

Suite à de nombreuses tentatives malveillantes, ESPRIT a décidé de renforcer la sécurité des données.

1. Assurer que les entrées d'audit seront stockées dans la table système de BDs (SQLBIND et SQLTEXT sont renseignées).
2. Créez un utilisateur « **ESPRIT** ». Attribuez-lui les privilèges pour se connecter, créer des tables, déclencheurs et des procédures. Sous ESPRIT, créer des tables avec quelques manipulations (insertion, sélection, MAJ et suppression).
3. Lister toutes les opérations effectuées par l'utilisateur « **ESPRIT** ».
4. On veut auditer :
  - a. La création d'utilisateur et la création de directory par accès.
  - b. Les tentatives de connexion par accès.
  - c. L'insertion et la suppression sur la table '**REGIONS**' du schéma HR par session pour tous les utilisateurs.
  - d. Toute mise à jour effectuées au niveau des tables par session.
  - e. Toutes les modifications de structures apportées aux objets qui vont être créés.
5. Lancer l'audit pour l'utilisateur « **ESPRIT** » afin de surveiller :
  - a. GRANT DIRECTORY et GRANT TABLE par l'utilisateur « **ESPRIT** » par accès si la requête est bien exécutée.
  - b. Toutes les instructions LDD et LMD
  - c. Toute instruction de création de déclencheurs effectuée avec succès par l'utilisateur « **ESPRIT** »
  - d. La suppression des synonymes dans n'importe quel schéma par l'utilisateur « **ESPRIT** »
6. Enlevez l'audit de select et de suppression de table '**REGIONS**' du schéma HR et vérifiez les options.
7. Créer une procédure stockée « **PROC\_AUDIT\_TRACE** » qui affiche les traces d'audit de création d'utilisateurs. On souhaite afficher l'utilisateur et la date de lancement de la commande.
8. Créer une fonction stockée « **FN\_AUDIT\_CONN** » qui retourne le nombre de connexions/déconnexions pour chaque utilisateur passé en paramètre. Tester avec l'utilisateur « **ESPRIT** ».
9. Ecrire une procédure stockée « **AUDIT\_OPTS (table varchar)** » qui permet d'afficher les options d'audit Pour une table passée en paramètre sur les commandes d'insertion, de mise à jour et de sélection. Tester avec la table '**REGIONS**'.
10. Créer une fonction stockée « **FN\_SQL (Nowner varcahr2, Ntable varchar2)** » qui prend en paramètre le nom d'une table et le nom de son propriétaire et permet de retourner le nombre toutes les requêtes exécutées sur la table.