**Salesforce Answers**

**Q.1. How many actions are there in process builder? What are they?**

Salesforce Process Builder is basically an automated tool that allows you to control the order of actions or evaluate the criteria for a record. It has eight actions associated with it:

* **Creating Records**: This will allow you to create a new record and add different field values for it.
* **Updating Records**: You can update one or more records that are somehow related to the record that started the process. This can be done either by manually entering the records or by using the records from a related record.
* **Posting on Chatter**: Process Builder in Salesforce allows you to post on Chatter for sharing any information with any user.
* **Quick Action**: If you already have global actions or objects, within Salesforce, then you can use quick action to use them in any record.
* **Launching or Triggering the Flow**: It is possible for you to launch the flow from your process to automate different processes.
* **Submitting the Record Automatically for Approval**: The record that started your process can be submitted. Any other record cannot be automatically submitted.
* **Calling or Triggering Apex Code**: You can invoke an Apex code that you have already written in Salesforce.
* **Invoking Another Process**: This action will actually invoke a process to another process.

**Q.2. Differences between Process Builder Flow and Work Flow?**

Ans: 1) Process builder is mainly used to avoid use of multiple workflow because one workflow can have only one condition and criteria action but one process builder can have multiple if/else condition.

2) Process builder allow you to update child record where as worflow doesnot.

3) Process builder can call your apex code so it can send email alerts, submit records for approvals using apex code whereas workflow cannot call apex but can send email/ outbound alert only without code.

**3. What is Salesforce & What is Visualforce?**

**Salesforce** is the world’s #1 customer relationship management (CRM) platform. We help your marketing, sales, commerce, service and IT teams work as one from anywhere.

Salesforce services allow businesses to use cloud technology to better connect with partners, customers, and potential customers.

**Visualforce** is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Lightning platform.

**4. Differences between Role & Profile?**

|  |  |
| --- | --- |
| **Role** | **Profile** |
| Roles provide access to records visibility for the users. | Profiles provide access control of CRED(create, read, edit, delete) records of the users. |
| It is basically a record level access. | It is basically an object and field level access. |
| Roles control access to records and fields only. | Profile control access to Objects, field-level security, page layouts, record types, and apps. |

**Q.3. What is trigger & what is the types of trigger? What are the events associated with trigger?**

Triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

There are two types of triggers:

1. Before triggers are used to update or validate record values before they’re saved to the database.
2. After triggers are used to access field values that are set by the system (such as a record's Id or LastModifiedDate field), and to affect changes in other records, such as logging into an audit table or firing asynchronous events with a queue. The records that fire the after trigger are read-only.

A trigger is Apex code that executes before or after the following types of operations:

* insert
* update
* delete
* merge
* upsert
* undelete

**Q. 4. What are sharing rules in Salesforce?**

Sharing rules can be based on who owns the record or on the values of fields in the record. For example, use sharing rules to extend sharing access to users in public groups or roles. As with role hierarchies, sharing rules can never be stricter than your org-wide default settings.

**Differences between trigger.new & trigger.old?**

**Trigger.new** : Returns a list of the new versions of the sObject records. Note that this sObject list is only available in insert and update triggers, and the records can only be modified in before triggers.

**Trigger.old** : Returns a list of the old versions of the sObject records. Note that this sObject list is only available in update and delete triggers.