

 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 15 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 15

Topic #: 1

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

You need to store and analyze social media postings in Google BigQuery at a rate of 10,000 messages per minute in near real-time. Initially, design the application to use streaming inserts for individual postings. Your application also performs data aggregations right after the streaming inserts. You discover that the queries after streaming inserts do not exhibit strong consistency, and reports from the queries might miss in-flight data. How can you adjust your application design?



- A. Re-write the application to load accumulated data every 2 minutes.
- B. Convert the streaming insert code to batch load for individual messages.
- C. Load the original message to Google Cloud SQL, and export the table every hour to BigQuery via streaming inserts.
- D. Estimate the average latency for data availability after streaming inserts, and always run queries after waiting twice as long.

[Show Suggested Answer](#)

by [deleted] at *March 16, 2020, 9:37 a.m.*

## Comments

[Submit](#)

  **noob\_master** Highly Voted  7 months, 1 week ago

**Selected Answer: D**

Answer: D. The only that describe a way to resolve the problem, with buffering the data.

(the question is possible old, the best approach would be Pub/Sub + Dataflow Streaming + Bigquery for streaming data instead near-real time)

   upvoted 8 times

  **MaxNRG** Highly Voted  3 years, 5 months ago

B. Streams data into BigQuery one record at a time without needing to run a load job:

<https://cloud.google.com/bigquery/docs/reference/rest/v2/tabledata/insertAll>

Instead of using a job to load data into BigQuery, you can choose to stream your data into BigQuery one record at a time by using the `tabledata.insertAll` method. This approach enables querying data without the delay of running a load job:

<https://cloud.google.com/bigquery/streaming-data-into-bigquery>

The BigQuery Storage Write API is a unified data-ingestion API for BigQuery. It combines the functionality of streaming ingestion and batch loading into a single high-performance API. You can use the Storage Write API to stream records into BigQuery that become available for query as they are written, or to batch process an arbitrarily large number of records and commit them in a single atomic operation.

Committed mode. Records are available for reading immediately as you write them to the stream. Use this mode for streaming workloads that need minimal read latency.

<https://cloud.google.com/bigquery/docs/write-api>

   upvoted 7 times

  **Abhi16820** 3 years, 5 months ago

IN THIS ALSO BIGQUERY HAS A BUFFER WHICH IT TAKES SLOWLY ANS INSERTS INTO REAL THING, WHAT YOU SAID IS HELPFULL IN REMOVING THE APPLICATION PART

   upvoted 1 times

  **MarcoDipa** 3 years, 4 months ago

could you please argue?

   upvoted 1 times

  **Rav761** Most Recent  4 months, 1 week ago

**Selected Answer: A**

To address the issue of strong consistency and ensure your reports do not miss in-flight data after streaming inserts, you should re-write the application to load accumulated data every 2 minutes (option A).

Here's why:

By accumulating and loading data in 2-minute intervals, you can balance between real-time data processing and ensuring data consistency.

This approach allows you to process the data in manageable batches, reducing the likelihood of inconsistencies that might occur with individual streaming inserts.

It maintains a near real-time analysis capability while allowing enough time for all in-flight data to be captured and accurately represented in your reports.

This adjustment should help improve the reliability of your data analysis and reporting.

   upvoted 2 times

  **imrane1995** 5 months, 1 week ago

**Selected Answer: A**

Accumulating data and loading it periodically (e.g., every 2 minutes) via batch inserts ensures strong consistency for queries. Batch loads in BigQuery allow you to avoid the latency issues inherent to streaming inserts and guarantee data availability for queries.

   upvoted 1 times

  **GHill1982** 6 months, 2 weeks ago

**Selected Answer: A**

For maintaining data consistency while handling high throughput streaming inserts and subsequent aggregations in Google BigQuery, the best approach is to re-write the application to load accumulated data every 2 minutes.

   upvoted 1 times

  **fire558787** 7 months, 1 week ago

"D" seems to use the typical approximate terminology of a wrong answer. "estimate the time" (how do you do that? do you do that over different times of the day?) and "wait twice as long" (who tells you that there are not a lot of cases when lag is twice as long?). Instead, "A" seems good. You don't need to show the exact results, but an approximation thereof, but you still want consistency. So an aggregation of the data every 2 minutes is a viable thing.

   upvoted 5 times

upvoted 3 times

**Parth\_P** 7 months, 1 week ago

**Selected Answer: D**

D is correct. The problem requirement is doing analytics on real-time data. You cannot do batch processing because the business requires it to be real-time even if it makes your job simpler, so B is incorrect. Other options are not streaming.

upvoted 2 times

**jkhong** 7 months, 1 week ago

**Selected Answer: D**

There are assumptions over the quality of data acceptable. If slight variations of the analytics against actual can be accepted, then D would be a good choice.

Many people chose B, but this also requires some form of waiting for the late data to arrive.

I think a combination of D and B can be applied, but for an initial fix, delaying the aggregation queries with D seems to make more sense. If the variance is small and the some late data leakage is acceptable, and we can remain as D.

If problems arise, we can always proceed to attempt B

upvoted 2 times

**korntewin** 7 months, 1 week ago

**Selected Answer: D**

The streaming mode may be in pending mode or buffered mode where the streaming data is not immediately available before committing or flushing. Thus, we need to wait before the data will be available. Or else we need to switch to committed mode (which is not present in the choices).

upvoted 2 times

**musumusu** 7 months, 1 week ago

Answer: D

What to learn or look for

1. In-Flight data = (Real Time data, i.e still in streaming pipeline and not landed in BigQuery)
2. Dataflow (assume in best case) streaming pipeline is running to send data to Bigquery.

Why not option B: change streaming to batch upload is not business requirement, we have to stuck to streaming and real time analysis.

Option D: make bigquery run after waiting for sometime (twice here), How will you do it?

- there is not setting in BQ to do it, right!. So, adjust it in your pipeline (dataflow)
- For example, add Fixed window, and you want to execute aggregation query after 2 min.

Code

```
```pipeline.apply(...)  
  .apply(Window.<TableRow>into(FixedWindows.of(Duration.standardMinutes(2))))  
  .apply(BigQueryIO.writeTableRows()  
    .to("my_dataset.my_table")  
  )  
```
```

upvoted 5 times

**philli1011** 1 year, 3 months ago

Answer: D

I agree with the first part of the D answer, but for the second part, I don't know how they came about the 2 mins, is it from a calculation?

upvoted 1 times

**imran79** 1 year, 7 months ago

A. Re-write the application to load accumulated data every 2 minutes.

By accumulating data and performing a batch load every 2 minutes, you can reduce the potential inconsistency caused by streaming inserts. While this introduces a slight delay, it provides a more consistent approach than streaming each individual message. This method can still meet the near real-time requirement, and the slight delay is often acceptable in scenarios where data consistency is paramount.

upvoted 3 times

**Nirca** 1 year, 7 months ago

**Selected Answer: B**

BBBBB is the only option

upvoted 1 times

**ckanaar** 1 year, 7 months ago

I'd argue that this question became outdated with the introduction of the BigQuery Storage Write API:  
<https://cloud.google.com/bigquery/docs/write-api>

upvoted 4 times

  **axantroff** 1 year, 6 months ago

Good point

   upvoted 1 times

  **klughund** 1 year, 9 months ago

Streaming inserts in BigQuery are not immediately available to be queried, which is causing the weak consistency you're observing. A better approach is to batch the data and load it at regular intervals. Loading the data every two minutes is still relatively real-time, and it should help solve the consistency problem.

Answer A.

   upvoted 3 times

  **NeoNitin** 1 year, 9 months ago

All the options aim to address the challenge of strong consistency in the data and potential missing data that may occur with streaming inserts. Each approach has its pros and cons, so the best choice depends on the specific needs and requirements of the application. It's like having different strategies for keeping track of all the fun things the kids do and say on the playground, making sure nothing gets left behind!

   upvoted 1 times

  **WillemHendr** 1 year, 11 months ago

Streaming Inserts is marked as Legacy now.

<https://cloud.google.com/bigquery/docs/streaming-data-into-bigquery#dataavailability>

The documentation is hinting on it can take up to 90 minutes to process the buffered data.

This question is testing if you are aware of the possible long times the buffer can build up.

   upvoted 3 times

[Load full discussion...](#)



## Platform

> [Home](#)

> [Examtopics PRO](#)

> [All Exams](#)

> [Training Courses](#)

