DUMPSARENA

Google Professional Data Engineer Exam

Google Professional-Data-Engineer

Version Demo

Total Demo Questions: 43

Total Premium Questions: 434

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Designing data processing systems	257
Topic 2, Building and operationalizing data processing systems	72
Topic 3, Operationalizing machine learning models	50
Topic 4, Ensuring solution quality	55
Total	434

QUESTION NO: 1

You have a data pipeline that writes data to Cloud Bigtable using well-designed row keys. You want to monitor your pipeline to determine when to increase the size of you Cloud Bigtable cluster. Which two actions can you take to accomplish this? Choose 2 answers.

A. Review Key Visualizer metrics. Increase the size of the Cloud Bigtable cluster when the Read pressure index is above 100.

B. Review Key Visualizer metrics. Increase the size of the Cloud Bigtable cluster when the Write pressure index is above 100.

C. Monitor the latency of write operations. Increase the size of the Cloud Bigtable cluster when there is a sustained increase in write latency.

D. Monitor storage utilization. Increase the size of the Cloud Bigtable cluster when utilization increases above 70% of max capacity.

E. Monitor latency of read operations. Increase the size of the Cloud Bigtable cluster of read operations take longer than 100 ms.

ANSWER: B C

Explanation:

For a pipeline whose primary workload is writing to Cloud Bigtable, use the Key Visualizer write pressure index and write-operation latency to identify when additional cluster nodes are needed. The write pressure index estimates the amount of serving capacity required for writes relative to the cluster's current capacity. A value above 100 indicates that the write workload is demanding more capacity than the cluster can comfortably provide, making it an appropriate signal to add nodes. Google's Key Visualizer documentation describes pressure metrics as a way to identify capacity-related performance pressure and determine whether scaling is appropriate.

Write latency is also a direct workload-level signal. A sustained increase in write latency can indicate that the cluster is approaching or exceeding its available serving capacity, assuming the workload and row-key design have been evaluated. Adding nodes increases the available Bigtable compute capacity and can restore stable write performance. Monitoring sustained behavior, rather than a single brief spike, helps distinguish a capacity trend from temporary variation. See the Cloud Bigtable Key Visualizer overview and Google's Bigtable performance guidance for capacity and performance-monitoring guidance.

QUESTION NO: 2

You want to migrate an on-premises Hadoop system to Cloud Dataproc. Hive is the primary tool in use, and the data format is Optimized Row Columnar (ORC). All ORC files have been successfully copied to a Cloud Storage bucket. You need to replicate some data to the cluster's local Hadoop Distributed File System (HDFS) to maximize performance. What are two ways to start using Hive in Cloud

Dataproc? (Choose two.)

A. Run the gsutil utility to transfer all ORC files from the Cloud Storage bucket to HDFS. Mount the Hive tables locally.

B. Run the gsutil utility to transfer all ORC files from the Cloud Storage bucket to any node of the Dataproc cluster. Mount the Hive tables locally.

C. Run the gsutil utility to transfer all ORC files from the Cloud Storage bucket to the master node of the Dataproc cluster. Then run the Hadoop utility to copy them to HDFS. Mount the Hive tables from
HDFS.

D. Leverage Cloud Storage connector for Hadoop to mount the ORC files as external Hive tables. Replicate external Hive tables to the native ones.

E. Load the ORC files into BigQuery. Leverage BigQuery connector for Hadoop to mount the BigQuery tables as external Hive tables. Replicate external Hive tables to the native ones.

ANSWER: C D

Explanation:

Running the gsutil utility to transfer all ORC files from the Cloud Storage bucket to the master node of the Dataproc cluster, then using the Hadoop utility to copy them to HDFS, is a valid migration path. gsutil can download objects from Cloud Storage to the master node's local filesystem, and the Hadoop filesystem commands can then place those files into the cluster's HDFS namespace. After the data is in HDFS, Hive tables can be created over the corresponding HDFS locations, providing local HDFS access for the data selected for replication.

Leveraging the Cloud Storage connector for Hadoop to expose the ORC files as external Hive tables is also a supported approach. Dataproc includes the Cloud Storage connector, which lets Hadoop-compatible tools access `gs://` paths through the Hadoop filesystem interface. Hive can define external tables whose locations point to Cloud Storage. Data that requires local HDFS performance can subsequently be copied or materialized into native Hive-managed tables stored in HDFS. This supports incremental migration: data can remain durably stored in Cloud Storage while performance-sensitive datasets are replicated to HDFS.

See the Cloud Storage connector documentation and Dataproc HDFS documentation.

QUESTION NO: 3

You are integrating one of your internal IT applications and Google BigQuery, so users can query BigQuery from the application's interface. You do not want individual users to authenticate to BigQuery and you do not want to give them access to the dataset. You need to securely access BigQuery from your IT application. What should you do?

A. Create groups for your users and give those groups access to the dataset

B. Integrate with a single sign-on (SSO) platform, and pass each user's credentials along with the query request

C. Create a service account and grant dataset access to that account. Use the service account's private key to access the dataset

D. Create a dummy user and grant dataset access to that user. Store the username and password for that user in a file on the files system, and use those credentials to access the BigQuery dataset

ANSWER: C

Explanation:

Create a service account and grant dataset access to that account. Use the service account's private key to access the dataset is correct because a service account is a non-human identity designed for an application, workload, or compute resource. The internal application authenticates as that identity, and BigQuery evaluates its permissions rather than requiring every application user to be a Google Cloud principal with direct dataset access. This cleanly separates application access from end-user identity management while allowing the application to submit queries and read only the data authorized by its assigned IAM roles or BigQuery dataset permissions.

The service account should receive the minimum permissions needed for the intended operations, following least privilege. For example, query execution commonly requires permission to create jobs in the billing project as well as appropriate read access to the target data. A service-account key can be used when the application runs outside Google Cloud and no keyless workload identity option is available. The private key must be stored and managed securely, such as in Secret Manager, never embedded in source code or exposed to users. Where the application runs on Google Cloud, attach the service account to its runtime and use Application Default Credentials instead of creating a downloadable key. Google documents service accounts at Service account overview and BigQuery access control at BigQuery access control.

QUESTION NO: 4

Your company has Apache Spark batch jobs that read data from Cloud Storage and write curated data back to Cloud Storage. The jobs run a few times each day, and the company wants to eliminate cluster administration and avoid paying for idle Dataproc clusters. Which two actions should you take? (Choose two.)

A. Submit the workloads as Dataproc Serverless for Spark batch jobs.

B. Store durable input and output data in Cloud Storage.

C. Create a permanent Dataproc cluster and keep it running between jobs.

D. Copy all Cloud Storage data to HDFS before each job and retain the cluster after completion.

E. Run the Spark jobs on manually managed Compute Engine instances.

ANSWER: A B

Explanation:

Submit the workloads as Dataproc Serverless for Spark batch jobs. Dataproc Serverless provisions and manages the compute resources required for each batch workload, so the company does not need to create, operate, or retain an idle Dataproc cluster between job runs. Charges are based on the resources used while the batch workload executes.

Continue using Cloud Storage for durable input and output data. Cloud Storage separates persistent data from temporary processing infrastructure, allowing each serverless batch execution to access the required data without depending on HDFS storage in a long-running cluster. See Dataproc Serverless overview and Cloud Storage connector documentation.

QUESTION NO: 5

Your team has created several BigQuery curated datasets containing anonymized industry benchmark data. You want to make these datasets easily discoverable and accessible for querying by external partner companies within their own Google Cloud projects. You need a **secure and scalable solution**. What should you do?

A. Grant the roles/bigquery.dataViewer IAM role to the partner group email addresses on the datasets.

B. Publish the datasets as listings within BigQuery sharing (Analytics Hub).

C. Create authorized views for each dataset and grant access to each partner.

D. Export the datasets to partner-specific Cloud Storage buckets.

ANSWER: B

Explanation:

Publish the datasets as listings within BigQuery sharing (Analytics Hub). is the appropriate solution because BigQuery sharing is designed to distribute governed data across organizational boundaries without duplicating the underlying dataset. A publisher creates a listing that describes the shared data and specifies the dataset or other shareable resource. External partner organizations can discover the listing in Analytics Hub and subscribe to it. Subscription creates a linked dataset in each subscriber's own Google Cloud project, allowing the partner to query the data through BigQuery while the provider continues to manage the source data centrally.

This model scales efficiently because one listing can serve multiple subscriber projects rather than requiring separate data copies or manually managed sharing arrangements for each partner. It also supports a formal discovery and subscription workflow, which is particularly valuable for curated benchmark datasets intended for a partner ecosystem. The publisher retains governance over the listing and can manage its availability and subscriber access. BigQuery sharing also supports egress controls for listings when stronger restrictions on subscriber data export are required. Google documents Analytics Hub as the BigQuery capability for securely sharing data across organizations: Analytics Hub introduction and Manage Analytics Hub listings.

QUESTION NO: 6

As your organization expands its usage of GCP, many teams have started to create their own projects. Projects are further multiplied to accommodate different stages of deployments and target audiences.

Each project requires unique access control configurations. The central IT team needs to have access to all projects. Furthermore, data from Cloud Storage buckets and BigQuery datasets must be shared for use in other projects in an ad hoc

way. You want to simplify access control management by minimizing the number of policies. Which two steps should you take? (Choose two.)

A. Use Cloud Deployment Manager to automate access provision.

B. Introduce resource hierarchy to leverage access control policy inheritance.

C. Create distinct groups for various teams, and specify groups in Cloud IAM policies.

D. Only use service accounts when sharing data for Cloud Storage buckets and BigQuery datasets.

E. For each Cloud Storage bucket or BigQuery dataset, decide which projects need access. Find all the active members who have access to these projects, and create a Cloud IAM policy to grant access to all these users.

ANSWER: B C

Explanation:

Introduce resource hierarchy to leverage access control policy inheritance. Google Cloud IAM policies are inherited down the resource hierarchy: organization-level permissions flow to folders and projects, and folder-level permissions flow to descendant projects. Placing projects in a meaningful organization and folder structure lets central IT receive the required administrative access through a small number of higher-level policies rather than separately maintaining a policy in every project. Project-specific access can still be added where needed without losing the centrally inherited access.

Create distinct groups for various teams, and specify groups in Cloud IAM policies. Google recommends granting roles to Google Groups rather than managing individual user bindings. Group membership can be updated as teams change, while the IAM policy remains stable. The same group-based approach supports ad hoc cross-project access to Cloud Storage buckets and BigQuery datasets: grant the appropriate resource-level role to the relevant team group, then manage additions and removals through group membership. Together, hierarchy-based inheritance and group principals minimize repeated policy definitions while retaining scalable, centrally governed access management. See Resource hierarchy and IAM access control and IAM security best practices.

QUESTION NO: 7

Your company produces 20,000 files every hour. Each data file is formatted as a comma separated values (CSV) file that is less than 4 KB. All files must be ingested on Google Cloud Platform before they can be processed. Your company site has a 200 ms latency to Google Cloud, and your Internet connection bandwidth is limited as 50 Mbps. You currently deploy a secure FTP (SFTP) server on a virtual machine in Google Compute Engine as the data ingestion point. A local SFTP client runs on a dedicated machine to transmit the CSV files as is. The goal is to make reports with data from the previous day available to the executives by 10:00 a.m. each day. This design is barely able to keep up with the current volume, even though the bandwidth utilization is rather low.

You are told that due to seasonality, your company expects the number of files to double for the next three months. Which two actions should you take? (choose two.)

A. Introduce data compression for each file to increase the rate file of file transfer.

B. Contact your internet service provider (ISP) to increase your maximum bandwidth to at least 100 Mbps.

C. Redesign the data ingestion process to use gsutil tool to send the CSV files to a storage bucket in parallel.

D. Assemble 1,000 files into a tape archive (TAR) file. Transmit the TAR files instead, and disassemble the CSV files in the cloud upon receiving them.

E. Create an S3-compatible storage endpoint in your network, and use Google Cloud Storage Transfer Service to transfer on-premices data to the designated storage bucket.

ANSWER: C D

Explanation:

The appropriate actions are to use gsutil to upload the CSV files to Cloud Storage in parallel and to assemble groups of 1,000 files into TAR archives before transmission. The limiting factor is not the available 50 Mbps bandwidth: the files are

extremely small, while the round-trip latency to Google Cloud is 200 ms. Sending tens of thousands of individual files causes substantial per-object connection, request, authentication, and acknowledgement overhead. That overhead prevents the existing SFTP-based design from efficiently using the available network capacity and will become more severe when the file count doubles.

Using gsutil with the `-m` option performs operations in parallel, allowing multiple uploads to remain in flight and making much better use of a high-latency network path. Uploading to Cloud Storage also provides a scalable managed landing zone for subsequent processing. Creating TAR files reduces the number of transfer operations from 20,000 per hour to roughly 20 archives per hour at the stated grouping size. This dramatically reduces latency-sensitive request overhead while preserving the source CSV contents for extraction and processing after arrival. Google documents parallel gsutil operations at Cloud Storage gsutil cp and describes Cloud Storage as an object-storage service suitable for scalable data ingestion at Cloud Storage introduction.

QUESTION NO: 8

A shipping company has live package-tracking data that is sent to an Apache Kafka stream in real time. This is then loaded into BigQuery. Analysts in your company want to query the tracking data in BigQuery to analyze geospatial trends in the lifecycle of a package. The table was originally created with ingest-date partitioning. Over time, the query processing time has increased. You need to implement a change that would improve query performance in BigQuery. What should you do?

A. Implement clustering in BigQuery on the ingest date column.

B. Implement clustering in BigQuery on the package-tracking ID column.

C. Tier older data onto Google Cloud Storage files and create a BigQuery table using GCS as an external data source.

D. Re-create the table using data partitioning on the package delivery date.

ANSWER: B

Explanation:

Implement clustering in BigQuery on the package-tracking ID column. The existing ingestion-date partitioning remains useful for restricting data to relevant ingestion-time ranges. Clustering then organizes the rows within each partition according to the package-tracking ID, placing records for the same or nearby tracking IDs into fewer storage blocks. Package lifecycle analysis commonly needs to retrieve the sequence of location and status events associated with a particular package or a defined collection of packages. When those queries filter on the tracking ID, BigQuery can use block pruning to read substantially less data, reducing bytes scanned and improving execution time as the table grows.

BigQuery automatically maintains clustered data over time, including as streaming-loaded data is incorporated. Clustering is particularly appropriate as a complement to date partitioning when there is a high-cardinality column that is frequently used to filter or aggregate data. The package-tracking ID is such a column: it identifies the entity whose event history forms the package lifecycle, while the partitioning layout continues to support temporal data management and date-bounded analysis. See the BigQuery clustered tables documentation and partitioned tables documentation.

QUESTION NO: 9

You need to create a data pipeline that copies time-series transaction data so that it can be queried from within BigQuery by your data science team for analysis. Every hour, thousands of transactions are updated with a new status. The size of the initial dataset is 1.5 PB, and it will grow by 3 TB per day. The data is heavily structured, and your data science team will build machine learning models based on this data. You want to maximize performance and usability for your data science team. Which two strategies should you adopt? (Choose two.)

A. Denormalize the data as must as possible.

B. Preserve the structure of the data as much as possible.

C. Use BigQuery UPDATE to further reduce the size of the dataset.

D. Develop a data pipeline where status updates are appended to BigQuery instead of updated.

E. Copy a daily snapshot of transaction data to Cloud Storage and store it as an Avro file. Use BigQuery's support for external data sources to query.

ANSWER: A D

Explanation:

Denormalize the data as must as possible is appropriate because BigQuery is an analytical, columnar data warehouse optimized for scanning large datasets. A denormalized schema reduces the need for repeated joins across extremely large transaction tables, which improves query performance and makes datasets easier for data scientists to use directly in SQL and BigQuery ML workflows. BigQuery documentation specifically recommends denormalizing data where appropriate and using nested and repeated fields to represent relationships, rather than relying on many joins.

Develop a data pipeline where status updates are appended to BigQuery instead of updated is also appropriate for this high-volume, time-series workload. Maintaining an append-only history records each status change as an event, avoids frequent mutations of a petabyte-scale table, and supports reproducible historical analysis and feature engineering. The current status can be derived when needed using SQL, such as by selecting the latest event for each transaction. This pattern aligns with BigQuery's strengths for large-scale analytical ingestion and queries. See BigQuery performance best practices for nested and repeated data and BigQuery data manipulation language documentation.

QUESTION NO: 10

You are designing storage for very large text files for a data pipeline on Google Cloud. You want to support ANSI SQL queries. You also want to support compression and parallel load from the input locations using Google recommended practices. What should you do?

A. Transform text files to compressed Avro using Cloud Dataflow. Use BigQuery for storage and query.

B. Transform text files to compressed Avro using Cloud Dataflow. Use Cloud Storage and BigQuery permanent linked tables for query.

C. Compress text files to gzip using the Grid Computing Tools. Use BigQuery for storage and query.

D. Compress text files to gzip using the Grid Computing Tools. Use Cloud Storage, and then import into Cloud Bigtable for query.

ANSWER: A

Explanation:

Transform text files to compressed Avro using Cloud Dataflow. Use BigQuery for storage and query. This design combines an efficient, strongly typed, schema-based interchange format with BigQuery's fully managed analytics storage and standard SQL interface. A Dataflow pipeline can read and transform the source text files at scale, serialize records as Avro, and write them in a form suitable for highly parallel ingestion into BigQuery.

Avro is particularly appropriate for this requirement because it is a block-based, splittable format. BigQuery can load Avro data in parallel, and Avro compression is applied at the block level, avoiding the single-stream processing limitation associated with a whole-file compressed text input. BigQuery then stores the data in its managed columnar format and makes it available for ANSI-compliant GoogleSQL queries, without requiring the pipeline to operate or tune a separate query-serving database. This pattern also separates transformation from analytics serving cleanly: Dataflow performs scalable ETL while BigQuery provides durable warehouse storage and interactive or batch SQL analysis.

Google documents Avro as a supported BigQuery load format and notes the loading behavior for compressed files in the BigQuery Avro loading documentation. For implementation details on scalable Dataflow pipelines, see the Dataflow overview.

QUESTION NO: 11

You work on a regression problem in a natural language processing domain, and you have 100M labeled exmaples in your dataset. You have randomly shuffled your data and split your dataset into train and test samples (in a 90/10 ratio). After you

trained the neural network and evaluatedyour model on a test set, you discover that the root-mean-squared error (RMSE) of your model is twice as high on the train set as on the test set. How should you improve the performance of your model?

A. Increase the share of the test sample in the train-test split.

B. Try to collect more data and increase the size of your dataset.

C. Try out regularization techniques (e.g., dropout of batch normalization) to avoid overfitting.

D. Increase the complexity of your model by, e.g., introducing an additional layer or increase sizing the size of vocabularies or n-grams used.

ANSWER: D

Explanation:

Increase the complexity of your model by, e.g., introducing an additional layer or increase sizing the size of vocabularies or n-grams used. is the appropriate action. With a randomly shuffled split and 100 million examples, the training and test sets should be highly representative of the same underlying distribution. An RMSE that remains high on the training data indicates that the current model is not capturing enough of the predictive relationships available even in the examples it learned from. This is a high-bias, or underfitting, pattern rather than a shortage-of-data problem.

Increasing model capacity can let the network represent richer linguistic relationships relevant to the regression target. Depending on the architecture, this may mean adding appropriately sized hidden layers, increasing embedding or representation capacity, or incorporating a broader and more informative vocabulary or n-gram feature set. The revised model should be validated using a held-out evaluation set, with the same preprocessing and inference-time evaluation procedure used for both datasets. Google's machine-learning guidance describes model complexity as a key lever for reducing underfitting, while also emphasizing validation when iterating on model architecture and features. See Google Machine Learning Crash Course: Underfitting and Vertex AI custom training overview.

QUESTION NO: 12

Your application uses Cloud Spanner to store customer orders. A commonly used query filters orders by customer ID and order status, but query latency has increased as the table grows. You want to improve performance while retaining Cloud Spanner scalability and strong consistency. Which two actions should you take? (Choose two.)

A. Create a secondary index on the customer ID and order status columns.

B. Use the query execution plan and profile to validate the query and index usage.

C. Export the orders table to BigQuery before every application query.

D. Store all customer orders in a single Cloud Spanner row.

E. Disable strong consistency for all Cloud Spanner reads.

ANSWER: A B

Explanation:

Create a secondary index that begins with the columns used by the query filter, such as customer ID and order status. A suitable secondary index lets Cloud Spanner locate matching rows without scanning the full base table. Use the query execution plan and profile to verify that the query uses the intended index and to identify remaining expensive operations.

Cloud Spanner query plans provide details about operators, rows processed, and execution behavior. Index design should follow actual query patterns, while avoiding unnecessary indexes because each index also has storage and write-maintenance cost. See Cloud Spanner secondary indexes and Query execution plans.

QUESTION NO: 13

You work for a global shipping company. You want to train a model on 40 TB of data to predict which ships in each geographic region are likely to cause delivery delays on any given day. The model will be based on multiple attributes collected from multiple sources. Telemetry data, including location in GeoJSON format, will be pulled from each ship and loaded every hour. You want to have a dashboard that shows how many and which ships are likely to cause delays within a region. You want to use a storage solution that has native functionality for prediction and geospatial processing. Which storage solution should you use?

A. BigQuery

B. Cloud Bigtable

C. Cloud Datastore

D. Cloud SQL for PostgreSQL

ANSWER: A

Explanation:

BigQuery is the appropriate storage solution because it combines scalable analytics storage with native machine learning and geospatial capabilities. Its serverless architecture can store and query the company's 40 TB dataset while supporting frequent hourly telemetry ingestion. BigQuery ML enables teams to create, train, evaluate, and run prediction models directly with SQL against data stored in BigQuery. This avoids exporting the shipping, operational, and telemetry features to a separate ML platform simply to identify ships with an elevated likelihood of delay.

For location data, BigQuery provides the `GEOGRAPHY` type and geospatial SQL functions. GeoJSON telemetry can be converted into geospatial values and used for region-based analysis, such as determining whether a ship lies within a geographic boundary or aggregating predicted delay risk by area. The resulting prediction and location query outputs can be consumed by a dashboard to show both the number of at-risk ships and their identities in each region. These capabilities make BigQuery well suited to the combined requirements for large-scale storage, in-place prediction, and geospatial processing. See BigQuery ML introduction and Introduction to geospatial analytics in BigQuery.

QUESTION NO: 14

You are migrating an application that tracks library books and information about each book, such as author or year published, from an on-premises data warehouse to BigQuery. In your current relational database, the author information is kept in a separate table and joined to the book information on a common key. Based on Google's recommended practice for schema design, how would you structure the data to ensure optimal speed of queries about the author of each book that has been borrowed?

A. Keep the schema the same, maintain the different tables for the book and each of the attributes, and query as you are doing today.

B. Create a table that is wide and includes a column for each attribute, including the author's first name, last name, date of birth, etc.

C. Create a table that includes information about the books and authors, but nest the author fields inside the author column.

D. Keep the schema the same, create a view that joins all of the tables, and always query the view.

ANSWER: C

Explanation:

Create a table that includes information about the books and authors, but nest the author fields inside the author column. This applies BigQuery's recommended denormalized schema design for hierarchical, commonly queried relationships. Store each book as a row and represent its author as a nested `STRUCT` (record) containing fields such as first name, last name, and date of birth. Author attributes can then be retrieved directly with field paths such as `author.last_name`.

BigQuery is designed for analytical queries over denormalized data. Nesting related data keeps the book and its author physically associated in the same table, avoiding the distributed shuffle and processing overhead that can occur when a

query must join separate large tables. This is particularly useful when queries frequently need to identify or group borrowed books by author. The design also preserves a logical hierarchy without duplicating the author's individual attributes as unrelated top-level columns. If a book can have multiple authors, the nested author record can instead be modeled as a repeated record, allowing multiple author structures for one book row.

Google's guidance on nested and repeated fields is available in Use nested and repeated fields and its schema-design guidance is available in Use nested and repeated fields for performance.

QUESTION NO: 15

You are training a spam classifier. You notice that you are overfitting the training data. Which three actions can you take to resolve this problem? (Choose three.)

A. Get more training examples

B. Reduce the number of training examples

C. Use a smaller set of features

D. Use a larger set of features

E. Increase the regularization parameters

F. Decrease the regularization parameters

ANSWER: A C E

Explanation:

Overfitting occurs when a model learns patterns that are overly specific to its training data, including noise, and consequently does not generalize well to unseen messages. **Get more training examples** is appropriate because a larger, representative training set gives the spam classifier more varied examples of legitimate and spam messages, reducing its tendency to memorize the existing data. **Use a smaller set of features** reduces model complexity. With fewer inputs, the model has fewer opportunities to fit accidental correlations that appear only in the training set. Feature selection is therefore a useful way to improve generalization when the current feature space is unnecessarily large or noisy.

Increase the regularization parameters is also appropriate. Regularization penalizes overly large model weights, encouraging a simpler model that relies less on individual training examples and is less sensitive to noise. In practice, the appropriate amount should be selected using validation data, since excessive regularization can lead to underfitting. These remedies align with Google's guidance that overfitting can be addressed by increasing training data, reducing model complexity, and applying regularization. See Google Machine Learning Crash Course: Regularization for Overfitting and Google Machine Learning Crash Course: Overfitting exercises.

QUESTION NO: 16

Your company is migrating its on-premises data warehousing solution to BigQuery. The existing data warehouse uses trigger-based change data capture (CDC) to apply daily updates from transactional database sources Your company wants to use BigQuery to improve its handling of CDC and to optimize the performance of the data warehouse Source system changes must be available for query m near-real time using tog-based CDC streams You need to ensure that changes in the BigQuery reporting table are available with minimal latency and reduced overhead. What should you do? Choose 2 answers

A. Perform a DML INSERT UPDATE, or DELETE to replicate each CDC record in the reporting table m real time.

B. Periodically use a DML MERGE to simultaneously perform DML INSERT. UPDATE, and DELETE operations in the reporting table

C. Insert each new CDC record and corresponding operation type into a staging table in real time

D. Insert each new CDC record and corresponding operation type into the reporting table in real time and use a materialized view to expose only the current version of each unique record.

ANSWER: B C

Explanation:

“Insert each new CDC record and corresponding operation type into a staging table in real time” is appropriate because it separates frequent, low-latency ingestion from mutations of the curated reporting table. The staging table can retain the incoming change payload and operation indicator, allowing CDC events to be written continuously without requiring a separate target-table mutation for every source transaction. This design also enables deduplication, ordering, and validation before changes are applied to the reporting data set.

“Periodically use a DML MERGE to simultaneously perform DML INSERT. UPDATE, and DELETE operations in the reporting table” is the complementary apply pattern. A single `MERGE` statement can use the staged CDC batch to insert new entities, update existing entities, and delete entities identified by the operation type. Executing this operation at a short, controlled interval provides near-real-time availability while substantially reducing the job, metadata, and table-modification overhead associated with issuing individual DML statements for each CDC event. This is a standard BigQuery pattern for consolidating staged changes into a reporting table. See the BigQuery documentation for the MERGE statement and guidance on streaming data into BigQuery.

QUESTION NO: 17

Which of the following are feature engineering techniques? (Select 2 answers)

A. Hidden feature layers

B. Feature prioritization

C. Crossed feature columns

D. Bucketization of a continuous feature

ANSWER: C D

Explanation:

Feature engineering transforms raw input data into representations that enable a model to learn useful patterns. **Bucketization of a continuous feature** is a feature-engineering transformation that converts a numeric range into discrete intervals, or buckets. For example, a raw age value can be represented by age ranges. This lets models capture non-linear relationships and treat the resulting bucket identifiers as categorical values, which can be particularly useful for linear models and wide-and-deep architectures.

Crossed feature columns are also a feature-engineering technique. A feature cross combines two or more categorical features into a synthetic feature that represents their joint occurrence. For example, crossing a geographic region with a device type enables a model to learn behavior associated with a particular region-device combination rather than considering each attribute only independently. Feature crosses therefore help model interactions that individual base features may not express.

Google Cloud’s machine-learning guidance identifies both transformations as ways to create more informative model inputs. See the Google Machine Learning Crash Course guidance on feature crosses and its documentation on bucketizing continuous features.

QUESTION NO: 18

Your team is building a data lake platform on Google Cloud. As a part of the data foundation design, you are planning to store all the raw data in Cloud Storage You are expecting to ingest approximately 25 GB of data a day and your billing department is worried about the increasing cost of storing old data. The current business requirements are:

• The old data can be deleted anytime

• You plan to use the visualization layer for current and historical reporting

• The old data should be available instantly when accessed

• There should not be any charges for data retrieval.

What should you do to optimize for cost?

A. Create the bucket with the Autoclass storage class feature.

B. Create an Object Lifecycle Management policy to modify the storage class for data older than 30 days to nearline, 90 days to coldline. and 365 days to archive storage class. Delete old data as needed.

C. Create an Object Lifecycle Management policy to modify the storage class for data older than 30 days to coldline, 90 days to nearline. and 365 days to archive storage class Delete old data as needed.

D. Create an Object Lifecycle Management policy to modify the storage class for data older than 30 days to nearlme. 45 days to coldline. and 60 days to archive storage class Delete old data as needed.

ANSWER: A

Explanation:

Create the bucket with the Autoclass storage class feature. Autoclass is designed for workloads whose future access patterns are uncertain, such as a data lake that retains raw data for both current and historical reporting. It automatically transitions objects between Cloud Storage classes according to observed access behavior, beginning with objects stored in Standard storage and moving inactive objects to lower-cost classes over time. This removes the operational burden of selecting and maintaining fixed age-based transition rules while continuously optimizing the storage cost for the actual usage pattern.

Most importantly, Autoclass preserves the stated access and cost requirements. Cloud Storage object access remains immediate across its storage classes; the classes do not impose an access-delay or restore process. For buckets with Autoclass enabled, retrieval fees are waived, including when an object in a colder class is read and subsequently moved back to Standard storage. This makes Autoclass appropriate for historical reporting that may access older data unexpectedly, without introducing per-read retrieval charges. Since old data may be deleted whenever appropriate, retention is not constrained by a required archival duration. See the Cloud Storage Autoclass documentation and the Cloud Storage pricing documentation.

QUESTION NO: 19

You want to migrate an on-premises Hadoop system to Cloud Dataproc. Hive is the primary tool in use, and the data format is Optimized Row Columnar (ORC). All ORC files have been successfully copied to a Cloud Storage bucket. You need to replicate some data to the cluster's local Hadoop Distributed File System (HDFS) to maximize performance. What are two ways to start using Hive in Cloud Dataproc? (Choose two.)

A. Run the gsutil utility to transfer all ORC files from the Cloud Storage bucket to HDFS. Mount the Hive tables locally.

B. Run the gsutil utility to transfer all ORC files from the Cloud Storage bucket to any node of the Dataproc cluster. Mount the Hive tables locally.

C. Run the gsutil utility to transfer all ORC files from the Cloud Storage bucket to the master node of the Dataproc cluster. Then run the Hadoop utility to copy them do HDFS. Mount the Hive tables from HDFS.

D. Leverage Cloud Storage connector for Hadoop to mount the ORC files as external Hive tables. Replicate external Hive tables to the native ones.

E. Load the ORC files into BigQuery. Leverage BigQuery connector for Hadoop to mount the BigQuery tables as external Hive tables. Replicate external Hive tables to the native ones.

ANSWER: C D

Explanation:

Running gsutil to copy the ORC files to the Dataproc master node and then using a Hadoop filesystem command to copy them into HDFS is a valid staged migration approach. The Hadoop command writes the data through HDFS, which distributes and replicates blocks across the cluster's worker nodes according to the HDFS configuration. Hive tables can

then be created as managed or external tables whose locations point to the HDFS-resident ORC data. This provides local HDFS access for workloads where that performance characteristic is required.

Using the Cloud Storage connector for Hadoop is also a supported way to begin using the existing ORC data without first placing every file in HDFS. Dataproc's Hadoop-compatible components, including Hive, can read paths using the `gs://` scheme. Hive external tables can therefore reference the ORC files directly in Cloud Storage. Required data can subsequently be materialized or replicated into native Hive tables backed by HDFS, such as through a Hive query that creates a table from the external source. This supports incremental migration while retaining Cloud Storage as durable object storage. See Cloud Storage connector documentation and Dataproc Hive documentation.

QUESTION NO: 20

You are building a Cloud Dataflow streaming pipeline that calculates sales totals in 10-minute event-time windows. Some purchase events arrive up to 30 minutes late. Business users need preliminary totals shortly after each window closes and corrected totals after late events arrive. Which three actions should you take? (Choose three.)

A. Use fixed windows based on event time.

B. Configure allowed lateness of at least 30 minutes.

C. Configure triggers to emit initial and late updated results.

D. Use processing-time windows for all purchase events.

E. Discard events that arrive after the watermark passes the end of a window.

ANSWER: A B C

Explanation:

Use event-time fixed windows so that purchases are assigned to the business interval in which they occurred, rather than the interval in which Dataflow received them. Configure allowed lateness of at least 30 minutes so that events arriving within the expected delay can still be incorporated into their original windows. Configure triggers that emit an initial result after the watermark passes the end of the window and emit updated panes when late data arrives.

Processing-time windows would assign delayed events based on arrival time and would not produce the intended business totals. Discarding late data would omit transactions that the business requires in corrected totals. See Streaming pipeline concepts and Apache Beam triggers documentation.

QUESTION NO: 21

Your company maintains a hybrid deployment with GCP, where analytics are performed on your anonymized customer data. The data are imported to Cloud Storage from your data center through parallel uploads to a data transfer server running on GCP. Management informs you that the daily transfers take too long and have asked you to fix the problem. You want to maximize transfer speeds. Which action should you take?

A. Increase the CPU size on your server.

B. Increase the size of the Google Persistent Disk on your server.

C. Increase your network bandwidth from your datacenter to GCP.

D. Increase your network bandwidth from Compute Engine to Cloud Storage.

ANSWER: C

Explanation:

Increase your network bandwidth from your datacenter to GCP. This upload path is constrained by the available throughput between the on-premises data center and Google Cloud. Because uploads are already performed in parallel, the transfer process is designed to use the available network capacity efficiently; increasing the capacity of the WAN or internet

connection to Google Cloud raises the maximum amount of data that can arrive at the transfer server per unit of time. This directly addresses the stated objective of maximizing daily transfer speed.

Google Cloud documentation notes that network bandwidth is a key factor in data-transfer performance and recommends evaluating the available bandwidth when moving data into Cloud Storage. For large, recurring hybrid transfers, the organization should provision sufficient connectivity capacity and, where appropriate, consider dedicated connectivity such as Cloud Interconnect to obtain predictable, high-throughput connectivity between the data center and Google Cloud. The relevant limiting segment is the connection bringing data into Google Cloud, so improving that segment increases the end-to-end ingestion rate available to the parallel upload workload. See Cloud Storage upload documentation and Cloud Interconnect overview.

QUESTION NO: 22

Your application publishes order-status events to a Cloud Pub/Sub topic. Events for the same order must be processed in the order in which they were published, while events for different orders can be processed concurrently. Which three actions should you take? (Choose three.)

A. Set the order ID as the ordering key for each published event.

B. Enable message ordering on the subscription.

C. Publish messages with the same ordering key from the same region.

D. Set the acknowledgement deadline to the maximum value.

E. Include a sequential timestamp field in each message body.

ANSWER: A B C

Explanation:

Cloud Pub/Sub message ordering is configured per ordering key. The publisher must assign the same ordering key, such as the order ID, to all events for a specific order. The subscriber must enable message ordering on the subscription so that Pub/Sub delivers messages with the same key in publish order. Publishers that use ordering keys must publish messages for a given ordering key in the same region to preserve ordering.

Different ordering keys can be delivered independently, allowing processing to remain concurrent across orders. Ordering is not achieved by placing a timestamp in the message body, because Pub/Sub does not use application payload fields to determine delivery order. See Order messages.

QUESTION NO: 23

Your company handles data processing for a number of different clients. Each client prefers to use their own suite of analytics tools, with some allowing direct query access via Google BigQuery. You need to secure the data so that clients cannot see each other's data. You want to ensure appropriate access to the data. Which three steps should you take? (Choose three.)

A. Load data into different partitions.

B. Load data into a different dataset for each client.

C. Put each client's BigQuery dataset into a different table.

D. Restrict a client's dataset to approved users.

E. Only allow a service account to access the datasets.

F. Use the appropriate identity and access management (IAM) roles for each client's users.

ANSWER: B D F

Explanation:

Load data into a different dataset for each client establishes a clear security boundary for each tenant. BigQuery permissions can be applied at the dataset level, so isolating each client's tables and views in its own dataset makes it straightforward to grant a client access only to its own data resources. This design also supports clients using their preferred tools for direct BigQuery queries without exposing data belonging to another client.

Restrict a client's dataset to approved users applies least-privilege access control at that security boundary. Access should be granted only to the authenticated users, groups, or workload identities that are authorized to work with that specific client's data. This makes the authorization scope explicit and independently manageable for every client.

Use the appropriate identity and access management (IAM) roles for each client's users ensures that approved identities receive only the capabilities required for their work, such as querying data or viewing metadata. Google Cloud recommends granting the minimum necessary permissions, preferably through predefined roles and groups where appropriate. BigQuery dataset-level access controls and IAM together provide practical tenant isolation and controlled direct-query access. See BigQuery access control with IAM and IAM security best practices.

QUESTION NO: 24

You need to modernize your existing on-premises data strategy. Your organization currently uses.

• Apache Hadoop clusters for processing multiple large data sets, including on-premises Hadoop Distributed File System (HDFS) for data replication.

• Apache Airflow to orchestrate hundreds of ETL pipelines with thousands of job steps.

You need to set up a new architecture in Google Cloud that can handle your Hadoop workloads and requires minimal changes to your existing orchestration processes. What should you do?

A. Use Dataproc to migrate Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases Convert your ETL pipelines to Dataflow.

B. Use Bigtable for your large workloads, with connections to Cloud Storage to handle any HDFS use cases Orchestrate your pipelines with Cloud Composer.

C. Use Dataproc to migrate your Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases. Use Cloud Data Fusion to visually design and deploy your ETL pipelines.

D. Use Dataproc to migrate Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases. Orchestrate your pipelines with Cloud Composer.

ANSWER: D

Explanation:

Use Dataproc to migrate Hadoop clusters to Google Cloud, and Cloud Storage to handle any HDFS use cases. Orchestrate your pipelines with Cloud Composer. Dataproc is Google Cloud's managed service for running open-source data workloads including Apache Hadoop and Spark. Its support for the Hadoop ecosystem allows existing jobs, tools, and processing patterns to move to managed clusters without redesigning the workloads around a different processing framework. Dataproc also integrates with Cloud Storage through the Cloud Storage connector, allowing durable, highly scalable object storage to be used in Hadoop-compatible workflows rather than retaining an on-premises HDFS dependency.

Cloud Composer is the appropriate orchestration service because it is Google Cloud's managed Apache Airflow offering. Existing Airflow DAG-based orchestration can therefore be migrated with minimal process and code changes, while Composer supplies managed Airflow infrastructure and integrates with Dataproc operators for submitting and monitoring cluster jobs. This combination directly preserves both key existing investments: Hadoop-based processing and Airflow-based orchestration, while replacing on-premises storage and operational management with Google Cloud managed services. See the Dataproc overview and the Cloud Composer overview.

QUESTION NO: 25

You want to use Google Stackdriver Logging to monitor Google BigQuery usage. You need an instant notification to be sent to your monitoring tool when new data is appended to a certain table using an insert job, but you do not want to receive notifications for other tables. What should you do?

A. Make a call to the Stackdriver API to list all logs, and apply an advanced filter.

B. In the Stackdriver logging admin interface, and enable a log sink export to BigQuery.

C. In the Stackdriver logging admin interface, enable a log sink export to Google Cloud Pub/Sub, and subscribe to the topic from your monitoring tool.

D. Using the Stackdriver API, create a project sink with advanced log filter to export to Pub/Sub, and subscribe to the topic from your monitoring tool.

ANSWER: D

Explanation:

Using the Stackdriver API, create a project sink with advanced log filter to export to Pub/Sub, and subscribe to the topic from your monitoring tool is correct because a Cloud Logging sink can apply an inclusion filter before routing matching log entries to a destination. The filter can target BigQuery audit log entries for completed insert jobs and constrain the match to the specific destination table, project, dataset, and table name as appropriate. Consequently, only the relevant table-append events are exported rather than activity for every BigQuery table.

Cloud Pub/Sub is the appropriate sink destination when an external monitoring system needs prompt, event-driven delivery. The monitoring tool can create a subscription on the topic and process each matching audit-log message as it arrives. A project-level sink is suitable because BigQuery audit logs for the project can be evaluated centrally, while the advanced filter provides the required precision. Cloud Logging documents how to create sinks and use inclusion filters at Configure log sinks. BigQuery audit logging details, including job and table-related audit metadata that can be used in filters, are documented in BigQuery audit logs.

QUESTION NO: 26

You have a requirement to insert minute-resolution data from 50,000 sensors into a BigQuery table. You expect significant growth in data volume and need the data to be available within 1 minute of ingestion for real-time analysis of aggregated trends. What should you do?

A. Use bq load to load a batch of sensor data every 60 seconds.

B. Use a Cloud Dataflow pipeline to stream data into the BigQuery table.

C. Use the INSERT statement to insert a batch of data every 60 seconds.

D. Use the MERGE statement to apply updates in batch every 60 seconds.

ANSWER: B

Explanation:

Use a Cloud Dataflow pipeline to stream data into the BigQuery table. Dataflow is a fully managed, autoscaling service for executing streaming data pipelines, making it appropriate for continuously ingesting telemetry from a large and growing sensor fleet. A streaming pipeline can read events as they arrive, apply any required validation, timestamp normalization, windowing, or aggregation, and continuously write the results to BigQuery. This design avoids relying on scheduled, minute-by-minute batch jobs and supports the required near-real-time analytics use case as input throughput grows.

BigQuery supports streaming ingestion through the BigQuery Storage Write API, and Dataflow provides BigQuery I/O connectors that support writing pipeline output to BigQuery. Data written through streaming ingestion is intended for low-latency availability for querying, while Dataflow provides operational capabilities such as autoscaling, monitoring, retry handling, and durable processing for the ingestion path. For aggregated trend analysis, the pipeline can also use event-time windows and emit continuously updated aggregates or raw events for downstream BigQuery SQL analysis. See the Dataflow overview and BigQuery Storage Write API documentation.

QUESTION NO: 27

You decided to use Cloud Datastore to ingest vehicle telemetry data in real time. You want to build a storage system that will account for the long-term data growth, while keeping the costs low. You also want to create snapshots of the data periodically, so that you can make a point-in-time (PIT) recovery, or clone a copy of the data for Cloud Datastore in a different environment. You want to archive these snapshots for a long time. Which two methods can accomplish this? (Choose two.)

A. Use managed export, and store the data in a Cloud Storage bucket using Nearline or Coldline class.

B. Use managed export, and then import to Cloud Datastore in a separate project under a unique namespace reserved for that export.

C. Use managed export, and then import the data into a BigQuery table created just for that export, and delete temporary export files.

D. Write an application that uses Cloud Datastore client libraries to read all the entities. Treat each entity as a BigQuery table row via BigQuery streaming insert. Assign an export timestamp for each export, and attach it as an extra column for each row. Make sure that the BigQuery table is partitioned using the export timestamp column.

E. Write an application that uses Cloud Datastore client libraries to read all the entities. Format the exported data into a JSON file. Apply compression before storing the data in Cloud Source Repositories.

ANSWER: A B

Explanation:

Using managed export and storing the export artifacts in a Cloud Storage bucket with the Nearline or Coldline storage class provides a durable, low-cost archival snapshot. A managed export produces a consistent copy of the Datastore data at a point in time, and the resulting files can be retained under Cloud Storage lifecycle and retention policies for the required archival period. Those export files can later be used by the managed import service to restore data, supporting point-in-time recovery. Cloud Storage's colder classes are designed for infrequently accessed data and are appropriate for long-lived backup copies.

Using managed export followed by managed import into Cloud Datastore in a separate project and a namespace dedicated to that export creates an isolated, usable clone of the snapshot. A separate project provides environment isolation, while the namespace prevents the imported entities from colliding with other entities in the destination database. Because the source export is a point-in-time managed export, the imported data represents a consistent historical copy suitable for recovery validation, development, testing, or migration workflows. Google documents managed exports and imports as the supported mechanism for transferring Datastore data between databases and projects. See Exporting and importing entities and Cloud Storage classes.

QUESTION NO: 28

You work for a large ecommerce company. You are using Pub/Sub to ingest the clickstream data to Google Cloud for analytics. You observe that when a new subscriber connects to an existing topic to analyze data, they are unable to subscribe to older data for an upcoming yearly sale event in two months, you need a solution that, once implemented, will enable any new subscriber to read the last 30 days of data. What should you do?

A. Create a new topic, and publish the last 30 days of data each time a new subscriber connects to an existing topic.

B. Set the topic retention policy to 30 days.

C. Set the subscriber retention policy to 30 days.

D. Ask the source system to re-push the data to Pub/Sub, and subscribe to it.

ANSWER: B

Explanation:

Set the topic retention policy to 30 days. Pub/Sub topic message retention keeps published messages available at the topic for the configured retention interval, independently of whether an existing subscription has acknowledged them. This is specifically useful when a subscription is created after messages have already been published: the new subscription can access messages retained by the topic, including historical clickstream events from the prior 30 days. Topic-level retention therefore meets the requirement without republishing data or creating separate topics for each analytics consumer.

After enabling a 30-day topic retention period, a newly created subscription can be positioned to consume the retained history, such as by using Pub/Sub seek functionality with a timestamp within that window when needed. The retention duration must be explicitly configured because topic message retention is not enabled by default. This design gives the ecommerce company a predictable replay horizon for new analysis workloads while retaining the scalable fan-out behavior of Pub/Sub: each subscriber can process the same retained clickstream data independently. Google Cloud documents topic message retention and its availability to subscriptions created after publication in the Pub/Sub message retention documentation. For operational use of historical replay positions, see Replay messages with seek.

QUESTION NO: 29

Your company is in the process of migrating its on-premises data warehousing solutions to BigQuery. The existing data warehouse uses trigger-based change data capture (CDC) to apply updates from multiple transactional database sources on a daily basis. With BigQuery, your company hopes to improve its handling of CDC so that changes to the source systems are available to query in BigQuery in near-real time using log-based CDC streams, while also optimizing for the performance of applying changes to the data warehouse. Which two steps should they take to ensure that changes are available in the BigQuery reporting table with minimal latency while reducing compute overhead? (Choose two.)

A. Perform a DML INSERT, UPDATE, or DELETE to replicate each individual CDC record in real time directly on the reporting table.

B. Insert each new CDC record and corresponding operation type to a staging table in real time.

C. Periodically DELETE outdated records from the reporting table.

D. Periodically use a DML MERGE to perform several DML INSERT, UPDATE, and DELETE operations at the same time on the reporting table.

E. Insert each new CDC record and corresponding operation type in real time to the reporting table, and use a materialized view to expose only the newest version of each unique record.

ANSWER: B D

Explanation:

Insert each new CDC record and corresponding operation type to a staging table in real time. This decouples low-latency ingestion from the more expensive task of reconciling changes in the reporting table. Each incoming change can include the source key, change timestamp or sequence value, operation type, and changed values, preserving the information needed to deterministically apply inserts, updates, and deletes. BigQuery can then make the staged stream available for querying quickly while avoiding a separate table mutation for every source event.

Periodically use a DML MERGE to perform several DML INSERT, UPDATE, and DELETE operations at the same time on the reporting table. A single `MERGE` statement can atomically compare staged source changes with the target reporting table and apply the appropriate action. Batching these operations substantially reduces DML job and table-rewrite overhead compared with executing individual mutations per CDC event, while the merge interval can be selected to meet the required reporting freshness. This staged micro-batch pattern is a standard BigQuery approach for efficiently maintaining a current-state analytical table from a continuously arriving change stream. See the BigQuery MERGE statement documentation and Google's change data capture guidance.

QUESTION NO: 30

You have Cloud Functions written in Node.js that pull messages from Cloud Pub/Sub and send the data to BigQuery. You observe that the message processing rate on the Pub/Sub topic is orders of magnitude higher than anticipated, but there is no error logged in Stackdriver Log Viewer. What are the two most likely causes of this problem? (Choose two.)

A. Publisher throughput quota is too small.

B. Total outstanding messages exceed the 10-MB maximum.

C. Error handling in the subscriber code is not handling run-time errors properly.

D. The subscriber code cannot keep up with the messages.

E. The subscriber code does not acknowledge the messages that it pulls.

ANSWER: C E

Explanation:

“Error handling in the subscriber code is not handling run-time errors properly” and “The subscriber code does not acknowledge the messages that it pulls” are the most likely causes because Cloud Pub/Sub uses at-least-once delivery. A pulled message remains outstanding until the subscriber acknowledges it. If acknowledgement does not occur before the acknowledgement deadline expires, Pub/Sub can redeliver the same message. Repeated delivery of the same set of messages can make observed subscriber processing throughput far exceed the publishing rate, even though the topic is not receiving that many distinct events.

In a Cloud Functions pipeline, an unhandled or inadequately handled runtime failure can prevent the successful acknowledgement path from being reached. If application code catches an error without logging it, or terminates processing without explicitly acknowledging the message, Stackdriver Log Viewer may show no useful error while Pub/Sub continues redelivering. The function must acknowledge messages only after the BigQuery write and any required processing complete successfully, while ensuring failures are logged and handled in a way that supports retries or dead-letter processing. Google documents that Pub/Sub may redeliver messages when acknowledgements are not received within the deadline and that subscribers should be designed for redelivery. See Acknowledging messages and Subscriber best practices.

QUESTION NO: 31

You need to create a new transaction table in Cloud Spanner that stores product sales data. You are deciding what to use as a primary key. From a performance perspective, which strategy should you choose?

A. The current epoch time

B. A concatenation of the product name and the current epoch time

C. A random universally unique identifier number (version 4 UUID)

D. The original order identification number from the sales system, which is a monotonically increasing integer

ANSWER: C

Explanation:

A random universally unique identifier number (version 4 UUID) is the appropriate primary-key strategy because Cloud Spanner stores rows in primary-key order and partitions data into contiguous key ranges. A version 4 UUID is generated randomly, so newly inserted sales transactions are distributed across many key ranges rather than continually targeting the same trailing range. This distribution allows Cloud Spanner to spread write work across its serving infrastructure and helps maintain throughput as transaction volume grows.

For a transaction table, the primary key must also uniquely identify each sale. Random UUIDs provide an extremely large identifier space, making collisions negligibly unlikely when generated correctly. This makes them suitable for independently created transaction records while avoiding a write concentration pattern. Cloud Spanner documentation specifically recommends avoiding monotonically increasing primary-key values for high-write tables and using a key design that distributes writes. See the guidance on preventing hotspots with primary-key design and the Cloud Spanner schema and data model documentation.

QUESTION NO: 32

You need to compose visualizations for operations teams with the following requirements:

Which approach meets the requirements?

A. Load the data into Google Sheets, use formulas to calculate a metric, and use filters/sorting to show only suboptimal links in a table.

B. Load the data into Google BigQuery tables, write Google Apps Script that queries the data, calculates the metric, and shows only suboptimal rows in a table in Google Sheets.

C. Load the data into Google Cloud Datastore tables, write a Google App Engine Application that queries all rows, applies a function to derive the metric, and then renders results in a table using the Google charts and visualization API.

D. Load the data into Google BigQuery tables, write a Google Data Studio 360 report that connects to your data, calculates a metric, and then uses a filter expression to show only suboptimal rows in a table.

ANSWER: D

Explanation:

Loading the data into Google BigQuery tables and using a Google Data Studio 360 report to connect to the data, calculate a metric, and filter the table to suboptimal rows is the appropriate managed analytics and visualization approach. BigQuery is designed to store and query large operational datasets efficiently using SQL, avoiding the scalability and maintenance constraints of spreadsheet-based processing or custom application code. A BI reporting layer can then define a calculated field for the operational metric and apply a report, chart, or control-level filter so that the operations team sees only links requiring attention. This provides an interactive, shareable visualization experience while keeping the underlying data processing in a purpose-built analytical warehouse. Google Data Studio has since been renamed Looker Studio; the relevant capability remains support for BigQuery connectors, calculated fields, and filters. See the BigQuery overview and the Looker Studio BigQuery connector documentation.

QUESTION NO: 33

Which of these rules apply when you add preemptible workers to a Dataproc cluster (select 2 answers)?

A. Preemptible workers cannot use persistent disk.

B. Preemptible workers cannot store data.

C. If a preemptible worker is reclaimed, then a replacement worker must be added manually.

D. A Dataproc cluster cannot have only preemptible workers.

ANSWER: B D

Explanation:

“Preemptible workers cannot store data.” is correct because Dataproc configures preemptible workers as secondary workers intended for scalable processing capacity rather than durable HDFS storage. They do not act as HDFS DataNodes, so HDFS does not place persistent cluster data on them. Although each VM has a boot persistent disk for its operating system and temporary local caching, that disk is not used to provide HDFS storage and its contents should not be treated as durable workload data. This design allows tasks such as Spark or MapReduce work to use low-cost, interruptible capacity while retaining durable data and essential cluster services on primary workers or external storage such as Cloud Storage.

“A Dataproc cluster cannot have only preemptible workers.” is also correct. A Dataproc cluster requires primary workers, which provide the stable worker capacity and run core components, including HDFS DataNode services when HDFS is used. Preemptible workers are supplemental secondary workers and can be reclaimed by Compute Engine; Dataproc uses managed instance groups to attempt replacement as capacity permits. Keeping primary workers ensures the cluster retains its foundational worker infrastructure even when secondary capacity is interrupted. See the Dataproc secondary worker documentation and Dataproc worker configuration documentation.

QUESTION NO: 34

Which of these numbers are adjusted by a neural network as it learns from a training dataset (select 2 answers)?

A. Weights

B. Biases

C. Continuous features

D. Input values

ANSWER: A B

Explanation:

Neural-network training adjusts the model's trainable parameters: **weights** and **biases**. A weight controls the strength and direction of the relationship between an input (or a prior layer's activation) and a neuron. During training, the optimizer uses the loss function's gradients to update each weight, reducing prediction error over successive iterations. A bias is also a learned parameter. It provides an offset for a neuron's weighted sum before the activation function is applied, allowing the model to fit patterns that do not need to pass through the origin. Together, weights and biases define the function that the network learns from the training data. Input values and continuous-feature values are supplied to the model as examples; they are not generally changed by the neural network's optimization process. Feature preprocessing, such as normalization or transformation, may occur in a separate data-preparation step, but this is distinct from learning the network parameters. Google's machine learning materials describe neural networks as learning weights and biases through training and gradient-based optimization. See Google Machine Learning Crash Course: Introduction to Neural Networks and TensorFlow: Training and evaluation with Keras.

QUESTION NO: 35

You have Google Cloud Dataflow streaming pipeline running with a Google Cloud Pub/Sub subscription as the source. You need to make an update to the code that will make the new Cloud Dataflow pipeline incompatible with the current version. You do not want to lose any data when making this update. What should you do?

A. Update the current pipeline and use the drain flag.

B. Update the current pipeline and provide the transform mapping JSON object.

C. Create a new pipeline that has the same Cloud Pub/Sub subscription and cancel the old pipeline.

D. Create a new pipeline that has a new Cloud Pub/Sub subscription and cancel the old pipeline.

ANSWER: C

Explanation:

Create a new pipeline that has the same Cloud Pub/Sub subscription and cancel the old pipeline. is the appropriate approach when the revised pipeline is incompatible with an in-place Dataflow update. The new job can read from the existing subscription, which preserves the subscription's message backlog and delivery state rather than introducing a separate subscription with a different message-retention history. After the new pipeline is running successfully, the old job can be canceled. Messages that had not been successfully acknowledged by the old job remain available for delivery from the shared subscription and can be processed by the new job. This approach prioritizes avoiding message loss; downstream processing should remain idempotent because Pub/Sub delivery is at-least-once and a message can be redelivered during the transition. Dataflow's update mechanism is intended for compatible graph changes and requires a mapping for changed transform names when applicable. For an incompatible graph, deploying a separate replacement job is the suitable migration pattern. See the Dataflow pipeline update guide and the Pub/Sub subscriber documentation for subscription delivery and acknowledgment behavior.

QUESTION NO: 36

Your company is migrating their 30-node Apache Hadoop cluster to the cloud. They want to re-use Hadoop jobs they have already created and minimize the management of the cluster as much as possible. They also want to be able to persist data beyond the life of the cluster. What should you do?

A. Create a Google Cloud Dataflow job to process the data.

B. Create a Google Cloud Dataproc cluster that uses persistent disks for HDFS.

C. Create a Hadoop cluster on Google Compute Engine that uses persistent disks.

D. Create a Cloud Dataproc cluster that uses the Google Cloud Storage connector.

E. Create a Hadoop cluster on Google Compute Engine that uses Local SSD disks.

ANSWER: D

Explanation:

Create a Cloud Dataproc cluster that uses the Google Cloud Storage connector. Dataproc is a managed service for running open-source data-processing frameworks, including Apache Hadoop and Spark, so the organization can migrate and continue running its existing Hadoop jobs without building and operating the cluster infrastructure itself. Google manages key cluster provisioning and operational integration, while Dataproc also supports creating ephemeral clusters for individual workloads or scheduled processing.

The Cloud Storage connector makes Cloud Storage available to Hadoop applications through the `gs://` filesystem scheme. Storing input data, output data, and other durable artifacts in Cloud Storage separates storage from the lifetime of Dataproc compute resources. The cluster can therefore be deleted after processing is complete without deleting the data, and a later Dataproc cluster can access the same Cloud Storage paths. This architecture reduces ongoing cluster management and avoids retaining a running HDFS-based storage cluster solely to preserve data. It also supports elasticity, because compute resources can be sized for the job rather than permanently sized around retained storage. See the Dataproc overview and Google's Cloud Storage connector documentation.

QUESTION NO: 37

Your company receives raw application logs in a Cloud Storage bucket. Logs are accessed frequently for the first 30 days, rarely during the following 23 months, and must be removed after two years. You want to minimize storage-management effort and storage cost. Which three actions should you take? (Choose three.)

A. Create a lifecycle rule that transitions objects to Nearline storage after 30 days.

B. Create a lifecycle rule that transitions objects to Coldline storage after 90 days.

C. Create a lifecycle rule that deletes objects after two years.

D. Enable object versioning and retain every noncurrent version indefinitely.

E. Copy all objects manually to a second bucket at the end of each month.

ANSWER: A B C

Explanation:

Create Cloud Storage lifecycle rules to transition objects to Nearline storage after 30 days, transition them to Coldline storage after 90 days, and delete them after two years. Lifecycle management automatically performs these actions as objects reach the configured age, avoiding manual movement or cleanup of the files.

Nearline and Coldline storage classes are designed for increasingly infrequent access. A deletion rule at two years implements the stated retention period and prevents unnecessary ongoing storage charges after the data is no longer required. See Object Lifecycle Management and Cloud Storage classes.

QUESTION NO: 38

You need to deploy additional dependencies to all of a Cloud Dataproc cluster at startup using an existing initialization action. Company security policies require that Cloud Dataproc nodes do not have access to the Internet so public initialization actions cannot fetch resources. What should you do?

A. Deploy the Cloud SQL Proxy on the Cloud Dataproc master

B. Use an SSH tunnel to give the Cloud Dataproc cluster access to the Internet

C. Copy all dependencies to a Cloud Storage bucket within your VPC security perimeter

D. Use Resource Manager to add the service account used by the Cloud Dataproc cluster to the Network User role

ANSWER: C

Explanation:

Copy all dependencies to a Cloud Storage bucket within your VPC security perimeter is correct because Dataproc initialization actions run on each relevant cluster VM during cluster creation and can retrieve scripts and supporting artifacts from Cloud Storage. Place the initialization script and every required package, archive, binary, or configuration artifact in a controlled bucket, then update the existing action to use those Cloud Storage paths rather than attempting to download content from public internet locations. The Dataproc VM service account must have the appropriate permission to read the objects, commonly through a narrowly scoped Cloud Storage IAM role such as Storage Object Viewer on the designated bucket. If the organization uses VPC Service Controls, keeping the bucket and Dataproc resources in the same service perimeter supports the policy goal by allowing approved Google Cloud API access while helping prevent data access outside the perimeter. This design provides repeatable, startup-time installation across cluster nodes without granting outbound internet access. Google documents that initialization actions can be stored in Cloud Storage and specifies the permissions required for the Dataproc VM service account. See Dataproc initialization actions and VPC Service Controls overview.

QUESTION NO: 39

Your company is using WHILECARD tables to query data across multiple tables with similar names. The SQL statement is currently failing with the following error:

Syntax error : Expected end of statement but got “-“ at [4:11]

SELECT age

FROM

bigquery-public-data.noaa_gsod.gsod

WHERE

age != 99

AND_TABLE_SUFFIX = ‘1929’

ORDER BY

age DESC

Which table name will make the SQL statement work correctly?

A. ‘bigquery-public-data.noaa_gsod.gsod‘

B. bigquery-public-data.noaa_gsod.gsod*

C. ‘bigquery-public-data.noaa_gsod.gsod’*

D. `bigquery-public-data.noaa_gsod.gsod*`

ANSWER: D

Explanation:

`bigquery-public-data.noaa_gsod.gsod*` is the required table expression because it uses a wildcard table name and encloses the complete project, dataset, and table pattern in backticks. In GoogleSQL, backticks delimit a fully qualified table identifier; they are particularly necessary here because the project ID contains hyphens. The asterisk must be inside the backticks so BigQuery interprets it as a wildcard matching all tables whose names begin with `gsod`, such as a table ending in `1929`. When querying wildcard tables, BigQuery provides the `_TABLE_SUFFIX` pseudocolumn. The predicate `_TABLE_SUFFIX = '1929'` then restricts the scanned wildcard match to the table with that suffix, while the remainder of the query filters the age values and sorts the result. This syntax uses standard GoogleSQL wildcard-table semantics and lets BigQuery resolve the table pattern correctly. See the BigQuery documentation for querying wildcard tables and the GoogleSQL reference for quoted identifiers.

QUESTION NO: 40

You're using Bigtable for a real-time application, and you have a heavy load that is a mix of read and writes. You've recently identified an additional use case and need to perform hourly an analytical job to calculate certain statistics across the whole database. You need to ensure both the reliability of your production application as well as the analytical workload.

What should you do?

A. Export Bigtable dump to GCS and run your analytical job on top of the exported files.

B. Add a second cluster to an existing instance with a multi-cluster routing, use live-traffic app profile for your regular workload and batch-analytics profile for the analytics workload.

C. Add a second cluster to an existing instance with a single-cluster routing, use live-traffic app profile for your regular workload and batch-analytics profile for the analytics workload.

D. Increase the size of your existing cluster twice and execute your analytics workload on your new resized cluster.

ANSWER: B

Explanation:

Add a second cluster to an existing instance with a multi-cluster routing, use live-traffic app profile for your regular workload and batch-analytics profile for the analytics workload. Bigtable clusters in the same replicated instance hold copies of the same data, allowing the analytical job to read the complete dataset without requiring a periodic export and without placing all workload demand on the production cluster. Application profiles are the mechanism Bigtable provides for associating a workload with its intended routing behavior. A live-traffic profile can use multi-cluster routing to preserve application availability if a cluster becomes unavailable, while a batch-analytics profile can direct the scheduled job according to its workload-routing configuration. This arrangement separates operational concerns at the client-routing layer and supplies independent cluster capacity for the hourly scan-heavy analytics work. Because replication is asynchronous, the analytical computation should also tolerate a small amount of replication lag if it requires data written immediately before the job begins. Google recommends using replication and application profiles to support workload isolation and availability requirements in Bigtable. See the Bigtable application profiles documentation and the Bigtable replication overview.

QUESTION NO: 41

Your company produces 20,000 files every hour. Each data file is formatted as a comma separated values (CSV) file that is less than 4 KB. All files must be ingested on Google Cloud Platform before they can be processed. Your company site has a 200 ms latency to Google Cloud, and your Internet connection bandwidth is limited as 50 Mbps. You currently deploy a secure FTP (SFTP) server on a virtual machine in Google Compute Engine as the data ingestion point. A local SFTP client runs on a dedicated machine to transmit the CSV files as is. The goal is to make reports with data from the previous day available to the executives by 10:00 a.m. each day. This design is barely able to keep up with the current volume, even though the bandwidth utilization is rather low. You are told that due to seasonality, your company expects the number of files to double for the next three months. Which two actions should you take? (Choose two.)

A. Introduce data compression for each file to increase the rate file of file transfer.

B. Contact your internet service provider (ISP) to increase your maximum bandwidth to at least 100 Mbps.

C. Redesign the data ingestion process to use gsutil tool to send the CSV files to a storage bucket in parallel.

D. Assemble 1,000 files into a tape archive (TAR) file. Transmit the TAR files instead, and disassemble the CSV files in the cloud upon receiving them.

E. Create an S3-compatible storage endpoint in your network, and use Google Cloud Storage Transfer Service to transfer on-premises data to the designated storage bucket.

ANSWER: C D

Explanation:

The appropriate actions are to upload the CSV objects in parallel with gsutil and to package groups of files into TAR archives before transmission. The workload consists of extremely small files and has a relatively high 200 ms network latency. In this situation, end-to-end throughput is limited primarily by per-file connection, request, acknowledgement, and protocol overhead rather than by the available network bandwidth. This matches the observation that the current design is near capacity while bandwidth utilization remains low.

Using gsutil's parallel composite upload capabilities enables multiple file uploads to proceed concurrently, increasing utilization of the available connection and avoiding a strictly serial transfer pattern. Google documents parallel operations through the `-m` option, which is designed to perform operations across multiple processes and threads: gsutil command-line documentation.

Combining 1,000 small CSV files into each TAR archive drastically reduces the number of remote transfer operations. Fewer, larger transfers amortize high-latency protocol costs over substantially more payload and make the projected doubled file count manageable. After upload, the archives can be extracted in Google Cloud for downstream processing. This follows Cloud Storage guidance to avoid excessive operations involving very small objects and to batch data where practical: Cloud Storage performance optimization guidance.

QUESTION NO: 42

Your financial services company is moving to cloud technology and wants to store 50 TB of financial time-series data in the cloud. This data is updated frequently and new data will be streaming in all the time. Your company also wants to move their existing Apache Hadoop jobs to the cloud to get insights into this data. Which product should they use to store the data?

A. Cloud Bigtable

B. Google BigQuery

C. Google Cloud Storage

D. Google Cloud Datastore

ANSWER: A

Explanation:

Cloud Bigtable is the appropriate storage service for this workload because it is a distributed, wide-column NoSQL database designed for very large data volumes, sustained high-throughput writes, and low-latency access. Financial time-series data commonly has a timestamped stream of new records and frequent updates, patterns that Bigtable can support effectively when the row-key schema is designed to distribute writes and enable the required time-range queries. A 50 TB dataset is also well aligned with Bigtable's scalable architecture.

Bigtable is especially suitable when an organization is migrating Hadoop-oriented data workloads because it supports the Apache HBase API. This compatibility enables applications and processing jobs that use HBase clients and Hadoop ecosystem integrations to access Bigtable with relatively limited application changes. The service can therefore act as the operational store for continuously arriving time-series records while cloud-based processing jobs derive insights from that data. Google's time-series schema guidance specifically discusses patterns for storing time-series data in Bigtable, including considerations for row keys, timestamped data, and query patterns. See Bigtable time-series schema design and Bigtable HBase compatibility.

QUESTION NO: 43

You are building a model to make clothing recommendations. You know a userâ€™s fashion preference is likely to change over time, so you build a data pipeline to stream new data back to the model as it becomes available. How should you use this data to train the model?

A. Continuously retrain the model on just the new data.

B. Continuously retrain the model on a combination of existing data and the new data.

C. Train on the existing data while using the new data as your test set.

D. Train on the new data while using the existing data as your test set.

ANSWER: B

Explanation:

Continuously retrain the model on a combination of existing data and the new data. This approach lets the recommendation model learn emerging fashion preferences and behavioral patterns while retaining the broader historical signal needed to make reliable recommendations. Training exclusively on the latest stream of data can overemphasize short-lived trends, seasonal effects, or a limited recent user cohort. Combining appropriately selected historical examples with newly arrived examples provides a more representative training distribution and helps prevent the model from losing patterns that remain useful.

In practice, the pipeline should periodically create a new training dataset that includes recent interactions plus retained historical data, then train and evaluate a candidate model before deployment. The historical portion may be sampled, weighted, or constrained to a recency window based on how quickly preferences change and on observed data drift. Google Cloud recommends monitoring production models for skew and drift, which can indicate that retraining with newer data is needed. Evaluation should also use a holdout dataset that reflects the intended production population and time horizon, rather than treating all newly streamed records as training-only data. See Google Cloud guidance on Vertex AI model monitoring and preparing data for Vertex AI training.