Prevent User Deletion if Assigned to an Incident

Category: ServiceNow Application Developer

Skills Required:

Script, Business Rules, Glide Records and APIs, User and Group

Management.

Problem Statement:

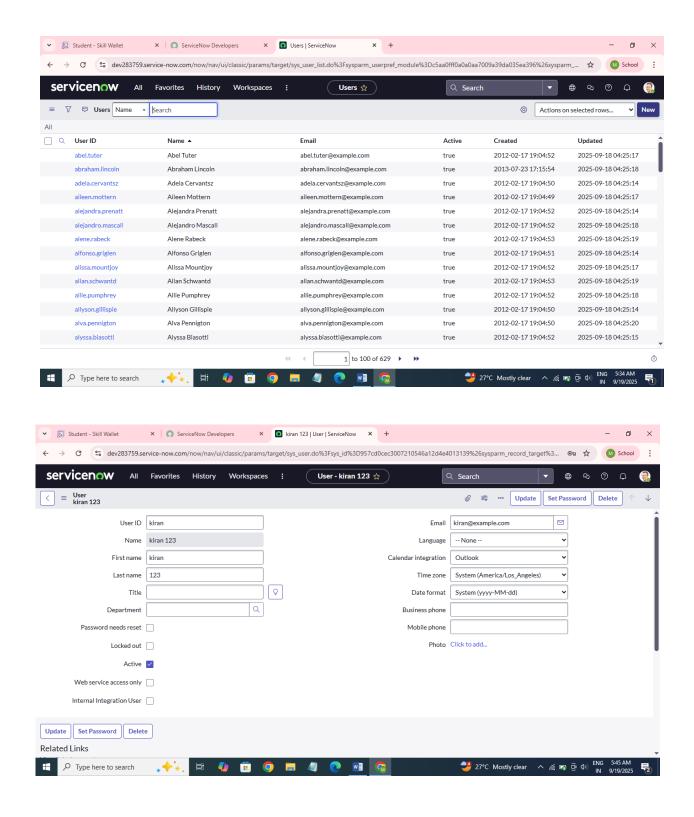
In an IT Service Management environment, users are frequently assigned to incidents for issue resolution and tracking. However, the current system lacks a validation mechanism to prevent the deletion of a user who is still actively assigned to incidents. This can lead to broken data references, loss of accountability, and disruption in workflow continuity.

There is a need to implement a safeguard that prevents such deletions unless all assigned incidents are closed or reassigned.

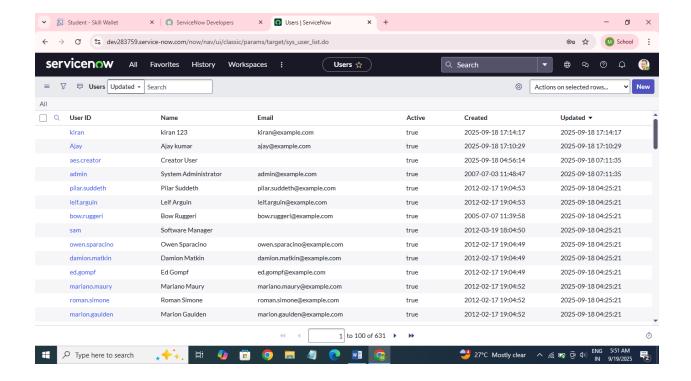
User Creation

Create Test Users

- 1. Go to ServiceNow >> All >> Users (under System Security)
- 2. Click on New
- 3. Create two users (e.g., kiran123,ajaykumar)
- 4. Submit and verify user records.



✓ Now the two users are created (Ajay kumar, kiran).



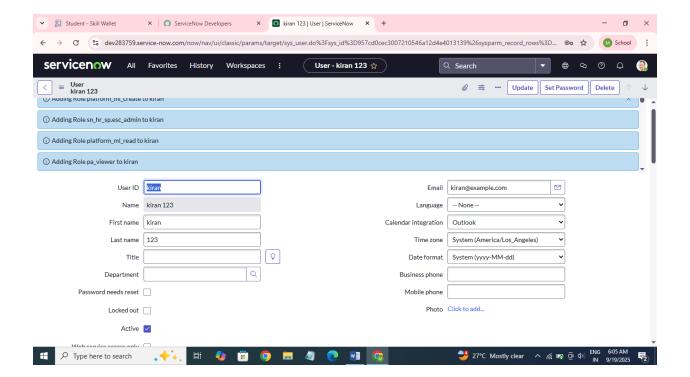
Assign Incident to User

Assign Incidents

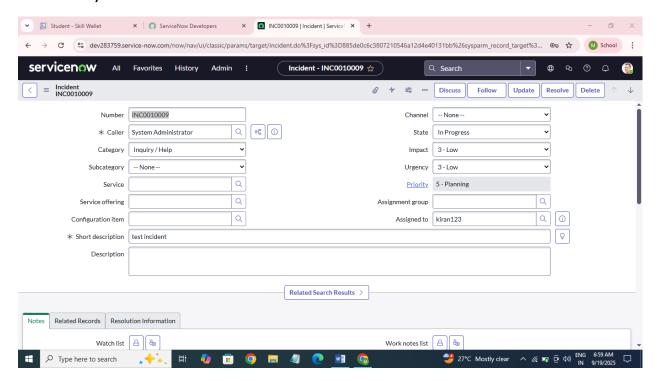
- 1. Navigate to the Incident table.
- 2. Create a new incident and assign it to one of the created users (e.g., kiran123)
- 3. Keep the incident Active = true and State = In Progress

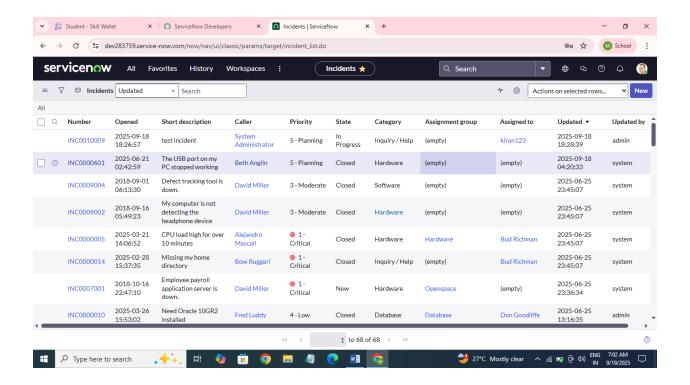
Note: To assign any user the user should have at least one role so assigned a role to the user before assigning incident

✓ First assign a role to one of the user (kiran). Then add the incident to the user.



✓ Now assign the incident to the user, who was assigned a role. (kiran 123)





Business Rule Creation

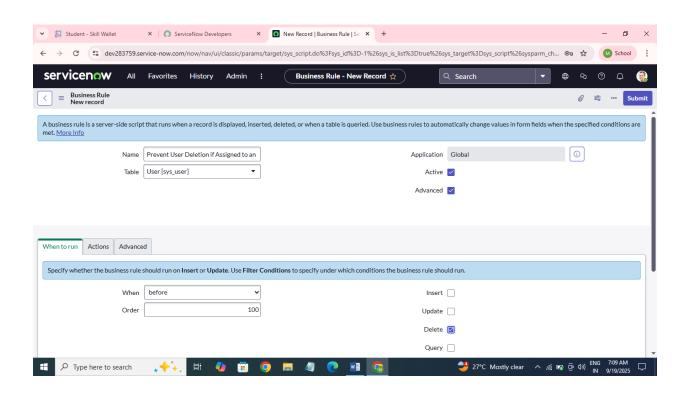
Create Business Rule

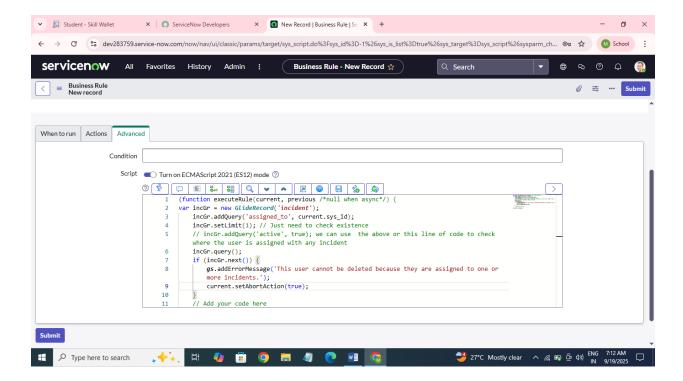
- 1. Go to System Definition >> Business Rules
- 2. Click on New
- 3. Fill in:
- 4. Name: Prevent User Deletion if Assigned to an Incident
- 5. Table: sys_user
- 6. When: Before
- 7. Delete: Checked
- 8. Script:// Add your code here
- 9. Click submit.

SCRIPT:

/// (function executeRule(current, previous /*null when async*/) {
var incGr = new GlideRecord('incident');

```
incGr.addQuery('assigned_to', current.sys_id);
incGr.setLimit(1); // Just need to check existence
  // incGr.addQuery('active', true); we can use the above or this line of code to check where the user is assigned with any incident
  incGr.query();
  if (incGr.next()) {
    gs.addErrorMessage('This user cannot be deleted because they are assigned to one or more incidents.');
    current.setAbortAction(true);
  }
// Add your code here
})(current, previous);
```

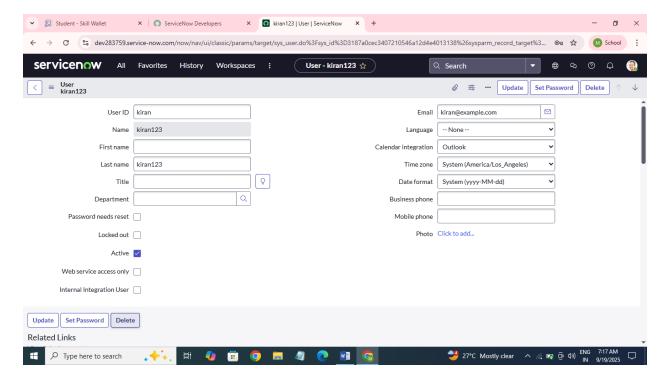




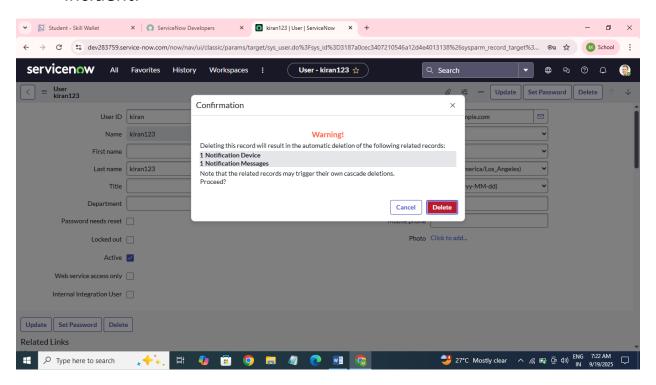
Test Deletion

Attempt to Delete Assigned User

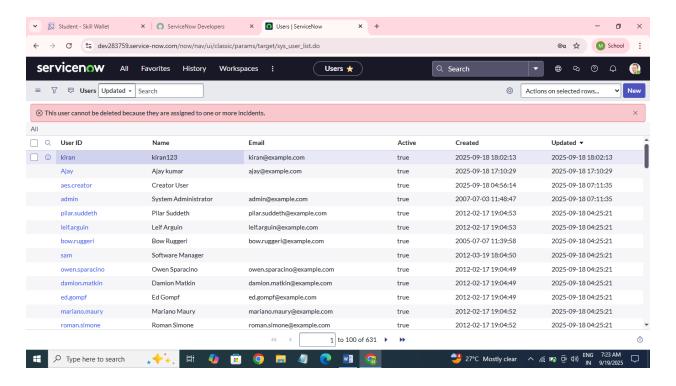
- 1. Go to the user record (kiran123)
- 2. Click Delete
- 3. Verify that deletion is blocked with an error message



✓ Now try deleting the user (kiran) to whom we have assigned an incident.



✓ It shows the error notification as "this user cannot be deleted because they are assigned to one or more incidents" and the user (kiran) is not deleted.

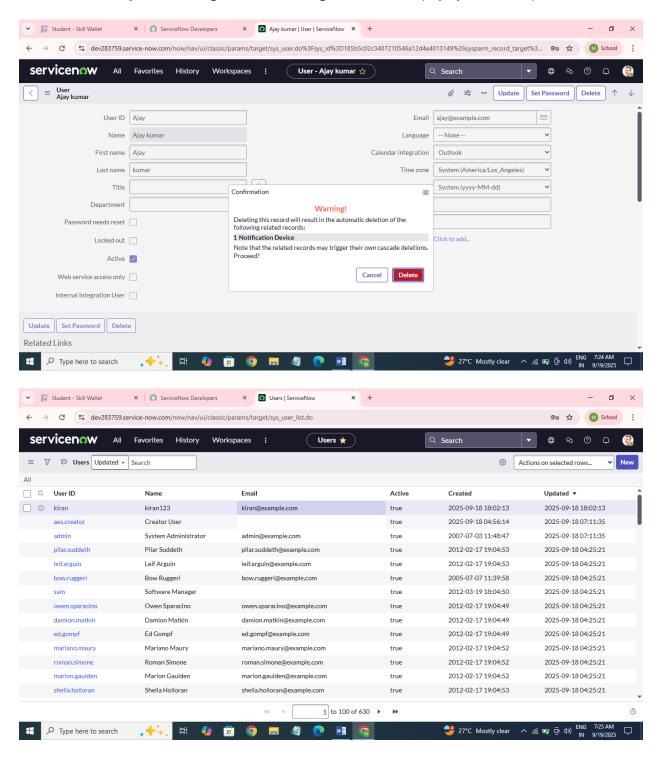


Test With Unassigned User

Attempt to Delete Unused User

- 1. Try deleting the second user (Ajay kumar) who is not assigned to any active incidents.
- 2. Deletion should succeed.

✓ Now try deleting an unassigned user (Ajay kumar).



✓ But the user from the table is get deleted. Because it is not assigned to any incident.

Conclusion

This project provides a safeguard mechanism against accidental or improper deletion of users who are still involved in active incidents. By using a Business Rule on the sys_user table, ServiceNow administrators can ensure that incident ownership and workflow integrity remain intact. This solution upholds data consistency and promotes operational continuity within IT service processes.

--- THE END ---