DUMPSBOSS.

Salesforce Certified Marketing Cloud Developer

Salesforce Marketing-Cloud-Developer

Version Demo

Total Demo Questions: 33

Total Premium Questions: 337

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Modeling	68
Topic 2, Security	20
Topic 3, Programmatic Languages	125
Topic 4, API	77
Topic 5, Marketing Cloud Platform	47
Total	337

QUESTION NO: 1

Which aspect should a developer consider before creating an Server-to-Server Installed Package and associated API Integration i* Marketing Cloud?

A. Using an Installed Package, APIs will have access to resources in all Business Units.

B. Scope (Permissions) will be granted based on the User who is creating the Installed Package.

C. Scope (Permissions) must be specifically granted when creating an API Integration component inside an Installed Package.

ANSWER: C

Explanation:

Scope (Permissions) must be specifically granted when creating an API Integration component inside an Installed Package. A server-to-server integration uses the OAuth 2.0 client-credentials flow: the external application authenticates using the client ID and client secret generated for the API integration, rather than acting as an individual Marketing Cloud user. Consequently, its access is controlled by the scopes selected in the Installed Package's API Integration component. The developer should identify the precise REST and SOAP API capabilities required—such as reading or writing specific data, managing assets, or interacting with journeys—and grant only those permissions. This is an important least-privilege decision because the integration can perform operations allowed by its assigned scopes whenever it obtains an access token. Scopes can be adjusted later by editing the API Integration component, but changes should be deliberate and followed by appropriate validation of the integration. Salesforce's Installed Package workflow explicitly includes selecting API permissions when configuring a server-to-server API integration. See Salesforce's integration considerations and the server-to-server integration documentation for the authentication model and permission-scoping process.

QUESTION NO: 2

In what order is AMPscript evaluated before an email is sent?

A. Subject Line, HTML Body, Text Body

B. HTML Body, Text Body, Subject Line

C. Text Body, HTML Body, Subject Line

D. HTML Body, Text Body, Text Body

ANSWER: B

Explanation:

AMPscript is evaluated in the order **HTML Body, Text Body, Subject Line**. Marketing Cloud processes the content areas of an email in a defined sequence before delivery. It first evaluates AMPscript in the HTML version of the message, then evaluates the plain-text version, and finally evaluates the subject line. This processing order matters whenever AMPscript performs actions that can affect later content, such as setting variables, writing to data extensions, or using conditional logic that depends on values established during processing. Developers should therefore design email AMPscript with the understanding that the subject line is processed last, rather than assuming it is evaluated before the message bodies. Keeping variable initialization and any required subscriber or send-context logic appropriately available to each content area helps ensure consistent personalization across the rendered email and its subject line. Salesforce documents AMPscript as a server-side scripting language for dynamically generating and personalizing Marketing Cloud content, and its email-processing guidance identifies the order in which those email components are evaluated. See Salesforce's AMPscript processing order documentation and the AMPscript Developer Guide.

QUESTION NO: 3

Northern Trail Outfitters' legal team is concerned about the daily import process that brings in subscribers to a Sendable Data Extension, even when records have already been targeted for deletion.

Which two true expected behaviors for these recordsoccur in the event a send is initiated directly to this Sendable Data Extension?

Choose 2 answers

A. Records still in the suppression phase will only be excluded if manually specified during send time.

B. Records still in the suppression phase will beexcluded from sends.

C. Records that have already been deleted will be treated as new records.

D. Records that have been deleted will be excluded from sends Indefinitely.

ANSWER: B C

Explanation:

Records still in the suppression phase will be excluded from sends. When Contact Delete is initiated, Marketing Cloud Engagement places the contact into a suppression period before permanently deleting the contact. This safeguards the deletion process by preventing the contact from receiving messages while the request is being processed, including when a send is initiated directly from a sendable data extension that still contains the record.

Records that have already been deleted will be treated as new records. Contact deletion removes the contact from Marketing Cloud Engagement's contact model; it does not necessarily remove matching rows from every data extension. If a deleted contact's identifying value remains in, or is later reimported to, a sendable data extension and that data extension is used as the send audience, Marketing Cloud Engagement can create the contact again as part of send processing. The newly created contact is not indefinitely suppressed merely because a prior instance of that contact was deleted. To avoid recreating contacts, the import and audience-selection processes must exclude identifiers that should remain absent from Marketing Cloud Engagement.

Salesforce documents the contact deletion lifecycle and its suppression period in Delete Contacts in Marketing Cloud Engagement and provides API-level contact-deletion behavior in the Contact Delete documentation.

QUESTION NO: 4

A developer wants to configure performance tracking of the content dynamically created via AMPscript in an email. Which two steps should be performed to achieve this objective? Choose 2

A. Request that the Impression Tracking feature be enabled on the account

B. Include the functions BeginImpressionRegion and EndImpressionRegion

C. Configure dynamic content block in Content Builder

D. Add a unique identifier in the HTML tags within the generated content

ANSWER: A B

Explanation:

Request that the Impression Tracking feature be enabled on the account because impression-region tracking is an account-level Marketing Cloud capability that must be provisioned before a developer can collect impression data for specific areas of an email. Once enabled, it allows reporting to distinguish engagement with an identified content region rather than treating the email body as a single undifferentiated unit.

Include the functions BeginImpressionRegion and EndImpressionRegion because these AMPscript functions define the exact start and end boundaries of the dynamically generated content to measure. The developer should place `BeginImpressionRegion` before the AMPscript output that renders the dynamic content and `EndImpressionRegion` immediately after it. This wrapping lets Marketing Cloud associate an impression region name with the rendered content during send and tracking processing. The region name supplied to `BeginImpressionRegion` should be meaningful and consistently used so that the resulting performance data can be identified in reporting. Salesforce documents these functions as the mechanism for creating and closing measurable impression regions in email content. See the Salesforce AMPscript references for BeginImpressionRegion and EndImpressionRegion.

QUESTION NO: 5

Certification Aid wants to create a file drop automation with a filename pattern. An import file is placed daily on the Marketing Cloud Enhanced FTP server, and thefile name always starts with the current month and day (e.g. OCT26). How should the filename pattern be defined? Choose 2.

A. %%Month%%%%Day%%

B. %%MMDD%%

C. Ends With operator

D. Begins With operator

ANSWER: A D

Explanation:

`%%Month%%%%Day%%` is the appropriate dynamic filename pattern because a File Drop automation can use Marketing Cloud date placeholders to evaluate the date when the automation runs. The `%%Month%%` placeholder supplies the current month in the format needed for a prefix such as `OCT`, and `%%Day%%` supplies the day portion, producing a value such as `OCT26`. This lets the automation locate the daily file without requiring someone to manually edit the pattern each day. The `Begins With operator` is required because the date value is only the leading portion of the filename; the imported file can include additional characters after that prefix, such as an extension or a descriptive suffix. Together, the pattern and operator instruct Automation Studio to identify files whose names begin with the date-derived value for the current run date. Salesforce documents File Drop activities as an Automation Studio mechanism for detecting files on Enhanced FTP and starting downstream processing when the configured file criteria are met. See Salesforce's File Drop Automation documentation and the Trailhead module on automating data imports.

QUESTION NO: 6

A developer is leveraging the SOAP API to dynamically display Profile and Preference Attributes in a custom profile center. Which method could be used to support the dynamic functionality?

A. Describe

B. Extract

C. Perform

D. Configure

ANSWER: A

Explanation:

Describe is the appropriate SOAP API method because it returns metadata about objects and their properties in Marketing Cloud Engagement. For a custom profile center that must dynamically render Profile and Preference Attributes, the developer needs to discover the available attributes, including their names, data types, and other property definitions, rather than hard-coding a fixed set of fields. A Describe request can be used to inspect the relevant API object metadata and build the profile-center interface based on the attributes configured in the account. This supports a more maintainable implementation: when profile or preference attributes are added or modified, the application can retrieve the updated metadata and adjust what it displays. The SOAP API's Describe operation is specifically intended to obtain information about an object's available properties and their definitions, making it suitable for metadata-driven user interfaces. Salesforce documents the SOAP API operations, including Describe, in the SOAP Web Service Methods reference. The details of object metadata returned by Describe are also covered in the Describe method documentation.

QUESTION NO: 7

Where can the SSJS Core library be used? Choose 2.

A. SMS messages

B. Marketing Cloud apps

C. Landing pages

D. Email messages

ANSWER: B C

Explanation:

The SSJS Core library can be used in **Marketing Cloud apps** and **Landing pages**. The Core library supplies higher-level server-side JavaScript objects and functions for common Marketing Cloud tasks, including working with data extensions, rows, dates, HTTP requests, and utility functions. In these server-side execution contexts, a developer loads the library with `Platform.Load("Core", "1")` before calling its Core functions. Landing pages are a standard web execution context for SSJS, while Marketing Cloud apps can also execute server-side code and use the Core library to support application logic and interactions with Marketing Cloud data. Salesforce distinguishes the Core library's supported contexts from other SSJS libraries and execution environments, so developers should select the library and scripting approach that matches where the code runs. Review Salesforce's Server-Side JavaScript documentation and the Core Library reference for usage details and available functions.

QUESTION NO: 8

Certification Aid wants to implement a custom profile center using SOAP API. Which SOAP API methods are relevant to achieve this? Choose 2.

A. Extract

B. Describe

C. Update

D. Configure

ANSWER: B C

Explanation:

Describe and **Update** are relevant to implementing a custom profile center with the Marketing Cloud SOAP API. A custom profile center needs to understand the available subscriber and profile-related fields before it can present an appropriate interface. The SOAP `Describe` operation returns metadata about an API object, including its properties and supported operations. This enables the application to identify the fields that can be surfaced and maintained for subscriber profile management.

After a subscriber submits profile changes, the application must persist those changes in Marketing Cloud. The SOAP `Update` operation updates existing objects, including Subscriber records and associated profile attributes when the request supplies the subscriber key and the applicable attributes. This supports the core profile-center workflow of changing contact information, preferences, or other subscriber data in response to a user's submission. Salesforce documents the object-metadata purpose of `Describe` and the process for modifying existing API objects with `Update` in its SOAP API documentation. See Describe and Update.

QUESTION NO: 9

A developer is designing a recurring Import Activity that loads a file into a Data Extension. The source file can contain both new records and changed records. Which two configurations should be used to add new records and update existing records? Choose 2.

A. Select Add and Update as the import type

B. Define a primary key on the target Data Extension

C. Select Overwrite as the import type

D. Remove all primary keys from the target Data Extension

ANSWER: A B

Explanation:

Configure the import type as **Add and Update** and define a primary key on the target Data Extension. The Add and Update import type allows Marketing Cloud to create records when no matching target row exists and update records when a matching row is found. Marketing Cloud uses the target Data Extension primary key to determine whether an incoming record already exists.

Without a primary key, Marketing Cloud cannot reliably identify the target row that should be updated. An Overwrite import replaces the target data rather than performing an incremental insert-or-update operation, while an Add Only import does not update existing records. Salesforce documents import types and primary-key behavior in the Import Activity documentation.

QUESTION NO: 10

A developer initiated a batch delete of Contacts in Contact Builder due to an import error during implementation. There are over two million records that need to be deleted.

Which two factors should be considered when batch deleting large volumes of contacts?Choose 2 answers

A. Up to one million records can be deleted in each batch.

B. To more quickly remove contact information, use the suppression period length of 14.

C. The deletion process supersedes other automated account activities.

D. The suppression status does not show for individual contacts until the entire batch processes.

ANSWER: C D

Explanation:

"The deletion process supersedes other automated account activities." is correct because a Contact Builder contact deletion is a resource-intensive, account-level operation. Salesforce processes the deletion as a priority operation, so the team must plan for its impact on concurrent Marketing Cloud activity, including automations and other processing that could be delayed while the large deletion is underway. This is especially important for a deletion involving more than two million contacts, where the operation can take substantial time.

"The suppression status does not show for individual contacts until the entire batch processes." is also correct. During a batch deletion, Marketing Cloud applies suppression as part of the deletion workflow to prevent the affected contacts from being used while removal is pending. The status is applied and reflected at the batch level rather than appearing incrementally as each individual contact is handled. Teams should therefore allow the batch to complete before using contact-level status checks to validate the outcome, and should monitor the deletion job rather than assuming that a contact is fully processed based on an intermediate state.

These behaviors support scheduling the operation carefully and validating completion after each deletion batch. See Salesforce's Delete Contacts documentation and Contact Builder documentation.

QUESTION NO: 11

A developer is writing a query to select unique subscribers who opened any emails sent since the beginning of the previous day.

Which querywould provide that result?

A)

B)

C)

A. Option A

B. Option B

C. Option C

D. SELECT DISTINCT SubscriberKey FROM _Open WHERE EventDate >= DATEADD(day, DATEDIFF(day, 0, GETDATE()) - 1, 0);

ANSWER: D

Explanation:

The query `SELECT DISTINCT SubscriberKey FROM _Open WHERE EventDate >= DATEADD(day, DATEDIFF(day, 0, GETDATE()) - 1, 0);` correctly returns each qualifying subscriber only once. The `_Open` data view records email open tracking events and includes both `SubscriberKey`, which identifies the subscriber, and `EventDate`, which records when the open occurred. Applying `DISTINCT` prevents a subscriber who opened multiple messages, or opened a message multiple times, from appearing more than once in the result set.

The date expression first uses `DATEDIFF` to determine the current day boundary, subtracts one day, and then uses `DATEADD` to produce midnight at the beginning of the previous day. This is preferable to a rolling 24-hour calculation because the requirement is based on a calendar-day boundary. The comparison includes all open events at or after that timestamp. Salesforce documents the available fields in the `_Open` data view at Open Data View and documents supported date functions, including `GETDATE`, `DATEDIFF`, and `DATEADD`, at Marketing Cloud T-SQL Date Functions.

QUESTION NO: 12

A developer wants to use the RaiseError Ampscript function when a subscriber does not have the necessary data to build an email. Which two outcomes are possible using this function? Choose 2 answer

A. The send fails

B. The email is not sent to the particular subscriber

C. An error message is sent to the From Address used in the email

D. The send is retried

ANSWER: A B

Explanation:

`RaiseError()` is intended to deliberately stop AMPscript processing when a condition makes the message invalid or unsafe to send, such as when required personalization data is absent. The outcome described by **"The send fails"** is possible because the function raises a processing error rather than allowing the email to continue rendering normally. Marketing Cloud records the error as part of send processing, providing a controlled way to prevent a message with incomplete required content from being sent.

"The email is not sent to the particular subscriber" is also possible because `RaiseError()` can terminate processing for the individual subscriber whose email execution encounters the error. This makes the function useful for data-validation logic embedded in an email: the developer can check for essential attributes, call `RaiseError()` when they are missing, and prevent delivery to that affected subscriber. The function also accepts a parameter that controls whether data-extension row changes made during execution are preserved, allowing developers to manage related data behavior while handling the send error. Salesforce documents the function syntax, send-processing behavior, and optional parameters in the RaiseError AMPscript reference. See also the AMPscript Developer Guide for AMPscript execution context.

QUESTION NO: 13

Certification Aid wants to import an encrypted CSV file from the Marketing Cloud Enhanced FTP server. Which two File Transfer activities are needed to achieve this? Choose 2.

A. To decrypt the import file on the Enhanced FTP server.

B. To move the import file from the Safehouse to Marketing Cloud.

C. To decrypt the import file on the Safehouse.

D. To move the import file from the Enhanced FTP server to the Safehouse.

ANSWER: C D

Explanation:

To make an encrypted CSV available for a Marketing Cloud import, the file must first be moved from the Enhanced FTP location into the Marketing Cloud Safehouse. The Safehouse is the secure temporary file location used by Automation Studio File Transfer activities for operations such as encryption, decryption, and unzipping. Therefore, moving the import file from the Enhanced FTP server to the Safehouse is the required first activity.

Once the encrypted file is in Safehouse, the next required activity is to decrypt the import file on the Safehouse. This produces a usable decrypted CSV that can then be selected by a subsequent Import Activity. The decryption operation is performed in Safehouse rather than directly on Enhanced FTP, so the correct sequence is transfer to Safehouse followed by Safehouse decryption. Salesforce describes the Safehouse as the staging location for File Transfer operations and documents decryption as a supported File Transfer action. See the Salesforce File Transfer Activity documentation and Marketing Cloud Enhanced FTP documentation.

QUESTION NO: 14

From which business unit could the Contact Delete feature be used within an Enterprise 2.0 account?

A. Any business unit

B. The Parent account

C. Only in Agency accounts

D. The business unit where the contact was introduced

ANSWER: B

Explanation:

The Parent account is correct. In an Enterprise 2.0 Marketing Cloud account, contact data and relationships can span the enterprise and its child business units. Because a contact deletion is intended to remove the contact comprehensively, the Contact Delete feature is initiated from the parent account rather than from an individual child business unit. This enterprise-level control helps ensure that the deletion request is evaluated and processed consistently for the contact across the account structure, including applicable contact records and related data managed at the enterprise level.

Contact deletion is a deliberate privacy and data-governance operation, not simply the removal of a subscriber from a single sendable data extension or channel list. The parent account is therefore the appropriate administrative context for initiating the process in an Enterprise 2.0 configuration. Before using the feature, administrators should understand the retention implications and ensure that the account is configured for Contact Delete according to Salesforce guidance. Salesforce documents the enterprise behavior and process in its Contact Delete documentation and provides additional operational guidance in the Marketing Cloud Contact Delete help article.

QUESTION NO: 15

A developer needs to generate a file containing email tracking data and place the resulting file on the Marketing Cloud Enhanced FTP server. Which two Automation Studio activities should be used? Choose 2.

A. Tracking Extract Activity

B. File Transfer Activity

C. Import Activity

D. SQL Query Activity

ANSWER: A B

Explanation:

A **Tracking Extract Activity** creates a file containing email tracking information, such as sends, opens, clicks, and bounces. The extract file is created in the Marketing Cloud Safehouse. A **File Transfer Activity** can then move the generated file from Safehouse to the Enhanced FTP location.

A SQL Query Activity writes query results to a data extension, not directly to an FTP file. An Import Activity loads file data into Marketing Cloud and is not used to deliver an extract file. Salesforce documents the Tracking Extract Activity and the File Transfer Activity.

QUESTION NO: 16

Certification Aid wants to add new customers to a cross-channel welcome campaign when they register on the company website. Which API should be used for this? Choose 1.

A. Personalization Builder API

B. Event Notification API

C. Transactional Messaging API

D. Journey Builder API

ANSWER: D

Explanation:

The **Journey Builder API** is the appropriate choice because a website registration is an external, real-time event that should enroll a new contact into a defined customer journey. A developer can configure an API Event entry source in Journey Builder and use the journey's event definition key to post the registrant's contact and event data to Marketing Cloud. Journey Builder then evaluates the entry event and starts the contact on the cross-channel welcome flow, where the journey can coordinate email, SMS, push notifications, advertising, or other configured activities.

This approach separates the website's registration action from the campaign orchestration itself: the website submits an event, and Journey Builder controls the timing, path, personalization, and channels used for the welcome experience. The event payload can also include registration attributes needed for segmentation or personalization later in the journey. Salesforce documents event-based journey entry and the API mechanism for injecting contacts through an API Event entry source in its Journey Builder API guide. For the endpoint and event-data requirements, see Salesforce's POST /interaction/v1/events reference.

QUESTION NO: 17

A developer started a Contact Delete process that is now complete.

In which two places would the Contact Delete process remove data? Choose 2 answers

A. Non-Sendable Data Extensions

B. Import Files on the Enhanced SFTP

C. Sendable Data Extensions

D. Mobile Lists

ANSWER: C D

Explanation:

Sendable Data Extensions and **Mobile Lists** are correct because Contact Delete is designed to remove a contact's personally identifiable contact data from Marketing Cloud contact-related storage after the deletion process completes. The process deletes the contact from the All Contacts population and removes associated records from sendable data extensions. A sendable data extension participates in subscriber/contact relationships through its configured send relationship, so its contact records are within the scope of Contact Delete processing.

The process also removes the contact's entries from MobileConnect mobile lists. Mobile lists are contact-associated mobile messaging data, and removing those entries prevents the deleted contact from continuing to exist as an addressable mobile contact in those lists. This supports privacy and data-subject deletion requirements across channels, rather than limiting deletion only to email-oriented subscriber records. The deletion is asynchronous and can take time to complete, so downstream sends and automations should account for the deletion status before relying on the data as removed. Salesforce documents the scope and operation of this feature in its Contact Delete documentation and its Contact Management Trailhead unit.

QUESTION NO: 18

A developer is leveraging the SOAP API to dynamically display Profile and PreferenceAttributes in a custom profile center. Which method could be used to support the dynamic functionality?

A. Describe

B. Extract

C. Perform

D. Configure

ANSWER: A

Explanation:

Describe is the appropriate SOAP API method because it retrieves metadata about Marketing Cloud API objects rather than requiring a developer to hard-code an object's available properties. For a custom profile center that must dynamically render Profile and PreferenceAttributes, this metadata can identify the relevant object structure and available fields, including information needed to build controls or determine how values should be handled. The application can call Describe for the applicable API object, inspect the returned definition, and use that response to construct or adjust the profile-management interface as attributes evolve. This approach supports a more maintainable profile center because the UI can be driven by the account's configured data model instead of a fixed list of fields. Salesforce documents Describe as a SOAP API operation for obtaining object definitions and metadata; its response includes the object's properties and associated details. See the Marketing Cloud SOAP API Describe documentation and the SOAP API methods reference for the operation's role in discovering API object information.

QUESTION NO: 19

A sendable data extension with a text field named 'Balance' contains the value S6.96 for a particular record. The following AMPscript statement is included in an email:

IF (Balance > 6.00) THEN

SET @Result = 'Balance is more than $6.00

ENDIF

Why would this IF statement yield unintended results?

A. The operands are not the same data type.

B. The comparison should use the < operator.

C. Balance is a protected keyword.

D. Double quotes should be used instead of single quotes.

ANSWER: A

Explanation:

The operands are not the same data type. `Balance` is defined as a Text field, so its value is supplied to AMPscript as character data, while `6.00` is a numeric literal. A relational comparison intended to test a monetary amount should use a value that is explicitly numeric. In particular, a value containing a currency-style prefix or other nonnumeric character, such as `$6.96` as shown, is not a clean decimal value and can produce unexpected coercion or comparison behavior. Before comparing the balance, the email should retrieve the attribute and normalize it into a numeric representation—for example, remove any formatting character that is not part of the number and then use AMPscript's `Value()` function. The resulting numeric variable can safely be compared with `6.00`. This ensures the condition evaluates based on arithmetic value rather than text formatting. Salesforce documents `Value()` as the AMPscript conversion function for interpreting a string as a number; see the Value() function reference. AMPscript syntax and expression behavior are also covered in the AMPscript Developer Guide.

QUESTION NO: 20

A developer is writing a SQL Query Activity that joins an Orders Data Extension to a Customers Data Extension. The query currently times out because both Data Extensions contain millions of records. Which two practices should the developer use to improve performance? Choose 2.

A. Select only fields required by the target Data Extension

B. Filter records as early as possible

C. Use SELECT * for every joined Data Extension

D. Apply functions to join fields in the WHERE clause

ANSWER: A B

Explanation:

Use only the fields required by the target Data Extension and filter records as early as possible. Selecting explicit fields instead of using `SELECT *` reduces the amount of data Marketing Cloud must read, process, and write. Applying a selective `WHERE` clause before processing unnecessary rows also reduces the size of the data set involved in joins and output operations.

Large joins should also use appropriately defined primary keys and can be divided into staged queries when needed. In contrast, applying functions to fields in the `WHERE` clause can increase processing work, and selecting every field expands the query payload unnecessarily. Salesforce provides supported SQL guidance in the Marketing Cloud SQL Reference and query-design guidance in the Query Data with SQL Trailhead module.

QUESTION NO: 21

Northern Trail Outfitters (NTO) wants to prevent competitors fromreceiving a coupon email. They also want to capture email addresses of competitors who are included in the targeted audience.

Which feature could NTO use to prevent the coupon from being sent and report the email addresses skipped?

A. Auto-Suppression list

B. RaiseError AMPscript function

C. Exclusion Script

D. Try/Catch SSJS functions

ANSWER: C

Explanation:

Exclusion Script is the appropriate feature because it evaluates recipient-level logic immediately before an email is sent. NTO can define criteria that identify competitor email addresses, such as a known domain, a flag in the sendable data extension, or a lookup against a dedicated competitor data extension. When the criteria are met, the script excludes that subscriber from the send, ensuring the coupon is not delivered even if the person was initially included in the target audience.

Because the logic is script-based, NTO can also retain an auditable record of the excluded recipients. The script can use the recipient data and lookup results to capture the applicable email address and other useful context in a reporting data extension, subject to the scripting functions and send-time design used by the account. This provides both the preventative control and the visibility NTO needs to review which targeted records were skipped. Salesforce documents exclusion scripts as a send-time mechanism for excluding subscribers based on defined conditions; see Exclusion Scripts and AMPscript for Marketing Cloud Engagement.

QUESTION NO: 22

Which AMPscript function returns the result of interpreted code within a code block and includes the result in the rendered content, where the code block is located?

A. V

B. Output

C. TreatAsContentArea

D. TreatAsContent

ANSWER: D

Explanation:

`TreatAsContent` is the AMPscript function that evaluates a string as Marketing Cloud content, including any AMPscript expressions or code contained in that string, and returns the resulting rendered output at the location where the function is called. This is useful when AMPscript is stored in a variable or returned dynamically and must be interpreted at send or render time rather than displayed as literal text. For example, a variable can contain an AMPscript expression, and `TreatAsContent` evaluates that expression so its final value becomes part of the email or landing-page content. The function's return value is therefore inserted directly into the surrounding rendered content. Salesforce documents `TreatAsContent()` as treating a string as content, enabling AMPscript in that string to be processed during rendering. See the TreatAsContent() AMPscript reference and the AMPscript Developer Guide.

QUESTION NO: 23

A developer wants to design a custom subscription center in CloudPages. The developer prefers to code in AMPscript, but is also skilled in Server-Side JavaScript. While the developer is confident their code is of high quality, they would still like to handle unexprected errors gracefully to ensure the best user experience. Which feature should handle this scenario?

A. Wrapping the code in a Server-Side JavaScript Try/Catch block

B. Using RaiseError AMPscript function when an error occurs

C. Marketing Cloud automatically handles any error scenario that may occur

D. Wrapping the code in a AMPscript HandleError block

ANSWER: A

Explanation:

Wrapping the code in a Server-Side JavaScript Try/Catch block is the appropriate error-handling feature for a CloudPages subscription center. Server-Side JavaScript provides `try`, `catch`, and optionally `finally` constructs that let a developer run page-processing logic while intercepting runtime exceptions. Within the `catch` block, the page can log contextual details, set a controlled response, and display a friendly message instead of exposing a raw platform error to the subscriber. This is particularly valuable for a custom subscription center, where failures can arise from data-extension access, request values, subscriber updates, or external calls. Although the page's primary business logic can be written in AMPscript, Server-Side JavaScript can be used as the outer control layer to provide structured exception handling around the relevant processing. This pattern preserves the desired AMPscript implementation while giving the developer a reliable place to handle unexpected failures and return an intentional user experience. Salesforce documents the Server-Side JavaScript `Try/Catch` construct and its use for exception handling in CloudPages and other Marketing Cloud server-side contexts. See Server-Side JavaScript Try/Catch and Server-Side JavaScript in Marketing Cloud Engagement.

QUESTION NO: 24

Which statements are true regarding the Marketing Cloud SOAP API? Choose 2.

A. More than 2000 SOAP calls can be performed per minute.

B. Most SOAP calls can be synchronous or asynchronous.

C. Uses XML in request and response body.

D. Uses JSON in request and response body.

ANSWER: B C

Explanation:

The statements **"Most SOAP calls can be synchronous or asynchronous."** and **"Uses XML in request and response body."** accurately describe the Marketing Cloud SOAP API. SOAP is an XML-based web-service protocol: a request is sent as an XML SOAP envelope, and Marketing Cloud returns an XML SOAP envelope containing the response, status information, and any requested objects or errors. This XML message structure is a core characteristic of SOAP integrations.

Marketing Cloud also supports synchronous and asynchronous processing patterns for SOAP operations. With synchronous processing, the client waits for Marketing Cloud to complete the requested operation and return its result. For operations that can require more processing time or involve larger workloads, asynchronous execution enables Marketing Cloud to accept the request and provide a request identifier that can be used to retrieve results or status later. The SOAP API documentation describes the available processing modes and asynchronous request handling, making this an important consideration when designing integrations that create, retrieve, update, or delete Marketing Cloud data.

See Salesforce's Marketing Cloud SOAP API overview and asynchronous API call documentation for details.

QUESTION NO: 25

NTO wants to exclude sends to specific subscribers based on a business rule, which is defined in an Exclusion Script. Which three types of sends would support this functionality? Choose 3

A. Journey Builder Send Email Activity

B. Content Builder Send Flow

C. Send Marketing Cloud Email in Sales or Service Cloud

D. Journey Builder Send SMS Activity

E. Automation Studio Send Email Activity

ANSWER: A B E

Explanation:

Journey Builder Send Email Activity, **Content Builder Send Flow**, and **Automation Studio Send Email Activity** support using an Exclusion Script to suppress individual subscribers at send time. An exclusion script is evaluated in the context of each subscriber being processed for an email send. Its business logic can use subscriber and send-context data to decide whether the subscriber should be omitted; when the script evaluates to true, Marketing Cloud excludes that subscriber from the send.

For a Journey Builder email activity, the script can enforce rules that must be applied when a journey reaches its email step. In the Content Builder send flow, it can be selected as part of configuring a one-time email send. In Automation Studio, it can be applied to the Send Email Activity so recurring or scheduled sends consistently enforce the same subscriber-level suppression rule. This makes exclusion scripts useful when ordinary publication-list or suppression-list membership is insufficient and the decision depends on dynamic data or more detailed business logic. Salesforce documents exclusion scripts as a send-time email exclusion feature and describes their use during email-send configuration. See Salesforce Help: Exclusion Scripts and Salesforce Help: Send Email Activity.

QUESTION NO: 26

A developer started a Contact Delete process that is now complete.

In which twoplaces would the Contact Delete process remove data? Choose 2 answers

A. Non-Sendable Data Extensions

B. Import Files on the Enhanced SFTP

C. Sendable Data Extensions

D. Mobile Lists

ANSWER: C D

Explanation:

The Contact Delete process removes the contact's data from **Sendable Data Extensions** and **Mobile Lists**. A sendable data extension is connected to the contact model through its subscriber or contact relationship and can be used as an audience for sends. Because those records represent sendable contact data, Marketing Cloud removes the matching contact data when the asynchronous contact deletion process completes. This prevents the deleted person from remaining available as a sendable contact through that connected audience source.

Mobile Lists are also included because they contain mobile messaging contact records, such as SMS subscribers. Contact deletion is designed to remove the contact from Marketing Cloud contact-management locations, including the mobile channel's list membership, so the deleted contact is no longer retained as an active mobile-list contact. The deletion process can take time to complete because Marketing Cloud processes the removal across the relevant contact and channel data stores. Salesforce's Contact Builder documentation describes deleting contacts and the associated data managed by the contact model; see Delete Contacts in Contact Builder. For the relationship between sendable data extensions and Marketing Cloud contacts, see Create a Sendable Data Extension.

QUESTION NO: 27

Northern Trails Outfitters (NTO) developers want to use the Transactional Messaging API to send email receipts to customers.

What is the first step required to send using the API?

A. POST to /messaging/vl/email/messages/ with clientjd

B. Request a token usingthe vl/requestToken endpoint

C. Request a token using the v2/authorize endpoint

D. POST to /messaging/vl with clientjd and client_secret

E. Request an access token using the /v2/token endpoint.

ANSWER: E

Explanation:

Requesting an OAuth access token from the `/v2/token` endpoint is the required first step. The Transactional Messaging API requires authentication before an application can submit a message definition or send an email message. The application makes a POST request to the Marketing Cloud OAuth token endpoint using its installed package credentials, including the client ID, client secret, and the `client_credentials` grant type. Marketing Cloud returns an access token and the appropriate REST instance URL. The returned token is then supplied as a Bearer token in the `Authorization` header for subsequent Transactional Messaging API requests. Using the instance-specific REST URL returned by authentication is important because API endpoints are tenant and stack specific. Access tokens are temporary, so an integration should obtain a new token when the current token expires rather than attempting to send API requests without valid authorization. Salesforce documents the OAuth token request and required credentials in its Marketing Cloud access token documentation. The authentication requirement and API request flow are also described in the Transactional Messaging API documentation.

QUESTION NO: 28

A developer wants to trigger an SMS message to a subscriber using a form published on CloudPages. How should the SMS message be triggered once the subscriber submits the form?

A. Outbound SMS template and Automation Send Method

B. InsertData AMPscript function to add the subscriber to a MobileConnect list

C. CreateSMSConversation AMPscript function

D. requestToken and messageContact REST API objects

ANSWER: C

Explanation:

The **CreateSMSConversation AMPscript function** is designed to initiate an SMS conversation and send an SMS message from a CloudPages form-processing page. When the form is submitted, the CloudPage can retrieve the submitted mobile number and any relevant values using AMPscript request functions, then call `CreateSMSConversation` with the applicable SMS conversation definition, recipient, message content, and optional subscriber attributes. This provides a synchronous, server-side way to trigger the MobileConnect message as part of processing the form submission.

The function uses a configured SMS conversation definition, which allows the developer to apply the MobileConnect configuration and messaging behavior established in the Marketing Cloud account. The recipient must be supplied in a valid mobile-number format, and the account must be configured for the appropriate SMS sending capability and compliance requirements. This approach is well suited to a CloudPage because AMPscript executes when Marketing Cloud processes the form request, allowing the submission event itself to initiate the SMS interaction. Salesforce documents the function, its parameters, and its behavior in the CreateSMSConversation AMPscript reference. See also Salesforce's AMPscript documentation for its use in CloudPages and other Marketing Cloud content.

QUESTION NO: 29

A developer receives a request to integrate Marketing Cloud with a lead capture tool. The lead capture tool will call the Marketing Cloud API to create a data extension every time a new lead form is published. The created data extension's name should match the name of the form exactly.

Which API feature could the developer use to dynamically create these data extensions?

A. SOAP API using Create Method and the DataExtension Object

B. REST API using POST on the /interaction/v1/EventDefinitions endpoint with Schema populated

C. REST API using POST on the /data/v1/customobjectdata/ endpoint

D. Creating the data extension using API is not possible

ANSWER: A

Explanation:

SOAP API using Create Method and the DataExtension Object is the appropriate feature because Marketing Cloud exposes data extensions as SOAP API objects that can be created programmatically. The integration can send a SOAP `CreateRequest` containing a `DataExtension` object, setting its `Name` property dynamically from the lead form name. The request can also define required metadata such as the customer key, folder/category identifier, and whether records are retained. To make the data extension usable for storing lead information, the request includes one or more `DataExtensionField` definitions, with each field's name, data type, length where applicable, and primary-key or required settings.

This approach directly meets the requirement that a distinct data extension be provisioned each time a form is published, while preserving the form's name as the data extension name. The calling application should generate a stable, unique customer key as well, since names are intended for display and customer keys are commonly used as durable identifiers in subsequent API calls. Salesforce documents creating data extensions with the SOAP API and the properties available on the DataExtension object in its Create a Data Extension Using the Web Service API guide and DataExtension SOAP API reference.

QUESTION NO: 30

Certification Aid wants to create Contacts in Marketing Cloud via API calls. Which API should be used for this? Choose 2.

A. POST /contacts/v1/contacts route

B. SOAP API

C. REST API

D. Contact object

ANSWER: A C

Explanation:

The REST API is the appropriate Marketing Cloud API family for creating and managing Contact Builder contacts programmatically. A contact is identified by its contact key, and the REST Contacts resource provides operations for creating the contact and associating any supplied attribute-set data with that contact. This approach supports an external application that needs to submit contact records directly into Marketing Cloud without relying on a send or other workflow to create them implicitly.

Specifically, `POST /contacts/v1/contacts` is the REST Contacts endpoint used to create a contact. A request to this route includes the contact key and can include attribute-set values in the request payload. The calling application must first obtain an OAuth access token from Marketing Cloud and send that token in the authorization header when invoking the endpoint. Using the REST API together with this POST route is therefore the documented mechanism for this contact-creation use case. Salesforce's REST Contacts reference describes the create-contact operation at Create a Contact, and the platform's API authentication process is documented in the Access Token documentation.

QUESTION NO: 31

A developer is troubleshooting the cause of incomplete results in the link tracking data for an email send.

How should the RedirectTo AMPscript function be described as it relates to link tracking'

A. It ensures link href values containing AMPscript variables are recorded in tracking

B. It ensures link href values containing HTML bookmarks or anchors are recorded in tracking

C. It prevents link href values from getting recorded in tracking

D. It ensures static link href values are recorded in tracking

ANSWER: A

Explanation:

It ensures link href values containing AMPscript variables are recorded in tracking. In Marketing Cloud Engagement, ordinary tracked links are identified when an email is prepared for sending. A URL that is assembled dynamically with AMPscript values cannot always be resolved as a conventional static href during that process. The `RedirectTo()` function is designed for this situation: it accepts the destination URL expression, generates a tracked redirect URL, and sends the recipient to the evaluated destination when they click it. This lets Marketing Cloud associate the click with the email send and preserve link-level tracking data even when the underlying URL includes subscriber-specific or other dynamically evaluated values.

For example, a link whose query-string parameters are populated from contact data can use `RedirectTo(Concat(...))` in its href. Marketing Cloud evaluates the expression for the recipient while retaining its redirect and tracking mechanism, so clicks can appear in tracking results. This usage is particularly relevant when diagnosing apparent gaps in click reporting for dynamically generated destinations. Salesforce documents `RedirectTo()` as a function that redirects a subscriber to a specified URL and tracks the click. See the Salesforce RedirectTo() reference and the Marketing Cloud click tracking documentation.

QUESTION NO: 32

Certification Aid wants to import data from a CSV file into a Data Extension. The CSV filecontains all relevant data. New records should be added to the Data Extension, and records which are not in the file should be removed from the Data Extension. Which import operation should be chosen for this? Choose 1.

A. Add only

B. Overwrite

C. Add andupdate

D. Update only

ANSWER: B

Explanation:

Overwrite is the appropriate import operation because the CSV file is stated to contain the complete, authoritative set of relevant data. An overwrite import first clears the existing rows from the target data extension and then loads the rows contained in the import file. Consequently, records represented in the CSV are present after the import, including newly introduced records, while records that existed only in the prior version of the data extension are removed because they are not reloaded. This makes overwrite suitable for a full-refresh workflow in which the source file should define the entire final contents of the data extension.

Before using this operation in production, the data extension’s fields, data types, and primary-key design should be validated against the incoming file, and the file should be treated as a complete snapshot rather than a partial delta. Salesforce documents the available import actions for data extensions, including the behavior of overwrite, in its Import Activity

documentation. For implementation context and automation patterns, see the Marketing Cloud Import Activity developer documentation.

QUESTION NO: 33

A developer is troubleshooting why a parent-level data extension cannot be accessed by a child business unit.

What should the developer check to validatethe data available can be accessed for child business unit queries?

A. The data extension is in the Shared Data Extensions folder and the query includes the ENT. prefix

B. The data extension is in the Shared Items root folder and is accessible to the child business unit

C. The data extension is in the Salesforce Data Extensions folder and Is accessible to the child business unit

D. The data extension is in the Synchronized Data Extensions folder and the query includes the ENT. prefix

ANSWER: A

Explanation:

The data extension is in the Shared Data Extensions folder and the query includes the ENT. prefix is correct because parent-level data extensions must be explicitly shared for use by child business units. In Marketing Cloud Engagement, placing a data extension in the Shared Data Extensions area makes it available to the applicable child business units according to the sharing configuration. This is distinct from merely storing an asset in a parent business unit, which does not by itself make the data extension queryable from a child context.

When a query activity runs in a child business unit, it must identify an enterprise-level shared data extension with the `ENT.` prefix. For example, a query can reference a shared data extension as `ENT.[SharedDataExtensionName]`. The prefix instructs Marketing Cloud Engagement to resolve the data extension in the enterprise/shared context rather than looking only within the child business unit's local data extensions. Therefore, validating both the shared-folder location and the enterprise-qualified query reference confirms the required setup for child-business-unit queries. Salesforce's SQL guidance and shared-data documentation provide the relevant enterprise and sharing behavior: Marketing Cloud Engagement SQL Reference and Shared Data Extensions.