**DECLARATIVE NETWORK**

**Declare Expression rule**

Use Declare Expression rules to define automatic computations of property values based on expressions.

We can create Declare Expression rule instead of using a property set method in an Activity.

When you include Declare Expression rules in calculations, you increase automation and efficiently reuse resources.

two methods of calculating values:

**1.Forward chaining**

Forward chaining occurs when the target property changes because the input values in the calculation change.

For example: online shop by selecting items that sum up to $60. A shopping application calculates the tax value that is 10% of the order amount and updates the target property with the value of $6. When the customer adds another item to the shopping cart, the application adds the price of the new item to the calculation. Consequently, input values change.

Consider a scenario in which customers provide width and length of a room to calculate the room area to order a correct number of floor tiles. Each time a customer changes the width or length of the room, the shopping application automatically updates the room area by performing forward chaining.

**Backward chaining**

Backward chaining occurs when an application seeks values of input parameters to calculate a target property.

For example, in an online shopping application, when a customer provides a code to get a discount on specific items in a shopping order, the application seeks properties that hold prices of the selected items, and then updates the total order amount.

In backward chaining, target values remain unchanged if input values change, and the application performs calculation only after receiving a call for a target property.

**Creating Declare Expression rule**

1.Dev Studio, click CreateDecisionDeclare Expression.

2.In the Target Property field, enter a property that you want to calculate, preceding the property name with a period.

For example: For the TaxValue property, enter .TaxValue.

3.Optional: To restrict search for the result of the declare expression to a specific page, in the Page Context field, enter a data page that stores available results.

For example: To situate a target property on the .pyItemNames() embedded page, enter .Invoices.pyOrders().pyItems().pyItemNames(). Leave the parentheses blank because a declare expression applies to all elements on a page.

For more information, see Data pages overview.

In the Context section, select the application layer in which you want to store the rule.

In the Apply to field, enter the class that stores the target property.

Note: When you create a declare expression, you can first provide the class, and then provide the target property in the Target Property field. If you select a class first, the system limits the available target properties to the properties that the selected class stores.

In the Add to ruleset field, enter a ruleset to store the rule

Click Create and open.

**Questions related to the declarative rules:**

what is declare expression

name some declarative rules

use of page context

rule declare On change how to implement

Declare triggers

Contrains

diff b/w Constraina and validation rules

Declare index how to implement

**How many declarative rules in pega**

**6**

1. Declare expression
2. Declare Onchnge
3. Declare Trigger
4. Declare index
5. Declare contrains
6. Declare pages

**Declarative rule**

* it is an instance of a class derived from the **Rule- Declare**- class
* when the value of property is invoked in any of these declarative rules
* They allow for automatic processing of Property values.
* No need to be called explicitly
* No need to run sequentially.
* The system manages re-evaluation when it detects a change.

**What is the use of PageContext?**

* Page Context is used if the target property is in PageList or PageGroup.
* If target property has mode single value and appears directly on the page of applied to a class, PageContext should be blank.
* This called Context-Free expression

**Explain about Rule-Declare Onchange?**

* Runs an activity automatically when the value of a specified property change.
* It is always Forward Chaining.
* If we specified more than one property, they all must be on the same page.
* If a *Declare OnChange* rule is used to suspend work, the when condition is called a business exception.

**How to implement Rule-Declare-Onchange?**

Ans:

1. Create an instance of Rule-declare-onchange
2. Select the properties we want this rule to keep a track of.
3. Optionally enter a when condition rule to be evaluated.
4. Provide the activity name of type OnChange.

**Explain about Declare Triggers?**

Declare Trigger runs an activity when instances of a specific class are created, updated, or deleted in the Database.

Declare Trigger is always Forward Chaining.

**Explain about Constraints?**

* Constraints are used for validation purpose.
* Define and enforce the relationship among property values.
* Prevent the user from advancing if the constraint is not satisfied.
* Prevent invalid data entered. Constrains is always Forward Chaining.

**What is the difference between Constraints and Validation Rules?**

Constraints are declarative rules triggered automatically when a property value changes. Validation rules must be told when to run When a form is submitted or Before running a Flow Action.

**Explain about Declare Index?'**

Declare Index rules are instances of the Rule-Declare-Index class. Declare Index Automatically extracts a value from an embedded property when its value has changed. They are part of the sysadmin category.

**How do we implement the Declare Index?**

Step 1: Create a class inherits from the Index- class. Provide these three properties: pxInsIndexedKey, pxIndexCount, pxIndexPurpose.

Step 2: Create Rule-Declare-Index And provide Source Page Context and Source page context class, Index class to write.

Step 3: In list view, Join tab provide Declare Index name.