SQL_DML_DCL_Cheatsheet_#5

DML Data Manipulation Language

Die DML wird verwendet, um in einer Datenbank Daten:

- Einzufügen → INSERT
- zu Ändern → UPDATE
- zu Löschen → DELETE

Beispiel Tabelle Mitglieder:

Field	Type	Null	Key	Default	Extra
PK_ID_Verein	int(11)	NO	PRI	NULL	auto_increment
Nachname	varchar(20)	NO		NULL	
Nutzername	varchar(20)	YES	UNI	NULL	
Telefonnummer	varchar(14)	YES		NULL	
Mitgliedsbeitrag	decimal(6,2)	YES		NULL	

Datensätze einfügen INSERT

Das Einfügen von Datensätzen erfolgt mithilfe der Anweisung INSERT. Geben Sie stets die Spalten an. Auto_Increment Spalten sollten nicht ausgefüllt werden.

Syntax:

```
INSERT INTO Tabellenname
(Spalte1, Spalte2, Spalte3, …, SpalteX)
VALUES
('Wert1', 'Wert2', 'Wert3', 'WertX');
```

Beispiel:

```
INSERT INTO Mitglieder
(Nachname, Nutzername, Telefonnummer, Mitgliedsbeitrag)
VALUES
('Müller', 'Joe', '0911996633','47.98');
```

Mehrere Datensätze einfügen

Syntax:

```
INSERT INTO Tabellenname
(Spalte1, Spalte2, Spalte3, …, SpalteX)
VALUES
('Wert1', 'Wert2', 'Wert3', 'WertX');
```

Beispiel:

```
INSERT INTO Mitglieder
(Nachname, Nutzername, Telefonnummer, Mitgliedsbeitrag)
VALUES
('Huahi', 'Julia', '09111236633','47.98'),
('Prim', 'Theresa', '0911993453','34.98'),
('Koch', 'Marvin', '09111325253','28.28');
```

Datensätze aus einer bestehenden Tabelle einfügen

Beim Einfügen aus einer anderen Tabelle müssen die Spalten (Datentyp und Anzahl) identisch sein.

Syntax:

```
INSERT INTO Tabellenname1
(Spalte1, Spalte2, Spalte3, …, SpalteX)
SELECT
(Spalte1, Spalte2, Spalte3, …, SpalteX)
FROM Tabellename2
WHERE Bedingung;
```

Datensätze löschen DELETE

Das Löschen von Datensätzen erfolgt mithilfe der Anweisung DELETE. Verwenden Sie stets den Primärschlüssel, um einen Datensatz eindeutig zu identifizieren.

Syntax:

```
DELETE FROM Tabellenname
WHERE SpalteX = 'Wert';
```

Beispiel:

```
DELETE FROM Mitglieder
WHERE PK_ID_Verein = 5;
```

Datensätze ändern UPDATE

Das Ändern von Datensätzen erfolgt mithilfe der Anweisung UPDATE. Um nicht alle Datensätze einer Spalte zu verändern, sollte mithilfe der WHERE – Klausel eine Bedingung eingegeben werden. Verwenden Sie stets den Primärschlüssel, um einen Datensatz eindeutig zu identifizieren.

Syntax:

```
UPDATE Tabellenname
SET FeldX = 'NeuerWert'
WHERE Bedingung;
```

Beispiel:

```
UPDATE Mitglieder
SET Telefonnummer = '095171144'
WHERE PK_ID_Verein = 22;
```

Beispiel Änderung aller Einträge einer Spalte:

Erhöhung des Mitgliedsbeitrags aller Mitglieder um 2 Prozent.

```
UPDATE Mitglieder
SET Mitgliedsbeitrag = Mitgliedsbeitrag * 1.02;
```

DCL Data Control Language

Die DCL wird verwendet, um in einem Datenbankmanagementsystem Benutzer und Berechtigungen zu verwalten:

- Benutzer anlegen → CREATE
- Rechte Vergeben → GRANT
- Rechte Entziehen → REVOKE
- Benutzer löschen → DROP

Die folgenden Beispiele zeigen nur einen kleinen Auszug aus der DCL. Die komplette Dokumentation finden Sie hier: https://mariadb.com/kb/en/account-management-sql-commands/

Benutzer anlegen CREATE

```
CREATE USER 'Benutzername' IDENTIFIED BY 'Passwort';
```

Rechte vergeben GRANT

Zugriff auf Tabelle Mitglieder in der Datenbank Verein:

```
GRANT ALL ON Verein.Mitglieder TO 'Benutzername';
```

Zugriff auf die komplette Datenbank Verein:

```
GRANT ALL ON Verein.* TO 'Benutzername';
```

Zugriff auf alle Datenbanken:

```
GRANT ALL ON *.* TO 'Benutzername';
```

Nutzer darf nur neue Werte in der Datenbank Verein eingeben:

```
GRANT INSERT ON Verein.* TO 'Benutzername';
```

Nutzer darf nur Werte aus der Datenbank Verein auslesen:

```
GRANT SELECT ON Verein.* TO 'Benutzername';
```

Rechte entziehen REVOKE

Nutzer Rechte von der Datenbank Verein entziehen:

```
REVOKE DROP ON Verein.* FROM 'Benutzername';
```

Benutzer löschen DROP

```
DROP USER 'Benutzername';
```