Question #1 *Topic 1*

Your retail company wants to predict customer churn using historical purchase data stored in BigQuery. The dataset includes customer demographics, purchase history, and a label indicating whether the customer churned or not. You want to build a machine learning model to identify customers at risk of churning. You need to create and train a logistic regression model for predicting customer churn, using the customer_data table with the churned column as the target label. Which BigQuery ML query should you use?

A.
```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT *
FROM customer_data;
```

B.
```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT * EXCEPT(churned),
       churned AS label
FROM customer_data;
```

C.
```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT * EXCEPT(churned)
FROM customer_data;
```

D.
```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT churned as label
FROM customer_data;
```

Suggested Answer: *B*

Community vote distribution

B (100%)

n2183712847 1 month, 2 weeks ago

Selected Answer: B

The best option is B. Option B is best because it correctly selects all columns as features except the churned column, and then explicitly designates the churned column as the label for BigQuery ML, which is required for training. Option A (SELECT *) is incorrect because while it includes the churned column, it doesn't explicitly define it as the label. Option C (SELECT * EXCEPT(churned)) is incorrect because it excludes the target churned column, which is essential for training a supervised model. Option D (SELECT churned as label) is incorrect because it only selects the label and excludes all feature columns needed for prediction. Therefore, Option B is the only query that correctly sets up the data for BigQuery ML logistic regression training.

upvoted 1 times

baviru 1 month ago

You can use SELECT * EXCEPT (foo) and SELECT foo AS bar.

https://ln.run/dFRAn

upvoted 1 times

trashbox 3 months ago

Selected Answer: B

You can use SELECT * EXCEPT (foo) and SELECT foo AS bar.

upvoted 2 times

Question #2 *Topic 1*

Your company has several retail locations. Your company tracks the total number of sales made at each location each day. You want to use SQL to calculate the weekly moving average of sales by location to identify trends for each store. Which query should you use?

A.

```
SELECT store_id, date, total_sales, AVG(total_sales) OVER (
PARTITION BY store_id
ORDER BY total_sales RANGE BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_avg
FROM store_sales_daily
```

B.

```
SELECT store_id, date, total_sales, AVG(total_sales)
OVER (
PARTITION BY date ORDER BY store_id ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_avg
FROM store_sales_daily
```

C.

```
SELECT store_id, date, total_sales, AVG(total_sales)
OVER (
PARTITION BY store_id
ORDER BY date ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_avg
FROM store_sales_daily
```

D.

```
SELECT store_id, date, total_sales, AVG(total_sales)
OVER (
PARTITION BY total_sales
ORDER BY date RANGE BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_avg
FROM store_sales_daily
```

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 2 weeks ago

Selected Answer: C

The best option is C. Option C is best because it correctly partitions by store_id to calculate the moving average for each store individually, and orders by date to ensure the moving average is calculated chronologically over the preceding 7 days (including the current day, thus 6 preceding). ROWS BETWEEN 6 PRECEDING AND CURRENT ROW then correctly defines the 7-day window for the average. Option A is incorrect because it orders by total_sales instead of date, making the rolling average based on sales value order, not time, which is illogical for a weekly trend. Option B is incorrect because it partitions by date instead of store_id, calculating a daily moving average across all stores, not per store. Option D is incorrect because it partitions by total_sales, which is nonsensical for analyzing trends by location, and orders by date within that illogical partition. Therefore, Option C is the only query that correctly calculates a weekly moving average of sales by store location.

upvoted 1 times

trashbox 3 months ago

Selected Answer: C

Should be partitioned by store_id to keep averages separate: A or C

Then, should be ordered by date for chronological window calculation: C

upvoted 1 times

Question #3 *Topic 1*

Your company is building a near real-time streaming pipeline to process JSON telemetry data from small appliances. You need to process messages arriving at a Pub/Sub topic, capitalize letters in the serial number field, and write results to BigQuery. You want to use a managed service and write a minimal amount of code for underlying transformations. What should you do?

A. Use a Pub/Sub to BigQuery subscription, write results directly to BigQuery, and schedule a transformation query to run every five minutes.

B. Use a Pub/Sub to Cloud Storage subscription, write a Cloud Run service that is triggered when objects arrive in the bucket, performs the transformations, and writes the results to BigQuery.

C. Use the “Pub/Sub to BigQuery” Dataflow template with a UDF, and write the results to BigQuery.

D. Use a Pub/Sub push subscription, write a Cloud Run service that accepts the messages, performs the transformations, and writes the results to BigQuery.

Suggested Answer: *C*

Community vote distribution

C (80%) A (20%)

JAGLees 3 weeks, 3 days ago

Selected Answer: C

Cloud Run is not minimal code (or recommended for Data Pipelines)

A scheduled job is not "near realtime"

So the answer is Dataflow with a UDF which gives a scalable managed solution with minimal code

upvoted 1 times

n2183712847 1 month, 2 weeks ago

Selected Answer: C

The best option is C. Use the “Pub/Sub to BigQuery” Dataflow template with a UDF. Option C is best because Dataflow templates are managed, serverless, and designed for streaming Pub/Sub data to BigQuery. UDFs allow minimal code for transformations within the pipeline. Option A (Pub/Sub to BigQuery + scheduled query) is incorrect because scheduled queries are not real-time transformations. Option B (Pub/Sub to Cloud Storage + Cloud Run) is incorrect because it adds unnecessary complexity with Cloud Storage as an intermediary and is not truly streaming. Option D (Pub/Sub push + Cloud Run) is incorrect because while real-time, it requires more code in Cloud Run than using a Dataflow UDF and is less purpose-built for data pipelines than Dataflow. Therefore, Option C, Dataflow template with UDF, is the best balance of managed service, minimal code, and near real-time streaming.

upvoted 1 times

bc3f222 1 month, 3 weeks ago

Selected Answer: A

Pub/Sub to BQ is now the recommended solution, no longer need dataflow

upvoted 1 times

rich_maverick 1 month, 3 weeks ago

Selected Answer: C

I agree that C is the best answer. However, answer A is doable and is also low/no code and also considered acceptable.

upvoted 1 times

trashbox 3 months ago

Selected Answer: C

A UDF of the Dataflow is a simpler coding option than a Cloud Run.

upvoted 1 times

Question #4 *Topic 1*

You want to process and load a daily sales CSV file stored in Cloud Storage into BigQuery for downstream reporting. You need to quickly build a scalable data pipeline that transforms the data while providing insights into data quality issues. What should you do?

A. Create a batch pipeline in Cloud Data Fusion by using a Cloud Storage source and a BigQuery sink.

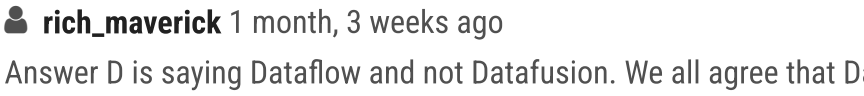
B. Load the CSV file as a table in BigQuery, and use scheduled queries to run SQL transformation scripts.

C. Load the CSV file as a table in BigQuery. Create a batch pipeline in Cloud Data Fusion by using a BigQuery source and sink.

D. Create a batch pipeline in Dataflow by using the Cloud Storage CSV file to BigQuery batch template.

Suggested Answer: *A*

Community vote distribution



n2183712847 1 month, 2 weeks ago

Selected Answer: A

The best option is A. Cloud Data Fusion pipeline (Cloud Storage to BigQuery). Option A is best because Cloud Data Fusion is visual and fast for pipeline building, scalable, handles transformations visually, and provides data quality insights within the pipeline. Option B (BigQuery load + SQL) is incorrect because scheduled queries are less of a pipeline and offer fewer built-in data quality features. Option C (BigQuery load + Data Fusion BQ to BQ) is incorrect because it's inefficient and redundant to load to BigQuery before Data Fusion. Option D (Dataflow template) is incorrect because while scalable, Data Fusion is often quicker to build visually for simpler pipelines. Therefore, Option A, Cloud Data Fusion, is the best balance of speed, scalability, and data quality for this task.

upvoted 2 times

jatinbhatia2055 1 month, 4 weeks ago

Selected Answer: D

There should be more detail in the question. Though both Dataflow and Datafusion can be used. Datafusion is more suitable if you dont want to code and let google do the work.

In case there is more complexity in daily analysis of the CSV, Dataflow is the best approach as it provide in built templates and custom template creation both.

upvoted 2 times

rich_maverick 1 month, 3 weeks ago

Answer D is saying Dataflow and not Datafusion. We all agree that Datafusion is the "quick and scalable" option. Also, Dataflow does not give insights to data quality issues. Answer is A.

upvoted 2 times

trashbox 3 months ago

Selected Answer: A

Cloud Data Fusion enables us to build a scalable data pipeline from Cloud Storage to BigQuery. In addition, the service provides us an end-to-end data lineage for root cause and impact analysis.

upvoted 4 times

Question #5 *Topic 1*

You manage a Cloud Storage bucket that stores temporary files created during data processing. These temporary files are only needed for seven days, after which they are no longer needed. To reduce storage costs and keep your bucket organized, you want to automatically delete these files once they are older than seven days. What should you do?

A. Set up a Cloud Scheduler job that invokes a weekly Cloud Run function to delete files older than seven days.

B. Configure a Cloud Storage lifecycle rule that automatically deletes objects older than seven days.

C. Develop a batch process using Dataflow that runs weekly and deletes files based on their age.

D. Create a Cloud Run function that runs daily and deletes files older than seven days.

Suggested Answer: *B*

Community vote distribution

B (100%)

n2183712847 1 month, 2 weeks ago

Selected Answer: B

The best option is B. Cloud Storage lifecycle rule. Option B is best because lifecycle rules are built-in, automatic, and efficient for Cloud Storage object management like deletion based on age. Option A (Cloud Scheduler + Cloud Run) is incorrect because it's overly complex using two services when one built-in feature exists. Option C (Dataflow batch) is incorrect because Dataflow is overkill for simple deletion and more costly. Option D (Cloud Run daily) is incorrect because it's still more complex than lifecycle rules and daily frequency is likely unnecessary. Therefore, Option B, lifecycle rule, is the simplest, most efficient, and cost-effective solution.

upvoted 1 times

rich_maverick 1 month, 3 weeks ago

Selected Answer: B

Simplest solution. All solutions can be made to work.

upvoted 1 times

jatinbhatia2055 1 month, 4 weeks ago

Selected Answer: B

The most effective solution.

upvoted 2 times

Question #6 *Topic 1*

You work for a healthcare company that has a large on-premises data system containing patient records with personally identifiable information (PII) such as names, addresses, and medical diagnoses. You need a standardized managed solution that de-identifies PII across all your data feeds prior to ingestion to Google Cloud. What should you do?

A. Use Cloud Run functions to create a serverless data cleaning pipeline. Store the cleaned data in BigQuery.

B. Use Cloud Data Fusion to transform the data. Store the cleaned data in BigQuery.

C. Load the data into BigQuery, and inspect the data by using SQL queries. Use Dataflow to transform the data and remove any errors.

D. Use Apache Beam to read the data and perform the necessary cleaning and transformation operations. Store the cleaned data in BigQuery.

Suggested Answer: *B*

Community vote distribution

B (100%)

JAGLees 3 weeks, 3 days ago

Selected Answer: B

A isnt right as Cloud Run isn't well suited to Data pipelines and generally isn't recommended. It *can* handle processing data, but not large volumes of batch data and isn't intended as an ETL tool.
C isn't right because Ingesting first into BigQuery doesn't meet the requirement to cleansing BEFORE ingestion.
D isn't right because Apache BEAM isn't a managed solution (although if the data had to be cleansed before leaving the on prem network it might be a suitable option to install on prem)

Therefore B (Data Fusion) is the best opinion. If DataFlow was here, then that would also be a good choice - perhaps a better one - but without it, Fusion is the obvious winner.

upvoted 1 times

n2183712847 1 month, 2 weeks ago

Selected Answer: B

The best option is B. Cloud Data Fusion. Option B is best because Data Fusion is a managed, visual, standardized data integration service ideal for building de-identification pipelines. Option A (Cloud Run functions) is incorrect because it requires more coding and is less inherently standardized for pipelines. Option C (Load to BigQuery first) is incorrect because it violates the requirement to de-identify before ingestion, creating a security risk. Option D (Apache Beam/Dataflow) is incorrect because while powerful, it's more code-centric and less of a pre-built managed solution compared to Data Fusion. Therefore, Option B, Cloud Data Fusion, is the best managed and standardized solution for pre-ingestion PII de-identification.

upvoted 1 times

rich_maverick 1 month, 3 weeks ago

Selected Answer: B

Cloud Data Fusion can be used to Sensitive Protection Service to de-identify feeds. However, Cloud Dataflow (a Google managed version of Beam) is the more general approach being used. I am only selecting Data Fusion over Beam because it is a named Google service. Had they said Dataflow, I would have gone there instead.

upvoted 1 times

Question #7 *Topic 1*

You manage a large amount of data in Cloud Storage, including raw data, processed data, and backups. Your organization is subject to strict compliance regulations that mandate data immutability for specific data types. You want to use an efficient process to reduce storage costs while ensuring that your storage strategy meets retention requirements. What should you do?

A. Configure lifecycle management rules to transition objects to appropriate storage classes based on access patterns. Set up Object Versioning for all objects to meet immutability requirements.

B. Move objects to different storage classes based on their age and access patterns. Use Cloud Key Management Service (Cloud KMS) to encrypt specific objects with customer-managed encryption keys (CMEK) to meet immutability requirements.

C. Create a Cloud Run function to periodically check object metadata, and move objects to the appropriate storage class based on age and access patterns. Use object holds to enforce immutability for specific objects.

D. Use object holds to enforce immutability for specific objects, and configure lifecycle management rules to transition objects to appropriate storage classes based on age and access patterns.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 2 weeks ago

Selected Answer: D

The best option is D. Object holds and lifecycle rules. Option D is best because Object holds enforce immutability for compliance, and lifecycle rules ensure cost efficiency by managing storage classes. Option A (Object Versioning) is incorrect because Versioning is for recovery, not strict immutability. Option B (CMEK) is incorrect because CMEK is for encryption, not immutability. Option C (Cloud Run) is incorrect because Cloud Run for storage class transition is less efficient than lifecycle rules. Therefore, Option D is the most effective and efficient solution for both immutability and cost.

upvoted 1 times

Question #8 *Topic 1*

You work for an ecommerce company that has a BigQuery dataset that contains customer purchase history, demographics, and website interactions. You need to build a machine learning (ML) model to predict which customers are most likely to make a purchase in the next month. You have limited engineering resources and need to minimize the ML expertise required for the solution. What should you do?

A. Use BigQuery ML to create a logistic regression model for purchase prediction.

B. Use Vertex AI Workbench to develop a custom model for purchase prediction.

C. Use Colab Enterprise to develop a custom model for purchase prediction.

D. Export the data to Cloud Storage, and use AutoML Tables to build a classification model for purchase prediction.

Suggested Answer: *A*

Community vote distribution

A (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: A

A.

Use BigQuery ML minimizes the ML Expertise required.

upvoted 2 times

Question #9 *Topic 1*

You are designing a pipeline to process data files that arrive in Cloud Storage by 3:00 am each day. Data processing is performed in stages, where the output of one stage becomes the input of the next. Each stage takes a long time to run. Occasionally a stage fails, and you have to address the problem. You need to ensure that the final output is generated as quickly as possible. What should you do?

A. Design a Spark program that runs under Dataproc. Code the program to wait for user input when an error is detected. Rerun the last action after correcting any stage output data errors.

B. Design the pipeline as a set of PTransforms in Dataflow. Restart the pipeline after correcting any stage output data errors.

C. Design the workflow as a Cloud Workflow instance. Code the workflow to jump to a given stage based on an input parameter. Rerun the workflow after correcting any stage output data errors.

D. Design the processing as a directed acyclic graph (DAG) in Cloud Composer. Clear the state of the failed task after correcting any stage output data errors.

Suggested Answer: *D*

Community vote distribution

D (100%)

JAGLees 3 weeks, 3 days ago

Selected Answer: D

Composer is the recommended orchestration tool for managing workloads in Google Cloud and perfectly meets these requirements

upvoted 1 times

n2183712847 1 month, 3 weeks ago

Selected Answer: D

The best option is D. Design the processing as a directed acyclic graph (DAG) in Cloud Composer. Clear the state of the failed task after correcting any stage output data errors. Cloud Composer (Apache Airflow) is specifically designed for orchestrating complex data pipelines, provides robust error handling with task-level rerun capabilities for efficient recovery, and offers a fully managed and scheduled environment, making it the most suitable choice for ensuring the final output is generated as quickly as possible in the face of occasional stage failures. Options A, B, and C are less efficient due to manual intervention, less granular error recovery, or lack of dedicated workflow orchestration features.

upvoted 1 times

Question #10 *Topic 1*

Another team in your organization is requesting access to a BigQuery dataset. You need to share the dataset with the team while minimizing the risk of unauthorized copying of data. You also want to create a reusable framework in case you need to share this data with other teams in the future. What should you do?

A. Create authorized views in the team's Google Cloud project that is only accessible by the team.

B. Create a private exchange using Analytics Hub with data egress restriction, and grant access to the team members.

C. Enable domain restricted sharing on the project. Grant the team members the BigQuery Data Viewer IAM role on the dataset.

D. Export the dataset to a Cloud Storage bucket in the team's Google Cloud project that is only accessible by the team.

Suggested Answer: *A*

Community vote distribution

JAGLees 3 weeks, 4 days ago

Selected Answer: B

Either Authorized Views or Analytics Hub could be reasonable, but I feel the mention of reusable framework and preventing data copying are pushing towards Analytics Hub woth egress restrictions

upvoted 1 times

Rio55 1 month ago

Selected Answer: A

Comparing the options, A (authorized views) and B (Analytics Hub) both seem great for job, authorized views are often a straightforward and effective way to share data while controlling access and minimizing copying.

I've choosing so far Authorized views for internal sharing and Analytics hub for external sharing.

upvoted 1 times

n2183712847 1 month, 3 weeks ago

Selected Answer: A

While Analytics Hub (Option B) is a powerful data sharing platform, Authorized Views (Option A) are the superior choice for this specific scenario of sharing a BigQuery dataset with another internal team while prioritizing minimizing unauthorized copying and ease of implementation. Authorized Views are simpler, more secure against data copying for internal use cases, easier to implement, and more cost-effective for this particular need.

I think it could be B or A, but I believe that analytics hub is mainly for sharing with an outside organization userbase.

upvoted 2 times

Question #11 *Topic 1*

Your company has developed a website that allows users to upload and share video files. These files are most frequently accessed and shared when they are initially uploaded. Over time, the files are accessed and shared less frequently, although some old video files may remain very popular.
You need to design a storage system that is simple and cost-effective. What should you do?

A. Create a single-region bucket with Autoclass enabled.

B. Create a single-region bucket. Configure a Cloud Scheduler job that runs every 24 hours and changes the storage class based on upload date.

C. Create a single-region bucket with custom Object Lifecycle Management policies based on upload date.

D. Create a single-region bucket with Archive as the default storage class.

Suggested Answer: *A*

Community vote distribution

A (100%)

trashbox Highly Voted 2 months, 2 weeks ago

Selected Answer: A

I will go with "A" because it should be automatically optimized based on actual access patterns.

upvoted 5 times

Pubb 1 month ago

I chose "A" too, because the problem does not describe any specific lifecycle policy.

upvoted 1 times

AmarT Most Recent 4 weeks ago

Selected Answer: A

Can anyone please share the complete question set.

If you already have please share it over tiwariamarnath77@gmail.com.

Thanks already for the community support.

upvoted 1 times

Rio55 1 month ago

Selected Answer: A

It's A because of the old video files that can remain popular

upvoted 1 times

n2183712847 1 month, 3 weeks ago

Selected Answer: A

A. Autoclass makes the most sense for some files accessed and shared less frequently, with some old video files remaining popular.

upvoted 2 times

jatinbhatia2055 1 month, 4 weeks ago

Selected Answer: A

Autoclass is a feature of Google Cloud Storage that automatically optimizes the storage class for objects based on their access patterns. When objects are newly uploaded and frequently accessed, Autoclass will place them in a more performant (and more expensive) storage class, like Standard. Over time, if access to the objects decreases, Autoclass will automatically transition them to a lower-cost storage class, such as Nearline, Coldline, or Archive, depending on their access frequency.
This solution is simple, cost-effective, and does not require manual management or custom lifecycle rules. It automates the optimization of storage classes without the need for any additional processes or scheduling jobs.

upvoted 1 times

Question #12 *Topic 1*

You recently inherited a task for managing Dataflow streaming pipelines in your organization and noticed that proper access had not been provisioned to you. You need to request a Google-provided IAM role so you can restart the pipelines. You need to follow the principle of least privilege. What should you do?

A. Request the Dataflow Developer role.

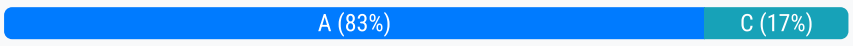
B. Request the Dataflow Viewer role.

C. Request the Dataflow Worker role.

D. Request the Dataflow Admin role.

Suggested Answer: *A*

Community vote distribution



Rio55 1 month ago

Selected Answer: A

A and D would work but because to follow the principle of least privilege, we're going to choose A, Dataflow developer.

upvoted 1 times

n2183712847 1 month, 3 weeks ago

Selected Answer: A

The best option is A. Request the Dataflow Developer role. Requesting the Dataflow Developer role is optimal because it grants the necessary permissions to restart Dataflow pipelines, aligning with the user's need, while adhering to the principle of least privilege by providing a focused set of permissions. Option B (Dataflow Viewer) is insufficient as it only provides read-only access and lacks the permission to restart pipelines. Option C (Dataflow Worker) is incorrect as it is intended for service accounts, not user access, and doesn't grant pipeline management permissions. Option D (Dataflow Admin) is excessive, granting far more permissions than needed for simply restarting pipelines, thus violating the principle of least privilege. Therefore, Option A is the clear and correct choice for requesting the minimum necessary IAM role to restart Dataflow pipelines.

upvoted 4 times

jatinbhatia2055 1 month, 4 weeks ago

Selected Answer: C

Dataflow Worker provides the necessary permissions for managing and interacting with Dataflow streaming pipelines, including restarting them. This role gives you the ability to execute and manage the pipelines without granting excessive permissions, aligning with the principle of least privilege.

upvoted 1 times

Question #13 *Topic 1*

You need to create a new data pipeline. You want a serverless solution that meets the following requirements:

• Data is streamed from Pub/Sub and is processed in real-time.

• Data is transformed before being stored.

• Data is stored in a location that will allow it to be analyzed with SQL using Looker.

Which Google Cloud services should you recommend for the pipeline?

A. 1. Dataproc Serverless
2. Bigtable

B. 1. Cloud Composer
2. Cloud SQL for MySQL

C. 1. BigQuery
2. Analytics Hub

D. 1. Dataflow
2. BigQuery

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: D

pubsub, dataflow, bigquery, looker

this architecture is very common in GCP.

upvoted 3 times

Question #14 *Topic 1*

Your team wants to create a monthly report to analyze inventory data that is updated daily. You need to aggregate the inventory counts by using only the most recent month of data, and save the results to be used in a Looker Studio dashboard. What should you do?

A. Create a materialized view in BigQuery that uses the SUM( ) function and the DATE_SUB( ) function.

B. Create a saved query in the BigQuery console that uses the SUM( ) function and the DATE_SUB( ) function. Re-run the saved query every month, and save the results to a BigQuery table.

C. Create a BigQuery table that uses the SUM( ) function and the _PARTITIONDATE filter.

D. Create a BigQuery table that uses the SUM( ) function and the DATE_DIFF( ) function.

Suggested Answer: *A*

Community vote distribution

A (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: A

The best option is A. Create a materialized view in BigQuery that uses the SUM( ) function and the DATE_SUB( ) function. Materialized views provide the efficiency of pre-computation, automatic updates, and are directly usable in Looker Studio, making them the ideal choice for creating an automated and performant monthly inventory report. Options B, C, and D are either manual, technically inaccurate in their description, or less efficient than leveraging the power of materialized views for this reporting task.

upvoted 2 times

Question #15 *Topic 1*

You have a BigQuery dataset containing sales data. This data is actively queried for the first 6 months. After that, the data is not queried but needs to be retained for 3 years for compliance reasons. You need to implement a data management strategy that meets access and compliance requirements, while keeping cost and administrative overhead to a minimum. What should you do?

A. Use BigQuery long-term storage for the entire dataset. Set up a Cloud Run function to delete the data from BigQuery after 3 years.

B. Partition a BigQuery table by month. After 6 months, export the data to Coldline storage. Implement a lifecycle policy to delete the data from Cloud Storage after 3 years.

C. Set up a scheduled query to export the data to Cloud Storage after 6 months. Write a stored procedure to delete the data from BigQuery after 3 years.

D. Store all data in a single BigQuery table without partitioning or lifecycle policies.

Suggested Answer: *B*

Community vote distribution

B (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: B

The best option is B. Partition a BigQuery table by month. After 6 months, export the data to Coldline storage. Implement a lifecycle policy to delete the data from Cloud Storage after 3 years. Option B provides the optimal balance of meeting the 3-year data retention requirement, achieving excellent cost optimization by tiering data to Coldline, and maintaining reasonable administrative overhead. Option C is more complex and less clear on overall retention and cost due to data duplication. Option D is the simplest administratively but provides the worst cost optimization. Option A is not cost-optimized and has moderate overhead. Therefore, Option B is the most practical and well-rounded solution for this data management scenario.

upvoted 2 times

Question #16 *Topic 1*

You have created a LookML model and dashboard that shows daily sales metrics for five regional managers to use. You want to ensure that the regional managers can only see sales metrics specific to their region. You need an easy-to-implement solution. What should you do?

A. Create a sales_region user attribute, and assign each manager's region as the value of their user attribute. Add an access_filter Explore filter on the region_name dimension by using the sales_region user attribute.

B. Create five different Explores with the sql_always_filter Explore filter applied on the region_name dimension. Set each region_name value to the corresponding region for each manager.

C. Create separate Looker dashboards for each regional manager. Set the default dashboard filter to the corresponding region for each manager.

D. Create separate Looker instances for each regional manager. Copy the LookML model and dashboard to each instance. Provision viewer access to the corresponding manager.

Suggested Answer: *A*

Community vote distribution

A (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: A

The best and easiest-to-implement solution is A. Create a sales_region user attribute, and assign each manager's region as the value of their user attribute. Add an access_filter Explore filter on the region_name dimension by using the sales_region user attribute. This option is optimal because it directly leverages Looker's built-in features for row-level security (Access Filters) and user personalization (User Attributes), making it easy to implement, scalable, maintainable, and secure. Option B (Separate Explores) is inefficient and less maintainable. Option C (Separate Dashboards) does not provide data security and is easily bypassed. Option D (Separate Looker Instances) is extreme overkill, costly, and impractical. Therefore, Option A is the clear and correct choice for an easy-to-implement and secure solution.

upvoted 1 times

Question #17 *Topic 1*

You need to design a data pipeline that ingests data from CSV, Avro, and Parquet files into Cloud Storage. The data includes raw user input. You need to remove all malicious SQL injections before storing the data in BigQuery. Which data manipulation methodology should you choose?

A. EL

B. ELT

C. ETL

D. ETLT

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: C

"remove all malicious SQL injections before storing the data in BigQuery"

If you are preforming transformation before, it is ETL.
If you are doing after, it is ELT.
If you are doing before and after, ETLT

upvoted 2 times

Question #18 *Topic 1*

You are working with a large dataset of customer reviews stored in Cloud Storage. The dataset contains several inconsistencies, such as missing values, incorrect data types, and duplicate entries. You need to clean the data to ensure that it is accurate and consistent before using it for analysis. What should you do?

A. Use the PythonOperator in Cloud Composer to clean the data and load it into BigQuery. Use SQL for analysis.

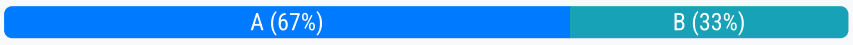
B. Use BigQuery to batch load the data into BigQuery. Use SQL for cleaning and analysis.

C. Use Storage Transfer Service to move the data to a different Cloud Storage bucket. Use event triggers to invoke Cloud Run functions to load the data into BigQuery. Use SQL for analysis.

D. Use Cloud Run functions to clean the data and load it into BigQuery. Use SQL for analysis.

Suggested Answer: *B*

Community vote distribution



n2183712847 1 month, 3 weeks ago

Selected Answer: B

The best option is B. Use BigQuery to batch load the data into BigQuery and use SQL for cleaning and analysis. Loading directly into BigQuery and using SQL provides the optimal balance of efficiency and simplicity for cleaning large datasets before analysis by leveraging BigQuery's scalable processing for both loading and transformation. Option A (Cloud Composer + PythonOperator) adds unnecessary complexity of workflow orchestration and external processing before loading, reducing efficiency. Option C (Storage Transfer Service + Cloud Run) overcomplicates the process with extra data movement and event-driven functions, making it less direct for data cleaning. Option D (Cloud Run functions) is less efficient for large-scale data cleaning compared to BigQuery SQL's parallel processing and adds complexity before data is in BigQuery for analysis. Therefore, loading into BigQuery and using SQL is the most efficient and straightforward approach for cleaning data before analysis in this scenario.

upvoted 1 times

SaquibHerman 2 months ago

Selected Answer: A

PythonOperator allows leveraging Python libraries (e.g., Pandas, PySpark) to perform robust data cleaning tasks:

Handle missing values (e.g., imputation, filtering).

Fix incorrect data types (e.g., string-to-date conversions).

Remove duplicates (e.g., using deduplication logic).

upvoted 2 times

Question #19 *Topic 1*

Your retail organization stores sensitive application usage data in Cloud Storage. You need to encrypt the data without the operational overhead of managing encryption keys. What should you do?

A. Use Google-managed encryption keys (GMEK).

B. Use customer-managed encryption keys (CMEK).

C. Use customer-supplied encryption keys (CSEK).

D. Use customer-supplied encryption keys (CSEK) for the sensitive data and customer-managed encryption keys (CMEK) for the less sensitive data.

Suggested Answer: *A*

Community vote distribution

A (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: A

Google can manage the encryption keys for you using GMEK.

CMEK is customer managed within Cloud KMS

CSEK is keys that are supplied outside, and managed outside.

upvoted 1 times

Question #20 *Topic 1*

You work for a financial organization that stores transaction data in BigQuery. Your organization has a regulatory requirement to retain data for a minimum of seven years for auditing purposes. You need to ensure that the data is retained for seven years using an efficient and cost-optimized approach. What should you do?

A. Create a partition by transaction date, and set the partition expiration policy to seven years.

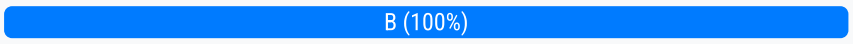
B. Set the table-level retention policy in BigQuery to seven years.

C. Set the dataset-level retention policy in BigQuery to seven years.

D. Export the BigQuery tables to Cloud Storage daily, and enforce a lifecycle management policy that has a seven-year retention rule.

Suggested Answer: *B*

Community vote distribution



n2183712847 1 month, 3 weeks ago

Selected Answer: B

In the context of the question, both are efficient and cost-optimized. If the regulatory requirement applies to all transaction data which is logically grouped into a dataset, dataset-level retention (Option C) might be slightly more practical for management. However, if the requirement is specifically for this BigQuery table, table-level (Option B) is equally valid and directly applicable. In a multiple choice scenario, both are strong candidates. However, for the most direct answer and often simpler approach when the requirement is somewhat broad ("transaction data"), dataset level policy (Option C) might be slightly favoured for ease of management if the retention is consistent for all transaction data. But for a single table scenario implied by "BigQuery table", table level is also perfectly valid. Given the options, and aiming for the most directly applicable and efficient, table-level retention is a very strong and valid choice.

Given the wording of the question is slightly less about dataset context and more about "the data" in BigQuery, table-level retention is a very direct and correct answer.

upvoted 1 times

Question #21 *Topic 1*

You need to create a weekly aggregated sales report based on a large volume of data. You want to use Python to design an efficient process for generating this report. What should you do?

A. Create a Cloud Run function that uses NumPy. Use Cloud Scheduler to schedule the function to run once a week.

B. Create a Colab Enterprise notebook and use the bigframes.pandas library. Schedule the notebook to execute once a week.

C. Create a Cloud Data Fusion and Wrangler flow. Schedule the flow to run once a week.

D. Create a Dataflow directed acyclic graph (DAG) coded in Python. Use Cloud Scheduler to schedule the code to run once a week.

Suggested Answer: *D*

Community vote distribution

Rio55 1 month ago

Selected Answer: D

Option D (Dataflow DAG coded in Python) and Option B using (bigframes.pandas in Colab Enterprise) seem to be the most suitable.

Option D, Dataflow is designed for parallel processing of large datasets, making it efficient. Python is the chosen language, and Cloud Scheduler handles the weekly scheduling.

Option B, using bigframes.pandas in Colab Enterprise is also a possibility for leveraging Python and BigQuery, but Dataflow is generally more robust and scalable for production ETL pipelines involving large datasets.

Option A with Cloud Run might face limitations with large data and execution time. Option C doesn't directly use Python code for the process design.

I would choose D.

upvoted 1 times

n2183712847 1 month, 3 weeks ago

Selected Answer: B

Option B (Colab Enterprise + bigframes.pandas) and Option D (Dataflow DAG in Python) are the most efficient options for handling large data volumes and using Python.

Option B is likely more straightforward and faster to implement for generating a report, especially if the analyst is familiar with Pandas and notebooks. bigframes.pandas simplifies interaction with BigQuery data within a Python environment for reporting purposes.

Option D is more robust and scalable for general data pipelines and potentially more complex transformations, but might be overkill for a weekly reporting task compared to the ease of use offered by bigframes.pandas in Colab Enterprise.

Given the need for an "efficient process for generating this report" and the desire to use Python, Option B provides a very efficient and relatively simpler path for creating a weekly aggregated sales report directly from BigQuery using Python's familiar Pandas-like syntax via bigframes.pandas in a scheduled Colab Enterprise notebook.

Final Answer: The final answer is
B
B

upvoted 1 times

Question #22 *Topic 1*

Your organization has decided to move their on-premises Apache Spark-based workload to Google Cloud. You want to be able to manage the code without needing to provision and manage your own cluster. What should you do?

A. Migrate the Spark jobs to Dataproc Serverless.

B. Configure a Google Kubernetes Engine cluster with Spark operators, and deploy the Spark jobs.

C. Migrate the Spark jobs to Dataproc on Google Kubernetes Engine.

D. Migrate the Spark jobs to Dataproc on Compute Engine.

Suggested Answer: *A*

Community vote distribution

A (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: A

Dataproc serverless to for spark workload without provisioning + managing your own infrastructure.

upvoted 2 times

Question #23 *Topic 1*

You are developing a data ingestion pipeline to load small CSV files into BigQuery from Cloud Storage. You want to load these files upon arrival to minimize data latency. You want to accomplish this with minimal cost and maintenance. What should you do?

A. Use the bq command-line tool within a Cloud Shell instance to load the data into BigQuery.

B. Create a Cloud Composer pipeline to load new files from Cloud Storage to BigQuery and schedule it to run every 10 minutes.

C. Create a Cloud Run function to load the data into BigQuery that is triggered when data arrives in Cloud Storage.

D. Create a Dataproc cluster to pull CSV files from Cloud Storage, process them using Spark, and write the results to BigQuery.

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: C

Given the choices, cloud run function is the only one that can

upvoted 1 times

Question #24 *Topic 1*

Your organization has a petabyte of application logs stored as Parquet files in Cloud Storage. You need to quickly perform a one-time SQL-based analysis of the files and join them to data that already resides in BigQuery. What should you do?

A. Create a Dataproc cluster, and write a PySpark job to join the data from BigQuery to the files in Cloud Storage.

B. Launch a Cloud Data Fusion environment, use plugins to connect to BigQuery and Cloud Storage, and use the SQL join operation to analyze the data.

C. Create external tables over the files in Cloud Storage, and perform SQL joins to tables in BigQuery to analyze the data.

D. Use the bq load command to load the Parquet files into BigQuery, and perform SQL joins to analyze the data.

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: C

The most efficient and quick solution for a one-time SQL analysis of petabyte-scale Parquet files in Cloud Storage joined with BigQuery data is C. Create external tables over the files in Cloud Storage and perform SQL joins. External tables allow you to query data directly in Cloud Storage with SQL, avoiding the time and cost of loading a petabyte of data into BigQuery. This is ideal for a fast, one-time analysis. Options A (Dataproc/Spark) and B (Cloud Data Fusion) are more complex and slower for a quick analysis. Option D (bq load) is inefficient and slow as it requires loading a petabyte of data into BigQuery, which is unnecessary for a one-time analysis of external files. Therefore, Option C provides the most direct, efficient, and SQL-centric approach for this scenario.

upvoted 1 times

Question #25 *Topic 1*

Your team is building several data pipelines that contain a collection of complex tasks and dependencies that you want to execute on a schedule, in a specific order. The tasks and dependencies consist of files in Cloud Storage, Apache Spark jobs, and data in BigQuery. You need to design a system that can schedule and automate these data processing tasks using a fully managed approach. What should you do?

A. Use Cloud Scheduler to schedule the jobs to run.

B. Use Cloud Tasks to schedule and run the jobs asynchronously.

C. Create directed acyclic graphs (DAGs) in Cloud Composer. Use the appropriate operators to connect to Cloud Storage, Spark, and BigQuery.

D. Create directed acyclic graphs (DAGs) in Apache Airflow deployed on Google Kubernetes Engine. Use the appropriate operators to connect to Cloud Storage, Spark, and BigQuery.

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: C

The best fully managed solution for scheduling and automating complex data pipelines is C. Use Cloud Composer with DAGs and appropriate operators. Cloud Composer, being a fully managed Apache Airflow service, is specifically designed for orchestrating complex workflows with dependencies and offers built-in operators to connect to Cloud Storage, Spark (via Dataproc), and BigQuery. Option D (Airflow on GKE) is not fully managed and adds operational overhead. Options A (Cloud Scheduler) and B (Cloud Tasks) are not designed for complex workflow orchestration and dependency management. Therefore, Option C is the optimal choice for a fully managed, robust, and feature-rich solution for data pipeline orchestration.

upvoted 1 times

Question #26 *Topic 1*

You are responsible for managing Cloud Storage buckets for a research company. Your company has well-defined data tiering and retention rules. You need to optimize storage costs while achieving your data retention needs. What should you do?

A. Configure the buckets to use the Archive storage class.

B. Configure a lifecycle management policy on each bucket to downgrade the storage class and remove objects based on age.

C. Configure the buckets to use the Standard storage class and enable Object Versioning.

D. Configure the buckets to use the Autoclass feature.

Suggested Answer: *B*

Community vote distribution

B (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: B

The best solution for optimizing Cloud Storage costs and achieving data retention needs is B. Configure a lifecycle management policy on each bucket. Lifecycle Management is designed to automate data tiering (moving data to cheaper classes as it ages) and data retention (deleting objects based on age or other rules), directly addressing the requirements for cost optimization and rule-based data management. Option D (Autoclass) helps with tiering based on access but doesn't handle retention rules. Option A (Archive storage) is too simplistic and inflexible, not addressing tiering or retention rules properly. Option C (Standard + Versioning) increases costs and is counter to the goal of optimization. Therefore, Option B provides the most comprehensive and rule-driven approach for cost optimization and data lifecycle management in Cloud Storage.

upvoted 1 times

Question #27 *Topic 1*

You are using your own data to demonstrate the capabilities of BigQuery to your organization's leadership team. You need to perform a one- time load of the files stored on your local machine into BigQuery using as little effort as possible. What should you do?

A. Write and execute a Python script using the BigQuery Storage Write API library.

B. Create a Dataproc cluster, copy the files to Cloud Storage, and write an Apache Spark job using the spark-bigquery-connector.

C. Execute the bq load command on your local machine.

D. Create a Dataflow job using the Apache Beam FileIO and BigQueryIO connectors with a local runner.

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: C

C. bq load is best for one-time transfer from local machine to BQ

upvoted 1 times

Question #28 *Topic 1*

Your organization uses Dataflow pipelines to process real-time financial transactions. You discover that one of your Dataflow jobs has failed. You need to troubleshoot the issue as quickly as possible. What should you do?

A. Set up a Cloud Monitoring dashboard to track key Dataflow metrics, such as data throughput, error rates, and resource utilization.

B. Create a custom script to periodically poll the Dataflow API for job status updates, and send email alerts if any errors are identified.

C. Navigate to the Dataflow Jobs page in the Google Cloud console. Use the job logs and worker logs to identify the error.

D. Use the gcloud CLI tool to retrieve job metrics and logs, and analyze them for errors and performance bottlenecks.

Suggested Answer: *C*

Community vote distribution

C (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: C

The quickest way to troubleshoot a failed Dataflow job is C. Use the Dataflow Jobs page in the Google Cloud console and examine job and worker logs. The console provides immediate and direct access to job status and detailed logs, allowing for rapid identification of errors. Option A (Cloud Monitoring Dashboard) is for proactive monitoring, not immediate failure diagnosis. Option B (Custom Script) is for future alerting, not current troubleshooting. Option D (gcloud CLI) is powerful but slightly less quick and user-friendly than the console for initial log browsing and error identification in this scenario. Therefore, Option C offers the most direct and efficient path to quickly diagnosing a Dataflow job failure.

upvoted 1 times

Question #29 *Topic 1*

Your company uses Looker to generate and share reports with various stakeholders. You have a complex dashboard with several visualizations that needs to be delivered to specific stakeholders on a recurring basis, with customized filters applied for each recipient. You need an efficient and scalable solution to automate the delivery of this customized dashboard. You want to follow the Google-recommended approach. What should you do?

A. Create a separate LookML model for each stakeholder with predefined filters, and schedule the dashboards using the Looker Scheduler.

B. Create a script using the Looker Python SDK, and configure user attribute filter values. Generate a new scheduled plan for each stakeholder.

C. Embed the Looker dashboard in a custom web application, and use the application's scheduling features to send the report with personalized filters.

D. Use the Looker Scheduler with a user attribute filter on the dashboard, and send the dashboard with personalized filters to each stakeholder based on their attributes.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: D

The optimal and Google-recommended solution is D. Use Looker Scheduler with User Attribute filters. This is the most efficient and scalable approach as it directly utilizes Looker's built-in features to automate personalized dashboard delivery based on user attributes. Option A (Separate LookML models) is inefficient and unscalable due to model duplication. Option B (Python SDK scripting) is more complex than necessary and less efficient than using built-in features. Option C (Embedded Dashboard and Custom App) is overly complex and inefficient, re-implementing Looker's native functionalities. Therefore, Option D is the most straightforward, efficient, and aligned with Google's recommended best practices for Looker.

upvoted 1 times

Question #30 *Topic 1*

You are predicting customer churn for a subscription-based service. You have a 50 PB historical customer dataset in BigQuery that includes demographics, subscription information, and engagement metrics. You want to build a churn prediction model with minimal overhead. You want to follow the Google-recommended approach. What should you do?

A. Export the data from BigQuery to a local machine. Use scikit-learn in a Jupyter notebook to build the churn prediction model.

B. Use Dataproc to create a Spark cluster. Use the Spark MLlib within the cluster to build the churn prediction model.

C. Create a Looker dashboard that is connected to BigQuery. Use LookML to predict churn.

D. Use the BigQuery Python client library in a Jupyter notebook to query and preprocess the data in BigQuery. Use the CREATE MODEL statement in BigQueryML to train the churn prediction model.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: D

The best and Google-recommended solution for building a churn model on a 50 PB BigQuery dataset with minimal overhead is D. Use BigQuery Python client and BigQueryML. BigQueryML enables in-database model training, eliminating data movement and minimizing overhead. This aligns with Google's best practices for BigQuery data. Option A (Local scikit-learn) is impractical due to the dataset size. Option B (Dataproc/Spark) introduces unnecessary data movement and cluster management overhead. Option C (Looker) is for BI, not ML model development. Therefore, Option D is the optimal choice for efficiency, scalability, and adherence to Google's recommendations for BigQuery-based machine learning.

upvoted 1 times

Question #31 *Topic 1*

You are a data analyst at your organization. You have been given a BigQuery dataset that includes customer information. The dataset contains inconsistencies and errors, such as missing values, duplicates, and formatting issues. You need to effectively and quickly clean the data. What should you do?

A. Develop a Dataflow pipeline to read the data from BigQuery, perform data quality rules and transformations, and write the cleaned data back to BigQuery.

B. Use Cloud Data Fusion to create a data pipeline to read the data from BigQuery, perform data quality transformations, and write the clean data back to BigQuery.

C. Export the data from BigQuery to CSV files. Resolve the errors using a spreadsheet editor, and re-import the cleaned data into BigQuery.

D. Use BigQuery's built-in functions to perform data quality transformations.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 2 weeks ago

Selected Answer: D

it's already in bigquery, so just preform the transformation in the dataset

upvoted 2 times

n2183712847 1 month, 3 weeks ago

Selected Answer: D

The best solution for effective and quick data cleaning is D. Use BigQuery's built-in functions. This is the most efficient and quickest approach as it leverages the power of BigQuery SQL for data transformations directly within the BigQuery environment. Option B (Cloud Data Fusion) is a good visual alternative but slower to set up than direct SQL. Option A (Dataflow) is powerful but more complex and time-consuming for initial cleaning. Option C (Spreadsheet Editor) is manual, inefficient, and not scalable for millions of records. Therefore, Option D offers the optimal balance of effectiveness and speed for cleaning data within BigQuery.

upvoted 1 times

Question #32 *Topic 1*

Your organization has several datasets in their data warehouse in BigQuery. Several analyst teams in different departments use the datasets to run queries. Your organization is concerned about the variability of their monthly BigQuery costs. You need to identify a solution that creates a fixed budget for costs associated with the queries run by each department. What should you do?

A. Create a custom quota for each analyst in BigQuery.

B. Create a single reservation by using BigQuery editions. Assign all analysts to the reservation.

C. Assign each analyst to a separate project associated with their department. Create a single reservation by using BigQuery editions. Assign all projects to the reservation.

D. Assign each analyst to a separate project associated with their department. Create a single reservation for each department by using BigQuery editions. Create assignments for each project in the appropriate reservation.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: D

The best solution for creating fixed BigQuery budgets per department is D. Separate projects per department, separate reservations per department, and project assignments to departmental reservations. This approach is optimal because separate reservations per department create fixed, predictable costs for each department. Assigning departmental projects to their respective reservations ensures that each department's query costs are contained within their allocated budget. Options A, B, and C fail to create fixed departmental budgets. Option A (Quotas) is for resource limits, not budgets. Options B and C (Single Reservation) provide overall cost predictability but don't enable departmental budget control. Therefore, Option D is the only option that effectively addresses the requirement of fixed departmental budgets.

upvoted 1 times

Question #33 *Topic 1*

You manage a web application that stores data in a Cloud SQL database. You need to improve the read performance of the application by offloading read traffic from the primary database instance. You want to implement a solution that minimizes effort and cost. What should you do?

A. Use Cloud CDN to cache frequently accessed data.

B. Store frequently accessed data in a Memorystore instance.

C. Migrate the database to a larger Cloud SQL instance.

D. Enable automatic backups, and create a read replica of the Cloud SQL instance.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: D

read the question, it says offloading read traffic from the primary database, meaning there would be a read-replica.

The best solution for improving Cloud SQL read performance by offloading traffic while minimizing effort and cost is D. Enable automatic backups and create a read replica. Read replicas are specifically designed to offload read traffic with minimal effort in Cloud SQL. Option C (Larger Instance) scales up the primary, not offloading, and can be more expensive. Option B (Memorystore) requires more application code changes and cache management, increasing effort. Option A (Cloud CDN) is less directly applicable to database read offloading and requires significant application changes, making it higher effort and less effective for this purpose. Therefore, Option D provides the most direct, efficient, and cost-effective way to improve read performance by offloading read traffic from your Cloud SQL database.

upvoted 1 times

Question #34 *Topic 1*

Your organization plans to move their on-premises environment to Google Cloud. Your organization's network bandwidth is less than 1 Gbps. You need to move over 500 TB of data to Cloud Storage securely, and only have a few days to move the data. What should you do?

A. Request multiple Transfer Appliances, copy the data to the appliances, and ship the appliances back to Google Cloud to upload the data to Cloud Storage.

B. Connect to Google Cloud using VPN. Use Storage Transfer Service to move the data to Cloud Storage.

C. Connect to Google Cloud using VPN. Use the gcloud storage command to move the data to Cloud Storage.

D. Connect to Google Cloud using Dedicated Interconnect. Use the gcloud storage command to move the data to Cloud Storage.

Suggested Answer: *A*

Community vote distribution

A (100%)

n2183712847 1 month, 3 weeks ago

Selected Answer: A

500 TB in a few days with 1gbps bandwidth would require option A. multiple Transfer Appliances

upvoted 1 times

Question #35 *Topic 1*

Your organization uses a BigQuery table that is partitioned by ingestion time. You need to remove data that is older than one year to reduce your organization's storage costs. You want to use the most efficient approach while minimizing cost. What should you do?

A. Create a scheduled query that periodically runs an update statement in SQL that sets the "deleted" column to "yes" for data that is more than one year old. Create a view that filters out rows that have been marked deleted.

B. Create a view that filters out rows that are older than one year.

C. Require users to specify a partition filter using the alter table statement in SQL.

D. Set the table partition expiration period to one year using the ALTER TABLE statement in SQL.

Suggested Answer: *D*

Community vote distribution

D (100%)

n2183712847 1 month, 2 weeks ago

Selected Answer: D

D. is the only one that has a partition expiration, meaning deletion. the other answer choices just filter views.

upvoted 2 times

n2183712847 1 month, 3 weeks ago

Selected Answer: D

The most efficient and cost-effective solution is D. Set the table partition expiration period to one year using the ALTER TABLE statement in SQL. Partition expiration is a built-in BigQuery feature specifically designed to automatically delete old partitions, directly reducing storage costs and requiring minimal effort. Options A and B (soft delete and views) do not reduce storage costs; they only filter or mark data without removing it. Option C (requiring partition filters) is about query optimization, not data removal or storage cost reduction. Therefore, Option D is the only option that directly and effectively addresses the requirement of removing old data to minimize storage costs in a BigQuery partitioned table.

upvoted 2 times