

🔗 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 175 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 175

Topic #: 1

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

Your company is implementing a data warehouse using BigQuery, and you have been tasked with designing the data model. You move your on-premises sales data warehouse with a star data schema to BigQuery but notice performance issues when querying the data of the past 30 days. Based on Google's recommended practices, what should you do to speed up the query without increasing storage costs?

- A. Denormalize the data.
- B. Shard the data by customer ID.
- C. Materialize the dimensional data in views.
- D. Partition the data by transaction date.

[Show Suggested Answer](#)

by [PhuocT](#) at Sept. 2, 2022, 7:10 p.m.

Comments

Type your comment...

[Submit](#)



[waiebdi](#) [Highly Voted](#) 1 year, 8 months ago

Selected Answer: D

D is the right answer because it does not increase storage costs.

A is not correct because denormalization typically increases the amount of storage needed.

   upvoted 14 times

  **Kimich** 11 months, 1 week ago



"Agree with you, denormalize usually increases storage, which may lead to an increase in cost. As for speeding up the query without increasing storage costs, another method is to partition the data by transaction date."

   upvoted 1 times

  **Aman47** **Most Recent** 10 months, 3 weeks ago

Bro, you are playing with words now. Gotta read the question fully.

   upvoted 2 times

  **philv** 1 year, 1 month ago

Some might say that Star schema is already denormalized, but it is considered relationnal (hence kind of normalized) from Google's perspective:

"BigQuery performs best when your data is denormalized. Rather than preserving a relational schema such as a star or snowflake schema, denormalize your data and take advantage of nested and repeated columns. Nested and repeated columns can maintain relationships without the performance impact of preserving a relational (normalized) schema."

I would go for A

https://cloud.google.com/bigquery/docs/nested-repeated#when_to_use_nested_and_repeated_columns

   upvoted 1 times

  **philv** 1 year ago

Changed my mind to D because of the "without increasing storage costs" part.

   upvoted 1 times

  **vamgcp** 1 year, 3 months ago

Selected Answer: D


Option D - BigQuery supports partitioned tables, where the data is divided into smaller, manageable portions based on a chosen column (e.g., transaction date). By partitioning the data based on the transaction date, BigQuery can efficiently query only the relevant partitions that contain data for the past 30 days, reducing the amount of data that needs to be scanned. Partitioning does not increase storage costs. It organizes existing data in a more structured manner, allowing for better query performance without any additional storage expenses.

   upvoted 1 times

  **WillemHendr** 1 year, 4 months ago

A is not a bad idea, but this questions is written around "please partition first on date", which is common best practice. The "storage" remark is hinting on we are not going to 'explode' the data for the sake of performance.

   upvoted 2 times

  **pcadolini** 1 year, 11 months ago

Selected Answer: A

I think better option is [A] considering GCP Documentation: <https://cloud.google.com/bigquery/docs/migration/schema-data-overview#denormalization> "BigQuery supports both star and snowflake schemas, but its native schema representation is neither of those two. It uses nested and repeated fields instead for a more natural representation of the data Changing your schema to use nested and repeated fields is an excellent evolutionary choice. It reduces the number of joins required for your queries, and it aligns your schema with the BigQuery internal data representation. Internally, BigQuery organizes data using the Dremel model and stores it in a columnar storage format called Capacitor."

   upvoted 4 times

  **zellck** 1 year, 11 months ago

Selected Answer: D

D is the answer.

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

A partitioned table is a special table that is divided into segments, called partitions, that make it easier to manage and query your data. By dividing a large table into smaller partitions, you can improve query performance, and you can control costs by reducing the number of bytes read by a query.

   upvoted 3 times

  **NicolasN** 1 year, 12 months ago

Selected Answer: D

A sneaky question.

[D] Yes - Since data is queried with date criteria, partitioning by transaction date will surely speed it up without further cost.

[A] Yes? - Star schema is a denormalized model but as user Reall01 pointed out, the option to use nested and repeated

fields can be considered a further denormalization. And if the model hasn't frequently changing dimensions, this kind of denormalization will result in increased performance, according to https://cloud.google.com/bigquery/docs/loading-data#loading_denormalized_nested_and_repeated_data :

"In some circumstances, denormalizing your data and using nested and repeated fields doesn't result in increased performance. Avoid denormalization in these use cases:

- You have a star schema with frequently changing dimensions"

I guess that the person who added this question, had in mind [D] as a correct answer. If the questioner had all the aforementioned under consideration, would state clearly if there are frequently changing dimensions in the schema.

👍 ↩ 🚩 upvoted 4 times

🗋 👤 **josrojgra** 2 years ago

Selected Answer: D

Star schema is supported by Big Query but is not the most efficient form, if you should design a schema from scratch google recommend to use nested and repeated fields.

In this case, you already have done a migration of the schema and data, so it sounds good and with less effort to do partitioning by transaction date than to redesign the schema.

And other aspect to consider is that this is a data warehouse, so is sure that there is an ETL process and if you change the schema you must adapt the ETL process.

I vote for D.

👍 ↩ 🚩 upvoted 1 times

🗋 👤 **devaid** 2 years, 1 month ago

Selected Answer: D

Star schema is not denormalized itself, but this assumes you already have moved ur data to big query, because you are querying. So, as BQ is not relational, the data already have been denormalized. I go with D.

👍 ↩ 🚩 upvoted 2 times

🗋 👤 **learner2610** 2 years, 1 month ago

I think Denormalizing here means ,using big queries native data representation and that is using nested and repeated columns .Thats is the best practice in GCP

<https://cloud.google.com/bigquery/docs/nested-repeated#example>

👍 ↩ 🚩 upvoted 1 times

🗋 👤 **[Removed]** 2 years, 2 months ago

Selected Answer: D

https://cloud.google.com/bigquery/docs/migration/schema-data-overview#migrating_data_and_schema_from_on-premises_to_bigquery

Star schema. This is a denormalized model, where a fact table collects metrics such as order amount, discount, and quantity, along with a group of keys. These keys belong to dimension tables such as customer, supplier, region, and so on. Graphically, the model resembles a star, with the fact table in the center surrounded by dimension tables.

Star schema is already denormalized so partition makes more sense going with D

👍 ↩ 🚩 upvoted 2 times

🗋 👤 **Real101** 2 years, 1 month ago

If you drill down within that link and land at: <https://cloud.google.com/architecture/bigquery-data-warehouse> it mentions " In some cases, you might want to use nested and repeated fields to denormalize your data." under schema design. Feels like a poorly written question since all depends on what context you take things in as "denormalization"

👍 ↩ 🚩 upvoted 2 times

🗋 👤 **GabyB** 1 year, 4 months ago

In some circumstances, denormalizing your data and using nested and repeated fields doesn't result in increased performance. For example, star schemas are typically optimized schemas for analytics, and as a result, performance might not be significantly different if you attempt to denormalize further.

<https://cloud.google.com/bigquery/docs/best-practices-performance-nested>

👍 ↩ 🚩 upvoted 1 times

🗋 👤 **NicolasN** 1 year, 12 months ago

You bring up a valid point. According to denormalization best practices, there is a critical info missing in order to decide whether further denormalization with nested and repeated fields could help, if there are frequently changing dimensions. Here's a quote from https://cloud.google.com/bigquery/docs/loading-data#loading_denormalized_nested_and_repeated_data :

"In some circumstances, denormalizing your data and using nested and repeated fields doesn't result in increased performance. Avoid denormalization in these use cases:

- You have a star schema with frequently changing dimensions."

   upvoted 2 times

  **AWSandeep** 2 years, 2 months ago

D. Partition the data by transaction date.

Star schema is already denormalized.

   upvoted 3 times

  **PhuocT** 2 years, 2 months ago

Selected Answer: D

should be D, not A

   upvoted 2 times



Platform

> [Home](#)

> [All Exams](#)

> [Examtopics PRO](#)

> [Training Courses](#)



© 2024 ExamTopics