**Interview Questions**

1. What is the purpose of using **GlideRecordSecure** in ServiceNow?

A: GlideRecordSecure is designed to prevent unauthorized access to data. GlideRecordSecure enforces access controls and ensures that the user has permission to access the records they are querying. Developers can use GlideRecordSecure in the same way as GlideRecord to query and manipulate ServiceNow records.

1. Can you describe your current project and company, and explain when you began working with ServiceNow?

A:

1. Could you share a complex scenario you’ve encountered and how you handled it?

A: I was working on a ServiceNow ITSM implementation for a client in the finance sector. The client needed an automated process to handle incident escalations based on multiple parameters, including priority, SLA breach, and impact on critical business services. Additionally, they wanted to integrate with their existing monitoring tools to trigger escalations automatically when certain thresholds were met.

1. **What are your specific roles and responsibilities in your current project?**

1. Requirement Gathering and Analysis

* Collaborating with Stakeholders: Work closely with business analysts, process owners, and end users to gather requirements for new functionalities or enhancements.
* Understanding Business Needs:

### 2. **ServiceNow Application Development**

* Custom Application Development: Design and build custom applications on the ServiceNow platform using ServiceNow Studio and scoped applications.
* Developing Workflows: Create and configure workflows, approval processes, and notifications to automate business processes.
* UI Development: Build and customize forms, lists, and dashboards. Develop user-friendly interfaces with UI Policies, Client Scripts, and UI Actions.
* Script Development: Write and maintain Business Rules, Client Scripts, Script Includes, UI Pages, Glide API scripting, and other server-side/client-side scripts.

### 3. **Customization and Configuration**

* Module Configuration: Configure and customize ServiceNow modules like ITSM
* Service Catalog & Portal Development: Customize the Service Catalog and Service Portal, including building catalog items, record producers, and workflows.
* Integration Development: Integrate ServiceNow with other third-party tools and platforms using REST, Scripted Web Services, MID Servers, and JDBC integrations.
* Reports and Dashboards: Develop custom reports, create performance analytics, and build dashboards to track KPIs and provide real-time insights to stakeholders.

### 4. **System Administration**

* Instance Management: Manage ServiceNow instances, including cloning, patching, and instance upgrades.
* User and Role Management: Configure ACLs (Access Control Rules), user roles, and permissions to maintain data security and restrict access.
* Table and Field Customization: Configure and manage fields, tables, and relationships in ServiceNow.

### 5. **Problem Solving and Debugging**

* Incident and Problem Management: Investigate and resolve issues related to the ServiceNow platform and applications.
* Troubleshooting Scripts: Debug client and server-side code (such as Business Rules, UI Policies, and Workflows) to identify and fix errors or inefficiencies.

### 6. **Implementation of Best Practices**

* Best Practices Adherence: Follow ServiceNow best practices for development, including code reusability, script optimization, and minimal customization to keep future upgrades smooth.

### 7. **Testing and Deployment**

* Unit and Integration Testing: Perform unit tests and collaborate with QA teams for end-to-end testing, ensuring that the solution meets the business requirements.
* Development to Production: Use Update Sets or ATF (Automated Test Framework) to move configurations from the development instance to production.

### 8. **Documentation and Knowledge Sharing**

* Technical Documentation: Document code, configurations, and processes to ensure maintainability and clarity for future development.
* Training: Attending internal training and ask if any doubts.
* Stay Updated: Stay current with new features, modules, and capabilities introduced by ServiceNow with each release.

### Example of Daily Responsibilities:

* Develop new features and enhancements in ITSM, like adding new Incident management processes.
* Build REST API integrations between ServiceNow and external tools (e.g., Slack, Jira).
* Automate repetitive tasks using workflows in the ServiceNow platform.
* Collaborate with stakeholders to define requirements and translate them into technical solutions.
* Troubleshoot and resolve ServiceNow issues encountered by users or in workflows.
* Perform regular instance health checks, reviewing performance logs, and resolving slow queries.

1. What is the most challenging customization you've implemented in your development work?

One of the most challenging customizations I implemented in ServiceNow involved integrating an external ITSM tool with ServiceNow using REST APIs to achieve seamless incident synchronization between two different platforms. This required complex scripting, API calls, error handling, and ensuring data consistency across both systems.

### **Scenario:**

We had two different companies using different ITSM systems—one using ServiceNow and the other using a legacy ITSM tool. Both organizations had a contractual agreement to collaborate on incident resolution. When an incident was raised in one system, it needed to be automatically created and synchronized in the other system, ensuring both sides were aware of the incident’s status, priority, and updates.

### **Challenge:**

1. Complex API Integration: The external system only supported certain REST API calls, which were not fully compatible with ServiceNow’s structure. Mapping the data fields between ServiceNow and the external ITSM tool was complex due to different field names and data formats (e.g., categories, priorities, and incident statuses).
2. Real-Time Synchronization: The integration needed to ensure that any updates (e.g., status changes, comments) made on the incident in one system would be automatically reflected in the other system in real time.
3. Handling Errors: We needed robust error handling, particularly to manage situations where the external ITSM tool was down or where the API rate limit was exceeded.
4. Security Concerns: Authentication and data security were important considerations as we were dealing with sensitive incident data across two organizations.

### **Solution:**

1. REST API Implementation:
   * Outbound REST Message: I set up an outbound REST message in ServiceNow to create, update, and delete incidents in the external system using their available API endpoints.
   * Inbound REST API: I also implemented an Inbound Scripted REST API in ServiceNow that allowed the external ITSM tool to communicate back and update incident records in ServiceNow.
2. Field Mapping and Transformation:
   * I used Script Includes to perform data transformations and map fields from the external system to ServiceNow fields. For example, if the external system had different categories or priorities, I implemented mapping logic that would transform them into ServiceNow-compatible values.
3. Handling Bi-Directional Updates:
   * To ensure real-time synchronization, I used Business Rules and Event Management. Business rules were triggered on the creation or update of an incident to call the external system’s API and synchronize the changes. Similarly, the inbound REST API processed updates from the external tool.
4. Error Handling and Logging:
   * To manage downtime or API limits, I implemented error handling and retry logic using GlideSystem logs and set a time-based retry mechanism using Scheduled Jobs. If the API call failed, the system would log the failure, and a retry would occur after a set interval.
5. Security and Authentication:
   * I used OAuth 2.0 for authentication between the two systems to ensure secure API communication. Roles and permissions were configured to limit who could interact with the external tool's API.

### **Outcome:**

The integration successfully enabled both systems to create and synchronize incidents seamlessly. Whenever an incident was created in ServiceNow, it was automatically created in the external system with all relevant details. Any updates to the incident (status changes, comments, resolutions) were reflected in both systems, ensuring transparency and collaboration between the two organizations.

### **Key Challenges and How I Handled Them:**

1. Data Mapping Issues: The category and priority fields in both systems did not align perfectly. I wrote a custom Script Include to handle the transformation of these values so they would match both systems correctly.
2. Error Recovery: Handling situations where the external system was down required writing robust error-handling logic, where failed API calls were retried at regular intervals without affecting the user experience in ServiceNow.
3. Real-Time Performance: Ensuring real-time updates with large volumes of incidents was a performance challenge. I optimized the API calls using asynchronous processing where possible, reducing the load on the ServiceNow instance.

### **Conclusion:**

This was a challenging yet rewarding experience, as the integration allowed both companies to collaborate more effectively by sharing critical incident data in real time. This project deepened my understanding of REST APIs, error handling, and real-time data synchronization in ServiceNow, and it also improved my ability to troubleshoot complex integrations while maintaining security and performance.

1. Based on your previous answers, further follow-up questions will be asked.
2. In the **Incident** table, you have three custom fields: X, Y, and Z. How would you populate field Z based on the selections in X and Y? Please explain the different methods you would use for this scenario.
3. In the coding context of the previous question, which approach would you take: client script or business rule? Where would you store the values of X and Y in the meantime?
4. How can you retrieve the **sys\_id** of an incident form using a Script Include and set it to a custom table?
5. How would you filter assignment group users based on the type of group?

A. Go to record list and right click on Assignment group and select group by Assignment group. Then it will display records group wise

1. How can you retrieve the time spent in each state of an incident (e.g., new, in progress, on hold)?

**Business Rules**

1. How would you enforce a rule to ensure that all child incidents are closed before closing the parent incident?

A: you can use the parent-child incident relationship to automatically resolve the children when the parent is resolved:

1. What is the role of **g\_scratchpad** in a business rule, and when would you use it—specifically in display rules or other types of business rules?

A:The g\_scratchpad object can be used to pass information from the server to the client when the client needs information that isn't available on the form. For example, if a client script needs to access a field that isn't on the form, the data can be passed to the client script through the g\_scratchpad object.

1. How would you change the state of an incident to "Resolved" when a related change request is set to "Closed"?
2. Can you introduce yourself and explain the details of your most recent project?
3. Which ITSM module does not include the approval process by default?
4. What are the different methods to migrate code between ServiceNow instances?
5. Which modules or tables are involved in the code migration process for each step?
6. What was the most recent instance you've worked on?

A: Washington DC

1. If there is a field on the Incident form specifying a day of the week (Mon, Tue, etc.), how would you set up a notification to be sent to the caller based on that day? Additionally, if the meeting doesn’t happen, how would you trigger a follow-up notification stating that the meeting was missed?
2. Does the custom field get captured in the audit log, and if so, how?