

M164_Datenbanken erstellen und Daten einfügen

Tag 3

Recap

Datentypen in drei Gruppen:

- String
- Numerische
- Datum/Zeit Datentypen

DATENTYP	BESCHREIBUNG	SPEICHER-BEDARF
VARCHAR(n)	Zeichenkette mit maximal n Zeichen, 65.535 möglich	n + 1 Bytes (n = maximale Länge)
CHAR(n)	Zeichenkette mit genau n Zeichen, max. 255 Zeichen	n Bytes
INT	Ganzzahl zwischen -2,147 und +2,147 Milliarden	4 Bytes
BIGINT	Ganzzahl zwischen -9223372036854775808 (-2 ⁶³) und 9223372036854775807 (2 ⁶³ -1)	8 Bytes
FLOAT	Fließkommazahl, Wertebereich: -3.40...E+38 bis 3.40...E+38	4 Bytes
DATE	Datum im Format YYYY-MM-DD (also bspw. 2009-07-29) (ab 01.01.1000)	3 Bytes
DATETIME	Datum + Uhrzeit im Format YYYY-MM-DD hh:mm:ss (also bspw. 2009-07-29 21:31:01) (ab 01.01.1000)	8 Bytes
TIMESTAMP	Datum + Uhrzeit im Format YYYYMMDDhhmmss (ab 1970)	4 Bytes

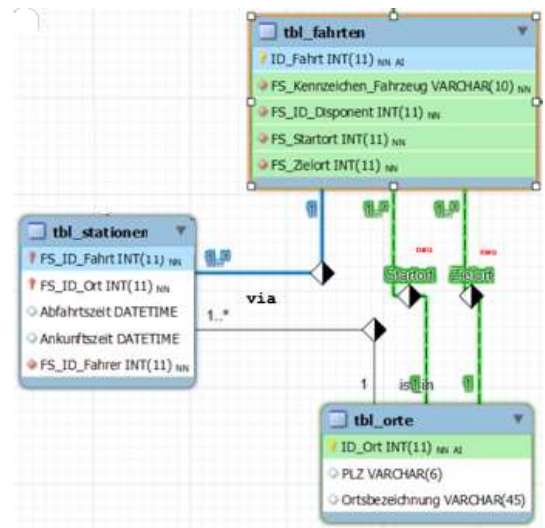
Datentyp	MariaDB (MySQL)	Beispiel	Bemerkung / Einstellungen
Ganze Zahlen	INT	1000	Standard-Ganzzahl
Natürliche Zahlen	UNSIGNED INT	5000	Nur positive Werte
Festkommazahlen (Dezimalzahlen)	DECIMAL(6,2)	1234.56	M = Gesamtstellen, D = Nachkommastellen
Aufzählungstypen	ENUM	ENUM('Ja', 'Nein')	Vordefinierte Werte
Boolean (logische Werte)	BOOLEAN	TRUE	Intern als TINYINT(1) gespeichert
Zeichen (einzelnes Zeichen)	CHAR(1)	'A'	Genau ein Zeichen
Gleitkommazahlen	FLOAT	3.14	Näherungswerte für Berechnungen
Zeichenkette fester Länge	CHAR(10)	'Hallo '	Immer exakt 10 Zeichen
Zeichenkette variabler Länge	VARCHAR(255)	'Beispieltext'	Länge flexibel bis 255 Zeichen
Datum und/oder Zeit	DATE	'2024-03-09'	Speichert nur das Datum
Zeitstempel	TIMESTAMP	'2024-03-09 12:34:56'	Automatische Aktualisierung möglich
Binäre Datenobjekte variabler Länge (z.B. Bild)	BLOB	(Bilddatei)	Speichert Binärdaten
Verbund	(Nicht direkt unterstützt)	-	Alternativ mit JSON umsetzbar
JSON	JSON	'{"name": "Max"}'	Strukturierte Daten im JSON-Format

Mehrfachbeziehungen

Zwei Tabellen haben **mehrere unabhängige Beziehungen**.

Kennzeichnung durch Rollen: Jede Beziehung muss klar benannt werden.

- Unterschiedliche Kardinalitäten: Jede Beziehung kann eigene Kardinalitäten haben.
- Beispiel: Fahrten & Orte
 - Beziehung 1: Startort (1:mc)
 - Beziehung 2: Zielort (1:mc)
 - Beziehung 3: Angefahrene Orte (mc:mc) → benötigt Transformationstabelle
- Transformationstabelle:
 - Löst mc:mc-Beziehung auf.
 - Enthält Fremdschlüssel zu beiden Tabellen.

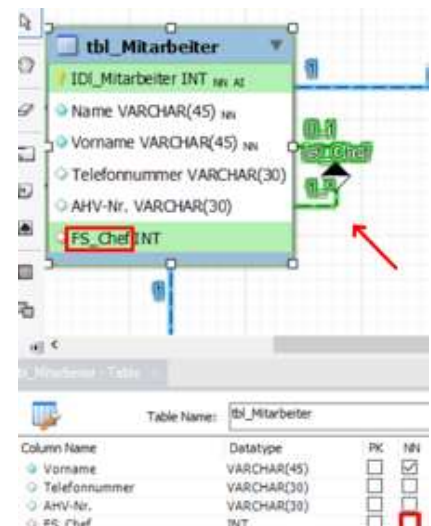


Rekursion (strenge Hierarchie)

Möglichkeit, dass an einer **Assoziation nur ein einziger Entitätstyp** beteiligt ist.

->bedeutet, dass ein Datensatz einer Tabelle mit einem anderen aus der gleichen Tabelle in Beziehung steht

- Ein Fremdschlüssel wird erhalten, der auf den Identifikationsschlüssel der eigenen Tabelle verweist.
- zB. jede Person hat genau eine andere Person als Vorgesetzten. Da die höchste Person keinen Vorgesetzten haben kann ist es eine c_mc Beziehung. Würden wir dieses Fremdschlüsselattribut auf „Not Null“ belassen (= 1:mc), könnten wir die „höchste Person“ nicht erfassen.



Einfache Hierarchie (Transformationstabelle)

- mehrere Oberelemente zugelassen sind (z. B. mehrere Projektleiter), dann handelt es sich um **eine mc:mc-Beziehung** und es wird eine Transformationstabelle benötigt, die für jeden Beziehungsgraphen einen Datensatz enthält, der angibt, welches **Oberelement zu welchem Unterelement** gehört.

Stücklistenproblem

Es gibt eine Tabelle mit Produkten und deren Eigenschaften. Jetzt kann sich aber auch ein Produkt aus mehreren anderen Produkten zusammensetzen (modulare Produkte).

- Das Problem: Liste mit Einzelteilen zu erzeugen
- Lösung mit Rekursion: Da es sich im Modell ja nicht um eine einfache Hierarchie handelt (c:mc), reicht die Erstellung eines Fremdschlüsselattributs in der Produktetabelle nicht aus. Neben der Produktetabelle benötigen wir eine weitere Tabelle, die die Zusammensetzungen enthält.