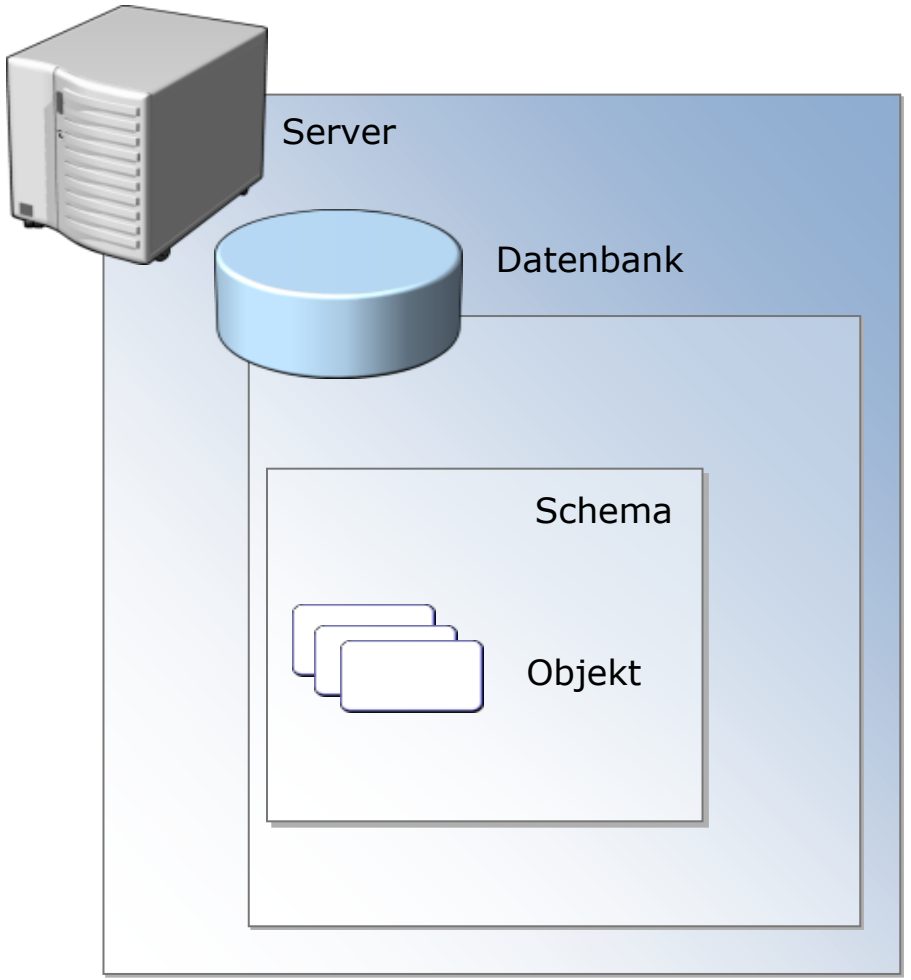
Anmeldungen, Rollen und Benutzer

Objekthierarchie

- Objekte im SQL-Server sind auf einer von drei Ebenen angesiedelt: Severebene, Datenbankebene, Schemaebene

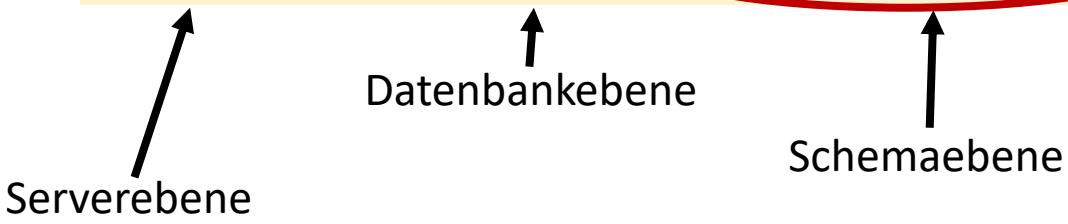
- Jedes Objekt muss auf genau einer der drei Ebenen existieren



Schema

- Benannter Container (Gruppierung) für Datenbankobjekte
- Vierteilige Benennunssyntax

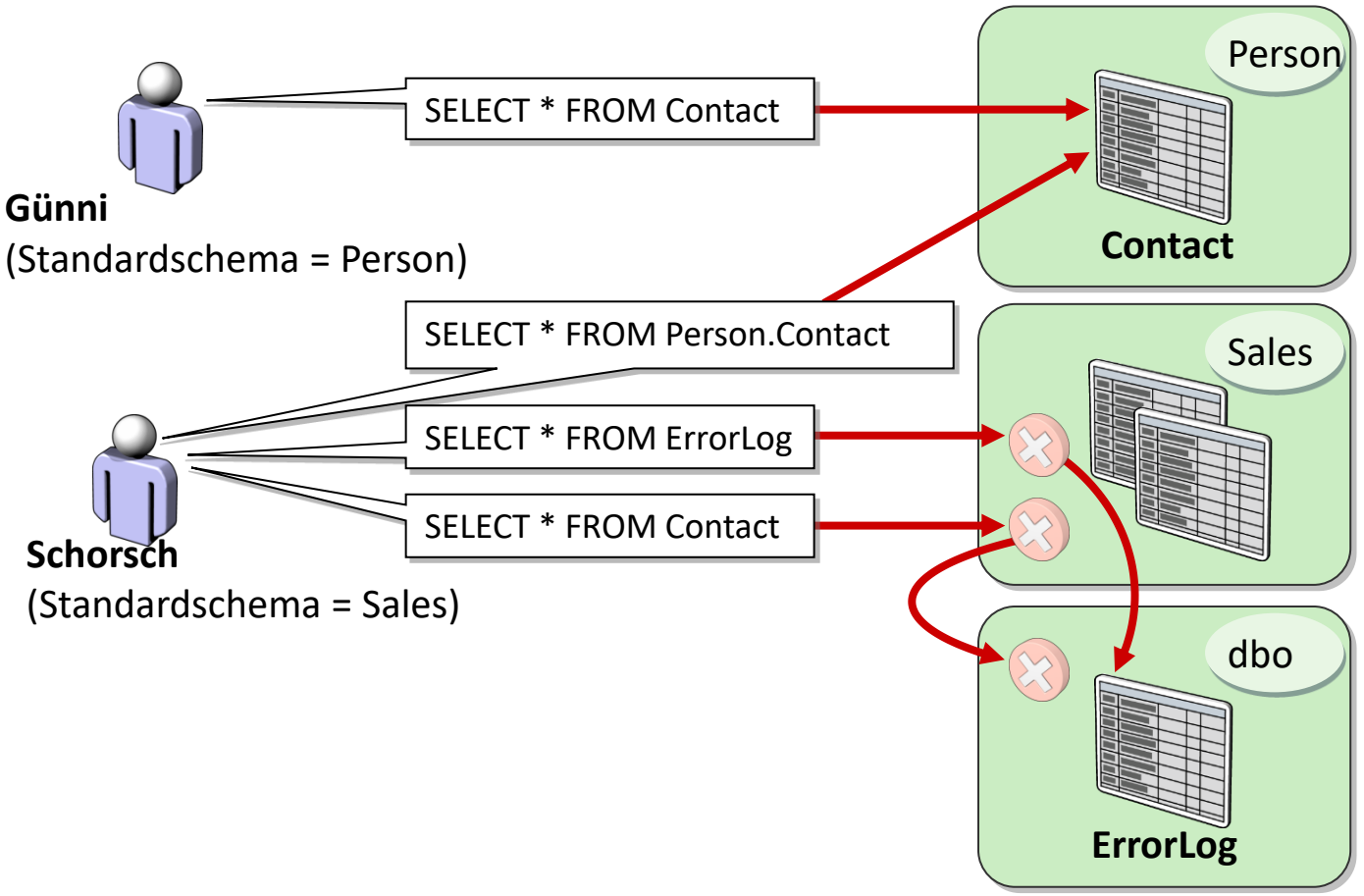
```
Server.Database.DatabaseScheme.DatabaseObject
```



- Schemata können Prinzipal gehören und Prinzipale können Schema zugeordnet sein
- Im SQL Server integrierte Schemata ***dbo, guest, sys, INFORMATION_SCHEMA*** (letzten beiden Systemreserviert)

Objekte suchen / finden

- Ein Prinzipal ist einem oder mehreren Schemata zugeordnet
- Die Schemata entscheiden über die Benutzbarkeit von Datenbankobjekten



Schema erstellen

- Syntax

```
CREATE SCHEMA {<Schemaname> | AUTHORISATION <Besitzername> |
<Schemaname> AUTHORISATION <Besitzername> }
[(<Tabellendefinition> | <Sichtdefinition>) |
GRANT | REVOKE | DENY <Anweisung>];
```

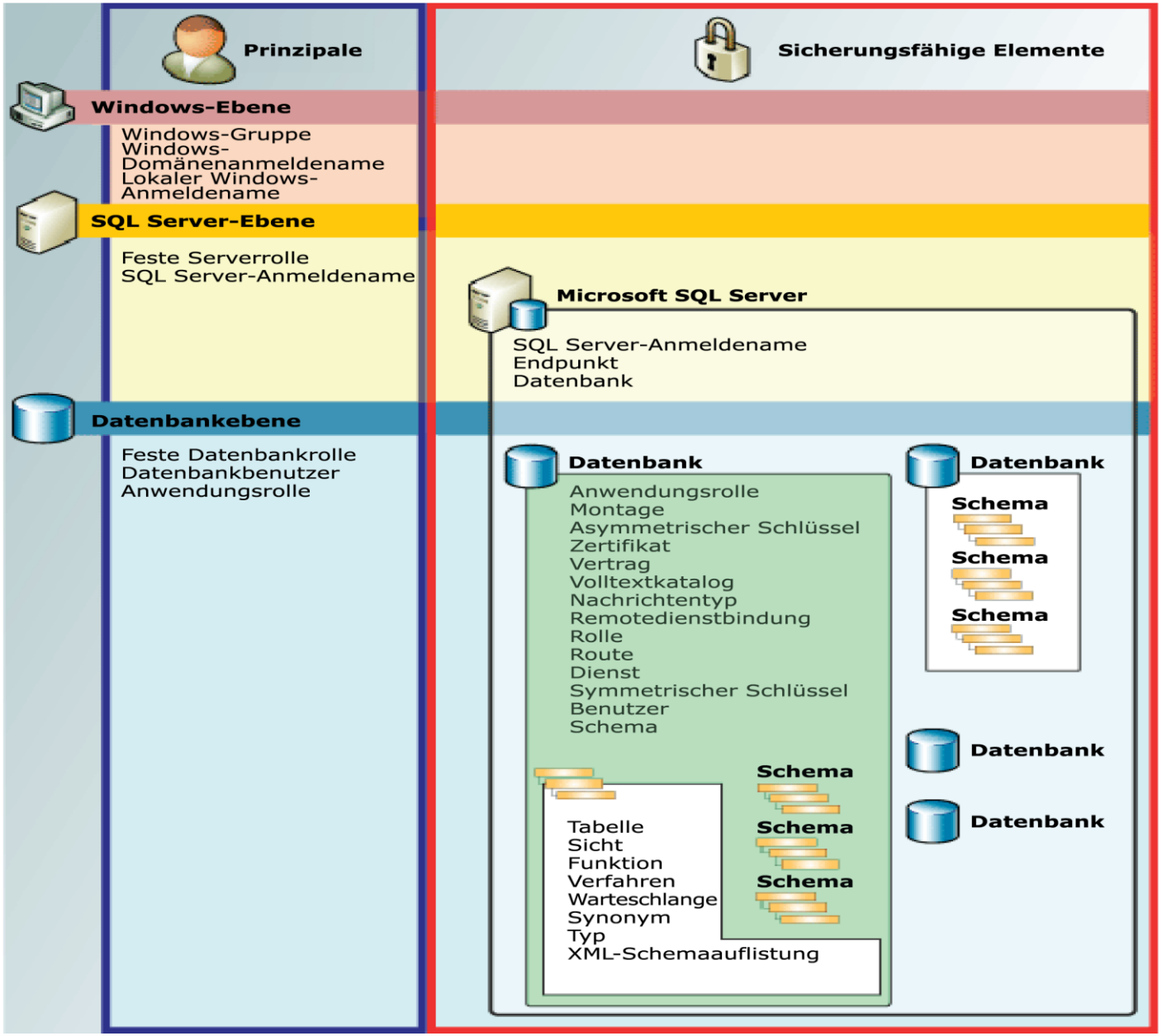
- Beispiel

```
CREATE SCHEMA Personal
AUTHORIZATION MrBean
```

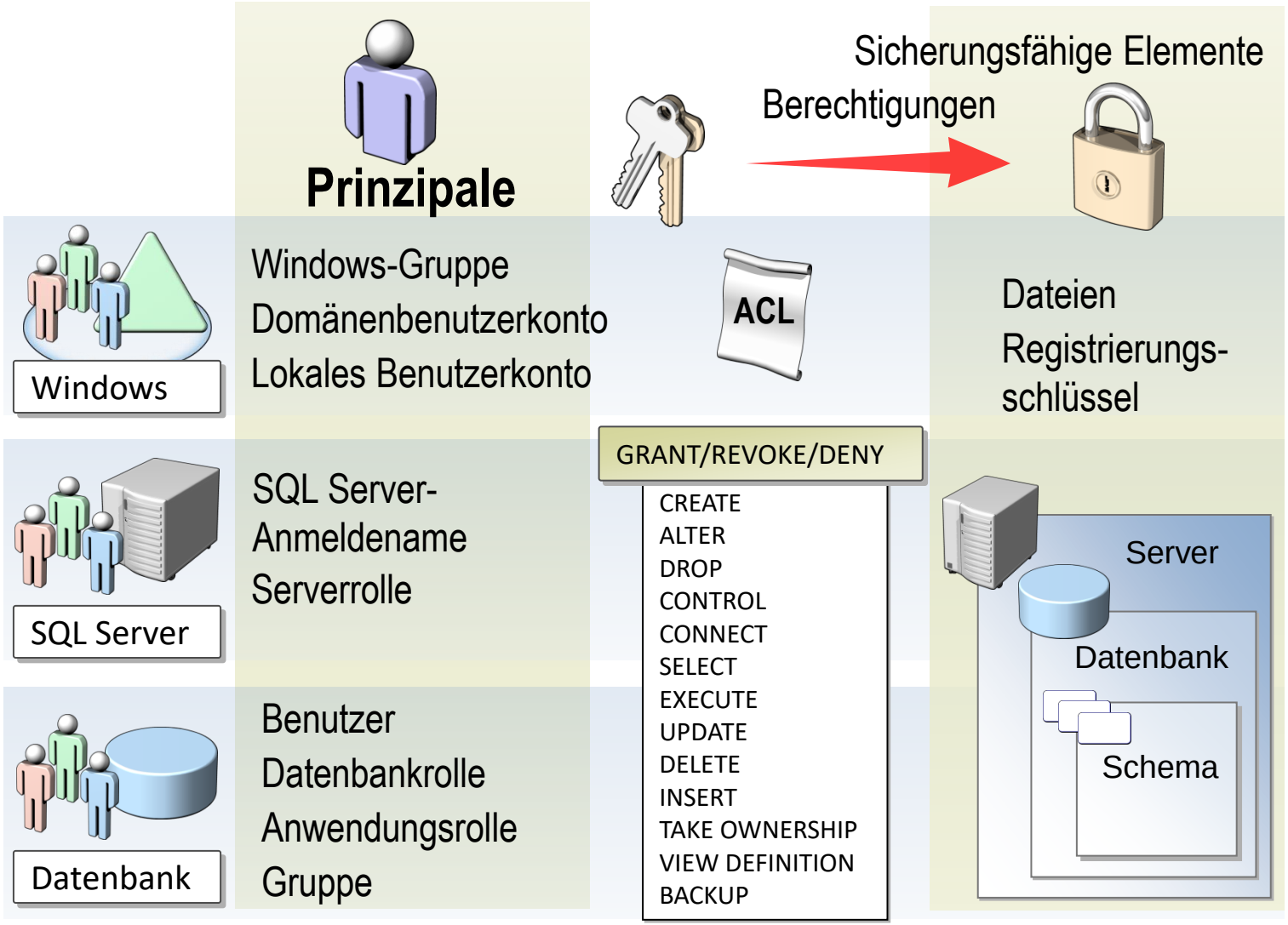
Sicherheits-Framework

Drei Hauptebenen, auf welchen Prizipale angelegt werden können.

- Windows-Ebene
- SQL Server-Ebene
- Datenbankebene



Sicherheits-Framework

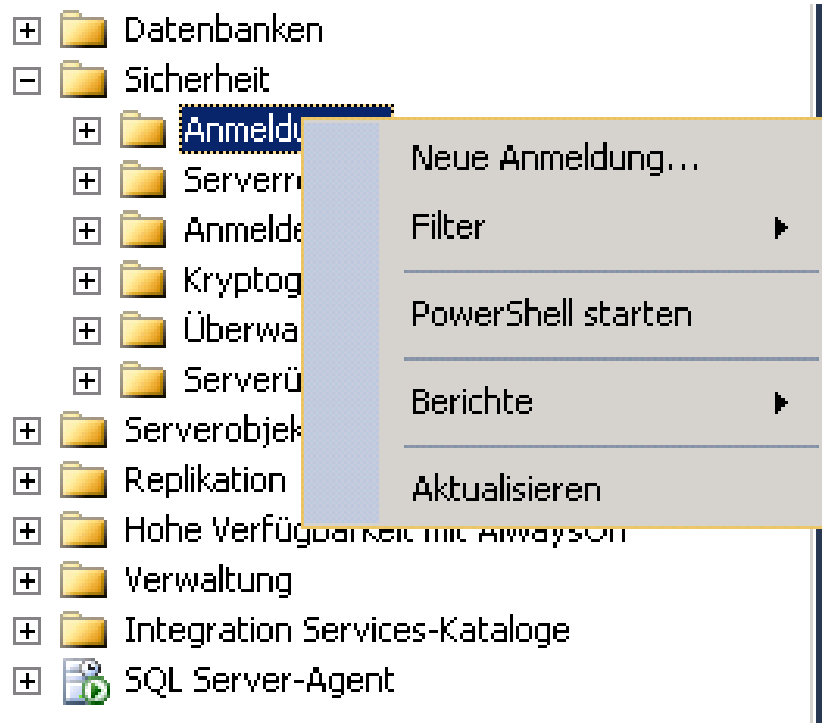


Windows Anmeldekonten

- Mit T-Sql:

```
CREATE LOGIN [AdventureWorks\Student]
FROM WINDOWS
WITH DEFAULT_DATABASE=[tempdb],
     DEFAULT_LANGUAGE=[us_english];
GO
CREATE LOGIN
[AdventureWorks\Salespeople]
FROM WINDOWS;
GO
```

- Mit Objekt-Explorer



Entfernen mit DROP LOGIN, Fehlermeldung falls Benutzer gerade angemeldet

SQL-Server Anmeldekonten

```
CREATE LOGIN MrBoo
WITH PASSWORD = 'P@ssw0rd',
CHECK_POLICY = ON;
GO
CREATE LOGIN MrBoo
WITH PASSWORD = 'P@ssw0rd',
CHECK_POLICY = OFF;
```

- CHECK_POLICY: Gültigkeit der Kennwortrichtlinie
- SQL-Server muss SQL-Server-Anmeldungen zulassen!

- Verwenden von ALTER LOGIN um Kennwörter zurückzusetzen und Anmeldungen aktivieren/deaktivieren

Authentifizierung und Autorisierung

Authentifizierung und Autorisierung werden häufig verwechselt

Authentifizierung

Ist die Überprüfung der Identität eines Prinzipals (beispielsweise das Ermitteln der Identität einer Person)

Autorisierung

Ist die Zuweisung von Berechtigungen für ein sicherungsfähiges Element für einen Prinzipal (beispielsweise die Entscheidung, welche Berechtigungen eine Person hat)

Kann durch das Zuweisen eines Prinzipals zu einer Rolle implementiert werden, die bereits über Berechtigungen verfügt

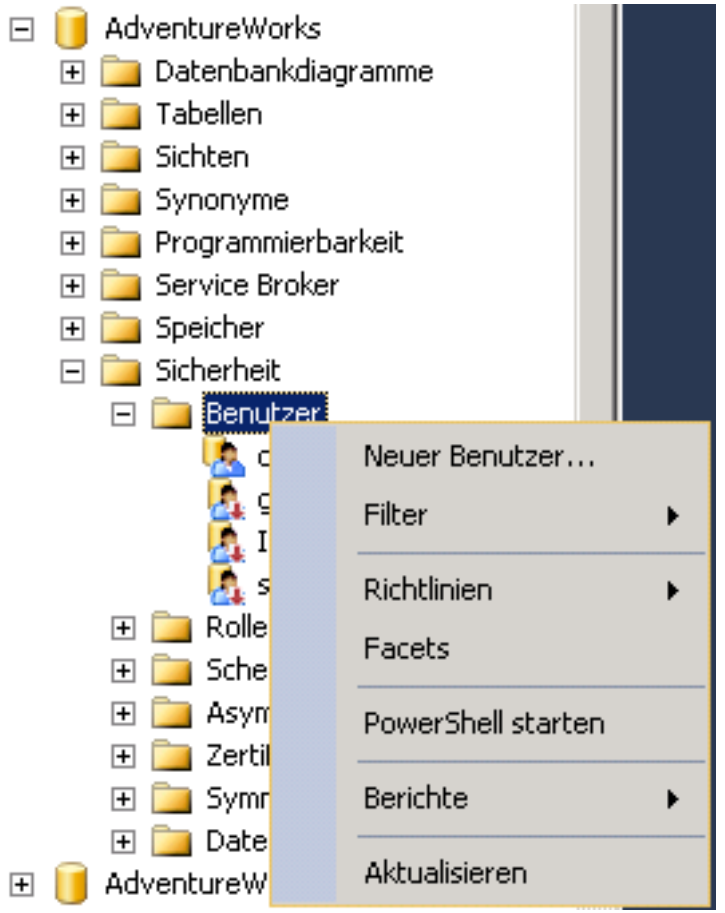
Wird über die Berechtigungen GRANT, DENY oder REVOKE für Berechtigungen für Datenbankobjekte implementiert

Beispiel: Sie möchten Geld von Ihrem Konto an einem Geldautomaten abheben. An diesem werden Sie zunächst aufgefordert, sich mithilfe Ihrer Karte und Ihrer dazugehörigen Geheimzahl zu authentifizieren. Anschließend geben Sie einen Betrag ein, welchen Sie abheben möchten. Dazu prüft das System, ob Sie autorisiert sind, diesen Betrag zu erhalten (Tageslimit, Dispo, Kontosperre etc.).

Steuern des Zugriffs auf Datenbanken

- Zugriff durch Erstellung von Datenbankbenutzern

```
CREATE USER Mumpitz
FOR LOGIN MrBoo;
GO
CREATE USER Student
FOR LOGIN [Northwind\Student];
GO
CREATE USER HRApp
FOR LOGIN HRUser;
GO
```



Berechtigungen auf Serverebene

- Berechtigungen als „feste Serverrolle“, „benutzerdefinierte Serverrolle“ oder bestimmte Berechtigungen für den Serverbetrieb
- Minimieren der Verwendung von festen Serverrollen

```
USE master;
GO
GRANT ALTER ON LOGIN::HRApp
TO [Northwind\Holly];
GO
GRANT ALTER ANY DATABASE
TO [Northwind\Holly];
GO
```

Typische Berechtigungen auf Serverebene

Die aktuelle Datenbank muss die master-Datenbank sein, wenn Berechtigungen im Serverbereich zugewiesen werden

Alle Rechte auf Server-Ebene werden durch Abfragen der Ansicht **sys.server_permissions** angezeigt

Typische Berechtigungen auf Serverebene	
ALTER ANY DATABASE	ALTER TRACE
BACKUP DATABASE	BACKUP LOG
CONNECT SQL	CONTROL SERVER
CREATE DATABASE	SHUTDOWN
VIEW ANY DEFINITION	VIEW SERVER STATE

Feste Serverrollen

Rolle	Beschreibung	Berechtigung auf Serverebene
sysadmin	Kann beliebige Aktionen ausführen	CONTROL SERVER (mit GRANT-Option)
dbcreator	Kann Datenbanken erstellen und ändern	ALTER ANY DATABASE
diskadmin	Kann Datenträgerdateien verwalten	ALTER RESOURCES
serveradmin	Kann serverweite Einstellungen konfigurieren	ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
securityadmin	Kann Serveranmeldungen verwalten und überwachen	ALTER ANY LOGIN
processadmin	Kann SQL Server-Prozesse verwalten	ALTER ANY CONNECTION ALTER SERVER STATE
bulkadmin	Kann die BULK INSERT-Anweisung ausführen	ADMINISTER BULK OPERATIONS
setupadmin	Kann Replikationen und verknüpfte Server konfigurieren	ALTER ANY LINKED SERVER

Serverrolle ‘public‘

- Besondere Rolle; Berechtigungen können geändert werden

Benutzerdefinierte Serverrollen

- Strengere Berechtigungssteuerung, sollte anstelle von festen Serverrollen verwendet werden

```
USE master;
GO
CREATE SERVER ROLE bebop;
GO
```

- Erteilen der Berechtigungen mithilfe des SSMS oder mit T-SQL

Berechtigungen auf Datenbankebene

- Berechtigungen als „feste Datenbankrolle“, „benutzerdefinierte Datenbankrolle“ oder bestimmte Berechtigungen für den Datenbankbereich

- Minimieren der Verwendung von festen Datenbankrollen

```
Use Northwind;
GO
GRANT CREATE TABLE TO HRManager;
GO
GRANT VIEW DEFINITION TO James;
GO
```

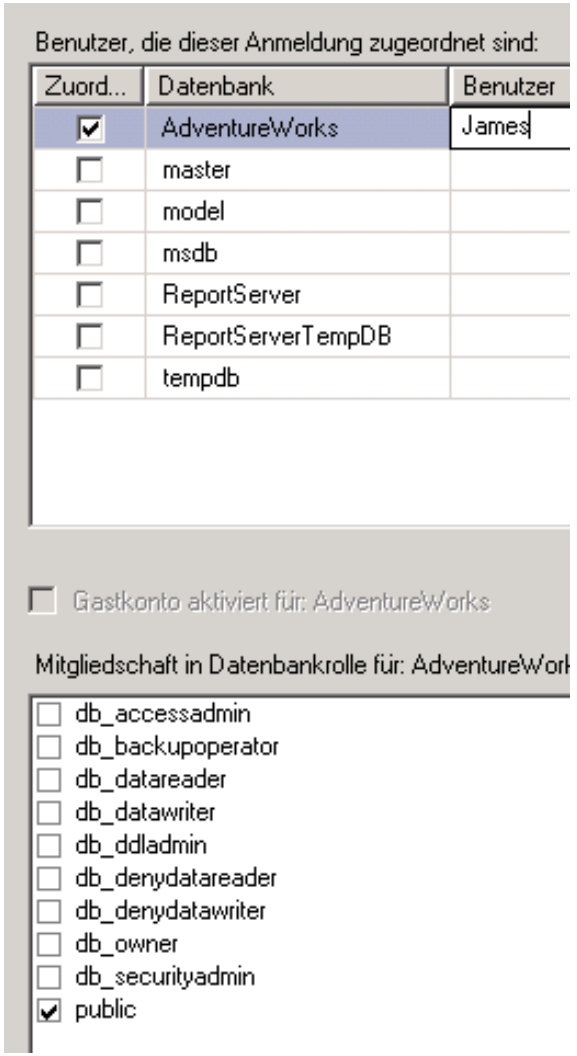
Feste Datenbankrollen

Rolle	Beschreibung
db_owner	Ausführen aller Aktivitäten zum Konfigurieren und Warten der Datenbank und Löschen der Datenbank
db_securityadmin	Ändern der Rollenmitgliedschaft und Verwalten von Berechtigungen
db_accessadmin	Hinzufügen oder Entfernen des Zugriffs auf die Datenbank für Anmeldungen
db_backupoperator	Sichern der Datenbank
db_ddladmin	Ausführen eines beliebigen DDL-Befehls in der Datenbank
db_datawriter	Hinzufügen, Löschen oder Ändern von Daten in allen Benutzertabellen
db_datareader	Lesen aller Daten aus allen Benutzertabellen
db_denydatawriter	Kein Hinzufügen, Löschen oder Ändern von Daten in Benutzertabellen
db_denydatareader	Kein Lesen von Daten in Benutzertabellen

Zuweisen von Rollen zu Benutzern

- Benutzer können Rollen zugewiesen werden
  - Verwendet GUI
  - Verwenden von T-SQL

```
Use Northwind;
GO
ALTER SERVER ROLE db_datareader
      ADD MEMBER James;
GO
```



Datenbankbesitzer

- dbo
  Die Anmeldung sa und Mitglieder der sysadmin-Rolle werden gemeinsam mit dem Datenbankbesitzer dem dbo-Konto zugeordnet

Arbeiten mit benutzerdefinierten Datenbankrollen

- Datenbankrollen können verwaltet werden (erstellt, geändert, gelöscht)
- Rollen haben Besitzer, der Rolle werden Berechtigungen erteilt, Berechtigungen werden von Rollenmitgliedern geerbt

```
CREATE ROLE MarketingReaders
AUTHORIZATION dbo;
GO
GRANT SELECT ON SCHEMA::Marketing
TO MarketingReaders;
GO
```

Beispiel

Typisches Szenario

- Definieren Sie dbo-Benutzer und andere Administratorrollen
- Definieren Sie Berechtigungsgruppen innerhalb der Datenbank
- Berücksichtigen Sie die Verwendung der öffentlichen Rolle für allgemeine Berechtigungen
- Erstellen Sie Rollen, und weisen Sie Ihnen Berechtigungen zu
- Fügen Sie Rollen Benutzer hinzu

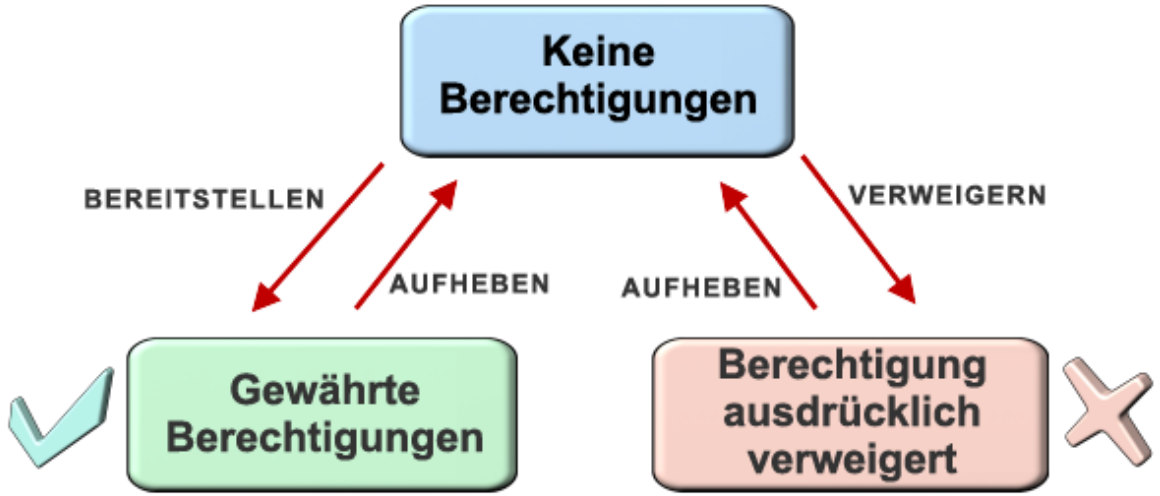
Für die Entscheidungsfindung innerhalb des Codes

- IS_SRVROLEMEMBER, IS_MEMBER

```
IF IS_MEMBER('BankManagers') = 0
BEGIN
      PRINT 'Operation is only for Bank Managers';
      ROLLBACK;
END;
```

GRANT, REVOKE, DENY

- GRANT wird verwendet, um eine Berechtigung zuzuweisen
- DENY wird verwendet, um eine Berechtigung explizit zu verweigern
  - Wird verwendet, wenn Berechtigungen durch Gruppen- oder Rollenmitgliedschaft vererbt werden
  - Sollte nur in Ausnahmefällen verwendet werden
- REVOKE entfernt entweder GRANT oder DENY



Sichern von Tabellen und Sichten

- Einige Objektberechtigungen gelten für Tabellen und Sichten
  - SELECT
  - INSERT, UPDATE, DELETE

```
GRANT SELECT ON OBJECT::Marketing.Salesperson
TO HRApp;
GO
GRANT SELECT ON Marketing.Salesperson
TO HRApp;
GO
```

Berechtigungen auf Spaltenebene

- Berechtigungen können auf Spaltenebene zugewiesen werden
- Mehrere Spaltenberechtigungen können in einer einzelnen Anweisung zugewiesen werden
- Eine GRANT-Berechtigung auf Spaltenebene überschreibt eine DENY-Berechtigung auf Tabellenebene

```
GRANT SELECT ON Marketing.Salesperson(SalespersonID, EmailAdr)
TO James;
GO
DENY SELECT ON Marketing.Salesperson TO Holly;
GO
GRANT SELECT ON Marketing.Salesperson (SalespersonID, FirstName,
Lastname) TO Holly;
GO
```

WITH GRANT OPTION

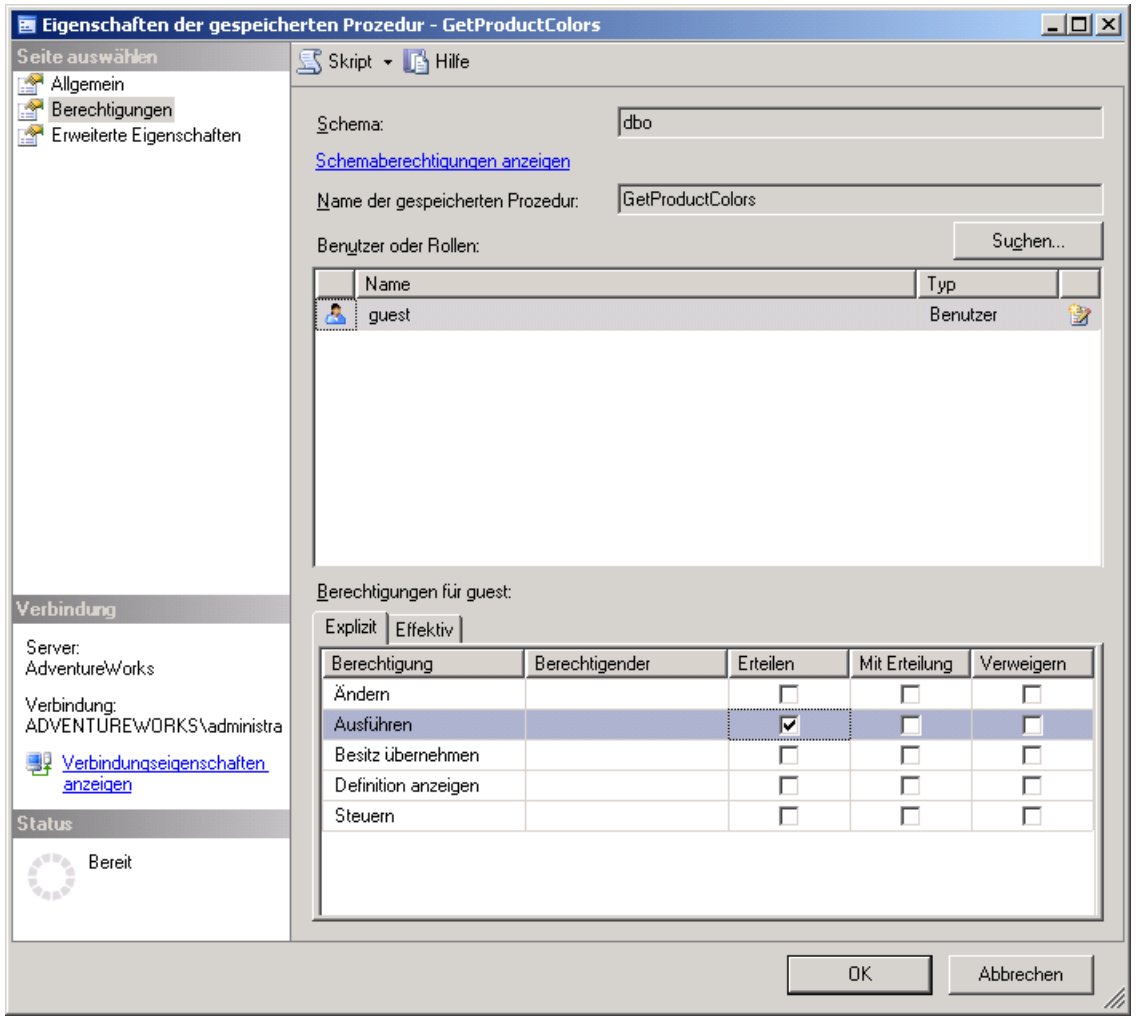
- Berechtigungen, die mit WITH GRANT OPTION erteilt werden, können anderen Prinzipalen durch den Berechtigten erteilt werden
- CASCADE wird auch verwendet, um die Berechtigungen aufzuheben, die von dem Berechtigten erteilt wurden
  - Dies kann auch für DENY gelten

```
GRANT UPDATE ON Marketing.Salesperson TO James
WITH GRANT OPTION;
GO
REVOKE UPDATE ON Marketing.Salesperson FROM James
CASCADE;
GO
```

Sichern von gespeicherten Prozeduren

- Gespeicherte Prozeduren erfordern
  - EXECUTE-Berechtigung, bevor sie aufgerufen werden können
  - ALTER-Berechtigung für Änderungen
  - VIEW DEFINITION für den Dokumentationszugriff

```
GRANT EXECUTE
ON Reports.GetProductPrices
TO Mod11User;
GO
```



Wofür?

- Benutzer benötigen die EXECUTE-Berechtigung, bevor sie skalare benutzerdefinierte Funktionen verwenden können
- Benutzer benötigen die SELECT-Berechtigung für Tabellenwertfunktionen
- Die REFERENCES-Berechtigung wird für CHECK-Einschränkungen, DEFAULT-Werte oder berechnete Spalten verwendet

```
GRANT EXECUTE ON dbo.FormatPhoneNumber TO public;
GO
```

Übersicht über Trennung Benutzer und Schema

- Schemas
  - Konzept geändert in SQL Server 2005
  - Nicht mehr mit Datenbankbenutzern äquivalent
  - Container für Datenbankobjekte
  - Über CREATE SCHEMA erstellt
  - Aufgelistet durch das Abfragen der sys.schemas-Ansicht
- Benutzer können Standardschemas haben
- Integrierte Schemas
  - dbo
  - guest (Gast)
  - sys
  - INFORMATION_SCHEMA

Gewähren von Berechtigungen auf Schemaebene

- Es können einzelne Berechtigungen für Tabellen, Sichten, gespeicherte Prozeduren usw. zugewiesen werden. Es ist aber auch möglich, stattdessen Berechtigungen auf Schemaebene zu erteilt
  - Anwendbar auf alle relevanten Objekte innerhalb des Schemas
  - Einfachere Verwaltung

```
GRANT EXECUTE ON SCHEMA::Marketing
TO Mod11User;
GO
GRANT SELECT ON SCHEMA::DirectMarketing
TO Mod11User;
GO
```