

## Salesforce Flow: A Brief Overview

Salesforce Flow is an automation tool in Salesforce that allows you to build complex business processes with a declarative, low-code approach. It enables users to collect, update, and process data across Salesforce objects using a visual interface.

### Types of Flows in Salesforce

1. **Screen Flow** – Requires user interaction with a UI.
2. **Schedule-Triggered Flow** – Runs at a scheduled time.
3. **Record-Triggered Flow** – Runs automatically when a record is created, updated, or deleted.
4. **Autolaunched Flow** – Runs automatically via Apex, Process Builder, or other automation tools.
5. **Platform Event-Triggered Flow** – Executes when a platform event occurs.

### Key Components of a Flow

- **Elements:** Actions, Assignments, Decisions, Screens, Loops, and Subflows.
- **Resources:** Variables, Constants, Formulas, and Collections.
- **Connectors:** Control the path between elements.

### Why Use Flows?

- Automate repetitive tasks
- Reduce the need for Apex triggers
- Provide guided UI for users
- Ensure data consistency and validation

### Steps to Use Salesforce Flow

1. **Navigate to Flow Builder:**
  - Go to **Setup** → Search for **Flows** → Click **New Flow**.
2. **Select the Flow Type** (Based on Requirement, See Below).
3. **Add Elements:**
  - Use elements like **Screen**, **Assignment**, **Decision**, **Loops**, **Actions**, etc.
4. **Configure Logic & Resources:**
  - Define variables, formulas, collections, and conditions to process data.
5. **Test the Flow:**
  - Use **Debug Mode** or test in a **Sandbox** before activation.
6. **Activate the Flow:**
  - Once verified, **Activate** the flow to make it live.

---

## Screen Flow

**Definition:** A flow that provides a **guided user interface** (UI) to collect input and execute actions based on user responses.

◆ **User Interaction?** ✓ Yes

◆ **Trigger Point:** Launched manually by users via **Lightning pages, buttons, actions, or community pages**.

◆ **Use Case:**

- Lead or Case creation wizard
  - Multi-step forms with conditional logic
  - Guided troubleshooting steps
- 

## Record-Triggered Flow

◆ **Definition:** A flow that runs **automatically when a record is created, updated, or deleted**.

◆ **User Interaction?** ✕ No

◆ **Trigger Point:** Triggered when a record meets specific conditions.

◆ **Use Case:**

- Automatically update a field when an Opportunity reaches a certain stage
  - Send email notifications when a Case status changes
  - Assign a newly created Lead to a specific user
- 

## Schedule-Triggered Flow

**Definition:** A flow that runs at a **specific time and frequency** (daily, weekly, or a custom schedule).

◆ **User Interaction?** ✕ No

◆ **Trigger Point:** Scheduled to run at a defined interval.

◆ **Use Case:**

- Send weekly reminders to inactive users
  - Delete outdated records every month
  - Assign follow-up tasks to sales reps at the start of each month
- 

## Auto-Launched Flow

**Definition:** A background process flow that runs **without user interaction** and is triggered via Apex, Process Builder, or another automation.

◆ **User Interaction?** ✕ No

◆ **Trigger Point:** Called from Apex, Process Builder, or another automation tool.

◆ **Use Case:**

- Automatically create related records after a new record is inserted
- Update multiple related records when a field value changes

- Integrate with external systems using APIs

---

## Platform Event-Triggered Flow

**Definition:** A flow that **listens to platform events** and performs actions when a specific event occurs.

◆ **User Interaction?** ✗ No

◆ **Trigger Point:** Triggered when a **Platform Event message** is received.

◆ **Use Case:**

- Sync customer order details from an external system into Salesforce
- Notify users when a system integration fails
- Trigger automated actions based on IoT (Internet of Things) data
- 

---

How to Choose the Right Flow?

Flow Type	Use Case	User Interaction?
Screen Flow	Guided UI for users	✓ Yes
Record-Triggered Flow	Automate on record changes	✗ No
Schedule-Triggered Flow	Run at a specific time	✗ No
Auto-Launched Flow	Background processing	✗ No
Platform Event-Triggered Flow	Respond to external events	✗ No

## 1. What is Salesforce Flow, and why is it used?

**Answer:**

Salesforce Flow is an automation tool that allows users to build business processes without writing code. It is used for **automating tasks, updating records, sending emails, and executing business logic** without Apex triggers or batch jobs.

## 2. What are the different types of Flows in Salesforce?

**Answer:**

- **Screen Flow** – User-interactive, used for data collection.
- **Record-Triggered Flow** – Runs automatically on record creation, update, or deletion.
- **Schedule-Triggered Flow** – Runs at specific time intervals.
- **Auto-Launched Flow** – Background process triggered by Apex, Process Builder, or another Flow.
- **Platform Event-Triggered Flow** – Runs in response to a platform event.

### 3. What is the difference between Record-Triggered Flow and Process Builder?

Answer:

- **Record-Triggered Flow** is more **efficient, faster, and supports before-save updates** (reducing SOQL and DML operations).
- **Process Builder** is **deprecated** in favor of Flow and lacks before-save efficiency.

### 4. What is the difference between Before-Save and After-Save Record-Triggered Flows?

Answer:

- **Before-Save Flow** → Runs **before** the record is saved. Used for **field updates** (more efficient than Process Builder or Apex Triggers).
  - **After-Save Flow** → Runs **after** the record is saved. Used for **actions like sending emails, creating related records, or complex automation**.
- 

### 5. How does Before-Save Flow improve performance over Apex Triggers?

Answer:

Before-Save Flows **execute field updates without DML operations**, reducing database transactions and improving efficiency compared to **Apex Triggers, Process Builder, or Workflow Rules**.

### 6. Can we create multiple Record-Triggered Flows on the same object?

Answer:

Yes, but it's best practice to **combine logic into a single Flow** for better performance and maintainability.

### 7. How do you handle recursion in Record-Triggered Flows?

Answer:

- Use **Flow Trigger Order (Execution Order)** to **control the sequence of Flows**.
- Use **custom fields or variables** to prevent multiple executions.

### 8. What are Flow Trigger Order numbers?

Answer:

Salesforce allows **Record-Triggered Flows to have execution order numbers (1-2000)**. Lower numbers run first, helping **control execution priority**.

---

## 9. What are the best practices for building Screen Flows?

**Answer:**

- Use **Decision elements** to guide users dynamically.
- Keep UI clean and simple using **Dynamic Forms for Flow**.
- Minimize API calls to reduce performance issues.

## 10. How do you prepopulate fields in a Screen Flow?

**Answer:**

- Use **Get Records** to fetch existing values.
- Use **Record variables and assignments** to set default values in screen components.

## 11. How can you use Screen Flow inside a Lightning Page?

**Answer:**

- Add a **Flow component** to the Lightning Record Page.
  - Pass **recordId** as an input variable to fetch record details dynamically.
- 

## 12. How does a Schedule-Triggered Flow work?

**Answer:**

It runs at a **specific time (daily, weekly, etc.)** to execute logic on multiple records, replacing batch Apex for scheduled jobs.

## 13. Can we use Schedule-Triggered Flow for bulk record processing?

**Answer:**

Yes, but **limit the number of records** using **Get Records with filters** to avoid hitting **governor limits**.

## 14. How do you prevent a Schedule-Triggered Flow from processing the same records multiple times?

**Answer:**

- Use a **custom checkbox field** (e.g., "Processed") and update it after execution.
  - Use a **Last Run Date** field to track when records were processed.
- 

## 15. How can an Auto-Launched Flow be triggered?

**Answer:**

- **Apex** (`Flow.Interview` class)
- **Process Builder** (now replaced by Flow)
- **Another Flow** (using a Subflow)

## 16. Can an Auto-Launched Flow have Screen elements?

**Answer:**

No, **Auto-Launched Flows** run in the background and do not support Screen components.

---

## 17. What is a Platform Event-Triggered Flow?

**Answer:**

A Flow that listens for **Platform Events** and performs actions in real-time when the event occurs.

## 18. How does a Platform Event-Triggered Flow differ from a Record-Triggered Flow?

**Answer:**

- **Record-Triggered Flow** → Runs when a record is created/updated.
  - **Platform Event Flow** → Runs when an **external system sends an event to Salesforce**.
- 

## 19. How can you debug a Flow in Salesforce?

**Answer:**

- Use **Debug Mode** in Flow Builder.
- Use **Fault Paths** to handle errors gracefully.
- Check **Flow Interview logs** in the **Setup → Paused & Failed Flow Interviews**.

## 20. What is a Subflow, and when should you use it?

**Answer:**

A **Subflow** is a Flow that can be called from another Flow.

**Use Cases:**

- **Reusability:** Common logic used in multiple Flows.
- **Modularity:** Break down complex processes into smaller Flows.

## 21. How do you handle bulkification in Flows?

**Answer:**

- Use **Get Records with filters** to limit query size.

- Use **Loop with Collection Variables** instead of multiple DML operations.

## 22. How can you call an Apex method from a Flow?

**Answer:**

Use **Apex-Defined Invocable Methods** with the `@InvocableMethod` annotation.

---

## 23. How do you handle errors in Flow?

**Answer:**

- Use **Fault Paths** to send error notifications.
- Use **Try-Catch Logic in Apex-Invoked Flows**.

## 24. What are some common Flow governor limits?

**Answer:**

- Max 2,000 elements per Flow interview.
- Max 50,000 records processed per transaction.
- Max 100 SOQL queries in a Flow.

## 25. How do you optimize a Flow for performance?

**Answer:**

- Use **Before-Save Flows** instead of triggers for field updates.
  - **Limit Loop Iterations** by filtering records early.
  - Use **Subflows** for common logic.
- 

## 26. How would you create a Flow to assign Leads based on region?

**Answer:**

Use a **Record-Triggered Flow with Decision Elements** to check Lead's location and assign it accordingly.

## 27. How would you automate case escalation using Flows?

**Answer:**

Use an **After-Save Record-Triggered Flow** to check **case status, priority, and last update time** and escalate accordingly.