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 60 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 60

Topic #: 1

[All Professional Data Engineer Questions]

You launched a new gaming app almost three years ago. You have been uploading log files from the previous day to a separate Google BigQuery table with the table name format LOGS_yyyymmdd. You have been using table wildcard functions to generate daily and monthly reports for all time ranges. Recently, you discovered that some queries that cover long date ranges are exceeding the limit of 1,000 tables and failing. How can you resolve this issue?

A. Convert all daily log tables into date-partitioned tables

B. Convert the sharded tables into a single partitioned table

C. Enable query caching so you can cache data from previous months

D. Create separate views to cover each month, and query from these views

Show Suggested Answer

by [deleted] at *March 21, 2020, 1:28 p.m.*

Comments

Type your comment...

Submit

⊟ [Removed] Highly Voted 4 years, 1 month ago

should be B
https://cloud.google.com/bigquery/docs/creating-partitioned-tables#converting_date-sharded_tables_into_ingestion-time_partitioned_tables

upvoted 38 times

jin0 1 year, 2 months ago

is it already partitioned? there is a table [table]_yyyymmdd it seems to partitioned by date from log files. but I confuse why D. is not a answer? if there is only reason to fail from query that exceeding 1,000 tables then I think creating views could be solution because querying views containing under 1,000 tables by a view could be queried.

upvoted 1 times

Rajuuu 3 years, 10 months ago

The above link does mention about shard ing benefits but only about partition tables.
A is correct.

upvoted 5 times

Tanzu 2 years, 2 months ago

https://cloud.google.com/bigquery/docs/partitioned-tables provides that info you are looking for. Shortly, partitioning performs better than sharding (PREFIX_yymmdd). and it is easy and supported that you can convert sharded tables into ingestion-time partitioned table.
So, B is only option and better one.

upvoted 3 times

vholti 2 years, 6 months ago

The question mentions tables are sharded. So B is more appropriate answer I think.
https://cloud.google.com/bigquery/docs/creating-partitioned-tables#convert-date-sharded-tables

upvoted 5 times

g2000 3 years, 3 months ago

keyword is single

upvoted 6 times

Chelseajcole 2 years, 7 months ago

you are right.

Partitioning versus sharding
Table sharding is the practice of storing data in multiple tables, using a naming prefix such as [PREFIX]_YYYYMMDD.

Partitioning is recommended over table sharding, because partitioned tables perform better. With sharded tables, BigQuery must maintain a copy of the schema and metadata for each table. BigQuery might also need to verify permissions for each queried table. This practice also adds to query overhead and affects query performance.

If you previously created date-sharded tables, you can convert them into an ingestion-time partitioned table.

upvoted 12 times

[Removed] Highly Voted 4 years, 1 month ago

Answer: B
Description: Google says that when you have multiple wildcard tables, best option is to shard it into single partitioned table. Time and cost efficient

upvoted 26 times

lgdantas 3 years, 8 months ago

Can you please share the reference?

upvoted 2 times

Tumri 2 years, 8 months ago

https://cloud.google.com/bigquery/docs/partitioned-tables#dt_partition_shard

upvoted 7 times

jatinbhatia2055 Most Recent 4 months, 3 weeks ago

Selected Answer: B

Sharded tables, like LOGS_yyyymmdd, are useful for managing data, but querying across a long date range with table wildcards can lead to inefficiencies and exceed the 1,000 table limit in BigQuery. Instead of using multiple sharded tables, you should consider converting these into a partitioned table.

A partitioned table allows you to store all the log data in a single table, but logically divides the data into partitions (e.g., by date). This way, you can efficiently query data across long date ranges without hitting the 1,000 table limit.

upvoted 1 times

Oleksandr0501 1 year ago

Oleksandr0501 1 year ago

Selected Answer: B

gpt: Thank you for your feedback and additional information. You are correct that partitioned tables have a limit of 4,000 partitions, so partitioning tables by date could potentially run into this limit in the future. In this case, option B, converting sharded tables into a single partitioned table, could be a reasonable solution to avoid exceeding the maximum number of tables in BigQuery.

As you mentioned, sharded tables require additional metadata and permissions verification, which can impact query performance. Converting sharded tables into a single partitioned table can improve performance and reduce query overhead.

Therefore, based on the information provided, option B seems to be the most appropriate solution for avoiding the limit of 1,000 tables in BigQuery and optimizing query performance.

upvoted 1 times

luks_skywalker 1 year, 1 month ago

The question seems pretty badly written. One important thing to remember is that partitioned tables also have a limit of 4000 partitions (https://cloud.google.com/bigquery/docs/partitioned-tables#ingestion_time), so moving everything to one table would just delay the problem. However, option A is not clear on how it will be done. One table per year with daily partitions? Best solution as no limit will be reached. One table per day? Then we have the same 1000 tables problem.
All things considered I'll stick to B, simply because the problem will definitely be solved for the next few years, so I'd say it's a reasonable solution.

upvoted 2 times

PolyMoe 1 year, 3 months ago

Selected Answer: B

Answer is B.
Table sharding is the practice of storing data in multiple tables, using a naming prefix such as [PREFIX]_YYYYMMDD.
Partitioning is recommended over table sharding, because partitioned tables perform better. With sharded tables, BigQuery must maintain a copy of the schema and metadata for each table. BigQuery might also need to verify permissions for each queried table. This practice also adds to query overhead and affects query performance.
In answer A. we still are creating tableS (even though partioned). So we still facing the issue of max 1000 tables. In B. we have only ONE table (partioned)

upvoted 2 times

samdhimal 1 year, 3 months ago

Why not A?
By converting all daily log tables into date-partitioned tables, you can take advantage of partition pruning to limit the number of tables that need to be scanned during a query. Partition pruning allows BigQuery to skip scanning partitions that are not within the date range specified in the query, thus reducing the number of tables that need to be scanned and can help to avoid reaching the 1,000 table limit.
A Seems like the correct answer but I can be wrong...

upvoted 3 times

RoshanAshraf 1 year, 3 months ago

Selected Answer: B

B. Convert the sharded tables into a single partitioned table
It was a sharded Table (format is the HINT here); converting to partition table is the option.
Also as per GCP its recommended to use Partition over Sharding

upvoted 1 times

korntewin 1 year, 3 months ago

Selected Answer: A

I chose option A. From all the comments I have seen, there are various things that are misunderstood.
1. Option A is a single table with multiple shards! Google does recommend to use partition rather than shard as it has a better performance (https://cloud.google.com/bigquery/docs/partitioned-tables#dt_partition_shard)
2. Option B is a single table with single partition! Single partition is a no for large table

upvoted 1 times

DipT 1 year, 4 months ago

Selected Answer: B

https://cloud.google.com/bigquery/docs/partitioned-tables

upvoted 1 times

DGames 1 year, 4 months ago

Selected Answer: B

Option A - already doing same loading data in separate table daily and reached 1000 table limit.
Option B - Use wild card to query the data
Option C & D - make no sense

upvoted 1 times

odacir 1 year, 4 months ago

⊟ **odacir** 1 year, 4 months ago

its B.
A - Even if you have 100+ partitioned tables, you still have the limit of less than 1000 tables. So this doesn't work for this problem.
C It's a no sense. Cache its 24h for every table that has been query in the last 24 and has no changes. Also, cache is not support with wildcard multiple tables.
D Will not work because it's a recursive issue. You still will have 100+ tables, beam query
B will work, you materialize in only one table, so will be working perfectly.

👍 ↩ ⚑ upvoted 1 times

⊟ **Nirca** 1 year, 6 months ago

Selected Answer: B

Convert MANY sharded tables into a single ONE (partitioned) table

👍 ↩ ⚑ upvoted 2 times

⊟ **rrr000** 1 year, 8 months ago

selecting for daily/monthly data from one single partition will be very expensive. I think A is the best answer

👍 ↩ ⚑ upvoted 1 times

⊟ **Preemptible_cerebrus** 1 year, 10 months ago

Selected Answer: B

C'mon, how much time are you going to take to partition every single table you have? second point and the most important, you have a table for every SINGLE DAY "LOGS_YYYYMMDD" partitioning every table will end on scanning all the records of each table when you query them by date ranges using the wildcards, there will be no difference on time-partitioning each table versus consuming them as described.

👍 ↩ ⚑ upvoted 2 times

⊟ **AmirN** 1 year, 11 months ago

If you follow option A, you will end up with the same amount of tables, e.g 1500 tables, though they will all be partitioned, which is not helpful.
Option B takes all the sharded tables and makes one large partitioned table.

👍 ↩ ⚑ upvoted 1 times

> ⊟ **rrr000** 1 year, 8 months ago
>
> Partitions are not tables. The issue is not performance. It is the limit imposed by bq regarding how many tables you can query.
>
> 👍 ↩ ⚑ upvoted 1 times

⊟ **mihaioff** 2 years ago

Selected Answer: B

It's B
https://cloud.google.com/bigquery/docs/creating-partitioned-tables#converting_date-sharded_tables_into_ingestion-time_partitioned_tables

👍 ↩ ⚑ upvoted 1 times

Load full discussion...

EXAMTOPICS

Platform

- Home
- All Exams
- Examtopics PRO
- Training Courses