DB2 V8.1 Family Fundamentals certification prep:
Security

Guido Thurmann

Datenbanken
Martin Luther Universität Halle / Wittenberg

21. Juli 2006

Worum geht es?

- Einführung in DB2 Sicherheit
- Grundkenntnis über Datenbanken und Betriebssysteme Voraussetzung
- 2.ter Teil der Vorbereitung und 2ter Teil des Test
- DB2 Installation empfohlen um selbst zu probieren, wird hier nicht behandelt
- Authentifizierung, Autorisierung und Privilegien

Bevor es losgeht, Voraussetzungen

1. Unter Windows einloggen mit einem Nutzer welcher Adminrechte besitzt. In diesem Tutorial Nutzen wir den Nutzer LISAC
2. installiertes DB2
3. Gruppe grp1 auf dem Computer erstellen, auf dem DB2 installiert ist
4. Ebenso einen Nutzer tst1, nicht Mitglied der Admingruppe

Bevor es losgeht, Voraussetzungen

1. Unter Windows einloggen mit einem Nutzer welcher Adminrechte besitzt. In diesem Tutorial Nutzen wir den Nutzer LISAC
2. installiertes DB2
3. Gruppe grp1 auf dem Computer erstellen, auf dem DB2 installiert ist
4. Ebenso einen Nutzer tst1, nicht Mitglied der Admingruppe

Bevor es losgeht, Voraussetzungen

1. Unter Windows einloggen mit einem Nutzer welcher Adminrechte besitzt. In diesem Tutorial Nutzen wir den Nutzer LISAC
2. installiertes DB2
3. Gruppe grp1 auf dem Computer erstellen, auf dem DB2 installiert ist
4. Ebenso einen Nutzer tst1, nicht Mitglied der Admingruppe

Bevor es losgeht, Voraussetzungen

1. Unter Windows einloggen mit einem Nutzer welcher Adminrechte besitzt. In diesem Tutorial Nutzen wir den Nutzer LISAC
2. installiertes DB2
3. Gruppe grp1 auf dem Computer erstellen, auf dem DB2 installiert ist
4. Ebenso einen Nutzer tst1, nicht Mitglied der Admingruppe

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

Aspekte der Datenbank Sicherheit

Was ein Sicherheitsplan definieren muss:

- Wer darf auf die Instanz und/oder die Datenbank zugreifen.
- Wo und wie wird das User Passwort überprüft.
- Welches Autorisationslevel ein User hat.
- Welche Befehle ein Nutzer ausführen darf.
- Welche Daten ein Nutzer sehen/änder darf.
- Welche Datenbank Objekte ein Nutzer manipulieren kann.

DB2 Sicherheitsmechanismen

- **Authentifizierung**, erste Sicherheitsfunktion die man beim Versuch zu einer DB2 Instanz oder Datenbank zu verbinden bemerkt. Hängt sehr mit den Sicherheitsfunktionen des zugrundeliegenden Betriebssystem. DB2 kann aber auch mit anderen Sicherheitsprotokollen (z.B. Kerberos) User authentifizieren.
- **Autorisierung** bezieht sich auf Operationen die ein User/Gruppe (und ggf. auch auf welchen Daten) ausführen kann. Die Berechtigungen eines User hängen von seinem Autorisationslevel ab. Es gibt 5 in DB2: SYSADM, SYSCTRL,SYSMAINT, DBADM und LOAD
- **Privilegien**, sind etwas genauer als Autorisierungen und können sowohl Nutzern als auch Gruppen zugewiesen werden.

DB2 Sicherheitsmechanismen

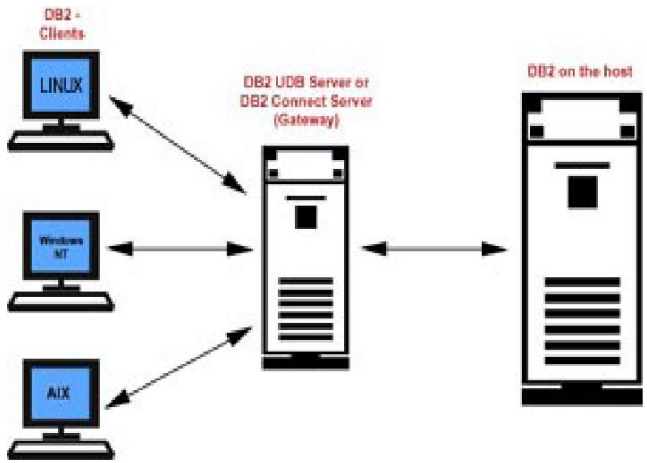
- **Authentifizierung**, erste Sicherheitsfunktion die man beim Versuch zu einer DB2 Instanz oder Datenbank zu verbinden bemerkt. Hängt sehr mit den Sicherheitsfunktionen des zugrundeliegenden Betriebssystem. DB2 kann aber auch mit anderen Sicherheitsprotokollen (z.B. Kerberos) User authentifizieren.
- **Autorisierung** bezieht sich auf Operationen die ein User/Gruppe (und ggf. auch auf welchen Daten) ausführen kann. Die Berechtigungen eines User hängen von seinem Autorisationslevel ab. Es gibt 5 in DB2: SYSADM, SYSCTRL,SYSMAINT, DBADM und LOAD
- **Privilegien**, sind etwas genauer als Autorisierungen und können sowohl Nutzern als auch Gruppen zugewiesen werden.

DB2 Sicherheitsmechanismen

- **Authentifizierung**, erste Sicherheitsfunktion die man beim Versuch zu einer DB2 Instanz oder Datenbank zu verbinden bemerkt. Hängt sehr mit den Sicherheitsfunktionen des zugrundeliegenden Betriebssystem. DB2 kann aber auch mit anderen Sicherheitsprotokollen (z.B. Kerberos) User authentifizieren.
- **Autorisierung** bezieht sich auf Operationen die ein User/Gruppe (und ggf. auch auf welchen Daten) ausführen kann. Die Berechtigungen eines User hängen von seinem Autorisationslevel ab. Es gibt 5 in DB2: SYSADM, SYSCTRL,SYSMAINT, DBADM und LOAD
- **Privilegien**, sind etwas genauer als Autorisierungen und können sowohl Nutzern als auch Gruppen zugewiesen werden.

Clients Server, Gateways, Hosts

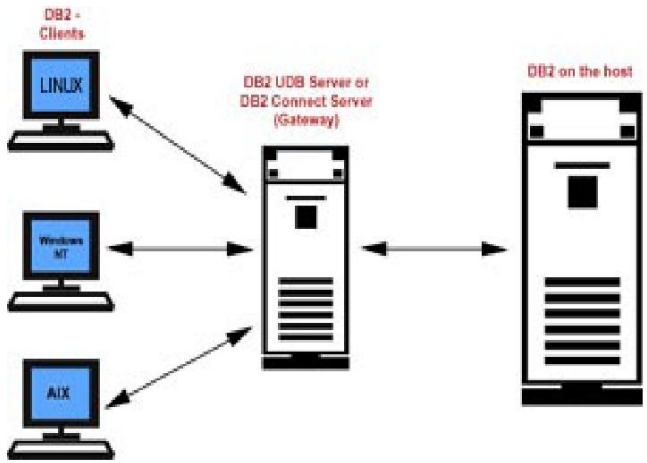
Jeder Punkt an dem Daten abgegriffen werden können, muss gesichert werden.



- Datenbank Server: ist/sind die Computer auf denen die Datenbank physikalisch läuft.
- DB2 Clients stellen Anfragen an den DB2 Server, sie können lokal als auch entfernt sein.
- Wenn sich die Datenbank in einem Großrechner mit Betriebssystemen wie z.B. AS/400 oder OS/390 befindet, wird dieser Host oder Host Server genannt.
- Ein Gateway ist ist ein Computer auf dem eine Version von “DB2 connect“ läuft. Clients können sich via Gateway zu DB2 Datenbank verbinden, die sich auf einem Host befinden.

Clients Server, Gateways, Hosts

Jeder Punkt an dem Daten abgegriffen werden können, muss gesichert werden.



- Datenbank Server: ist/sind die Computer auf denen die Datenbank physikalisch läuft.
- DB2 Clients stellen Anfragen an den DB2 Server, sie können lokal als auch entfernt sein.
- Wenn sich die Datenbank in einem Großrechner mit Betriebssystemen wie z.B. AS/400 oder OS/390 befindet, wird dieser Host oder Host Server genannt.
- Ein Gateway ist ist ein Computer auf dem eine Version von “DB2 connect“ läuft. Clients können sich via Gateway zu DB2 Datenbank verbinden, die sich auf einem Host befinden.

Wann authentifiziert DB2?

Authentifizierung kontrolliert folgende Aspekte des Datenbanken Sicherheitsplans:

1. Wer ist berechtigt die Instanz und oder Datenbank zu betreten
2. Wo und wie wird das User Passwort verifiziert

Dies geschieht mit Hilfe der Sicherheitsmechanismen des Betriebssystems, wann immer ein attach oder eine connect Kommando ausgeführt wird.

- attach: zur Herstellung einer Verbindung zu einer DB2 Instanz
- connect: zur Herstellung einer Verbindung zu einer Datenbank in einer DB2 Instanz

Wann authentifiziert DB2?

Authentifizierung kontrolliert folgende Aspekte des Datenbanken Sicherheitsplans:

1. Wer ist berechtigt die Instanz und oder Datenbank zu betreten
2. Wo und wie wird das User Passwort verifiziert

Dies geschieht mit Hilfe der Sicherheitsmechanismen des Betriebssystems, wann immer ein attach oder eine connect Kommando ausgeführt wird.

- attach: zur Herstellung einer Verbindung zu einer DB2 Instanz
- connect: zur Herstellung einer Verbindung zu einer Datenbank in einer DB2 Instanz

Wann authentifiziert DB2?

Authentifizierung kontrolliert folgende Aspekte des Datenbanken Sicherheitsplans:

1. Wer ist berechtigt die Instanz und oder Datenbank zu betreten
2. Wo und wie wird das User Passwort verifiziert

Dies geschieht mit Hilfe der Sicherheitsmechanismen des Betriebssystems, wann immer ein attach oder eine connect Kommando ausgeführt wird.

- attach: zur Herstellung einer Verbindung zu einer DB2 Instanz
- connect: zur Herstellung einer Verbindung zu einer Datenbank in einer DB2 Instanz

Wann authentifiziert DB2?

Authentifizierung kontrolliert folgende Aspekte des Datenbanken Sicherheitsplans:

1. Wer ist berechtigt die Instanz und oder Datenbank zu betreten
2. Wo und wie wird das User Passwort verifiziert

Dies geschieht mit Hilfe der Sicherheitsmechanismen des Betriebssystems, wann immer ein attach oder eine connect Kommando ausgeführt wird.

- attach: zur Herstellung einer Verbindung zu einer DB2 Instanz
- connect: zur Herstellung einer Verbindung zu einer Datenbank in einer DB2 Instanz

Wann authentifiziert DB2?

Authentifizierung kontrolliert folgende Aspekte des Datenbanken Sicherheitsplans:

1. Wer ist berechtigt die Instanz und oder Datenbank zu betreten
2. Wo und wie wird das User Passwort verifiziert

Dies geschieht mit Hilfe der Sicherheitsmechanismen des Betriebssystems, wann immer ein attach oder eine connect Kommando ausgeführt wird.

- attach: zur Herstellung einer Verbindung zu einer DB2 Instanz
- connect: zur Herstellung einer Verbindung zu einer Datenbank in einer DB2 Instanz

Wann authentifiziert DB2? Beispiele

In den folgenden Beispielen nutzen wir den Standard-Authentifizierungs-Typ: `SERVER` (in der Datenbank Konfigurationsdatei). Wir nehmen an wir sind mit der User ID auf dem Computer angemeldet, mit der wir die Instanz `db2inst1` erstellt haben.

1. `db2 attach to db2inst1`
   Die Authentifizierung geschieht hier implizit. Es wird die User ID des Users benutzt der sich auf dem Computer angemeldet hat und schon vom Betriebssystem verifiziert wurde.
2. `db2 connect to sample user tst1 using mypass`
   ```
   Database Connection Information
   Database server = DB2/NT 8.1.0
   SQL authorization ID = TST1
   Local database alias = SAMPLE
   ```
   Hier geschieht die Authentifizierung explizit. User `tst1` mit Passwort `mypass` wird durch das Betriebssystem verifiziert. User `tst1` wird erfolgreich zur Beispiel Datenbank verbunden.
3. `db2 connect to sample user tst1 using mypass new chgpass confirm chgpass`
   User `tst1` mit Passwort `mypass` wird durch das Betriebssystem verifiziert. Sein Passwort wird auf `chgpass` geändert.
   Nun würde das 2. Beispiel fehlschlagen.

DB2 Authentifizierungstypen

Durch die Typen wird der Ort der Authentifizierung festgelegt.

Type	Description
SERVER	Authentication takes place on the server.
SERVER_ENCRYPT	Authentication takes place on the server. Passwords are encrypted at the client machine before being sent to the server.
CLIENT	Authentication takes place on the client machine (see Dealing with untrusted clients on page for exceptions).
∗KERBEROS	Authentication is performed by the Kerberos security software.
∗KRB_SERVER_ENCRYPT	Authentication is performed by Kerberos security software if the client setting is KERBEROS. Otherwise, SERVER_ENCRYPT is used.

*These settings are valid only for Windows 2000 operating systems.

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    ```
    db2 get dbm cfg
    ```
  - zur Veränderung auf `server_encrypt`:
    ```
    C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
    server_encrypt
    C:\PROGRA1\SQLLIB\BIN> db2stop
    C:\PROGRA1\SQLLIB\BIN> db2start
    ```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    ```
    db2 get dbm cfg
    ```
  - zur Veränderung auf `server_encrypt`:
    ```
    C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
    server_encrypt
    C:\PROGRA1\SQLLIB\BIN> db2stop
    C:\PROGRA1\SQLLIB\BIN> db2start
    ```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    ```
    db2 get dbm cfg
    ```
  - zur Veränderung auf `server_encrypt`:
    ```
    C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
    server_encrypt
    C:\PROGRA1\SQLLIB\BIN> db2stop
    C:\PROGRA1\SQLLIB\BIN> db2start
    ```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    ```
    db2 get dbm cfg
    ```
  - zur Veränderung auf `server_encrypt`:
    ```
    C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
    server_encrypt
    C:\PROGRA1\SQLLIB\BIN> db2stop
    C:\PROGRA1\SQLLIB\BIN> db2start
    ```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    ```
    db2 get dbm cfg
    ```
  - zur Veränderung auf `server_encrypt`:
    ```
    C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
    server_encrypt
    C:\PROGRA1\SQLLIB\BIN> db2stop
    C:\PROGRA1\SQLLIB\BIN> db2start
    ```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    `db2 get dbm cfg`
  - zur Veränderung auf `server_encrypt`:

```
C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
server_encrypt
C:\PROGRA1\SQLLIB\BIN> db2stop
C:\PROGRA1\SQLLIB\BIN> db2start
```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    ```
    db2 get dbm cfg
    ```
  - zur Veränderung auf `server_encrypt`:
    ```
    C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
    server_encrypt
    C:\PROGRA1\SQLLIB\BIN> db2stop
    C:\PROGRA1\SQLLIB\BIN> db2start
    ```

Einstellungen auf dem Server

- werden in der Datenbank Manager Konfigurationsdatei (`DBM CFG`) unter Nutzung des `AUTHENTICATION` Parameters vorgenommen
- weil die `DBM CFG` Datei eine Instanz-Level Konfigurationsdatei ist, gelten die Einstellungen `AUTHENTICATION` Parameters in der gesamten Instanz
- der Parameter kann wie folgt bearbeitet werden:
  - um die derzeitige Einstellung des `AUTHENTICATION` Parameter anzuzeigen:
    `db2 get dbm cfg`
  - zur Veränderung auf `server_encrypt`:

```
C:\PROGRA1\SQLLIB\BIN> db2update dbm cfg using authentication
server_encrypt
C:\PROGRA1\SQLLIB\BIN> db2stop
C:\PROGRA1\SQLLIB\BIN> db2start
```

Einstellungen auf dem Gateway

- wird mit Hilfe des catalog database Kommandos gesetzt
- hier nutzen wir die host Datenbank mit Namen `myhostdb`
- zum Ändern des Typs auf `SERVER`:
  ```
  db2 catalog database myhostdb at node nd1 authentication
  dcs db2 terminate
  ```
- **Die Authentifizierung findet niemals auf dem Gateway selbst statt.**
- In DB2 Version 8 muss diese immer entweder auf dem Client oder Server geschehen.

Einstellungen auf dem Gateway

- wird mit Hilfe des catalog database Kommandos gesetzt
- hier nutzen wir die host Datenbank mit Namen `myhostdb`
- zum Ändern des Typs auf `SERVER`:
  ```
  db2 catalog database myhostdb at node nd1 authentication
  dcs db2 terminate
  ```
- **Die Authentifizierung findet niemals auf dem Gateway selbst statt.**
- In DB2 Version 8 muss diese immer entweder auf dem Client oder Server geschehen.

Einstellungen auf dem Gateway

- wird mit Hilfe des catalog database Kommandos gesetzt
- hier nutzen wir die host Datenbank mit Namen `myhostdb`
- zum Ändern des Typs auf `SERVER`:

```
db2 catalog database myhostdb at node nd1 authentication
dcs db2 terminate
```

- **Die Authentifizierung findet niemals auf dem Gateway selbst statt.**
- In DB2 Version 8 muss diese immer entweder auf dem Client oder Server geschehen.

Einstellungen auf dem Gateway

- wird mit Hilfe des catalog database Kommandos gesetzt
- hier nutzen wir die host Datenbank mit Namen `myhostdb`
- zum Ändern des Typs auf `SERVER`:
  ```
  db2 catalog database myhostdb at node nd1 authentication
  dcs db2 terminate
  ```
- **Die Authentifizierung findet niemals auf dem Gateway selbst statt.**
- In DB2 Version 8 muss diese immer entweder auf dem Client oder Server geschehen.

Einstellungen auf dem Gateway

- wird mit Hilfe des catalog database Kommandos gesetzt
- hier nutzen wir die host Datenbank mit Namen `myhostdb`
- zum Ändern des Typs auf `SERVER`:
  ```
  db2 catalog database myhostdb at node nd1 authentication
  dcs db2 terminate
  ```
- **Die Authentifizierung findet niemals auf dem Gateway selbst statt.**
- In DB2 Version 8 muss diese immer entweder auf dem Client oder Server geschehen.

Einstellungen auf dem Gateway

- wird mit Hilfe des catalog database Kommandos gesetzt
- hier nutzen wir die host Datenbank mit Namen `myhostdb`
- zum Ändern des Typs auf `SERVER`:
  ```
  db2 catalog database myhostdb at node nd1 authentication
  dcs db2 terminate
  ```
- **Die Authentifizierung findet niemals auf dem Gateway selbst statt.**
- In DB2 Version 8 muss diese immer entweder auf dem Client oder Server geschehen.

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Einstellungen auf dem Client, 2 Szenerien:

1. einen Client der sich zu einer DB2 Datenbank auf einem Server verbindet
   - Der A.-Typ von Client und Server muss passen (außer bei `KRB_SERVER_ENCRYPT`)
   - angenommen der A.-Typ des Servers ist `SERVER`, dann nutzt folgendes Kommando ebenfalls diesen A.-Typ
     ```
     db2 catalog database sample at node nd1 authentication server
     ```
   - wenn der Typ nicht spezifiziert wird, wird standardmäßig `SERVER_ENCRYPT` genutzt.
2. einen Client der sich zu einer DB2 Datenbank auf einem Host verbindet (z.B.: DB2 auf OS(390)
   - Angenommen der A.-Typ auf dem Gateway ist auf `SERVER` gesetzt.
   - wenn kein A.-Typ gesetzt ist wird standardmäßig `SERVER` angenommen, wenn die Verbindung via DB2 connect geschieht
   - folgendes Kommando auf dem Client bewirkt das ein verschlüsseltes Passwort zum Gateway geschickt wird:
     ```
     db2 catalog database myhostdb at node nd1 authentication server_encrypt
     ```
   - wenn der A.-Typ auf dem Gateway auf `SERVER_ENCRYPT` gesetzt ist, wird die Authentifizierung erneut auf der Host Datenbank stattfinden, das Passwort ist nun aber zwischen Gateway und Host verschlüsselt

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Behandlung von nicht vertrauenswürdigen Clients

- wenn auf dem Server oder Gateway der A.-Typ auf `CLIENT` gesetzt ist, wird erwartet das der Client UserID und Passwort verifiziert
- einige DB2 Clients laufen auf Betriebssysteme die dies nicht können, z.b.: Windows98 und Windows ME
- 2 weitere Parameter in der DBM CFG Datei legen fest was geschehen soll, wenn Server bzw. Gateway A.-Typ `CLIENT` ist und ein nicht vertrauenswürdiger Client versucht sich mit der DB2 Instanz zu verbinden:
  - `TRUST_ALLCLNTS`
  - `TRUST_CLNTAUTH`
- zusätzlich gibt es noch 2 weitere Faktoren die eine Rolle spielen:
  - werden Passwort oder UserID zur Verfügung gestellt
  - Typ der Verbindung:
    - **untrusted** Clients, siehe oben
    - **Host** Clients, Clients die auf Host Operation System wie OS/390
    - **Trusted** Clients, auf nicht-host BS aber mit native Sicherheitseigenschaften

Autorisationstyp `CLIENT`

User ID/ Password ?	TRUST_ALLCLNTS	TRUST_CLNTAUTH	Untrusted Client	Trusted Client	Host Client
No	Yes	CLIENT	CLIENT	CLIENT	CLIENT
No	Yes	SERVER	CLIENT	CLIENT	CLIENT
No	No	CLIENT	SERVER	CLIENT	CLIENT
No	No	SERVER	SERVER	CLIENT	CLIENT
No	DRDAONLY	CLIENT	SERVER	SERVER	CLIENT
No	DRDAONLY	SERVER	SERVER	SERVER	CLIENT
Yes	Yes	CLIENT	CLIENT	CLIENT	CLIENT
Yes	Yes	SERVER	SERVER	SERVER	SERVER
Yes	No	CLIENT	SERVER	CLIENT	CLIENT
Yes	No	SERVER	SERVER	SERVER	SERVER
Yes	DRDAONLY	CLIENT	SERVER	SERVER	CLIENT
Yes	DRDAONLY	SERVER	SERVER	SERVER	SERVER

DRDAONLY bezieht sich nur auch Host Clients, trotz der Tatsache, dass DB2 Version 8 CLients beim Verbinden ebenfalls DRDA benutzen.

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Autorisationstyp `CLIENT` ffd.

Einstellungen auf dem Server:

```
db2 update dbm cfg using authentication client
db2 update dbm cfg using trust_allclnts yes
db2 update dbm cfg using trust_clntauth server
db2stop
db2start
```

Einstellung auf dem Client:

```
db2 catalog database sample at node nd1 authentication client
```

Wirkung:

- `db2 connect to sample` bewirkt auf Grund der Einstellungen die Authentifizierung auf dem Client
- `db2 connect to sample user tst1 using mypass` von einem beliebigen Client, wird weiterhin eine Authentifizierung auf dem Server bedingen

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

Einführung in Autorisationen

- DB2 Autorisationen kontrollieren folgende Teile des Datenbank Sicherheitsplans:
  - das Autorisationslevel welches ein Nutzer erhält
  - die Kommandos die ein Nutzer absetzen darf
  - die Daten die ein User sehen bzw. manipulieren darf
  - die Datenbankobjekte die ein Nutzer erstellen, ändern oder löschen darf
- Autorisationen sind aus Gruppen von Privilegien, High-Level-Datenbank-Management (Instanz-Level) Verwaltung und Utility Operationen zusammengesetzt
- 3 von 5 Autorisationen sind auf Instanz-Level: `SYSADM`, `SYSCTRL`, `SYSMAINT`, sie beinhalten Instanz-Level Kommandos genauso wie Kommandos an alle Datenbanken innerhalb der Instanz. Sie können nur zu Gruppen zugeordnet werden.
- `DBADM` und `LOAD` Autoritäten können zu User und Gruppen für bestimmte Datenbanken mit Hilfe des `GRANT` Befehls zugewiesen werden.
- Anzeige der eigenen Autorisationen mittels: `db2 get authorizations`

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:
  `db2 update dbm cfg using SYSADM_GROUP grp1`
  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:
  `db2 update dbm cfg using SYSADM_GROUP grp1`
  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:

  `db2 update dbm cfg using SYSADM_GROUP grp1`

  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:
  `db2 update dbm cfg using SYSADM_GROUP grp1`
  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:

  `db2 update dbm cfg using SYSADM_GROUP grp1`

  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:

  `db2 update dbm cfg using SYSADM_GROUP grp1`

  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:

`db2 update dbm cfg using SYSADM_GROUP grp1`

die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:
  `db2 update dbm cfg using SYSADM_GROUP grp1`
  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSADM

- vergleichbar mit der root Autorität bei Unix
- jedes Kommando kann an die Instanz, an jede Datenbank innerhalb der Instanz und an jedes Objekt innerhalb der Datenbank
- können alle Daten manipulieren sowie Autoritäten und Privilegien entziehen und vergeben
- dürfen als einzige die DBM CFG Datei ändern
- SYSADM wird in der DBM CFG mit dem SYSADM_GROUP Parameter verwaltet.
- bei erstellen einer Instanz wird der Parameter
  - unter Windows auf Administrator,
  - unter UNIX auf die Hauptgruppe des Nutzers der sie erstellt hat,

  gesetzt
- SYSADM können als einzige die SYS* Autoritäten vergeben:
  `db2 update dbm cfg using SYSADM_GROUP grp1`
  die Instanz muss noch neu gestartet werden, auch kann man wenn man nicht als Mitglied der Gruppe `grp1` eingeloggt war, nun die Maschine nicht neu starten,man muss sich erst mit einer ID die Mitglied der Gruppe `grp1` einloggen

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:

```
db2 update dbm cfg using SYSCTRL_GROUP group name
```

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:

```
db2 update dbm cfg using SYSCTRL_GROUP group name
```

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:

```
db2 update dbm cfg using SYSCTRL_GROUP group name
```

SYSCTRL

- ermöglicht alle administrativen und verwaltenden Kommandos innerhalb der Instanz
- sie können aber nicht alle Daten bearbeiten/einsehen, dazu müssen sie zusätzliche Privilegien besitzen
- Beispiele:
  - `db2start/db2stop`
  - `db2 create/drop database`
  - `db2 create/drop tablespac`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- Nutzer mit SYSADM Recht können SYSCRTL zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSCTRL_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:

```
db2 update dbm cfg using SYSMAINT_GROUP group name
```

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

SYSMAINT

- Teilmenge von SYSCTRL
- nur verwaltende Kommandos, wie
  - `db2start/db2stop`
  - `db2 backup/restore/rollforward database`
  - `db2 runstats` (against any table)
  - `db2 update db cfg for database dbname`
- können **nicht** Datenbanken oder Tabellenplatz erstellen oder löschen
- können nur Daten einsehen wenn sie die Privilegien zugewiesen bekommen haben
- Nutzer mit SYSADM Recht können SYSMAINT zu Gruppen mit folgendem Kommando zuweisen:
  `db2 update dbm cfg using SYSMAINT_GROUP group name`

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

DBADM

- DBADM ist eher Datenbank-Level Autorität als Instanz-Level
- DBADM User haben vollständige Kontrolle über die Datenbank (fast), folgendes können sie z.B. **nicht**:
  - `drop database`
  - `drop/create tablespace`
  - `backup/restore database`
  - `update db cfg for database db name`
- erlaubt sind:
  - `db2 create/drop table`
  - `db2 grant/revoke` (any privilege)
  - `db2 runstats` (any table)
- DBADM erhalten automatisch alle Privilegien über die Datenbankobjekte und ihren Inhalt
- kann an User und Gruppen vergeben werden, Beispiele der Vergabe:
  - `db2 create database test` (User der diesen Befehl ausführt erhält automatisch DBADM an Datenbank test)
  - `db2 connect tosample`
    `db2 grant dbadm on database to user tst1`
    nur durch SYSADM ausführbar, weist DBADM für `sample` an User `tst1` zu
  - `db2 grant dbadm on database to group grp1`
    nur durch SYSADM ausführbar, wie eben nur an Gruppe

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

LOAD

- wird auch als Datenbank-Level Autorität betrachtet
- kann an User und Gruppen zugewiesen werden
- erlaubt den `LOAD` Befehl, der eine schnelle Alternative zu `insert` oder `import` ist, wenn man große Daten in eine Tabelle laden möchte
- je nach dem Typ von `LOAD` der ausgeführt werden soll, reicht die `LOAD` Autorität evtl. nicht, spezielle Privilegien werden evtl. benötigt
- folgende Befehle können ausgeführt werden:
  - `db2 quiesce tablespaces for table`
  - `db2 list tablespaces`
  - `db2 runstats` (any table)
  - `db2 load insert` (must have insert privilege on table)
  - `db2 load restart/terminate after load insert` (must have insert privilege on table)
  - `db2 load replace` (must have insert and delete privilege on table)
  - `db2 load restart/terminate after load replace` (must have insert and delete privilege on table
- Nur SYSADM und DBADM können DBADM vergeben oder entziehen:
  - `db2 grant load on database to user tst1`
    `db2 grant insert on table sales to user tst1`
  - `db2 grant load on database to group grp1`
    `db2 grant delete on table sales to group grp1`
    `db2 grant insert on table sales to group grp1`

Datenbank- und Objektprivilegien

wir unterscheiden

- Datenbanken Level Privilegien
  - beinhaltet alle Objekte in einer Datenbank
- Objekt Level Privilegien
  - Datenbankobjekte beinhalten Tabellen, Views, Indexes, Schemas und Packages
  - Informationen über Objelkt-Level Privilegien werden in System Catalog View gespeichert, die View Namen sind: `syscat.tabauth`, `syscat.colauth`, `syscat.indexauth`, `syscat.schemaauth`, `syscat.routineauth`, und `syscat.packageauth`.

Datenbank- und Objektprivilegien

wir unterscheiden

- Datenbanken Level Privilegien
  - beinhaltet alle Objekte in einer Datenbank
- Objekt Level Privilegien
  - Datenbankobjekte beinhalten Tabellen, Views, Indexes, Schemas und Packages
  - Informationen über Objelkt-Level Privilegien werden in System Catalog View gespeichert, die View Namen sind: `syscat.tabauth`, `syscat.colauth`, `syscat.indexauth`, `syscat.schemaauth`, `syscat.routineauth`, und `syscat.packageauth`.

Datenbank- und Objektprivilegien

wir unterscheiden

- Datenbanken Level Privilegien
  - beinhaltet alle Objekte in einer Datenbank
- Objekt Level Privilegien
  - Datenbankobjekte beinhalten Tabellen, Views, Indexes, Schemas und Packages
  - Informationen über Objelkt-Level Privilegien werden in System Catalog View gespeichert, die View Namen sind: `syscat.tabauth`, `syscat.colauth`, `syscat.indexauth`, `syscat.schemaauth`, `syscat.routineauth`, und `syscat.packageauth`.

Datenbank- und Objektprivilegien

wir unterscheiden

- Datenbanken Level Privilegien
  - beinhaltet alle Objekte in einer Datenbank
- Objekt Level Privilegien
  - Datenbankobjekte beinhalten Tabellen, Views, Indexes, Schemas und Packages
  - Informationen über Objelkt-Level Privilegien werden in System Catalog View gespeichert, die View Namen sind: `syscat.tabauth`, `syscat.colauth`, `syscat.indexauth`, `syscat.schemaauth`, `syscat.routineauth`, und `syscat.packageauth`.

Datenbank- und Objektprivilegien

wir unterscheiden

- Datenbanken Level Privilegien
  - beinhaltet alle Objekte in einer Datenbank
- Objekt Level Privilegien
  - Datenbankobjekte beinhalten Tabellen, Views, Indexes, Schemas und Packages
  - Informationen über Objelkt-Level Privilegien werden in System Catalog View gespeichert, die View Namen sind: `syscat.tabauth`, `syscat.colauth`, `syscat.indexauth`, `syscat.schemaauth`, `syscat.routineauth`, und `syscat.packageauth`.

Datenbank- und Objektprivilegien

wir unterscheiden

- Datenbanken Level Privilegien
  - beinhaltet alle Objekte in einer Datenbank
- Objekt Level Privilegien
  - Datenbankobjekte beinhalten Tabellen, Views, Indexes, Schemas und Packages
  - Informationen über Objelkt-Level Privilegien werden in System Catalog View gespeichert, die View Namen sind: `syscat.tabauth`, `syscat.colauth`, `syscat.indexauth`, `syscat.schemaauth`, `syscat.routineauth`, und `syscat.packageauth`.

Datenbankprivilegien

- CREATETAB: Users can create tables within the database.
- BINDADD: Users can create packages in the database using the BIND command.
- CONNECT: Users can connect to the database.
- CREATE_NOT_FENCED: Users can create unfenced user-defined functions (UDFs).
- IMPLICIT_SCHEMA: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- LOAD: Users can load data into a table
- QUIESCE_CONNECT: Users can access a database while it is in a quiesced state.
- CREATE_EXTERNAL_ROUTINE: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- `CREATETAB`: Users can create tables within the database.
- `BINDADD`: Users can create packages in the database using the BIND command.
- `CONNECT`: Users can connect to the database.
- `CREATE_NOT_FENCED`: Users can create unfenced user-defined functions (UDFs).
- `IMPLICIT_SCHEMA`: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- `LOAD`: Users can load data into a table
- `QUIESCE_CONNECT`: Users can access a database while it is in a quiesced state.
- `CREATE_EXTERNAL_ROUTINE`: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- CREATETAB: Users can create tables within the database.
- BINDADD: Users can create packages in the database using the BIND command.
- CONNECT: Users can connect to the database.
- CREATE_NOT_FENCED: Users can create unfenced user-defined functions (UDFs).
- IMPLICIT_SCHEMA: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- LOAD: Users can load data into a table
- QUIESCE_CONNECT: Users can access a database while it is in a quiesced state.
- CREATE_EXTERNAL_ROUTINE: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- `CREATETAB`: Users can create tables within the database.
- `BINDADD`: Users can create packages in the database using the BIND command.
- `CONNECT`: Users can connect to the database.
- `CREATE_NOT_FENCED`: Users can create unfenced user-defined functions (UDFs).
- `IMPLICIT_SCHEMA`: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- `LOAD`: Users can load data into a table
- `QUIESCE_CONNECT`: Users can access a database while it is in a quiesced state.
- `CREATE_EXTERNAL_ROUTINE`: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- `CREATETAB`: Users can create tables within the database.
- `BINDADD`: Users can create packages in the database using the BIND command.
- `CONNECT`: Users can connect to the database.
- `CREATE_NOT_FENCED`: Users can create unfenced user-defined functions (UDFs).
- `IMPLICIT_SCHEMA`: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- `LOAD`: Users can load data into a table
- `QUIESCE_CONNECT`: Users can access a database while it is in a quiesced state.
- `CREATE_EXTERNAL_ROUTINE`: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- `CREATETAB`: Users can create tables within the database.
- `BINDADD`: Users can create packages in the database using the BIND command.
- `CONNECT`: Users can connect to the database.
- `CREATE_NOT_FENCED`: Users can create unfenced user-defined functions (UDFs).
- `IMPLICIT_SCHEMA`: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- `LOAD`: Users can load data into a table
- `QUIESCE_CONNECT`: Users can access a database while it is in a quiesced state.
- `CREATE_EXTERNAL_ROUTINE`: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- `CREATETAB`: Users can create tables within the database.
- `BINDADD`: Users can create packages in the database using the BIND command.
- `CONNECT`: Users can connect to the database.
- `CREATE_NOT_FENCED`: Users can create unfenced user-defined functions (UDFs).
- `IMPLICIT_SCHEMA`: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- `LOAD`: Users can load data into a table
- `QUIESCE_CONNECT`: Users can access a database while it is in a quiesced state.
- `CREATE_EXTERNAL_ROUTINE`: Users can create a procedure for use by applications and other users of the database.

Datenbankprivilegien

- `CREATETAB`: Users can create tables within the database.
- `BINDADD`: Users can create packages in the database using the BIND command.
- `CONNECT`: Users can connect to the database.
- `CREATE_NOT_FENCED`: Users can create unfenced user-defined functions (UDFs).
- `IMPLICIT_SCHEMA`: Users can implicitly create schemas within the database without using the CREATE SCHEMA command.
- `LOAD`: Users can load data into a table
- `QUIESCE_CONNECT`: Users can access a database while it is in a quiesced state.
- `CREATE_EXTERNAL_ROUTINE`: Users can create a procedure for use by applications and other users of the database.

Objektprivilegien

Privilege name	Relevant object(s)	Description
CONTROL	Table, View, Index, Package, Alias, Distinct Type, User Defined function, Sequence	Provides full authority on the object. Users with this privilege can also grant or revoke privileges on the object to other users.
DELETE	Table, View	Allows users to delete records from the object.
INSERT	Table, View	Allows users to insert records into the object via the INSERT or the IMPORT commands.
SELECT	Table, View	Provides the ability to view the contents of the object using the select statement.
UPDATE	Table, View	Allows users to modify records within the object using the update statement.
ALTER	Table	Allows users to alter the object definition using the alter statement.
INDEX	Table	Allows users to create indexes on the object using the create index statement.
REFERENCES	Table	Provides the ability to create or drop foreign key constraints on the object.
BIND	Package	Allows users to rebind existing packages.
EXECUTE	Package, Procedure, Function, Method	Allows users to execute packages and routines.
ALTERIN	Schema	Allows users to modify definitions of objects within the schema.
CREATEIN	Schema	Allows users to create objects within the schema.
DROPIN	Schema	Allows users to drop objects within the schema.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man: `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten: `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man: `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten: `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man: `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten: `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man: `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten: `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man: `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten: `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man: `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten: `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man:
  `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten:
  `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien

Privilegien können Nutzer explizit mit den Kommandos `GRANT` und `REVOKE` gegeben und entzogen werden.
Beispiel:

- unter Windows als Admin angemeldet, öffnen von 2 Fenstern mit Instanz `db2inst1`
- im 1. Fenster: `db2 connect to sample`, ein Nutzer wird (als Admin) angemeldet
- im 2. Fenster: `db2 connect to sample user tst1 using passwd`, Nutzer tst1 wird angemeldet, mit keinen Privilegien an der Sample DB.
- im 2. Fenster: `db2 select * from lisac.org`, da tst1 keine Rechte hat erhält man:
  `SQL0551N TST1"does not have the privilege to perform operation SSELECTön object LISAC.ORG".`
- im 1. Fenster: `db2 grant select on table lisac.org to user tst1`, gibt User `tst1` das Privileg select, nun würde obige Anfrage Erfolg haben
- im 2. Fenster: `db2 insert into lisac.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')`, wir erhalten:
  `SQL0551N TST1"does not have the privilege to perform operation ÏNSERTön object LISAC.ORG"`
- im 1. Fenster: `db2 grant insert on table lisac.org to group grp1, user mytest`, da tst1 zur Gruppe `grp1` gehört wird nun der obige Insert-Befehl Erfolg haben.

Explizite Privilegien ffd.

- im 2. Fenster: `db2 drop table lisac.emp_photo` führt zu:
  `SQL0551N "TST1" does not have the privilege to perform operation "DROP TABLE" on object "LISAC.EMP_PHOTO".`
- im 1. Fenster: `db2 grant dropin on schema lisac to all`, korrigiert dies (da **alle** Nutzer nun im Schema `LISAC` löschen dürfen)
- im 1. Fenster:
  ```
  db2 revoke select on table lisac.org from user tst1
  db2 revoke insert on table lisac.org from group grp1
  db2 revoke dropin on schema lisac from all
  ```
- man beachte das wenn man einer Gruppe Rechte entzieht sich dies nur auf die auswirkt die nicht direkt an einen einzelner Nutzer vergeben wurden.

Explizite Privilegien ffd.

- im 2. Fenster: `db2 drop table lisac.emp_photo` führt zu:
  `SQL0551N "TST1" does not have the privilege to perform operation "DROP TABLE" on object "LISAC.EMP_PHOTO".`
- im 1. Fenster: `db2 grant dropin on schema lisac to all`, korrigiert dies (da **alle** Nutzer nun im Schema `LISAC` löschen dürfen)
- im 1. Fenster:
  ```
  db2 revoke select on table lisac.org from user tst1
  db2 revoke insert on table lisac.org from group grp1
  db2 revoke dropin on schema lisac from all
  ```
- man beachte das wenn man einer Gruppe Rechte entzieht sich dies nur auf die auswirkt die nicht direkt an einen einzelner Nutzer vergeben wurden.

Explizite Privilegien ffd.

- im 2. Fenster: `db2 drop table lisac.emp_photo` führt zu:
  ```
  SQL0551N "TST1" does not have the privilege to perform operation "DROP TABLE"
  on object "LISAC.EMP_PHOTO".
  ```
- im 1. Fenster: `db2 grant dropin on schema lisac to all`, korrigiert dies (da **alle** Nutzer nun im Schema `LISAC` löschen dürfen)
- im 1. Fenster:
  ```
  db2 revoke select on table lisac.org from user tst1
  db2 revoke insert on table lisac.org from group grp1
  db2 revoke dropin on schema lisac from all
  ```
- man beachte das wenn man einer Gruppe Rechte entzieht sich dies nur auf die auswirkt die nicht direkt an einen einzelner Nutzer vergeben wurden.

Explizite Privilegien ffd.

- im 2. Fenster: `db2 drop table lisac.emp_photo` führt zu:
  `SQL0551N "TST1" does not have the privilege to perform operation "DROP TABLE" on object "LISAC.EMP_PHOTO".`
- im 1. Fenster: `db2 grant dropin on schema lisac to all`, korrigiert dies (da **alle** Nutzer nun im Schema `LISAC` löschen dürfen)
- im 1. Fenster:
  ```
  db2 revoke select on table lisac.org from user tst1
  db2 revoke insert on table lisac.org from group grp1
  db2 revoke dropin on schema lisac from all
  ```
- man beachte das wenn man einer Gruppe Rechte entzieht sich dies nur auf die auswirkt die nicht direkt an einen einzelner Nutzer vergeben wurden.

Implizite Privilegien

DB2 vergibt Rechte u.U. automatisch, wenn bestimmte Kommandos benutzt werden, ohne das ein extra `GRANT` notwendig ist. Diese Rechte werden auch wieder automatisch entzogen wenn die zugehörigen Objekte gelöscht werden, außer Higher-Level Privilegien wurden vergeben.

Command issued	Privilege granted	To whom it is granted
`CREATE TABLE mytable`	CONTROL on `mytable`	User issuing command
`CREATE SCHEMA myschema`	CREATEIN, ALTERIN, DROPIN on `myschema`, plus the ability to grant these to others	User issuing command
`CREATE VIEW myview`	CONTROL on `myview` only if CONTROL is held on all tables and views referenced in the definition of `myview`	User issuing command
`CREATE DATABASE mydb`	SELECT on `mydb 's` system catalog tables, IMPLICIT_SCHEMA on `mydb`*	PUBLIC**

* beim erstellen einer DB, erhält der User automatisch DBADM Autorisierung, welche implizit `CONNECT`, `CREATETAB`, `BINDADD`, `IMPLICIT_SCHEMA` und `CREATE_NOT_FENCED` beinhalten. Diese bleiben sogar nach dem entziehen von DBADM erhalten.

**PUBLIC ist spezielle DB2 Gruppe, welche alle Nutzer einer bestimten Datenbank beinhaltet. Sie muss im Gegensatz zu den anderen Gruppen nicht auf Betriebssystemebene Ebene definiert werden. Einige Privilegien wie CONNECT und SELCECT (auf Katalog Tabellen) werden automatisch an diese Gruppe vergeben. GRANT und REVOKE kann auf PUBLIC wie gewohnt angewandt werden.

Implizite Privilegien

DB2 vergibt Rechte u.U. automatisch, wenn bestimmte Kommandos benutzt werden, ohne das ein extra `GRANT` notwendig ist. Diese Rechte werden auch wieder automatisch entzogen wenn die zugehörigen Objekte gelöscht werden, außer Higher-Level Privilegien wurden vergeben.

Command issued	**Privilege granted**	**To whom it is granted**
`CREATE TABLE mytable`	CONTROL on `mytable`	User issuing command
`CREATE SCHEMA myschema`	CREATEIN, ALTERIN, DROPIN on `myschema`, plus the ability to grant these to others	User issuing command
`CREATE VIEW myview`	CONTROL on `myview` only if CONTROL is held on all tables and views referenced in the definition of `myview`	User issuing command
`CREATE DATABASE mydb`	SELECT on `mydb 's` system catalog tables, IMPLICIT_SCHEMA on `mydb`*	PUBLIC**

* beim erstellen einer DB, erhält der User automatisch DBADM Autorisierung, welche implizit `CONNECT`, `CREATETAB`, `BINDADD`, `IMPLICIT_SCHEMA` und `CREATE_NOT_FENCED` beinhalten. Diese bleiben sogar nach dem entziehen von DBADM erhalten.

**PUBLIC ist spezielle DB2 Gruppe, welche alle Nutzer einer bestimten Datenbank beinhaltet. Sie muss im Gegensatz zu den anderen Gruppen nicht auf Betriebssystemebene Ebene definiert werden. Einige Privilegien wie CONNECT und SELCECT (auf Katalog Tabellen) werden automatisch an diese Gruppe vergeben. GRANT und REVOKE kann auf PUBLIC wie gewohnt angewandt werden.

Indirekte Privilegien

- Privilegien können indirekt vergeben werden wenn Pakete (ein oder mehrere SQL Statements, welche in ein DB2 internes Format umgewandelt wurden) durch einen Datenbank Manager ausgeführt werden.
- wenn ein Paket nur statische Statements enthält, kann dieses schon mit nur dem EXECUTE Privileg ausgeführt werden.
- angenommen das `db2package1` führt: `db2 select * from org` `db2 insert into test values (1, 2, 3)` aus.
- in diesem Fall würde ein Nutzer mit nur dem EXECUTE Privileg auf `db2package1` indirekt ein SELECT und INSERT Privileg auf die Tabelle `test` erhalten.

Indirekte Privilegien

- Privilegien können indirekt vergeben werden wenn Pakete (ein oder mehrere SQL Statements, welche in ein DB2 internes Format umgewandelt wurden) durch einen Datenbank Manager ausgeführt werden.
- wenn ein Paket nur statische Statements enthält, kann dieses schon mit nur dem EXECUTE Privileg ausgeführt werden.
- angenommen das `db2package1` führt: `db2 select * from org` `db2 insert into test values (1, 2, 3)` aus.
- in diesem Fall würde ein Nutzer mit nur dem EXECUTE Privileg auf `db2package1` indirekt ein SELECT und INSERT Privileg auf die Tabelle `test` erhalten.

Indirekte Privilegien

- Privilegien können indirekt vergeben werden wenn Pakete (ein oder mehrere SQL Statements, welche in ein DB2 internes Format umgewandelt wurden) durch einen Datenbank Manager ausgeführt werden.
- wenn ein Paket nur statische Statements enthält, kann dieses schon mit nur dem EXECUTE Privileg ausgeführt werden.
- angenommen das `db2package1` führt: `db2 select * from org` `db2 insert into test values (1, 2, 3)` aus.
- in diesem Fall würde ein Nutzer mit nur dem EXECUTE Privileg auf `db2package1` indirekt ein SELECT und INSERT Privileg auf die Tabelle `test` erhalten.

Indirekte Privilegien

- Privilegien können indirekt vergeben werden wenn Pakete (ein oder mehrere SQL Statements, welche in ein DB2 internes Format umgewandelt wurden) durch einen Datenbank Manager ausgeführt werden.
- wenn ein Paket nur statische Statements enthält, kann dieses schon mit nur dem EXECUTE Privileg ausgeführt werden.
- angenommen das `db2package1` führt: `db2 select * from org` `db2 insert into test values (1, 2, 3)` aus.
- in diesem Fall würde ein Nutzer mit nur dem EXECUTE Privileg auf `db2package1` indirekt ein SELECT und INSERT Privileg auf die Tabelle `test` erhalten.

Zusammenfassung

- Elements of a DB2 security plan: You should understand the structure of the entire DB2 environment, which includes client, servers, gateways, and hosts. You should also understand authentication, authorization, and privileges.
- DB2 authentication types: You should know how to set authentication types using the db2 update dbm cfg using authentication type command on the server, and using the db2 catalog database command on the gateway and client.
- DB2 authorities: You should understand the basics of the SYSADM, SYSCTRL, and SYSMAINT authorities, which are set in the DBM CFG file, and DBADM and LOAD authorities, which are set via the GRANT command and revoked using the REVOKE command. You should definitely know what command each authority is allowed to run.
- DB2 privileges: You should have an understanding of the different types of privileges and what they allow a user to do. Examples are CONTROL, INSERT, DELETE, CREATEIN, DROPIN, REFERENCES, and SELECT. You should also know how a privilege is obtained/revoked explicitly (GRANT/REVOKE commands), implicitly, or (for packages only) indirectly.

Zusammenfassung

- Elements of a DB2 security plan: You should understand the structure of the entire DB2 environment, which includes client, servers, gateways, and hosts. You should also understand authentication, authorization, and privileges.
- DB2 authentication types: You should know how to set authentication types using the db2 update dbm cfg using authentication type command on the server, and using the db2 catalog database command on the gateway and client.
- DB2 authorities: You should understand the basics of the SYSADM, SYSCTRL, and SYSMAINT authorities, which are set in the DBM CFG file, and DBADM and LOAD authorities, which are set via the GRANT command and revoked using the REVOKE command. You should definitely know what command each authority is allowed to run.
- DB2 privileges: You should have an understanding of the different types of privileges and what they allow a user to do. Examples are CONTROL, INSERT, DELETE, CREATEIN, DROPIN, REFERENCES, and SELECT. You should also know how a privilege is obtained/revoked explicitly (GRANT/REVOKE commands), implicitly, or (for packages only) indirectly.

Zusammenfassung

- Elements of a DB2 security plan: You should understand the structure of the entire DB2 environment, which includes client, servers, gateways, and hosts. You should also understand authentication, authorization, and privileges.
- DB2 authentication types: You should know how to set authentication types using the db2 update dbm cfg using authentication type command on the server, and using the db2 catalog database command on the gateway and client.
- DB2 authorities: You should understand the basics of the SYSADM, SYSCTRL, and SYSMAINT authorities, which are set in the DBM CFG file, and DBADM and LOAD authorities, which are set via the GRANT command and revoked using the REVOKE command. You should definitely know what command each authority is allowed to run.
- DB2 privileges: You should have an understanding of the different types of privileges and what they allow a user to do. Examples are CONTROL, INSERT, DELETE, CREATEIN, DROPIN, REFERENCES, and SELECT. You should also know how a privilege is obtained/revoked explicitly (GRANT/REVOKE commands), implicitly, or (for packages only) indirectly.

Zusammenfassung

- Elements of a DB2 security plan: You should understand the structure of the entire DB2 environment, which includes client, servers, gateways, and hosts. You should also understand authentication, authorization, and privileges.
- DB2 authentication types: You should know how to set authentication types using the db2 update dbm cfg using authentication type command on the server, and using the db2 catalog database command on the gateway and client.
- DB2 authorities: You should understand the basics of the SYSADM, SYSCTRL, and SYSMAINT authorities, which are set in the DBM CFG file, and DBADM and LOAD authorities, which are set via the GRANT command and revoked using the REVOKE command. You should definitely know what command each authority is allowed to run.
- DB2 privileges: You should have an understanding of the different types of privileges and what they allow a user to do. Examples are CONTROL, INSERT, DELETE, CREATEIN, DROPIN, REFERENCES, and SELECT. You should also know how a privilege is obtained/revoked explicitly (GRANT/REVOKE commands), implicitly, or (for packages only) indirectly.

Zusammenfassung

Fragen?

Rechtschreibung nur interpoliert.