W03 Scenario:

Your new manager has many concerns about database security. After reading about a recent data breach in the news, it is one of the things that keeps him up at night. Help calm your manager's nerves by investigating how to control database security more precisely.

W03 Scenario Submission Guidelines:

Be sure you submit all elements labeled by the bolded word, **SHOW**.

NOTE:

- If you have doubts about these security tasks, the best approach is to experiment. **TRY** it, **GRANT** it, **LOGIN** (with the new account), and **TEST** it.

1. The owner of the company has some security questions for you…
   a. What is Windows authentication in SQL Server? What is one benefit to Windows Authentication over SQL Authentication?
   b. Explain the difference between authentication and authorization. Give an example of authorization in the database.
   c. Verify your SQL Server installation is in mixed authorization mode and can accept both Windows and SQL Server Authentication.
   d. What would happen if you grant SELECT permission on a table to the fixed database role called '*public*?' Would this granted permission apply to future users also (users that are not created yet)? Why could this be dangerous? **HINT**: Look under 'Fixed Database Roles' in Chapter 12 or here in the Microsoft documentation.

   SHOW 1: Your understanding by answering these questions confidently. Use the textbook or Microsoft documentation to verify your answers.

2. You have heard that using 'schemas' can give you added flexibility and control in database security. You decide to test this by doing the following:
   a. Create two new schemas for the Bowling database and two more for an additional database of your choice. You will be creating four schemas total.
   b. Transfer the tables in the bowling database and your chosen database out of the dbo (database owner) schema and into the four new schemas. How you choose to separate the tables into these schemas is completely up to you (you will not be graded on that choice). **NOTE**: Tables can only belong to one schema.
   c. Create four new logins and map them to each database (two for each database). Issue a grant command that will give SELECT rights on an entire schema (one for each user). Do this for each of the four logins. Test this authorization by logging in with these new users.

HINT: Remember that a “login” is on the instance/server level and is used for authentication. Each login can map to one or more users in each of the databases in the instance. Logins receive instance/server level authorizations. Users receive database/schema/table/etc level authorizations. Consider users to be under the umbrella of a login.

SHOW 2:

I. The four new schemas you created and the process you used to do so.
II. The process you used to transfer the tables into the four new schemas.
III. Proof that each of the four logins can access the schema intended and no other schemas.

3. With “user-defined roles,” determine a common level of authorization privileges new users should have in ***one database*** of your choice. This may be different for each business model (database) according to your discretion.
    a. First, you decide to create a list of DCL (Data Control Language or “GRANT commands”) to assign to every future entry-level user of a given database. You can choose whatever you would like for the users to be authorized to do.
    HINT: Here is a student example of practicing DCL for two test users.

    b. Then, you get smarter and realize you can use a user-defined role as explained in chapter 12 instead of issuing so many separate GRANTS for each individual user. Create a role for new employees and grant the permissions you listed in ‘3a’ directly to the new role instead.
    HINT: Here is the student example from this week’s preparation post on how to use a fixed role for permissions. In your case, you will instead add users to the custom role you create.

    c. Create two new database logins/users and add them as members to the new role from ‘3b’ **instead** of granting the permissions one by one directly to the users. In this manner, you save time and are less error prone.

    SHOW 3:

    I. All DCL code (“GRANT” statements) from step ‘a’ above.
    II. The process you used to create a role with the needed DCL authorization commands instead.
    III. Proof that the new role works as it should for your two new logins/users.

4. Investigate these security related data dictionary entries (or others you may find) to see where you can see evidence of the new schemas, logins, users, or role from this assignment in the data dictionary.

    SHOW 4:

I. A query and results that include data dictionary information showing evidence of something you did in this assignment (perhaps a query that shows a new schema, login, or user-defined role you created in steps 1, 2, or 3).

II. One additional data dictionary query regarding *anything* in database security that might be useful to the business going forward. Include the query, the results, and your explanation for why it would be a useful security report.