LU04c – Die Crow's-Foot-Notation

Lernziele

- Ich kann erklären, worin sich die Crow's-Foot- von der Chen-Notation unterscheidet.
- Ich kann die Symbole für 1:1, 1:n und n:m in der Crow's-Foot-Notation lesen und zeichnen.
- Ich kann in einem Crow's-Foot-Diagramm Primär- und Fremdschlüssel benennen.

Woher der Name kommt



„Crow's Foot„ bedeutet **Krähenfuss**. Der Name kommt vom Symbol für *many* (viele), das wie ein kleiner Krähenfuss aussieht:



Wozu eine zweite Notation?

In LU04b haben wir Beziehungen und Kardinalitäten in der **Chen-Notation** dargestellt. Diese ist gut geeignet, um mit Auftraggeber:innen zu sprechen – sie bleibt aber **fachlich und abstrakt**.

Die **Crow's-Foot-Notation** zeigt dieselbe Information **technischer**: Sie stellt Entitäten bereits als Tabellen dar. Damit ist sie die passende Darstellung für das **logische Datenmodell** und die letzte Station vor der Umsetzung in SQL.

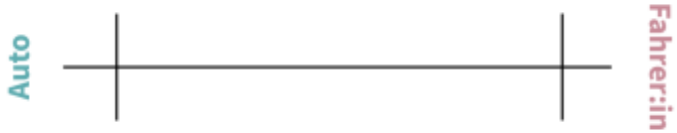
	Chen-Notation	**Crow's-Foot-Notation**
Entität	Rechteck	Tabelle mit Spaltenliste
Attribut	eigenes Oval	Zeile innerhalb der Tabelle
Beziehung	Raute mit Beschriftung	Linie mit Symbolen an den Enden
Schlüssel	unterstrichenes Attribut	mit **PK** / **FK** markiert
Eignet sich für	konzeptuelles Modell	logisches Datenmodell

Die Kardinalitäten selbst (1:1, 1:n, n:m) sind in beiden Notationen dieselben – nur die **Zeichen** unterscheiden sich. Was die Kardinalitäten bedeuten, steht in der vorherigen Seite (LU04b).

Die Symbole

1:1 – One-to-One

Darstellung: Linie mit **kurzem Querstrich** auf beiden Seiten.



Ein *Auto kann gleichzeitig nur* eine *Person fahren, und* eine *Person fährt gleichzeitig nur* ein *Auto.*

1:n – One-to-Many

Darstellung: **Querstrich** („eins“) auf der einen Seite, **Krähenfuss** („viele„) auf der anderen.



Eine *Regisseurin kann* mehrere *Filme drehen.* Ein *Film wird in der Regel von* einer *Regie gedreht.*

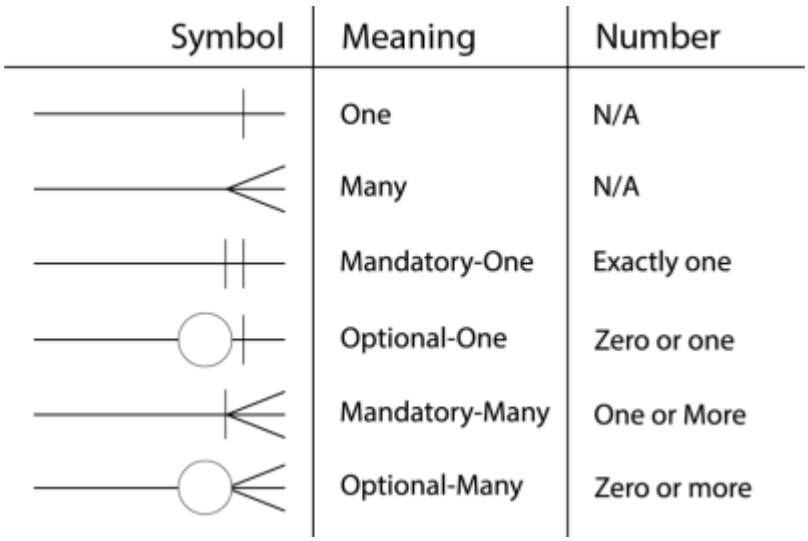
n:m – Many-to-Many

Darstellung: **Krähenfuss auf beiden Seiten**.



Ein:e *Schauspieler:in kann in* mehreren *Filmen spielen.* Ein *Film kann* mehrere *Schauspieler:innen haben.*

Die folgende Übersicht zeigt alle möglichen Symbolkombinationen:



Beispiel: Online-Buchshop

Wir haben die Entitäten **Kunde**, **Bestellung** und **Produkt**:

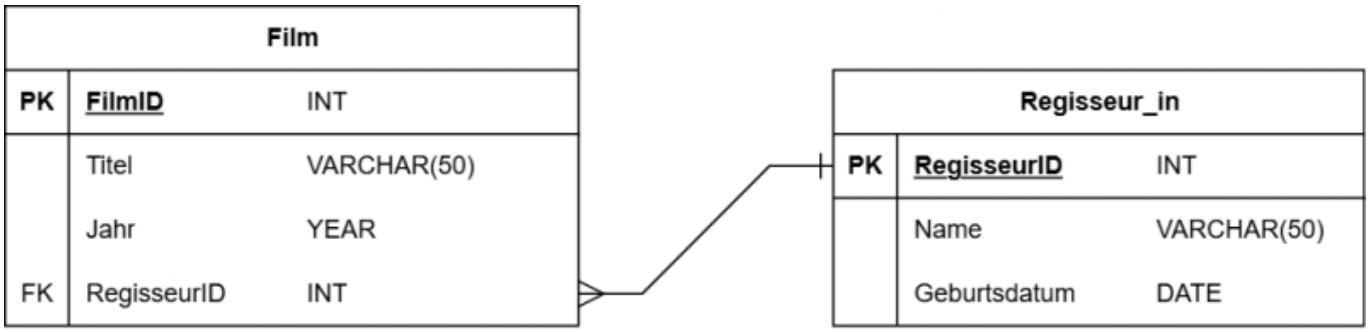
- Ein Kunde kann viele Bestellungen aufgeben → **1:n**
- Eine Bestellung kann mehrere Produkte enthalten, und ein Produkt kann in vielen Bestellungen vorkommen → **n:m**



In der Zeichnung erkennen Sie das an den Linienenden: Zwischen Kunde und Bestellung steht ein Querstrich auf der Kundenseite und ein Krähenfuss auf der Bestellungsseite. Zwischen Bestellung und Produkt stehen Krähenfüsse auf beiden Seiten.

Schlüssel im logischen Modell

Im Crow's-Foot-Diagramm werden die Schlüssel direkt eingetragen:



- **Regisseur** (RegisseurID **PK**, Name)
- **Film** (FilmID **PK**, Titel, Jahr, RegisseurID **FK**)
- Beziehung: **1 Regisseur ↔ n Filme**

Damit die Daten korrekt und widerspruchsfrei bleiben, gelten drei Regeln:

Regel	Bedeutung
Entitätsintegrität	Jede Tabelle braucht einen Primärschlüssel, und dieser darf nie leer (NULL) sein.
Referentielle Integrität	Ein Fremdschlüssel muss auf einen Primärschlüssel verweisen, den es tatsächlich gibt.
Domänenintegrität	Die Werte müssen zum vorgesehenen Datentyp passen – in einer Zahlenspalte steht kein Text.

Wie geht es weiter? In LU05 setzen wir ein solches logisches Modell in eine echte Datenbank um: Wir wählen passende **Datentypen** und legen die Tabellen mit SQL-Befehlen (CREATE TABLE …) an.

M290-LU04

Guido Koch

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

Permanent link:
https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu04/theorie/c_crowsfoot

Last update: **2026/09/06 16:22**