# Lab 16: Security

This lab creates a **custom** authentication scheme and an **IS\_ADMIN** authorization scheme.

* Add logon columns to the EMP\_INSUM table.
  + SQL Workshop
  + Object Browser
  + Add the following columns to EMP\_INSUM
    - USERNAME VARCHAR2(50)
    - PASSWORD VARCHAR2(50)
    - IND\_ADMIN NUMBER
* Add text items to page 4 (Create / Edit Employee) to maintain the USERNAME and PASSWORD columns.
  + Edit page 4 (Create / Edit Employee)
  + Center of Page Designer
    - Drag and drop two Text Fields into the Create / Edit Employee Region.
    - Select **P4\_NEW**
  + Right Hand Side
    - Identification
      * Name **P4\_USERNAME**
    - Label
      * Label **Username**
  + Center of Page Designer
    - Select **P4\_NEW\_1**
  + Right Hand Side
    - Identification
      * Name **P4\_PASSWORD**
    - Label
      * Label **Password**
  + Save and Run
  + Add USERNAMEs and PASSWORDs to at least two employees in the EMP\_INSUM table.
* Create a PL/SQL function to authenticate the user.
  + SQL Workshop
  + SQL Commands
  + Run the following code...

CREATE OR REPLACE FUNCTION LOGIN\_EMP\_INSUM (

p\_username IN VARCHAR2

,p\_password IN VARCHAR2)

RETURN BOOLEAN AS

l\_dummy number;

BEGIN

select 1 into l\_dummy

from emp\_insum

where upper(username) = upper(p\_username)

and password = p\_password;

return true;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

return false;

END LOGIN\_EMP\_INSUM;

Create a custom authentication scheme that uses the new PL/SQL function.

* + Application Builder
  + Application Home Page (Training)
  + Shared Components
  + Security
  + Authentication Schemes
  + Create >
  + Method
    - Create Scheme **Based on a pre-configured scheme from the gallery**
  + Name
    - Name **INSUM\_CUSTOM**
    - Scheme Type **Custom**
  + Settings
    - Authentication Function Name **LOGIN\_EMP\_INSUM**
  + Create Authentication Scheme
  + Note that the new Authentication Scheme is “current” by default.
  + Test with the two users that have a username and password
* Make one of the users an administrator
  + SQL Workshop
  + Object Browser
  + EMP\_INSUM Data
  + Edit one of your users: IND\_ADMIN = 1
* Create an Application Item to store the IND\_ADMIN value and a process to load the value.
  + Application Builder
  + Drill into your application
  + Shared Components
  + Application Logic
  + Application Items
  + Create >
  + Name
    - Name **P\_IND\_ADMIN**
  + Security
    - Session State Protection **Restricted - May not be set from browser**
  + Create Application Item
* Create a PL/SQL procedure that sets the value of P\_IND\_ADMIN
  + SQL Workshop
  + SQL Commands
  + Run the following code:

CREATE OR REPLACE PROCEDURE POST\_LOGIN\_EMP\_INSUM AS

l\_ind\_admin number;

BEGIN

select ind\_admin into l\_ind\_admin

from EMP\_INSUM

where upper(username) = upper(v('APP\_USER'));

APEX\_util.set\_session\_state('P\_IND\_ADMIN',l\_ind\_admin);

END POST\_LOGIN\_EMP\_INSUM;

* Register the POST\_LOGIN\_EMP\_INSUM procedure with the Authorization Scheme.
  + Application Builder
  + Drill into your application
  + Shared Components
  + Security
  + Authentication Schemes
  + Edit INSUM\_CUSTOM - Current
  + Login Processing
    - Post-Authentication Procedure Name **POST\_LOGIN\_EMP\_INSUM**
  + Test:
    - Logout
    - Login with your admin user
    - Developer Tool Bar
      * Session
        + View **Application Items** Set
    - Logout
    - Login with your non-admin user
    - Developer Tool Bar
      * Session
        + View **Application Items** Set
* Create an **Authorization** Scheme for the ADMIN role.
  + Application Builder
  + Drill into your application
  + Shared Components
  + Security
  + Authorization Schemes
  + Create >
  + Creation Method
    - Create Authorization Scheme **From Scratch**
  + Details
    - Name **IS\_ADMIN**
    - Scheme Type **PL/SQL Function Returning Boolean**
    - PL/SQL Function Body

**IF :P\_IND\_ADMIN = 1 THEN**

**return true;**

**ELSE**

**return false;**

**END IF;**

* + - Identify error message when scheme violated **You are not an administrator.**
    - Evaluation Point **Once per page view**
  + Create Authorization Scheme
* Now restrict the application so that only administrators can edit employees using the Employee tabular form (page 5). NOTE: this is a **two-step** process because you must secure the page itself **and** the navigation links that point to the page.
* 1) Apply the IS\_ADMIN authorization scheme to the page (this is the most important step).
  + Edit Page 5 (Employees)
  + Left Hand Side
    - Select **Page 5: Employees**
  + Right Hand Side
    - Security
      * Authorization Scheme **IS\_ADMIN**
  + Save and run
  + Test
* 2) Apply the IS\_ADMIN authorization scheme to the navigation link that points to the page.
  + Shared Components
  + Navigation
  + Navigation Menu
  + Edit **Desktop Navigation Menu**
  + Edit **Employees**
    - Authorization
      * Authorization Scheme **IS\_ADMIN**
  + Run as a non-administrator
  + Test
  + Run as an administrator
  + Test