PassQueen
a.k.a dumpsqueen.com

Salesforce Process Automation Accredited Professional

Salesforce Process-Automation

Version Demo

Total Demo Questions: 19

Total Premium Questions: 196

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

What are three best practices a business analyst should keep in mind when creating a Flow?

A. Create a draft version of the flow and delete it after the real version has been successfully run at least once.

B. Plan out their flow before they start building.

C. Identify the Salesforce IDs and hardcode them in a process so they can be easily referenced.

D. Provide an error handler.

E. Wait until the end of the flow to make changes to the database.

ANSWER: B D E

Explanation:

Planning the flow before building it is a core practice because it lets the business analyst map the business process, define entry criteria and outcomes, identify required data, and decide where decisions, loops, and reusable logic belong. A clear design reduces rework and makes the finished automation easier for others to understand and maintain. Providing an error handler is equally important. Fault paths enable a flow to respond predictably when an operation fails, such as by showing an appropriate message to a user or notifying an administrator with useful troubleshooting context. Salesforce recommends using fault paths so failures are handled intentionally rather than leaving users with an unhelpful generic error. Waiting until the end of the flow to make changes to the database supports efficient automation design. Flow interviews should collect record changes and perform them together when practical, minimizing database operations and helping the automation stay within Salesforce platform limits. This pattern also makes the transaction easier to reason about. Salesforce's guidance on flow building and error handling is available in Flow Best Practices and Handle Flow Errors with Fault Paths.

QUESTION NO: 2

Zephyrus Relocation Services (ZRS) is looking to automate some of its processes that use third-party data stored in an external system. The Administrator recently learned about using Local Action in Lightning Aura Components for accessing and interacting with third-party data. Which two limitations should the Administrator keep in mind?

A. Flows that include Lightning Components are supported only in Lightning Runtime

B. Local Action can only be used in Screen Flows

C. Lightning Components that access third-party data from within a Flow can be tested only in Partial Copy or Full Sandbox

D. Autolaunched Flows with at least one Local Action can only run in system context

ANSWER: A B

Explanation:

"Flows that include Lightning Components are supported only in Lightning Runtime" is correct because a flow that renders an Aura Lightning component, including a component exposed as a local action, requires the Lightning flow runtime. This runtime provides the client-side environment in which the component can render, accept user interaction, and execute its JavaScript-based behavior. Administrators should therefore ensure that users launch the screen flow from a Lightning-supported experience when the automation depends on such a component.

"Local Action can only be used in Screen Flows" is also correct. A local action is designed for client-side execution during a user's flow interview. It can use an Aura component to perform custom client-side work, such as interacting with browser capabilities or an external system through component logic, and can display an interface where appropriate. Because this behavior requires an active user session and Lightning runtime, it is limited to screen flows rather than background automation. Salesforce documents local actions as a flow action type for extending screen-flow behavior with Lightning components. See Salesforce Flow Reference: Local Actions and Salesforce Developer Guide: Flow Screen Components.

QUESTION NO: 3

A record-triggered flow updates a related Account whenever a Contact is changed. The Account update causes another flow to update Contacts, and the automation repeats. What should the Administrator do to help prevent this recursion?

A. Add entry criteria that check whether the relevant field changed.

B. Move every Update Records element into a loop.

C. Replace the record-triggered flows with screen flows.

D. Make both flows run only when a user clicks Debug.

ANSWER: A

Explanation:

The administrator should add **entry criteria that verify a relevant change occurred** before allowing the flow to run. For example, the Contact-triggered flow can evaluate whether the field that requires the Account update changed from its prior value. This prevents the flow from performing the same update again when an unrelated downstream update occurs.

Careful entry criteria and conditions that compare current and prior values are important when multiple record-triggered flows update related objects. See Salesforce guidance on Flow best practices.

QUESTION NO: 4

Once a flow is activated, what are the two requirements for any user to run the flow?

A. Access to all the records referenced inside the flow and all the referenced fields in those records.

B. Access to all the records referenced inside the flow.

C. The"Flow User" field enabled on their user detail page.

D. The “Run Flows" user permission.

ANSWER: A D

Explanation:

Users must have the **Run Flows** user permission and appropriate access to the records and fields that the flow references. The Run Flows permission is the baseline permission that allows a user to execute active flows. Administrators can provide it through a profile or permission set, which makes it possible to grant flow-running capability without broadly changing unrelated permissions.

Record and field access remains essential when a flow runs in user context. The user needs the ability to access the records the flow reads, creates, updates, or otherwise references, and the applicable field-level security for the fields used by the flow. This ensures that flow automation respects the organization’s sharing model and data-security configuration while carrying out its work. Flow builders should therefore test with representative users, rather than only as an administrator, to confirm that the required object, record, and field access is present. Salesforce documents flow security contexts and the interaction between flows, permissions, and data access in its Flow Security guide. For details on granting permissions through profiles and permission sets, see Salesforce’s Profiles and Permission Sets documentation.

QUESTION NO: 5

Which two benefits does an autolaunched flow provide when it is invoked as a subflow?

A. It allows common automation logic to be reused by multiple flows.

B. It can accept input values and return output values to the parent flow.

C. It automatically displays its own screens to users.

D. It can be activated without a flow version.

ANSWER: A B

Explanation:

An autolaunched flow invoked as a subflow supports **reusable automation logic**. A flow builder can maintain common steps, such as calculations or record-processing logic, in one flow and invoke it from multiple parent flows. It can also receive input values and return output values, allowing parent flows to pass context and use the result.

Autolaunched flows do not display screens to users. A screen flow is required when the process must collect user input through an interactive interface. See Salesforce documentation for calling flows from flows.

QUESTION NO: 6

When using a Decision element, what does a flow do if no outcome conditions are met?

A. It fails and sends the Administrator an email.

B. Provide a Screen element where users can verify entered values, make corrections, and proceed.

C. It ends with no action taken.

D. It takes the path for the default outcome.

ANSWER: D

Explanation:

It takes the path for the default outcome. In Salesforce Flow Builder, a Decision element evaluates its defined outcomes in the configured order. Each outcome contains condition logic that determines whether the flow should follow that outcome's connector. When none of those explicitly defined outcome conditions evaluates to true, Flow uses the Decision element's Default Outcome. This ensures that every record, user interaction, or other flow interview arriving at the Decision element has a defined route forward, including situations that were not anticipated by the more specific conditional outcomes. The default path can contain any appropriate follow-up logic, such as updating data, showing a screen, invoking an action, or ending the flow. Administrators should design that path intentionally, because it is the handling route for all cases that do not satisfy a named outcome. Salesforce identifies the Default Outcome as the result used when no other outcome's conditions are met. See Salesforce's Decision Element reference and the Flow logic Trailhead module for details on configuring decision branches and default handling.

QUESTION NO: 7

Which three things should an Administrator consider when drafting a list of test cases prior to testing a flow?

A. When the tester expects actions to not occur.

B. How many users will be accessing the flow.

C. How formulas should resolve.

D. How long should the flow take to execute.

E. When the tester expects actions to occur.

ANSWER: A C E

Explanation:

Effective flow test cases must cover both the intended automation behavior and the boundaries that prevent automation from running. “When the tester expects actions to occur” is essential because each test should identify the record data, user inputs, or triggering event that should cause the flow to follow a particular path and produce a verifiable result, such as updating a field, creating a related record, sending an alert, or displaying the appropriate screen outcome. “When the tester expects actions to not occur” is equally important: administrators need test data for situations in which entry criteria, decisions, validations, or conditional paths should prevent an action. This confirms that the flow does not create unintended changes. “How formulas should resolve” belongs in the test plan because formulas often determine routing, decisions, assignments, and values written by the flow. Test cases should specify representative values, boundary values, and the expected formula result so that the administrator can verify that the flow takes the appropriate path. Salesforce’s implementation guidance emphasizes planning for positive and negative scenarios and validating expected outcomes before deployment. See Plan for Flow Implementation and Test Flows.

QUESTION NO: 8

Which three main categories can Flow elements be broken down into?

A. Guided visual processes, behind-the-scenes automation, and approval automations.

B. Interaction, logic, and actions.

C. Logic, actions, and connectors.

D. Variables, choices, and stages.

ANSWER: B

Explanation:

Interaction, logic, and actions. is correct because Flow Builder organizes its elements by the function they perform in an automation. Interaction elements support user-facing steps, most notably Screen, where a flow displays information or collects input. Logic elements control the path and processing of the flow, such as evaluating criteria, assigning values, looping through collections, or pausing execution. Action elements perform work, including creating, updating, retrieving, or deleting records; invoking a subflow; sending notifications; or calling other available actions. This functional grouping helps a builder choose the appropriate element based on whether the flow needs to engage a user, determine processing behavior, or carry out an operation. Salesforce’s Flow Builder documentation describes the available elements and their roles, while Trailhead’s Flow Builder learning material explains how logic and action elements are used to implement automation behavior. See Salesforce’s Flow Element Reference and the Flow Basics Trailhead module for the element categories and usage details.

QUESTION NO: 9

Cloud Kicks (CK) is evaluating outbound message actions to send pricing updates to ….

A. If the endpoint is unavailable, outbound messages are lost after 3 unsuccessful retries.

B. Outbound messages could potentially be delivered out of order.

C. Audit trail is not available for outbound messages.

D. Admin can configure up to 5 outbound message types for guaranteed delivery.

ANSWER: B

Explanation:

Outbound messages could potentially be delivered out of order. Salesforce outbound messaging is an asynchronous integration mechanism: when the relevant workflow rule or approval process is evaluated, Salesforce places a SOAP notification in the outbound-message queue for delivery to the configured endpoint. Because notifications are processed asynchronously and delivery can involve retries, a receiving system must not assume that message arrival order always matches the order in which the underlying Salesforce records were changed. For pricing updates, Cloud Kicks should

therefore design the endpoint to be idempotent and to evaluate a version, timestamp, or other sequencing value before applying an update. This prevents an older notification that arrives later from overwriting newer pricing data. Salesforce's outbound messaging documentation specifically describes the queued, asynchronous delivery model and advises implementers to account for delivery behavior when building the listener. See Salesforce SOAP API Developer Guide: Outbound Messaging and Salesforce Help: Monitor Outbound Messages.

QUESTION NO: 10

Northern Trail Outfitters (NTO) is looking to build an app for its logistics team where users will be able to submit inventory refill requests. The current manual process involves filling out a long form containing many fields. NTO is planning to replace the current process with Flows. Users frequently change field values as they are filling out the form. What should NTO keep in mind about letting users go backward in the Flow to make updates to field values?

A. "Allow Navigation" needs to be set to TRUE on the Flow.

B. Only users with "Navigate Flows" permission can navigate to previous screens.

C. Letting users navigate from a later screen to a previous screen could result in duplicates.

D. Once the user navigates to a previous screen, all field values on the current screen will be lost and will need to be repopulated.

ANSWER: D

Explanation:

Once the user navigates to a previous screen, all field values on the current screen will be lost and will need to be repopulated. In a screen flow, screen-component input is committed to the flow only when the user proceeds forward from that screen. If the user selects the standard Previous navigation control while completing a later screen, the flow returns to the earlier screen without committing the in-progress values from the screen they left. Consequently, if the user later moves forward again, they must re-enter the values that had been typed or changed on that later screen.

This behavior is especially important for NTO's long inventory-refill form because users are likely to review earlier answers and make corrections. The flow design should therefore minimize unnecessary screen changes, group related inputs thoughtfully, and clearly communicate that going back can discard unfinished data on the active screen. Where appropriate, designers can use flow variables and screen configuration to preserve values that have already been submitted from earlier screens. Salesforce documents screen-flow navigation and the behavior of screen input in its Screen Flow reference and provides implementation guidance in Trailhead's Screen Flows module.

QUESTION NO: 11

The architect is designing a flow where a screen flow is used to create a contact and display a

confirmation screen. While the confirmation screen is displayed, remote API is invoked to update

the contact in the external system. The update fails. How should the architect resolve the design?

A. Add error handling mechanism to the flow and test often and early.

B. Use a separate transaction to update the contact in the external system.

C. Use Apex since updating a contact in the remote system is not possible in flow.

D. Use the Process builder instead of flow.

ANSWER: A

Explanation:

Add error handling mechanism to the flow and test often and early. A flow that invokes a remote API must anticipate that the external service can return an error, time out, or be temporarily unavailable. The architect should configure fault handling for

the element that performs the external update, such as an action or invocable callout. Its fault path can capture the flow error details, present an appropriate outcome to the user when applicable, and create a record or notify an administrator so the failed synchronization can be investigated and retried through a defined process. This design makes the integration failure explicit and manageable rather than allowing an unhandled flow error after the contact has been created. Testing the fault path early is important because integration failures commonly depend on authentication, response formats, network conditions, and downstream availability; a successful-path-only test does not validate the operational design. Salesforce recommends using fault paths to manage errors from flow elements and make flows more resilient. See Salesforce Trailhead's guidance on adding fault handling to flows and the Salesforce documentation on fault connectors.

QUESTION NO: 12

Which of the following should be used to branch a flow?

A. Branching Element

B. Decision Element

C. If condition

D. If Elase condition

ANSWER: B

Explanation:

Use a **Decision Element** to branch a Salesforce flow. A Decision evaluates conditions using the data and resources available at that point in the flow, then directs the interview down one of several outcome paths. Each outcome has condition requirements, such as whether a record field has a particular value, whether a variable is blank, or whether a formula evaluates to true. Flow follows the first outcome whose conditions are met. If no defined outcome matches, Flow can follow the Decision element's default outcome, which ensures the automation has a defined path for records that do not meet any explicit criteria. This makes the Decision element the standard declarative tool for implementing conditional logic, routing work, and controlling which actions run next. Salesforce's Flow Builder documentation describes Decision as the element used to evaluate conditions and control the path that a flow takes. See Salesforce Help: Decision Element and Trailhead: Define Flow Logic.

QUESTION NO: 13

Which three flow resources can be used to store or calculate values during a flow interview?

A. Variable

B. Constant

C. Formula

D. Connector

E. Fault Path

ANSWER: A B C

Explanation:

A **Variable** stores a value that can change as the flow runs. A **Constant** stores a value that is set when the flow is configured and remains unchanged during the interview. A **Formula** calculates a value dynamically from other available resources and record fields.

These resources help a flow manage data and logic without requiring hard-coded record IDs or repeated configuration. A connector controls the path between elements; it is not a resource used to store or calculate a value. See Salesforce documentation on Flow resources.

QUESTION NO: 14

What is a flow interview?

A. Questions posed by flow designer to potential flow users.

B. A flow that takes the same path as the original flow.

C. Instance of a flow.

D. Connection or interlink between two to more internal elements of a flow.

ANSWER: C

Explanation:

An instance of a flow is correct. In Salesforce terminology, a flow interview is a particular execution, or run, of a flow. Each time a user starts a screen flow, or Salesforce invokes an eligible automated flow, Salesforce creates a separate interview to hold that run's context. The interview tracks the flow's progress and the values held in its variables while the flow is executing. For a screen flow, it represents the user's individual journey through screens and decisions; for an automated flow, it represents the specific execution processing the relevant record or event. Interviews can also be paused when the flow supports pausing, allowing Salesforce to preserve the run's state so it can be resumed later. This definition is important for administration and troubleshooting because flow errors, paused-flow records, and debug details relate to individual flow interviews rather than to the flow definition as a whole. Salesforce's metadata and API documentation likewise uses the term FlowInterview for a flow's runtime instance. See the Salesforce FlowInterview object reference and Salesforce documentation on pausing flow interviews.

QUESTION NO: 15

Cloud Kicks is looking to build an orchestration that will most likely span across departments and roles. The lead architect advised the Administrator to monitor the running instances after they build out the orchestration.

What is a running instance of an orchestration called?

A. Session

B. Interview

C. Spool

D. Run

ANSWER: D

Explanation:

Run is the term Salesforce Flow Orchestration uses for an individual execution of an orchestration. Every time the orchestration is started—whether it is initiated by an autolaunched flow, a record-triggered flow, or another supported mechanism—Salesforce creates an orchestration run. A run represents the complete, end-to-end instance of the orchestration as it progresses through its stages and steps, including work assigned to users or queues across departments.

Administrators can monitor these instances from the Orchestration Runs page in Setup. This view provides operational visibility into a run's status and progress, and it helps identify runs that are waiting for assigned work, have completed, or encountered an error. Monitoring at the run level is especially important for cross-functional processes because a single orchestration can coordinate multiple human tasks and automated actions over time.

Salesforce's Flow Orchestration documentation describes an orchestration run as an instance of an orchestration and explains how to monitor runs in Setup. See Salesforce Help: Flow Orchestration and Salesforce Help: Monitor Orchestration Runs.

QUESTION NO: 16

In which two ways does Salesforce Flow for Service help customer service agent?

A. It shows a checklist that agents can print.

B. It allows an agent to pen a record and seamlessly resume a customer conversion.

C. It uses flows and quick action to walk agents through customer engagement.

D. It helps an experienced agent show a new agent what to do.

ANSWER: B C

Explanation:

Salesforce Flow for Service helps agents deliver consistent, efficient customer support by embedding guided automation directly in their Service Cloud workspace. It uses flows and quick action to walk agents through customer engagement because a screen flow can present step-by-step instructions, collect information, make decisions based on entered data, and update related records without requiring the agent to leave the case or customer context. Quick actions provide a convenient, contextual way to launch these guided interactions from a record page or console utility.

It also allows an agent to pen a record and seamlessly resume a customer conversion because flows can be designed to preserve progress and support resumed work when an interaction cannot be completed in one sitting. This is valuable for service processes that require follow-up, additional customer input, approvals, or work across multiple interactions. By guiding agents through standardized screens and automating record actions, Flow reduces manual steps while still allowing the agent to handle the conversation in the appropriate service context. See Salesforce's Flow documentation and Quick Actions documentation.

QUESTION NO: 17

Insightopia aims to automate accessing real-time transaction details stored in an external system. The transaction records are updated frequently but the volume of transactions stays the same over time.

Which could help Insightopia support this automation business requirement?

A. Flows, ETL (Extract, Transform, Load) process

B. Flow Orchestration, External Services

C. Flows, Salesforce Connect

D. Flow Orchestration, Local Action

ANSWER: C

Explanation:

Flows, Salesforce Connect is the appropriate combination because the requirement is to access transaction data that remains in an external system, changes frequently, and must be available in real time. Salesforce Connect exposes external-system data in Salesforce as external objects. Rather than copying and storing transaction records in Salesforce, external objects access the source system when users or automation query the data. This approach keeps the information current with the external source and avoids building a repeated import or synchronization process for a stable overall data volume.

Flow provides the process-automation layer. A flow can use the available external-object data as part of its logic and automate the associated business process, such as evaluating transaction information and carrying out follow-up actions in Salesforce. Together, these capabilities separate real-time data access from process automation: Salesforce Connect supplies the live external data, while Flow coordinates the business logic.

Salesforce describes Salesforce Connect as a way to access data stored outside Salesforce without copying it into the org, using external objects. See Salesforce Help: Salesforce Connect and Salesforce Help: Flow Builder.

QUESTION NO: 18

Which three of the following are the key component to build aprocess in Process Builder?

A. Action

B. Scheduler

C. Timer

D. Criteria Node

E. Trigger

ANSWER: A D E

Explanation:

In Process Builder, a process is structured around a start condition, one or more decision points, and the work that is performed after a decision evaluates as true. **Trigger** is the key starting component: it defines the object and specifies when the process begins, such as when a record is created or edited. **Criteria Node** is the decision component that evaluates record data and determines whether a particular branch of the process should proceed. A process can contain multiple criteria nodes, allowing separate business conditions to be evaluated in sequence. **Action** is the outcome component attached to qualifying criteria. Actions perform the automation work, such as creating a record, updating a related record, submitting for approval, posting to Chatter, invoking Apex, or launching a flow. Together, these components represent the fundamental Process Builder pattern: start the process from a record change, evaluate business criteria, and execute the required automated action. Salesforce has retired Process Builder for creating new automation in favor of Flow Builder, but these remain the core concepts for understanding and maintaining existing Process Builder processes. See Salesforce's Process Builder overview and criteria and actions documentation.

QUESTION NO: 19

Which two use cases is it appropriate to combine a process and a flow?

A. Clone a record and its children.

B. Post to an internal Chatter group.

C. Delete a related record.

D. Open up visibility of a Chatter feed item to Portal users.

ANSWER: A C

Explanation:

Clone a record and its children. is an appropriate use case because duplicating a parent record together with an arbitrary set of related child records requires querying collections, iterating through them, assigning new parent references, and creating the copied records. A flow supplies these collection and looping capabilities, while a process can serve as the record-change entry point and pass the relevant record ID into the flow.

Delete a related record. is also appropriate because deletion is a data operation that can be performed by flow logic after the process identifies that the triggering business condition has been met. The flow can receive the triggering record or its ID, locate the related record or records, and execute the needed delete operation. This combination was especially useful in legacy Process Builder designs when a simple record-trigger condition needed to initiate automation requiring more advanced flow elements. Salesforce now recommends building new record-triggered automation in Flow, but understanding this process-to-flow pattern remains relevant when maintaining existing automations. See Salesforce's Flow Basics and Process Builder overview.