PassQueen
a.k.a dumpsqueen.com

Pega Certified Senior System Architect (PCSSA) 87V1

Pegasystems PEGAPCSSA87V1

Version Demo

Total Demo Questions: 14

Total Premium Questions: 145

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Application Design	17
Topic 2, Case Management	28
Topic 3, Data Modeling	13
Topic 4, Data Integration	13
Topic 5, User Experience	20
Topic 6, Security	18
Topic 7, Reporting	9
Topic 8, Background Processing	5
Topic 9, DevOps	22
Total	145

QUESTION NO: 1

Which three items are generated and displayed in the compliance score of an application? (Choose three.)

A. Rules with warnings in the application

B. Pega Platform core rules used in the application

C. Rules with unjustified warnings in the application

D. Weighted compliance score

E. Percentage of compliant rules in the application

ANSWER: A C D

Explanation:

The application compliance score is a Guardrails-oriented assessment that Pega generates from the rules in the application. It displays the rules with warnings in the application so that development teams can identify rule instances that require review or improvement. This visibility supports targeted remediation by linking the score to concrete rule-level findings rather than presenting only an aggregate result.

It also displays rules with unjustified warnings in the application. A warning can be justified when the team records an appropriate rationale for retaining the implementation; separately identifying unjustified warnings helps teams focus on findings that have not yet been assessed or accepted with a documented business or technical reason.

The score itself is a weighted compliance score. Pega weights Guardrail warnings according to their relative severity or impact, so the result reflects more than the simple number of warnings. This weighted measurement gives teams a more meaningful indication of application quality and maintainability and helps prioritize remediation work. Pega's Guardrail compliance tooling is intended to be reviewed throughout development and before application release. See Pega Academy: Guardrail compliance and Pega Documentation.

QUESTION NO: 2

An assignment service-level agreement (SLA) is configured with the following details:

â™¦ Initial urgency: 20

â™¦ Assignment ready: Timed delay of 1 hour

â™¦ Goal: 5 hours and increase urgency by 10

â™¦ Deadline: 8 hours and increase urgency by 20

â™¦ Passed deadline: 2 hours, increase urgency by 20, and limit events to 5

Assuming no other urgency adjustments, what is the assignment urgency 16 hours after the case reaches the assignment?

A. 100

B. 90

C. 130

D. 70

ANSWER: C

Explanation:

130 is correct. The assignment begins with an initial urgency of 20. The assignment-ready delay controls when the assignment becomes available for processing; it does not add an urgency adjustment. At the five-hour goal, the SLA adds 10 urgency, bringing the total to 30. At the eight-hour deadline, it adds another 20, bringing urgency to 50.

After the deadline, the passed-deadline interval occurs every two hours. By 16 hours after the assignment is reached, passed-deadline processing occurs at hours 10, 12, 14, and 16. Each event increases urgency by 20, for a total additional increase of 80. The passed-deadline event limit is five, and only four such events have occurred by this point, so the limit does not prevent any of these increases. Therefore, the final calculation is 20 + 10 + 20 + (4 × 20) = 130.

Pega SLAs use goal, deadline, and recurring passed-deadline events to update assignment urgency and drive escalation behavior. See Pega's Service-level agreements learning topic and Service-level agreements documentation.

QUESTION NO: 3

While running an application, a user notes that a system exception occurred. Which log do you review to identify the issue?

A. Pega

B. Alert

C. AlertSecunty

D. BIX

ANSWER: A

Explanation:

Pega is correct because Pega Platform records application runtime exceptions, Java stack traces, rule-processing failures, and related diagnostic details in the primary Pega log, commonly named `PegaRULES`. This is the log that a system architect or administrator reviews first when a user reports a system exception during application processing. The exception entry provides the timestamp, requestor or operator context where available, affected rules or activities, the exception message, and a stack trace. Those details allow the team to correlate the reported failure with the processing path and identify the underlying configuration, rule, integration, or platform issue. Log files are available through Pega Platform administrative tooling and on the server hosting the node; the exact file location and rolling-file naming can vary by deployment and logging configuration. Pega documentation describes log files as the operational diagnostic source for errors and exceptions, while the primary Pega log captures general platform and application processing events. See the Pega documentation portal's Platform documentation and Pega Academy's System administration learning resources for current guidance on viewing and configuring logs.

QUESTION NO: 4

A requirement states that when a case is assigned to a user for review, its work status is set to Open-Review. Which two steps do you perform to implement this requirement? (Choose Two)

A. Configure a data transform to apply the Open-Review status on the appropriate assignments.

B. Add Open-Review as an allowed status on the case type record.

C. Create a field value record for the Open-Review status.

D. Apply the Open-Review status to the appropriate assignments.

ANSWER: B D

Explanation:

To use **Open-Review** as a case status, first add it as an allowed status on the case type record. Case status configuration defines the status values that the case can use throughout its life cycle. Adding the status at the case-type level makes **Open-Review** available for selection in the case workflow and maintains a consistent, governed status model for the case.

Next, apply **Open-Review** to the appropriate assignments. An assignment represents work routed to a user or work queue, and its configuration can set the case status when the flow reaches that assignment. Configuring the review assignment with this status ensures that the case automatically changes to **Open-Review** at the point it is assigned for review, without requiring the reviewer to update the status manually. This directly aligns the status with the assignment's business purpose and provides accurate reporting and visibility into work currently awaiting review. Pega's case-management training describes configuring case statuses and using them in a case life cycle: Pega Academy: Case status.

QUESTION NO: 5

An external partner invokes a REST service to create cases in an application. The partner must authenticate before the service can access application rules.

Which rule configures the authentication and access group used by the service?

A. Service package

B. Authentication profile

C. REST connector

D. Node type

ANSWER: A

Explanation:

A service package configures the authentication and authorization context for services. The service package specifies the authentication type and can identify the access group under which authenticated service requests run. The access group establishes the application context and the access roles that control which rules and data the service can use.

The REST service rule defines the service endpoint and methods, while the service package provides shared security and runtime configuration for the services in that package. See Pega documentation on REST services.

QUESTION NO: 6

How do you ensure that only a manager can run a specific flow action?

A. Add a pre-processing activity to the flow action form to determine whether the user is a manager.

B. Add a privilege to the flow action form, then add the privilege to a role assigned to the access group for managers.

C. Add an Access When condition to the flow action to determine whether the user is a manager.

D. Add a validate rule to the flow action form to determine whether the user is a manager.

ANSWER: B

Explanation:

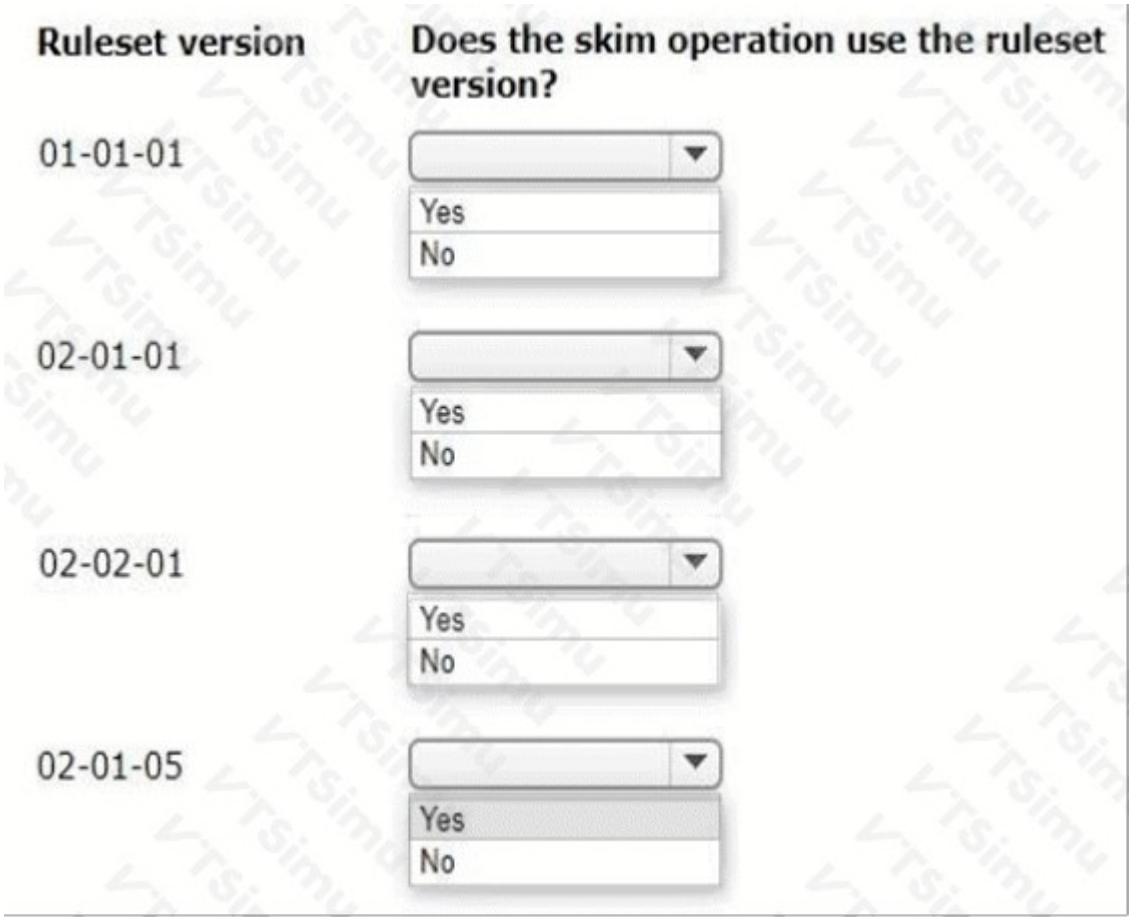
Add a privilege to the flow action form, then add the privilege to a role assigned to the access group for managers. This is the appropriate Pega authorization design because a privilege is a named capability that can be required before a user performs a protected action. Configure the flow action to require that privilege, and grant the privilege through an access role that is included in the managers' access group. At run time, Pega evaluates the operator's effective access roles and privileges; only an operator whose access group provides the required privilege can execute the flow action.

This approach separates authorization policy from flow-processing logic. The flow action remains reusable, while the access group and role configuration centrally controls which population is allowed to use it. It also supports maintainable administration: when a person becomes or ceases to be a manager, assigning the appropriate access group changes the person's available capabilities without modifying the flow action or application rules. Pega's access-control model uses access groups, access roles, and privileges together to enforce such feature-level permissions. See Pega Academy's Access control topic and the Pega documentation on privileges for the role-based authorization model and privilege usage.

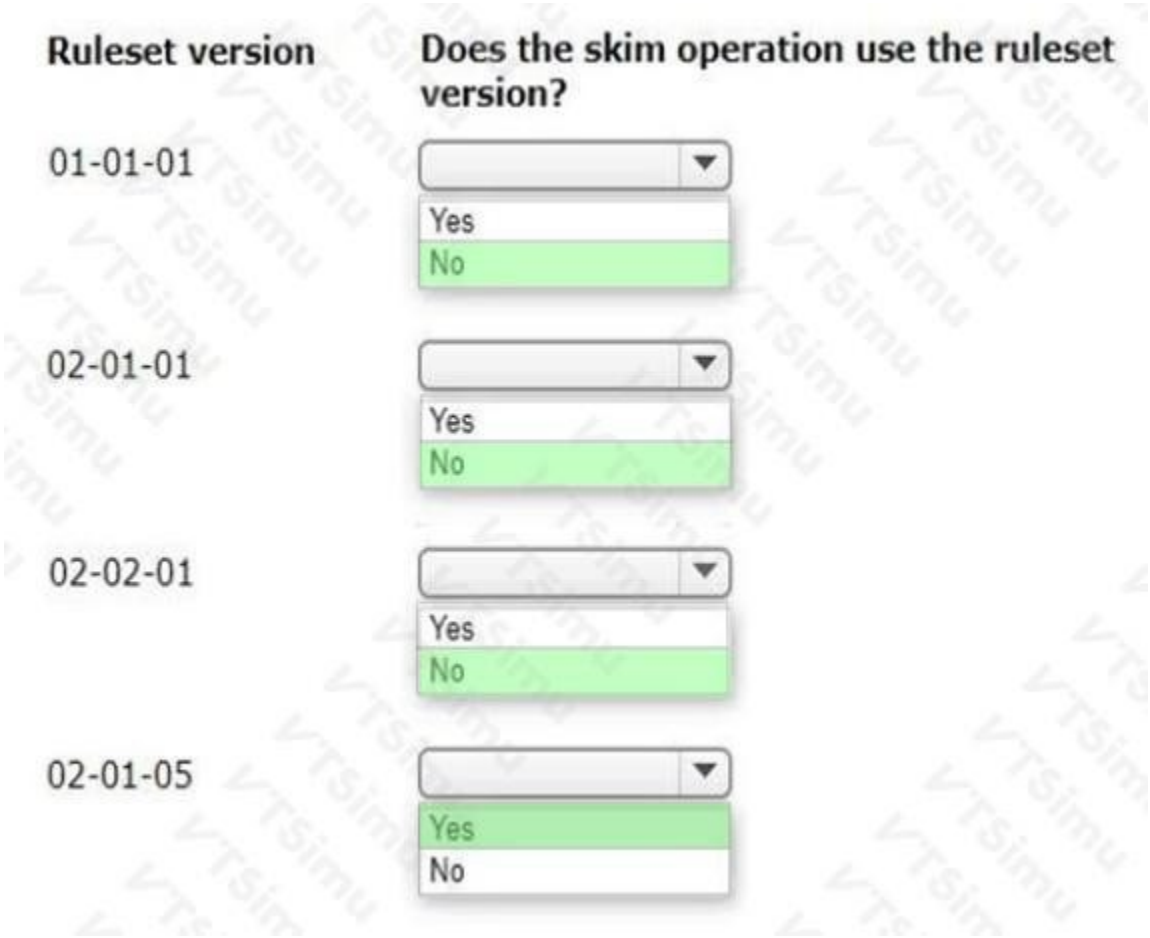
QUESTION NO: 7 - (HOTSPOT)

You perform a major skim on the ruleset ABC:02-02 to ABC:03-01.

In the Answer area, select the ruleset versions that the skim operation uses.



ANSWER:



Explanation:

A major skim creates a new major ruleset version by collecting the effective rules from the source major version. Here, the source is ABC:02-02 and the destination is ABC:03-01. The source major version is therefore version 02. Pega considers the rules available in that major version across its applicable minor and patch versions, so the skim uses ruleset versions 02-01-01, 02-01-05, and 02-02-01. This lets the new 03-01 version contain the resolved rule content from the complete version 02 ruleset lineage, including rules saved in the earlier 02-01 minor version as well as rules in 02-02.

The version 01-01-01 is not part of source major version 02. A major skim does not pull that earlier major-version branch into the new ruleset version. Accordingly, the marker should select Yes beside 02-01-01, 02-02-01, and 02-01-05, and select No beside 01-01-01. This preserves the expected major-version boundary while creating ABC:03-01 from the complete contents of ABC version 02. Pega ruleset versioning uses major, minor, and patch components to organize rule history and control which rule instances are available during rule resolution. See the Pega Academy material on ruleset versioning for the versioning model used when managing and advancing ruleset versions.

QUESTION NO: 8

A case type uses a child case to obtain an approval from each member of an approval committee.

Which two configurations support processing the approvals in parallel? (Choose Two)

A. Configure a Split for Each shape based on a Page List of committee members.

B. Configure the parent process to resume after all subprocesses complete.

C. Configure a single assignment that is routed to all committee members.

D. Configure a wait shape for each committee member.

ANSWER: A B

Explanation:

Configure a Split for Each shape based on a Page List of committee members. The shape creates a subprocess for each embedded page in the Page List, allowing independent approval processing for every committee member.

Configure the parent process to resume after all subprocesses complete. This join behavior ensures that the parent case waits until every required approval subprocess has finished before it proceeds to the next step. See Pega documentation on parallel case processing.

QUESTION NO: 9

An application must create a case for each valid row in a file that is uploaded nightly. Invalid rows must be recorded for review without preventing valid rows from being processed.

Which Pega capability is most appropriate?

A. Data flow

B. Declare expression

C. Correspondence rule

D. Access control policy

ANSWER: A

Explanation:

Data flow is correct because a data flow is designed to process a stream or collection of data records. The data flow can read the uploaded file, evaluate each row, create cases for valid records, and route invalid records to an error-handling destination or data set for later review.

This design supports scalable, record-by-record processing and separates invalid-record handling from successful case creation. See Pega documentation on data flows.

QUESTION NO: 10 - (SIMULATION)

You are creating a report that lists all open Personal Injury cases with related Auto Claim case information. Personal Injury and Auto Claim cases belong to the same class group.

The report includes three columns:

♦ Case ID for the Personal Injury case

♦ Case ID for the parent Auto Claim case

♦ Claim adjuster for the Auto Claim case

Personal Injury case data includes the Auto Claim case ID.

What join type do you use to configure the report?

ANSWER: See the explanation for the answer

Explanation:

Use a Class Join.

Configure the Personal Injury case class as the report's primary class and add the Auto Claim case class as a **Class Join**. A class join is appropriate because both case types are in the same class group.

Join the Personal Injury property that stores the Auto Claim case ID to the Auto Claim case `pyID` (Case ID).

Select the Personal Injury `pyID`, the joined Auto Claim `pyID`, and the joined Auto Claim claim-adjuster property as report columns.
Apply the filter for open Personal Injury cases.

The class join lets the report retrieve the related Auto Claim case data using the Auto Claim ID stored on each Personal Injury case.

QUESTION NO: 11

A data type stores reference information about product categories. Business users must be able to add and update categories in production without a deployment.

Which configuration supports this requirement?

A. Configure the data type as a local data source and delegate the data records.

B. Save each product category as a circumstance of a flow rule.

C. Create a separate production ruleset for every product category.

D. Store each product category as a field value in a locked ruleset.

ANSWER: A

Explanation:

Configure the data type as a local data source and delegate the data records. A local data source stores data records in Pega, and delegation can provide authorized business users with controlled access to maintain those records in production. Product categories are reference data, so maintaining them as data records is more appropriate than changing application rules for each ordinary update.

Delegation should be configured for the appropriate access group and with suitable authorization so that only permitted users can make changes. See Pega documentation on data types and Pega Academy: Delegating rules.

QUESTION NO: 12

Which two configurations can you use to include access groups into your Product rule? (Choose two.)

A. Export the access groups using the Access Manager.

B. Add the access groups to the application record.

C. Add the access group to the Product record in the Individual instances to include section.

D. Associate the access groups with a ruleset included in the application.

ANSWER: B C

Explanation:

Access groups are data instances, so a Product rule can package them explicitly by listing each required access-group instance in the *Individual instances to include* section. This is useful when the product needs a specific access group, including its application association, role list, and other configuration, without relying on the access group being collected indirectly through another included rule.

Access groups can also be included through the application definition. When access groups are added to the application record and that application is included in the Product rule, Pega packages the application's associated access-group configuration as part of the application content. This approach is generally appropriate for packaging a complete application and its normal security configuration together.

These two methods support both targeted packaging of individual access-group instances and application-driven packaging. During import, the target system receives the required access-group data so that operators can be assigned to the application with the intended authorization context. See Pega documentation on Product rules and the Pega Academy material on application rules.

QUESTION NO: 13

Which three mobile device features can you leverage by using Pega Mobile Client? (Choose Three)

A. Barcode scanning

B. Text messages

C. Biometric identification

D. Push notifications

E. Phone calls

ANSWER: A C D

Explanation:

Pega Mobile Client packages a Pega application as a native mobile app and exposes supported device capabilities to the application. **Barcode scanning** is supported through the device camera, allowing a case workflow to capture a barcode value without requiring users to enter it manually. **Biometric identification** can be used to authenticate or reauthenticate a user with the device's biometric mechanism, such as fingerprint or facial recognition, helping provide a secure and convenient sign-in experience. **Push notifications** allow the mobile app to alert users about assignments, case updates, and other relevant events even when the app is not actively open. These capabilities are particularly useful for field and task-oriented applications, where rapid data capture, secure access, and timely notification of new work are essential. Pega Mobile Client is designed to bridge Pega application functionality with native mobile-device services while retaining Pega's case management and user-interface behavior. See Pega's Pega Mobile Client documentation and the Pega Academy mobile topic for mobile capability and configuration guidance.

QUESTION NO: 14 - (DRAG DROP)

Drag each security solution on the left to its appropriate use case.

ANSWER:

Explanation:

The completed mapping correctly applies Pega's different authorization mechanisms at the appropriate level. **Access when** is the appropriate security solution for preventing dealers from modifying orders in the Fulfillment stage because it evaluates a Boolean condition at run time. The condition can use the order's current stage or status and allow the modification action only while that condition is satisfied. This makes the authorization responsive to the life-cycle state of the case or record.

Access Deny correctly addresses the requirement that dealers cannot run reports. Pega access-deny rules provide an explicit restriction for an operator's access, ensuring that the reporting capability is unavailable to the affected users even if broader access might otherwise be present. This is useful for enforcing a firm security boundary around a protected feature.

Access Control Policy is correct for allowing customers to view only their own credit report. Access control policies support record-level authorization by evaluating a policy condition against the requestor and the data being accessed. A policy can compare the customer identity associated with the credit-report record to the logged-in customer, so access is limited to records that belong to that customer. Pega describes this approach in its Access control policies documentation.

Access of Role to Object correctly permits dealers to modify evaluations because it associates an access role with the evaluation object and its allowed operations. The role-based configuration grants the required update privilege to authorized dealers in a consistent, reusable way. This aligns with Pega's role-based access model, described in the role-based access control documentation.