



Rekursive Beziehungen tauchen nur einmal auf und zwar bei Autor empfiehlt einen anderen Autor.

Der Pfeil zwischen Benutzer und Autor ist so zu lesen: "Der Autor IST ein Benutzer"

Ich habe die Bewertung als eine Beziehung zwischen Benutzer und Autor gemacht, da ich es nicht als nötig gesehen habe eine weitere Entität für Bewertung zu erstellen, weil die nötigen Attribute einer Bewertung auch als Attribute in der Beziehung sein können. Unter anderem war eine kritische Entscheidung die Kardinalitäten zwischen Benutzer bewertet Autor, da im Text zwar steht ein Benutzer kann zu jedem Autor genau eine Bewertung abgegeben, aber in meinem ER-Modell ist es so zu lesen, dass ein Autor von beliebig vielen Benutzern bewertet werden kann, deshalb $[0, *]$ und ebenso kann ein Benutzer beliebig viele Autoren 1 mal bewerten, daher wieder $[0, *]$.

Wenn ein Autor eine Seite anlegt, dann mit dem Datum als Information, daher kommt hier wieder Attribut bei einer Beziehung ins Spiel. Ein Autor kann eine Seite anlegen, muss aber nicht unbedingt daher $[0, 1]$.

Jede Seite enthält einen Eintrag mind. und ansonsten beliebig viele, daher $[1, *]$

Ein Eintrag kann beliebig viele Bilder enthalten, daher $[0, *]$ und jedes Bild ist in genau einem Eintrag enthalten, daher $[1, 1]$.

Ebenfalls wird ein Eintrag immer von genau einem Autor geschrieben, daher $[1, 1]$, aber ein Autor kann beliebig viele Einträge schreiben, daher $[0, *]$.

Dies gilt ebenso für die Bild und Tag.

An dieser Stelle könnte vielleicht eine mehrstellige Beziehung auftauchen, da sich "erstellt", "erstellt" und "schreibt" ähneln. Da man aber in der Praxis keine mehrstelligen Beziehungen benutzt lasse ich hier jede einzeln.

Das Attribut bei Bild namens Bild ist bei der Implementierung ein Blob. Ich erwähne es nur um den doppelten Namen von "Bild" einmal erwähnt zu haben und um zwischen Entität Bild und Attribut Bild unterscheiden zu können.

Tag hat als Primärschlüssel einen Tag, da jeder Tag eindeutig ist und sonst keine weiteren.

Außerdem kann ein Tag bei beliebig vielen Bildern gesetzt werden und ein Bild kann genauso beliebig viele Tags setzen, daher bei beiden $[0, *]$.

Ein Autor kann beliebig viele Transaktionen empfangen, daher $[0, *]$.

Ebenso kann ein Benutzer beliebig viele Transaktionen ausführen, daher auch $[0, *]$