



2.3.2 Datenbanksysteme II

Lerninhalte 232-05 Modellierung der RC Wildbach Datenbank (1)

Modellierung der RC Wildbach Datenbank (1)

Im Folgenden wenden wir unser bisheriges theoretisches Wissen über die Modellierung einer Datenbank an.



Abb. 01: Rafting Anbieter RC Wildbach

Aufgabe:

Der **Rafting-Club** RC Wildbach **verleiht** seit kurzem seine **Schlauchboote** samt **Guide** gegen eine Gebühr.

Mit einer Datenbank sollen die Kunden, die Boote, die Guides und alle Buchungen verwaltet werden. Aus organisatorischen Gründen gibt es festgelegte Startzeiten für jede Tour.

Achtung:

In den folgenden Kapiteln wird die Modellierung der Datenbank RC Wildbach auf **Grundlage** der Kundentabelle aus dem **Modul 2.3.1 Datenbanksysteme I** fortgeführt.

Solltest du dein **Grundwissen** zu bestimmten Themen (z. B. Formulare, Abfragen, Bericht oder Serienbrief) auffrischen wollen, dann schau dir einfach noch einmal die entsprechenden **Kapitel** und **Videos** (ab 231-05) an.

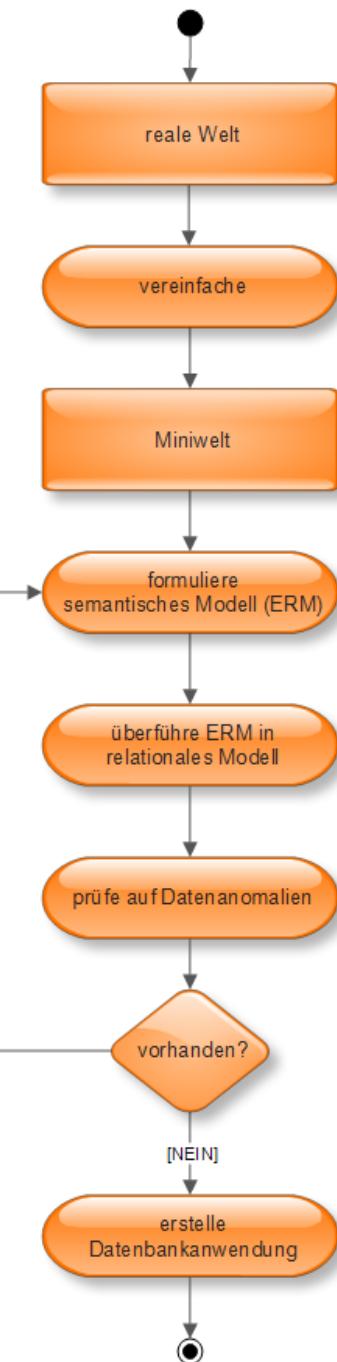


Abb. 02: Aktivitätsdiagramm Datenmodellierung

Von der realen Welt zur Miniwelt

Natürlich kann und muss ein Datenmodell nicht alle Aspekte der Wirklichkeit berücksichtigen. Was genau erforderlich ist, richtet sich immer nach der zu lösenden konkreten Aufgabe. Im Berufsleben hat ein Datenbankentwickler (oder Projektteam) eine detaillierte schriftliche Beschreibung der Fähigkeiten, die die Datenbank einmal haben soll. Dieses Pflichtenheft wird vorher im Gespräch mit dem Auftraggeber gemeinsam erstellt.



2.3.2 Datenbanksysteme II

Lerninhalte 232-05 Modellierung der RC Wildbach Datenbank (1)

Für unsere Rafting-Datenbank haben wir kein Pflichtenheft, sondern nur eine knappe Aufgabenbeschreibung. Um herauszufinden, was mindestens verlangt wird, sollte man sich die gestellte Aufgabe zunächst genau durchlesen.

Suche nach Entitätstypen

Ein praktischer Weg Entitätstypen zu finden besteht darin, die Substantive der Aufgabenbeschreibung unter die Lupe zu nehmen. In dieser Version sind alle relevanten Hauptwörter **fett** markiert:

Mit der Datenbank für den RC Wildbach sollen die **Kunden**, die **Boote**, die **Guides** und alle **Buchungen** verwaltet werden. Aus organisatorischen Gründen gibt es festgelegte **Startzeiten** für jede Tour.

Der Entitätstyp **Kunde**

Unsere Kundenverwaltung haben wir bereits im Modul **2.3.1 Datenbanksysteme I** in der Datenbank für den RC-Wildbach (s. Ordner Materialien - Datenbanken: *RC-Wildbach-06.odb*) schrittweise implementiert. Die Entität **Kunde**, sowie die dazugehörigen Attribute¹ stellt sich im ER-Modell so dar:

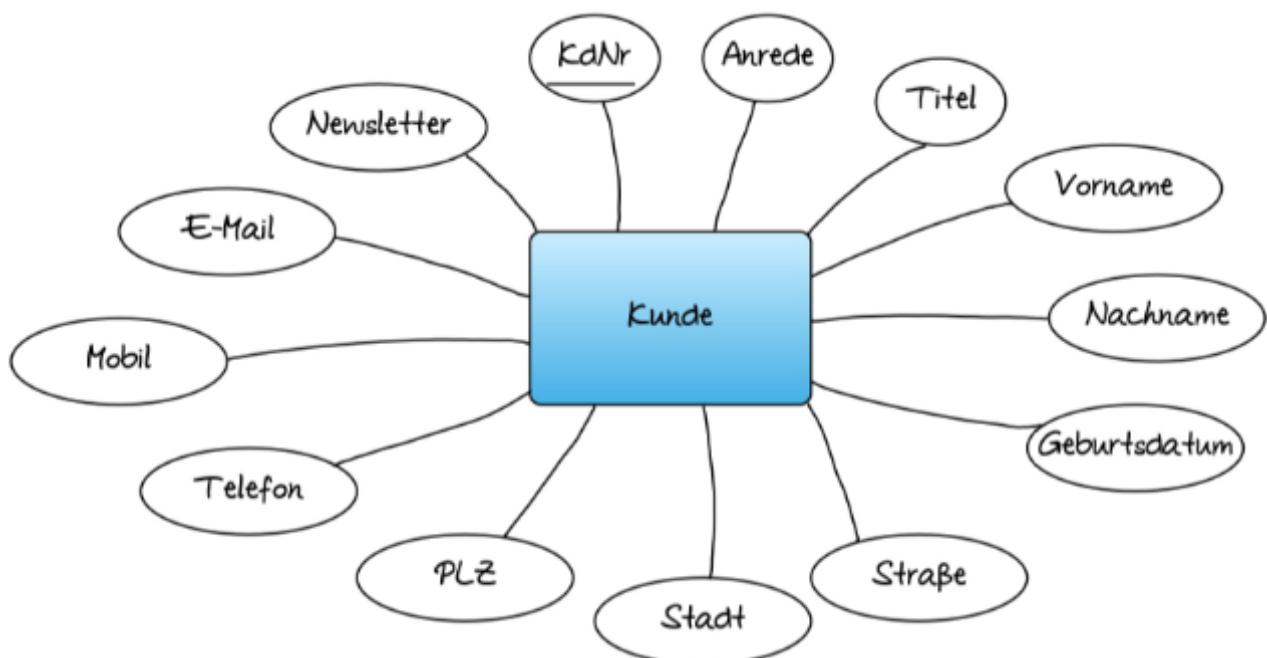


Abb. 03: Entität Kunde mit Attributen

¹ Der farbige Hintergrund für die Entität Kunde soll die Lesbarkeit des Diagramms erleichtern. Die verwendete Schrift unterstreicht den Eindruck, dass die Modellierung mit Papier und Bleistift gemacht werden kann.



Der Entitätstyp Boot

Es sollen Touren mit Raftingbooten gebucht werden können. Der Geschäftsführer, Max Huber, möchte den Preis nicht pro Person, sondern pro Boot kalkulieren. Wir benötigen also das Attribut **PreisProBoot** und erfassen zusätzlich weitere, sinnvolle Eigenschaften, wie den Bootsnamen (**Bootsname**) und die Anzahl der verfügbaren Sitzplätze (**Sitze**). Das neue, künstliche Attribut **BootNr** hilft uns maschinell alle Boote eindeutig zu identifizieren. So ein **identifizierendes Attribut** nennt man auch **Schlüsselfeld**. Weil es so wichtig ist, wird dieses Attribut im ER-Modell auch unterstrichen.

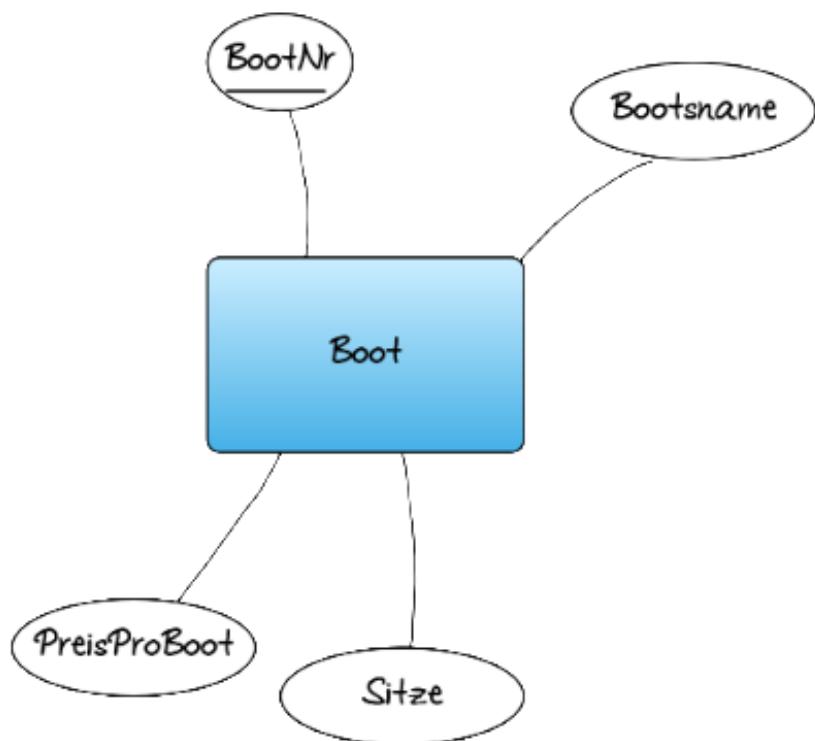


Abb. 04: Entität Boot mit Attributen

Es ergibt sich folgendes Klassendiagramm, welches wir in LibreOffice Base mit den jeweils passenden Datentypen implementieren. Es entsteht die **Relation** Boot(BootNr, Bootsname, Sitze, PreisProBoot).

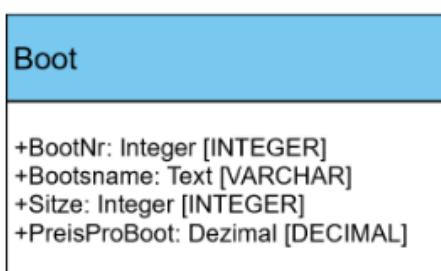


Abb. 05: Klassendiagramm mit Attributen und Datentypen

	Feldname	Feldtyp
▶	BootNr	Integer [INTEGER]
	Bootsname	Text [VARCHAR]
	Sitze	Integer [INTEGER]
	PreisProBoot	Dezimal [DECIMAL]

Abb. 06: Tabellenobjekt *tblBoote* in LibreOffice Base



Der Entitätstyp Guide

Kunden bekommen ein Boot nur zusammen mit einem ausgebildeten Guide. Der Guide ist im Bootspreis inbegriffen. Natürlich könnten wir – wie bei den Kunden – gleich viele Datenfelder (Attribute) erfassen. Für unsere „Miniwelt“ sollen die minimalen Kontaktdaten ausreichen.

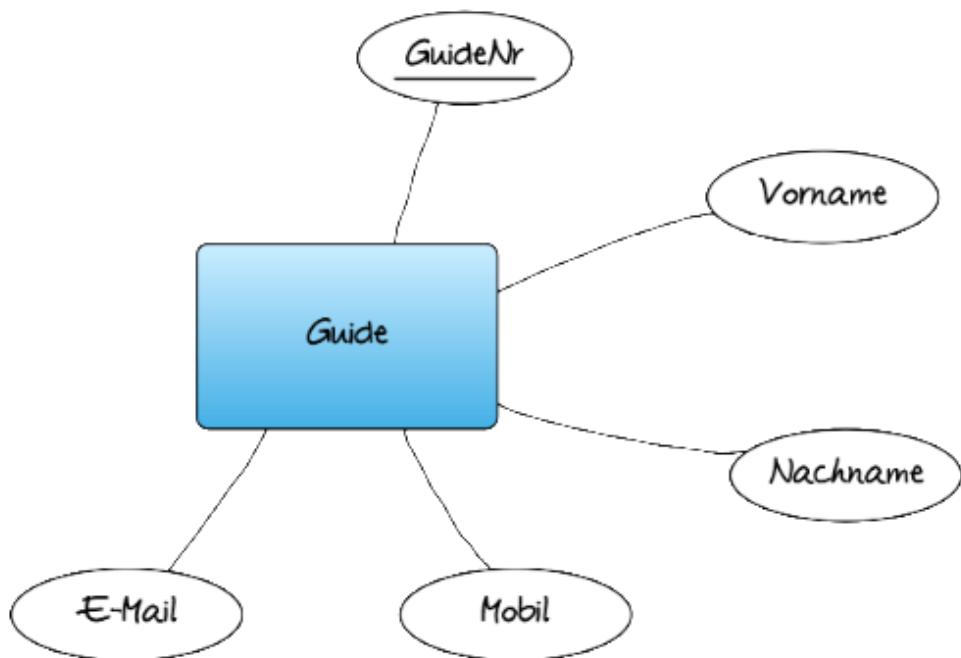


Abb. 07: Entität Guide mit Attributen

Es ergibt sich folgendes Klassendiagramm, welches wir in LibreOffice Base mit den jeweils passenden Datentypen implementieren. Es entsteht die **Relation** Guide(GuideNr, Nachname, Vorname, Mobil, E-Mail).

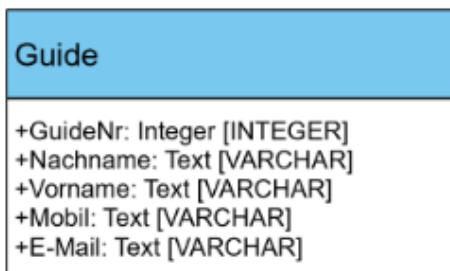


Abb. 08: Klassendiagramm mit Attributen und Datentypen

	Feldname	Feldtyp
►	GuideNr	Integer [INTEGER]
	Nachname	Text [VARCHAR]
	Vorname	Text [VARCHAR]
	Mobil	Text [VARCHAR]
	E-Mail	Text [VARCHAR]

Abb. 09: Tabellenobjekt *tblGuides* in LibreOffice Base



Der Entitätstyp Startzeit

Aus organisatorischen Gründen gibt es festgelegte Startzeiten für jede Tour. Damit sie später problemlos angepasst werden können, erfassen wir ihre Eigenschaften in einer eigenen Entität.

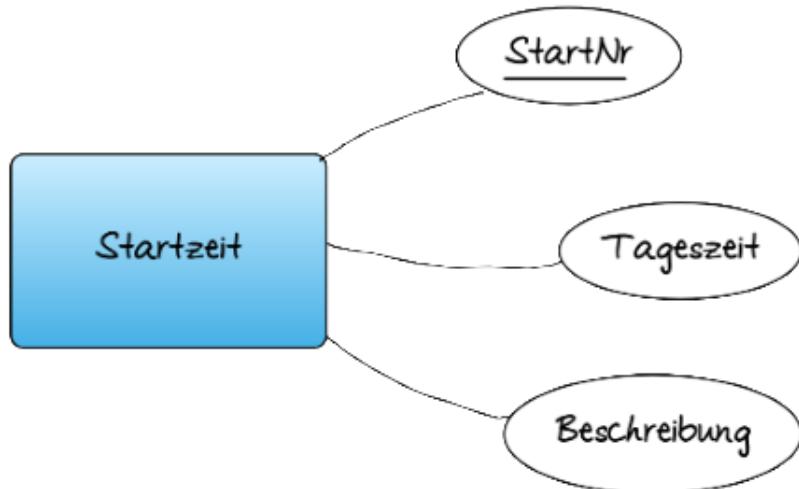


Abb. 10: Entität Startzeit mit Attributen

Es ergibt sich folgendes Klassendiagramm, welches wir in LibreOffice Base mit den jeweils passenden Datentypen implementieren. Es entsteht die **Relation** Startzeit(StartNr, Beschreibung, Tageszeit).

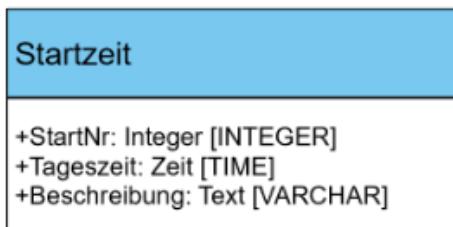


Abb. 11: Klassendiagramm mit Attribut- und Datentypen

	Feldname	Feldtyp
▶	StartNr	Integer [INTEGER]
	Beschreibung	Text [VARCHAR]
	Tageszeit	Zeit [TIME]

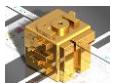
Abb. 12: Tabellenobjekt *tblStartzeiten* in LibreOffice Base

Damit du dir besser vorstellen kannst, welche konkreten Datensätze aktuell in der Relation Startzeit erfasst sind, wirf einen Blick auf die unteren Objektkarten.



Abb. 13: Objektkarten zur Klasse Startzeit

Im nächsten Kapitel - 232-06 Modellierung der RC Wildbach Datenbank (2) – machen wir uns Gedanken über die Beziehungen zwischen den gefundenen Entitätstypen und den dazu gehörenden Kardinalitäten.



Wie die bisherigen Tabellenobjekte unseres Datenmodells in der Datenbank RC-Wildbach implementiert werden, kannst du mit Hilfe dieses **Lernvideos** nachvollziehen:



Abb. 14: Startbild des Videotutoriums *Implementierung Teil 1*