Adobe

AD0-E605 Exam

Adobe Real-Time Customer Data Profile Developer Expert

Questions & Answers
Demo

Version: 4.0

Question: 1

A data architect is designing a Real-Time Customer Profile to capture user interactions across multiple channels for an online media company. The company tracks user interactions such as article reads, video views, and ad clicks across its website, app, and email newsletters. Currently, the Real-Time Customer Profile schema design contains a User Profile Class and an Experience Event Class. The Experience Event Class captures each interaction as a separate event record and contains an identity field (user_id) linking to the user profile.

Upon review, the data architect realizes that the schema design is unable to accurately capture the sequence of interactions made by a single user during one session (defined as a continuous period of activity without more than 30 minutes of inactivity).

How should the data architect modify the schema design to better capture the sequence of user interactions within a single session in the Real-Time Customer Profile?

A. Add a session.id field to the Experience Event Schema

B. Add a session.id field to the User Profile Schema

C. Add an event.type field to the User Profile Schema

D. Create a new Session Schema and add a session.id field to it

Answer: A

Explanation:

In Adobe Real-Time CDP, the Experience Data Model (XDM) is designed to separate static attributes from time-series data. The XDM ExperienceEvent Class is specifically intended to capture "point-in-time" occurrences, such as clicks, views, or purchases. To accurately track and sequence user interactions within a specific session, the most effective architectural approach is to include a session identifier directly within the ExperienceEvent schema.

By adding a session.id (typically via the Adobe Analytics or Web SDK mixin) to the ExperienceEvent record, each discrete event is tagged with a unique identifier that persists for the duration of the user's activity. This allows the Real-Time Customer Profile to not only link events to a specific individual via the Identity Map but also to group and sequence those events chronologically within a specific visit.

Options B and C are incorrect because the XDM Individual Profile Class represents the "state" of a user (e.g., name, email, subscription status) rather than a sequence of transient actions; storing a session ID there would result in data overwriting and a loss of historical session context. Option D is unnecessary because XDM is built on a flat, denormalized event structure; creating a separate schema for sessions would introduce unnecessary complexity in relationship mapping and decrease performance for real-time segmentation. Therefore, modifying the ExperienceEvent schema to include a session identifier is the standard best practice for session-based behavioral analysis and journey orchestration.

Question: 2

A media company uses Adobe Experience Platform to process large quantities of media consumption data. This data was previously stored in a relational database management system (RDBMS) but has been migrated to the Adobe Real-Time CDP's NoSQL data model for improved scalability and performance. The data set includes information such as user ID, media content ID, play duration, pause durations, and timestamps of each interaction. Which combination of Experience Data Model (XDM) schemas should be used to efficiently capture and retrieve this data with the Adobe Real-Time CDP's NoSQL data model, considering the real-time analytics needs?

A. An XDM Individual Profile schema for user ID, media content ID and an XDM Experience Event schema for play duration, pause durations, and timestamps

B. An XDM Experience Event schema for user ID, media content ID and an XDM Individual Profile schema for play duration, pause durations, and timestamps

C. Two XDM schemas, one for user ID and media content ID, the other for play duration, pause durations, and timestamps

D. An XDM Individual Profile schema for user ID and an XDM Experience Event schema for media consumption data

Answer: D

Explanation:

In the Adobe Real-Time Customer Data Platform, the architecture is built upon the Experience Data Model (XDM), which utilizes two primary base classes to represent different types of data: XDM Individual Profile and XDM ExperienceEvent. To efficiently handle media consumption data in a NoSQL environment, it is critical to distinguish between the "actor" and the "action."

The XDM Individual Profile class is designed to store the "record" data, representing the identity and attributes of the user (the actor). This includes the user ID and other persistent traits like name, email, or preferences. This schema provides the centralized view of the customer. Conversely, the XDM ExperienceEvent class is purpose-built for time-series data (the actions). Media consumption data—such as content IDs, play/pause durations, and timestamps—are inherently behavioral and occur at specific points in time.

By using an ExperienceEvent schema for the media interactions, the system can capture an unlimited stream of events without bloating the profile record itself. Each event is linked to the Individual Profile via a common identity (the user ID). This separation is vital for the NoSQL data model used by the Real-Time Customer Profile, as it allows for high-throughput ingestion and enables real-time segmentation based on recent behaviors (e.g., "users who watched more than 50% of a video in the last 24 hours"). Options A, B, and C incorrectly mix behavioral event data into the profile schema or fail to leverage the standard class structure required for profile enrichment.

Question: 3

A company collects customer data from various sources, including customer relationship management (CRM) systems, website interactions, and in-store purchases. To create a unified Real-Time Customer Profile in Adobe Experience Platform, which approach should be taken?

A. Use a single identity namespace for all data sources to simplify identity resolution

B. Configure identity namespaces based on Experience Data Model (XDM) schemas for all profile-enabled data sources

C. Ingest data without mapping to XDM schemas to maintain raw data integrity

D. Prioritize online data over offline data to ensure real-time responsiveness

Answer: B

Explanation:

To create a unified Real-Time Customer Profile, Adobe Experience Platform relies on the Identity Service to stitch data together across disparate sources. The correct approach involves configuring Identity Namespaces within the XDM schemas used by each data source. An Identity Namespace provides context for an identity value (e.g., distinguishing an "Email" from a "CRM ID").

When multiple datasets are "Profile-enabled," the system looks for common identity markers. For example, a CRM record might contain a CRM_ID and an Email, while a website interaction contains an ECID and an Email. By defining these fields as identities within their respective schemas and assigning them to the correct namespaces, the Identity Graph can link the CRM_ID to the ECID via the shared Email.

Option A is incorrect because using a single namespace for different ID types would lead to data collisions and failed resolution. Option C is incorrect because data must be mapped to XDM and enabled for Profile to be included in the unified view; raw, unmapped data remains in the data lake and cannot participate in real-time services. Option D is a strategic choice but not a technical requirement for unification; the platform is designed to handle both high-velocity behavioral data and high-volume batch data (offline) simultaneously to form a complete 360-degree view.

Question: 4

A multinational company is transitioning its on-premises data warehouse to the Adobe Experience Platform in an attempt to reap the benefits of real-time data and cloud scalability. The current data warehouse includes a complex set of relational databases with numerous tables including Orders, OrderDetails, Customers, Products, and Suppliers. The Orders and OrderDetails are interconnected with a one-to-many relationship, while the rest of the tables have many-to-many relationships.

Which two approaches should be followed while translating this release database management system (RDBMS) schema to Adobe Real-Time Customer Data Platform's (Adobe Real-Time CDP) NoSQL data model, considering the maintenance of data relationships? (Choose two.)

A. Develop a custom schema for each entity, linking them together using relationship identifiers within nested fields of the schema

B. Create only two schemas: one for Customers and another for all related entities

C. Create separate experience data model schemas for each entity and use nested fields to store related data

D. Create separate schemas for each entity and build foreign key relationships similar to the SQL model

E. Create separate experience data model schemas for each entity, use lookup fields for one-to-many relationships, and use arrays to maintain many-to-many relationships

Answer: C, E

Explanation:

Translating a relational (RDBMS) model to the Adobe Experience Platform's NoSQL-based XDM requires a shift from "joined tables" to "hierarchical and linked objects."

Approach C is essential for handling 1:N relationships where the data is highly coupled, such as

Orders and OrderDetails. In XDM, rather than having a separate table for line items, you should use nested fields (or an array of objects) within the Order schema. This ensures that when an "Order" event is retrieved, all its details are available in a single document, maximizing performance for real-time segmentation and activation without needing complex joins.

Approach E addresses the broader relational structure. For entities like Products or Suppliers, Adobe utilizes Lookup Schemas. By defining a relationship between an ExperienceEvent (the Order) and a Lookup Schema (the Product), the Real-Time Customer Profile can "hydrate" event data with descriptive attributes from the lookup table at the time of processing. Furthermore, for many-to-many (N:N) relationships, XDM utilizes arrays of strings or objects to store multiple identifiers within a single profile or event record. This denormalized approach is fundamental to NoSQL scalability, as it allows the platform to maintain data integrity and relationship context while providing the sub-second query speeds required for real-time use cases.

Question: 5

A financial institution is migrating its customer transaction data from a relational database (RDBMS) to Adobe Real-Time CDP. The institution's transaction records include data points like customer ID, account type, transaction type, transaction amount, and transaction date. The data architect must ensure the transaction data can be linked to individual customer profiles in Adobe Real-Time CDP while also ensuring the data model maintains performance for real-time analysis and personalization use cases. What is the best approach to model this data in Adobe Real-Time CDP's NoSQL data model?

A. Create an XDM Experience Event schema for each transaction type

B. Create a custom entity schema for each transaction type

C. Create an Experience Data Model (XDM) Individual Profile schema for each customer and link transactions via relationships

D. Create an XDM Experience Event schema for transactions and link it to the individual customer profile via the customer ID

Answer: D

Explanation:

In Adobe Real-Time CDP, transaction data is inherently behavioral and time-bound. The XDM ExperienceEvent class is the optimized choice for this data type because it is designed to capture immutable, point-in-time actions. Each transaction (containing amount, type, and date) should be treated as an event. By including the customer ID within this schema and marking it as an Identity, the platform’s Identity Service automatically associates these events with the corresponding XDM Individual Profile.

This approach is superior to Option C because the Individual Profile schema is intended for stateful attributes (like "current balance" or "account level"), not a growing list of transactions. Storing transactions in the profile would lead to extremely large profile fragments, degrading performance. Option A is inefficient as it creates schema sprawl; instead, a single ExperienceEvent schema should use a "transaction type" field to differentiate between deposits, withdrawals, or transfers.

By leveraging the NoSQL architecture of the Real-Time Customer Profile, these events are stored in a way that allows the Segmentation Service to evaluate them in milliseconds. For example, a segment could instantly identify "customers who made a transaction over $1,000 in the last hour." Linking via the customer ID ensures that as soon as a transaction is ingested, it is immediately visible on the unified profile for real-time personalization.