DUMPSBOSS.

Adobe Analytics Developer Professional Exam

Adobe AD0-E213

Version Demo

Total Demo Questions: 7

Total Premium Questions: 79

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Technical Implementation	61
Topic 2, Reporting and Visualization	8
Topic 3, Administration	10
Total	79

QUESTION NO: 1

Identify the correct features of Rollup Reports from below options, kindly select 2 options which imply the same? Select all that apply.

A. Rollup report suites can include a maximum of 40 child report suites.

B. When you add a report suite to an existing rollup, historical data is included in the rollup.

C. Data contained in rollup report suites does not support breakdowns or segments.

D. Rollup report suites do not have any limit for events.

E. Rollup report suites combine data from multiple child report suites into one reporting view.

ANSWER: A E

Explanation:

“Rollup report suites can include a maximum of 40 child report suites” is correct because Adobe Analytics defines a rollup report suite as a reporting suite that aggregates data from a limited set of source (child) report suites, with a maximum of 40 child suites. This limit is an important implementation constraint when designing consolidated reporting across sites, brands, regions, or other digital properties.

“Rollup report suites combine data from multiple child report suites into one reporting view” is also correct. The central purpose of a rollup is to provide a unified view of data collected in its included child report suites, allowing analysts to report across those suites without having to manually combine separate reports. The child suites remain independently reportable, while the rollup provides the aggregate reporting layer. Adobe’s report-suite administration documentation describes rollups as report suites that aggregate data from multiple source report suites and documents their configuration and limits. See Adobe’s Rollup report suites documentation and the Report suite administration documentation for implementation details.

QUESTION NO: 2

Which of the following scenarios would justify the use of the feature "processing rules" in adobe analytics? Select all that apply.

A. Product manager has launched a new iOS/Android app and data is being captured in context variable. It needs to be mapped over to eVar/props

B. A product manager is looking to capture user agent in an eVar but doesn't want to request the release manager for another build in production

C. Product Marketing team has launched a new social media marketing campaign and would like to see campaign performance in last touch channel report

D. The page name for checkout is "checkout". Product manager would like to fix the historical values that have been captured so far in pages report

ANSWER: A B

Explanation:

Processing rules are appropriate when Adobe Analytics can transform or populate reporting variables at collection time without requiring a change to the site or app implementation. “Product manager has launched a new iOS/Android app and data is being captured in context variable. It needs to be mapped over to eVar/props” is a core processing-rules use case: context-data variables sent by an SDK can be mapped to report-suite variables such as eVars and props as the hit is processed. This allows the existing mobile implementation to continue sending context data while the report suite defines how that data is stored for reporting.

“A product manager is looking to capture user agent in an eVar but doesn't want to request the release manager for another build in production” is also justified because processing rules can use values available on incoming Analytics hits, including user-agent information, to populate an eVar at processing time. This enables reporting changes without an application release, provided the rule is configured carefully and the eVar’s expiration and allocation settings support the intended

analysis. Processing rules affect data received after the rule is enabled; they are therefore suited to forward-looking collection and variable-population changes. Adobe documents processing rules and their variable-setting capabilities at Processing rules overview and context-data mapping at Context data variables.

QUESTION NO: 3

Within Adobe Analytics, how long does it take for data to be processed and ready for reporting?

A. Approximately 5 minutes

B. Within seconds

C. Approximately 1 hour

D. Typically 30–90 minutes

ANSWER: D

Explanation:

Typically 30–90 minutes is the appropriate answer for standard Adobe Analytics reporting availability. After a hit is collected, Adobe Analytics must ingest it, apply processing rules and other report-suite processing, persist the data, and make the processed results available to reporting interfaces such as Analysis Workspace. This is asynchronous processing rather than an immediate, hit-by-hit reporting update, so implementations and analysts should allow for reporting latency when validating a deployment or assessing very recent activity.

Adobe documents that Analytics reporting data is generally available within a 30-to-90-minute window. The exact time can vary with processing conditions and with the report or capability being used. Some specialized workflows, such as data feeds, may follow their own delivery schedules, while real-time reporting is a separate feature and should not be treated as the normal availability expectation for standard reports. For practical QA, collect test hits, confirm them in the Experience Cloud Debugger or network request, then allow the normal processing window before expecting them in Analysis Workspace. See Adobe's data latency documentation and Analysis Workspace documentation.

QUESTION NO: 4

What is the first step for data to get from the developer's page into the reports in Adobe Analytics?

A. Analytics code is served via the data layer and Adobe Launch.

B. The website invokes the image request.

C. Visitor arrives at the digital experience.

ANSWER: C

Explanation:

Visitor arrives at the digital experience. is the first event in the data-collection sequence. Adobe Analytics cannot collect page-view or interaction data until a person opens the site, application, or other tracked digital property. That arrival causes the relevant page or experience to load in the visitor's browser. During loading and subsequent interaction, the implemented Adobe Analytics library or Adobe Experience Platform Web SDK can run, read the configured page and visitor context, and send a tracking request to Adobe's collection servers. Adobe then processes the received data for the configured report suite or data stream, making it available through Analytics reporting after processing. In other words, the visitor's arrival is the initiating business event that makes all later technical steps possible: page code execution, request generation, data ingestion, processing, and reporting. Adobe's implementation documentation describes how Analytics implementations collect data from digital properties and send it to Adobe, while its data-collection overview explains the role of client-side tags and SDKs in capturing data when users engage with an experience. See Adobe Analytics implementation overview and Adobe Experience Platform Tags documentation.

QUESTION NO: 5

In Adobe Launch, what should be the type of data element to refer a data layer variable?

A. Data Layer

B. Javascript Variable

C. DOM Attribute

ANSWER: B

Explanation:

Javascript Variable is the appropriate data-element type for referring to a variable exposed by a site's data layer in Adobe Launch (now Adobe Experience Platform Tags). A data layer commonly makes values available as properties of a global JavaScript object, such as `window.digitalData` or another site-defined object. A JavaScript Variable data element lets the implementation specify that object/property path and resolve its current value when a rule uses the data element. For example, a value held at `digitalData.page.pageInfo.pageName` can be made available to rule conditions, Adobe Analytics variables, and other extension actions through a reusable data element.

This approach keeps the mapping between the site's data layer and tag configuration centralized. Rules can reference the data element rather than repeating the JavaScript property path, making the implementation easier to maintain when the same value is needed in multiple places. Adobe documents JavaScript Variable as a Core extension data-element type and describes data elements as reusable variables that capture values from page content, cookies, local storage, query parameters, and other sources. See Adobe's Core extension overview and its data elements documentation.

QUESTION NO: 6

Which two statements correctly describe Adobe Analytics page-view and custom-link tracking calls? Select all that apply.

A. A page-view call increments the Page views metric.

B. A custom-link call can be sent using the s.tl() method.

C. A custom-link call automatically increments the Page views metric.

D. A page-view call can only contain the pageName variable.

E. A custom-link call cannot contain success events.

ANSWER: A B

Explanation:

A page-view call increments the Page views metric is correct. In an AppMeasurement implementation, a page-view tracking call is sent using `s.t()`. When the call is processed, it contributes to Page views and can populate the Page dimension using the `pageName` value or the page URL fallback.

A custom-link call can be sent using the s.tl() method is also correct. The `s.tl()` method sends a link-tracking call for interactions such as downloads, exit links, and custom links. Unlike a normal page-view call, it does not increment Page views. Developers can set events, eVars, props, and other applicable variables before sending the custom-link call. See Adobe's documentation for the t() method and the tl() method.

QUESTION NO: 7

Identify the function from below which is not performed by ID Service cookies?

A. Access or store personally identifiable information (PII) like your email address.

B. Persist this unique ID so the ID service can collect and share data with other Experience Cloud solutions.

C. Set and store a unique ID for your site visitors (the MID).

ANSWER: A

Explanation:

Access or store personally identifiable information (PII) like your email address. is the function not performed by Experience Cloud ID Service cookies. The Experience Cloud Identity Service creates and maintains a persistent Experience Cloud ID (ECID, formerly called MID) in a first-party `AMCV` cookie. This identifier lets Adobe Experience Cloud applications recognize the same browser or device over time and use the common identity to support data sharing and integration across Experience Cloud solutions. The cookie's purpose is identification through an opaque Adobe-generated visitor ID, rather than retaining directly identifying details such as an email address. Adobe's documentation specifically describes the AMCV cookie as containing the Experience Cloud ID and related region information; it does not define the cookie as storage for customer PII. Implementations should therefore avoid placing PII in identity values, URLs, variables, or other Analytics collection fields unless the organization has an appropriate, documented, privacy-compliant process. This distinction matters because the ECID is a pseudonymous technical identifier used for visitor recognition, while an email address directly identifies an individual. See Adobe's Experience Cloud Identity Service overview and its cookie documentation.