

🔗 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 119 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 119

Topic #: 1

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

You operate a database that stores stock trades and an application that retrieves average stock price for a given company over an adjustable window of time. The data is stored in Cloud Bigtable where the datetime of the stock trade is the beginning of the row key. Your application has thousands of concurrent users, and you notice that performance is starting to degrade as more stocks are added. What should you do to improve the performance of your application?

- A. Change the row key syntax in your Cloud Bigtable table to begin with the stock symbol.
- B. Change the row key syntax in your Cloud Bigtable table to begin with a random number per second.
- C. Change the data pipeline to use BigQuery for storing stock trades, and update your application.
- D. Use Cloud Dataflow to write a summary of each day's stock trades to an Avro file on Cloud Storage. Update your application to read from Cloud Storage and Cloud Bigtable to compute the responses.

[Show Suggested Answer](#)

by [deleted] at *March 22, 2020, 12:41 p.m.*

Comments

Type your comment...

[Submit](#)

🗄️ 👤 [Removed] Highly Voted 5 years, 1 month ago

Answer: A

Description: Timestamp at starting of rowkey causes bottleneck issues

👍 ↩️ 🚩 upvoted 41 times

🗄️ 👤 kichukonr Highly Voted 5 years ago

Stock symbol will be similar for most of the records, so it's better to start with random number.. Answer should be B

👍 ↩️ 🚩 upvoted 13 times

🗄️ 👤 Abhi16820 3 years, 5 months ago

You never use something called random number in bigtable rowkey because it gives you no use in querying possibilities, since we can't run sql queries in bigtable we should not randomise rowkeys in bigtable. Don't confuse the above point with the hotspot logic, both are different if you think so.

And another thing is, what you said can be good choice if we are using cloud spanner and trying to comeup with primary key situation, since there we can always run sql query.

I think you got the point now.

👍 ↩️ 🚩 upvoted 13 times

🗄️ 👤 taepyung 5 years ago

I agree with u

👍 ↩️ 🚩 upvoted 3 times

🗄️ 👤 karthik89 4 years, 2 months ago

it can start with stock symbol concated with timestamp can be a good row key design

👍 ↩️ 🚩 upvoted 6 times

🗄️ 👤 Yonghai 3 years, 4 months ago

for a given company, the data poits starts with the same stock symbol. The dataset is not distrubuted. It is not a good option.

👍 ↩️ 🚩 upvoted 3 times

🗄️ 👤 clouditis Most Recent 4 months, 3 weeks ago

Selected Answer: B

the most plausible option to pick here is B, A can introduce hot-spotting

👍 ↩️ 🚩 upvoted 1 times

🗄️ 👤 Vineet_Mor 8 months, 2 weeks ago

B is correct, By introducing a random number or a hash at the beginning of the row key, you distribute the writes and reads more evenly across the Bigtable cluster, thereby improving performance under heavy load.

WHY NOT A?

This might still cause hotspots if certain stocks are more popular than others. It could lead to uneven load distribution, which wouldn't solve the performance degradation problem.

👍 ↩️ 🚩 upvoted 2 times

🗄️ 👤 Sofiia98 1 year, 3 months ago

Selected Answer: A

Answer is A.

👍 ↩️ 🚩 upvoted 2 times

🗄️ 👤 musumusu 2 years, 2 months ago

Answer A:

Trick to remember: Row-key adjustment always be like in decending order.

#<<Least value>>#<<Lesser value>>

For example:

1. #<<Earth>>#<<continents>>#<<countries>>#<<cities>> and so on..

2. #<<Stock>>#<<users>>#timestamp..

in 99% cases timestamp will be in the end, as its smallest division...

👍 ↩️ 🚩 upvoted 12 times

🗄️ 👤 piyush7777 1 year, 8 months ago

Awesome!

👍 ↩️ 🚩 upvoted 1 times

🗄️ 👤 zelick 2 years, 5 months ago

Selected Answer: A

A is the answer.

<https://cloud.google.com/bigtable/docs/schema-design#row-keys>


It's important to create a row key that makes it possible to retrieve a well-defined range of rows. Otherwise, your query requires a table scan, which is much slower than retrieving specific rows.

<https://cloud.google.com/bigtable/docs/schema-design#row-keys-avoid>

Some types of row keys can make it difficult to query your data, and some result in poor performance. This section describes some types of row keys that you should avoid using in Bigtable.

- Row keys that start with a timestamp. This pattern causes sequential writes to be pushed onto a single node, creating a hotspot. If you put a timestamp in a row key, precede it with a high-cardinality value like a user ID to avoid hotspots.

   upvoted 6 times

  **AzureDP900** 2 years, 4 months ago

I agree with you . A is right

   upvoted 1 times

  **MaxNRG** 3 years, 4 months ago

Selected Answer: A

A: https://cloud.google.com/bigtable/docs/schema-design-time-series#prefer_rows_to_column_versions

   upvoted 4 times

  **JG123** 3 years, 5 months ago

Correct: A

   upvoted 1 times

  **JayZeeLee** 3 years, 5 months ago

A and B would both work, since both would distribute the work. This question is not framed properly.

   upvoted 1 times

  **sumanshu** 3 years, 10 months ago

Vote for A

   upvoted 3 times




  **Jay3244** 4 years, 2 months ago

Option A.

Below document explains

Having EXCHANGE and SYMBOL in the leading positions in the row key will naturally distribute activity.

<https://cloud.google.com/bigtable/docs/schema-design-time-series>

   upvoted 5 times

  **arghya13** 4 years, 5 months ago

I think A

   upvoted 2 times

  **kavs** 4 years, 5 months ago


Catch here is current Rowkey starts with timestamp which should not be in the starting or end position so symbols should be prefixed before timestamp

   upvoted 1 times

  **Cloud_Enthusiast** 4 years, 5 months ago

A is correct..A Good ROW KEY has to be an ID followed by timestamp. Stock symbol in this case works as an ID

   upvoted 6 times

  **kino2020** 4 years, 7 months ago

A.

You can find an example in Google's introductory guide.

https://cloud.google.com/bigtable/docs/schema-design-time-series?hl=ja#financial_market_data

   upvoted 2 times

  **Diqtator** 4 years, 7 months ago

I think A would be best practice. Adding random numbers as start of rowkey doesn't help with troubleshooting

   upvoted 3 times

[Load full discussion...](#)

Platform

> Home

> Examtopics PRO

> All Exams

> Training Courses

