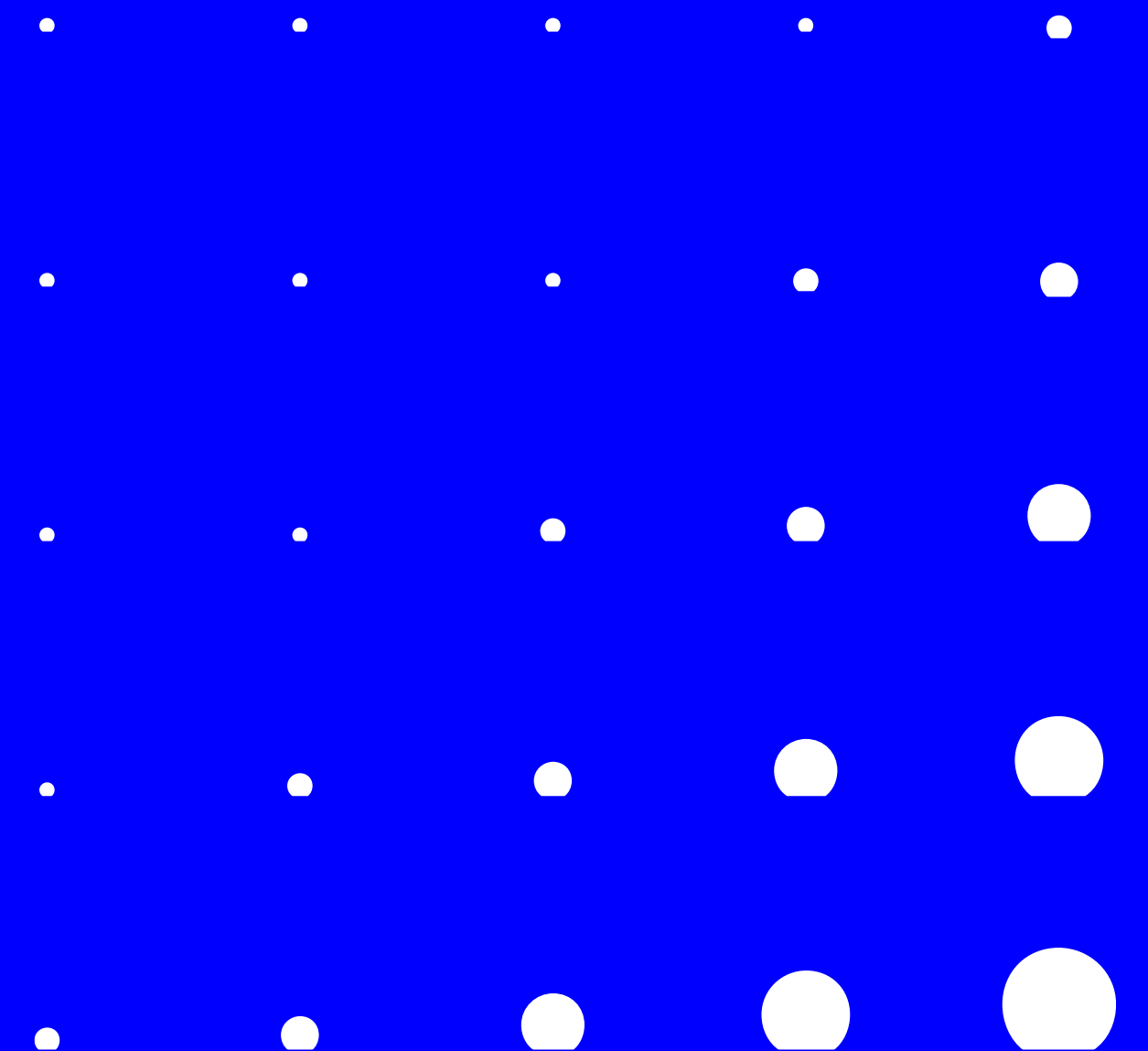


SQL: DML & DCL

Data Manipulation & Control Language



Was ist die DML?

- Die Data Manipulation Language DML ist derjenige Teil einer Datenbanksprache, der verwendet wird, um Daten
 - zu schreiben,
 - zu ändern und
 - zu löschen.

DML – INSERT

Neue Daten einfügen

- **INSERT INTO Student (MatrNr, Name) VALUES (27123, `Meier`)**
fügt eine Zeile mit den geg. Werten für die Spalten MatrNr und Name in die Tabelle Student hinzu.
- **INSERT INTO Student (MatrNr, Name)**
VALUES (27124, `Schulz`), (27125, `Schmidt`)
fügt zwei Zeilen mit den angegebenen Werten für die Spalten MatrNr und Name in die Tabelle Student hinzu.
- **INSERT INTO Student VALUES (27126, `Schmidt`)**
 - Beim INSERT-Statement kann die erste Klammer mit den Attribut-Namen auch weggelassen und direkt mit values() die Werte eingefügt werden.
 - Allerdings müssen dann die Werte in der gleichen Reihenfolge wie in der Tabellendefinition angegeben werden.
 - Zusätzlich müssen Werte für alle Spalten der Tabelle bereitgestellt werden.
- **INSERT INTO Student (MatrNr, Name)**
SELECT MatrNr, Name FROM Student_alt
kopiert alle Studenten von der Tabelle Student_alt in die Tabelle Student und kann daher gut als Backup verwendet werden!

DML – UPDATE

Vorhandene Daten aktualisieren

- **UPDATE Student SET Name = `Meier` WHERE MatrNr = 27124**
ändert den Wert der Spalte Name in der Tabelle Student für eine bestimmte MatrNr.
- **UPDATE Student SET vorname = 'Lisa', nachname = 'Müller', geburtsdatum = '2000-05-10' WHERE id = 8**
aktualisiert mehrere Spalten gleichzeitig (vorname, nachname, geburtsdatum) für den Studenten mit id = 8.
- **UPDATE Bestellung b JOIN Kunde k ON b.kundennummer = k.kundennummer SET b.status = 'Versendet' WHERE k.name = 'Meier'**
verknüpft die Tabelle Bestellung mit Kunde und setzt den Bestellstatus auf 'Versendet' für alle Bestellungen des Kunden 'Meier'.
- **UPDATE Produkt SET preis = preis * 1.10 WHERE kategorie = 'Elektronik' ;**
erhöht den Preis aller Produkte der Kategorie 'Elektronik' um 10% und berechnet den neuen Wert direkt aus der bestehenden Spalte.

DML – Löschen & Leeren

- **DELETE FROM Student WHERE . . .**
 - Löscht Daten aus der Tabelle, aber die Struktur bleibt erhalten.
 - Kann mit einer WHERE-Bedingung nur bestimmte Zeilen löschen.
 - Auto-Increment-Werte werden nicht zurückgesetzt.
 - Ist langsam, weil jede Zeile einzeln gelöscht und protokolliert wird.
- **TRUNCATE TABLE Student**
 - Löscht alle Daten, aber behält die Tabellenstruktur.
 - Schneller als DELETE, da es keine Zeilen einzeln löscht.
 - Kann nicht mit WHERE verwendet werden; löscht also immer alles.
 - Auto-Increment wird zurückgesetzt; wenn vorhanden, beginnt die Zählung wieder bei 1.
 - Kann nicht rückgängig gemacht werden; ROLLBACK funktioniert nicht.

DML – Löschen & Leeren

- **DROP TABLE Student**
 - Löscht die gesamte Tabelle inklusive Daten und Struktur.
 - Kann nicht mit WHERE verwendet werden; löscht also immer alles.
 - Auto-Increment geht verloren.
 - Kann nicht rückgängig gemacht werden; ROLLBACK funktioniert nicht.
 - Falls benötigt, muss die Tabelle neu erstellt werden.

DML – Löschen & Leeren: Wann was verwenden?

Befehl	Löscht Daten?	Löscht Tabellenstruktur?	Kann WHERE-Bedingung verwenden?	Auto-Increment zurückgesetzt?	Transaktionsprotokoll?	Rollback möglich?	Geschwindigkeit
<code>DELETE</code> <code>FROM</code> <code>Student</code>	✓ Ja	✗ Nein	✓ Ja	✗ Nein	✓ Ja (jede Zeile einzeln)	✓ Ja (mit TRANSACTION)	🐢 Langsam (Zeile für Zeile)
<code>TRUNCATE</code> <code>TABLE</code> <code>Student</code>	✓ Ja	✗ Nein	✗ Nein (löscht alles)	✓ Ja	✗ Nein	✗ Nein (kein Rollback)	🚀 Sehr schnell
<code>DROP</code> <code>TABLE</code> <code>Student</code>	✓ Ja	✓ Ja	✗ Nein (löscht alles)	✗ Nicht mehr vorhanden	✗ Nein	✗ Nein	🚀 Sehr schnell

DML – Löschen & Leeren: Wann was verwenden?

- **DELETE FROM**
 - Wenn nur bestimmte Zeilen gelöscht werden sollen und Rollback möglich sein soll.
- **TRUNCATE TABLE**
 - Wenn alle Zeilen entfernt werden sollen, aber die Struktur der Tabelle erhalten bleibt.
- **DROP TABLE**
 - Wenn die ganze Tabelle (inkl. Struktur) komplett gelöscht werden soll.

DCL

Data Control Language

Was ist die DCL?

- Die Data Control Language (DCL) ist eine Unterkategorie von SQL, die sich mit der Verwaltung von Benutzerrechten und Berechtigungen in einer Datenbank befasst.
- Es gibt zwei Hauptbefehle in DCL:
 - **GRANT** → Erteilt Rechte an Benutzer oder Rollen.
 - **REVOKE** → Entzieht Rechte von Benutzern oder Rollen.
- Die Kommandos zur Rechteverwaltung sind in SQL spezifiziert, nicht jedoch die zur Benutzerverwaltung.
 - Daher implementiert jedes DBMS seine eigene Benutzerverwaltung.
 - In modernen DBMS können Rechte nicht nur auf einzelne Tabellen vergeben werden.

DCL-Syntax: GRANT – Rechte vergeben

- `@ ' % '`
Der Benutzer kann sich von überall aus verbinden.
- `@ ' 192.168.1. % '`
Der Benutzer kann sich nur aus einem bestimmten Subnetz verbinden.
- `@ ' company . com '`
Zugriff nur für Benutzer, die sich von dieser Domain anmelden.
- **WITH GRANT OPTION**
Der Benutzer darf die erhaltenen Rechte an andere weitergeben.

DCL-Syntax: GRANT – Rechte vergeben

- Lesezugriff für einen lokalen Benutzer
 - **GRANT SELECT ON schule.Student TO 'benutzer1'@'localhost'**
- Lese- und Schreibrechte für einen Benutzer, der sich von überall verbinden kann
 - **GRANT SELECT, INSERT, UPDATE ON schule.Student TO 'benutzer2'@'%'**
- Alle Rechte auf eine Datenbank für einen Admin, der sich nur von einer bestimmten IP verbinden kann
 - **GRANT ALL PRIVILEGES ON schule.* TO 'admin'@'192.168.1.10'**
- Nur DELETE-Rechte für einen Benutzer, der sich von einem bestimmten Subnetz aus verbindet
 - **GRANT DELETE ON schule.Lehrer TO 'benutzer3'@'192.168.1.%'**

DCL-Syntax: GRANT – Rechte vergeben

- Zugriff auf eine bestimmte Spalte für einen Benutzer mit einer festen IP-Adresse erlauben
 - **GRANT SELECT (Noten) ON schule.Prüfung TO 'benutzer4'@'10.0.0.5'**
- Rechte für mehrere Benutzer gleichzeitig vergeben (lokale und entfernte Benutzer)
 - **GRANT SELECT, INSERT ON schule.Kurs TO 'user1'@'localhost', 'user2'@'192.168.1.20'**
- EXECUTE-Rechte für gespeicherte Prozeduren für einen entfernten Benutzer erlauben
 - **GRANT EXECUTE ON PROCEDURE schule.berechne_durchschnitt TO 'benutzer5'@'company.com'**
- Erlaubnis zur Erstellung neuer Benutzer für einen Superadmin von überall aus
 - **GRANT CREATE USER ON *.* TO 'superadmin'@'%'**

DCL-Syntax: GRANT – Rechte vergeben

- Temporäre Tabellenrechte für einen Benutzer von einer internen IP-Adresse erlauben
 - **GRANT CREATE TEMPORARY TABLES ON schule.* TO 'benutzer6'@'172.16.0.100'**
- Grant mit Weitergaberecht; erlaubt dem Benutzer, Rechte weiterzugeben
 - **GRANT SELECT ON schule.Student TO 'benutzer7'@'%' WITH GRANT OPTION**
- Einem Benutzer das Ausführen einer Stored Procedure erlauben
 - **GRANT EXECUTE ON PROCEDURE schule.berechne_durchschnitt TO 'benutzer1'@'%'**
- Einem Benutzer das Ändern einer Stored Procedure erlauben
 - **GRANT ALTER ROUTINE ON PROCEDURE schule.berechne_durchschnitt TO 'admin'@'192.168.1.10'**

DCL-Syntax: REVOKE – Rechte entziehen

- Lesezugriff für einen Benutzer entfernen
 - **REVOKE SELECT ON schule.Student FROM 'benutzer1'@'localhost'**
- bestimmte Schreibrechte von einem globalen Benutzer entfernen
 - **REVOKE INSERT, UPDATE ON schule.Student FROM 'benutzer2'@'%'**
- Alle Rechte von einem Administrator entfernen, der sich nur von einer bestimmten IP verbinden kann
 - **REVOKE ALL PRIVILEGES ON schule.* FROM 'admin'@'192.168.1.10'**
- Nur DELETE-Rechte für einen Benutzer aus einem bestimmten Subnetz entfernen
 - **REVOKE DELETE ON schule.Lehrer FROM 'benutzer3'@'192.168.1.%'**

DCL-Syntax: REVOKE – Rechte entziehen

- Zugriff auf eine bestimmte Spalte entziehen
 - **REVOKE SELECT (Noten) ON schule.Prüfung FROM 'benutzer4'@'10.0.0.5';**
- Rechte für mehrere lokale und entfernte Benutzer gleichzeitig entziehen
 - **REVOKE SELECT, INSERT ON schule.Kurs FROM 'user1'@'localhost', 'user2'@'192.168.1.20';**
- EXECUTE-Rechte für gespeicherte Prozeduren für einen entfernten Benutzer entziehen
 - **REVOKE EXECUTE ON PROCEDURE schule.berechne_durchschnitt FROM 'benutzer5'@'company.com';**
- Erlaubnis zur Erstellung neuer Benutzer entziehen
 - **REVOKE CREATE USER ON *.* FROM 'superadmin'@'%';**

DCL-Syntax: REVOKE – Rechte entziehen

- Temporäre Tabellenrechte für einen Benutzer aus einer internen IP entfernen
 - **REVOKE CREATE TEMPORARY TABLES ON schule.* FROM 'benutzer6'@'172.16.0.100'**
- Weitergaberecht verhindern
 - **REVOKE GRANT OPTION ON schule.Student FROM 'benutzer7'@'%'**
- Der Benutzer benutzer1 darf die Stored Procedure nicht mehr ausführen
 - **REVOKE EXECUTE ON PROCEDURE schule.berechne_durchschnitt FROM 'benutzer1'@'%'**
- Der Benutzer admin kann die Stored Procedure nicht mehr ändern oder neu kompilieren
 - **REVOKE ALTER ROUTINE ON PROCEDURE schule.berechne_durchschnitt FROM 'admin'@'192.168.1.10'**