CERTLABZ CERTIFICATION BOARD

Certified Data Engineering Professional (CDEP)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working data engineer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1 — Domain 1: Data Engineering Foundations and Architecture — DATA

Emma Wilson matches data stores to their roles.

Store	Role
OLTP database	Many small transactions
Data warehouse	Integrated structured analytics
Data lake	Raw data at scale on cheap storage
Lakehouse	Lake storage with warehouse features

Table 1.1 Stores and roles.

A team wants to run heavy analytical queries without harming the production application. What should they use?

A. Run the analytical queries directly against the OLTP database, because it already holds the freshest copy of all the data.

B. Use a data warehouse (or lakehouse) for analytics, so large scans do not slow or destabilise the transactional workload.

C. Add more CPUs to the OLTP database, because heavy analytics on a transactional system is fine once the hardware is large enough.

D Build the dashboards to query the OLTP database less often, since reducing query frequency removes any impact on the application.

Correct answer: B
Heavy analytical scans on the production OLTP database can slow or destabilise the transactional workload, so analytics belongs on a system optimised for it, a warehouse or lakehouse. Adding CPUs or querying less often does not make running large analytics on the operational database a sound design.

Question 2 — Domain 2: Data Modeling and Storage — DATA

James Carter has slow, costly analytical queries scanning huge tables.

Choice	Current state
File format	Row-based text
Partitioning	None
Columns read per query	A few of many
Cost	High, full scans

Table 2.1 Current storage.

Which changes will most improve query performance and cost?

A Keep row-based text but run the queries more frequently, because repeated runs let the warehouse cache and speed up the scans.

B Switch to a columnar format such as Parquet and partition the data, so queries read only the columns and partitions they need.

C Store everything in one giant unpartitioned file, because a single file is simplest for the engine to scan quickly each time.

D Remove all keys and constraints from the tables, because fewer constraints always makes analytical queries run faster and cheaper.

Correct answer: B
Columnar formats let queries read only the needed columns with strong compression, and partitioning lets them skip irrelevant partitions, together cutting scan cost dramatically. Running row-based full scans more often, using one giant file, or dropping constraints does not address the cost of scanning far more data than needed.

Question 3 — Domain 3: Ingestion, ETL/ELT and Pipelines — CONFIG

Olivia Davis designs a robust incremental ingestion, listed out of order.

```
A. Load atomically (stage then swap)
B. Read only changed data via a watermark
C. Validate schema and data quality
D. Deduplicate on the key
```

Listing 3.1 Steps to order.

What is the correct order of these steps?

A. A, then C, then B, then D, loading atomically before any changed data has been read from the source or validated at all.

B. B, then C, then D, then A, reading changed data, validating it, deduplicating, then loading it atomically into the target.

C. D, then A, then B, then C, deduplicating before any data has been read and loading before it has been validated for quality.

D. C, then A, then D, then B, validating and loading before the changed data has been read from the source using the watermark.

Correct answer: B
A robust incremental load reads only changed data via a watermark (B), validates schema and quality (C), deduplicates on the key (D), then loads atomically with a stage-then-swap (A). The other orders load or deduplicate before the data has even been read or validated.

Question 4

Domain 4: Batch and Streaming Processing

DATA

Liam Anderson chooses processing models for four needs.

Need	Best model
Fraud alerts within seconds	(to choose)
Report viewed once a day	(to choose)
Hourly totals with late events	(to choose)
One partition holds most data	(to choose)

Table 4.1 Processing needs.

Which set of choices is correct?

A. Batch for fraud alerts, streaming for the daily report, processing-time windows for late events, and ignoring the data-skew problem.

B. Streaming for fraud alerts, scheduled batch for the daily report, event-time windows with late handling, and repartitioning to fix skew.

C. Streaming for both the fraud alerts and the daily report, processing-time windows for late events, and adding a dashboard to fix skew.

D. Batch for both needs, ignoring event time for the late events, and buying a bigger warehouse to resolve the data-skew problem entirely.

Correct answer: B
Fraud alerts within seconds need streaming, a once-a-day report is best served by a simpler, cheaper scheduled batch, hourly totals with late events need event-time windows with late-data handling, and one overloaded partition is data skew fixed by repartitioning. The other options misapply the models or ignore event time and skew.

Question 5

Domain 5: Data Quality, Orchestration and Reliability

CONFIG

Ava Thompson reviews a pipeline after a downstream report was wrong for days.

```
orchestration : cron job, no dependency handling
on failure    : no alert, silent
data checks   : none (nulls, volume, freshness)
retries       : none; a failed run leaves stale data
```

Listing 5.1 The current pipeline.

Which improvements would have caught or prevented the incident?

A. Run the same cron job more frequently, because running it more often would have refreshed the stale data before anyone noticed the issue.

B. Add failure alerting, data quality and freshness checks, and idempotent retries, so silent failures and stale data are caught and handled.

C. Replace the report with a nicer dashboard, because a clearer visual presentation would have made the wrong numbers easier to spot sooner.

D. Move the pipeline to a larger warehouse, because more compute is what prevents a pipeline from silently failing and serving stale data.

Correct answer: B
The incident was a silent failure that left stale data with no checks or alerts. Failure alerting, data quality and freshness checks, and idempotent retries would have caught it and kept the data correct. Running the broken job more often, a prettier dashboard, or a bigger warehouse does not address the missing reliability controls.

Question 6

Domain 6: Governance, Security and Cost

DATA

A cloud warehouse bill spikes and a test environment holds sensitive data.

Finding	Detail
Query pattern	Dashboards run full-table scans
Storage	Row-based, unpartitioned
Test environment	Full copy of production PII
Old data	Never expired

Table 6.1 Findings from a review.

Which set of actions best addresses cost and privacy together?

A. Buy a larger warehouse, keep the full-table scans, copy the production PII into more environments, and keep all old data indefinitely.

B. Partition and use columnar storage to scan less, mask or use synthetic data in test, and set retention and lifecycle policies on old data.

C. Run the dashboards more frequently, encrypt only the dashboard, leave the PII copy in test, and archive nothing so all data stays available.

D. Disable monitoring to cut cost, delete the warehouse, keep the raw PII in test for realism, and expire data at random to reduce storage.

Correct answer: B
Partitioning and columnar storage make queries scan far less, cutting cost; masking or synthetic data removes needless PII exposure in test; and retention and lifecycle policies control both cost and privacy risk from old data. The other options increase spend, keep sensitive data exposed, or remove needed controls.

Question 7

Domain 1: Data Engineering Foundations and Architecture

A pipeline that is rerun after a failure duplicates rows in the target table. What property was missing, and why does it matter?

A. Nothing was missing, because duplicating rows on a rerun is expected behaviour and the duplicates can simply be ignored downstream.

B. Idempotence: a step should be safe to retry so a rerun produces the same result without creating duplicates or corrupting data.

C. The pipeline needed a larger warehouse, because duplicates on rerun are caused by insufficient compute rather than by the pipeline design.

D. The pipeline should never be rerun, so the fix is to forbid retries entirely and require a manual full rebuild after every single failure.

Correct answer: B
A step that duplicates data when rerun is not idempotent. Idempotent design means a retry produces the same result without duplication or corruption, which is essential because failures and retries are normal. Ignoring duplicates, adding compute, or banning retries does not fix the underlying design flaw.

Question 8

Domain 3: Ingestion, ETL/ELT and Pipelines

A nightly full reload of a very large table has become too slow and expensive. What is the better approach?

A. Run the same full reload more frequently, because splitting the full reload across more runs each night reduces its total cost.

B. Switch to incremental loading, using change data capture or a watermark so each run processes only the data that changed.

C. Stop loading the table altogether, because a table that is expensive to reload is not worth keeping current in the warehouse.

D. Load the full table into more environments, because spreading the same full reload across environments makes each one cheaper.

Correct answer: B
Reloading an entire large table every night is slow and costly; incremental loading with CDC or a watermark processes only changed data, dramatically cutting time and cost. Running full reloads more often or in more places increases cost, and abandoning the table is not a real solution.

Question 9 Domain 4: Batch and Streaming Processing

A team uses an always-on streaming pipeline to produce a report that is only viewed once a day. What is the concern?

A. There is no concern, because streaming is always the superior choice and should be used for every reporting pipeline regardless of need.

B. Streaming adds cost and operational complexity with no benefit here; a scheduled batch job would meet the once-a-day need more simply and cheaply.

C. The concern is that streaming is too slow for a daily report, so the pipeline should be rewritten to run even more continuously than it does now.

D. The only concern is the report format, so converting the streaming output into a different chart type resolves the mismatch entirely.

Correct answer: B
A report viewed once a day has no real-time requirement, so an always-on streaming pipeline adds cost and complexity for no benefit; a scheduled batch job is simpler and cheaper. Streaming is not universally superior, it is not too slow for a daily report, and the issue is the processing model, not the chart type.

Question 10 Domain 5: Data Quality, Orchestration and Reliability

An upstream schema change unexpectedly breaks five downstream jobs. What practices would reduce this risk?

A. Forbid all schema changes forever, because the only way to protect downstream jobs is to freeze every upstream dataset permanently.

B. Data contracts plus lineage-based impact analysis, so schema changes are coordinated and affected consumers are identified and warned in advance.

C. Ignore the downstream jobs, because it is the responsibility of each downstream team to constantly watch the upstream schema for any changes.

D. Delete the downstream jobs whenever the upstream schema changes, because rebuilding them from scratch each time is simpler than coordinating.

Correct answer: B
Data contracts set agreed expectations between producers and consumers, and lineage-based impact analysis identifies which downstream jobs a schema change affects, so changes are coordinated and consumers warned. Freezing all schemas is impractical, ignoring downstream invites breakage, and deleting jobs on every change is not viable.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CDEP-001. https://certlabz.com/certifications/certified-data-engineering-professional.html