ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 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;
```

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

B)

```
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;
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

Explanation:

In BigQuery ML, when creating a logistic regression model to predict customer churn, the correct query should:

Exclude the target label column (in this case, churned) from the feature columns, as it is used for training and not as a feature input.

Rename the target label column to label, as BigQuery ML requires the target column to be named label.

The chosen query satisfies these requirements:

SELECT * EXCEPT(churned), churned AS label: Excludes churned from features and renames it to label.

The OPTIONS(model_type='logistic_reg') specifies that a logistic regression model is being trained.

This setup ensures the model is correctly trained using the features in the dataset while targeting the churned column for predictions.

Question: 2

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 totalsales RANGE BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling FROM store_sales_daily
```

B)

```
SELECT storeid, date, totalsales, AVG(totalsales)
OVER (
PARTITION BY date ORDER BY store_id ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ) as r 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

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C

Explanation:

To calculate the weekly moving average of sales by location:

The query must group by store_id (partitioning the calculation by each store).

The ORDER BY date ensures the sales are evaluated chronologically.

The ROWS BETWEEN 6 PRECEDING AND CURRENT ROW specifies a rolling window of 7 rows (1 week if each row represents daily data).

The AVG(total_sales) computes the average sales over the defined rolling window.

Chosen query meets these requirements:

- PARTITION BY store_id groups the calculation by each store.
- ORDER BY date orders the rows correctly for the rolling average.
- ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ensures the 7-day moving average.

Extract from Google Documentation: From "Analytic Functions in BigQuery" (https://cloud.google.com/bigquery/docs/reference/standard-sql/analytic-function-concepts): "Use ROWS BETWEEN n PRECEDING AND CURRENT ROW with ORDER BY a time column to compute moving averages over a fixed number of rows, such as a 7-day window, partitioned by a grouping key like store_id."
Reference: Google Cloud Documentation - "BigQuery Window Functions" (https://cloud.google.com/bigquery/docs/reference/standard-sql/window-function-calls).

Question: 3

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.

Answer: C

Explanation:

Using the "Pub/Sub to BigQuery" Dataflow template with a UDF (User-Defined Function) is the optimal choice because it combines near real-time processing, minimal code for transformations, and scalability. The UDF allows for efficient implementation of custom transformations, such as capitalizing letters in the serial number field, while Dataflow handles the rest of the managed pipeline seamlessly.

Question: 4

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.

Answer: A

Explanation:

Using Cloud Data Fusion to create a batch pipeline with a Cloud Storage source and a BigQuery sink is the best solution because:

Scalability: Cloud Data Fusion is a scalable, fully managed data integration service.

Data transformation: It provides a visual interface to design pipelines, enabling quick transformation of data.

Data quality insights: Cloud Data Fusion includes built-in tools for monitoring and addressing data quality issues during the pipeline creation and execution process.

Question: 5

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.

Answer: B

Explanation:

Configuring a Cloud Storage lifecycle rule to automatically delete objects older than seven days is the best solution because:

Built-in feature: Cloud Storage lifecycle rules are specifically designed to manage object lifecycles, such as automatically deleting or transitioning objects based on age.

No additional setup: It requires no external services or custom code, reducing complexity and maintenance.

Cost-effective: It directly achieves the goal of deleting files after seven days without incurring additional compute costs.

Question: 6

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.

Answer: B

Explanation:

Using Cloud Data Fusion is the best solution for this scenario because:

Standardized managed solution: Cloud Data Fusion provides a visual interface for building data pipelines and includes prebuilt connectors and transformations for data cleaning and de-identification.

Compliance: It ensures sensitive data such as PII is de-identified prior to ingestion into Google Cloud, adhering to regulatory requirements for healthcare data.

Ease of use: Cloud Data Fusion is designed for transforming and preparing data, making it a managed and user-friendly tool for this purpose.

It's a fully managed, cloud-native data integration service for building ETL/ELT data pipelines visually.

It offers built-in transformations and connectors, including those suitable for data masking and de-identification.

It provides a standardized, visual interface, making it easier to create and manage data pipelines across various data sources.

It's designed for data integration and transformation, making it ideal for this scenario.

It helps to achieve a standardized managed solution.

Question: 7

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.

Answer: D

Explanation:

Using object holds and lifecycle management rules is the most efficient and compliant strategy for this scenario because:

Immutability: Object holds (temporary or event-based) ensure that objects cannot be deleted or overwritten, meeting strict compliance regulations for data immutability.

Cost efficiency: Lifecycle management rules automatically transition objects to more cost-effective storage classes based on their age and access patterns.

Compliance and automation: This approach ensures compliance with retention requirements while reducing manual effort, leveraging built-in Cloud Storage features.

Question: 8

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.

Answer: A

Explanation:

Using BigQuery ML is the best solution in this case because:

Ease of use: BigQuery ML allows users to build machine learning models using SQL, which requires minimal ML expertise.

Integrated platform: Since the data already exists in BigQuery, there's no need to move it to another service, saving time and engineering resources.

Logistic regression: This is an appropriate model for binary classification tasks like predicting the likelihood of a customer making a purchase in the next month.

Question: 9

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.

Answer: D

Explanation:

Using Cloud Composer to design the processing pipeline as a Directed Acyclic Graph (DAG) is the most suitable approach because:

Fault tolerance: Cloud Composer (based on Apache Airflow) allows for handling failures at specific stages. You can clear the state of a failed task and rerun it without reprocessing the entire pipeline.

Stage-based processing: DAGs are ideal for workflows with interdependent stages where the output of one stage serves as input to the next.

Efficiency: This approach minimizes downtime and ensures that only failed stages are rerun, leading to faster final output generation.

Question: 10

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 dat

a. 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.

Answer: B

Explanation:

Using Analytics Hub to create a private exchange with data egress restrictions ensures controlled sharing of the dataset while minimizing the risk of unauthorized copying. This approach allows you to provide secure, managed access to the dataset without giving direct access to the raw data. The egress restriction ensures that data cannot be exported or copied outside the designated boundaries. Additionally, this solution provides a reusable framework that simplifies future data sharing with other teams or projects while maintaining strict data governance.

Extract from Google Documentation: From "Analytics Hub Overview" (https://cloud.google.com/analytics-hub/docs): "Analytics Hub enables secure, controlled data sharing with private exchanges. Combine with organization policies like restrictDataEgress to prevent data copying, providing a reusable framework for sharing BigQuery datasets across teams." Reference: Google Cloud Documentation - "Analytics Hub" (https://cloud.google.com/analytics-hub).

Question: 11

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 custom Object Lifecycle Management policies based on upload date.
B. Create a single-region bucket with Autoclass enabled.
C. Create a single-region bucket. Configure a Cloud Scheduler job that runs every 24 hours and changes the storage class based on upload date.
D. Create a single-region bucket with Archive as the default storage class.

Answer: B

Explanation:

The storage system must balance cost, simplicity, and access patterns: high initial access, decreasing over time, with some files remaining popular. Google Cloud Storage offers tailored options for this:

Option A: Custom Object Lifecycle Management (OLM) policies (e.g., transition to Nearline after 30 days, Archive after 90 days) are effective but static. They don't adapt to actual usage, so popular old files in Archive would incur high retrieval costs.

Option B: Autoclass automatically adjusts storage classes (Standard, Nearline, Coldline, Archive) based on object access patterns, not just age. It keeps frequently accessed files in Standard (low latency/cost for access) and moves inactive ones to cheaper classes, minimizing costs while preserving simplicity. This fits the "some files remain popular" nuance.

Option C: A Cloud Scheduler job to manually change classes daily is complex (requires scripting, monitoring), error-prone, and less cost-effective than automated solutions like Autoclass or OLM.

Option D: Defaulting to Archive is cheapest for storage but disastrous for access—retrieval costs and latency would

skyrocket for initial high-access periods.

Why B is Best: Autoclass simplifies management (no rules to define) and optimizes costs dynamically. For videos, where access varies unpredictably, it ensures popular files stay accessible without manual intervention, aligning with Google's cost-optimization guidance.

Extract from Google Documentation: From "Autoclass in Cloud Storage" (https://cloud.google.com/storage/docs/autoclass): "Autoclass automatically transitions objects to the most cost-effective storage class based on access patterns, simplifying management and reducing costs for workloads with variable access, such as media files."

Reference: Google Cloud Documentation - "Cloud Storage Autoclass" (https://cloud.google.com/storage/docs/autoclass).

Question: 12

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.

Answer: A

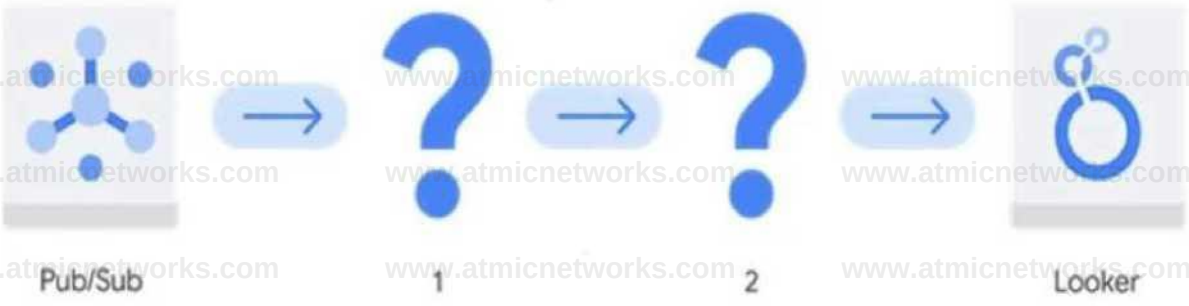
Explanation:

The Dataflow Developer role provides the necessary permissions to manage Dataflow streaming pipelines, including the ability to restart pipelines. This role adheres to the principle of least privilege, as it grants only the permissions required to manage and operate Dataflow jobs without unnecessary administrative access. Other roles, such as Dataflow Admin, would grant broader permissions, which are not needed in this scenario.

Question: 13

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

Answer: D

Explanation:

To build a serverless data pipeline that processes data in real-time from Pub/Sub, transforms it, and stores it for SQL-based analysis using Looker, the best solution is to use Dataflow and BigQuery. Dataflow is a fully managed service for real-time data processing and transformation, while BigQuery is a serverless data warehouse that supports SQL-based querying and integrates seamlessly with Looker for data analysis and visualization. This combination meets the requirements for real-time streaming, transformation, and efficient storage for analytical queries.

Question: 14

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.

Answer: A

Explanation:

Creating a materialized view in BigQuery with the SUM() function and the DATE_SUB() function is the best approach. Materialized views allow you to pre-aggregate and cache query results, making them efficient for repeated access, such as monthly reporting. By using the DATE_SUB() function, you can filter the inventory data to include only the most recent month. This approach ensures that the aggregation is up-to-date with minimal latency and provides efficient integration with Looker Studio for dashboarding.

Question: 15

You have a BigQuery dataset containing sales dat

a. 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.

Answer: B

Explanation:

Partitioning the BigQuery table by month allows efficient querying of recent data for the first 6 months, reducing query costs. After 6 months, exporting the data to Coldline storage minimizes storage costs for data that is rarely accessed but needs to be retained for compliance. Implementing a lifecycle policy in Cloud Storage automates the deletion of the data after 3 years, ensuring compliance while reducing administrative overhead. This approach balances cost efficiency and compliance requirements effectively.

Question: 16

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.

Answer: A

Explanation:

Using a sales_region user attribute is the best solution because it allows you to dynamically filter data based on each manager's assigned region. By adding an access_filter Explore filter on the region_name dimension that references the sales_region user attribute, each manager sees only the sales metrics specific to their region. This approach is easy to implement, scalable, and avoids duplicating dashboards or Explores, making it both efficient and maintainable.

Question: 17

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

Answer: C

Explanation:

The ETL (Extract, Transform, Load) methodology is the best approach for this scenario because it allows you to extract data from the files, transform it by applying the necessary data cleansing (including removing malicious SQL injections), and then load the sanitized data into BigQuery. By transforming the data before loading it into BigQuery, you ensure that only clean and safe data is stored, which is critical for security and data quality.

Question: 18

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.

Answer: B

Explanation:

Using BigQuery to batch load the data and perform cleaning and analysis with SQL is the best approach for this scenario. BigQuery provides powerful SQL capabilities to handle missing values, enforce correct data types, and remove duplicates efficiently. This method simplifies the pipeline by leveraging BigQuery's built-in processing power for both cleaning and analysis, reducing the need for additional tools or services and minimizing complexity.

Question: 19

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.

Answer: A

Explanation:

Using Google-managed encryption keys (GMEK) is the best choice when you want to encrypt sensitive data in Cloud Storage without the operational overhead of managing encryption keys. GMEK is the default encryption mechanism in Google Cloud, and it ensures that data is automatically encrypted at rest with no additional setup or maintenance required. It provides strong security while eliminating the need for manual key management.

Google Cloud encrypts all data at rest by default, and the simplest way to avoid key management overhead is to use Google-managed encryption keys (GMEK).

Option A: GMEK is fully managed by Google, requiring no user intervention, and meets the requirement of no operational overhead while ensuring encryption.

Option B: CMEK requires managing keys in Cloud KMS, adding operational overhead.

Option C: CSEK requires users to supply and manage keys externally, increasing complexity significantly.

Option D: Combining CSEK and CMEK adds unnecessary complexity and overhead for a uniform sensitive dataset.
Extract from Google Documentation: From "Encryption at Rest in Google Cloud" (https://cloud.google.com/security/encryption-at-rest): "By default, Google Cloud automatically encrypts data at rest using Google-managed encryption keys (GMEK), providing strong security without requiring you to manage keys."
Reference: Google Cloud Documentation - "Encryption Options" (https://cloud.google.com/storage/docs/encryption).

Question: 20

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.

Answer: B

Explanation:

Setting a table-level retention policy in BigQuery to seven years is the most efficient and cost- optimized solution to meet the regulatory requirement. A table-level retention policy ensures that the data cannot be deleted or overwritten before the specified retention period expires, providing compliance with auditing requirements while keeping the data within BigQuery for easy access and analysis. This approach avoids the complexity and additional costs of exporting data to Cloud Storage.

Question: 21

You need to create a weekly aggregated sales report based on a large volume of dat

a. 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.

Answer: D

Explanation:

Using Dataflow with a Python-coded Directed Acyclic Graph (DAG) is the most efficient solution for generating a weekly aggregated sales report based on a large volume of data. Dataflow is optimized for large-scale data processing and can handle

aggregation efficiently. Python allows you to customize the pipeline logic, and Cloud Scheduler enables you to automate the process to run weekly. This approach ensures scalability, efficiency, and the ability to process large datasets in a cost-effective manner.

Question: 22

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.

Answer: A

Explanation:

Migrating the Spark jobs to Dataproc Serverless is the best approach because it allows you to run Spark workloads without the need to provision or manage clusters. Dataproc Serverless automatically scales resources based on workload requirements, simplifying operations and reducing administrative overhead. This solution is ideal for organizations that want to focus on managing their Spark code without worrying about the underlying infrastructure. It is cost-effective and fully managed, aligning well with the goal of minimizing cluster management.

Question: 23

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.

Answer: C

Explanation:

Using a Cloud Run function triggered by Cloud Storage to load the data into BigQuery is the best solution because it minimizes both cost and maintenance while providing low-latency data ingestion. Cloud Run is a serverless platform that automatically scales based on the workload, ensuring efficient use of resources without requiring a dedicated instance or cluster. It integrates seamlessly with Cloud Storage event notifications, enabling real-time processing of incoming files and loading them into BigQuery. This approach is cost-effective, scalable, and easy to manage.

The goal is to load small CSV files into BigQuery upon arrival (event-driven) with minimal latency, cost, and maintenance. Google Cloud provides serverless, event-driven options that align with this requirement. Let's evaluate each option in detail:

Option A: Cloud Composer (managed Apache Airflow) can schedule a pipeline to check Cloud Storage every 10 minutes, but this polling approach introduces latency (up to 10 minutes) and incurs
costs for running Composer even when no files arrive. Maintenance includes managing DAGs and the Composer environment, which adds overhead. This is better suited for scheduled batch jobs, not event-driven ingestion.

Option B: A Cloud Run function triggered by a Cloud Storage event (via Eventarc or Pub/Sub) loads files into BigQuery as soon as they arrive, minimizing latency. Cloud Run is serverless, scales to zero when idle (low cost), and requires minimal maintenance (deploy and forget). Using the BigQuery API in the function (e.g., Python client library) handles small CSV loads efficiently. This aligns with Google's serverless, event-driven best practices.

Option C: Dataproc with Spark is designed for large-scale, distributed processing, not small CSV ingestion. It requires cluster management, incurs higher costs (even with ephemeral clusters), and adds unnecessary complexity for a simple load task.

Option D: The bq command-line tool in Cloud Shell is manual and not automated, failing the "upon arrival" requirement. It's a one-off tool, not a pipeline solution, and Cloud Shell isn't designed for persistent automation.

Why B is Best: Cloud Run leverages Cloud Storage's object creation events, ensuring near-zero latency between file arrival

and BigQuery ingestion. It's serverless, meaning no infrastructure to manage, and costs scale with usage (free when idle). For small CSVs, the BigQuery load job is lightweight, avoiding processing overhead.

Extract from Google Documentation: From "Triggering Cloud Run with Cloud Storage Events" (https://cloud.google.com/run/docs/triggering/using-events): "You can trigger Cloud Run services in response to Cloud Storage events, such as object creation, using Eventarc. This serverless approach minimizes latency and maintenance, making it ideal for real-time data pipelines." Additionally, from "Loading Data into BigQuery" (https://cloud.google.com/bigquery/docs/loading-data-cloud-storage- csv): "Programmatically load CSV files from Cloud Storage using the BigQuery API, enabling automated ingestion with minimal overhead."

Reference: Google Cloud Documentation - "Cloud Run Events" (https://cloud.google.com/run/docs), "BigQuery Load Jobs" (https://cloud.google.com/bigquery/docs/loading-data).

Question: 24

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.

Answer: C

Explanation:

Creating external tables over the Parquet files in Cloud Storage allows you to perform SQL-based analysis and joins with data already in BigQuery without needing to load the files into BigQuery. This approach is efficient for a one-time analysis as it avoids the time and cost associated with loading large volumes of data into BigQuery. External tables provide seamless integration with Cloud Storage, enabling quick and cost-effective analysis of data stored in Parquet format.

Question: 25

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.

Answer: C

Explanation:

Using Cloud Composer to create Directed Acyclic Graphs (DAGs) is the best solution because it is a fully managed, scalable workflow orchestration service based on Apache Airflow. Cloud Composer allows you to define complex task dependencies and schedules while integrating seamlessly with Google Cloud services such as Cloud Storage, BigQuery, and Dataproc for Apache Spark jobs. This approach minimizes operational overhead, supports scheduling and automation, and provides an efficient and fully managed way to orchestrate your data pipelines.

Extract from Google Documentation: From "Cloud Composer Overview" (https://cloud.google.com/composer/docs): "Cloud Composer is a fully managed workflow orchestration service built on Apache Airflow, enabling you to schedule and automate complex data pipelines with dependencies across Google Cloud services like Cloud Storage, Dataproc, and BigQuery."

Reference: Google Cloud Documentation - "Cloud Composer" (https://cloud.google.com/composer).

Question: 26

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.

Answer: B

Explanation:

Configuring a lifecycle management policy on each Cloud Storage bucket allows you to automatically transition objects to lower-cost storage classes (such as Nearline, Coldline, or Archive) based on their age or other criteria. Additionally, the policy can automate the removal of objects once they are no longer needed, ensuring compliance with retention rules and optimizing storage costs. This approach aligns well with well-defined data tiering and retention needs, providing cost efficiency and automation.

Extract from Google Documentation: From "Object Lifecycle Management" (https://cloud.google.com/storage/docs/lifecycle): "Use lifecycle management policies to automatically transition objects to lower-cost storage classes (e.g., Nearline, Coldline) and delete them based on age, optimizing costs according to your specific tiering and retention requirements." Reference: Google Cloud Documentation - "Cloud Storage Lifecycle" (https://cloud.google.com/storage/docs/lifecycle).

Question: 27

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.

Answer: C

Explanation:

To troubleshoot a failed Dataflow job as quickly as possible, you should navigate to the Dataflow Jobs page in the Google Cloud console. The console provides access to detailed job logs and worker logs, which can help you identify the cause of the failure. The graphical interface also allows you to visualize pipeline stages, monitor performance metrics, and pinpoint where the error occurred, making it the most efficient way to diagnose and resolve the issue promptly.

Extract from Google Documentation: From "Monitoring Dataflow Jobs" (https://cloud.google.com/dataflow/docs/guides/monitoring-jobs): "To troubleshoot a failed Dataflow job quickly, go to the Dataflow Jobs page in the Google Cloud Console, where you can view job logs and worker logs to identify errors and their root causes."

Reference: Google Cloud Documentation - "Dataflow Monitoring" (https://cloud.google.com/dataflow/docs/guides/monitoring-jobs).

Question: 28

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.

Answer: D

Explanation:

Using the Looker Scheduler with user attribute filters is the Google-recommended approach to efficiently automate the delivery of a customized dashboard. User attribute filters allow you to dynamically customize the dashboard's content based on the recipient's attributes, ensuring each stakeholder sees data relevant to them. This approach is scalable, does not require creating separate models or custom scripts, and leverages Looker's built-in functionality to automate recurring deliveries effectively.

Question: 29

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.

Answer: D

Explanation:

Using the BigQuery Python client library to query and preprocess data directly in BigQuery and then leveraging BigQueryML to train the churn prediction model is the Google-recommended approach for this scenario. BigQueryML allows you to build machine learning models directly within BigQuery using SQL, eliminating the need to export data or manage additional infrastructure. This minimizes overhead, scales effectively for a dataset as large as 50 PB, and simplifies the end-to-end process of building and training the churn prediction model.

Question: 30

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 dat

a. 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 reimport the cleaned data into BigQuery.

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

Answer: D

Explanation:

Using BigQuery's built-in functions is the most effective and efficient way to clean the dataset directly within BigQuery. BigQuery provides powerful SQL capabilities to handle missing values, remove duplicates, and resolve formatting issues without needing to export data or create complex pipelines. This approach minimizes overhead and leverages the scalability

of BigQuery for large datasets, making it an ideal solution for quickly addressing data quality issues.

Question: 31

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.

Answer: D

Explanation:

Assigning each analyst to a separate project associated with their department and creating a single reservation for each department using BigQuery editions allows for precise cost management. By assigning each project to its department's reservation, you can allocate fixed compute resources and budgets for each department, ensuring that their query costs are predictable and controlled. This approach aligns with your organization's goal of creating a fixed budget for query costs while maintaining departmental separation and accountability.

Question: 32

You manage a web application that stores data in a Cloud SQL database. You need to improve the read performance of the application by ofloading 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.

Answer: D

Explanation:

Enabling automatic backups and creating a read replica of the Cloud SQL instance is the best solution to improve read performance. Read replicas allow you to ofload read traffic from the primary database instance, reducing its load and improving overall performance. This approach is costeffective and easy to implement within Cloud SQL. It ensures that the primary instance focuses on write operations while replicas handle read queries, providing a seamless performance boost with minimal effort.

Question: 33

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 dat

a. 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.

Answer: A

Explanation:

Using Transfer Appliances is the best solution for securely and efficiently moving over 500 TB of data to Cloud Storage within a limited timeframe, especially with network bandwidth below 1 Gbps. Transfer Appliances are physical devices provided by Google Cloud to securely transfer large amounts of data. After copying the data to the appliances, they are shipped back to Google, where the data is uploaded to Cloud Storage. This approach bypasses bandwidth limitations and ensures the data is migrated quickly and securely.

Question: 34

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.

Answer: D

Explanation:
Setting the table partition expiration period to one year using the ALTER TABLE statement is the most efficient and cost-effective approach. This automatically deletes data in partitions older than one year, reducing storage costs without requiring manual intervention or additional queries. It minimizes administrative overhead and ensures compliance with your data retention policy while optimizing storage usage in BigQuery.

Extract from Google Documentation: From "Managing Partitioned Tables in BigQuery" (https://cloud.google.com/bigquery/docs/partitioned-tables#expiration): "Set a partition expiration time using ALTER TABLE to automatically remove partitions older than a specified duration, reducing storage costs efficiently for ingestion-

time partitioned tables." Reference: Google Cloud Documentation - "BigQuery Partitioned Tables" (https://cloud.google.com/bigquery/docs/partitioned-tables).

Question: 35

Your company is migrating their batch transformation pipelines to Google Cloud. You need to choose a solution that supports programmatic transformations using only SQL. You also want the technology to support Git integration for version control of your pipelines. What should you do?

A. Use Cloud Data Fusion pipelines.

B. Use Dataform workflows.

C. Use Dataflow pipelines.

D. Use Cloud Composer operators.

Answer: B

Explanation:

Dataform workflows are the ideal solution for migrating batch transformation pipelines to Google Cloud when you want to perform programmatic transformations using only SQL. Dataform allows you to define SQL-based workflows for data transformations and supports Git integration for version control, enabling collaboration and version tracking of your pipelines. This approach is purpose-built
for SQL-driven data pipeline management and aligns perfectly with your requirements.

The solution must use SQL for transformations and integrate with Git for version control, focusing on batch pipelines.
Let's evaluate:

Option A: Cloud Data Fusion uses a visual UI with plugins, not SQL-only transformations. It lacks native Git integration (requires external tools), missing a key requirement.

Option B: Dataform is a SQL-based workflow tool for BigQuery transformations, defining pipelines as SQLX scripts. It integrates natively with Git for version control, supporting batch ELT processes with minimal overhead.

Option C: Cloud Composer uses Python DAGs and operators, not SQL-only transformations. Git is possible but not

intrinsic to its workflow design.

Option D: Dataflow uses Apache Beam (Python/Java), not SQL, and lacks built-in Git support for pipeline definitions.
Why B is Best: Dataform is purpose-built for SQL-driven ELT in BigQuery, with Git integration baked in (e.g., GitHub sync). It's serverless, aligns with batch migration, and simplifies pipeline management. Extract from Google Documentation: From "Dataform Overview" (https://cloud.google.com/dataform/docs): "Dataform lets you define SQL-based transformation workflows for BigQuery, with native Git integration for version control, enabling teams to manage batch data pipelines programmatically and collaboratively."
Reference: Google Cloud Documentation - "Dataform" (https://cloud.google.com/dataform).

Question: 36

You manage a BigQuery table that is used for critical end-of-month reports. The table is updated weekly with new sales dat

a. You want to prevent data loss and reporting issues if the table is accidentally deleted. What should you do?

A. Configure the time travel duration on the table to be exactly seven days. On deletion, re-create the deleted table solely from the time travel data.

B. Schedule the creation of a new snapshot of the table once a week. On deletion, re-create the deleted table using the snapshot and time travel data.

C. Create a clone of the table. On deletion, re-create the deleted table by copying the content of the clone.

D. Create a view of the table. On deletion, re-create the deleted table from the view and time travel data.

Answer: B

Explanation:

Scheduling the creation of a snapshot of the table weekly ensures that you have a point-in-time backup of the table. In case of accidental deletion, you can re-create the table from the snapshot. Additionally, BigQuery's time travel feature allows you to recover data from up to seven days prior to deletion. Combining snapshots with time travel provides a robust solution for preventing data loss and ensuring reporting continuity for critical tables. This approach minimizes risks while offering flexibility for recovery.

Question: 37

Your organization sends IoT event data to a Pub/Sub topic. Subscriber applications read and perform transformations on the messages before storing them in the data warehouse. During particularly busy times when more data is being written to the topic, you notice that the subscriber applications are not acknowledging messages within the deadline. You need to modify your pipeline to handle these activity spikes and continue to process the messages. What should you do?

A. Retry messages until they are acknowledged.

B. Implement flow control on the subscribers

C. Forward unacknowledged messages to a dead-letter topic.

D. Seek back to the last acknowledged message.

Answer: B

Explanation:
Implementing flow control on the subscribers allows the subscriber applications to manage message processing during activity spikes by controlling the rate at which messages are pulled and processed.
This prevents overwhelming the subscribers and ensures that messages are acknowledged within the deadline. Flow control helps maintain the stability of your pipeline during high-traffic periods without dropping or delaying messages unnecessarily.

Question: 38

You have millions of customer feedback records stored in BigQuery. You want to summarize the data by using the large language model (LLM) Gemini. You need to plan and execute this analysis using the most efficient approach. What should you do?

A. Query the BigQuery table from within a Python notebook, use the Gemini API to summarize the data within the notebook, and store the summaries in BigQuery.

B. Use a BigQuery ML model to pre-process the text data, export the results to Cloud Storage, and use the Gemini API to summarize the pre- processed data.

C. Create a BigQuery Cloud resource connection to a remote model in Vertex AI, and use Gemini to summarize the data.

D. Export the raw BigQuery data to a CSV file, upload it to Cloud Storage, and use the Gemini API to summarize the data.

Answer: C

Explanation:

Creating a BigQuery Cloud resource connection to a remote model in Vertex AI and using Gemini to summarize the data is the most efficient approach. This method allows you to seamlessly integrate BigQuery with the Gemini model via Vertex AI, avoiding the need to export data or perform manual steps. It ensures scalability for large datasets and minimizes data movement, leveraging Google Cloud's ecosystem for efficient data summarization and storage.

Question: 39

You are working on a data pipeline that will validate and clean incoming data before loading it into BigQuery for real-time analysis. You want to ensure that the data validation and cleaning is performed efficiently and can handle high volumes of dat

a. What should you do?

A. Write custom scripts in Python to validate and clean the data outside of Google Cloud. Load the cleaned data into BigQuery.

B. Use Cloud Run functions to trigger data validation and cleaning routines when new data arrives in Cloud Storage.

C. Use Dataflow to create a streaming pipeline that includes validation and transformation steps.

D. Load the raw data into BigQuery using Cloud Storage as a staging area, and use SQL queries in BigQuery to validate and clean the data.

Answer: C

Explanation:

Using Dataflow to create a streaming pipeline that includes validation and transformation steps is the most efficient and scalable approach for real-time analysis. Dataflow is optimized for high-volume data processing and allows you to apply validation and cleaning logic as the data flows through the pipeline. This ensures that only clean, validated data is loaded into

BigQuery, supporting real-time analysis while handling high data volumes effectively.

Question: 40

Your organization needs to implement near real-time analytics for thousands of events arriving each second in Pub/Sub. The incoming messages require transformations. You need to configure a pipeline that processes, transforms, and loads the data into BigQuery while minimizing development time. What should you do?

A. Use a Google-provided Dataflow template to process the Pub/Sub messages, perform transformations, and write the results to BigQuery.

B. Create a Cloud Data Fusion instance and configure Pub/Sub as a source. Use Data Fusion to process the Pub/Sub messages, perform transformations, and write the results to BigQuery.

C. Load the data from Pub/Sub into Cloud Storage using a Cloud Storage subscription. Create a Dataproc cluster, use PySpark to perform transformations in Cloud Storage, and write the results to BigQuery.

D. Use Cloud Run functions to process the Pub/Sub messages, perform transformations, and write the results to BigQuery.

Answer: A

Explanation:

Using a Google-provided Dataflow template is the most efficient and development-friendly approach to implement near real-time analytics for Pub/Sub messages. Dataflow templates are pre-built and optimized for processing streaming data, allowing you to quickly configure and deploy a pipeline with minimal development effort. These templates can handle message ingestion from Pub/Sub, perform necessary transformations, and load the processed data into BigQuery, ensuring scalability and low latency for near real-time analytics.

Question: 41

Your organization needs to store historical customer order dat

a. The data will only be accessed once a month for analysis and must be readily available within a few seconds when it is accessed. You need to choose a storage class that minimizes storage costs while ensuring that the data can be

retrieved quickly. What should you do?

A. Store the data in Cloud Storaqe usinq Nearline storaqe.

B. Store the data in Cloud Storaqe usinq Coldline storaqe.

C. Store the data in Cloud Storage using Standard storage.

D. Store the data in Cloud Storage using Archive storage.

Answer: A

Explanation:

Using Nearline storage in Cloud Storage is the best option for data that is accessed infrequently (such as once a month) but must be readily available within seconds when needed. Nearline offers a balance between low storage costs and quick retrieval times, making it ideal for scenarios like monthly analysis of historical data. It is specifically designed for infrequent access patterns while avoiding the higher retrieval costs and longer access times of Coldline or Archive storage.

Question: 42

You have a Dataflow pipeline that processes website traffic logs stored in Cloud Storage and writes the processed data to BigQuery. You noticed that the pipeline is failing intermittently. You need to troubleshoot the issue. What should you do?

A. Use Cloud Logging to identify error groups in the pipeline's logs. Use Cloud Monitoring to create a dashboard that tracks the number of errors in each group.

B. Use Cloud Logging to create a chart displaying the pipeline's error logs. Use Metrics Explorer to validate the findings from the chart.

C. Use Cloud Logging to view error messages in the pipeline's logs. Use Cloud Monitoring to analyze the pipeline's metrics, such as CPU utilization and memory usage.

D. Use the Dataflow job monitoring interface to check the pipeline's status every hour. Use Cloud Profiler to analyze the pipeline's metrics, such as CPU utilization and memory usage.

Answer: C

Explanation:

To troubleshoot intermittent failures in a Dataflow pipeline, you should use Cloud Logging to view detailed error messages in the pipeline's logs. These logs provide insights into the specific issues causing failures, such as data format errors or resource limitations. Additionally, you should use Cloud Monitoring to analyze the pipeline's metrics, such as CPU utilization, memory usage, and throughput, to identify performance bottlenecks or resource constraints that may contribute to the failures. This approach provides a comprehensive view of the pipeline's health and helps pinpoint the root cause of the intermittent issues.

Question: 43

Your organization's business analysts require near real-time access to streaming dat

a. However, they are reporting that their dashboard queries are loading slowly. After investigating BigQuery query performance, you discover the slow dashboard queries perform several joins and aggregations.

You need to improve the dashboard loading time and ensure that the dashboard data is as up-to-date as possible. What should you do?

A. Disable BiqQuery query result caching.

B. Modify the schema to use parameterized data types.

C. Create a scheduled query to calculate and store intermediate results.

D. Create materialized views.

Answer: D

Explanation:

Creating materialized views is the best solution to improve dashboard loading time while ensuring that the data is as up-to-

date as possible. Materialized views precompute and cache the results of complex joins and aggregations, significantly reducing query execution time for dashboards. They also automatically update as the underlying data changes, ensuring near real-time access to fresh data. This approach optimizes query performance and provides an efficient and scalable solution for streaming data dashboards.

For near real-time dashboards with slow queries involving joins and aggregations, Google recommends materialized views in BigQuery to precompute and refresh results automatically.

Option A: Scheduled queries compute intermediate results periodically, but they aren't near real-time and require manual scheduling, adding latency.

Option B: Parameterized data types don't address query performance for joins/aggregations; they're for query flexibility, not optimization.

Option C: Disabling caching worsens performance by forcing full recomputation each time, contrary to the goal.

Option D: Materialized views precompute joins and aggregations, auto-refresh with streaming data (near real-time), and optimize query performance for dashboards.

Extract from Google Documentation: From "Using Materialized Views in BigQuery" (https://cloud.google.com/bigquery/docs/materialized-views): "Materialized views are precomputed views that periodically cache the results of a query, improving performance for complex queries like joins and aggregations, and they support automatic refresh for near real-time data."

Reference: Google Cloud Documentation - "Materialized Views" (https://cloud.google.com/bigquery/docs/materialized-views).

Question: 44

You need to create a data pipeline that streams event information from applications in multiple Google Cloud regions into BigQuery for near real-time analysis. The data requires transformation before loading. You want to create the pipeline using a visual interface. What should you do?

A. Push event information to a Pub/Sub topic. Create a Dataflow job using the Dataflow job builder.

B. Push event information to a Pub/Sub topic. Create a Cloud Run function to subscribe to the Pub/Sub topic, apply transformations, and insert the data into BigQuery.

C. Push event information to a Pub/Sub topic. Create a BigQuery subscription in Pub/Sub.

D. Push event information to Cloud Storage, and create an external table in BigQuery. Create a BigQuery scheduled job that executes once each day to apply transformations.

Answer: A

Explanation:

Pushing event information to a Pub/Sub topic and then creating a Dataflow job using the Dataflow job builder is the most suitable solution. The Dataflow job builder provides a visual interface to design pipelines, allowing you to define transformations and load data into BigQuery. This approach is ideal for streaming data pipelines that require near real-time transformations and analysis. It ensures scalability across multiple regions and integrates seamlessly with Pub/Sub for event ingestion and BigQuery for analysis.

The best solution for creating a data pipeline with a visual interface for streaming event information from multiple Google Cloud regions into BigQuery for near real-time analysis with transformations is A . Push event information to a Pub/Sub topic. Create a Dataflow job using the Dataflow job builder.

Here's why:

Pub/Sub and Dataflow:

Pub/Sub is ideal for real-time message ingestion, especially from multiple regions.

Dataflow, particularly with the Dataflow job builder, provides a visual interface for creating data pipelines that can perform real-time stream processing and transformations.

The Dataflow job builder allows creating pipelines with visual tools, fulfilling the requirement of a visual interface.

Dataflow is built for real time streaming and applying transformations.

Let's break down why the other options are less suitable:

B . Push event information to Cloud Storage, and create an external table in BigQuery. Create a BigQuery scheduled job that executes once each day to apply transformations:

This is a batch processing approach, not real-time.

Cloud Storage and scheduled jobs are not designed for near real-time analysis.

This does not meet the real time requirement of the question.

C . Push event information to a Pub/Sub topic. Create a Cloud Run function to subscribe to the Pub/Sub topic, apply transformations, and insert the data into BigQuery:

While Cloud Run can handle transformations, it requires more coding and is less scalable and manageable than Dataflow for complex streaming pipelines.

Cloud run does not provide a visual interface.

D . Push event information to a Pub/Sub topic. Create a BigQuery subscription in Pub/Sub:

BigQuery subscriptions in Pub/Sub are for direct loading of Pub/Sub messages into BigQuery, without the ability to perform transformations.

This option does not provide any transformation functionality.

Therefore, Pub/Sub for ingestion and Dataflow with its job builder for visual pipeline creation and transformations is the most appropriate solution.

Question: 45

You work for an online retail company. Your company collects customer purchase data in CSV files and pushes them to Cloud Storage every 10 minutes. The data needs to be transformed and loaded into BigQuery for analysis. The transformation involves cleaning the data, removing duplicates, and enriching it with product information from a separate table in BigQuery. You need to implement a low-overhead solution that initiates data processing as soon as the files are loaded into Cloud Storage. What should you do?

A. Use Cloud Composer sensors to detect files loading in Cloud Storage. Create a Dataproc cluster, and use a Composer task to execute a job on the cluster to process and load the data into BigQuery.

B. Schedule a direct acyclic graph (DAG) in Cloud Composer to run hourly to batch load the data from Cloud Storage to BigQuery, and process the data in BigQuery using SQL.

C. Use Dataflow to implement a streaming pipeline using an OBJECT_FINALIZE notification from Pub/Sub to read the data from Cloud Storage, perform the transformations, and write the data to BigQuery.

D. Create a Cloud Data Fusion job to process and load the data from Cloud Storage into BigQuery. Create an OBJECT_FINALIZE notification in Pub/Sub, and trigger a Cloud Run function to start the Cloud Data Fusion job as soon as new files are loaded.

Answer: C

Explanation:

Using Dataflow to implement a streaming pipeline triggered by an OBJECT_FINALIZE notification from Pub/Sub is the best solution. This approach automatically starts the data processing as soon as new files are uploaded to Cloud Storage, ensuring low latency. Dataflow can handle the data cleaning, deduplication, and enrichment with product information from the BigQuery table in a scalable and efficient manner. This solution minimizes overhead, as Dataflow is a fully managed service,

and it is well-suited for real-time or near-real-time data pipelines.

Extract from Google Documentation: From "Dataflow: Streaming from Cloud Storage" (https://cloud.google.com/dataflow/docs/guides/streaming-data): "Use Dataflow with Pub/Sub notifications (e.g., object_finalize) to process files from Cloud Storage in near real-time, applying transformations and loading results into BigQuery with low operational overhead." Reference: Google Cloud Documentation - "Dataflow Streaming" (https://cloud.google.com/dataflow/docs).

Question: 46

You work for a home insurance company. You are frequently asked to create and save risk reports with charts for specific areas using a publicly available storm event dataset. You want to be able to quickly create and re-run risk reports when new data becomes available. What should you do?

A. Export the storm event dataset as a CSV file. Import the file to Google Sheets, and use cell data in the worksheets to create charts.

B. Copy the storm event dataset into your BigQuery project. Use BigQuery Studio to query and visualize the data in Looker Studio.

C. Reference and query the storm event dataset using SQL in BigQuery Studio. Export the results to Google Sheets, and use cell data in the worksheets to create charts.

D. Reference and query the storm event dataset using SQL in a Colab Enterprise notebook. Display the table results and document with Markdown, and use Matplotlib to create charts.

Answer: B

Explanation:

Copying the storm event dataset into your BigQuery project and using BigQuery Studio to query and visualize the data in Looker Studio is the best approach. This solution allows you to create reusable and automated workflows for generating risk reports. BigQuery handles the querying efficiently, and Looker Studio provides powerful tools for creating and sharing dynamic charts and dashboards. This setup ensures that reports can be easily re-run with updated data, minimizing manual effort and providing a scalable, interactive solution for visualizing risk reports.

Question: 47

Your company currently uses an on-premises network file system (NFS) and is migrating data to Google Cloud. You want to be able to control how much bandwidth is used by the data migration while capturing detailed reporting on the migration status. What should you do?

A. Use a Transfer Appliance.

B. Use Cloud Storage FUSE.

C. Use Storage Transfer Service.

D. Use gcloud storage commands.

Answer: C

Explanation:

Using the Storage Transfer Service is the best solution for migrating data from an on-premises NFS to Google Cloud. This service allows you to control bandwidth usage by configuring transfer speed limits and provides detailed reporting on the migration status. Storage Transfer Service is specifically designed for large-scale data migrations and supports scheduling, monitoring, and error handling, making it an efficient and reliable choice for your use case.

Question: 48

You are a Looker analyst. You need to add a new field to your Looker report that generates SQL that will run against your company's database. You do not have the Develop permission. What should you do?

A. Create a new field in the LookML layer, refresh your report, and select your new field from the field picker.

B. Create a calculated field using the Add a field option in Looker Studio, and add it to your report.

C. Create a table calculation from the field picker in Looker, and add it to your report.

D. Create a custom field from the field picker in Looker, and add it to your report.

Answer: D

Explanation:

Creating a custom field from the field picker in Looker allows you to add new fields to your report without requiring the Develop permission. Custom fields are created directly in the Looker UI, enabling you to define calculations or transformations that generate SQL for the database query. This approach is user-friendly and does not require access to the LookML layer, making it the appropriate choice for your situation.

Question: 49

Your organization's ecommerce website collects user activity logs using a Pub/Sub topic. Your organization's leadership team wants a dashboard that contains aggregated user engagement metrics. You need to create a solution that transforms the user activity logs into aggregated metrics, while ensuring that the raw data can be easily queried. What should you do?

A. Create a Dataflow subscription to the Pub/Sub topic, and transform the activity logs. Load the transformed data into a BigQuery table for reporting.

B. Create an event-driven Cloud Run function to trigger a data transformation pipeline to run. Load the transformed activity logs into a BigQuery table for reporting.

C. Create a Cloud Storage subscription to the Pub/Sub topic. Load the activity logs into a bucket using the Avro file format. Use Dataflow to transform the data, and load it into a BigQuery table for reporting.

D. Create a BigQuery subscription to the Pub/Sub topic, and load the activity logs into the table. Create a materialized view in BigQuery using SQL to transform the data for reporting

Answer: A

Explanation:

Using Dataflow to subscribe to the Pub/Sub topic and transform the activity logs is the best approach for this scenario. Dataflow is a managed service designed for processing and transforming streaming data in real time. It allows you to

aggregate metrics from the raw activity logs efficiently and load the transformed data into a BigQuery table for reporting. This solution ensures scalability, supports realtime processing, and enables querying of both raw and aggregated data in BigQuery, providing the flexibility and insights needed for the dashboard.

Question: 50

You are constructing a data pipeline to process sensitive customer data stored in a Cloud Storage bucket. You need to ensure that this data remains accessible, even in the event of a single-zone outage. What should you do?

A. Set up a Cloud CDN in front of the bucket.

B. Enable Object Versioning on the bucket.

C. Store the data in a multi-region bucket.

D. Store the data in Nearline storaqe.

Answer: C

Explanation:
Storing the data in a multi-region bucket ensures high availability and durability, even in the event of a single-zone outage. Multi-region buckets replicate data across multiple locations within the selected region, providing resilience against zone-level failures and ensuring that the data remains accessible. This approach is particularly suitable for sensitive customer data that must remain available without interruptions.

A single-zone outage requires high availability across zones or regions. Cloud Storage offers locationbased redundancy options:

Option A: Cloud CDN caches content for web delivery but doesn't protect against underlying storage outages—it's for performance, not availability of the source data.

Option B: Object Versioning retains old versions of objects, protecting against overwrites or deletions, but doesn't ensure availability during a zone failure (still tied to one location).

Option C: Multi-region buckets (e.g., us or eu) replicate data across multiple regions, ensuring accessibility even if a single zone or region fails. This provides the highest availability for sensitive data in a pipeline.

Option D: Nearline storage is a low-cost class with lower availability (single-region by default), not designed for outage resilience.

Why C is Best: Multi-region storage offers geo-redundancy, guaranteeing data access during outages with no additional configuration. It's the simplest, most reliable choice for pipeline continuity. Extract from Google Documentation: From "Cloud Storage Bucket Locations" (https://cloud.google.com/storage/docs/locations): "Multi-region buckets replicate data across multiple regions within a geography, providing high availability and durability, ensuring access even if a single zone or region experiences an outage." Reference: Google Cloud Documentation - "Cloud Storage Locations" (https://cloud.google.com/storage/docs/locations).

Question: 51

Your retail company collects customer data from various sources:

- Online transactions: Stored in a MySQL database
- Customer feedback: Stored as text files on a company server
- Social media activity: Streamed in real-time from social media platforms

You are designing a data pipeline to extract this dat

a. Which Google Cloud storage system(s) should you select for further analysis and ML model training?

A. 1. Online transactions: Cloud Storage

2. Customer feedback: Cloud Storage

3. Social media activity: Cloud Storage

B. 1. Online transactions: BigQuery

2. Customer feedback: Cloud Storage

3. Social media activity: BigQuery

C. 1. Online transactions: Bigtable

2. Customer feedback: Cloud Storage

3. Social media activity: CloudSQL for MySQL

D. 1. Online transactions: Cloud SQL for MySQL

2. Customer feedback: BigQuery

3. Social media activity: Cloud Storage

Answer: B

Explanation:

Online transactions: Storing the transactional data in BigQuery is ideal because BigQuery is a serverless data warehouse optimized for querying and analyzing structured data at scale. It supports SQL queries and is suitable for structured transactional data.

Customer feedback: Storing customer feedback in Cloud Storage is appropriate as it allows you to store unstructured text files reliably and at a low cost. Cloud Storage also integrates well with data processing and ML tools for further analysis.

Social media activity: Storing real-time social media activity in BigQuery is optimal because BigQuery supports streaming inserts, enabling real-time ingestion and analysis of data. This allows immediate analysis and integration into dashboards or ML pipelines.

Question: 52

Your company uses Looker as its primary business intelligence platform. You want to use LookML to visualize the profit margin for each of your company's products in your Looker Explores and dashboards. You need to implement a solution quickly and efficiently. What should you do?

A. Create a derived table that pre-calculates the profit margin for each product, and include it in the Looker model.

B. Define a new measure that calculates the profit margin by using the existing revenue and cost fields.

C. Create a new dimension that categorizes products based on their profit margin ranges (e.g., high, medium, low).

D. Apply a filter to only show products with a positive profit margin.

Answer: B

Explanation:

Defining a new measure in LookML to calculate the profit margin using the existing revenue and cost fields is the most

efficient and straightforward solution. This approach allows you to dynamically compute the profit margin directly within your Looker Explores and dashboards without needing to pre-calculate or create additional tables. The measure can be defined using LookML syntax, such as:

```
measure: profit_margin {
 type: number
 sql: (revenue - cost) / revenue ;;
 value_format: "0.0%"
}
```

This method is quick to implement and integrates seamlessly into your existing Looker model, enabling accurate visualization of profit margins across your products.

Question: 53

You are a data analyst working with sensitive customer data in BigQuery. You need to ensure that only authorized personnel within your organization can query this data, while following the principle of least privilege. What should you do?

A. Enable access control by using IAM roles.
B. Update dataset privileges by using the SQL GRANT statement.
C. Export the data to Cloud Storage, and use signed URLs to authorize access.
D. Encrypt the data by using customer-managed encryption keys (CMEK).

Answer: A

Explanation:

Comprehensive and Detailed In-Depth

BigQuery uses IAM for access control, adhering to least privilege by granting only necessary permissions.

Option A: IAM roles (e.g., roles/bigquery.dataViewer for read-only) restrict query access to authorized users, aligning with Google's security best practices.

Option B: BigQuery doesn't support SQL GRANT for dataset privileges; access is managed via IAM or authorized views.

Option C: Exporting to Cloud Storage with signed URLs bypasses BigQuery's native controls and adds complexity.

Option D: CMEK encrypts data at rest but doesn't control query access.

Extract from Google Documentation: From "Controlling Access to BigQuery" (https://cloud.google.com/bigquery/docs/access-control): "Use IAM to manage access to BigQuery resources. Predefined roles like roles/bigquery.dataViewer allow you to grant query access while following the principle of least privilege."

Reference: Google Cloud Documentation - "BigQuery IAM" (https://cloud.google.com/bigquery/docs/access-control).

Question: 54

Your organization stores highly personal data in BigQuery and needs to comply with strict data privacy regulations. You need to ensure that sensitive data values are rendered unreadable whenever an employee leaves the organization. What should you do?

A. Use AEAD functions and delete keys when employees leave the organization.

B. Use dynamic data masking and revoke viewer permissions when employees leave the organization.

C. Use customer-managed encryption keys (CMEK) and delete keys when employees leave the organization.

D. Use column-level access controls with policy tags and revoke viewer permissions when employees leave the organization.

Answer: C

Explanation:

Using customer-managed encryption keys (CMEK) allows you to encrypt highly sensitive data in BigQuery with encryption keys managed by your organization. When an employee leaves the organization, you can render the data unreadable by deleting or revoking access to the encryption keys associated with the data. This approach ensures compliance with strict data privacy regulations by making the data inaccessible without the encryption keys, providing strong control over data access and security.

Question: 55

You used BigQuery ML to build a customer purchase propensity model six months ago. You want to compare the current serving data with the historical serving data to determine whether you need to retrain the model. What should you do?

A. Compare the two different models.

B. Evaluate the data skewness.

C. Evaluate data drift.

D. Compare the confusion matrix.

Answer: C

Explanation:

Evaluating data drift involves analyzing changes in the distribution of the current serving data compared to the historical data used to train the model. If significant drift is detected, it indicates that the data patterns have changed over time, which can impact the model's performance. This analysis helps determine whether retraining the model is necessary to ensure its predictions remain accurate and relevant. Data drift evaluation is a standard approach for monitoring machine learning models over time.

Question: 56

Your company uses Looker to visualize and analyze sales dat

a. You need to create a dashboard that displays sales metrics, such as sales by region, product category, and time period. Each metric relies on its own set of attributes distributed across several tables. You need to provide users the ability to filter the data by specific sales representatives and view individual transactions. You want to follow the Google-recommended approach. What should you do?

A. Create multiple Explores, each focusing on each sales metric. Link the Explores together in a dashboard using drill-down functionality.

B. Use BigQuery to create multiple materialized views, each focusing on a specific sales metric. Build the dashboard using these views.

C. Create a single Explore with all sales metrics. Build the dashboard using this Explore.

D. Use Looker's custom visualization capabilities to create a single visualization that displays all the sales metrics with filtering and drill-down functionality.

Answer: C

Explanation:

Creating a single Explore with all the sales metrics is the Google-recommended approach. This Explore should be designed to include all relevant attributes and dimensions, enabling users to analyze sales data by region, product category, time period, and other filters like sales representatives. With a well-structured Explore, you can efficiently build a dashboard that supports filtering and drill-down functionality. This approach simplifies maintenance, provides a consistent data model, and ensures users have the flexibility to interact with and analyze the data seamlessly within a unified framework.

Looker's recommended approach for dashboards is a single, unified Explore for scalability and usability, supporting filters and drill-downs.

Option A: Materialized views in BigQuery optimize queries but bypass Looker's modeling layer, reducing flexibility.

Option B: Custom visualizations are for specific rendering, not multi-metric dashboards with filtering/drill-down.

Option C: Multiple Explores fragment the data model, complicating dashboard cohesion and maintenance.

Option D: A single Explore joins tables in LookML, enabling all metrics, filters (e.g., sales rep), and drill-downs to transactions, per Looker's best practices.

Extract from Google Documentation: From "Building Dashboards in Looker" (https://cloud.google.com/looker/docs/creating-dashboards): "Create a single Explore that models your data relationships, allowing users to filter and drill into details like individual transactions directly within a dashboard."

Reference: Looker Documentation - "Explores and Dashboards" (https://cloud.google.com/looker/docs).

Question: 57

Your company's ecommerce website collects product reviews from customers. The reviews are loaded as CSV files daily to a Cloud Storage bucket. The reviews are in multiple languages and need to be translated to Spanish. You need to configure a pipeline that is serverless, efficient, and requires minimal maintenance. What should you do?

A. Load the data into BigQuery using Dataproc. Use Apache Spark to translate the reviews by invoking the Cloud Translation API. Set BigQuery as the sink.U

B. Use a Dataflow templates pipeline to translate the reviews using the Cloud Translation API. Set BigQuery as the sink.

C. Load the data into BigQuery using a Cloud Run function. Use the BigQuery ML create model statement to train a translation model. Use the model to translate the product reviews within BigQuery.

D. Load the data into BigQuery using a Cloud Run function. Create a BigQuery remote function that invokes the Cloud Translation API. Use a scheduled query to translate new reviews.

Answer: D

Explanation:

Loading the data into BigQuery using a Cloud Run function and creating a BigQuery remote function that invokes the Cloud Translation API is a serverless and efficient approach. With this setup, you can use a scheduled query in BigQuery to invoke the remote function and translate new product reviews on a regular basis. This solution requires minimal maintenance, as BigQuery handles storage and querying, and the Cloud Translation API provides accurate translations without the need for custom ML model development.

Question: 58

You have a Dataproc cluster that performs batch processing on data stored in Cloud Storage. You need to schedule a daily Spark job to generate a report that will be emailed to stakeholders. You need a fully-managed solution that is easy to implement and minimizes complexity. What should you do?

A. Use Cloud Composer to orchestrate the Spark job and email the report.

B. Use Dataproc workflow templates to define and schedule the Spark job, and to email the report.

C. Use Cloud Run functions to trigger the Spark job and email the report.

D. Use Cloud Scheduler to trigger the Spark job. and use Cloud Run functions to email the report.

Answer: B

Explanation:

Using Dataproc workflow templates is a fully-managed and straightforward solution for defining and scheduling your Spark

job on a Dataproc cluster. Workflow templates allow you to automate the execution of Spark jobs with predefined steps, including data processing and report generation. You can integrate email notifications by adding a step to the workflow that sends the report using tools like a Cloud Function or external email service. This approach minimizes complexity while leveraging Dataproc's managed capabilities for batch processing.

Question: 59

Your organization has highly sensitive data that gets updated once a day and is stored across multiple datasets in BigQuery. You need to provide a new data analyst access to query specific data in BigQuery while preventing access to sensitive dat

a. What should you do?

A. Grant the data analyst the BigQuery Job User IAM role in the Google Cloud project.

B. Create a materialized view with the limited data in a new dataset. Grant the data analyst BigQuery Data Viewer IAM role in the dataset and the BigQuery Job User IAM role in the Google Cloud project.

C. Create a new Google Cloud project, and copy the limited data into a BigQuery table. Grant the data analyst the BigQuery Data Owner IAM role in the new Google Cloud project.

D. Grant the data analyst the BigQuery Data Viewer IAM role in the Google Cloud project.

Answer: B

Explanation:

Creating a materialized view with the limited data in a new dataset and granting the data analyst the BigQuery Data Viewer role on the dataset and the BigQuery Job User role in the project ensures that the analyst can query only the non-sensitive data without access to sensitive datasets. Materialized views allow you to predefine what subset of data is visible, providing a secure and efficient way to control access while maintaining compliance with data governance policies. This approach follows the principle of least privilege while meeting the requirements.

Question: 60

You are a database administrator managing sales transaction data by region stored in a BigQuery table. You need to ensure that each sales representative can only see the transactions in their region. What should you do?

A. Add a policy tag in BigQuery.

B. Create a row-level access policy.

C. Create a data masking rule.

D. Grant the appropriate 1AM permissions on the dataset.

Answer: B

Explanation:

Creating a row-level access policy in BigQuery ensures that each sales representative can see only the transactions relevant to their region. Row-level access policies allow you to define fine-grained access control by filtering rows based on specific conditions, such as matching the sales representative's region. This approach enforces security while providing tailored data access, aligning with the principle of least privilege.

Extract from Google Documentation: From "Row-Level Security in BigQuery" (https://cloud.google.com/bigquery/docs/row-level-security): "Row-level access policies let you restrict access to specific rows in a table based on a filter condition, such as a user's region, providing fine-grained control over data visibility without creating separate tables or views."

Reference: Google Cloud Documentation - "BigQuery Row-Level Security" (https://cloud.google.com/bigquery/docs/row-level-security).

Question: 61

Your company's customer support audio files are stored in a Cloud Storage bucket. You plan to analyze the audio files' metadata and file content within BigQuery to create inference by using BigQuery ML. You need to create a corresponding table in BigQuery that represents the bucket containing the audio files. What should you do?

A. Create an external table.

B. Create a temporary table.

C. Create a native table.

D. Create an object table.

Answer: D

Explanation:

To analyze audio files stored in a Cloud Storage bucket and represent them in BigQuery, you should create an object table. Object tables in BigQuery are designed to represent objects stored in Cloud Storage, including their metadata. This enables you to query the metadata of audio files directly from BigQuery without duplicating the data. Once the object table is created, you can use it in conjunction with other BigQuery ML workflows for inference and analysis.

Question: 62

You work for a financial services company that handles highly sensitive dat
a. Due to regulatory requirements, your company is required to have complete and manual control of data encryption. Which type of keys should you recommend to use for data storage?

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

B. Use a dedicated third-party key management system (KMS) chosen by the company.

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

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

Answer: A

Explanation:

For regulatory requirements that mandate complete and manual control of data encryption, you should use customer-supplied encryption keys (CSEK). With CSEK, your company provides the encryption keys for data storage, and Google Cloud does not store or manage these keys. This approach ensures that your organization retains full control and responsibility over the encryption process, meeting strict regulatory compliance requirements.

Question: 63

Your team needs to analyze large datasets stored in BigQuery to identify trends in user behavior. The analysis will involve complex statistical calculations, Python packages, and visualizations. You need to recommend a managed collaborative

environment to develop and share the analysis. What should you recommend?

A. Create a Colab Enterprise notebook and connect the notebook to BigQuery. Share the notebook with your team. Analyze the data and generate visualizations in Colab Enterprise.

B. Create a statistical model by using BigQuery ML. Share the query with your team. Analyze the data and generate visualizations in Looker Studio.

C. Create a Looker Studio dashboard and connect the dashboard to BigQuery. Share the dashboard with your team. Analyze the data and generate visualizations in Looker Studio.

D. Connect Google Sheets to BigQuery by using Connected Sheets. Share the Google Sheet with your team. Analyze the data and generate visualizations in Gooqle Sheets.

Answer: A

Explanation:

Using a Colab Enterprise notebook connected to BigQuery provides a managed, collaborative environment ideal for complex statistical calculations, Python packages, and visualizations. Colab Enterprise supports Python libraries for advanced analytics and offers seamless integration with BigQuery for querying large datasets. It allows teams to collaboratively develop and share analyses while taking advantage of its visualization capabilities. This approach is particularly suitable for tasks involving sophisticated computations and custom visualizations.

Question: 64

Your organization has several datasets in BigQuery. The datasets need to be shared with your external partners so that they can run SQL queries without needing to copy the data to their own projects. You have organized each partner's data in its own BigQuery dataset. Each partner should be able to access only their dat
a. You want to share the data while following Google-recommended practices. What should you do?

A. Use Analytics Hub to create a listing on a private data exchange for each partner dataset. Allow each partner to subscribe to their respective listings.

B. Create a Dataflow job that reads from each BigQuery dataset and pushes the data into a dedicated Pub/Sub topic for each partner. Grant each partner the pubsub. subscriber IAM role.

C. Export the BigQuery data to a Cloud Storage bucket. Grant the partners the storage.objectUser IAM role on the bucket.

D. Grant the partners the bigquery.user IAM role on the BigQuery project.

Answer: A

Explanation:

Using Analytics Hub to create a listing on a private data exchange for each partner dataset is the Google-recommended practice for securely sharing BigQuery data with external partners. Analytics Hub allows you to manage data sharing at scale, enabling partners to query datasets directly without needing to copy the data into their own projects. By creating separate listings for each partner dataset and allowing only the respective partner to subscribe, you ensure that partners can access only their specific data, adhering to the principle of least privilege. This approach is secure, efficient, and designed for scenarios involving external data sharing.

Question: 65

Your organization has decided to migrate their existing enterprise data warehouse to BigQuery. The existing data pipeline tools already support connectors to BigQuery. You need to identify a data migration approach that optimizes migration speed. What should you do?

A. Create a temporary file system to facilitate data transfer from the existing environment to Cloud Storage. Use Storage Transfer Service to migrate the data into BigQuery.

B. Use the Cloud Data Fusion web interface to build data pipelines. Create a directed acyclic graph (DAG) that facilitates pipeline orchestration.

C. Use the existing data pipeline tool's BigQuery connector to reconfigure the data mapping.

D. Use the BigQuery Data Transfer Service to recreate the data pipeline and migrate the data into BigQuery.

Answer: C

Explanation:

Since your existing data pipeline tools already support connectors to BigQuery, the most efficient approach is to use the existing data pipeline tool's BigQuery connector to reconfigure the data mapping. This leverages your current tools, reducing migration complexity and setup time, while optimizing migration speed. By reconfiguring the data mapping within the existing pipeline, you can seamlessly direct the data into BigQuery without needing additional services or intermediary steps.

Question: 66

Your organization uses scheduled queries to perform transformations on data stored in BigQuery. You discover that one of your scheduled queries has failed. You need to troubleshoot the issue as quickly as possible. What should you do?

A. Navigate to the Logs Explorer page in Cloud Logging. Use filters to find the failed job, and analyze the error details.

B. Set up a log sink using the gcloud CLI to export BigQuery audit logs to BigQuery. Query those logs to identify the error associated with the failed job ID.

C. Request access from your admin to the BigQuery information_schema. Query the jobs view with the failed job ID, and analyze error details.

D. Navigate to the Scheduled queries page in the Google Cloud console. Select the failed job, and analyze the error details.

Answer: D

Explanation:

Question: 67

Your team uses the Google Ads platform to visualize metrics. You want to export the data to BigQuery to get more granular insights. You need to execute a one-time transfer of historical data and automatically update data daily. You want a solution that is low-code, serverless, and requires minimal maintenance. What should you do?

A. Export the historical data to BigQuery by using BigQuery Data Transfer Service. Use Cloud Composer for daily automation.

B. Export the historical data to Cloud Storage by using Storage Transfer Service. Use Pub/Sub to trigger a Dataflow template that loads data for daily automation.

C. Export the historical data as a CSV file. Import the file into BigQuery for analysis. Use Cloud Composer for daily automation.

D. Export the historical data to BigQuery by using BigQuery Data Transfer Service. Use BigQuery Data Transfer Service for daily automation.

Answer: D

Explanation:

Question: 68

Your organization has a BigQuery dataset that contains sensitive employee information such as salaries and performance reviews. The payroll specialist in the HR department needs to have continuous access to aggregated performance data, but they do not need continuous access to other sensitive dat

E. You need to grant the payroll specialist access to the performance data without granting them access to the entire dataset using the simplest and most secure approach. What should you do?

A. Use authorized views to share query results with the payroll specialist.

B. Create row-level and column-level permissions and policies on the table that contains performance data in the dataset. Provide the payroll specialist with the appropriate permission set.

C. Create a table with the aggregated performance data. Use table-level permissions to grant access to the payroll specialist.

D. Create a SQL query with the aggregated performance data. Export the results to an Avro file in a Cloud Storage bucket. Share the bucket with the payroll specialist.

Answer: A

Explanation:

Using authorized views is the simplest and most secure way to grant the payroll specialist access to aggregated performance data without exposing the entire dataset. Authorized views allow you to create a view in BigQuery that contains only the query results for the aggregated performance data. The payroll specialist can query the view without being granted access to the underlying sensitive data. This approach ensures security, adheres to the principle of least privilege, and eliminates the

need to manage complex row-level or column-level permissions.

Question: 69

You work for a global financial services company that trades stocks 24/7. You have a Cloud SGL for PostgreSQL user database. You need to identify a solution that ensures that the database is continuously operational, minimizes downtime, and will not lose any data in the event of a zonal outage. What should you do?

A. Continuously back up the Cloud SGL instance to Cloud Storage. Create a Compute Engine instance with PostgreSCL in a different region. Restore the backup in the Compute Engine instance if a failure occurs.

B. Create a read replica in another region. Promote the replica to primary if a failure occurs.

C. Configure and create a high-availability Cloud SQL instance with the primary instance in zone A and a secondary instance in any zone other than zone A.

D. Create a read replica in the same region but in a different zone.

Answer: C

Explanation:

Configuring a high-availability (HA) Cloud SQL instance ensures continuous operation, minimizes downtime, and prevents data loss in the event of a zonal outage. In this setup, the primary instance is located in one zone (e.g., zone A), and a synchronous secondary instance is located in a different zone within the same region. This configuration ensures that all data is replicated to the secondary instance in real-time. In the event of a failure in the primary zone, the system automatically promotes the secondary instance to primary, ensuring seamless failover with no data loss and minimal downtime. This is the recommended approach for mission-critical, highly available databases.

Question: 70

You are migrating data from a legacy on-premises MySQL database to Google Cloud. The database contains various tables with different data types and sizes, including large tables with millions of rows and transactional dat

a. You need to migrate this data while maintaining data integrity, and minimizing downtime and cost. What should you do?

A. Set up a Cloud Composer environment to orchestrate a custom data pipeline. Use a Python script to extract data from

the MySQL database and load it to MySQL on Compute Engine.

B. Export the MySQL database to CSV files, transfer the files to Cloud Storage by using Storage Transfer Service, and load the files into a Cloud SQL for MySQL instance.

C. Use Database Migration Service to replicate the MySQL database to a Cloud SQL for MySQL instance.

D. Use Cloud Data Fusion to migrate the MySQL database to MySQL on Compute Engine.

Answer: C

Explanation:

Using Database Migration Service (DMS) to replicate the MySQL database to a Cloud SQL for MySQL instance is the best approach. DMS is a fully managed service designed for migrating databases to Google Cloud with minimal downtime and cost. It supports continuous data replication, ensuring data integrity during the migration process, and handles schema and data transfer efficiently. This solution is particularly suited for large tables and transactional data, as it maintains real-time synchronization between the source and target databases, minimizing downtime for the migration.

Question: 71

You created a customer support application that sends several forms of data to Google Cloud. Your application is sending:

1. Audio files from phone interactions with support agents that will be accessed during trainings.
2. CSV files of users' personally identifiable information (PII) that will be analyzed with SQL.
3. A large volume of small document files that will power other applications.

You need to select the appropriate tool for each data type given the required use case, while following Google-recommended practices. Which should you choose?

A.

1. Cloud Storage
2. CloudSQL for PostgreSQL
3. Bigtable

B.

1. Filestore
2. Cloud SQL for PostgreSQL
3. Datastore

C.

1. Cloud Storage
2. BigQuery
3. Firestore

D.

1. Filestore
2. Bigtable
3. BigQuery

Answer: C

Explanation:

Audio files from phone interactions: Use Cloud Storage. Cloud Storage is ideal for storing large binary objects like audio files, offering scalability and easy accessibility for training purposes.

CSV files of users' personally identifiable information (PII): Use BigQuery. BigQuery is a serverless data warehouse optimized for analyzing structured data, such as CSV files, using SQL. It ensures compliance with PII handling through access controls and data encryption.

A large volume of small document files: Use Firestore. Firestore is a scalable NoSQL database designed for applications requiring fast, real-time interactions and structured document storage, making it suitable for powering other applications.

Question: 72

You are designing an application that will interact with several BigQuery datasets. You need to grant the application's service account permissions that allow it to query and update tables within the datasets, and list all datasets in a project within your application. You want to follow the principle of least privilege. Which pre-defined IAM role(s) should you apply to the service account?

A. roles/bigquery.jobUser and roles/bigquery.dataOwner

B. roles/bigquery.connectionUser and roles/bigquery.dataViewer

C. roles/bigquery.admin

D. roles/bigquery.user and roles/bigquery.filteredDataViewer

Answer: A

Explanation:

roles/bigquery.jobUser:

This role allows a user or service account to run BigQuery jobs, including queries. This is necessary for the application to interact with and query the tables.

From Google Cloud documentation: "BigQuery Job User can run BigQuery jobs, including queries, load jobs, export jobs, and copy jobs."

roles/bigquery.dataOwner:

This role grants full control over BigQuery datasets and tables. It allows the service account to update tables, which is a requirement of the application.

From Google Cloud documentation: "BigQuery Data Owner can create, delete, and modify BigQuery datasets and tables. BigQuery Data Owner can also view data and run queries."

Why other options are incorrect:

B . roles/bigquery.connectionUser and roles/bigquery.dataViewer:

roles/bigquery.connectionUser is used for external connections, which is not required for this task.
roles/bigquery.dataViewer only allows viewing data, not updating it.

C . roles/bigquery.admin:

roles/bigquery.admin grants excessive permissions. Following the principle of least privilege, this role is too broad.

D . roles/bigquery.user and roles/bigquery.filteredDataViewer:

roles/bigquery.user grants the ability to run queries, but not the ability to modify data. roles/bigquery.filteredDataViewer only provides permission to view filtered data, which is not sufficient for updating tables.

Principle of Least Privilege:

The principle of least privilege is a security concept that states that a user or service account should be granted only the permissions necessary to perform its intended tasks.

By assigning roles/bigquery.jobUser and roles/bigquery.dataOwner, we provide the application with the exact permissions it needs without granting unnecessary access.

Google Cloud Documentation Reference:

BigQuery IAM roles: https://cloud.google.com/bigquery/docs/access-control-basic-roles

IAM best practices: https://cloud.google.com/iam/docs/best-practices-for-using-iam

Question: 73

Your company is setting up an enterprise business intelligence platform. You need to limit data access between many different teams while following the Google-recommended approach. What should you do first?

A. Create a separate Looker Studio report for each team, and share each report with the individuals within each team.

B. Create one Looker Studio report with multiple pages, and add each team's data as a separate data source to the report.

C. Create a Looker (Google Cloud core) instance, and create a separate dashboard for each team.

D. Create a Looker (Google Cloud core) instance, and configure different Looker groups for each team.

Answer: D

Explanation:

Comprehensive and Detailed In-Depth

For an enterprise BI platform with data access control across teams, Google recommends Looker (Google Cloud core) over Looker Studio for its robust access management. The "first" step focuses on setting up the foundation.

Option A: Looker Studio reports are lightweight but lack granular access control beyond sharing. Creating separate reports per team is inefficient and unscalable.

Option B: One Looker Studio report with multiple pages and data sources doesn't enforce team-level access control natively—users could access all pages/data.

Option C: Creating a Looker instance with separate dashboards per team is a step forward but skips the foundational access control setup (groups), reducing scalability.

Option D: Setting up a Looker instance and configuring groups aligns with Google's recommendation for enterprise BI. Groups allow role-based access control (RBAC) at the model, Explore, or dashboard level, ensuring teams see only their data. This is

the scalable, foundational step per Looker's "Access Control" documentation.

Reference: Looker Documentation - "Managing Users and Groups" (https://cloud.google.com/looker/docs/admin-users-groups).

Question: 74

Your company is adopting BigQuery as their data warehouse platform. Your team has experienced Python developers. You need to recommend a fully-managed tool to build batch ETL processes that extract data from various source systems, transform the data using a variety of Google Cloud services, and load the transformed data into BigQuery. You want this tool to leverage your team's Python skills. What should you do?

A. Use Dataform with assertions.
B. Deploy Cloud Data Fusion and included plugins.
C. Use Cloud Composer with pre-built operators.
D. Use Dataflow and pre-built templates.

Answer: C

Explanation:

Comprehensive and Detailed In-Depth

The tool must be fully managed, support batch ETL, integrate with multiple Google Cloud services, and leverage Python skills.

Option A: Dataform is SQL-focused for ELT within BigQuery, not Python-centric, and lacks broad service integration for extraction.

Option B: Cloud Data Fusion is a visual ETL tool, not Python-focused, and requires more UI-based configuration than coding.

Option C: Cloud Composer (managed Apache Airflow) is fully managed, supports batch ETL via DAGs, integrates with various Google Cloud services (e.g., BigQuery, GCS) through operators, and allows custom Python code in tasks. It's ideal for Python developers per the "Cloud Composer" documentation.

Option D: Dataflow excels at streaming and batch processing but focuses on Apache Beam (Python SDK available), not broad service orchestration. Pre-built templates limit customization.

Reference: Google Cloud Documentation - "Cloud Composer Overview" (https://cloud.google.com/composer/docs).

Question: 75

You need to create a data pipeline for a new application. Your application will stream data that needs to be enriched and cleaned. Eventually, the data will be used to train machine learning models. You need to determine the appropriate data manipulation methodology and which Google Cloud services to use in this pipeline. What should you choose?

A. ETL; Dataflow -> BigQuery
B. ETL; Cloud Data Fusion -> Cloud Storage
C. ELT; Cloud Storage -> Bigtable
D. ELT; Cloud SQL -> Analytics Hub

Answer: A

Explanation:
Comprehensive and Detailed In-Depth
Streaming data requiring enrichment and cleaning before ML training suggests an ETL (Extract, Transform, Load) approach, with a focus on real-time processing and a data warehouse for ML.

Option A: ETL with Dataflow (streaming transformations) and BigQuery (storage/ML training) is Google's recommended pattern for streaming pipelines. Dataflow handles enrichment/cleaning, and BigQuery supports ML model training (BigQuery ML).

Option B: ETL with Cloud Data Fusion to Cloud Storage is batch-oriented and lacks streaming focus. Cloud Storage isn't ideal for ML training directly.

Option C: ELT (load then transform) with Cloud Storage to Bigtable is misaligned—Bigtable is for NoSQL, not ML training or post-load transformation.

Option D: ELT with Cloud SQL to Analytics Hub is for relational data and data sharing, not streaming or ML.
Reference: Google Cloud Documentation - "Dataflow: ETL Patterns" (https://cloud.google.com/dataflow/docs/guides), "BigQuery ML" (https://cloud.google.com/bigquery-ml).

Question: 76

You need to transfer approximately 300 TB of data from your company's on-premises data center to Cloud Storage. You have 100 Mbps internet bandwidth, and the transfer needs to be completed as quickly as possible. What should you do?

A. Use Cloud Client Libraries to transfer the data over the internet.

B. Use the gcloud storage command to transfer the data over the internet.

C. Compress the data, upload it to multiple cloud storage providers, and then transfer the data to Cloud Storage.

D. Request a Transfer Appliance, copy the data to the appliance, and ship it back to Google.

Answer: D

Explanation:

Comprehensive and Detailed In-Depth

Transferring 300 TB over a 100 Mbps connection would take an impractical amount of time (over 300 days at theoretical maximum speed, ignoring real-world constraints like latency). Google Cloud provides the Transfer Appliance for large-scale, time-sensitive transfers.

Option A: Cloud Client Libraries over the internet would be slow and unreliable for 300 TB due to bandwidth limitations.

Option B: The gcloud storage command is similarly constrained by internet speed and not designed for such large transfers.

Option C: Compressing and splitting across multiple providers adds complexity and isn't a Google- supported method for Cloud Storage ingestion.

Option D: The Transfer Appliance is a physical device shipped to your location, capable of handling terabytes of data quickly (e.g., 300 TB in days via high-speed local copy), then shipped back to Google for upload to Cloud Storage.

Extract from Google Documentation: From "Transferring Data to Google Cloud Storage" (https://cloud.google.com/storage/docs/transferring-data): "For transferring large amounts of data (hundreds of terabytes or more), consider using the Transfer Appliance, a high-capacity storage server that you load with data and ship to a Google data center. This is ideal when transferring over the internet would take too long."

Reference: Google Cloud Documentation - "Transfer Appliance Overview" (https://cloud.google.com/transfer-appliance).

Question: 77

You are working with a small dataset in Cloud Storage that needs to be transformed and loaded into BigQuery for analysis. The transformation involves simple filtering and aggregation operations. You want to use the most efficient and cost-effective data manipulation approach. What should you do?

A. Use Dataproc to create an Apache Hadoop cluster, perform the ETL process using Apache Spark, and load the results into BigQuery.

B. Use BigQuery's SQL capabilities to load the data from Cloud Storage, transform it, and store the results in a new

BigQuery table.

C. Create a Cloud Data Fusion instance and visually design an ETL pipeline that reads data from Cloud Storage, transforms it using built-in transformations, and loads the results into BigQuery.

D. Use Dataflow to perform the ETL process that reads the data from Cloud Storage, transforms it using Apache Beam, and writes the results to BigQuery.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth

For a small dataset with simple transformations (filtering, aggregation), Google recommends leveraging BigQuery's native SQL capabilities to minimize cost and complexity.

Option A: Dataproc with Spark is overkill for a small dataset, incurring cluster management costs and setup time.

Option B: BigQuery can load data directly from Cloud Storage (e.g., CSV, JSON) and perform transformations using SQL in a serverless manner, avoiding additional service costs. This is the most efficient and cost-effective approach.

Option C: Cloud Data Fusion is suited for complex ETL but adds overhead (instance setup, UI design) unnecessary for simple tasks.

Option D: Dataflow is powerful for large-scale or streaming ETL but introduces unnecessary complexity and cost for a small, simple batch job.

Extract from Google Documentation: From "Loading Data into BigQuery from Cloud Storage" (https://cloud.google.com/bigquery/docs/loading-data-cloud-storage): "You can load data directly from Cloud Storage into BigQuery and use SQL queries to transform it without needing additional processing tools, making it cost-effective for simple transformations."

Reference: Google Cloud Documentation - "BigQuery Data Loading" (https://cloud.google.com/bigquery/docs/loading-data).

Question: 78

You want to build a model to predict the likelihood of a customer clicking on an online advertisement. You have historical data in BigQuery that includes features such as user demographics, ad placement, and previous click behavior. After training the model, you want to generate predictions on new dat

a. Which model type should you use in BigQuery ML?

A. Linear regression

B. Matrix factorization

C. Logistic regression

D. K-means clustering

Answer: C

Explanation:

Comprehensive and Detailed In-Depth

Predicting the likelihood of a click (binary outcome: click or no-click) requires a classification model. BigQuery ML supports this use case with logistic regression.

Option A: Linear regression predicts continuous values, not probabilities for binary outcomes.

Option B: Matrix factorization is for recommendation systems, not binary prediction.

Option C: Logistic regression predicts probabilities for binary classification (e.g., click likelihood), ideal for this scenario and supported in BigQuery ML.

Option D: K-means clustering is for unsupervised grouping, not predictive modeling.

Extract from Google Documentation: From "BigQuery ML: Logistic Regression" (https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax- create#logistic_reg): "Logistic regression models are used to predict the probability of a binary outcome, such as whether an event will occur, making them suitable for classification tasks like click prediction."

Reference: Google Cloud Documentation - "BigQuery ML Model Types" (https://cloud.google.com/bigquery-ml/docs/introduction).

Question: 79

Your data science team needs to collaboratively analyze a 25 TB BigQuery dataset to support the development of a machine learning model. You want to use Colab Enterprise notebooks while ensuring efficient data access and minimizing cost. What should you do?

A. Export the BigQuery dataset to Google Drive. Load the dataset into the Colab Enterprise notebook using Pandas.

B. Use BigQuery magic commands within a Colab Enterprise notebook to query and analyze the data.

C. Create a Dataproc cluster connected to a Colab Enterprise notebook, and use Spark to process the data in BigQuery.

D. Copy the BigQuery dataset to the local storage of the Colab Enterprise runtime, and analyze the data using Pandas.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth

For a 25 TB dataset, efficiency and cost require minimizing data movement and leveraging BigQuery's scalability within Colab Enterprise.

Option A: Exporting 25 TB to Google Drive and loading via Pandas is impractical (size limits, transfer costs) and slow.

Option B: BigQuery magic commands (%%bigquery) in Colab Enterprise allow direct querying of BigQuery data, keeping processing in the cloud, reducing costs, and enabling collaboration.

Option C: Dataproc with Spark adds cluster costs and complexity, unnecessary when BigQuery can handle the workload.

Option D: Copying 25 TB to local storage is infeasible due to size and cost.

Extract from Google Documentation: From "Using BigQuery with Colab Enterprise" (https://cloud.google.com/colab/docs/bigquery): "You can use BigQuery magic commands (%%bigquery) in Colab Enterprise to execute SQL queries directly against BigQuery datasets, providing efficient access to large-scale data without moving it."

Reference: Google Cloud Documentation - "Colab Enterprise with BigQuery" (https://cloud.google.com/colab/docs).

Question: 80

You manage an ecommerce website that has a diverse range of products. You need to forecast future product demand accurately to ensure that your company has sufficient inventory to meet customer needs and avoid stockouts. Your company's historical sales data is stored in a BigQuery table. You need to create a scalable solution that takes into account the seasonality and historical data to predict product demand. What should you do?

A. Use the historical sales data to train and create a BigQuery ML time series model. Use the ML.FORECAST function call to output the predictions into a new BigQuery table.

B. Use Colab Enterprise to create a Jupyter notebook. Use the historical sales data to train a custom prediction model in Python.

C. Use the historical sales data to train and create a BigQuery ML linear regression model. Use the ML.PREDICT function call to output the predictions into a new BigQuery table.

D. Use the historical sales data to train and create a BigQuery ML logistic regression model. Use the ML.PREDICT function call to output the predictions into a new BigQuery table.

Answer: A

Explanation:

Comprehensive and Detailed In-Depth

Forecasting product demand with seasonality requires a time series model, and BigQuery ML offers a scalable, serverless solution. Let's analyze:

Option A: BigQuery ML's time series models (e.g., ARIMA_PLUS) are designed for forecasting with seasonality and trends. The ML.FORECAST function generates predictions based on historical data, storing them in a table. This is scalable (no infrastructure) and integrates natively with BigQuery, ideal for ecommerce demand prediction.

Option B: Colab Enterprise with a custom Python model (e.g., Prophet) is flexible but requires coding, maintenance, and potentially exporting data, reducing scalability compared to BigQuery ML's in-place processing.

Option C: Linear regression predicts continuous values but doesn't handle seasonality or time series patterns effectively, making it unsuitable for demand forecasting.

Option D: Logistic regression is for binary classification (e.g., yes/no), not time series forecasting of demand quantities.
Why A is Best: ARIMA_PLUS in BigQuery ML automatically models seasonality and trends, requiring only SQL knowledge. It's serverless, scales with BigQuery's capacity, and keeps data in one place, minimizing complexity and cost. For example, CREATE MODEL ...

OPTIONS(model_type='ARIMA_PLUS') followed by ML.FORECAST delivers accurate, scalable forecasts.
Extract from Google Documentation: From "BigQuery ML Time Series Forecasting" (https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create- time-series): "The ARIMA_PLUS model type in BigQuery ML is designed for time series forecasting, accounting for seasonality and trends, making it ideal for predicting future values like product demand based on historical data."
Reference: Google Cloud Documentation - "BigQuery ML Time Series" (https://cloud.google.com/bigquery-ml/docs/time-series).

Question: 81

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.

Answer: C

Explanation:
Comprehensive and Detailed In-Depth

A one-time load with minimal effort points to a simple, out-of-the-box tool. The files are local, so the solution must bridge on-premises to BigQuery easily.

Option A: A Python script with the Storage Write API requires coding, setup (authentication, libraries), and debugging—more effort than necessary for a one-time task.

Option B: Dataproc with Spark involves cluster creation, file transfer to Cloud Storage, and job scripting—far too complex for a simple load.

Option C: The bq load command (part of the Google Cloud SDK) is a CLI tool that uploads local files (e.g., CSV, JSON) directly to BigQuery with one command (e.g., bq load --source_format=CSV dataset.table file.csv). It's pre-built, requires no coding, and leverages existing SDK installation, minimizing effort.

Option D: Dataflow with Beam requires pipeline development and isn't designed for local execution to BigQuery without Cloud Storage staging, adding complexity.

Why C is Best: For a demo, bq load is the fastest, simplest way to get data into BigQuery. Install the SDK, run a command, and you're done—no infrastructure or coding needed. It's Google's go-to for ad-hoc loads.

Extract from Google Documentation: From "Loading Data into BigQuery with the bq Command-Line Tool" (https://cloud.google.com/bigquery/docs/bq-command-line-tool#loading_data): "The bq load command allows you to upload data files from your local machine to BigQuery with a single command, providing a quick and easy way to perform one-time data loads." Reference: Google Cloud Documentation - "bq Command-Line Tool" (https://cloud.google.com/bigquery/docs/bq-command-line-tool).

Question: 82

You are storing data in Cloud Storage for a machine learning project. The data is frequently accessed during the model training phase, minimally accessed after 30 days, and unlikely to be accessed after 90 days. You need to choose the appropriate storage class for the different stages of the project to minimize cost. What should you do?

A. Store the data in Nearline storage during the model training phase. Transition the data to Coldline storage 30 days after model deployment, and to Archive storage 90 days after model deployment.

B. Store the data in Standard storage during the model training phase. Transition the data to Nearline storage 30 days after model deployment, and to Coldline storage 90 days after model deployment.

C. Store the data in Nearline storage during the model training phase. Transition the data to Archive storage 30 days after model deployment, and to Coldline storage 90 days after model deployment.

D. Store the data in Standard storage during the model training phase. Transition the data to Durable Reduced Availability (DRA) storage 30 days after model deployment, and to Coldline storage 90 days after model deployment.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth

Cost minimization requires matching storage classes to access patterns using lifecycle rules. Let's assess:

Option A: Nearline during training (frequent access) incurs high retrieval costs and latency, unsuitable for ML workloads. Coldline after 30 days and Archive after 90 days are reasonable but misaligned initially.

Option B: Standard storage (no retrieval fees, low latency) is ideal for frequent access during training. Transitioning to Nearline (30-day minimum, low access) after 30 days and Coldline (90-day minimum, rare access) after 90 days matches the pattern and minimizes costs effectively.

Option C: Nearline during training is costly for frequent access, and Archive to Coldline is illogical (Archive is cheaper than Coldline).

Option D: DRA storage doesn't exist in Google Cloud (legacy AWS term); the progression should be Standard -> Nearline -> Coldline.

Why B is Best: Standard ensures training efficiency, while Nearline and Coldline reduce costs as access drops, all manageable via lifecycle rules (e.g., SetStorageClass actions). This is Google's recommended tiering strategy.

Extract from Google Documentation: From "Cloud Storage Classes" (https://cloud.google.com/storage/docs/storage-classes): "Use Standard storage for frequently accessed data, such as during active ML training. Transition to Nearline after 30 days for infrequent

access, and Coldline after 90 days for rare access, optimizing costs with lifecycle management." Reference: Google Cloud Documentation - "Storage Classes" (https://cloud.google.com/storage/docs/storage-classes).

Question: 83

You need to design a data pipeline to process large volumes of raw server log data stored in Cloud Storage. The data needs to be cleaned, transformed, and aggregated before being loaded into BigQuery for analysis. The transformation involves complex data manipulation using Spark scripts that your team developed. You need to implement a solution that leverages your team's existing skillset, processes data at scale, and minimizes cost. What should you do?

A. Use Dataflow with a custom template for the transformation logic.
B. Use Cloud Data Fusion to visually design and manage the pipeline.
C. Use Dataform to define the transformations in SQLX.
D. Use Dataproc to run the transformations on a cluster.

Answer: D

Explanation:

Comprehensive and Detailed In-Depth

The pipeline must handle large-scale log processing with existing Spark scripts, prioritizing skillset reuse, scalability, and cost. Let's break it down:

Option A: Dataflow uses Apache Beam, not Spark, requiring script rewrites (losing skillset leverage). Custom templates scale well but increase development cost and effort.

Option B: Cloud Data Fusion is a visual ETL tool, not Spark-based. It doesn't reuse existing scripts, requiring redesign, and is less cost-efficient for complex, code-driven transformations.

Option C: Dataform uses SQLX for BigQuery ELT, not Spark. It's unsuitable for pre-load transformations of raw logs and doesn't leverage Spark skills.

Option D: Dataproc runs Spark natively, allowing direct use of your team's scripts. It scales for large datasets (ephemeral clusters minimize cost) and integrates with Cloud Storage and BigQuery seamlessly.

Why D is Best: Dataproc is Google's managed Spark platform, ideal for large-scale, script-based processing. For example, a script cleaning logs (e.g., parsing, deduplicating) runs as-is on a cluster,

writing results to BigQuery via the Spark BigQuery Connector. Cost is minimized with preemptible

VMs or auto-scaling clusters. It's the most practical fit for your team's expertise and requirements.

Extract from Google Documentation: From "Dataproc Overview"

(https://cloud.google.com/dataproc/docs): "Dataproc is a managed Spark and Hadoop service that lets you run existing Spark scripts to process large-scale data from Cloud Storage, with cost-effective scaling and integration to BigQuery for analysis."

Reference: Google Cloud Documentation - "Dataproc" (https://cloud.google.com/dataproc).

Question: 84

You are designing a BigQuery data warehouse with a team of experienced SQL developers. You need to recommend a cost-effective, fully-managed, serverless solution to build ELT processes with SQL pipelines. Your solution must include source code control, environment parameterization, and data quality checks. What should you do?

A. Use Cloud Data Fusion to visually design and manage the pipelines.
B. Use Dataform to build, orchestrate, and monitor the pipelines.
C. Use Dataproc to run MapReduce jobs for distributed data processing.
D. Use Cloud Composer to orchestrate and run data workflows.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth

The solution must support SQL-based ELT, be serverless and cost-effective, and include advanced features like version control and quality checks. Let's dive in:

Option A: Cloud Data Fusion is a visual ETL tool, not SQL-centric (uses plugins), and isn't fully serverless (requires instance management). It lacks native source code control and parameterization.

Option B: Dataform is a serverless, SQL-based ELT platform for BigQuery. It uses SQLX scripts, integrates with Git for version control, supports environment variables (parameterization), and offers assertions for data quality—all meeting the requirements cost-effectively.

Option C: Dataproc is for Spark/MapReduce, not SQL ELT, and requires cluster management, contradicting serverless and cost goals.

Option D: Cloud Composer orchestrates workflows (Python DAGs), not SQL pipelines natively. It's managed but not optimized for ELT within BigQuery alone.

Why B is Best: Dataform leverages your team's SQL skills, runs in BigQuery (no extra infrastructure), and provides Git integration (e.g., GitHub), parameterization (e.g., DECLARE env STRING DEFAULT 'prod';), and quality checks (e.g., assert 'no_nulls' AS SELECT COUNT(*) FROM table WHERE col IS NULL). It's the perfect fit.

Extract from Google Documentation: From "Dataform Overview" (https://cloud.google.com/dataform/docs): "Dataform is a fully managed, serverless solution for building SQL-based ELT pipelines in BigQuery, with built-in Git version control, environment parameterization, and data quality assertions for robust data warehouse management." Reference: Google Cloud Documentation - "Dataform" (https://cloud.google.com/dataform).

Question: 85

You have a Cloud SQL for PostgreSQL database that stores sensitive historical financial dat

a. You need to ensure that the data is uncorrupted and recoverable in the event that the primary region is destroyed. The data is valuable, so you need to prioritize recovery point objective (RPO) over recovery time objective (RTO). You want to recommend a solution that minimizes latency for primary read and write operations. What should you do?

A. Configure the Cloud SQL for PostgreSQL instance for multi-region backup locations.
B. Configure the Cloud SQL for PostgreSQL instance for regional availability (HA). Back up the Cloud SQL for PostgreSQL database hourly to a Cloud Storage bucket in a different region.
C. Configure the Cloud SQL for PostgreSQL instance for regional availability (HA) with synchronous replication to a secondary instance in a different zone.
D. Configure the Cloud SQL for PostgreSQL instance for regional availability (HA) with asynchronous replication to a secondary instance in a different region.

Answer: A

Explanation:

Comprehensive and Detailed In-Depth

The priorities are data integrity, recoverability after a regional disaster, low RPO (minimal data loss), and low latency for primary operations. Let's analyze:

Option A: Multi-region backups store point-in-time snapshots in a separate region. With automated backups and transaction logs, RPO can be near-zero (e.g., minutes), and recovery is possible postdisaster. Primary operations remain in one zone, minimizing latency.

Option B: Regional HA (failover to another zone) with hourly cross-region backups protects against

zone failures, but hourly backups yield an RPO of up to 1 hour—too high for valuable data. Manual backup management adds overhead.

Option C: Synchronous replication to another zone ensures zero RPO within a region but doesn't protect against regional loss. Latency increases slightly due to sync writes across zones.

Option D: Asynchronous replication to another region reduces RPO (e.g., seconds) but increases primary write latency (waiting for async confirmation) and doesn't guarantee zero data loss in a crash.

Why A is Best: Multi-region backups balance low RPO (continuous logs + frequent backups) with no latency impact on primary operations (single-zone instance). For example, enabling backups to us- east1 from a us-west1 instance ensures recoverability without affecting performance, prioritizing RPO as requested.

Extract from Google Documentation: From "Cloud SQL Backup and Recovery" (https://cloud.google.com/sql/docs/postgres/backup-recovery): "Configure multi-region backups to store data in a different region, ensuring recoverability after a regional failure with minimal RPO using automated backups and transaction logs, while keeping primary operations low-latency." Reference: Google Cloud Documentation - "Cloud SQL Backups" (https://cloud.google.com/sql/docs/postgres/backup-recovery).

Question: 86

You work for a retail company that collects customer data from various sources:

Online transactions: Stored in a MySQL database

Customer feedback: Stored as text files on a company server

Social media activity: Streamed in real-time from social media platforms

You need to design a data pipeline to extract and load the data into the appropriate Google Cloud storage system(s) for further analysis and ML model training. What should you do?

A. Copy the online transactions data into Cloud SQL for MySQL. Import the customer feedback into BigQuery. Stream the social media activity into Cloud Storage.

B. Extract and load the online transactions data into BigQuery. Load the customer feedback data into Cloud Storage. Stream the social media activity by using Pub/Sub and Dataflow, and store the data in BigQuery.

C. Extract and load the online transactions data, customer feedback data, and social media activity into Cloud Storage.

D. Extract and load the online transactions data into Bigtable. Import the customer feedback data into Cloud Storage. Store the social media activity in Cloud SQL for MySQL.

Answer: B

Explanation:

Comprehensive and Detailed In-Depth

The pipeline must extract diverse data types and load them into systems optimized for analysis and ML. Let's assess:

Option A: Cloud SQL for transactions keeps data relational but isn't ideal for analysis/ML (less scalable than BigQuery). BigQuery for feedback is fine but skips staging. Cloud Storage for streaming social media loses real-time context and requires extra steps for analysis.

Option B: BigQuery for transactions (via export from MySQL) supports analysis/ML with SQL. Cloud Storage stages feedback text files for preprocessing, then BigQuery ingestion. Pub/Sub and Dataflow stream social media into BigQuery, enabling real-time analysis—optimal for all sources.

Option C: Cloud Storage for all data is a staging step, not a final solution for analysis/ML, requiring additional pipelines.

Option D: Bigtable for transactions is for NoSQL workloads, not analytics. Cloud Storage for feedback is staging-only. Cloud SQL for streaming social media is impractical (not real-time optimized).

Why B is Best: BigQuery is Google's analytics/ML hub (e.g., BigQuery ML). Staging feedback in Cloud Storage allows preprocessing, and streaming via Pub/Sub/Dataflow ensures real-time data in BigQuery. For example, MySQL data exports to BigQuery via bq load, feedback uploads to GCS, and Dataflow processes social media into BigQuery tables.

Extract from Google Documentation: From "Data Analytics on Google Cloud" (https://cloud.google.com/architecture/data-analytics): "Load structured data (e.g., MySQL) and unstructured data (e.g., text) into BigQuery for analysis and ML, using Cloud Storage for staging and Pub/Sub with Dataflow for streaming real-time data like social media activity." Reference: Google Cloud Documentation - "BigQuery Data Ingestion" (https://cloud.google.com/bigquery/docs/loading-data).

Question: 87

Your organization is building a new application on Google Cloud. Several data files will need to be stored in Cloud Storage. Your organization has approved only two specific cloud regions where these data files can reside. You need to determine a Cloud Storage bucket strategy that includes automated high availability. What should you do?

A. Create a dual-region bucket, and upload the files to this bucket.

B. Create a single-region bucket in each of the two regions, and use the gcloud storage command to replicate the data across the buckets in both regions.

C. Create a multi-region bucket, and upload the files to this bucket.

D. Create a single-region bucket in each of the two regions, and use Storage Transfer Service to replicate the data

across the buckets in both regions.

Answer: A

Explanation:

Comprehensive and Detailed In-Depth

The strategy requires storage in two specific regions with automated high availability (HA). Cloud Storage location options dictate the solution:

Option A: A dual-region bucket (e.g., us-west1 and us-east1) replicates data synchronously across two user-specified regions, ensuring HA without manual intervention. It's fully automated and meets the requirement.

Option B: Two single-region buckets with gcloud storage replication is manual, not automated, and lacks real-time HA (requires scripting and monitoring).

Option C: Multi-region buckets (e.g., us) span multiple regions within a geography but don't let you specify exactly two regions, potentially violating the restriction.

Option D: Two single-region buckets with Storage Transfer Service automate replication but aren't synchronous (batch-based), reducing HA compared to dual-region's real-time sync.

Why A is Best: Dual-region buckets provide geo-redundancy across two exact regions (e.g., nam4 for us-central1/us-east1), ensuring data is always available with no manual setup. For example, gsutil mb -l nam4 gs://my-bucket creates this setup, aligning with Google's HA recommendations.

Extract from Google Documentation: From "Cloud Storage Bucket Locations" (https://cloud.google.com/storage/docs/locations): "Dual-region buckets provide high availability by synchronously replicating data across two specific regions you choose, ensuring automated redundancy and accessibility within your approved locations."

Reference: Google Cloud Documentation - "Cloud Storage Dual-Region" (https://cloud.google.com/storage/docs/locations#dual-region).

Question: 88

Your organization is conducting analysis on regional sales metrics. Data from each regional sales team is stored as separate tables in BigQuery and updated monthly. You need to create a solution

that identifies the top three regions with the highest monthly sales for the next three months. You want the solution to automatically provide up-to-date results. What should you do?

A. Create a BigQuery table that performs a union across all of the regional sales tables. Use the row_number() window

function to query the new table.

B. Create a BigQuery table that performs a cross join across all of the regional sales tables. Use the rank() window function to query the new table.

C. Create a BigQuery materialized view that performs a union across all of the regional sales tables. Use the rank() window function to query the new materialized view.

D. Create a BigQuery materialized view that performs a cross join across all of the regional sales tables. Use the row_number() window function to query the new materialized view.

Answer: C

Explanation:

Comprehensive and Detailed in Depth

Why C is correct:Materialized views in BigQuery are precomputed views that periodically cache the results of a query. This ensures up-to-date results automatically.

A UNION is the correct operation to combine the data from multiple regional sales tables.

RANK() function is correct to rank the sales regions. ROW_NUMBER() would create a unique number for each row, even if sales amount is the same, this is not the desired function.

Why other options are incorrect:A and B: Standard tables do not provide automatic updates.

E. A CROSS JOIN would produce a Cartesian product, which is not appropriate for combining regional sales data.

Cross join is used when you want every combination of rows from tables, not a aggregation of data.

Reference:

BigQuery Materialized Views: https://cloud.google.com/bigquery/docs/materialized-views

BigQuery Window Functions: https://cloud.google.com/bigquery/docs/reference/standard- sql/window-functions

BigQuery UNION: https://cloud.google.com/bigquery/docs/reference/standard-sql/query- syntax#union_all

BigQuery CROSS JOIN: https://cloud.google.com/bigquery/docs/reference/standard-sql/query- syntax#cross_join

Question: 89

Your company stores historical data in Cloud Storage. You need to ensure that all data is saved in a bucket for at least three years. What should you do?

A. Enable Object Versioning.

B. Change the bucket storage class to Archive.

C. Set a bucket retention policy.

D. Set temporary object holds.

Answer: C

Explanation:

Comprehensive and Detailed in Depth

Why C is correct:Bucket retention policies are specifically designed to enforce a minimum retention period for objects within a Cloud Storage bucket. This ensures that data cannot be deleted or overwritten before the specified period.

Why other options are incorrect:A: Object versioning allows you to keep multiple versions of an object, but it doesn't guarantee a minimum retention period.

8: Changing the storage class to Archive is for cost optimization, not data retention enforcement.

D: Object holds are for legal holds, not general retention.

Reference:

Cloud Storage Retention Policies: https://cloud.google.com/storage/docs/bucket-lock

Cloud Storage Object Versioning: https://cloud.google.com/storage/docs/object-versioning

Cloud Storage Storage Classes: https://cloud.google.com/storage/docs/storage-classes

Cloud Storage Object Holds: https://cloud.google.com/storage/docs/legal-holds

Question: 90

Your organization consists of two hundred employees on five different teams. The leadership team is concerned that any employee can move or delete all Looker dashboards saved in the Shared folder. You need to create an easy-to-manage solution that allows the five different teams in your organization to view content in the Shared folder, but only be able to move or delete their teamspecific dashboard. What should you do?

A. 1. Create Looker groups representing each of the five different teams, and add users to their corresponding group. 2. Create five subfolders inside the Shared folder. Grant each group the View access level to their corresponding subfolder.

B. 1. Move all team-specific content into the dashboard owner s personal folder. 2. Change the access level of the Shared folder to View for the All Users group. 3. Instruct each user to create content for their team in the user's personal folder.

C. 1. Change the access level of the Shared folder to View for the All Users group. 2. Create Looker groups representing each of the five different teams, and add users to their corresponding group. 3. Create five subfolders inside the Shared folder. Grant each group the Manage Access, Edit access level to their corresponding subfolder.

D. 1. Change the access level of the Shared folder to View for the All Users group. 2. Create five subfolders inside the Shared folder. Grant each team member the Manage Access, Edit access level to their corresponding subfolder.

Answer: C

Explanation:

Comprehensive and Detailed in Depth

Why C is correct:Setting the Shared folder to "View" ensures everyone can see the content.

Creating Looker groups simplifies access management.

Subfolders allow granular permissions for each team.

Granting "Manage Access, Edit" allows teams to modify only their own content.

Why other options are incorrect:A: Grants View access only, so teams can't edit.

B: Moving content to personal folders defeats the purpose of sharing.

D: Grants edit access to all members of the team, not the team as a whole, which is not ideal.

Reference:

Looker Access Control: https://cloud.google.com/looker/docs/access-control

Looker Groups: https://cloud.google.com/looker/docs/groups

Question: 91

Your retail company wants to analyze customer reviews to understand sentiment and identify areas for improvement. Your company has a large dataset of customer feedback text stored in BigQuery that includes diverse language patterns, emojis, and slang. You want to build a solution to classify customer sentiment from the feedback text. What should you do?

A. Preprocess the text data in BigQuery using SQL functions. Export the processed data to AutoML Natural Language for model training and deployment.

B. Export the raw data from BigQuery. Use AutoML Natural Language to train a custom sentiment analysis model.

C. Use Dataproc to create a Spark cluster, perform text preprocessing using Spark NLP, and build a sentiment analysis model with Spark MLlib.

D. Develop a custom sentiment analysis model using TensorFlow. Deploy it on a Compute Engine instance.

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:AutoML Natural Language is designed for text classification tasks, including sentiment analysis, and can handle diverse language patterns without extensive preprocessing.

AutoML can train a custom model with minimal coding.

Why other options are incorrect:A: Unnecessary extra preprocessing. AutoML can handle the raw data.

C: Dataproc and Spark are overkill for this task. AutoML is more efficient and easier to use.

D: Developing a custom TensorFlow model requires significant expertise and time, which is not efficient for this scenario.

Reference:

AutoML Natural Language: https://cloud.google.com/natural-language/automl/docs

Question: 92

Your organization's website uses an on-premises MySQL as a backend database. You need to migrate the on-premises MySQL database to Google Cloud while maintaining MySQL features. You want to minimize administrative overhead and downtime. What should you do?

A. Install MySQL on a Compute Engine virtual machine. Export the database files using the mysqldump command. Upload the files to Cloud Storage, and import them into the MySQL instance on Compute Engine.

B. Use Database Migration Service to transfer the data to Cloud SQL for MySQL, and configure the on premises MySQL database as the source.

C. Use a Google-provided Dataflow template to replicate the MySQL database in BigQuery.

D. Export the database tables to CSV files, and upload the files to Cloud Storage. Convert the MySQL schema to a Spanner schema, create a JSON manifest file, and run a Google-provided Dataflow template to load the data into Spanner.

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:Database Migration Service (DMS) is designed for migrating databases to Cloud SQL with minimal downtime and administrative overhead.

Cloud SQL for MySQL is a fully managed MySQL service, which aligns with the requirement to minimize administrative overhead.

Why other options are incorrect:A: Installing MySQL on Compute Engine requires manual management of the database

instance, which increases administrative overhead.

C: BigQuery is not a direct replacement for a relational MySQL database. It's an analytical data warehouse.

D: Spanner is a globally distributed, scalable database, but it requires schema conversion and is not a direct replacement for MySQL, and it is also much more complex than cloud SQL.

Reference:

Database Migration Service: https://cloud.google.com/database-migration

Cloud SQL for MySQL: https://cloud.google.com/sql/docs/mysql

Question: 93

Your company uses Looker as its primary business intelligence platform. You want to use LookML to visualize the profit margin for each of your company's products in your Looker Explores and dashboards. You need to implement a solution quickly and efficiently. What should you do?

A. Apply a filter to only show products with a positive profit margin.

B. Define a new measure that calculates the profit margin by using the existing revenue and cost fields.

C. Create a new dimension that categorizes products based on their profit margin ranges (e.g., high, medium, low).

D. Create a derived table that pre-calculates the profit margin for each product, and include it in the Looker model.

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:Defining a new measure in LookML is the most efficient and direct way to calculate and visualize aggregated metrics like profit margin.

Measures are designed for calculations based on existing fields.

Why other options are incorrect:A: Filtering doesn't calculate or visualize the profit margin itself.

C: Dimensions are for categorizing data, not calculating aggregated metrics.

D: Derived tables are more complex and unnecessary for a simple calculation like profit margin, which can be done using a measure.

Reference:

Looker Measures: https://cloud.google.com/looker/docs/reference/field-params/measure

Looker Dimensions: https://cloud.google.com/looker/docs/reference/field-params/dimension

Looker Derived Tables: https://cloud.google.com/looker/docs/data-modeling/derived-tables

Question: 94

Your company has an on-premises file server with 5 TB of data that needs to be migrated to Google Cloud. The network operations team has mandated that you can only use up to 250 Mbps of the total available bandwidth for the migration. You need to perform an online migration to Cloud Storage. What should you do?

A. Use Storage Transfer Service to configure an agent-based transfer. Set the appropriate bandwidth limit for the agent pool.

B. Use the gcloud storage cp command to copy all files from on-premises to Cloud Storage using the - -daisy-chain option.

C. Request a Transfer Appliance, copy the data to the appliance, and ship it back to Google Cloud.

D. Use the gcloud storage cp command to copy all files from on-premises to Cloud Storage using the - -no-clobber option.

Answer: A

Explanation:

Comprehensive and Detailed in Depth

Why A is correct:Storage Transfer Service with agent-based transfer allows for online migrations and provides the ability to set bandwidth limits.

Agents are installed on-premises and can be configured to respect network constraints.

Why other options are incorrect:B: The --daisy-chain option is not related to bandwidth control.

C: Transfer Appliance is for ofline migrations and is not suitable for online transfers with bandwidth constraints.

D: The --no-clobber option prevents overwriting existing files but does not control bandwidth.

Reference:

Storage Transfer Service: https://cloud.google.com/storage-transfer-service/docs

Storage Transfer Service Agents: https://cloud.google.com/storage-transfer-service/docs/agent- overview

gcloud storage cp: https://cloud.google.com/storage/docs/gsutil/commands/cp

Question: 95

You work for a gaming company that collects real-time player activity dat

a. This data is streamed into Pub/Sub and needs to be processed and loaded into BigQuery for analysis. The processing involves filtering, enriching, and aggregating the data before loading it into partitioned BigQuery tables. You need to design a pipeline that ensures low latency and high throughput while following a Google-recommended approach. What should you do?

A. Use Cloud Composer to orchestrate a workflow that reads the data from Pub/Sub, processes the data using a Python script, and writes it to BigQuery.

B. Use Dataproc to create an Apache Spark streaming job that reads the data from Pub/Sub, processes the data, and writes it to BigQuery.

C. Use Dataflow to create a streaming pipeline that reads the data from Pub/Sub, processes the data, and writes it to BigQuery using the streaming API.

D. Use Cloud Run functions to subscribe to the Pub/Sub topic, process the data, and write it to BigQuery using the streaming API.

Answer: C

Explanation:

Comprehensive and Detailed in Depth

Why C is correct:Dataflow is the recommended service for real-time stream processing on Google Cloud.

It provides scalable and reliable processing with low latency and high throughput.

Dataflow's streaming API is optimized for Pub/Sub integration and BigQuery streaming inserts.

Why other options are incorrect:A: Cloud Composer is for batch orchestration, not real-time streaming.

B: Dataproc and Spark streaming are more complex and not as efficient as Dataflow for this task.

D: Cloud Run functions are for stateless, event-driven applications, not continuous stream processing.

Reference:

Dataflow Streaming: https://cloud.google.com/dataflow/docs/concepts/streaming-pipelines

Pub/Sub to BigQuery with Dataflow: https://cloud.google.com/dataflow/docs/tutorials/pubsub-to- bigquery

Question: 96

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_uype='logisric_reg') AS SELECT * from cusromer_data;

B. CREATE OR REPLACE MODEL churn_prediction_model OPTIONS (rr.odel_type=' logisric_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 churned as label FROM customer_data;

D. CREATE OR REPLACE MODEL churn_prediction_model options(model_type='logistic_reg*) as select ' except(churned) FROM customer data;

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:BigQuery ML requires the target label to be explicitly named label.

EXCEPT(churned) selects all columns except the churned column, which becomes the features.

churned AS label renames the churned column to label, which is required for BigQuery ML.

logistic_reg is the correct model_type option.

Why other options are incorrect:A: Does not rename the target column to label. Also has a typo in the model type.

C: Only selects the target label, not the features.

D: Has a syntax error with the single quote before except.

Reference:

BigQuery ML Logistic Regression: https://cloud.google.com/bigquery-ml/docs/reference/standard- sql/bigqueryml-syntax-create-logistic-regression

BigQuery ML Syntax: https://cloud.google.com/bigquery-ml/docs/reference/standard-

sql/bigqueryml-syntax-create

Question: 97

You work for a healthcare company. You have a daily ETL pipeline that extracts patient data from a legacy system, transforms it, and loads it into BigQuery for analysis. The pipeline currently runs manually using a shell script. You want to automate this process and add monitoring to ensure pipeline observability and troubleshooting insights. You want one centralized solution, using opensource tooling, without rewriting the ETL code. What should you do?

A. Create a direct acyclic graph (DAG) in Cloud Composer to orchestrate a pipeline trigger daily. Monitor the pipeline's execution using the Apache Airflow web interface and Cloud Monitoring.

B. Configure Cloud Dataflow to implement the ETL pipeline, and use Cloud Scheduler to trigger the Dataflow pipeline daily. Monitor the pipelines execution using the Dataflow job monitoring interface and Cloud Monitoring.

C. Use Cloud Scheduler to trigger a Dataproc job to execute the pipeline daily. Monitor the job's progress using the Dataproc job web interface and Cloud Monitoring.

D. Create a Cloud Run function that runs the pipeline daily. Monitor the functions execution using Cloud Monitoring.

Answer: A

Explanation:

Comprehensive and Detailed in Depth

Why A is correct:Cloud Composer is a managed Apache Airflow service, which is a popular opensource workflow orchestration tool.

DAGs in Airflow can be used to automate ETL pipelines.

Airflow's web interface and Cloud Monitoring provide comprehensive monitoring capabilities.

It also allows you to run existing shell scripts.

Why other options are incorrect:B: Dataflow requires rewriting the ETL pipeline using its SDK.

C: Dataproc is for big data processing, not orchestration.

D: Cloud Run functions are for stateless applications, not long-running ETL pipelines.

Reference:

Cloud Composer: https://cloud.google.com/composer/docs

Apache Airflow: https://airflow.apache.org/

Question: 98

You manage data at an ecommerce company. You have a Dataflow pipeline that processes order data from Pub/Sub, enriches the data with product information from Bigtable, and writes the processed data to BigQuery for analysis. The pipeline runs continuously and processes thousands of orders every minute. You need to monitor the pipeline's performance and be alerted if errors occur. What should you do?

A. Use Cloud Monitoring to track key metrics. Create alerting policies in Cloud Monitoring to trigger notifications when metrics exceed thresholds or when errors occur.

B. Use the Dataflow job monitoring interface to visually inspect the pipeline graph, check for errors, and configure notifications when critical errors occur.

C. Use BigQuery to analyze the processed data in Cloud Storage and identify anomalies or inconsistencies. Set up scheduled alerts based when anomalies or inconsistencies occur.

D. Use Cloud Logging to view the pipeline logs and check for errors. Set up alerts based on specific keywords in the logs.

Answer: A

Explanation:

Comprehensive and Detailed in Depth

Why A is correct:Cloud Monitoring is the recommended service for monitoring Google Cloud services, including Dataflow.

It allows you to track key metrics like system lag, element throughput, and error rates.

Alerting policies in Cloud Monitoring can trigger notifications based on metric thresholds.

Why other options are incorrect:B: The Dataflow job monitoring interface is useful for visualization, but Cloud Monitoring provides more comprehensive alerting.

C: BigQuery is for analyzing the processed data, not monitoring the pipeline itself. Also Cloud Storage is not where the data resides during processing.

D: Cloud Logging is useful for viewing logs, but Cloud Monitoring is better for metric-based alerting.

Reference:

Cloud Monitoring for Dataflow: https://cloud.google.com/dataflow/docs/guides/using-monitoring

Cloud Monitoring: https://cloud.google.com/monitoring/docs

Question: 99

Your team uses Google Sheets to track budget data that is updated daily. The team wants to compare budget data against

actual cost data, which is stored in a BigQuery table. You need to create a solution that calculates the difference between each day's budget and actual costs. You want to ensure that your team has access to daily-updated results in Google Sheets. What should you do?

A. Create a BigQuery external table by using the Drive URI of the Google sheet, and join the actual cost table with it. Save the joined table as a CSV file and open the file in Google Sheets.

B. Download the budget data as a CSV file and upload the CSV file to a Cloud Storage bucket. Create a new BigQuery table from Cloud Storage, and join the actual cost table with it. Open the joined BigQuery table by using Connected Sheets.

C. Download the budget data as a CSV file, and upload the CSV file to create a new BigQuery table. Join the actual cost table with the new BigQuery table, and save the results as a CSV file. Open the CSV file in Google Sheets.

D. Create a BigQuery external table by using the Drive URI of the Google sheet, and join the actual cost table with it. Save the joined table, and open it by using Connected Sheets.

Answer: D

Explanation:

Comprehensive and Detailed in Depth

Why D is correct:Creating a BigQuery external table directly from the Google Sheet allows for real time updates.

Joining the external table with the actual cost table in BigQuery performs the calculation.

Connected Sheets allows the team to access and analyze the results directly in Google Sheets, with the data being updated.

Why other options are incorrect:A: Saving as a CSV file loses the live connection and daily updates.

B: Downloading and uploading as a CSV file adds unnecessary steps and loses the live connection.

C: Same issue as B, losing the live connection.

Reference:

BigQuery External Tables: https://cloud.google.com/bigquery/docs/external-tables

Connected Sheets: https://support.google.com/sheets/answer/9054368?hl=en

Question: 100

You have a Cloud SQL for PostgreSQL database that stores sensitive historical financial dat

a. You need to ensure that the data is uncorrupted and recoverable in the event that the primary region is destroyed. The data is valuable, so you need to prioritize recovery point objective (RPO) over recovery time objective (RTO). You want to recommend a solution that minimizes latency for primary read and write operations. What should you do?

A. Configure the Cloud SQL for PostgreSQL instance for regional availability (HA) with asynchronous replication to a secondary instance in a different region.

B. Configure the Cloud SQL for PostgreSQL instance for multi-region backup locations.

C. Configure the Cloud SQL for PostgreSQL instance for regional availability (HA). Back up the Cloud SQL for PostgreSQL database hourly to a Cloud Storage bucket in a different region.

D. Configure the Cloud SQL for PostgreSQL instance for regional availability (HA) with synchronous replication to a secondary instance in a different zone.

Answer: D

Explanation:

Comprehensive and Detailed in Depth

Why D is correct:Synchronous replication ensures that data is written to both the primary and secondary instances at the same time, minimizing data loss (RPO).

Regional availability (HA) within different zones provides redundancy within the same region, minimizing latency.

Why other options are incorrect:A: Asynchronous replication has a potential for data loss.

B: Multiregion backups are for disaster recovery, not minimizing latency.

C: Hourly backups do not provide the lowest possible RPO.

Reference:

Cloud SQL high availability: https://cloud.google.com/sql/docs/postgres/high-availability

Cloud SQL backups: https://cloud.google.com/sql/docs/postgres/backup-restore

Question: 101

You are building a batch data pipeline to process 100 GB of structured data from multiple sources for daily reporting. You need to transform and standardize the data prior to loading the data to ensure that it is stored in a single dataset. You want to use a low-code solution that can be easily built and managed. What should you do?

A. Use Cloud Data Fusion to ingest data and load the data into BigQuery. Use Looker Studio to perform data cleaning and transformation.

B. Use Cloud Data Fusion to ingest the data, perform data cleaning and transformation, and load the data into BigQuery.

C. Use Cloud Data Fusion to ingest the data, perform data cleaning and transformation, and load the data into Cloud SQL for PostgreSQL.

D. Use Cloud Storage to store the data. Use Cloud Run functions to perform data cleaning and transformation, and load the data into BigQuery.

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:Cloud Data Fusion is a fully managed, cloud-native data integration service for building and managing ETL/ELT data pipelines.

It provides a graphical interface for building pipelines without coding, making it a low-code solution.

Cloud data fusion is perfect for the ingestion, transformation and loading of data into BigQuery.

Why other options are incorrect:A: Looker studio is for visualization, not data transformation.

C: Cloud SQL is a relational database, not ideal for large-scale analytical data.

D: Cloud run is for stateless applications, not batch data processing.

Reference:

Cloud Data Fusion: https://cloud.google.com/data-fusion/docs

Question: 102

You are working on a project that requires analyzing daily social media dat

a. You have 100 GB of JSON formatted data stored in Cloud Storage that keeps growing.

You need to transform and load this data into BigQuery for analysis. You want to follow the Google- recommended approach. What should you do?

A. Manually download the data from Cloud Storage. Use a Python script to transform and upload the data into BigQuery.

B. Use Cloud Run functions to transform and load the data into BigQuery.

C. Use Dataflow to transform the data and write the transformed data to BigQuery.

D. Use Cloud Data Fusion to transfer the data into BigQuery raw tables, and use SQL to transform it.

Answer: C

Explanation:

Comprehensive and Detailed in Depth

Why C is correct:Dataflow is a fully managed service for transforming and enriching data in both batch and streaming modes.

Dataflow is googles recomended way to transform large datasets.

It is designed for parallel processing, making it suitable for large datasets.

Why other options are incorrect:A: Manual downloading and scripting is not scalable or efficient.

B: Cloud Run functions are for stateless applications, not large data transformations.

D: While Cloud Data fusion could work, Dataflow is more optimized for large scale data transformation.

Reference:

Dataflow: https://cloud.google.com/dataflow/docs

Query successful

Question: 103

You created a curated dataset of market trends in BigQuery that you want to share with multiple external partners. You want to control the rows and columns that each partner has access to. You want to follow Google-recommended practices. What should you do?

A. Publish the dataset in Analytics Hub. Grant dataset-level access to each partner by using subscriptions.

B. Create a separate Cloud Storage bucket for each partner. Export the dataset to each bucket and assign each partner to their respective bucket. Grant bucket-level access by using 1AM roles.

C. Grant each partner read access to the BigQuery dataset by using 1AM roles.

D. Create a separate project for each partner and copy the dataset into each project. Publish each dataset in Analytics Hub. Grant dataset-level access to each partner by using subscriptions.

Answer: A

Explanation:

Comprehensive and Detailed in Depth

Why A is correct:Analytics Hub allows you to share datasets with external partners while maintaining control over access.

Subscriptions allow granular control.

Why other options are incorrect:B: Cloud storage is for files, not bigquery datasets.

C: IAM roles do not allow for granular row and column level control.

D: Creating a separate project for each partner is complex and not scalable.

Reference:

Analytics Hub: https://cloud.google.com/analytics-hub/docs

Question: 104

Your company wants to implement a data transformation (ETL) pipeline for their BigQuery data warehouse. You need to identify a managed transformation solution that allows users to develop with SQL and JavaScript, has version control, allows for modular code, and has data quality checks. What should you do?

A. Create a Cloud Composer environment, and orchestrate the transformations by using the BigQueryinsertJob operator.

B. Create BigQuery scheduled queries to define the transformations in SQL.

C. Use Dataform to define the transformations in SQLX.

D. Use Dataproc to create an Apache Spark cluster and implement the transformations by using PySpark SQL.

Answer: C

Explanation:

Comprehensive and Detailed in Depth

Why C is correct:Dataform is a managed data transformation service that allows you to define data pipelines using SQL and JavaScript.

It provides version control, modular code development, and data quality checks.

Why other options are incorrect:A: Cloud Composer is an orchestration tool, not a data transformation tool.

B: Scheduled queries are not suitable for complex ETL pipelines.

D: Dataproc requires setting up a Spark cluster and writing code, which is more complex than using Dataform.

Reference:

Dataform: https://cloud.google.com/dataform/docs

Question: 105

Following a recent company acquisition, you inherited an on-premises data infrastructure that needs to move to Google Cloud. The acquired system has 250 Apache Airflow directed acyclic graphs (DAGs) orchestrating data pipelines. You need to

migrate the pipelines to a Google Cloud managed service with minimal effort. What should you do?

A. Convert each DAG to a Cloud Workflow and automate the execution with Cloud Scheduler.

B. Create a new Cloud Composer environment and copy DAGs to the Cloud Composer dags/ folder.

C. Create a Google Kubernetes Engine (GKE) standard cluster and deploy Airflow as a workload. Migrate all DAGs to the new Airflow environment.

D. Create a Cloud Data Fusion instance. For each DAG, create a Cloud Data Fusion pipeline.

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:Cloud Composer is a managed Apache Airflow service that provides a seamless migration path for existing Airflow DAGs.

Simply copying the DAGs to the Cloud Composer folder allows them to run directly on Google Cloud.

Why other options are incorrect:A: Cloud Workflows is a different orchestration tool, not compatible with Airflow DAGs.

C: GKE deployment requires setting up and managing a Kubernetes cluster, which is more complex.

D: Cloud Data Fusion is a data integration tool, not suitable for orchestrating existing pipelines.

Reference:

Cloud Composer: https://cloud.google.com/composer/docs

Question: 106

You have an existing weekly Storage Transfer Service transfer job from Amazon S3 to a Nearline Cloud Storage bucket in Google Cloud. Each week, the job moves a large number of relatively small files. As the number of files to be transferred each week has grown over time, you are at risk of no longer completing the transfer in the allocated time frame. You need to decrease the total transfer time by replacing the process. Your solution should minimize costs where possible. What should you do?

A. Create a transfer job using the Google Cloud CLI, and specify the Standard storage class with the — custom-storage-class flag.

B. Create parallel transfer jobs using include and exclude prefixes.

C. Create a batch Dataflow job that is scheduled weekly to migrate the data from Amazon S3 to Cloud Storage.

D. Create an agent-based transfer job that utilizes multiple transfer agents on Compute Engine instances.

Answer: B

Explanation:

Comprehensive and Detailed in Depth

Why B is correct:Creating parallel transfer jobs by using include and exclude prefixes allows you to split the data into smaller chunks and transfer them in parallel.

This can significantly increase throughput and reduce the overall transfer time.

Why other options are incorrect:A: Changing the storage class to Standard will not improve transfer speed.

C: Dataflow is a complex solution for a simple file transfer task.

D: Agent-based transfer is suitable for large files or network limitations, but not for a large number of small files.