



DATENBANKEN

Dokumentation zum dritten Praktikum

PRAKTIKUM 3

In diesem Dokument sind die Änderungen beschrieben die an der Struktur der Datenbank vorgenommen wurden, sowie die Beschreibung der Lösung für die Aufgaben des dritten Praktikums

Steffen Giersch & Maria Lüdemann

HAW Hamburg 18.06.2013

INHALTSVERZEICHNIS

ÄNDERUNGEN DER STRUKTUR	2
AUFGABE 3	2
SCRIPT 9.1	2
SCRIPT 9.2	2
SCRIPT 9.3	3
SICHERHEITSABFRAGEN FÜR EINGABEN	3
CHECKLISTE DER ABNAHME.....	3

Änderungen der Struktur

Aus der Besprechung der letzten Aufgabe ergaben sich noch einige kleine Änderungen die vorgenommen werden mussten. Sie sind im Folgenden beschrieben.

In der Tabelle „Beobachtungsort“ ist nun die „Zoographische Region“ „not null“ anstelle vom „Beobachtungsort“. Dies kommt daher, dass das Lastenheft ungenau beschrieben hat, welche Angabe zwangsweise gegeben werden muss. Daraus folgt, dass die „Range“ aus der Tabelle „Merlin.Birds“ nicht mehr als Beobachtungsort eingesetzt werden kann.

Stattdessen wird nun für alle Vögel ein Ort „Paläarktis, De“ definiert, der in der Checkliste zugewiesen wird.

Des Weiteren wird die Checkliste nun nur noch mit den Vögeln aus der Tabelle „Merlin.Birds_de“ gefüllt, die alle den Ort „Paläarktis, De“ zugewiesen bekommen.

Zuletzt wurde noch die Priorität der Tabelle Merlin.Birds_DE über die der Tabelle Merlin.Birds_IOC gesetzt, weil wir sowieso nur die Einträge aus Birds_DE abfragen.

Aufgabe 3

Im Folgenden sind die Scripte aufgeführt und erklärt die wir im Praktikum vorführten.

Script 9.1

Welche Vögel der Deutschen Liste wurden schon beobachtet.

```
/* Aufgabe 9 a*/
select distinct * from voegel v
  WHERE EXISTS (select * from beobachtet b where v.vogelid = b.vogelid
and b.benutzername = 'Maria Luedemann');
```

Script 9.2

Welche Vogelarten noch nicht?

```
/* Aufgabe 9 b*/
select * from voegel v
  WHERE NOT EXISTS (select * from beobachtet b where v.vogelid =
  b.vogelid and b.benutzername = 'Maria Luedemann')
AND EXISTS (select * from checkliste ch where v.vogelid = ch.vogelid);
```

Script 9.3

Wann wurden Lifer vermerkt?

```
SELECT BenutzerName, VogelID, min(BeobBeginn) FROM beobachtet b
Group BY b.BenutzerName, b.VogelID;
```

Sicherheitsabfragen für Eingaben

Dieses Insert stellt sicher, dass sowohl der Vogel wie auch der Ort bereits vorher in der Datenbank vorhanden sind. Nur dann gelingt das Erstellen der neuen Beobachtung.

```
INSERT INTO Beobachtet
SELECT 12, 'Maria Luedemann', 1, 256, '01-01-2005', '01-01-2005', 'Flying and
stuff' FROM dual
WHERE EXISTS (SELECT BeobachtungsortID FROM Beobachtungsort WHERE
BeobachtungsortID = 1)
AND EXISTS (SELECT VogelID FROM Checkliste ch WHERE ch.VogelID =
256) ;
```

Checkliste der Abnahme

Anmeldung als Beobachter

Die Datenbank beinhaltet drei Rechtstufen die in der Tabelle Beobachter festgelegt werden. 1 steht für Admin, 2 für Moderator und 3 für Benutzer.

Anlegen von Beobachtungen

Beobachtungen sollen über eine Maske eingegeben werden können die Sorge trägt dass die Daten legal sind. So wird die ID automatisch von der Oberfläche erzeugt, der Benutzername ausgelesen und weitergereicht. Die VogelID wird durch den Benutzer ausgewählt, die Oberfläche bietet dem Benutzer an aus dem Englischen, Lateinischen oder Deutschen Namen zu wählen sofern sie vorhanden sind und reicht dann aber nur die ID an die Datenbank weiter. Sowohl Beginn wie auch Ende wird über eine Oberfläche eingegeben die das Datum als Drop Down Menü darstellt sodass der Nutzer gezwungen ist das gewünschte Format anzugeben. Ein Kommentar ist optional und wird unbearbeitet in der Datenbank abgespeichert.

Der Beobachtungsort

Es ist durch die Abfrage festgelegt, dass der Ort vorher schon in der Datenbank vorhanden sein muss damit er angegeben werden kann. Erstellen kann nur der Content-Admin oder der DB-Admin einen Ort, der Benutzer kann ihn dann in der Oberfläche nur noch auswählen.

```
INSERT INTO Beobachtet
/* (BeobachtungsID, BenutzerName, BeobachtungsortID, VogelID, BeobBeginn, BeobEnde, Kommentar) */
SELECT 10, 'Steffen Giersch', 1, 256, '01-01-2000', '01-01-2000', 'Dance'
FROM dual
WHERE EXISTS (SELECT BeobachtungsortID FROM Beobachtungsort WHERE BeobachtungsortID = 1)
AND EXISTS (SELECT VogelID FROM Checkliste ch WHERE ch.VogelID = 256);
```

Was war ein Lifer

Lifer werden durch Abfragen in Form von 9.3 herausgesucht die im Falle einer persönlichen Abfrage des Birdwatchers auf seine ID angepasst werden. Dort werden dann die Lifer angezeigt welcher Vogel, wann und wo von diesem Benutzer das erste Mal gesehen wurden.