Google Discussions

EXAM PROFESSIONAL DATA ENGINEER TOPIC 1 QUESTION 24 DISCUSSION

Actual exam question from Google's Professional Data Engineer
Question #: 24
Topic #: 1
[All Professional Data Engineer Questions]

You have spent a few days loading data from comma-separated values (CSV) files into the Google BigQuery table CLICK_STREAM. The column DT stores the epoch time of click events. For convenience, you chose a simple schema where every field is treated as the STRING type. Now, you want to compute web session durations of users who visit your site, and you want to change its data type to the TIMESTAMP. You want to minimize the migration effort without making future queries computationally expensive. What should you do?

A. Delete the table CLICK_STREAM, and then re-create it such that the column DT is of the TIMESTAMP type. Reload the data.

B. Add a column TS of the TIMESTAMP type to the table CLICK_STREAM, and populate the numeric values from the column TS for each row. Reference the column TS instead of the column DT from now on.

C. Create a view CLICK_STREAM_V, where strings from the column DT are cast into TIMESTAMP values. Reference the view CLICK_STREAM_V instead of the table CLICK_STREAM from now on.

D. Add two columns to the table CLICK STREAM: TS of the TIMESTAMP type and IS_NEW of the BOOLEAN type. Reload all data in append mode. For each appended row, set the value of IS_NEW to true. For future queries, reference the column TS instead of the column DT, with the WHERE clause ensuring that the value of IS_NEW must be true.

E. Construct a query to return every row of the table CLICK_STREAM, while using the built-in function to cast strings from the column DT into TIMESTAMP values. Run the query into a destination table NEW_CLICK_STREAM, in which the column TS is the TIMESTAMP type. Reference the table NEW_CLICK_STREAM instead of the table CLICK_STREAM from now on. In the future, new data is loaded into the table NEW_CLICK_STREAM.

Show Suggested Answer

by jvg637 at *March 11, 2020, 7:13 p.m.*

Comments

Type your comment...

Submit

jvg637 Highly Voted 5 years, 1 month ago
"E" looks better. For D, the database will be double in size (which increases the storage price) and the user has to spend some more days reloading all the data.
upvoted 31 times

jkhong 2 years, 4 months ago
Also D doesn't make sense since we're filtering IS_NEW to true to only consider future data, which disregards our previously loaded data
upvoted 5 times

assU2 2 years, 5 months ago
"You want to minimize the migration effort without making future queries computationally expensive." Nothing about storage price.
upvoted 4 times

[Removed] Highly Voted 5 years, 1 month ago
E - more simple and reasonable. Also recommended if not concerned about cost but simplicity.
https://cloud.google.com/bigquery/docs/manually-changing-schemas#changing_a_columns_data_type
upvoted 22 times

Tanzu 3 years, 3 months ago
Due to the hard limitations of bq, Not E is the simple answer by the way!
upvoted 1 times

Parandhaman_Margan Most Recent 1 month, 3 weeks ago
Selected Answer: C
Changing DT to TIMESTAMP. Create a view casting the string. Answer C.
upvoted 1 times

Abizi 2 months ago
Selected Answer: E
I was hesitating between C and E, but E seems to be the good one
upvoted 1 times

LP_PDE 3 months, 1 week ago
Selected Answer: C
Both options C (creating a view) and E (creating a new table) avoid reloading the original data. However, C is the better choice for minimizing effort and maintaining performance. Views don't store any data themselves. They simply act as a layer on top of the existing table. This means you avoid the cost of storing duplicate data, which can be significant for large tables.
upvoted 1 times

Nittin 9 months ago
Selected Answer: C
Create a view no data migration easy to do but computational efficient queries not sure (?)
upvoted 3 times

mark1223jkh 11 months, 3 weeks ago
E. It recreates the table one time and everything is fixed. Next time you load, load to the new table, you can delete the previous one.
Definitely not C. The question says I have to minimize future query effort, which literally means "don't create a view that converts from STR to TIMESTAMP for every row."
upvoted 1 times

suwalsageen12 11 months, 3 weeks ago
Selected Answer: E
Option E is correct.
The question is asking to consider the Query cost for future.
This is a one time job to fix the Timestamp column, no views were created

This is a one time job to fix the Timestamp column. no views were created.

upvoted 1 times

teka112233 1 year ago

Selected Answer: E

Why Option E is the best choice:

It modifies the schema with minimal data movement.
The original table remains untouched for potential future needs.
Future data loads can directly go to the new table with the desired schema.
Queries referencing the new table (NEW_CLICK_STREAM) will benefit from the optimized data type for timestamp operations.

upvoted 1 times

GYORK 1 year, 1 month ago

Selected Answer: C

minimizing effort is key.

upvoted 1 times

TVH_Data_Engineer 1 year, 4 months ago

Selected Answer: C

A view in Google BigQuery is a virtual table defined by a SQL query. By creating a view that casts the DT column as a TIMESTAMP, you can transform the data format without altering the underlying data in the CLICK_STREAM table. This means you don't have to reload any data, thereby minimizing migration effort.

upvoted 4 times

apoio.certificacoes.closer 4 months, 3 weeks ago

Depends. If you create a materialized view, that tracks. If it's not a materialized view, the underlying query will run every time there's a query against the view.

upvoted 1 times

axantroff 1 year, 5 months ago

Selected Answer: E

Good point about the logical views and the desire to reduce costs. I would vote for E

upvoted 1 times

mk_choudhary 1 year, 6 months ago

The best way to minimize the migration effort without making future queries computationally expensive is to create a view and reference it instead of the table. This is because views are materialized when they are queried, so they do not incur any additional overhead.
So the answer is (C).

upvoted 1 times

brokeasspanda 1 year, 5 months ago

C doesn't say materialized view, there's a difference with a regular view so it'll be slower and more expensive on every call to that view.

upvoted 1 times

rtcpost 1 year, 6 months ago

Selected Answer: E

Option "E"
It avoids the need to delete and recreate the entire CLICK_STREAM table, which is time-consuming and requires reloading all data.

It allows you to use a simple query to cast the existing DT column as TIMESTAMP values and store the results in a new table, NEW_CLICK_STREAM.

You can gradually migrate to the new data format, and your future queries will be able to utilize the TIMESTAMP data type for more efficient processing.

upvoted 2 times

sergiomujica 1 year, 8 months ago

Option D duplicates, not a good solution

upvoted 1 times

NeoNitin 1 year, 9 months ago

E. E. You can use a special command to change the time on the old cards to the better type "TIMESTAMP" and create a new box called "NEW_CLICK_STREAM." From now on, you'll look at the new box whenever you want to know the time. It's like having a new and better box to keep things tidy and organized.

So, the best way to change the time on the little cards to the better type "TIMESTAMP" is option E. It's like using magic to create a new box and making sure everything is still easy to find and work with. It's a clever way to keep track of time and make your website even better!

upvoted 1 times

tal_ 1 year, 10 months ago

Selected Answer: E

they asked to "change its data type"

upvoted 1 times

Load full discussion...