

🔗 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 183 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 183

Topic #: 1

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

You are using Bigtable to persist and serve stock market data for each of the major indices. To serve the trading application, you need to access only the most recent stock prices that are streaming in. How should you design your row key and tables to ensure that you can access the data with the simplest query?

- A. Create one unique table for all of the indices, and then use the index and timestamp as the row key design.
- B. Create one unique table for all of the indices, and then use a reverse timestamp as the row key design.
- C. For each index, have a separate table and use a timestamp as the row key design.
- D. For each index, have a separate table and use a reverse timestamp as the row key design.

[Show Suggested Answer](#)

by [AWSandeep](#) at *Sept. 2, 2022, 10:25 p.m.*

Comments

Type your comment...

[Submit](#)

👤 [John_Pongthorn](#) [Highly Voted](#) 🍎 2 years, 7 months ago



This is special case , please Take a look carefully the below link and read at last paragraph at the bottom of this comment, let everyone share idea. We will go with B. C.

everyone share idea, we will go with B, C
<https://cloud.google.com/bigtable/docs/schema-design#time-based>

Don't use a timestamp by itself or at the beginning of a row key, because this will cause sequential writes to be pushed onto a single node, creating a hotspot.

If you usually retrieve the most recent records first, you can use a reversed timestamp in the row key by subtracting the timestamp from your programming language's maximum value for long integers (in Java, `java.lang.Long.MAX_VALUE`). With a reversed timestamp, the records will be ordered from most recent to least recent.

   upvoted 17 times

  **Ryannn23** 3 months ago

According to the link you provided:

If you usually retrieve the most recent records first in your queries, a pattern to consider is using reversed timestamps in the row key. This pattern causes rows to be ordered from most recent to least recent, so more recent data is earlier in the table.

----- READ CAREFULLY -----

As with any timestamp, avoid starting a row key with a reversed timestamp so that you don't cause hotspots.

You can get a reversed timestamp by subtracting the timestamp from your programming language's maximum value for long integers (in Java, `java.lang.Long.MAX_VALUE`).

Hence, starting row key with timestamp should be avoided (normal or reversed).

That leads to answer A, which is a best practice.

   upvoted 1 times

  **Mcloudgirl** 2 years, 5 months ago

I agree, based on the docs, B. Leading with a non-reversed timestamp will lead to hotspotting, reversed is the way to go.

   upvoted 2 times

  **zellck** Highly Voted  2 years, 5 months ago

Selected Answer: B

B is the answer.

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

If you usually retrieve the most recent records first, you can use a reversed timestamp in the row key by subtracting the timestamp from your programming language's maximum value for long integers (in Java, `java.lang.Long.MAX_VALUE`). With a reversed timestamp, the records will be ordered from most recent to least recent.

   upvoted 13 times

  **Ryannn23** Most Recent  3 months ago

Selected Answer: A

Vote A, as explained by Augustax:

Agree A is the best option because:

1. Multi-tenancy solution
2. As with any timestamp, avoid starting a row key with a reversed timestamp so that you don't cause hotspots.


   upvoted 3 times

  **Augustax** 3 months, 2 weeks ago

Selected Answer: A

Agree A is the best option because:

1. Multi-tenancy solution
2. As with any timestamp, avoid starting a row key with a reversed timestamp so that you don't cause hotspots.

   upvoted 4 times

  **shangning007** 4 months, 2 weeks ago

Selected Answer: D

I don't think any answer is correct.

A lot people upvote for B, but based on <https://cloud.google.com/bigtable/docs/schema-design#time-based>, "As with any timestamp, avoid starting a row key with a reversed timestamp so that you don't cause hotspots."

   upvoted 4 times

  **ToiToi** 6 months ago

Selected Answer: D

Why other options are not as suitable:

A and B (One table for all indices): Storing all indices in a single table can lead to performance issues as the table grows larger. It also makes it harder to scale individual indices independently.

C. (Timestamp as row key): Using a regular timestamp would place the most recent data at the end of the table. making it

Using a regular timestamp would place the most recent data at the end of the table, making it less efficient to retrieve the latest prices.

   upvoted 2 times

  **SamuelTsch** 6 months, 1 week ago

Selected Answer: D

Option B and Option D are both from my point of view correct. It depends on the situation. If there is a need to get the information from each stock index, then D is more suitable. Otherwise B.

   upvoted 2 times

  **mayankazyour** 8 months ago

Selected Answer: D

1. Reverse Timestamp for most recent stock prices
2. Having different tables for each stock is more efficient, improves the query performance and option B doesn't specify stock in row key.

   upvoted 2 times

  **iooj** 9 months ago

Selected Answer: A

Row keys that start with a timestamp (irrespective reversed or not) cause 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 (index in our case) to avoid hotspots.

The ideal option would be: "use the index and reversed timestamp as the row key design".

   upvoted 6 times

  **datapassionate** 1 year, 3 months ago

Selected Answer: B

B is a correct answer because "you need to access only the most recent stock prices"

"If you usually retrieve the most recent records first, you can use a reversed timestamp in the row key by subtracting the timestamp from your programming language's maximum value for long integers (in Java, `java.lang.Long.MAX_VALUE`). With a reversed timestamp, the records will be ordered from most recent to least recent."

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

   upvoted 5 times

  **TVH_Data_Engineer** 1 year, 4 months ago

Selected Answer: B

B. One unique table for all indices, reverse timestamp as row key:

A single table for all indices keeps the structure simple.

Using a reverse timestamp as part of the row key ensures that the most recent data comes first in the sorted order. This design is beneficial for quickly accessing the latest data.

For example: you can convert the timestamp to a string and format it in reverse order, like "yyyyMMddHHmmss", ensuring newer dates and times are sorted lexicographically before older ones.

   upvoted 2 times

  **kshehadyx** 1 year, 7 months ago

Correct is B

   upvoted 1 time



  **arien_chen** 1 year, 8 months ago

Selected Answer: D

Option B using reverse timestamp only, this is not the answer.
The right answer should be using the index and reverse timestamp as the row key.

So, Option D is the only answer, because not A, B, C.

   upvoted 6 times


  **Lanro** 1 year, 9 months ago

Selected Answer: B

<https://cloud.google.com/bigtable/docs/schema-design#row-keys> - If you usually retrieve the most recent records first, you can use a reversed timestamp

B it is.

   upvoted 2 times


  **Chom** 1 year, 10 months ago

Selected Answer: A

A is the answer

   upvoted 2 times

upvoted 2 times


 **vaga1** 1 year, 10 months ago

Selected Answer: B

the answer relies on whether the application needs to access the whole indexes at the same time or not. If yes then is B, if no is A.

In mind the answer is yes, so B makes more sense: I retrieve all the list at the same time.

   upvoted 2 times

 **ajdf** 1 year, 11 months ago

Selected Answer: B

<https://cloud.google.com/bigtable/docs/schema-design#time-based> If you usually retrieve the most recent records first, you can use a reversed timestamp in the row key by subtracting the timestamp from your programming language's maximum value for long integers (in Java, `java.lang.Long.MAX_VALUE`). With a reversed timestamp, the records will be ordered from most recent to least recent.

   upvoted 2 times

[Load full discussion...](#)



Platform

> [Home](#)

> [All Exams](#)

> [Examtopics PRO](#)

> [Training Courses](#)



© 2024 ExamTopics