

Modellierung der RC Wildbach Datenbank (2)

Im letzten Kapitel haben wir durch Überlegung die folgenden Entitätstypen gefunden:

- Kunde
- Boot
- Guide
- Startzeit

Die Beziehungen zwischen Objekten sind sehr wichtig. Wenn wir die Beziehungen kennen, können wir später die verschiedenen Tabellen unserer Datenbank entsprechend verbinden.

Beziehungstypen und Kardinalitäten

Beziehung zwischen Kunde und Boot

Die Kunden des RC Wildbachs leihen oder buchen Raftingboote. Der Beziehungstyp kann also wie in Abb. 02 festgelegt werden. Beziehungstypen werden im ER-Modell grafisch durch Rauten abgebildet.

Um die **Kardinalität** zu finden, betrachtet man die Beziehung aus zwei Leserichtungen:

a) **Leserichtung ->**

Als erstes müssen wir uns fragen, ob ein Kunde nur ein bestimmtes Boot, oder auch mehrere Raftingboote ausleihen kann. Diese Frage ist leicht zu beantworten. Natürlich wollen wir, dass ein Kunde immer wieder kommt und über die Zeit viele Boote buchen möchte. (-> **n**)

b) **Leserichtung <-**

Umgekehrt kann man sagen, dass ein bestimmtes Boot von einem oder mehreren Kunden gebucht werden kann (**n <-**)

Die Beziehung zwischen den Entitätstypen Kunde und Boot besitzt also die Kardinalität **m:n** (sprich: m zu n).

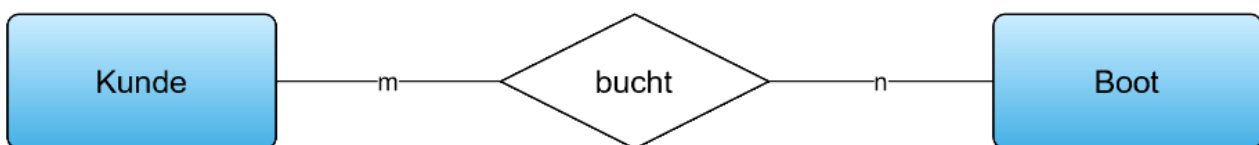
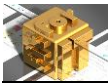


Abb. 01: Beziehungstyp „bucht“ und passende Kardinalität (**m:n**)

Im Falle eines **m:n**-Beziehungstyps kann aus dem Beziehungstyp ein eigenständiger Entitätstyp geschaffen werden. Dem neuen Entitätstyp können dann auch Attribute zugeordnet werden¹.

¹ **1:n**-Beziehungstypen haben hingegen üblicherweise keine Attribute, da diese immer einem der beteiligten Entitätstypen zugeordnet werden können.



In unserem Beispiel ist also der Beziehungstyp „bucht“ etwas Besonderes, nämlich ein eigenständiger Entitätstyp, weil er die Kardinalität **m:n** aufweist². Der Beziehungs- und zugleich Entitätstyp „bucht“ (od. Buchung) beinhaltet später in unserer Datenbank alle Buchungsdatensätze.

In unserer vereinfachten Miniwelt legen wir die nützlichen Attribute für die Buchungsdatensätze wie in Abb. 02 fest.

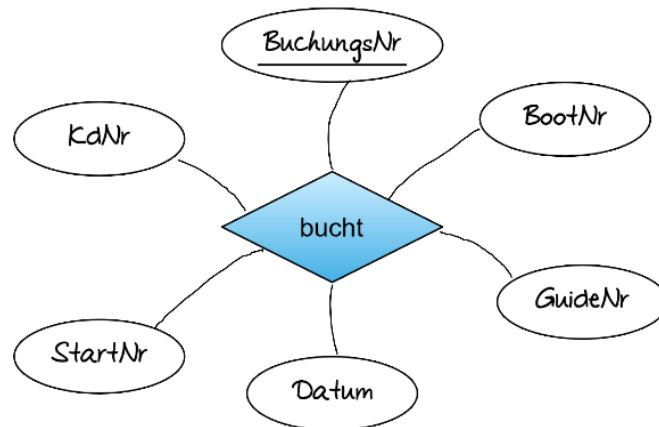


Abb. 02: Beziehungs- und Entitätstyp „bucht“ mit Attributen

Beziehung zwischen Buchung („bucht“) und Guide

Zu jeder Buchung gehört zwingend ein Guide. So lässt sich der Beziehungstyp wie in Abb. 03 schnell festlegen.

Um die **Kardinalität** zu finden, betrachtet man die Beziehung mengenmäßig von zwei Seiten:

- a) Für eine Buchung („bucht“) stehen mehrere Guides zur Wahl ($\rightarrow n$)
- b) Es kann höchstens 1 Guide ein gebuchtes Boot begleiten ($1 \leftarrow$)

Es handelt sich also um eine **1:n-Beziehung**.

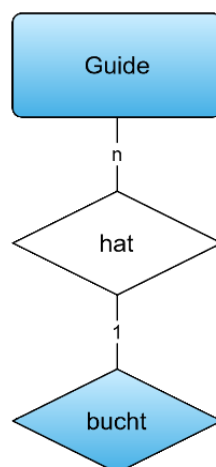
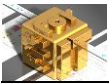


Abb. 03: Beziehungstyp „hat“ und passende Kardinalität (1:n)

² Der farbige Hintergrund für den Beziehungs- und Entitätstyp „bucht“ soll ab Abb. 02 diese Doppelrolle betonen.



Beziehung zwischen Buchung („bucht“) und Startzeit

Aus organisatorischen Gründen gibt es festgelegte Startzeiten für jede Tour. Die Startzeit ist aber nicht vom gewählten Boot abhängig, sondern man wählt sie als Kunde bei der Buchung. Der passende Name für den Beziehungstyp ist ebenfalls „hat“, weil jede Buchung eine Startzeit hat (s. Abb. 04).

Um die **Kardinalität** zu finden, betrachtet man die Beziehung wieder aus zwei Leserichtungen:

- a) **Leserichtung ->**
Für jede Buchung steht mehr als eine Startzeit zur Wahl. ($\rightarrow n$)
- b) **Leserichtung <-**
Umgekehrt gilt, dass es höchstens eine Startzeit pro Buchung geben kann ($n \leftarrow$)

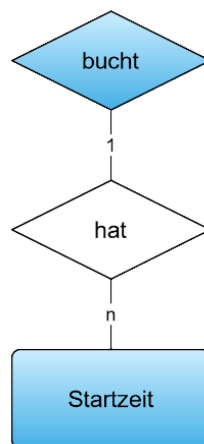


Abb. 04: Beziehungstyp „hat“ und passende Kardinalität (1:n)

Setzen wir nun alle gefundenen Entitätstypen und Beziehungstypen zusammen, haben wir unser fertiges ER-Modell (s. Abb. 05).

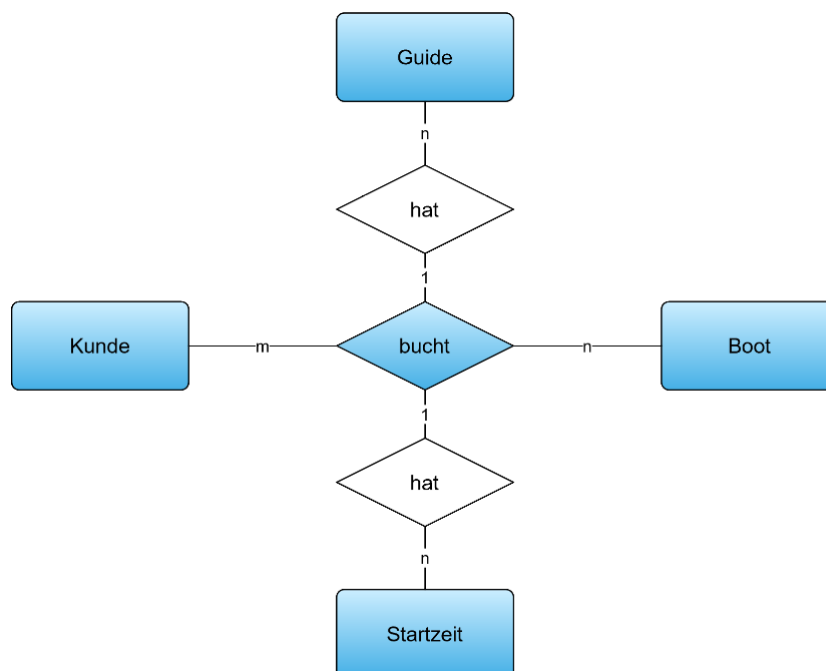
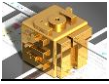


Abb. 05: ER-Modell zur Datenbank RC Wildbach



Wie der neue Beziehungs- und Entitätstyp „bucht“ mit seinen Attributen als Tabelle in der Datenbank RC-Wildbach implementiert wird, kannst du mit Hilfe dieses **Lernvideos** nachvollziehen:



Abb. 06: Startbild des Videotutoriums *Implementierung Teil 2*