Einführung

In dieser Dokumentation beschreiben wir unsere Lösungen zu dem vollständigen Lastenheft für das Beobachtungs-Informationssystem Merlin. Merlin soll dabei den Austausch für die Beobachtung von Vögeln *(also birdwatching)* ermöglichen. Diese Dokumentation und die Aufgabe wurde von Anton Kirakozov (Matrikelnummer: ) und Mesut Koc (Matrikelnummer: ) bearbeitet.

Erforderliche software

Für die Umsetzung unseres ER-Diagramms verwenden wir die Open-Source-Software „Dia“. Mit Hilfe der Software können wir unsere Gedankengänge besser darstellen, da die Software die Möglichkeit bietet schematische Zeichnungen und Diagramme zu erstellen.

**Quelle:** <http://www.heise.de/download/dia.html>

Allgemeine Fragen und annahmen

Laut der Aufgabenstellung haben sich vorerst folgende Fragen ergeben:

* Das Lastenheft gibt Information über die Unterart *(3. Glossar)*. Wir haben uns dabei gefragt, ob die „Unterart“ von Vögeln ein eigener Entitätstyp sein soll.

**Annahme:** Wir sind nun davon ausgegangen, dass nicht jede Art eine Unterart haben kann. Aus diesem Grund haben wir, „Unterart“ als Attribut bei dem Entitätstyp “ Vogelart“ aufgenommen. Dabei Unterteilt sich die Unterart in zwei wesentliche Attribute und zwar in Art und Gattung.

* Da man zwischen den Benutzern in drei Kategorien unterscheiden muss in DB-Administrator,

Content-Admin und Birdwatcher, haben wir uns entschieden eine eigene Entität für die Rolle zu erstellen, damit die referenzielle Datenintegrität gewährleistet wird.

* Bei unseren ersten Überlegungen, entschieden wir uns die Checkliste als eine Entität zu entwickeln. Allerdings stellten wir fest, dass die Relationen, die zwischen Checkliste –Vogelarten und Checkliste- Beobachtungsort, in einer Art eine mehrfach zu mehrfach Beziehung bilden, sodass man die Checkliste auch zu einer Relation umwandeln kann.

Das ER-Diagramm

Erklärungen zum ER-Diagramm

Für die Entwicklung eines ER-Diagramms, ist es wichtig wesentliche Informationen aus einem Text/Lastenheft herauszufiltern und diese regelkonform umzusetzen. Man entnimmt relevante Informationen und versucht zu analysieren in welchem Kontext sie zu einander stehen.

Unsere Herangehensweise war recht strukturiert aufgebaut und wurde schritt für schritt durchgeführt. Am Anfang haben wir versucht festzustellen welche Entitäten sich jeweils ergeben und in welcher Beziehung sie zu einander stehen. Es ergeben sich laut Aufgabenstellung vier Entitätenstypen und zwar:

**Beobachtungsort**, **Vorgelart**, **Benutzer** und **Rolle**.

Im Verlauf dieser Dokumentation werden wir auf einzelne Entitäten, Entitätstypen, Relationen und Attribute eingehen und diese erklären.

**Beobachtungsort und Vogelart**

Zu Beginn fangen wir mit dem Beobachtungsort an. Der Ort der Beobachtung ist wichtig um feststellen zu können in welcher Region eine Vogelart gesichtet wurde oder auch nicht. Dabei ist die Region in drei Kategorien aufgeteilt in Kontinent, Land und Stadt. Diese drei Kategorien bilden die Attribute des Entitätstypen. Zusätzlich bilden wir einen Primärschlüssel, damit wir jede Entität eindeutig identifizieren können.

Es entsteht der Entitätstyp Vogelart , da man genauso wie bei dem Beobachtungsort die Daten eines Vogels dokumentieren muss. Es entstehen also auch hier Attribute wie Name, Art und Gattung. Die letzteren beiden sind für die Unterscheidung der Vögel essenziell wichtig, da es von einer Art mehrere Gattungen geben kann. Der Entitätstyp wird auch mit einem Primarykey versehen(siehe Beobachtung).

Die oben genannten Entitätstypen stehen in einer Beziehung zu einander. Diese Beziehung bezeichnet sich als eine mehrfach zu mehrfach Beziehung. Diese Annahme kann man mit einfachen Fragen feststellen. Es können mehrere Vogelarten in einem Ort gesichtet werden. Also ergibt sich eine n:1 Beziehung. Da wir aber auch die andere Seite betrachten müssen, stellen wir fest, dass es einen Vogel auch an mehreren Orten geben kann, was wiederum auch zu einer 1:n Beziehung führt. Diese beiden Relationen kann man zu einer Relation zusammenfügen, sodass es zu der oben genannten mehrfach zu mehrfach Beziehung kommt(m:n). Aus dieser m:n-Beziehung entsteht eine „Zwischentabelle“ welche den Zusammengesetzten Schüssel beinhaltet und als Primärschlüssel fungiert(BeobachtungsID,VID). Auf diese Tabelle möchten wir nicht genauer eingehen, da diese in erst in einem ERM von relevanter Bedeutung ist.

**Benutzer und Rolle**

Betrachten wir nun den Entitätstypen Benutzer. Da man davon ausgehen **kann**, dass es mehrere Benutzer/Beobachter geben kann, müssen die Informationen wie Name, Adresse dokumentiert werden. Dieser Entitätstyp ist besonders , da er gleich mehrere Relationen besitzt. Neben dem Namen der Adresse usw. besitzt jeder Benutzer eine besondere Funktionalität(Rolle). Diese Rolle definiert sich aus einem Administrator, Content-Admin und einem Birdwatcher. Wie man unserer Annahme entnehmen kann, haben wir uns für einen vierten Entitätstypen entschieden. Diese Annahme belegen wir mit folgendem Gedanken. Mit Hilfe der Frage, ob ein Benutzer mehrere Tätigkeiten haben haben, bildet sich eine 1:n Relation, sodass man den Entitästypen Rolle erstellen kann. Des Weiteren wäre die Datenbank inkonsistent und es wäre umständlicher, falls man bestimme Datensätze löschen sollte und somit jedes einzelne Tupel ändern müsste.

Die beiden Entitätstypen versehen wir mit einem PrimaryKey(s.o.), des Weiteren beinhaltet Benutzer mehrere Attribute wie z.B. einen vollständigen Namen, Registrierungsdatum , Email, Passwort, möglicherweise auch die Adresse usw. Wohingegen Rolle nur ein Attribut beinhaltet und zwar die Funktionalität des Users.

**Beobachtungsort, Vogelart und Benutzer**