PassQueen
a.k.a dumpsqueen.com

Salesforce Order Management Administrator Accredited Professional

Salesforce Order-Management-Administrator

Version Demo

Total Demo Questions: 13

Total Premium Questions: 132

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Order Management Fundamentals	23
Topic 2, Order Management Setup	62
Topic 3, Order Management Operations	24
Topic 4, Order Management Integrations	19
Topic 5, Mix Questions	4
Total	132

QUESTION NO: 1

An administrator is building a record-triggered flow that evaluates Fulfillment Order Products across several fulfillment locations and updates the products selected for reassignment. Which two practices should the administrator use to reduce the risk of Salesforce limit exceptions? Choose 2 answers

A. Retrieve the required records before entering the loop

B. Update each Fulfillment Order Product during its loop iteration

C. Store changed records in a collection and update them after the loop

D. Create a separate flow interview for every fulfillment location

ANSWER: A C

Explanation:

Get Records elements should be placed outside loops whenever possible. Retrieving related fulfillment records once and processing them in a collection avoids issuing a separate query for each loop iteration.

The administrator should also add changed records to a collection and update the collection after the loop. This reduces the number of DML operations and helps the flow scale when an order contains many products or is distributed across multiple locations.

Creating or updating records inside a loop can cause excessive DML statements. Multiple locations do not inherently require Apex code; a properly designed flow can handle the process declaratively.

QUESTION NO: 2

When can an Invoice be created?

A. At anypoint before the Fulfillment Order is created, by customizing flow "Create Invoice and Ensure Funds"

B. At any point before the Fulfillment Order is created, by customizing flow "Create invoice"

C. At any point after the Fulfillment Order is created, by customizing flow 'Create Invoice and Ensure Funds*

D. At any point after the Fulfillment Order is created, by customizing flow "Create Invoice"

ANSWER: D

Explanation:

An invoice can be created at any point after the Fulfillment Order has been created by customizing the *Create Invoice* flow. In Salesforce Order Management, the Create Invoice core action operates from a Fulfillment Order record and creates the related Invoice record. Because the Fulfillment Order is the source context for the action, it must exist before invoice creation can occur. This gives an administrator flexibility to align invoicing with the organization's operational process—for example, creating an invoice when fulfillment begins, after items are shipped, or at another appropriate post-fulfillment-order milestone.

The flow is intended to be customized rather than treated as a fixed timing mechanism. An administrator can add decision logic, additional updates, notifications, or subflows around the core action to enforce business-specific requirements while retaining the supported invoice-creation behavior. The action also updates the relevant record statuses as part of its processing, which helps keep fulfillment and invoicing progress synchronized. Salesforce documents the required Fulfillment Order context and behavior of this action in the Create Invoice Core Action reference. For broader implementation context, see Salesforce's Order Management overview.

QUESTION NO: 3

An administrator is creating a flow that evaluates Fulfillment Order Products from several locations and updates the products selected for fulfillment. Which two design practices should the administrator use to improve scalability? Choose 2 answers.

A. Get the required Fulfillment Order Products before entering the loop

B. Update a collection of changed records after the loop completes

C. Use an Update Records element inside each loop iteration

D. Use a Get Records element inside each loop iteration

E. Convert the automation to a screen flow

ANSWER: A B

Explanation:

Getting the required records before entering a loop reduces repeated queries as the number of fulfillment products and locations increases. Updating a record collection after the loop similarly avoids performing separate DML operations for every item processed.

Using Get Records or Update Records inside each loop iteration can cause the flow to exceed transaction limits when order volume grows. A screen flow is not required for background fulfillment automation and does not address limit consumption.

QUESTION NO: 4

Which option is available for an admin to use as middleware to import data to. or export data from Order Management?

A. Use Workbench to access the database

B. Use the Developer console

C. Find an application on AppExchange

D. Write Apex classes to query the database and write to a file

E. Use Data Loader

ANSWER: E

Explanation:

Use Data Loader is correct because Data Loader is Salesforce's supported client application for bulk importing and exporting Salesforce records through comma-separated value files. An administrator can use it as an integration or migration utility when Order Management data must be loaded into Salesforce or extracted for reconciliation, reporting, archival, or use by an external process. Data Loader provides insert, update, upsert, delete, hard delete, and export operations, allowing an administrator to map file columns to the relevant Salesforce object fields and process substantial record volumes efficiently. For Order Management implementations, the tool is particularly useful during data migration, operational corrections, and controlled bulk maintenance where the target Order Management objects and fields support the required data operation. Administrators should prepare valid record relationships and required field values before loading data, and use export operations when a portable file-based extract is needed. Salesforce documents Data Loader as a desktop client for importing and exporting data, including its supported operations and file-based workflow. See Salesforce Data Loader documentation and Salesforce Data Loader Developer Guide.

QUESTION NO: 5

A service manager asks why agents use Order Summary records instead of updating the original Order after checkout. Which statement best describes the purpose of an Order Summary?

A. A dynamic view of underlying order data

B. A replacement record for the original Order

C. A record that represents one financial adjustment

D. A record that represents one fulfillment shipment

ANSWER: A

Explanation:

An **Order Summary** provides a dynamic, consolidated view of the underlying order data for downstream Order Management work. It is the operational order record used for activities such as fulfillment, returns, payments, and service. It does not replace the source Order with an unrelated transaction, and it is not simply a record for a single discount or financial adjustment.

QUESTION NO: 6

A customer places an order containing products fulfilled from two locations. Before fulfillment, the customer cancels one item. After delivery, the customer returns another item. Which three Order Management record types can be created to support these lifecycle events? Choose 3 answers

A. Fulfillment Order

B. Change Order

C. Return Order

D. Payment Gateway Log

E. Invoice

ANSWER: A B C

Explanation:

A Fulfillment Order represents the portion of the order assigned to a fulfillment location. Because the order is fulfilled from two locations, Order Management can create fulfillment work separately for the applicable locations.

A Change Order records the requested post-order cancellation or amendment. A Return Order manages the later reverse-logistics process for the delivered item that the customer returns.

An Invoice is a financial document and is not created merely because an item is cancelled or returned. A Payment Gateway Log records gateway communications rather than the operational order-change or return process.

QUESTION NO: 7

An administrator has completed testing of a new Order Management flow in a sandbox and needs to move it to production using a change set. Which three actions are part of the required deployment process? Choose 3 answers.

A. Create an outbound change set in the sandbox

B. Upload the outbound change set to production

C. Deploy the inbound change set in production

D. Approve the change set in the source sandbox

E. Add the change set to a release schedule in production

ANSWER: A B C

Explanation:

The administrator creates an outbound change set in the source sandbox, adds the required metadata and dependencies, and uploads it to the connected target organization. In production, the uploaded change set appears as an inbound change set and must be deployed to apply the changes.

Scheduling the change set or approving it in the source sandbox is not a standard change-set deployment step. Production can validate an inbound change set before deployment, but validation does not itself apply the metadata.

QUESTION NO: 8

An administrator is preparing a new Order Management org to test authorization, capture, and refund processing against a mock payment service. Which three configuration records are required for the mock payment gateway setup? Choose 3 answers

A. Payment Gateway Provider

B. Named Credential

C. Payment Gateway

D. CORS Allowlist Entry

E. Trusted Site Entry

ANSWER: A B C

Explanation:

A Payment Gateway Provider defines the provider-side integration configuration used for payment processing. A Named Credential securely stores the endpoint and authentication details required for the callout to the mock service. A Payment Gateway makes the configured provider available for payment operations.

CORS and Trusted Site entries can be relevant to browser-based integrations, but they do not establish the Order Management payment gateway configuration required for this server-side payment processing scenario.

QUESTION NO: 9

Some admins are exploring the optimal Data Model for their QMS Org. What should be considered when choosing between Person Accounts vs Contacts?

A. Person Accounts once enabled cannot be rolled back and makes changes to the data model

B. Person Accounts once enabled can be rolled back

C. Person Accounts are appropriate for B2B transactions while Account-Contact model is appropriate for B2C transactions

D. Person Accounts are appropriate for B2C transactions while Account-Contact model is appropriate for B2B transactions

ANSWER: D

Explanation:

Person Accounts are appropriate for B2C transactions while Account-Contact model is appropriate for B2B transactions. This distinction follows the core purpose of each Salesforce data model. A Person Account represents an individual consumer and combines relevant Account and Contact characteristics into one record. It is therefore well suited to organizations that sell directly to people, where each customer generally has an individual relationship with the business and does not need to be modeled as an employee or representative of a company.

For business customers, the Account-Contact model provides the necessary organizational structure. The Account represents the company, while related Contact records represent the people associated with that company, such as buyers,

administrators, or stakeholders. This supports common B2B requirements, including maintaining multiple relationships under one organization and tracking activities or transactions at both the business and individual levels. Administrators should begin by identifying whether the organization's customers are primarily individual consumers or businesses with multiple associated people, because that decision drives the most appropriate model. Salesforce describes Person Accounts and their intended use in its Person Accounts documentation and explains the standard relationship between accounts and contacts in Accounts and Contacts documentation.

QUESTION NO: 10

What type of relationship exists between FulfilmentOrder and Location?

A. One-to-one Lookup (Location) FulfilledFromLocationId

B. Many-to-One Lookup (Location) FulfilledfromLocationId

C. Junction (Many to Many)

D. One-to-many Lookup (Location) FulfilledFromLocation

ANSWER: B

Explanation:

Many-to-One Lookup (Location) FulfilledfromLocationId is correct because a FulfillmentOrder identifies the location from which its products are to be fulfilled by using the `FulfilledFromLocationId` field. This field is a lookup reference from the FulfillmentOrder record to one Location record. Therefore, each individual FulfillmentOrder has one fulfilled-from location, while the same Location can be referenced by numerous FulfillmentOrder records over time. From the FulfillmentOrder side of the relationship, this is many orders to one location.

In Salesforce Order Management, Location records model physical inventory, fulfillment, or operational sites. Associating the fulfillment order to its source location enables fulfillment processes to determine where items should be picked, packed, and shipped. The relationship is implemented as a standard lookup field rather than a unique one-to-one association, which supports the normal business case of processing many customer orders from the same warehouse, store, or distribution center. Salesforce documents `FulfilledFromLocationId` as the Location reference on the FulfillmentOrder object; see the FulfillmentOrder object reference and the Location object reference.

QUESTION NO: 11

An Administrator has created a change set with two items, a new apex class

and a new apex trigger. When attempting to deploy the change set, and error

occurs? What could be the cause?

A. < br >

B. Different Metadata types cannot be deployed within a single change set < br >

C. An Apex Trigger cannot be deployed via change set < br >

D. Apex Classes require a header file when deployed < br >

E. The new Apex class and trigger have lowered the org's overall Apex code coverage below the required 75% threshold.

ANSWER: E

Explanation:

The new Apex class and trigger have lowered the org's overall Apex code coverage below the required 75% threshold. Salesforce validates Apex code as part of a production deployment, including deployments made with change sets. The deployment process runs the applicable tests in the target organization and calculates test coverage for the organization's

Apex code. To complete deployment successfully, the organization must maintain at least 75% aggregate Apex code coverage. In addition, every Apex trigger must receive some test coverage. A newly deployed class or trigger that is not adequately exercised by test methods can reduce the aggregate percentage enough to make the validation or deployment fail. Administrators should ensure that test classes cover the new business logic and trigger behavior, then run the relevant tests in the target environment before retrying the change set deployment. Salesforce documents the deployment coverage requirement and trigger-coverage requirement in its Apex Testing documentation. Change sets are a supported mechanism for deploying metadata between related Salesforce organizations; see Salesforce Change Sets documentation.

QUESTION NO: 12

Which of the following options represent two ways that an administrator can utilize APIs in Workbench to query OrderSummaries?

A. Apex API and Connect API

B. Connect API and Platform API

C. Platform API and Composite API

D. Composite API and Invocable API

E. Invocable API and Apex API

ANSWER: B

Explanation:

Connect API and Platform API are the two applicable API approaches for querying order summary information through Workbench. Salesforce Order Management exposes order-summary resources through the Commerce section of Connect API. An administrator can use Workbench's REST Explorer to send an authenticated request to the Order Summaries Connect API resource, which is designed to retrieve order summary data and related commerce information in the context of a buyer or order-management workflow. Salesforce also makes the `OrderSummary` standard object available through the Salesforce Platform REST API. In Workbench, an administrator can use the REST query resource and SOQL to retrieve selected `OrderSummary` fields, filter records, and inspect the returned JSON response. This is particularly useful for validating data, investigating an order's processing state, and testing integrations without writing custom code. The two methods complement each other: Connect API provides the commerce-oriented order-summary resource, while the Platform API provides direct object access through standard REST and SOQL patterns. Salesforce documents the order-summary Connect API resource in the Connect API Reference and SOQL query requests in the REST API Query Resource documentation.

QUESTION NO: 13

Once the administrator hasactivated the integration between B2C Commerce and Order Management, orders being placed by customers will be sent to Order Management if the order status is set to which two values?

A. Created

B. Active

C. Drafted

D. New

E. Open

ANSWER: D E

Explanation:

New and **Open** are the B2C Commerce order statuses used by the B2C Commerce–Order Management integration to identify orders eligible for export to Salesforce Order Management. A customer checkout ordinarily transitions an order into the New state. When the integration detects an order in New or Open status, it can send the order payload to Order Management so that downstream fulfillment processes, such as allocation, shipping, and service operations, can begin. Open is also an eligible status because it represents an order that remains active for processing in the B2C Commerce order lifecycle. These status values are part of B2C Commerce's defined order-status model; the platform exposes its order status constants through the Order API, including New and Open. Using the expected status values ensures that the integration processes confirmed, actionable customer orders consistently and avoids relying on a status that is not part of the applicable B2C Commerce export criteria. See the B2C Commerce Order API status documentation and Salesforce's Order Management API documentation.