**Überblick über DCL-Befehle und Logins/Benutzer in SQL Server (mit "MeinGeschaeftDo"**

**Server Authentifizierung Arten**

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Berechtigungen und deren Bedeutung:**

In SQL Server werden Berechtigungen auf Datenbankobjekte (wie Tabellen, Sichten, gespeicherte Prozeduren) vergeben, um zu steuern, welche Aktionen Benutzer ausführen dürfen.

**2. Erstellen von Logins und Benutzern in der Datenbank "MeinGeschaeftDo"**

Bevor wir mit der Verwaltung von Berechtigungen fortfahren, müssen wir sicherstellen, dass die richtigen Logins und Benutzer existieren. Ein **Login** erlaubt es einem Benutzer, sich bei SQL Server anzumelden, während ein **Benutzer** es ermöglicht, auf eine bestimmte Datenbank zuzugreifen und darauf zu operieren.

**2.1 Erstellen eines SQL Server Logins**

**SQL Server-Authentifizierung:**

CREATE LOGIN Benutzer1 WITH PASSWORD = 'SicheresPasswort2025';

**Windows-Authentifizierung:**

CREATE LOGIN [Domain\Benutzer1] FROM WINDOWS;

**2.2 Erstellen eines Benutzers in der Datenbank "MeinGeschaeftDo"**

Nun, nachdem wir das Login erstellt haben, müssen wir einen Benutzer in der Datenbank **"MeinGeschaeftDo"** erstellen, der mit diesem Login verknüpft wird.

USE MeinGeschaeftDo;

CREATE USER Benutzer1 FOR LOGIN Benutzer1;

**3. Zuweisen von Berechtigungen (GRANT)**

Mit dem **GRANT**-Befehl können wir Benutzern oder Rollen Berechtigungen für Datenbankobjekte zuweisen. In dieser Übung geht es darum, Berechtigungen auf Tabellen wie Kunden, Bestellungen, Produkte und andere, die in Ihrer **"MeinGeschaeftDo"**-Datenbank enthalten sind, zu vergeben.

**3.1 Beispiel: Lesezugriff auf die Tabelle Kunden**

Gewähren Sie dem Benutzer Benutzer1 nur Lesezugriff auf die Tabelle Kunden:

GRANT SELECT ON dbo.Kunde TO Benutzer1;

**Erklärung:**  
Der Benutzer Benutzer1 erhält das Recht, alle Daten in der Tabelle Kunde zu lesen.

**3.2 Beispiel: Schreibzugriff auf Bestellungen gewähren**

Gewähren Sie dem Benutzer Benutzer1 Schreibzugriff (INSERT) auf die Tabelle Bestellung:

GRANT INSERT ON dbo.Bestellungen TO Benutzer1;

**Erklärung:**  
Der Benutzer Benutzer1 darf neue Bestellungen in die Tabelle Bestellungen einfügen.

**3.3 Beispiel: Zugriff auf mehrere Tabellen**

Gewähren Sie dem Benutzer Benutzer1 auf mehrere Tabellen Zugriffsrechte:

GRANT SELECT, INSERT, UPDATE ON dbo.Kunde TO Benutzer1;

GRANT SELECT, DELETE ON dbo.Bestellung TO Benutzer1;

GRANT SELECT ON dbo.Produkt TO Benutzer1;

**Erklärung:**

* Der Benutzer Benutzer1 darf die Tabellen Kunden und Produkte lesen, die Tabelle Bestellungen sowohl lesen als auch löschen und in der Tabelle Kunden neue Datensätze einfügen oder bestehende Datensätze ändern.

**4. Verweigern von Berechtigungen (DENY)**

Der **DENY**-Befehl wird verwendet, um einem Benutzer bestimmte Berechtigungen zu verweigern. Wenn Sie einem Benutzer eine Berechtigung verweigern, können sie diese Berechtigung nicht einmal erhalten, wenn sie ihr später durch einen GRANT-Befehl zugewiesen wird.

**4.1 Beispiel: Verweigern des Löschens von Bestellungen**

Verweigern Sie dem Benutzer Benutzer1 das Löschen von Bestellung:

DENY DELETE ON dbo.Bestellung TO Benutzer1;

**Erklärung:**  
Der Benutzer Benutzer1 darf keine Datensätze in der Tabelle Bestellungen löschen, selbst wenn er später mit GRANT auch Löschrechte erhält.

**4.2 Beispiel: Verweigern des Zugriffs auf die Tabelle Produkt**

Verweigern Sie dem Benutzer Benutzer1 den Zugriff auf die Tabelle Produkt:

DENY SELECT ON dbo.Produkt TO Benutzer1;

**Erklärung:**  
Der Benutzer Benutzer1 kann keine Daten aus der Tabelle Produkt lesen.

**5. Entziehen von Berechtigungen (REVOKE)**

Mit dem **REVOKE**-Befehl können Sie Berechtigungen entziehen, die einem Benutzer entweder durch GRANT oder DENY zugewiesen wurden.

**5.1 Beispiel: Entziehen der Berechtigung zum Einfügen von Bestellungen**

REVOKE INSERT ON dbo.Bestellung TO Benutzer1;

**Erklärung:**  
Der Benutzer Benutzer1 verliert das Recht, neue Bestellungen in die Tabelle Bestellungen einzufügen.

**5.2 Beispiel: Entziehen der Berechtigung zum Aktualisieren der Tabelle Kunden**

REVOKE UPDATE ON dbo.Kunde TO Benutzer1;

**Erklärung:**  
Der Benutzer Benutzer1 verliert das Recht, Daten in der Tabelle Kunden zu aktualisieren.

**6. Übungsaufgaben**

1. **Aufgabe 1: Erstellen Sie ein neues Login und einen neuen Benutzer.**
   * Erstellen Sie ein Login namens Mitarbeiter1 mit einem sicheren Passwort.
   * Erstellen Sie einen Benutzer Mitarbeiter1 in der Datenbank **"MeinGeschaeftDo"**.
2. **Aufgabe 2: Gewähren Sie Zugriff auf die Tabelle Kunde.**
   * Gewähren Sie dem Benutzer Mitarbeiter1 Lese- und Schreibzugriff auf die Tabelle Kunden.
3. **Aufgabe 3: Verweigern Sie den Zugriff auf die Tabelle Bestellung.**
   * Verweigern Sie dem Benutzer Mitarbeiter1 das Recht, die Tabelle Bestellungen zu lesen.
4. **Aufgabe 4: Entziehen Sie alle Berechtigungen für Mitarbeiter1.**
   * Entfernen Sie alle Berechtigungen, die Sie dem Benutzer Mitarbeiter1 gewährt haben (einschließlich INSERT und UPDATE auf Kunden).
5. **Aufgabe 5: Erstellen Sie eine Rolle Manager und weisen Sie dieser Rolle Berechtigungen zu.**
   * Erstellen Sie eine Rolle Manager, die Lese- und Schreibzugriff auf die Tabellen Kunden und Bestellungen hat.
   * Fügen Sie den Benutzer Mitarbeiter1 zu dieser Rolle hinzu.

**7. Zusammenfassung**

* Mit **GRANT**, **DENY** und **REVOKE** können Sie sehr spezifische Berechtigungen auf Benutzer und Rollen anwenden.
* **GRANT** gewährt Berechtigungen, **DENY** verweigert sie, und **REVOKE** entzieht bereits gewährte Berechtigungen.
* Logins sind notwendig, damit sich Benutzer bei SQL Server anmelden können. Benutzer ermöglichen den Zugriff auf Datenbanken.
* Rollen bieten eine einfache Möglichkeit, Berechtigungen für mehrere Benutzer zu verwalten.

**Beispiel Setzung von Berechtigungen in Gespeicherte Prozeduren**

Ein Bild, das Text, Screenshot, Display, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.