

[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 21 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 21

Topic #: 1

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

Your company uses a proprietary system to send inventory data every 6 hours to a data ingestion service in the cloud. Transmitted data includes a payload of several fields and the timestamp of the transmission. If there are any concerns about a transmission, the system re-transmits the data. How should you deduplicate the data most efficiently?

- A. Assign global unique identifiers (GUID) to each data entry.
- B. Compute the hash value of each data entry, and compare it with all historical data.
- C. Store each data entry as the primary key in a separate database and apply an index.
- D. Maintain a database table to store the hash value and other metadata for each data entry.

[Show Suggested Answer](#)

by [deleted] at March 18, 2020, 4:13 p.m.

Comments

Type your comment...

[Submit](#)

dg63 [Highly Voted](#) 4 years, 10 months ago

The best answer is "A".

Answer "D" is not as efficient or error-proof due to two reasons

However, B is not as efficient or error proof due to two reasons:

1. You need to calculate hash at sender as well as at receiver end to do the comparison. Waste of computing power.
 2. Even if we discount the computing power, we should note that the system is sending inventory information. Two messages sent at different can denote same inventory level (and thus have same hash). Adding sender time stamp to hash will defeat the purpose of using hash as now retried messages will have different timestamp and a different hash.
- if timestamp is used as message creation timestamp than that can also be used as a UUID.

   upvoted 67 times

  **emmylou** 1 year, 7 months ago

If you add a unique ID aren't you by definition not getting a duplicate record. Honestly I hate all these answers.

   upvoted 4 times



  **billalltf** 11 months, 3 weeks ago

You can add a function or condition that verifies if the global unique id already exists or just do a deduplication later

   upvoted 1 times

  **retax** 4 years, 6 months ago

If the goal is to ensure at least ONE of each pair of entries is inserted into the db, then how is assigning a GUID to each entry resolving the duplicates? Keep in mind if the 1st entry fails, then hopefully the 2nd (duplicate) is successful.

   upvoted 13 times

  **ralf_cc** 3 years, 10 months ago

A - In D, same message with different timestamp will have different hash, though the message content is the same.

   upvoted 12 times

  **MaxNRG** 3 years, 3 months ago

agreed, the key here is "payload of several fields and the timestamp"

   upvoted 2 times

  **MaxNRG** 3 years, 3 months ago

"payload of several fields and the timestamp of the transmission"

   upvoted 2 times

  **BigDataBB** 3 years, 3 months ago

Hi Max, I also think that the hash value would be wrong because the timestamp is part of payload and is not written that the hash value is generated without the ts; but it also not written if GUID is linked or not with sending. Often this is a point where the answer is vague. Because don't specify if the GUID is related to the data or to the send.

  upvoted 1 times

  **omakin** 3 years, 9 months ago

Strong Answer is A - in another question on the gcp sample questions: the correct answer to that particular question was "You are building a new real-time data warehouse for your company and will use 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?"


This means you need a "uniqueid" and timestamps to properly dedupe a data.

   upvoted 8 times

  **Tanzu** 3 years, 3 months ago

U need a uniqueid but in this scenario, there is none. So u have to calculate by hashing w/ some of the fields in the dataset.

A is assigning guid in processing side will not solve the issue. Cause u will assign diff. ids...

   upvoted 1 times

  **cetanx** 2 years, 3 months ago

Answer - D

Key statement is "Transmitted data includes a payload of several fields and the timestamp of the transmission."

So the timestamp is appended to message while sending, in other words that field is subject to change if message is retransmitted. However, adding a GUID doesn't help much because if message is transmitted twice you will have different GUID for both messages but they will be the same/duplicate data.

You can simply calculate a hash based on not all data but from a select of columns (with the payload of several fields AND definitely by excluding the timestamp). By doing so, you can assure a different hash for each message.

  upvoted 5 times

  **MarcoDipa** 3 years, 4 months ago

Answer is D. Using Hash values we can remove duplicate values from a database. Hash values will be same for duplicate data and thus can be easily rejected. Obviously you won't check hash for timestamp.
D is better than B because maintaining a different table will reduce cost for hash computation for all historical data

   upvoted 5 times

  **Mathew106** 1 year, 9 months ago

Why can't it be A, where the GUID is a hash value? Why do we need to store the hash with the metadata in a separate database to do the deduplication?

   upvoted 1 times

  **[Removed]** Highly Voted  5 years, 1 month ago

Answer: D

Description: Using Hash values we can remove duplicate values from a database. Hash values will be same for duplicate data and thus can be easily rejected.

   upvoted 24 times

  **stefanop** 3 years ago

Hash values for same data will be the same, but in this case data contains also the timestamp

   upvoted 2 times

  **DGames** 2 years, 4 months ago

While calculating Hash value we exclude the timestamp.



   upvoted 1 times

  **fassil** Most Recent  3 weeks, 4 days ago

Selected Answer: D

A is incorrect. how can you find duplicates if you assign a unique id to every record? The answer is D.

   upvoted 1 times

  **Mo5454545454** 1 month ago

Selected Answer: D

The most efficient way to deduplicate your inventory data would be:

D. Maintain a database table to store the hash value and other metadata for each data entry.

This approach is optimal because:

It creates a lightweight reference table that stores just the hash values and essential metadata (like timestamps) rather than the full payload data

Hash values can be quickly compared to identify duplicates without expensive full-data comparisons

The metadata can help with auditing and troubleshooting transmission issues

This solution scales well as your data volume grows

Option A (using GUIDs) doesn't address the retransmission scenario well, as new GUIDs might be generated each time.

Option B requires comparing against all historical data, which becomes increasingly inefficient over time. Option C creates unnecessary storage overhead by using entire data entries as primary keys when only a hash value is needed for comparison.

   upvoted 1 times

  **Parandhaman_Margan** 1 month, 3 weeks ago

Selected Answer: D

Deduplicate data with retransmissions. Use a database table with hash

   upvoted 2 times

  **Abizi** 2 months ago

Selected Answer: A

most obvious answer

   upvoted 1 times

  **Rav761** 4 months, 1 week ago

Selected Answer: D

Option D: Maintain a database table to store the hash value and other metadata for each data entry.

This approach is efficient and scalable. By storing a computed hash value (as a compact representation of the data) along with metadata, deduplication can be performed by comparing new entries with the stored hashes. This minimizes storage requirements and improves lookup efficiency.

   upvoted 1 times

  **vbrege** 10 months, 3 weeks ago

1. My original vote was 'B'. I chose it over 'D' because option 'D' does not explicitly say anything about how that table will be

used for deduplication. In hindsight, explicit usage of table should not be given much weightage so after review and seeing other comments, I thought of 'D' as the correct answer.

2. Now looking more clearly at option 'D' (and 'B' also), it's a little ambiguous of what keys will be used to create the hash. So, if you use the payload PLUS the timestamp, the hash is of no use. This is a little confusing

3. Finally, although I never thought this is the right option, 'A' seems to be the correct option. The GUID is created at Data entry, NOT at the transmission stage. So, the GUID should be representative of the payload only and NOT the timestamp which will make it unique per payload, not per transmission of the same payload. So, in the end, I feel like 'A' is the correct choice.

   upvoted 1 times

  **TVH_Data_Engineer** 1 year, 4 months ago

Selected Answer: D

To deduplicate the data most efficiently, especially in a cloud environment where the data is sent periodically and re-transmissions can occur, the recommended approach would be:

D. Maintain a database table to store the hash value and other metadata for each data entry.

This approach allows you to quickly check if an incoming data entry is a duplicate by comparing hash values, which is much faster than comparing all fields of a data entry. The metadata, which includes the timestamp and possibly other relevant information, can help resolve any ambiguities that may arise if the hash function ever produces collisions.

   upvoted 1 times

  **JustQ** 1 year, 5 months ago

B. Compute the hash value of each data entry, and compare it with all historical data.

Explanation:

Efficiency: Hashing is a fast and efficient operation, and comparing hash values is generally quicker than comparing the entire payload or using other methods.

Space Efficiency: Storing hash values requires less storage space compared to storing entire payloads or using global unique identifiers (GUIDs).

Deduplication: By computing the hash value of each data entry and comparing it with historical data, you can easily identify duplicate transmissions. If the hash value matches an existing one, it indicates that the payload is the same.

   upvoted 3 times

  **steghe** 1 year, 5 months ago

I thought the answer was A 'cos it's more efficient. But I read the answer with more attention: GUID is given "at each data entry". It's not said that GUID was given from publisher. If GUID is given in data entry (subscriber), two equal messages can have different GUID.




D is not complete 'cos it's not so precise about hash field that are used.

I'm in doubt on this answer :-)

   upvoted 2 times

  **Lestrang** 1 year, 1 month ago

Data entry means record, it is not an action. that means that each record will have a unique id. so assuming our sink will not accept duplicates based on a key, the GUID will work.

   upvoted 1 times




  **rocky48** 1 year, 6 months ago

Selected Answer: A

Answer : A

"D" is not as efficient or error-proof due to two reasons

1. You need to calculate hash at sender as well as at receiver end to do the comparison. Waste of computing power.
2. Even if we discount the computing power, we should note that the system is sending inventory information. Two messages sent at different can denote same inventory level (and thus have same hash). Adding sender time stamp to hash will defeat the purpose of using hash as now retried messages will have different timestamp and a different hash.
if timestamp is used as message creation timestamp than that can also be used as a UUID.

   upvoted 1 times

  **rtcpoost** 1 year, 6 months ago

Selected Answer: D

D. Maintain a database table to store the hash value and other metadata for each data entry.

Storing a database table with hash values and metadata is an efficient way to deduplicate data. When new data is transmitted, you can calculate the hash of the payload and check whether it already exists in the database. This approach allows for efficient duplicate detection without the need to compare the new data with all historical data. It's a common and scalable technique used to ensure data consistency and avoid processing the same data multiple times.

Options A (assigning GUIDs to each data entry) and C (storing each data entry as the primary key) can work, but they might be less efficient than using hash values when dealing with a large volume of data. Option B (computing the hash value of each data entry and comparing it with all historical data) can be computationally expensive and slow, especially if there's a significant amount of historical data to compare against. Storing hash values in a table allows for fast and efficient deduplication.

👍 🔄 🚩 upvoted 1 times

🗨️ 👤 **alihabib** 1 year, 9 months ago

Why not D ? Generate a Hash for payload entry and maintain the value as metadata. Do the validation check on Dataflow..... A GUID will generate 2 different entries for same payload entry, it will not tackle duplication check

👍 🔄 🚩 upvoted 2 times

🗨️ 👤 **Hungry_guy** 1 year, 9 months ago

Answer is B - although the time stamp is diff for each transmission - the hash value is computed for the payload, not for the timestamp - which is just an added field for transmission. So, has val remains the same for all transmissions of the same data - which is what we can use for comparision.

So, much more efficient to just directly compare the hash values with the historical data - to check and remove duplicates - instead of again wasting space storing stuff - in option D

👍 🔄 🚩 upvoted 3 times

🗨️ 👤 **Mark_86** 1 year, 9 months ago

Selected Answer: D

This question is formulated very badly.

From the way that A is formulated, you would not deduplicate but rather the duplicates would have the same GUID.

Then we have D, which is storing the information (assuming the hash is created without the timestamp). B is doing it right away. D only alludes to the actual deduplication. But it would be more efficient.

👍 🔄 🚩 upvoted 2 times

🗨️ 👤 **boca_2022** 2 years ago

Selected Answer: A

A is best choice. D doesn't make sense.

👍 🔄 🚩 upvoted 2 times

🗨️ 👤 **FP77** 1 year, 8 months ago

A is incorrect. how can you find duplicates if you assign a unique id to every record? The answer is either B or D. I first selected B, but reading through the answers D may be better.

👍 🔄 🚩 upvoted 2 times

[Load full discussion...](#)



Platform

> [Home](#)

> [Examtopics PRO](#)

> [All Exams](#)

> [Training Courses](#)



