Database sikkerhed & adgang 2.HF

It & Data, Odense

ANNL/april 2019

SYDDANSK ERHVERVSSKOLE

Sikkerhed / SQL injection

Når der arbejdes med databaser / OOP, skal der tænkes sikkerhed ind i koden.

Nogle gode råd:

- Anvend server bruger/login
- Begræns adgang til det nødvendige
- Opret få bruger med administrator rettigheder
- Undgå sa konto
- Anvend parametre
- Brug Views / Stored procedures
- Kryptering af kodeord
- Anvendelse af token – evt. med tidsbegrænsning

Hvad er SQL injection?

- SQL injektion er kode skrevet ind i input felter
- Hvor koden f.eks. giver uønsket adgang til databasen
- Eller udfører ondsindet kommandoer
  - Sletter data
  - Ændre data

Læs mere på:

https://www.w3schools.com/Sql/sql_injection.asp

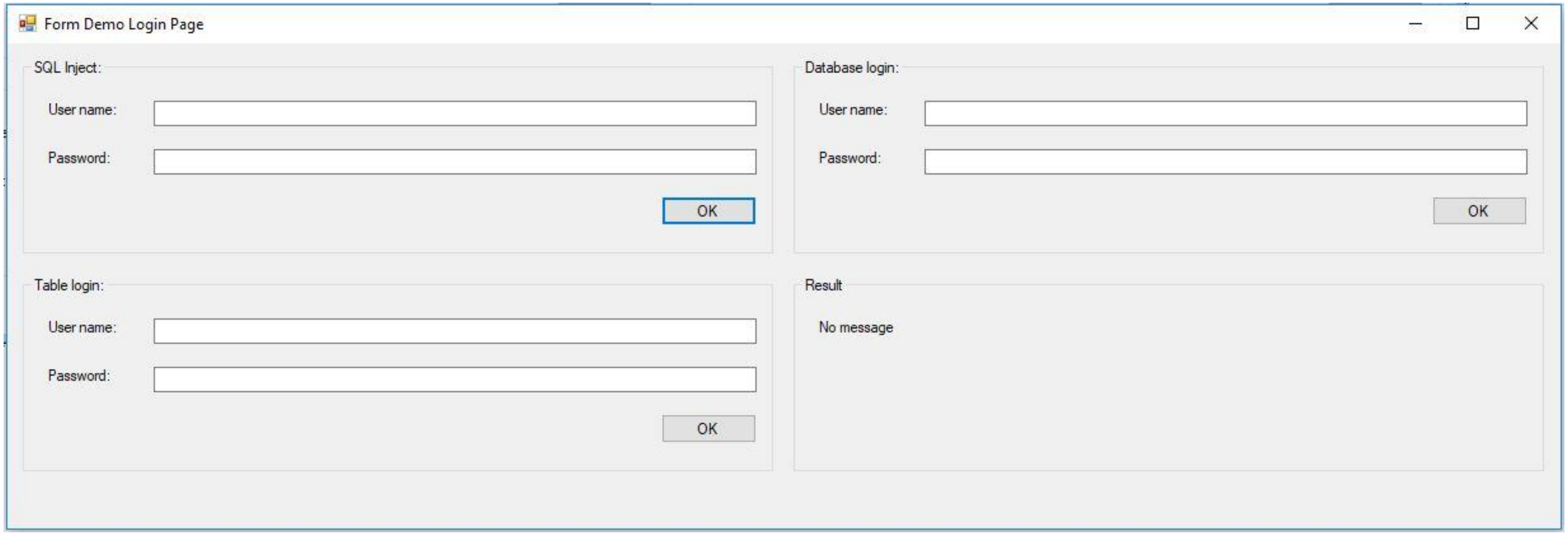
https://www.acunetix.com/websitesecurity/sql-injection/

https://www.acunetix.com/blog/articles/exploiting-sql-injection-example/

https://www.perspectiverisk.com/mysql-sql-injection-practical-cheat-sheet/

https://www.veracode.com/security/sql-injection

Demonstration



Form Demo Login Page

SQL Inject:

User name:

Password:

Database login:

User name:

Password:

Table login:

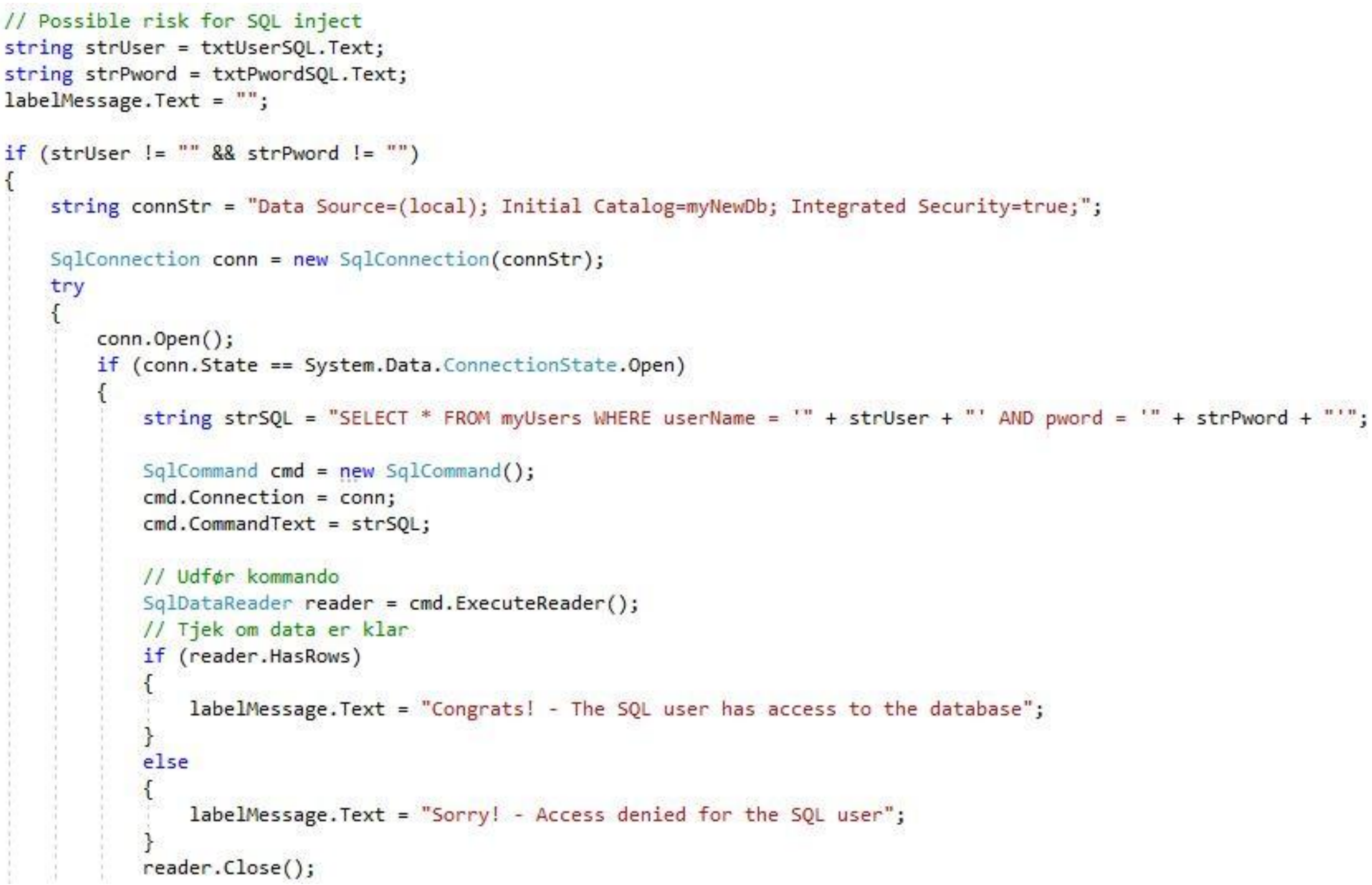
User name:

Password:

Result

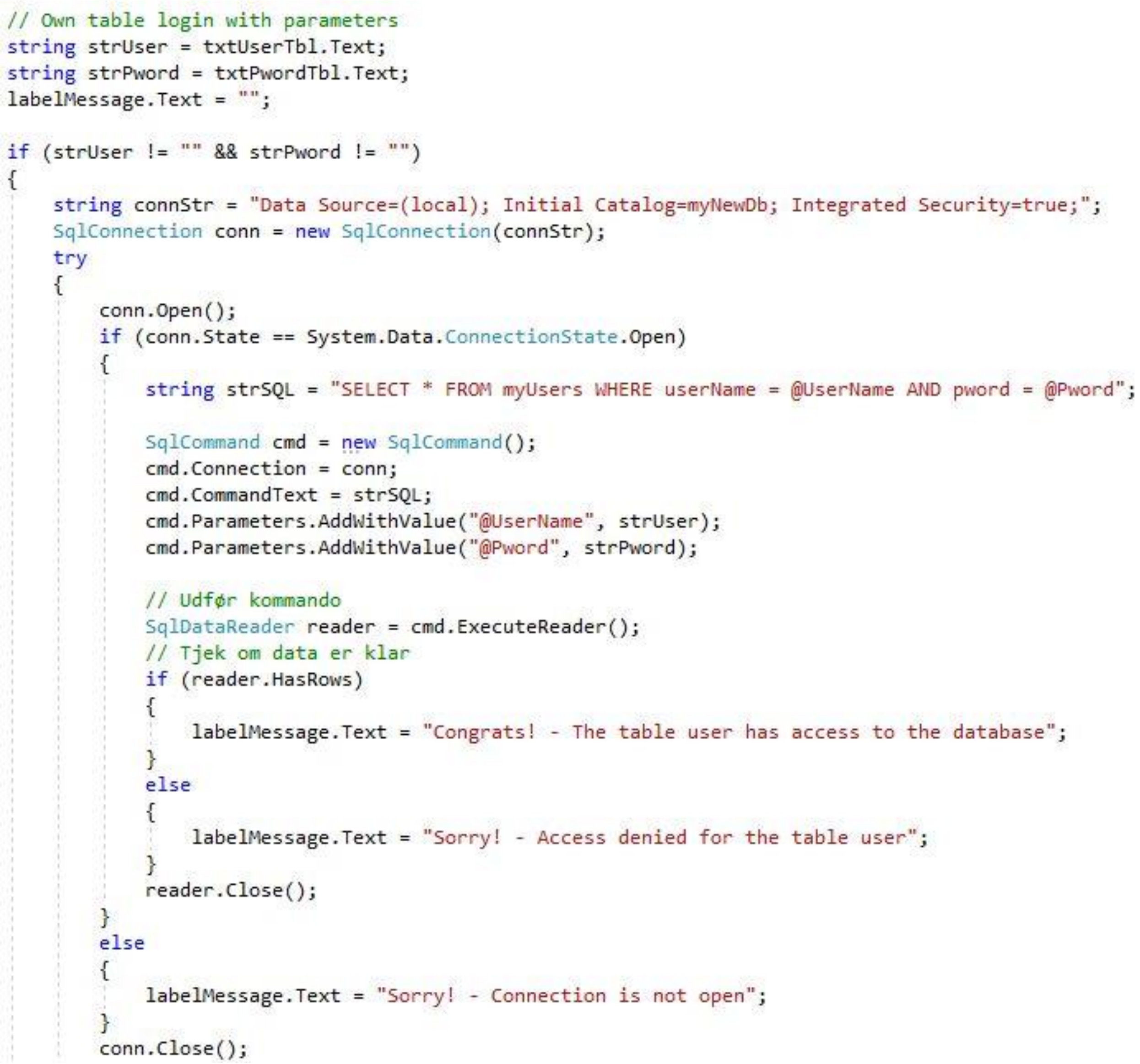
No message

Potentiel risiko for SQL injektion



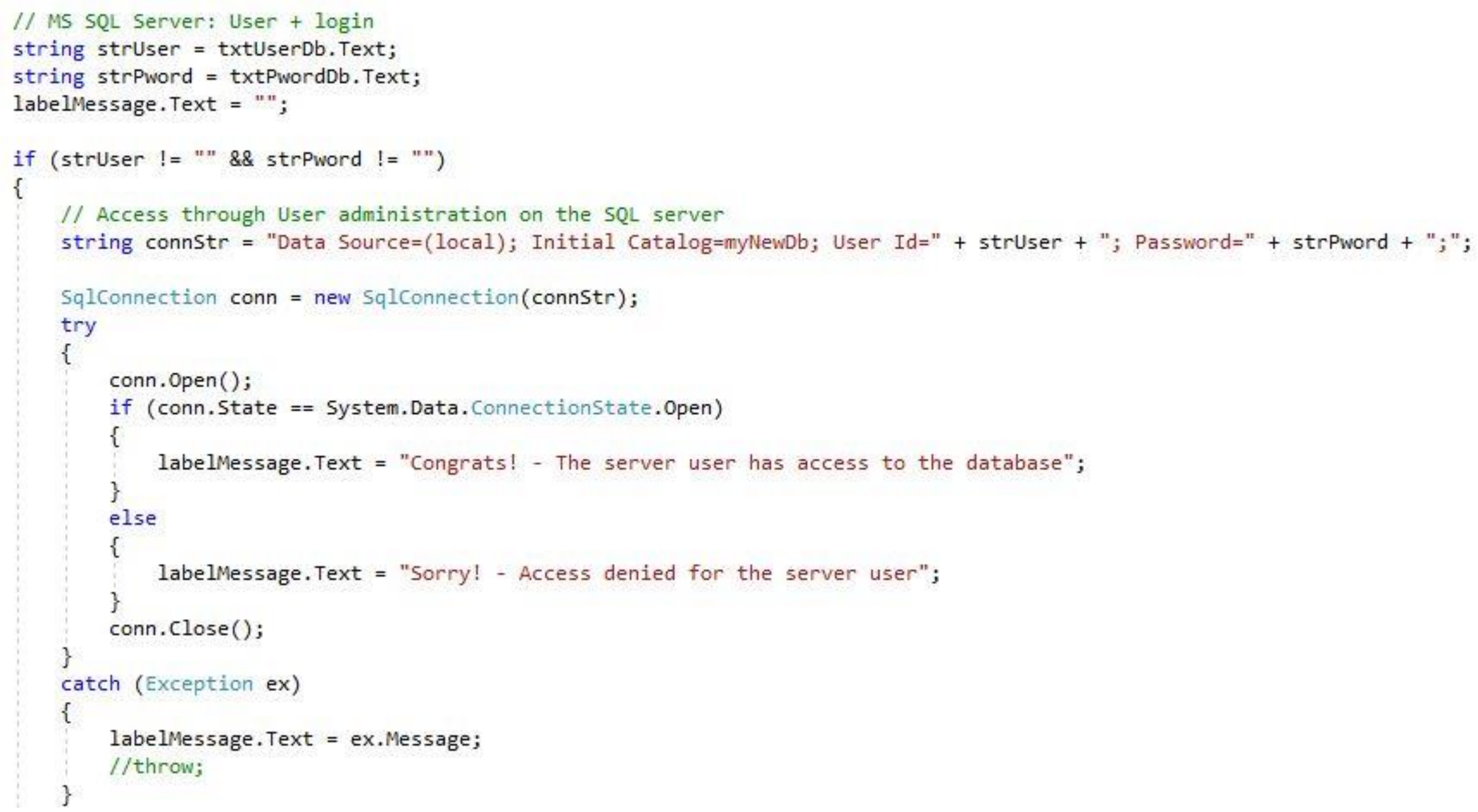
```
// Possible risk for SQL inject
string strUser = txtUserSQL.Text;
string strPword = txtPwordSQL.Text;
labelMessage.Text = "";

if (strUser != "" && strPword != "")
{
    string connStr = "Data Source=(local); Initial Catalog=myNewDb; Integrated Security=true;";

    SqlConnection conn = new SqlConnection(connStr);
    try
    {
        conn.Open();
        if (conn.State == System.Data.ConnectionState.Open)
        {
            string strSQL = "SELECT * FROM myUsers WHERE userName = '" + strUser + "' AND pword = '" + strPword + "'";

            SqlCommand cmd = new SqlCommand();
            cmd.Connection = conn;
            cmd.CommandText = strSQL;

            // Udfør kommando
            SqlDataReader reader = cmd.ExecuteReader();
            // Tjek om data er klar
            if (reader.HasRows)
            {
                labelMessage.Text = "Congrats! - The SQL user has access to the database";
            }
            else
            {
                labelMessage.Text = "Sorry! - Access denied for the SQL user";
            }
            reader.Close();
```

Samme login med parametre



```
// Own table login with parameters
string strUser = txtUserTbl.Text;
string strPword = txtPwordTbl.Text;
labelMessage.Text = "";

if (strUser != "" && strPword != "")
{
    string connStr = "Data Source=(local); Initial Catalog=myNewDb; Integrated Security=true;";
    SqlConnection conn = new SqlConnection(connStr);
    try
    {
        conn.Open();
        if (conn.State == System.Data.ConnectionState.Open)
        {
            string strSQL = "SELECT * FROM myUsers WHERE userName = @UserName AND pword = @Pword";

            SqlCommand cmd = new SqlCommand();
            cmd.Connection = conn;
            cmd.CommandText = strSQL;
            cmd.Parameters.AddWithValue("@UserName", strUser);
            cmd.Parameters.AddWithValue("@Pword", strPword);

            // Udfør kommando
            SqlDataReader reader = cmd.ExecuteReader();
            // Tjek om data er klar
            if (reader.HasRows)
            {
                labelMessage.Text = "Congrats! - The table user has access to the database";
            }
            else
            {
                labelMessage.Text = "Sorry! - Access denied for the table user";
            }
            reader.Close();
        }
        else
        {
            labelMessage.Text = "Sorry! - Connection is not open";
        }
        conn.Close();
```

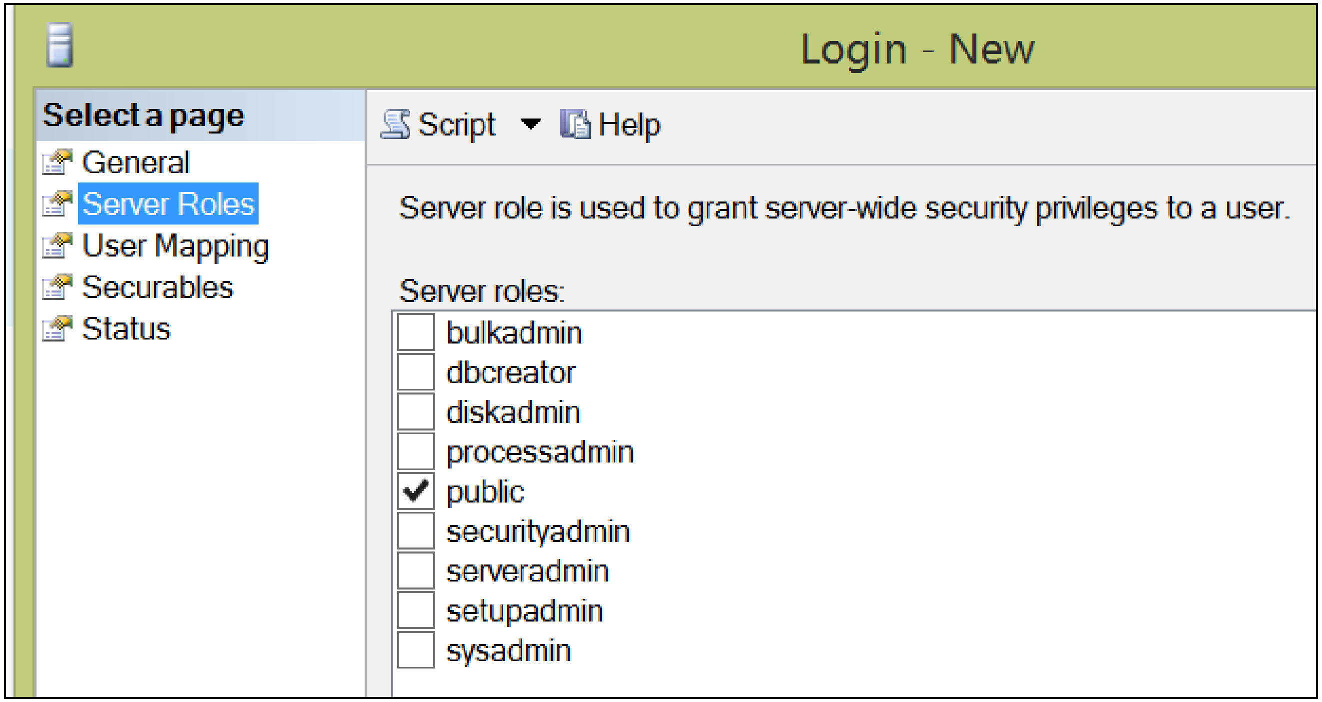
Database bruger administration



```
// MS SQL Server: User + login
string strUser = txtUserDb.Text;
string strPword = txtPwordDb.Text;
labelMessage.Text = "";

if (strUser != "" && strPword != "")
{
    // Access through User administration on the SQL server
    string connStr = "Data Source=(local); Initial Catalog=myNewDb; User Id=" + strUser + "; Password=" + strPword + ";";

    SqlConnection conn = new SqlConnection(connStr);
    try
    {
        conn.Open();
        if (conn.State == System.Data.ConnectionState.Open)
        {
            labelMessage.Text = "Congrats! - The server user has access to the database";
        }
        else
        {
            labelMessage.Text = "Sorry! - Access denied for the server user";
        }
        conn.Close();
    }
    catch (Exception ex)
    {
        labelMessage.Text = ex.Message;
        //throw;
    }
```

Genopfrisk: Login / Server Roller

- *"Server Roles"* hjælper dig med at administrere tilladelserne på serveren

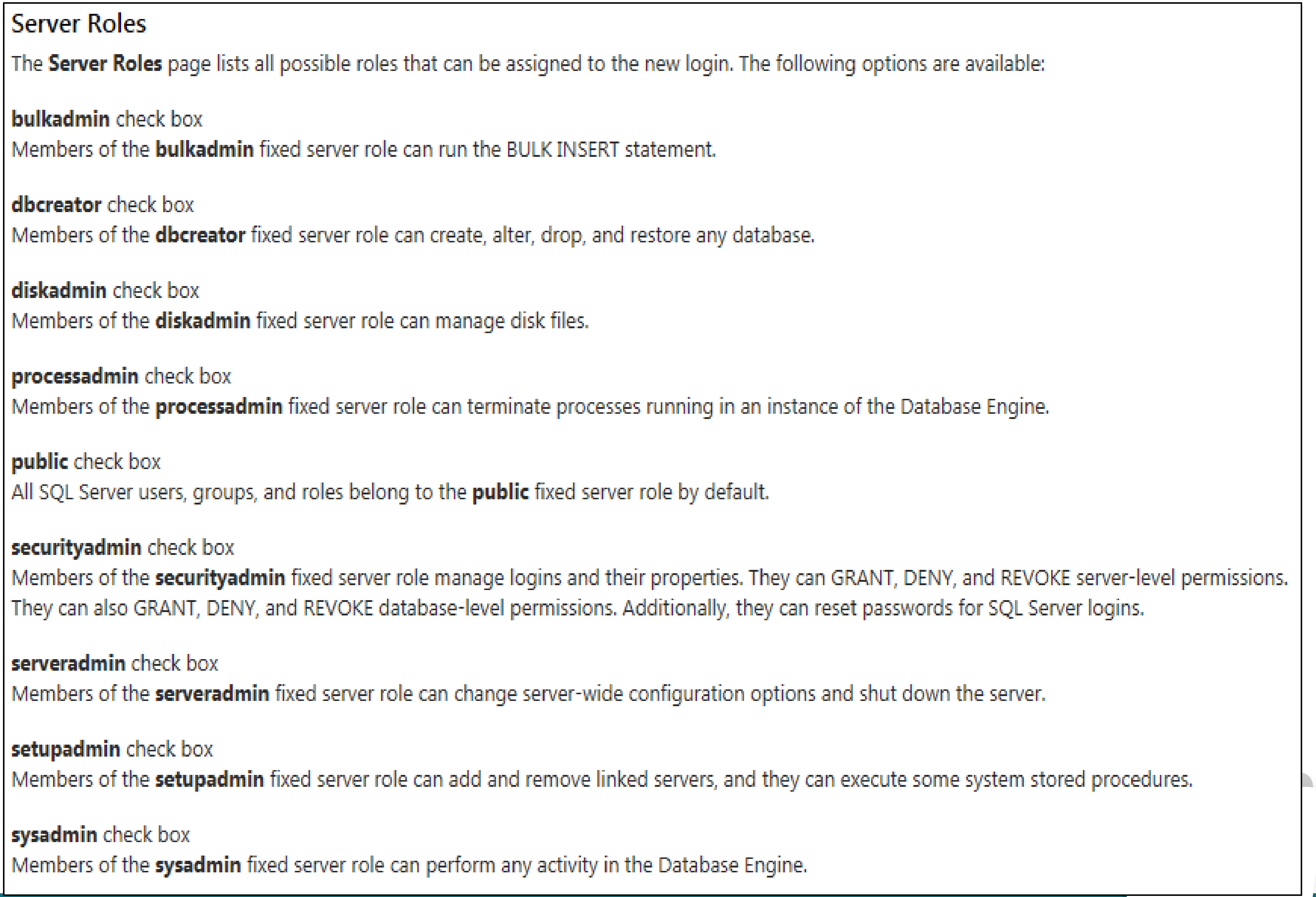


Server role is used to grant server-wide security privileges to a user.

Server roles:

- □ bulkadmin
- □ dbcreator
- □ diskadmin
- □ processadmin
- ☑ public
- □ securityadmin
- □ serveradmin
- □ setupadmin
- □ sysadmin

Server Roles / Administration



Server Roles

The **Server Roles** page lists all possible roles that can be assigned to the new login. The following options are available:

bulkadmin check box
Members of the **bulkadmin** fixed server role can run the BULK INSERT statement.

dbcreator check box
Members of the **dbcreator** fixed server role can create, alter, drop, and restore any database.

diskadmin check box
Members of the **diskadmin** fixed server role can manage disk files.

processadmin check box
Members of the **processadmin** fixed server role can terminate processes running in an instance of the Database Engine.

public check box
All SQL Server users, groups, and roles belong to the **public** fixed server role by default.

securityadmin check box
Members of the **securityadmin** fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server-level permissions. They can also GRANT, DENY, and REVOKE database-level permissions. Additionally, they can reset passwords for SQL Server logins.

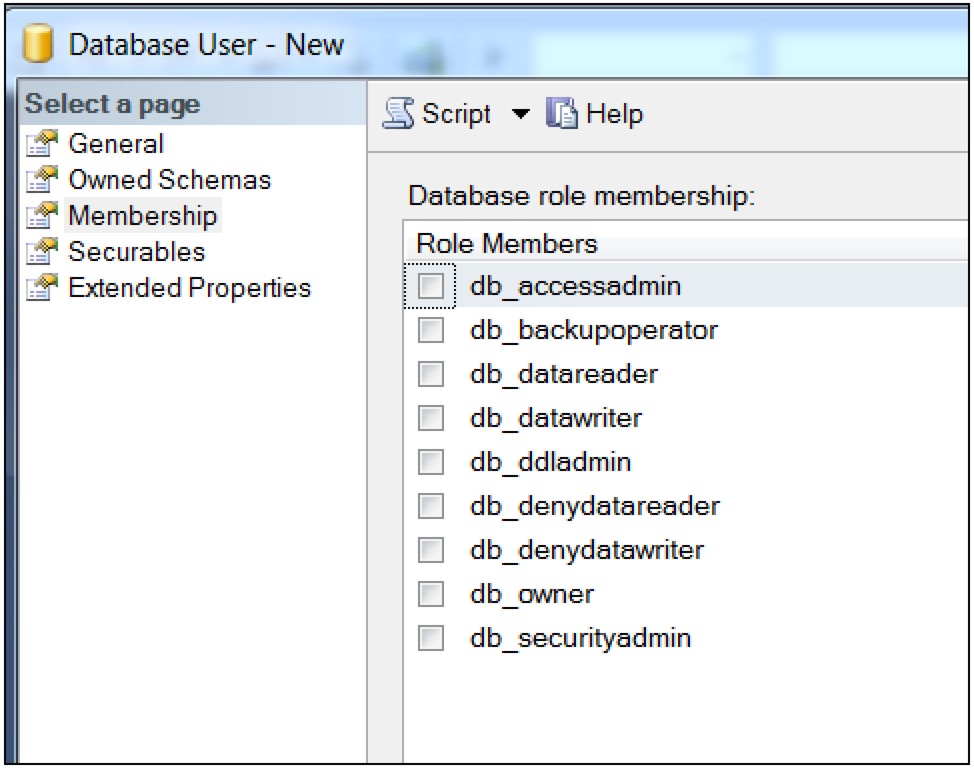
serveradmin check box
Members of the **serveradmin** fixed server role can change server-wide configuration options and shut down the server.

setupadmin check box
Members of the **setupadmin** fixed server role can add and remove linked servers, and they can execute some system stored procedures.

sysadmin check box
Members of the **sysadmin** fixed server role can perform any activity in the Database Engine.

Database User

- Database roles hjælper dig med at administrere tilladelserne på selve databasen.



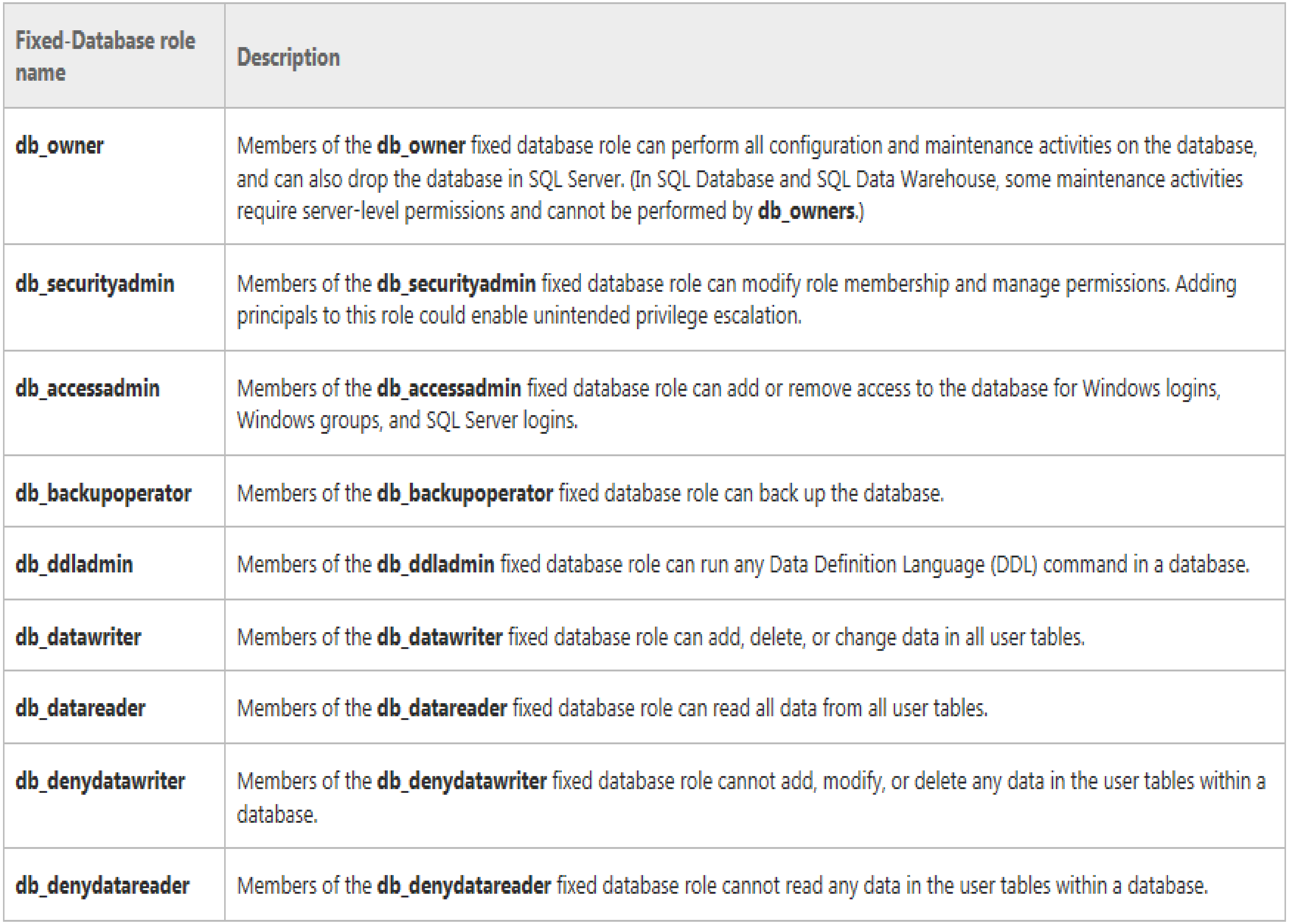
Database User - New

Database role membership:

Role Members
□ db_accessadmin
□ db_backupoperator
□ db_datareader
□ db_datawriter
□ db_ddladmin
□ db_denydatareader
□ db_denydatawriter
□ db_owner
□ db_securityadmin

- Her man ydereligere også give rettigheder til, at man f.eks. kun skal kunne se views eller køre stored procedure osv.

Database Roles



Fixed-Database role name	Description
db_owner	Members of the **db_owner** fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database in SQL Server. (In SQL Database and SQL Data Warehouse, some maintenance activities require server-level permissions and cannot be performed by **db_owners**.)
db_securityadmin	Members of the **db_securityadmin** fixed database role can modify role membership and manage permissions. Adding principals to this role could enable unintended privilege escalation.
db_accessadmin	Members of the **db_accessadmin** fixed database role can add or remove access to the database for Windows logins, Windows groups, and SQL Server logins.
db_backupoperator	Members of the **db_backupoperator** fixed database role can back up the database.
db_ddladmin	Members of the **db_ddladmin** fixed database role can run any Data Definition Language (DDL) command in a database.
db_datawriter	Members of the **db_datawriter** fixed database role can add, delete, or change data in all user tables.
db_datareader	Members of the **db_datareader** fixed database role can read all data from all user tables.
db_denydatawriter	Members of the **db_denydatawriter** fixed database role cannot add, modify, or delete any data in the user tables within a database.
db_denydatareader	Members of the **db_denydatareader** fixed database role cannot read any data in the user tables within a database.

Sikkerhed / Adgang