

# ServiceNow Administration

---

## Prevent User Deletion If Assigned To An Incident

**Team ID:** LTVIP2026TMIDS81090

**Team Size:** 4

### Team Members and Roles

Name	Role
Satyanagaraju Yesuri	Team Leader / ServiceNow Administrator
Naimisha Kundrapu	Member / Incident Management Specialist
Poojitha Reddi	Member / Security & Role Configuration Manager
Durga Mallesh Nunna	Member / Testing & Documentation Engineer

## 1. Introduction

### **Project Title: Prevent User Deletion If Assigned To An Incident**

This project focuses on implementing a validation mechanism in ServiceNow to prevent deletion of users who are assigned to active incidents. The solution ensures system reliability, maintains referential integrity, and prevents workflow disruption in IT Service Management (ITSM) environments.

## 2. Project Overview

### **Purpose**

In ServiceNow, users are assigned to incidents for issue resolution and service tracking. If a user assigned to an active incident is deleted, it may lead to broken references, loss of accountability, and data inconsistency.

The purpose of this project is to implement a Business Rule that validates incident assignments before allowing user deletion.

### **Features**

- User Deletion Validation: Prevents deletion of users who are assigned to active incidents.
- Real-Time Incident Check: Automatically verifies incident assignments before executing delete operation.
- Active Status Filtering: Ensures only active incidents are considered during validation.
- Server-Side Business Rule Execution: Uses a Before Delete Business Rule to enforce logic securely at the backend level.
- GlideRecord Integration: Implements GlideRecord API to query incident records dynamically.
- Role-Based Assignment Control: Requires ITIL role for users to be assigned to incidents.
- Data Integrity Protection: Maintains referential consistency between User and Incident tables.
- Error Message Notification: Displays system alert when deletion attempt is blocked.

## 3. Architecture

### **Platform Architecture**

The project is implemented entirely within the ServiceNow platform using server-side scripting and table relationships.

### **Components Used**

- User Table (sys\_user): Stores user records in the system.
- Incident Table (incident): Stores incident records and assignment details.
- Business Rule (Before Delete): Executes validation logic before a user record is deleted.
- GlideRecord API: Queries the incident table to check for active assignments.

## Architecture Flow

- Admin attempts to delete a user record.
- The “Before Delete” Business Rule is triggered.
- GlideRecord queries the Incident table.
- System checks for active incidents assigned to that user.
- If an active record exists → Deletion is blocked.
- If no active record exists → Deletion is allowed.

## System Architecture

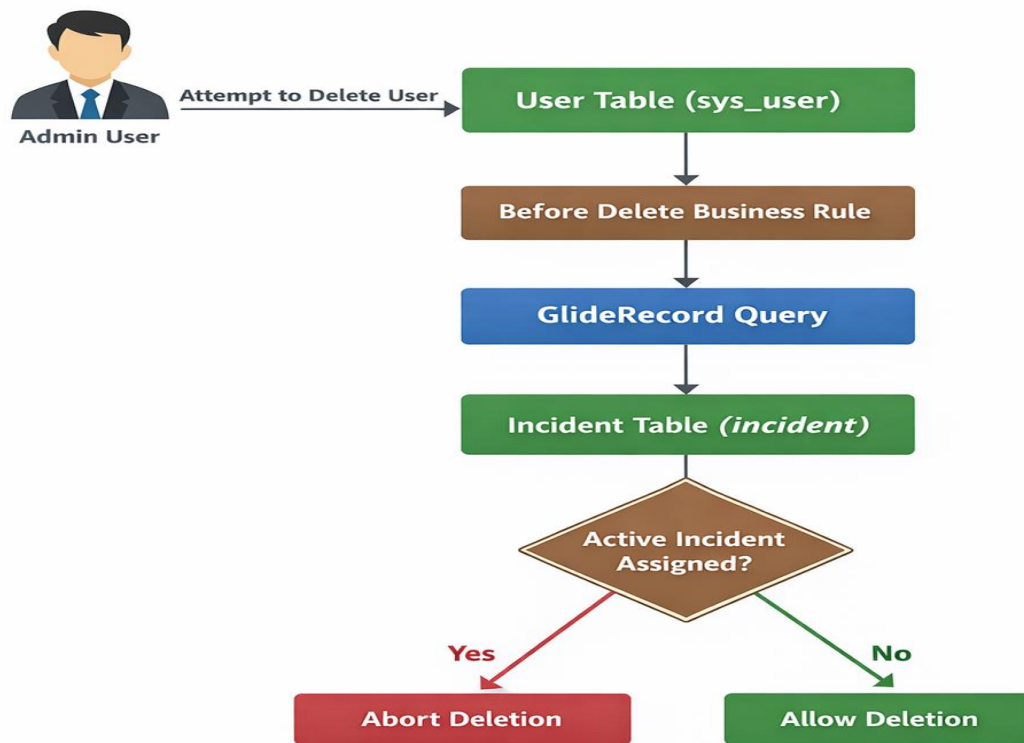


fig 1: User Deletion Validation Architecture in ServiceNow

## Figure Description

The diagram illustrates the workflow executed when an administrator attempts to delete a user record. The process begins from the Admin User, who initiates the deletion request in the User Table (sys\_user). A Before Delete Business Rule is triggered automatically.

This Business Rule uses a GlideRecord query to check the Incident Table (incident) for any active incidents assigned to the selected user.

A decision point evaluates whether an active incident exists:

If Yes, the system aborts the deletion to maintain data integrity.

If No, the system allows the user record to be deleted.

This architecture ensures referential integrity and prevents accidental removal of users who are still associated with active incident records.

## 4. Setup Instructions

### Prerequisites

- Active ServiceNow Developer Instance
- Admin Access
- ITIL Role for assigned users

### Configuration Steps

#### 1. Create Test User:

Navigate to:

User Administration → Users → New

Create a test user record.

#### 2. Assign ITIL Role:

Open the created user → Add ITIL role under Roles related list.

This allows the user to appear in the Assigned to field.

#### 3. Create Incident:

Navigate to:

Incident → Create New

Fill:

Short Description

State = In Progress

Active = True

Assigned To = Test User

#### 4. Create Business Rule:

Navigate to:

System Definition → Business Rules → New

Configuration:

- Table: User [sys\_user]
- When: Before
- Delete: Checked
- Advanced: Checked
- Active: Checked

## 5. Business Rule Script

```
(function executeRule(current, previous) {  
  
    var inc = new GlideRecord('incident');  
    inc.addQuery('assigned_to', current.sys_id);  
    inc.addQuery('active', true);  
    inc.query();  
  
    if (inc.hasNext()) {  
        gs.addErrorMessage("User cannot be deleted because they are assigned to active incidents.");  
        current.setAbortAction(true);  
    }  
  
})(current, previous);
```

This script checks whether any active incident is assigned to the user and blocks deletion accordingly.

## 6. Running the Application

To test the functionality:

- Assign an incident to a user
- Attempt to delete the user
- System displays error message
- User record remains in database

### Execution Validation Process

- Ensure the Business Rule is active on the User (sys\_user) table.
- The rule must be configured as Before Delete.
- Verify that the incident assigned to the user is Active = True.
- Attempting deletion triggers server-side validation.
- If validation fails, an error message is displayed.
- The user record remains unchanged in the database.

## 7. Functional Validation

Functional testing was conducted to verify:

- Business Rule triggers before deletion
- Incident query executes correctly
- Active condition filtering works properly
- System error message displays accurately
- Deletion is aborted when condition is met

## 8. User Interface

The user interface includes:

User Form – Displays user details and roles

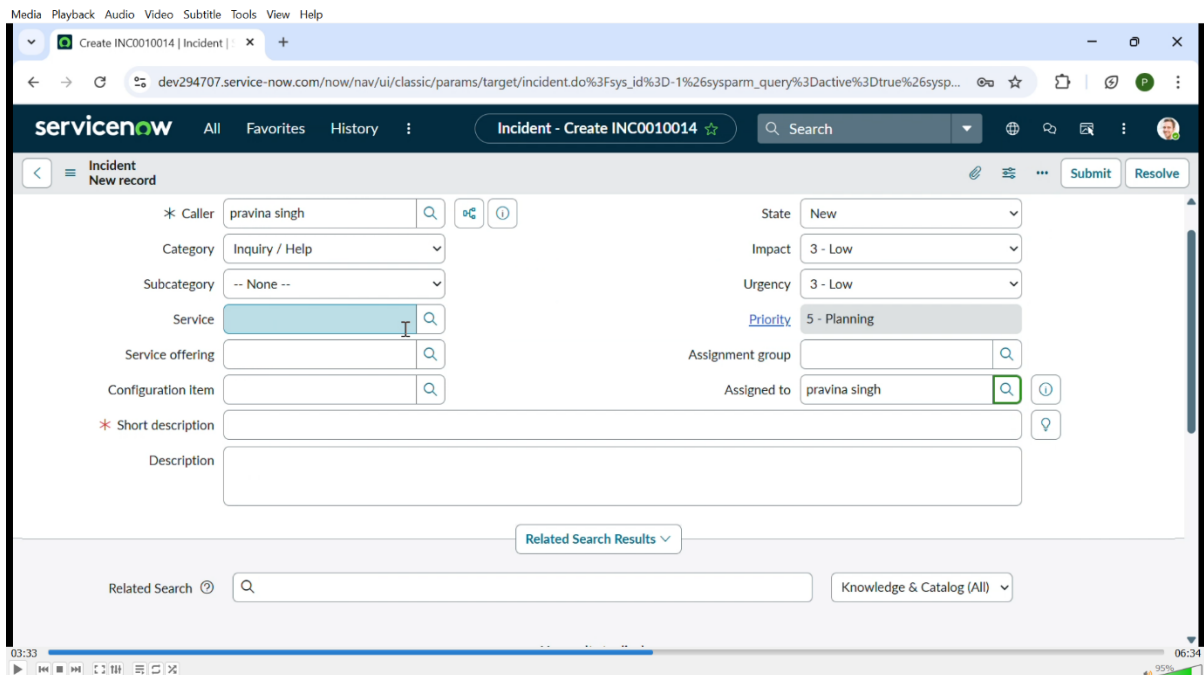
Incident Form – Shows assignment information

Error Message Banner – Displays validation error

### User Record Form

The screenshot shows the 'User - New Record' form in the ServiceNow interface. The form is organized into two main columns. The left column contains fields for User ID (pravina), First name (pravina), Last name (singh), Title (IT Technician), Department (IT), Password needs reset (checkbox), Locked out (checkbox), Active (checkbox, checked), Identity type (Human), and Internal Integration User (checkbox). The right column contains fields for Email (pravina@gmail.com), Language (None), Calendar integration (Outlook), Time zone (System (America/Los Angeles)), Date format (System (yyyy-MM-dd)), Business phone, Mobile phone, and Photo (Click to add...). A 'Submit' button is located at the bottom left of the form. The browser window shows the URL 'dev294707.service-now.com/now/nav/ui/classic/params/target/sys\_user.do%3Fsys\_id%3D-1%26sys\_is\_list%3Dtrue%26sys\_target%3Dsys\_us...' and the ServiceNow logo in the top left corner.

## Incident Assignment Form:

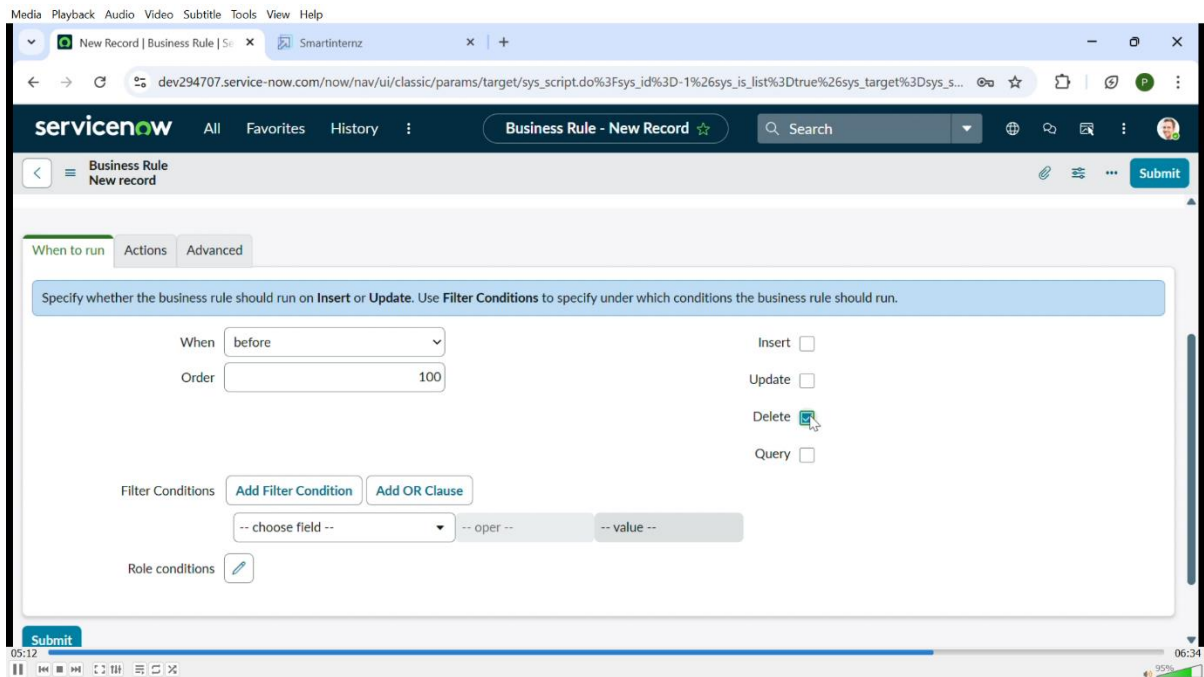


The screenshot shows the 'Incident - Create INC0010014' form in the ServiceNow interface. The form is divided into several sections for data entry:

- Caller:** pravina singh
- Category:** Inquiry / Help
- Subcategory:** -- None --
- Service:** (Dropdown menu is open, showing a search bar)
- Service offering:** (Empty field)
- Configuration Item:** (Empty field)
- State:** New
- Impact:** 3 - Low
- Urgency:** 3 - Low
- Priority:** 5 - Planning
- Assignment group:** (Empty field)
- Assigned to:** pravina singh
- Short description:** (Empty field)
- Description:** (Empty text area)

At the bottom, there is a 'Related Search Results' section with a search bar and a 'Knowledge & Catalog (All)' dropdown.

## Business Rule Configuration:

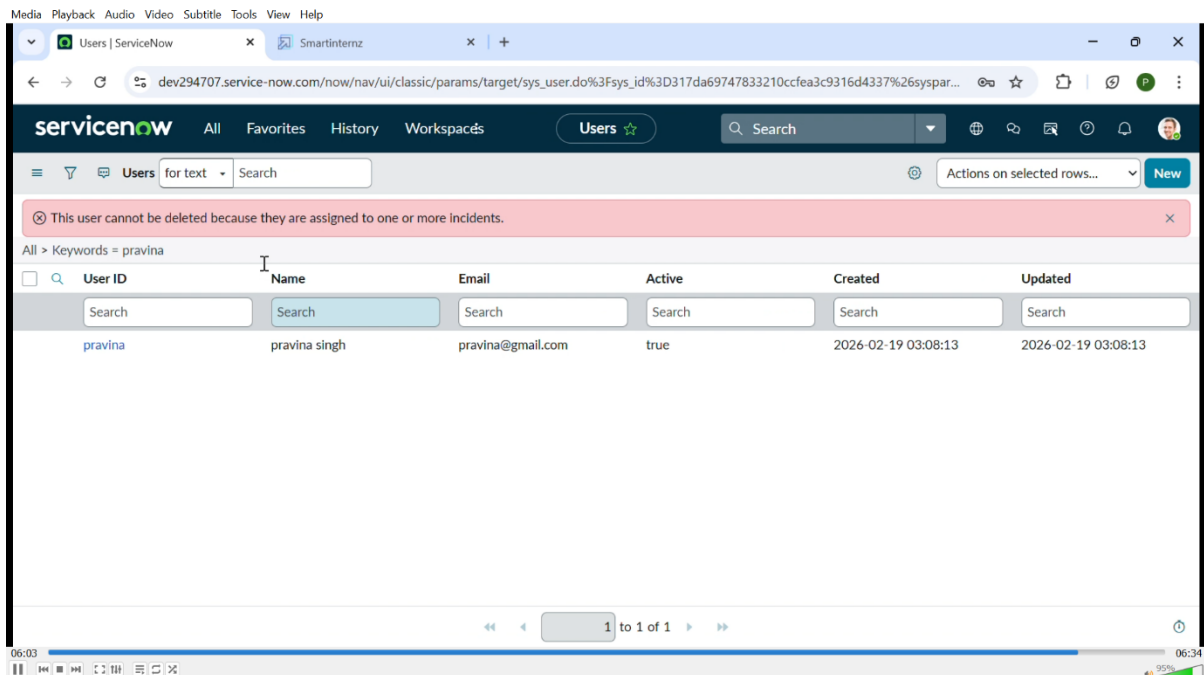


The screenshot shows the 'Business Rule - New Record' form in the ServiceNow interface. The form is divided into several sections for configuration:

- When to run:** before
- Order:** 100
- Filter Conditions:** (Add Filter Condition, Add OR Clause)
- Role conditions:** (Edit icon)
- Actions:** Insert ☐, Update ☐, Delete ☒, Query ☐

The 'Delete' action is selected. The 'Filter Conditions' section is currently empty, with a dropdown menu showing '-- choose field --' and a button to 'Add Filter Condition'.

## Error Message Display:



## 9. Testing

Manual testing was performed for the following scenarios:

- Assigned user deletion attempt
- Unassigned user deletion attempt
- Role visibility validation
- Self Service vs Default view behavior

All scenarios were validated successfully.

## 10. Known Issues

- Assigned To field not visible in Self Service view
- ITIL role required for user to appear in assignment field
- Business Rule will not execute if Delete checkbox is not enabled



## 11. Future Enhancements

- Extend validation to Change and Problem tables
- Implement logging mechanism for deletion attempts
- Send email notification when deletion is blocked
- Create UI policy to guide administrator