DUMPSBOSS.

Pega Certified Senior System Architect (PCSSA) 86V1

Pegasystems PEGAPCSSA86V1

Version Demo

Total Demo Questions: 12

Total Premium Questions: 121

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Application Design	30
Topic 2, Case Management	23
Topic 3, Data Modeling	14
Topic 4, User Experience	10
Topic 5, Security	17
Topic 6, Integration	14
Topic 7, Background Processing	5
Topic 8, Reporting	8
Total	121

QUESTION NO: 1

In a hiring case type, only HR Managers should have read access to a candidate's reference letter attachment. Which two tasks are used to fulfill this requirement? (Choose Two)

A. Configure read access for a user according to a privilege

B. Restrict read access according to a When condition

C. Configure attribute-based access control on the Attachment Category

D. Define a separate Attachment Category for HR Managers

ANSWER: A D

Explanation:

Configure read access for a user according to a privilege and **Define a separate Attachment Category for HR Managers** work together to secure the reference letter at the attachment level. In Pega, an attachment category classifies attachments and can be configured with security settings, including the privilege required to perform actions such as viewing or downloading attachments in that category. Creating a dedicated category for reference letters isolates these sensitive files from ordinary hiring-case attachments and gives the application a specific policy point for their access control.

The required privilege is then granted only to the access role associated with HR Managers. At run time, Pega evaluates whether the current operator has that privilege before permitting read access to an attachment in the protected category. This approach is appropriate because the requirement concerns a particular attachment type, rather than general access to the complete hiring case. It also makes the rule reusable and maintainable: future sensitive document types can use their own categories and privilege policies without changing the overall case-access model. See Pega documentation on attachments in case management and access control.

QUESTION NO: 2

Which two rule types can you mark as a relevant record? (Choose Two)

A. Property

B. Report

C. Skin

D. Section

ANSWER: A D

Explanation:

Property and **Section** can be marked as relevant records. Relevant records provide a curated set of Dev Studio rules for use in App Studio, allowing low-code application developers to select approved, reusable implementation assets without having to locate them through the full rule hierarchy. A property is a supported relevant-record type because it represents a reusable element of the application data model. Marking a property as relevant makes it available where App Studio users configure data-related behavior. A section is also supported because it is a reusable user-interface rule; making it relevant enables application builders to add an approved section to views while maintaining the implementation defined by the system architect. This mechanism helps make App Studio configuration simpler and safer while supporting reuse and governance of rules created in Dev Studio. Pega documents relevant records as rules that developers explicitly expose for use by application developers, including supported data-model and UI rule types. See Pega's documentation on creating relevant records and its Relevant records learning topic.

QUESTION NO: 3

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. In Pega, a privilege is the standard authorization mechanism for restricting access to specific capabilities, including flow actions. Configuring the privilege on the flow action makes Pega evaluate whether the current operator has that privilege before the action is available for use. The privilege is then granted through an access role, and that access role is included in the access group used by manager operators. This provides declarative, role-based security: authorization is based on the operator's current access group and associated roles rather than on custom procedural logic in the flow action. It also centralizes administration, because access can be granted or withdrawn by changing role and access-group configuration without modifying the flow or adding implementation-specific checks. Privilege-based access control is appropriate when the requirement is to ensure that only users in an authorized business role can perform an operation. Pega documents privileges as access-control records that can be associated with rules and granted to access roles. See Pega Platform documentation on privileges and Pega Platform documentation on access groups.

QUESTION NO: 4 - (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 I

D.

What join type do you use to configure the report?

ANSWER: See the explanation for the answer

Explanation:

Use a Class join.

Configure the report with Personal Injury as the primary class and add the Auto Claim case class as a **Class join**. This join type 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` property.
Select the Personal Injury `.pyID`, the joined Auto Claim `.pyID`, and the joined Auto Claim adjuster property as report columns.
Filter the Personal Injury cases to open cases.

A class join is used rather than an association join because the relationship is identified by the parent Auto Claim case ID and both classes share the same class group.

QUESTION NO: 5

An application must route an assignment to the manager of the operator who created the case.

Which routing option do you configure?

A. To the work party manager

B. To the current operator

C. To a work queue

D. To a specific operator

ANSWER: A

Explanation:

To the work party manager is correct because this routing approach identifies the manager associated with an operator work party and routes the assignment to that manager. In this scenario, the case creator is represented by the appropriate operator work party, allowing Pega to determine the manager dynamically rather than requiring a fixed operator ID in the process.

This routing configuration supports organizational changes because manager information is maintained with the operator and organization configuration. When the creator changes, the assignment is routed according to the current manager relationship. See Pega Academy: Routing work.

QUESTION NO: 6 - (HOTSPOT)

The current version of an application lists ABC:02-02 as an application ruleset. The ruleset versions listed in the Answer Area are also present on the system.

In the Answer Area, identify the ruleset versions that the skim operation considers when you perform a major version ruleset skim on ruleset ABC.



ANSWER:



Explanation:

A major-version skim for the ABC ruleset uses the application's ruleset version entry, ABC:02-02, to determine the relevant version family. The application can resolve rules from versions in major version 02 up to the specified minor version, including patch versions. Consequently, the skim considers ABC:02-01-01 and ABC:02-01-05 because they are patch versions under the earlier 02-01 minor version, as well as ABC:02-02-01 because it is a patch version under the application's 02-02 minor version. These are all part of the major version 02 ruleset lineage that is available to the application and therefore belongs in the set evaluated by the major skim operation.

The purpose of this skim is to bring the eligible rules from the applicable major-version lineage into the newly created skimmed ruleset version while preserving the ruleset hierarchy and rule-resolution behavior associated with the application. Patch numbers do not exclude a ruleset version from consideration; they identify versions within the applicable minor-version branch. The version 01-01-01 is outside the application's major version 02 lineage, so it is not part of the versions considered for this major-version skim. Pega's ruleset and versioning concepts are described in the Pega Platform rules documentation.

QUESTION NO: 7

Which two statements are true about a data type in Pega Platform? (Choose Two)

A. A data type defines a reusable business data structure.

B. A data type can be configured with a system of record.

C. A data type can be used only by one case type.

D. A data type automatically creates a separate ruleset for each field.

ANSWER: A B

Explanation:

A data type defines a reusable business data structure, including the fields that describe an entity such as a customer, product, or location. Application developers can reference that structure from case types, data pages, views, and other rules that require the same business information.

A data type can be configured with a system of record. The system of record identifies where the data is sourced and maintained, such as a Pega data store or an external system accessed through a connector. This supports a consistent data-access design across the application. See Pega Academy: Data types and Pega Platform documentation.

QUESTION NO: 8

A class group is used to_____.

A. identify a layer of the Enterprise Class Structure

B. combine related properties by usage

C. assign work to a specific group of users

D. associate child classes with a single database table

ANSWER: D

Explanation:

A class group is used to associate multiple concrete classes, typically a parent class and its descendant work or data classes, with one physical database table. In Pega, the class group defines the shared persistence mapping: instances of the member classes are stored in the table specified by the class group rather than requiring a separate table for each class. This is particularly common for work classes, where related case types can share a table while retaining their individual class identities and rules. The class group configuration also identifies which class is used for history instances, supporting consistent persistence for the grouped classes. Sharing a table in this way simplifies the database schema and lets Pega distinguish the stored instances through their class metadata. The Pega data-model guidance describes class groups as the mechanism for mapping classes to database tables and managing shared storage arrangements. See the official Pega Academy Data model topic and the Pega Platform documentation on class groups for details on configuring class-to-table mappings.

QUESTION NO: 9

A company with multiple applications spanning different business units wants to send a standard confirmation email to customers whenever a case is created.

Which is the appropriate layer of the Enterprise Class Structure (ECS) for the email correspondence?

A. Unit

B. Organization

C. Implementation

D. Division

ANSWER: B

Explanation:

Organization is the appropriate ECS layer because the confirmation email is a common capability required by multiple applications across different business units. In Pega's Enterprise Class Structure, rules defined at the organization layer are intended for enterprise-wide reuse. A correspondence rule placed there can be inherited by the classes for the organization's divisions, units, and implementation applications, allowing every relevant application to use one centrally maintained email definition.

Keeping the standard confirmation correspondence at this layer also supports consistent customer communications. For example, shared branding, common wording, sender details, and standard delivery behavior can be updated once and made available throughout the enterprise hierarchy. An implementation can still specialize the inherited correspondence when a genuine application-specific requirement exists, while the shared organization-level rule remains the reusable baseline. This approach follows ECS's purpose of locating a rule at the highest layer where it is broadly applicable, maximizing reuse and reducing duplicate rule maintenance. See Pega Academy's Enterprise Class Structure material and the Pega documentation on Enterprise Class Structure.

QUESTION NO: 10 - (DRAG DROP)

In the Answer Area, drag each log name on the left to its intended purpose on the right.



ANSWER:



Explanation:

The completed mapping is correct because each selected log corresponds to the type of operational information that Pega Platform writes to that log. The **Pega** log is the primary application log for internal platform activity. It records informational messages as well as warnings and errors generated during normal request processing and internal operations. This makes it the appropriate destination for the purpose describing warnings, errors, and information about internal operations.

The **AlertSecurity** log is specifically intended for security alert events, including attempts to manipulate or tamper with request URLs. Recording these events separately supports security monitoring and investigation, because administrators can review suspected URL tampering without mixing those events into general operational logging.

The **Alert** log is dedicated to performance-related alert messages. Pega generates alerts when monitored operations exceed configured thresholds or exhibit behavior that may require performance analysis. These messages help teams identify slow interactions, database activity, requestor behavior, and other conditions relevant to application performance and tuning.

This separation of general operational logging, security alerts, and performance alerts lets administrators filter and investigate system activity efficiently. For additional detail on Pega Platform logging and alerting, see Pega documentation on logging and Pega documentation on alerts.

QUESTION NO: 11

Which two configurations are required when adding a Split for Each shape to your case life cycle? (Choose Two)

A. Enter a rule that stores the audit note that you want to include in the subprocess

B. Define when the parent process resumes processing

C. Enter the Page List or Page Group property that is the basis of the split

D. Enable users to get back to the subprocess after the case moves forward

ANSWER: B C

Explanation:

A Split for Each shape creates a separate subprocess instance for every item in a collection, enabling work to proceed in parallel for each embedded page. Therefore, you must enter the Page List or Page Group property that is the basis of the split. This property supplies the collection that Pega iterates over, with each page serving as the context for one spawned subprocess.

You must also define when the parent process resumes processing. The parent flow needs a join condition that establishes whether it continues after every subprocess completes, after any subprocess completes, or according to the configured completion behavior. This synchronization setting is essential because it governs how the parallel child work reconnects to the primary case flow and ensures that subsequent case processing occurs at the intended point.

These configurations define both the workload to distribute and the completion behavior that controls the parent case's progression. Pega documentation describes Split for Each as a mechanism for processing each item in a Page List or Page Group through parallel subprocesses, with configuration for resuming the parent flow. See Pega Academy: Flow shapes and Pega Platform documentation: Flow shapes.

QUESTION NO: 12

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

♦ Initial urgency: 20

♦ Assignment ready: Timed delay of 1 hour

♦ Goal: 5 hours and increase urgency by 10

♦ Deadline: 8 hours and increase urgency by 20

♦ 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 its configured initial urgency of 20. The timed Assignment ready setting controls when the assignment becomes available for processing; it does not add an urgency adjustment in this configuration. At the goal interval, the SLA increases urgency by 10, resulting in an urgency of 30. At the deadline interval, it adds another 20, raising urgency to 50.

After the deadline, the passed-deadline escalation runs every 2 hours. By 16 hours after the assignment is reached, passed-deadline processing has occurred four times: at 10, 12, 14, and 16 hours. Each event adds 20 urgency, for a total passed-deadline increase of 80. The configured limit of 5 events has not yet been reached. Therefore, the final urgency is 20 initial urgency + 10 goal adjustment + 20 deadline adjustment + 80 from four passed-deadline events, which equals 130.

Pega service-level agreements use goal, deadline, and passed-deadline intervals to automate escalation and adjust assignment urgency. See Pega Academy: Service-level agreements and Pega Documentation: Service-level agreements.