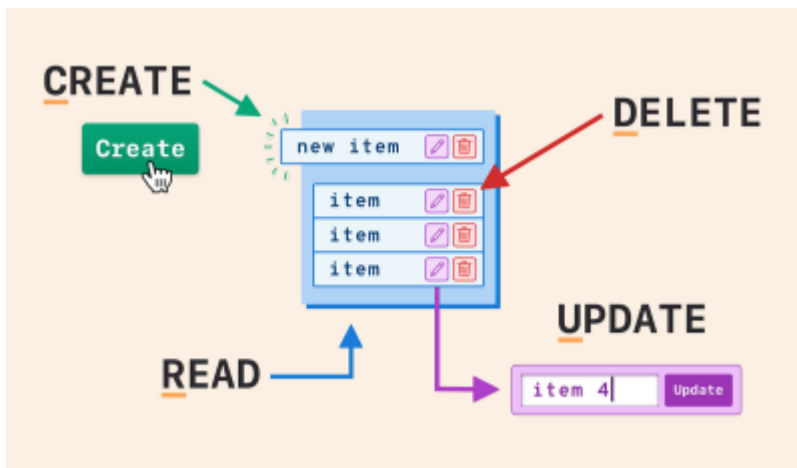


# LU07a - CRUD: Die Basics der Datenverwaltung

**CRUD** ist ein Akronym für **Create, Read, Update, Delete**. Diese vier Operationen sind die **Grundbausteine für den Umgang mit Daten** – egal ob in einer Datenbank, einer App oder einem Webservice.

Stellen Sie sich CRUD wie einen Werkzeugkasten vor, mit dem Sie immer wieder die gleichen Tätigkeiten machen:

- **Create** – etwas Neues hinzufügen
- **Read** – vorhandene Daten abrufen
- **Update** – bestehende Daten ändern
- **Delete** – Daten löschen



## CRUD in der Praxis

CRUD begegnet Ihnen nicht nur in der Datenbank, sondern überall in der Informatik. Beispiel: Ein Social-Media-Post

- **Create** → Sie schreiben einen neuen Post → er wird in der Datenbank gespeichert.
- **Read** → Andere Nutzer:innen sehen Ihren Post → die App liest die Daten aus der Datenbank.
- **Update** → Sie ändern den Text oder fügen ein Bild hinzu → die Daten in der Datenbank werden angepasst.
- **Delete** → Sie löschen den Post → der Eintrag wird aus der Datenbank entfernt.

## CRUD und SQL

In diesem Kapitel lernen Sie die sogenannten **DML-Befehle** (Data Manipulation Language). Mit DML-Befehlen setzen Sie CRUD direkt in SQL um:

CRUD	SQL-Befehl (typisch)
------	----------------------

<b>Create</b>	INSERT INTO ... VALUES (...);
<b>Read</b>	SELECT ... FROM ... WHERE ...;
<b>Update</b>	UPDATE ... SET ... WHERE ...;
<b>Delete</b>	DELETE FROM ... WHERE ...;

## CRUD in der Praxis

CRUD/DML-Befehle werden in der Praxis auf **zwei Wegen** ausgeführt:

- **Als SQL-Skript:**

Ein Skript ist eine einfache **Textdatei mit SQL-Befehlen** (Dateiendung `.sql`).

Sie können das Skript speichern, teilen, mehrfach ausführen → das Ergebnis ist immer gleich.

→ Beispiel: 100 neue Produkte (Sneaker) in einer Datenbank einfügen – ein Skript erledigt das in Sekunden.

- **Über die Backend-Logik:**

Moderne Apps (z. B. ein Social-Media-Dienst oder ein Webshop) schicken Befehle vom Frontend ins Backend.

Dort werden die passenden SQL-Operationen ausgeführt.

→ Beispiel: Beim Klicken auf "Post erstellen" sendet die App einen **\*\*REST-Call\*\*** ((REST-Call|Ein Aufruf von einem Programm (z. B. einer Website oder App) an ein Backend. Dabei wird ein HTTP-Befehl wie GET, POST, PUT oder DELETE verwendet, um Daten von einer Datenbank abzufragen oder zu verändern.))

an den Server, der wiederum ein **INSERT** in die Datenbank macht.

Sie werden CRUD sowohl in SQL-Skripten (z. B. für Migrationen, Seeds oder Wartungsarbeiten) als auch indirekt über Backend-Logik (REST, APIs) antreffen. Der Vorteil von SQL-Skripten liegt darin, dass sie nachvollziehbar sind: Sie lassen sich speichern, mehrfach ausführen und es genau dokumentiert, welche Befehle ausgeführt werden. Gleichzeitig sind sie effizient und sauber, da grosse Datenmengen damit schnell und mit den richtigen Datentypen eingefügt werden können – im Gegensatz zu fehleranfälligeren Importen über grafische Oberflächen.

## Beispiele (Film-DB)

- **Create:** einen Film hinzufügen

```
INSERT INTO imdb_top_1000 (Series_Title, Released_Year, Genre, IMDB_Rating)
```

```
VALUES ('Dune', 2021, 'Action, Adventure, Sci-Fi', 8.0);
```

- **Read:** Filme ab 2010 mit Top-Bewertung auslesen

```
SELECT Series_Title, Released_Year, IMDB_Rating
FROM imdb_top_1000
WHERE Released_Year >= 2010
ORDER BY IMDB_Rating DESC;
```

- **Update:** Genre aktualisieren

```
UPDATE imdb_top_1000
SET Genre = 'Drama, War'
WHERE Series_Title = '1917' AND Released_Year = 2019;
```

- **Delete:** Tuple (Datensatz) entfernen

```
DELETE FROM imdb_top_1000
WHERE Series_Title = 'Dummy Movie' AND Released_Year = 1900;
```

## Merksatz

**CRUD ist das „Alphabet“ der Datenarbeit.** Egal ob API-Call oder SQL-Skript – am Ende passieren Create/Read/Update/Delete auf Tabellen.

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
[https://wiki.bzz.ch/modul/m290\\_guko/learningunits/lu07/theorie/a\\_crud?rev=1758661647](https://wiki.bzz.ch/modul/m290_guko/learningunits/lu07/theorie/a_crud?rev=1758661647)

Last update: **2025/09/23 23:07**

