### **Access Control for Project table**

**1.Project Objectives**

The Access Control for project table aims to address the challenge of Role-Based Access Optimization for Enhanced Efficiency within the ServiceNow platform. The project's key objectives include:

1. **Streamline User Interactions**: The primary focus is to optimize user access based on their roles, ensuring that each user has personalized access to the features and functionalities they need, without unnecessary complexity. This will result in a smoother, more efficient interaction with the ServiceNow platform.
2. **Personalized and Role-Specific Access**: Implementing a system that dynamically adjusts the access privileges based on the user’s role and responsibilities. This means users will only see the tools, applications, and data relevant to their tasks, reducing the cognitive load and increasing productivity.
3. **Advanced User-Centric Design:** Leverage principles of user-centric design to ensure the solution is intuitive, easy to navigate, and adaptable to diverse user needs. The design will focus on simplifying workflows, minimizing redundancies, and enhancing user satisfaction.
4. **Adaptive Configuration Tools**: Integrate tools that allow for flexible configuration of user roles and permissions, ensuring that access control mechanisms can be adjusted as business requirements evolve, without disrupting the user experience.
5. **Intelligent Access Control:** Utilize smart access control mechanisms, such as context-aware permissions and automated workflows, to ensure that users receive the appropriate access at the right time, further strengthening operational efficiency and security.

**Specific Outcomes**

**Improved Data Security and Privacy**:

* **Outcome**: By restricting access to project tables based on user roles, sensitive project data will be better protected from unauthorized access. This minimizes the risk of data breaches or inadvertent data exposure, ensuring compliance with internal security policies and external regulations (e.g., GDPR, HIPAA).

**Streamlined User Experience**:

* **Outcome**: Users will only see the project data and actions relevant to their role. This reduces clutter, simplifies navigation, and enables team members to focus on tasks that are directly related to their responsibilities, improving productivity and satisfaction.

**Faster Onboarding and Role Adjustments**:

* **Outcome**: Automation of access control will ensure that new users or employees who change roles automatically receive the correct permissions for project tables. This will eliminate the manual process of updating user access, reducing the chances of errors or delays during onboarding.

**Enhanced Compliance and Audit Trails**:

* **Outcome**: By implementing detailed role-based access control, the system will automatically track and log user interactions with project tables. This provides an accurate audit trail that can be reviewed during internal or external audits, making it easier to demonstrate compliance with security and data protection standards.

**Reduction in Access-Related Errors**:

1. **Outcome**: With well-defined roles and permissions, users will be less likely to make errors related to accessing or modifying project data that is outside of their scope. This reduces the risk of mistakes that can delay projects or lead to data inconsistencies.
2. **Key features and Concepts Used**

**Role-Based Access Control (RBAC):**

* Feature: Role-based access control is the core concept for managing who can view, edit, or manage project data based on predefined roles (e.g., Project Manager, Team Member, Executive). This ensures that each user has the appropriate level of access to project tables based on their responsibilities, minimizing the risk of unauthorized access and ensuring data security.
* Concept: Users are assigned roles, and each role has specific permissions that dictate access to various project data and actions within ServiceNow.

**Granular Permissions:**

* Feature: Permissions are set at a granular level to control access to individual fields or records within a project table. For example, a project manager might have the ability to edit project milestones, while a team member can only update their task statuses.
* Concept: This feature allows for fine-tuned access control, where not only access to the project table itself is restricted, but also specific operations such as creating, editing, or viewing individual records.

**Automated Access Assignment:**

* Feature: Automatically assign or update user access based on their role or department using rules and workflows. For instance, when a new user joins a project, their access permissions to the project table can be automatically applied based on their job role without manual intervention.
* Concept: Workflow automation reduces manual errors and improves efficiency, ensuring the right permissions are always applied at the right time.

**Context-Aware Access Control:**

* Feature: Access permissions can be dynamically adjusted based on factors such as project status, user location, or time of access. For instance, certain users may be granted access to sensitive data during specific project phases, while access is restricted during other times.
* Concept: This concept enhances security and flexibility, ensuring that users are granted access only when it is contextually appropriate, which also minimizes the risk of unauthorized actions.

**Audit Trails and Activity Logs:**

* Feature: Detailed logs are generated for every action a user takes within project tables, including who accessed which records and what actions were performed (e.g., view, update, delete). This ensures accountability and can be reviewed for compliance purposes.
* Concept: Keeping track of user activities supports transparency and provides an audit trail, which is essential for regulatory compliance, troubleshooting, and security monitoring.

**Dynamic Access Groups:**

Feature: Group users based on specific criteria (e.g., project team, department) and manage access as a group rather than individually. This makes it easier to apply access rules to multiple users simultaneously, saving time and effort when managing user access.

Concept: Dynamic groups simplify user access management, especially in larger organizations where teams and projects change frequently.

1. **Detailed Steps to Solution Design**

**Pre-Requisites: -**

1. Knowledge on **Form, Application, Module,Tables.**
2. Knowledge on **ACL.**
3. **Skills used to solve the problem statement: -**
4. Service Now Administration.

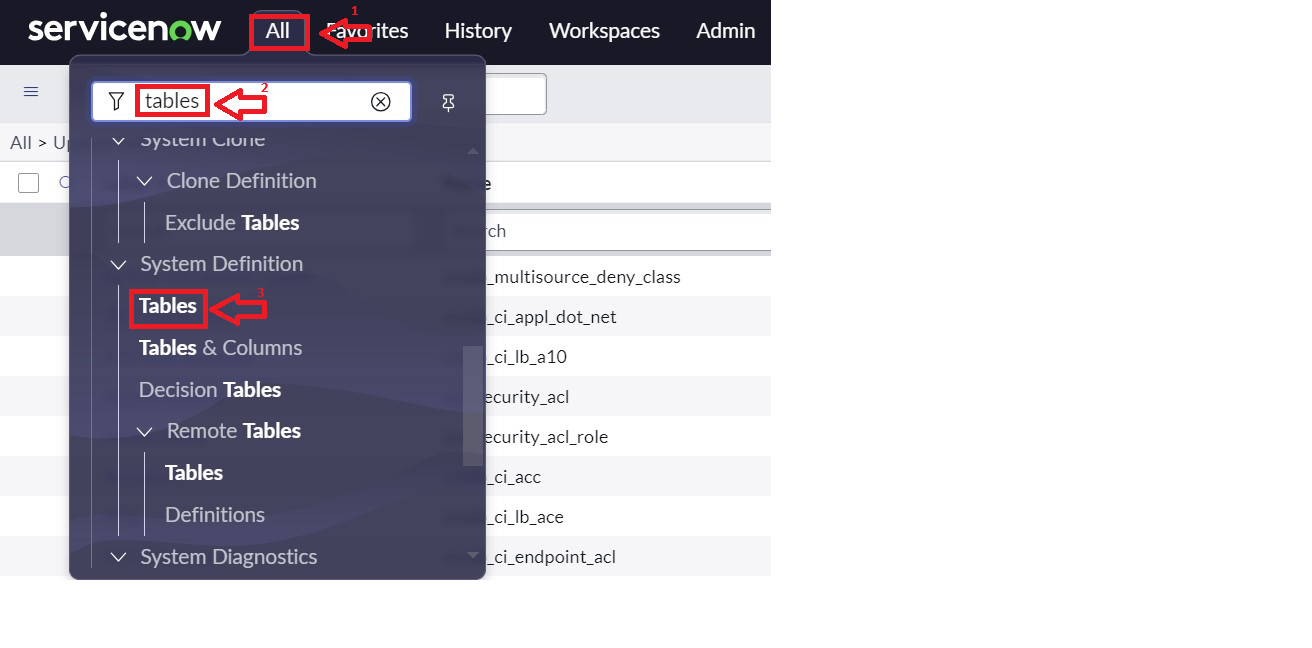
**Activity – 1:**

I started by opening the ServiceNow Developer Instance and clicked on "All" in the left-hand menu. Then, I searched for "Tables" in the search bar and selected "Tables" under the "System Definition" section. After that, I clicked on "New" to create a new table. I filled in the following details:

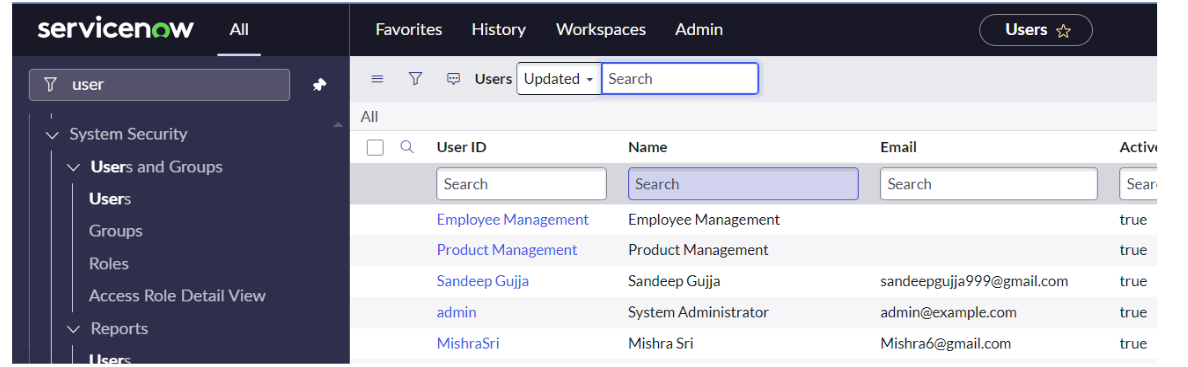
* **Label**: Project
* **Name**: u\_st\_project
* **Add module to menu**: Selected "Create New"
* I left everything else as the default settings.

Next, under the "Columns" section, I clicked on "Insert a new row" and added the following columns:

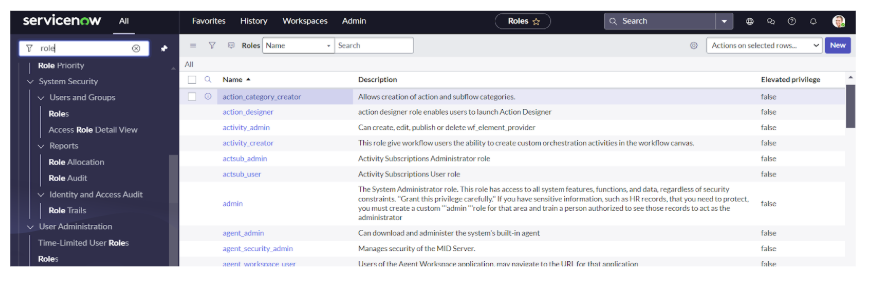
* **Column label**: Name >> **Type**: String>> Max Length:100
* **Column label**: Project Overview >> **Type**: String>>Max Length>> 200
* **Column label:**Budget >>**Type** : Price
* **Column label:**Total Expenses>> **Type:** Price

Finally, I clicked on "Submit" to create the table.

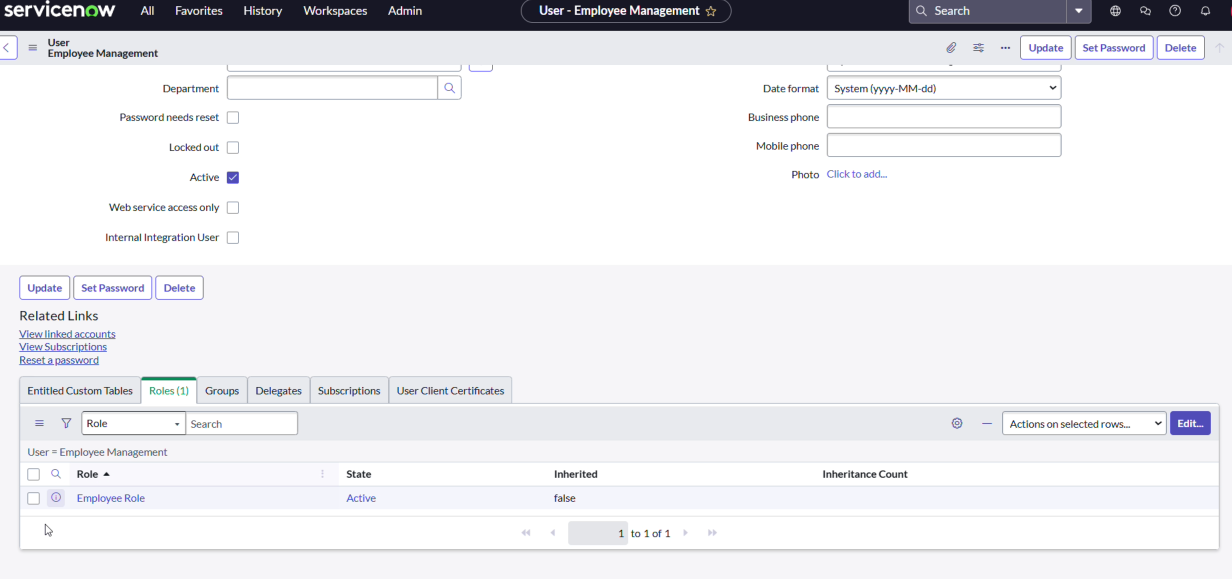
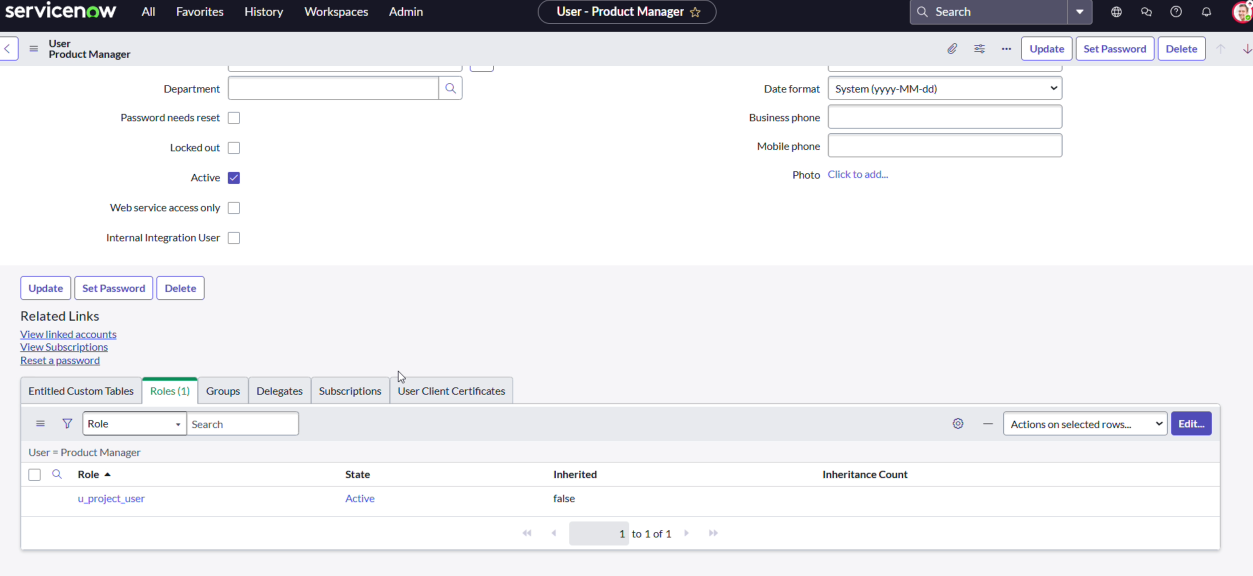
**Activity – 2:**

I opened ServiceNow and clicked on "All" in the left-hand menu. Then, I searched for "Users" in the search bar and selected "Users" under the "System Security" section. After that, I clicked on "New" to create a new users(Product Manager and Employee Management) I filled in the necessary details to create the user and then clicked on "Submit" to save the new user.

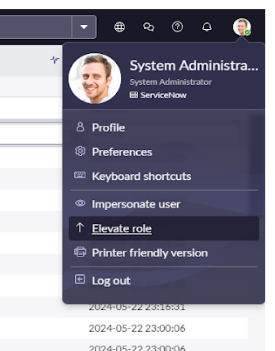
Activity – 3:

I opened ServiceNow and clicked on "All" in the left-hand menu. Then, I searched for "Roles” in the search bar and After that, I clicked on "New" to create a roles and created Employee Role.

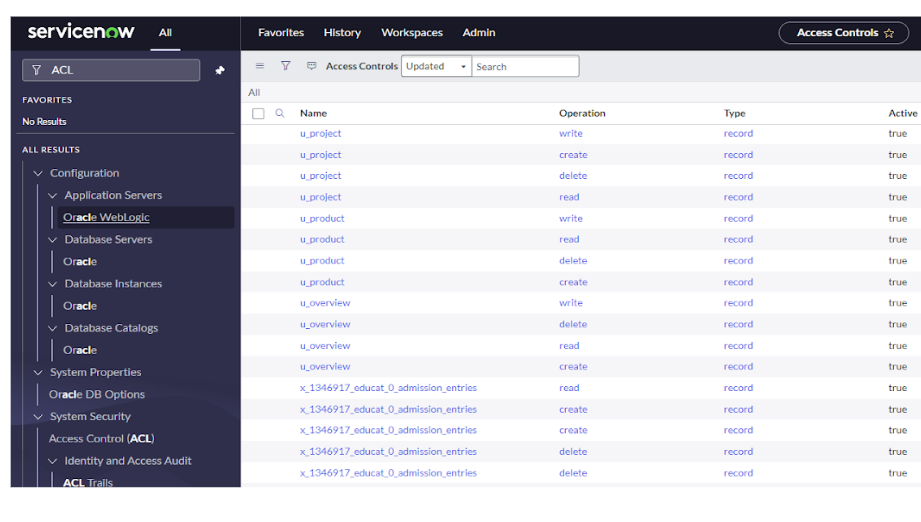
**Activity – 4:**

Then I copied role name(u\_project\_user) from the project table which I have created earlier. Under Project Manager user ID, I have added the u\_project\_user role by editing it.Likewise, I added Employee Role to the Employee Management User.

**Activity – 5:**

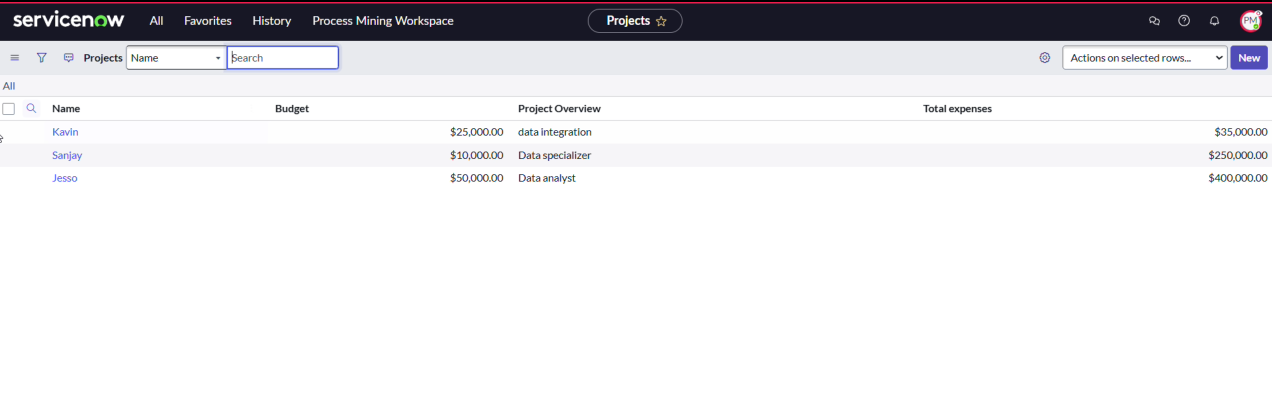
Then In Service Now, I clicked On the Profile Icon>>Elevate Role>>Grant the high security by checking in the check box and clicked save.

Next I searched Access Controls(ACL) in All and clicked on New.



Next,I created read operation table level ACL(none) on Employee role,u\_project\_user(budget),u\_project\_user(Total expenses) and clicked Save.

**Activity 6:**

Then I selected Impersonate User option under profile icon, and clicked Product Management.Then, I went to ALL and searched for Project and clicked New. Then I have created some records with some details as you can see below.

**Key Scenarios Addressed by ServiceNow in the Implementation Project:**

**Role-Based Access Control**: ServiceNow's role-based access control was implemented to ensure that employees can only access applications, modules, and data relevant to their specific job functions. This prevents unauthorized access and simplifies the user experience by eliminating unnecessary options.

**Streamlined User Management**: Through the creation of specific user roles and groups, ServiceNow enabled the management of users in a more organized manner. Each user is assigned appropriate permissions, ensuring they have access to only the resources needed for their tasks.

**Customized Service Request Management**: The application for **Service Request** was created, tailored to the organization’s needs. It allowed users to access the relevant modules (**Create New** and **All**) based on their role, streamlining the service request process.

**Efficient Role and Module Configuration**: By defining roles and modules such as **Create New** and **All**, ServiceNow allowed for the customization of the interface, ensuring that users see only the modules they need, improving workflow and reducing distractions.

**Scalable User and Group Management**: The system’s setup is designed to scale, allowing the addition of new roles, departments, or user groups in the future without disrupting current configurations. This ensures that ServiceNow will continue to meet the evolving needs of the organization.

**Improved Security and Compliance**: With the setup of custom roles and permissions, sensitive data is protected, and employees only have access to the information necessary for their tasks. This strengthens the organization's security measures and helps maintain compliance with internal policies.

Conclusion:

1. **Streamlined User Experience**: By creating specific roles and assigning relevant modules, employees now have access only to the applications and tools necessary for their job functions, reducing unnecessary distractions.
2. **Enhanced Access Control**: I implemented role-based access controls, ensuring that each user can only access the tools and data they need, thereby improving security and protecting sensitive information.
3. **Improved Workflow Efficiency**: The configuration of roles, groups, and modules has simplified the navigation and functionality of ServiceNow, allowing employees to find and use the required tools more quickly.
4. **Scalable Setup**: The structure I created is flexible and scalable, making it easy to add new roles, modules, or departments in the future without disrupting existing processes.
5. **Improved Productivity**: With streamlined access and reduced complexity, employees are now able to complete tasks more efficiently, leading to measurable improvements in overall productivity