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 168 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 168

Topic #: 1

[All Professional Data Engineer Questions]

You work for a financial institution that lets customers register online. As new customers register, their user data is sent to Pub/Sub before being ingested into
BigQuery. For security reasons, you decide to redact your customers' Government issued Identification Number while allowing customer service representatives to view the original values when necessary. What should you do?

A. Use BigQuery's built-in AEAD encryption to encrypt the SSN column. Save the keys to a new table that is only viewable by permissioned users.

B. Use BigQuery column-level security. Set the table permissions so that only members of the Customer Service user group can see the SSN column.

C. Before loading the data into BigQuery, use Cloud Data Loss Prevention (DLP) to replace input values with a cryptographic hash.

D. Before loading the data into BigQuery, use Cloud Data Loss Prevention (DLP) to replace input values with a cryptographic format-preserving encryption token.

Show Suggested Answer

by AWSandeep at *Sept. 2, 2022, 7:01 p.m.*

Comments

Type your comment...

Submit

AWSandeep Highly Voted 2 years, 8 months ago

Selected Answer: B

B. While C and D are intriguing, they don't specify how to enable customer service representatives to receive access to the encryption token.

upvoted 11 times

cloud_rider 5 months ago

B will show the values to the customer support service all the time as they have access to it, so no redaction as per the ask. Another thing is the requirement is to view when necessary, so D fits this requirement and format preserving encryption can be reverted when necessary.

upvoted 1 times

MaxNRG 1 year, 4 months ago

B. BigQuery column-level security:

Pros: Granular control over column access, ensures only authorized users see the SSN column.
Cons: Doesn't truly redact the data. The SSN values are still stored in BigQuery, even if hidden from unauthorized users. A potential security breach could expose them.

upvoted 1 times

ffggrre 1 year, 6 months ago

there is no SSN in question, it can be any ID.

upvoted 1 times

Lanro Highly Voted 1 year, 9 months ago

Selected Answer: D

I don't see why we should use DLP since we know exactly the column that should be locked or encrypted. On the other hand having a cryptographic representation of SSN helps to aggregate/analyse entries. So I will vote for D, but B is much more easy to implement. Garbage question indeed.

upvoted 6 times

SamuelTsch Most Recent 6 months, 1 week ago

Selected Answer: D

In the question, there is no mention of SSN column.

upvoted 2 times

SamuelTsch 6 months, 1 week ago

also, in the question, "you decide to REDACT ...". Option B does not redact the values.

upvoted 1 times

MohaSa1 6 months, 2 weeks ago

Selected Answer: D

Authorized users can decrypt the FPE tokens back to the original GIINs, D is the best option.

upvoted 2 times

baimus 6 months, 4 weeks ago

Selected Answer: D

D (FPE) does indeed allow encryption to be reversed if desired, allowing operatives to review the original key. This makes it preferable to B, as it's also more secure.

upvoted 2 times

Topg4u 11 months ago

D:
SSN is only tied to USA not in any other countries, The question did not mention SSN.

upvoted 2 times

MaxNRG 1 year, 4 months ago

Selected Answer: D

The best option is D - Before loading the data into BigQuery, use Cloud Data Loss Prevention (DLP) to replace input values with a cryptographic format-preserving encryption token.

The key reasons are:

DLP allows redacting sensitive PII like SSNs before loading into BigQuery. This provides security by default for the raw SSN values

values.
Using format-preserving encryption keeps the column format intact while still encrypting, allowing business logic relying on SSN format to continue functioning.
The encrypted tokens can be reversed to view original SSNs when required, meeting the access requirement for customer service reps.

upvoted 3 times

MaxNRG 1 year, 4 months ago

Option A does encrypt SSN but requires managing keys separately.

Option B relies on complex IAM policy changes instead of encrypting by default.

Option C hashes irreversibly, preventing customer service reps from viewing original SSNs when required.

Therefore, using DLP format-preserving encryption before BigQuery ingestion balances both security and analytics requirements for SSN data.

upvoted 1 times

MaxNRG 1 year, 4 months ago

Why not B. BigQuery column-level security:
Doesn't truly redact the data. The SSN values are still stored in BigQuery, even if hidden from unauthorized users. A potential security breach could expose them.

upvoted 2 times

Aman47 1 year, 4 months ago

Selected Answer: D

Even if you provide Column level access control, The Data Owners or other hierarchies above it will also be able to view very sensitive data. Better to just use encryption and decryption. As this data can also never be used for any analytic workloads

upvoted 3 times

spicebits 1 year, 5 months ago

Selected Answer: D

Answer has to be D. Question says "you decide to redact your customers' Government issued Identification Number while allowing customer service representatives to view the original values when necessary"... Redact... view the original values... D is the only choice.

upvoted 3 times

Nirca 1 year, 6 months ago

Selected Answer: B

It might not be D!
Since - only the Frame is kept. the data will be changed.
Format Preserving Encryption (FPE), endorsed by NIST, is an advanced encryption technique that transforms data into an encrypted format while preserving its original structure. For instance, a 16-digit credit card number encrypted with FPE will still be a 16-digit number

upvoted 1 times

Helinia 1 year, 4 months ago

No, the value using FPE can be decrypted with key.
"Encrypted values can be re-identified using the original cryptographic key and the entire output value, including surrogate annotation."

https://cloud.google.com/dlp/docs/pseudonymization#supported-methods

upvoted 1 times

ffggrre 1 year, 6 months ago

Selected Answer: B

Customer service needs to see the original value, not possible with other options.

upvoted 1 times

kcl10 1 year, 7 months ago

Selected Answer: B

of course B

upvoted 1 times

ckanaar 1 year, 7 months ago

Selected Answer: D

I believe the crux to the question is that the cryptographic format-preserving encryption token is re-identifiable, whereas the cryptographic hash is not: https://cloud.google.com/dlp/docs/transformations-reference

Therefore, customer service can view the original values when necessary in case of D.

upvoted 3 times

ckanaar 1 year, 7 months ago

Nevermind, this can actually also be done in the case of answer B. They are both correct, just different implementations. No idea

upvoted 2 times

knith66 1 year, 9 months ago

the question mentions that "user data is sent to Pub/Sub before being ingested" instead of just saying data goes to big query through pub/sub. So some alteration is expected before being injected into the big query. So option D should work.

upvoted 2 times

sr25 1 year, 9 months ago

Selected Answer: D

D. The question says giving CSR's access to values "when necessary" - not default access like given in B. D is a better option using the token.

upvoted 2 times

ZZHZZH 1 year, 9 months ago

Selected Answer: B

One of the key requirement is to be able to let authorized personel see the ID. D doesn't specify that.

upvoted 1 times

vaga1 1 year, 11 months ago

Selected Answer: D

The answer is between B and D as well described in many comments.

I personally do not see any reason to keep the information available using a token or a mask. It is not a PAN card number, it's just a personal ID. It should not be useful for analytical purposes.

I'm gonna go for D then

upvoted 2 times

vaga1 1 year, 11 months ago

sorry B

upvoted 1 times

Load full discussion...