**Unit Outcomes:**

* Examine Application Security concepts.
* Examine Security Layers.
* Modify the database so that tables are protected.

**Database-2**: Implement a database schema with security and optimization.

**Purpose**

In this assignment you will continue to improve on the assignment from last week.

**Assignment Instructions**

* You will continue to practice using a layered approach to building software in this Assignment.
* First, make copy of the Unit 2 application and use that as a starting point for the Unit 3 application.
* You will want to store this as the baseline for your Unit 3 application in your source code management system you picked in Unit 1.
* Add a login screen that also asks for the data needed to construct the connection string. Those items are the server, database, user and password. These data will get passed into the business layer on construction.
* Add the ability to access three tables.
* Display the record count and create a list of data from each table.
* Create a SQL script that prepares the database by creating three users, each allowed a separate role.
* The script must also grant permissions to allow some users more access than others.
* Add new methods into the business layer.
* Create methods to return the number of employees and orders in the same fashion as customers.
* Include a constructor that accepts the four data items used to create a connection string.
* Modify the data layer to make use of the two other tables.
* Remember that the methods return raw data, not a table structure.

Reminders:

* The presentation layer only talks to the business layer.
* The business layer will do the actual interaction with the data access layer.
* All of the interactions with the data layer are performed within the business layer.

**Assignment Requirements**

1. Using the Northwind database, prepare a sql script to do the following:
   1. Create roles/groups for security in the database.
      * SalesRole
      * HRRole
      * CEORole
   2. Modify access for those roles/groups.
      * Grant SalesRole the ability to view Orders and Customers table
      * Grant HRRole the ability to view Employee table
      * Grant CEORole the ability to view Orders, Customers and Employee tables.
   3. Create three new database users:
      * User\_CEO
      * User\_HR
      * User\_Sales
   4. Add each user to the appropriate role\group.
      * User\_Sales to SalesRole
      * User\_HR to HRRole
      * User\_CEO to CEORole
   5. Save SQL statements as IT481\_Unit3\_YourName.SQL

Take all of your source code and save it to the source code management system.

* Remember that this is separate from the Unit 2 Project.
* Save a screenshot to prove you have completed this step, and save it in a Word document. Name this IT481\_Unit3\_YourName.doc
  1. Take all of your source code and save it to the source code management system.
  2. Remember that this is separate from the Unit 2 Project. Save a screenshot to prove you have completed this step, and save it in a Word document. Name this IT481\_Unit3\_YourName.doc
  3. Modify your application to require user credentials.
     1. Add a GUI interface to require user credentials.
     2. Add error handling to your application.
     3. Add the capability to view more than one table.
  4. Test your app, using each of the three users:
     1. User\_CEO should be able to view all tables.
     2. User\_HR can only see Employees.
     3. User\_Sales can only view Customers and Orders.
     4. Bad login data should be trapped.

Place all of your project code, the SQL script, and the screenshot into a zip file named FirstName\_LastName\_zip\_Unit3