.
- B. Use GROUP BY on the unique ID column and timestamp column and SUM on the values.
- C. Use the LAG window function with PARTITION by unique ID along with WHERE LAG IS NOT NULL.
- D. Use the ROW\_NUMBER window function with PARTITION by unique ID along with WHERE row equals 1.

[Show Suggested Answer](#)

by [deleted] at March 15, 2020, 8:44 a.m.

### Comments

Type your comment...



[Submit](#)

🗨️ **Ender\_H** Highly Voted 2 years, 7 months ago

I personally don't think any answer is correct,

D is the closest one but it's missing a "ORDER BY timestamp DESC" to ensure to get only the latest record based in the timestamp

   upvoted 12 times

  **ndimu** 5 months, 1 week ago

the idea is you can have multiple events occurring at the same time so the only way to distinct is id

   upvoted 1 times

  **Davijde13** 2 years, 3 months ago

The question mention only duplicated data and nothing about taking only the latest ones. Therefore I assume there is no need to always take the latest, we should ensure we take only one record for each ID.




   upvoted 6 times

  **daghayeghi** Highly Voted  4 years, 1 month ago

D:

[https://cloud.google.com/bigquery/streaming-data-into-bigquery#manually\\_removing\\_duplicates](https://cloud.google.com/bigquery/streaming-data-into-bigquery#manually_removing_duplicates)


   upvoted 9 times

  **willyunger** Most Recent  1 month, 2 weeks ago

**Selected Answer: D**

D is closest, as there will always be at least 1 row for each ID. Would have rather used SELECT DISTINCT.



   upvoted 1 times

  **RT\_G** 7 months, 1 week ago

**Selected Answer: D**

D ensures data is partitioned by the unique id and only one record is picked thereby ensuring results are de-duplicated

   upvoted 1 times

  **rtcpost** 7 months, 1 week ago

**Selected Answer: D**

This approach will assign a row number to each row within a unique ID partition, and by selecting only rows with a row number of 1, you will ensure that duplicates are excluded in your query results. It allows you to filter out redundant rows while retaining the latest or earliest records based on your timestamp column.

Options A, B, and C do not address the issue of duplicates effectively or interactively as they do not explicitly remove duplicates based on the unique ID and event timestamp.

   upvoted 2 times

  **Radhika7983** 7 months, 1 week ago

Correct answer is D. Group by column us used to check for the duplicates where you can have the count(\*) for each of the unique id column. If the count is greater than 1, we will know duplicate exists. The easiest way to remove duplicates while streaming inserts is to use row\_number. Use GROUP BY on the unique ID column and timestamp column and SUM on the values will not remove duplicates.

I also executed LAG function and LAG function will return NULL on unique id when no previous records with same unique id exist. Hence LAG is also not an option here.

   upvoted 8 times

  **MaxNRG** 7 months, 1 week ago



D is correct because it will just pick out a single row for each set of duplicates.

A is not correct because this will just return one row.

B is not correct because this doesn't get you the latest value, but will get you a sum of the same event over time which doesn't make too much sense if you have duplicates.

C is not correct because if you have events that are not duplicated, it will be excluded.

   upvoted 6 times

  **Zosby** 2 years, 2 months ago

Correct D



   upvoted 1 times

  **Morock** 2 years, 2 months ago

**Selected Answer: D**

Row number gives the unique number ranking based on target column.

   upvoted 3 times

  **odacir** 2 years, 4 months ago

**Selected Answer: D**

It's the only valid option, try it your self with examples in QB.

   upvoted 1 times

  **Mamta072** 2 years, 10 months ago

Ans is D as Row number is the clause to fetch unique record from duplicate

   upvoted 1 times

  **Arkon88** 3 years, 2 months ago

Answer: D

   upvoted 1 times

  **samdhimal** 3 years, 3 months ago


correct answer -> Use the ROW\_NUMBER window function with PARTITION by unique ID along with WHERE row equals 1.

You can use the ROW\_NUMBER() to turn non-unique rows into unique rows and then delete the duplicate rows.

Reference:

[https://www.mysqltutorial.org/mysql-window-functions/mysql-row\\_number-function/](https://www.mysqltutorial.org/mysql-window-functions/mysql-row_number-function/)

   upvoted 3 times

  **samdhimal** 7 months, 1 week ago

When you are using BigQuery streaming inserts, there is no guarantee that data will only be sent once. However, you can use the ROW\_NUMBER window function to ensure that duplicates are not included while interactively querying data. By using a PARTITION BY clause on the unique ID column, you can assign a unique number to each row within a result set, based on the order specified in the timestamp column. Then, a WHERE clause can be used to select only the row with the number 1. This will return the first row for each unique ID based on the timestamp column, which will ensure that duplicates are not included in your query results.

   upvoted 4 times

  **samdhimal** 2 years, 3 months ago

Option A is not recommended because it will only return the first row based on the timestamp column, it doesn't consider the unique ID, so you could have multiple rows with the same timestamp, and you will get one of them arbitrarily.

Option B is not recommended because it's used for aggregation, it doesn't return the first row for each unique ID based on the timestamp column.

Option C is not recommended because it's used for comparing rows, it doesn't return the first row for each unique ID based on the timestamp column.

   upvoted 2 times

  **nofaruccio** 3 years, 5 months ago

Sorry, but IMHO no response is correct, because, in addition to making the ID field unique, it occurs consider the record with most recent timestamp

   upvoted 1 times

  **anji007** 3 years, 6 months ago

Ans: D

   upvoted 1 times

  **lbhhoya82** 4 years, 1 month ago

Correct : D

   upvoted 1 times

  **sid091** 4 years, 2 months ago

D is correct

   upvoted 3 times

[Load full discussion...](#)



Platform

> Home

> All Exams

