



Betreuer: Frau Prof. Dr. I. Wilfert

Studienarbeit

zum Thema

„Überführung analoger Radwegeinformationen des
Kreisinformationssystems für den Landkreis Leipziger Land in ein
GIS (ArcView) mit gleichzeitigem Aufbau einer Oracle-Datenbank“



Olaf Schnabel
Studiengang 1996
Dresden, 17. Juni 2001

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alles aus anderen Quellen oder von anderen Personen übernommene Material, das in dieser Arbeit verwendet wurde oder auf das direkt Bezug genommen wird, wurde als solches kenntlich gemacht.

Leipzig, 17.06.2001

Olaf Schnabel

Themenstellung der Studienarbeit

für: Herrn Olaf Schnabel

Thema: Überführung analoger Radwegeinformationen des Kreisinformationssystems für den Landkreis Leipziger Land in ein GIS (ArcView) mit gleichzeitigem Aufbau einer Oracle-Datenbank

Anliegen dieser Studienarbeit ist es, aus vorhandenen Unterlagen eine GIS-basierte digitale Radwegkarte zu erstellen und mit weiteren Planungsdaten in einer Datenbank zu hinterlegen. Ziel ist es, diese Radwegkarte multifunktional für die Bauplanung, den Tourismus sowie die Umweltplanung zu nutzen.

Prof. Dr. Ingeborg Wilfert

Inhaltsverzeichnis

	Seite
1. Einleitung	6
2. Vorarbeiten	8
2.1. Das GIS ArcView	8
2.2. Das Datenbankmanagementsystem Oracle	11
2.3. Das Prinzip der relationalen Datenbanken	14
2.4. Erstellung des logischen Konzepts	19
3. Überführung der analogen Daten in ein digitales Format	23
3.1. Vorüberlegungen zur Digitalisierung	23
3.2. Segmentierung	25
4. Aufbau der relationalen Datenbank	27
4.1. Erstellung des physischen Konzepts	27
4.2. Erstellung der Datenbank	32
4.3. Anbindung der Datenbank an die Geometrie im GIS	39
5. Möglichkeiten und Grenzen der Aktualisierung	43
6. Planungsrelevante Zusatzinformationen	49
7. Grafische Gestaltung der Radwegekarte	52
8. Zusammenfassung	56
9. Danksagung	58
10. Literatur- und Quellenverzeichnis	59
11. Abbildungsverzeichnis	61

Anlagen

- I. Segmentierscript
- II. Segmentierscript zur Aktualisierung
- III. Koordinatenscript
- IV. Controlfile
- V. Anleitung zur Aktualisierung

1. Einleitung

Das Landratsamt Leipziger Land ist die regionale Verwaltungseinheit für die 24 Gemeinden des Landkreises Leipziger Land. Das Bauplanungs- und Wirtschaftsförderungsamt ist neben dem Amt für Kommunalaufsicht, dem Amt für Umweltschutz sowie dem Bauverwaltungsamt dem Dezernat IV des Landratsamtes Leipziger Land zugeordnet.

1999 wurde der Beschluss gefasst, ein Geographisches Informationssystem (GIS) für das Bauplanungs- und Wirtschaftsförderungsamt zu erwerben, welches seit dieser Zeit mit dem Desktop-GIS ArcView im Landratsamt vorliegt [HEINRICH, 2000, S. 1].

Ziel dieser Studienarbeit ist es, ein Radwegeinformationssystem für das Gebiet des Landkreises Leipziger Land für das Bauplanungs- und Wirtschaftsförderungsamt des Landratsamtes Leipziger Land zu kreieren, um eine Koordinierungsgrundlage für die Radwegenetzplanung der Kommunen sowie der anderen Träger öffentlicher Belange zu schaffen. Dabei soll eine relationale Datenbank mit Radwegemerkmale und -informationen erstellt werden und diese mit der Geometrie im GIS verbunden werden.

Bisher lag im Leipziger Landratsamt nur eine analoge Folie (zwei Blätter) mit Radwegen im Maßstab 1:30.000 vor, von der eine generalisierte Papierkopie im Maßstab 1:40.000 erstellt wurde. Dort sind farbige Themenelemente sowie teilweise auch farbige Hintergrundelemente zu großen Teilen per Hand mit Farbstiften eingezeichnet, so dass die Darstellung der Radwege erheblich in ihrer Lesbarkeit eingeschränkt ist. Außerdem hat diese Karte durch den kleinen Maßstab nur Übersichtscharakter. Daher ist deren Zweck, vorhandene Problempunkte des Radwegenetzes aufzuzeigen und ein Entscheidungsinstrument für Lösungsmöglichkeiten dieser Probleme zu sein, nicht erfüllt worden.

Damit stellte sich für das Bauplanungs- und Wirtschaftsförderungsamt des Landratsamtes Leipziger Land die Aufgabe, eine zweckmäßigere Grundlage für Problemlösungen und die Planung von Radwegen neben der analogen großformatigen Folie zu schaffen, die ohne großen Personal- und Zeitaufwand

zu aktualisieren und zu existierenden Daten anderer Ämter des Landratsamtes sowie höherer Planungsorgane (Regierungspräsidium Leipzig) und anderen staatlichen Einrichtungen (zum Beispiel: Straßenbauamt) kompatibel ist. Dabei soll sie erweiterbar sein, das heißt, sich mit anderen Themen wie Sehenswürdigkeiten, Biotopen und Baugebieten verschneiden lassen, um so neue Erkenntnisse zu finden. Für diese Aufgaben ist ein GIS sehr gut geeignet und auch mit dem ArcView 3.2 im Landratsamt vorhanden. Weitere Arbeitsunterlagen für dieses Projekt wurden vom Landratsamt als Fotografien von den Radwegen, Planungsunterlagen in Textform und Gutachten von örtlichen Radwandervereinen bereitgestellt. Des Weiteren ist im Leipziger Landratsamt ein Datenbankserver im lokalen Netzwerk (local area network - LAN) vorhanden, von dem aus die Software für alle Rechner des Landratsamtes gestartet wird. Für den GIS-Arbeitsplatz des Autors handelte es sich um die oben erwähnte bereitgestellte GIS-Software ArcView 3.2 von der Firma Environmental Systems Research Institute (ESRI) sowie die Datenbanksoftware Oracle 7.3.4 . Da der Datenbankserver im LAN mit dem Betriebssystem UNIX läuft, ist dieser als stabil genug für den Mehrbenutzerbetrieb zu werten. Die Speicherung und Haltung der Sachdaten ist daher unter dem vom Server bereitgestellten Oracle-Datenbankmanagementsystem als günstig einzuschätzen. Die Geometrie dagegen sollte im ArcView bearbeitet und die thematischen Sachdaten mit dieser Geometrie verknüpft werden.

2. Vorarbeiten

2.1. Das GIS ArcView

In diesem Kapitel wird eine kurze grundlegende Beschreibung von ArcView gegeben. Im Bauplanungs- und Wirtschaftsförderungsamt vollzieht sich die Arbeit meist in Projekten. Dafür ist ArcView sehr gut geeignet, da es alle Bestandteile einer Projektarbeit (Views, Tabellen, Diagramme, Layouts, Scripte) in einer Projektdatei speichert, die jedoch nur die Verweise auf die Daten (Verzeichnispfade) enthält. Dadurch können dieselben Daten ohne Duplizierung in beliebig vielen Projekten benutzt werden. Somit kann einerseits die Aktualität dieser Daten gewährleistet werden, andererseits können mehrere Benutzer mit den gleichen Daten arbeiten.

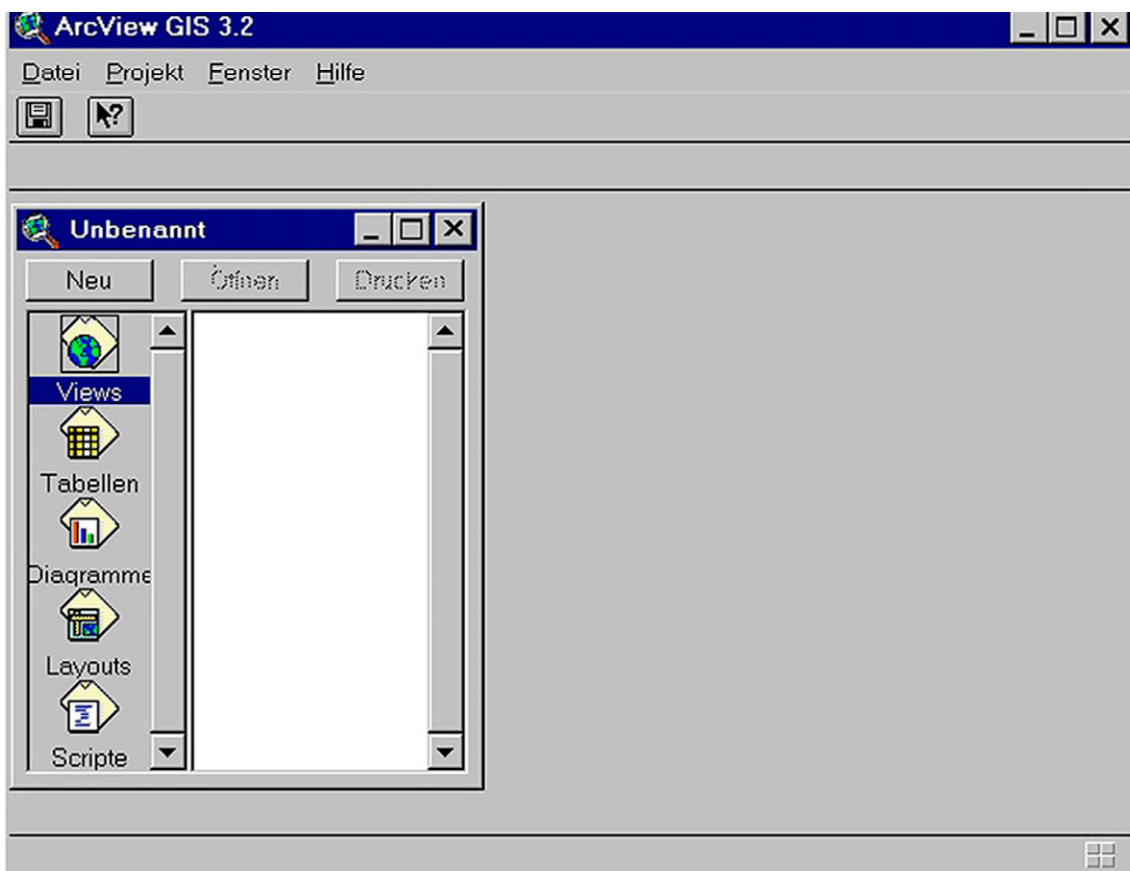


Abbildung 2: Das ArcView-Projektfenster

Der Menüpunkt „Views“ (Abbildung 2) stellt eine Art Bildschirmkarte dar, in der beliebig viele thematische Ebenen (Themen) übereinander projiziert und

dargestellt werden können. Dabei besteht jedes Thema aus einer Menge gleichartiger Geoobjekte (Punkt, Linie, Fläche, Pixel), die raumbezogen sind und als Vektor- oder Rasterdaten dargestellt werden. Die Geometrien werden dabei getrennt nach ihrer graphischen Grundstruktur gespeichert [HEINRICH, 200, S. 4].

Unter dem Menüpunkt „Tabellen“ (Abbildung 2) kann man sich die Attribute der graphischen Daten in einer Sachdatentabelle zusammengefasst anzeigen lassen und diese auch editieren. Aber auch eigene Tabellen kann man dort erstellen.

Im Menüpunkt „Diagramme“ (Abbildung 2) können Diagramme dargestellt werden, die Daten aus den Tabellen benutzen.

Der Menüpunkt „Layouts“ (Abbildung 2) bietet die Möglichkeit, thematische Daten aus den Views, Tabellen, Diagrammen, importierten Rastergrafiken sowie Kartengestaltungselementen wie Nordpfeil, Legende und Kartenrahmen für den Druck zusammenzustellen.

Der Menüpunkt „Scripte“ (Abbildung 2) wird benutzt, um Programme (Scripte) und Anwendungen (Extensions) in der ArcView-internen objektorientierten Programmiersprache Avenue zu schreiben und auszuführen. Dabei ist wichtig, dass ArcView selbst modular aus Avenue-Scripten zusammengesetzt ist. Das heißt, dass jeder Button und jeder Menüpunkt im ArcView selbst programmierbar ist. Damit ist ArcView individuell erweiterbar.

Gravierende Nachteile von ArcView sind das fehlende Datenbankmanagement sowie die nichttopologische Speicherung der Geometriedaten, das heißt, dass Redundanzen bei identischen Polygongrenzen auftreten. Da jedoch in dieser Arbeit Radwege (Linien) und Sehenswürdigkeiten (Punkte) dargestellt werden sollen, hat dieser Nachteil hierfür keine Relevanz. Räumliche Analysen können ohne weiteres durchgeführt werden und sind dank modularem Aufbau auch beliebig erweiterbar.

Denkbar war es für diese Studienarbeit, als Arbeitsgrundlage Vektor-Daten aus der Automatisierten Liegenschaftskarte (ALK) zu übernehmen. Da aber die ALK laut Mitarbeitern des Landratsamtes Leipziger Land für den Landkreis Leipziger Land nicht flächendeckend vorlag und immer noch nicht vorliegt, wurden georeferenzierte topographische Karten im Maßstab 1:10.000 benutzt, die für dieses Projekt flächendeckend, randangepasst und blattschnittfrei vorlagen

[FEHSE, mündliche Mitteilung, 2000]. Auf dieser großmaßstäbigen Kartenbasis konnten nun Daten erfasst werden.

2.2. Das Datenbankmanagementsystem Oracle

Das Datenbankmanagementsystem Oracle wird für diese Arbeit benutzt, weil die Informationen (Merkmale) der Radwege unabhängig von der Geometrie gespeichert werden sollen. Auf diese Weise können sie für andere Anwendungen durchaus auch von Belang sein und ohne Probleme dafür verwendet werden.

Oracle besteht in der Standardkonfiguration aus sieben Komponenten. Das Utility „WebDB“ erstellt Internetseiten als Programme, die auf Anforderung Daten aus der Datenbank ins Internet stellen.

Der „Enterprise-Manager“ ist ein Datenbankmanagementsystem (DBMS), welches über Fenster und Menüs viele Datenbank-Verwaltungsaufgaben wie die Verwaltung von Daten und Tabellen erledigen kann, so dass der Nutzer keinen Programmiercode eingeben muss. Außerdem kann man mit dem Enterprise-Manager Benutzer einrichten und Passwörter vergeben. Er ist das wichtigste Werkzeug des Datenbank-Administrators, da damit eindeutige Rollen für die Benutzer zugewiesen werden, das heißt, dass durch den Datenbank-Administrator bestimmt wird, welche Privilegien (Lese- und Schreibrechte) der Benutzer erhält.

Ein weiteres Utility ist „SQL*Plus“. Damit können Abfragen erstellt und ausgeführt, Tabellen hinzugefügt und geändert sowie Daten editiert werden. Es wird die Programmiersprache SQL (Structured Query Language) benutzt, die in fast jedem relationalen Datenbankmanagementsystem Standard ist [McCULLOUGH-DIETER, 1999, S. 26]. SQL ist eine sogenannte nichtprozedurale Programmiersprache. Muss in einer prozeduralen Programmiersprache der Zugriff auf die Daten sowie die Bearbeitung derselben beschrieben werden, ermittelt Oracle im Gegensatz dazu intern die nötigen Befehle [LOCKMAN, 1998, S. 33]. Zusätzlich wurde SQL in Oracle noch um weitere Befehle erweitert (Einrichtung von Benutzern, Variablendefinition, Dateieditierungen usw.).

Das vierte Utility „EXP und IMP“ kann benutzt werden, um Daten aus einer Oracle-Datenbank in eine Datei zu exportieren und diese in einer Oracle-Datenbank (zum Beispiel in einem anderen Betriebssystem) wieder zu importieren.

Das Utility „Präcompiler“ bietet für verschiedene Programmiersprachen wie C, C++, Pascal und FORTRAN einige Präcompiler, um diese Programmiersprachen innerhalb einer SQL-Anweisung einzusetzen.

Das sechste Utility „Assistenten“ besteht aus Hilfsprogrammen, die bei komplizierteren Aufgaben Hilfestellungen bieten.

Das letzte Utility „Net8“ ist ein Kommunikationswerkzeug, mit dem Computer für Oracle-Datenbanken in einem Netzwerk konfiguriert werden können [McCULLOUGH-DIETER, 1999, S. 27].

Während dem Datenbankadministrator im Landratsamt Leipziger Land alle oben beschriebenen Utilities zur Verfügung stehen, sind die Computer für Nutzer von Oracle-Datenbanken nur mit dem Utility „SQL*Plus“ sowie dem Konfigurationstool „SQL Net Easy Configuration“ (Einrichtung der Datenbank auf dem Server) ausgestattet.

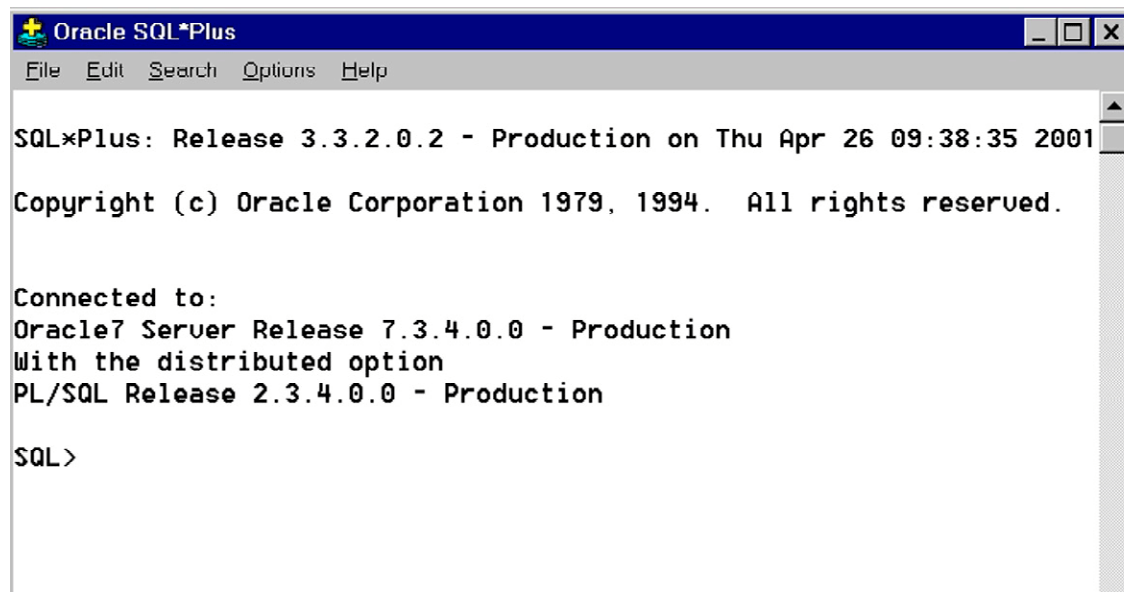


Abbildung 3: Das Oracle SQL*Plus - Eingabefenster

Dem Autor stand daher nur ein Befehlseingabefenster für SQL-Anweisungen zur Verfügung (Abbildung 3), so dass eine Zugriffsrechteverwaltung und Rollenprofilfestlegung sowie eine Vereinfachung im Aktualisierungs- und Editierungsablauf durch Programmierung im Oracle durch den Autor nicht vorgenommen werden konnten (keine Administratorrechte). Daher wurde nach Alternativen gesucht, um optimale Voraussetzungen für den Aufbau und vor allem für die später durch ungeübte Nutzer auszuführende Aktualisierung der

Datenbank zu haben. Eine Alternative bietet das Datenbankmanagementsystem Access der Firma Microsoft, welches zwar im Leipziger Landratsamt vorhanden ist, allerdings für diese Arbeit aus lizenzrechtlichen Gründen nicht zur Verfügung stand. Somit musste auf das Utility „SQL*Plus“ zurückgegriffen werden.

2.3. Das Prinzip der relationalen Datenbanken

Das vom Server bereitgestellte Datenbankmanagementsystem Oracle 7.3.4 ist ein relationales Datenbankmanagementsystem, also eine Sammlung von Tabellen, die durch Beziehungen miteinander verbunden sind. Dabei werden Daten in Zeilenform in Tabellen mit ein oder mehreren Spalten gespeichert, wobei jeder Datensatz einem eindeutigen Wert (Schlüssel) zugeordnet ist. Das heißt, dass ein oder mehrere Felder einer Tabelle eine eindeutige Identifizierung der Datensätze erlauben. So können Daten in beliebiger Reihenfolge angeordnet und abgespeichert werden, wobei niemals zwei Datensätze den gleichen Schlüssel besitzen dürfen. Im Extremfall können alle Spalten einer Tabelle einen eindeutigen Schlüssel bilden. Es ist möglich, dass eine Tabelle mehrere (zum Beispiel zwei) eindeutige Felder besitzt, die als Schlüssel dienen könnten. Dann wird das eine Feld als Hauptschlüssel, als sogenannter Primärschlüssel, ausgesucht und definiert, während der andere Schlüssel als Nebenschlüssel (Alternativschlüssel) verwendet wird. Eine dritte Art der Schlüssel ist der Fremdschlüssel. Das ist ein Schlüssel, der in einer Tabelle gespeichert ist und auf Zeilen einer anderen Tabelle verweist. Ist eine Datenbank gut aufgebaut, besteht keinerlei Redundanz zwischen nicht als Schlüssel verwendeten Feldern der Tabellen mehr.

Die Vorteile des relationalen Datenbankmanagementsystems liegen auf der Hand: Im Gegensatz zu hierarchischen und vernetzten Datenbankmanagementsystemen werden Daten einerseits als lineare Datensätze gespeichert, andererseits sind diese logisch und physisch unabhängig von anderen Datensätzen. Dies ermöglicht einfache Editierungen, ohne andere Datensätze zu verändern. Außerdem ist eine Kombination (Verknüpfung) von Tabellen ohne Probleme über den Schlüssel möglich.

Dabei sollten Tabellen einer relationalen Datenbank in der dritten Normalform vorliegen, um wirklich keine Abhängigkeiten zwischen den Datensätzen aufzuweisen. Die erste Normalform wird erreicht, wenn sich alle Spalten einer Tabelle und alle Daten dieser Spalten auf den Schlüssel der Tabelle beziehen [McCULLOUGH-DIETER, 1999, S. 142]. Wenn also die Daten einer Beispieltabelle „Radweg1“ wie in Abbildung 4 angeordnet sind, verstößt die

Tabelle gegen die erste Normalform, da die Spalten „Bearbeiter“ und „Bearb_Adresse“ nicht direkt vom Primärschlüssel „Radweg_Nr“ abhängen.

Tabelle RADWEG1

Radweg_Nr	Radwegname	Bearbeiter	Zustand	Bearb_Adresse
1	Westweg	Schnabel	gut	Nordstr. 1
2	Ostweg	Schnabel	mittel	Nordstr. 1
3	Südweg	Fehse	sehr gut	Südstr. 1
4	Nordweg	Fehse	schlecht	Südstr. 1

Abbildung 4: Die Tabelle RADWEG1 verstößt gegen die erste Normalform.

Richtig wäre die Einrichtung einer zweiten Tabelle „Personendaten“ wie in Abbildung 5, um bei der Änderung der Bearbeiteradresse oder bei der Änderung des Bearbeiternamens nicht die gleichen Daten mehrfach in verschiedenen Zeilen ändern zu müssen.

Tabelle RADWEG1

Radweg_Nr	Radwegname	Zustand	Bearb_Nr
1	Westweg	gut	1
2	Ostweg	mittel	1
3	Südweg	sehr gut	2
4	Nordweg	schlecht	2

Tabelle PERSONENDATEN

Bearb_Nr	Bearbeiter	Bearb_Adresse
1	Schnabel	Nordstr. 1
2	Fehse	Südstr. 1

Abbildung 5: Die Tabellen RADWEG1 und PERSONENDATEN erfüllen die erste Normalform.

Die zweite Normalform besagt, dass sich alle Spalten und deren Daten auf den gesamten Schlüssel und nicht nur auf Teile des Schlüssels beziehen müssen [McCULLOUGH-DIETER, 1999, S. 142]. Die Beispieltabelle „Radweg2“ in der Abbildung 6 verstößt gegen die zweite Normalform. Der Schlüssel dieser Tabelle wird aus den Spalten „Radweg_Nr“ und „Belag“ gebildet. Die Daten der

Spalte „Zustand“ hängen zwar vom gesamten Schlüssel ab, die Daten der Spalte „Radwegname“ jedoch nur von der Spalte „Radweg_Nr“.

Tabelle RADWEG2

Radweg_Nr	Belag	Radwegname	Zustand
1	Asphalt	Westweg	gut
1	Beton	Westweg	mittel
2	Asphalt	Ostweg	sehr gut
3	Verbundstein	Nordweg	schlecht

Abbildung 6: Die Tabelle RADWEG2 verstößt gegen die zweite Normalform.

Daher wird eine separate Tabelle mit den Spalten „Radweg_Nr“ und „Radwegname“ angelegt. Ändert sich nun der Name des Radweges, muss er nur noch in der neuen Tabelle „Name“ editiert werden, wie in der Abbildung 7 unschwer zu erkennen ist.

Tabelle RADWEG2

Radweg_Nr	Belag	Zustand
1	Asphalt	gut
1	Beton	mittel
2	Asphalt	sehr gut
3	Verbundstein	schlecht

Tabelle NAME

Radweg_Nr	Radwegname
1	Westweg
2	Ostweg
3	Nordweg

Abbildung 7: Beide Tabellen entsprechen der zweiten Normalform.

Ist eine Tabelle in der dritten Normalform aufgebaut, hängen alle Nicht-Schlüssel-Spalten und deren Daten nur noch vom Schlüssel und nicht mehr von anderen Nicht-Schlüssel-Spalten der Tabelle ab [McCULLOUGH-DIETER, 1999, S. 142]. Die Tabelle „Radweg3“ in der Abbildung 8 verstößt zum Beispiel gegen die dritte Normalform, nicht aber gegen die zweite Normalform, denn die drei Spalten „Radwegname“, „Gemeinde_Nr“ und „Gemeinde“ hängen ja von der Schlüsselspalte „Radweg_Nr“ ab. Allerdings tritt eine Redundanz zwischen den Spalten „Gemeinde_Nr“ und „Gemeinde“ auf.

Tabelle RADWEG3

Radweg_Nr	Radwegname	Gemeinde_Nr	Gemeinde
1	Westweg	2	Markkleeberg
2	Ostweg	3	Pegau
3	Nordweg	2	Markkleeberg
4	Südweg	1	Zwenkau

Abbildung 8: Die Tabelle RADWEG3 verstößt gegen die dritte Normalform.

Tabelle RADWEG3		
Radweg_Nr	Radwegname	Gemeinde_Nr
1	Westweg	2
2	Ostweg	3
3	Nordweg	2
4	Südweg	1

Tabelle GEMEINDE	
Gemeinde_Nr	Gemeinde
1	Zwenkau
2	Markkleeberg
3	Pegau

Abbildung 9: Die Tabellen in dritter Normalform

Die Lösung ist, wie in Abbildung 9 gezeigt, die Speicherung der Gemeinde-Daten in einer separaten Tabelle. Es bietet sich allerdings an, Tabellen immer nur mit zwei Spalten zu versehen (Ausnahme: Tabellen mit Koordinatenangaben bekommen zum Beispiel drei oder vier Felder „ID“, „X“, „Y“, „Z“, da die Koordinatenspalten zusammen gehören), um jeden Eintrag in der Tabelle genau einer ID eindeutig zuordnen zu können und die Zugriffszeiten auf Daten in den Tabellen zu minimieren. Dabei besteht der eindeutige Schlüssel zweckmäßigerweise aus fortlaufenden Nummern.

Relationale Datenbanken sind Sammlungen von Tabellen, die miteinander in Beziehungen (Relationen) stehen. Es werden zwei verschiedene Beziehungen unterschieden, nämlich die 1:1-Relation und die 1:n-Relation.

Bei 1:1-Relationen entspricht jeder Datensatz einer Zieltabelle genau einem Datensatz einer anderen (Quell-)tabelle.

Bei 1:n-Relationen dagegen kann ein Datensatz der Quelltable beliebig vielen Datensätzen der Zieltabelle zugeordnet werden.

2.4. Erstellung des logischen Konzepts

Nachdem die Voraussetzungen (siehe Kapitel 2.1. bis 2.3.) gegeben waren, musste ein logisches Konzept erstellt werden. Aufgrund von Literaturrecherchen und eigenen Überlegungen sowie Befragungen von Mitarbeitern des Landratsamtes wurde von folgenden Merkmalen der Radwege ausgegangen: dem Namen, der überwiegenden Nutzung, dem Zustand, dem Eigentümer, dem Belag, der Neigung, der Breite des Radweges, der Gemeinde, in der der Radweg liegt sowie der Verlauf des Weges auf einer Straße oder separat. Für Radwege, die auf einer Straße verlaufen, könnte es sinnvoll sein, deren Verkehrsbelastung und damit die Gefährdung des Radfahrers zu erfassen.

Alle oben genannten Überlegungen sollten sowohl für bestehende als auch für geplante Radwege gelten. Des Weiteren wurden Informationen zur Herkunft der Daten, zum Datum der Erfassung (Digitalisierung) des Radwegs, zur Genauigkeit und Vollständigkeit sowie zur Aktualität der Daten in die Überlegungen mit einbezogen. Diese Metainformationen sind wichtig, um die Qualität der Daten einschätzen zu können.

Aus dieser Fülle von Merkmalen wurden die für diese Arbeit relevanten nach Absprache mit der Abteilungsleiterin der Bauleitplanung und den Mitarbeitern des Bauplanungs- und Wirtschaftsförderungsamtes des Landratsamtes Leipziger Land extrahiert und mit den schon im Landratsamt vorhandenen Daten (siehe Einleitung) abgeglichen. Dabei kristallisierten sich folgende Merkmale heraus:

- das Erfassungsdatum: Das Erfassungsdatum ist der Zeitpunkt der Aufnahme in das GIS, zum Beispiel der 2. Februar 2001. Dieser Metadatenteil wurde extrahiert, um die Aktualität der Daten abschätzen zu können. Dabei wird davon ausgegangen, dass die aufzunehmenden Daten zum Zeitpunkt der Digitalisierung auf dem aktuellsten Stand sind.
- der Bearbeiter: Der Name des die GIS-Aufnahme durchführenden Bearbeiters, zum Beispiel Schnabel, als Teil der Metadaten wurde gewählt, um genau zu dokumentieren, wer Veränderungen an Daten durchgeführt hat und damit einen Ansprechpartner zu haben.

- die Datenquelle: Die Herkunft der Daten, zum Beispiel vom Straßenbauamt oder vom Landratsamt selbst, als Teil der Metadaten wurde ausgesucht, um die Qualität und Verlässlichkeit der Daten überprüfen zu können.
- der Radwegname: Der Name der Radroute, zum Beispiel Elsterradroute, ist ein weiteres Kriterium der Erfassung, wobei nur Radwege erster und zweiter Priorität (laut Radverkehrskonzeption des Freistaates Sachsen vom 01.01.1996) einen Namen haben. Das sind für den Landkreis Leipziger Land die Elsterradroute in erster Priorität, die Nord-Süd-Route, die Route Altenburg-Colditz sowie der Radweg Südliche Schlachtfelder in der zweiten Priorität.
- die Nutzung: Die Art des Verkehrs auf dem Radweg ist insofern interessant, da die Beseitigung von Gefahrenquellen und Problempunkten auf ständig befahrenen Radwegen (unter anderem durch Kinder) oberste Priorität hat. Es werden daher zwei Nutzarten unterschieden, der Schüler- und Alltagsverkehr sowie die touristische Nutzung.
- der Zustand: Der Radwegzustand wird nach der Art des Belages und seiner Beschaffenheit (Schlaglöcher und Pfützen) beurteilt. Dabei gingen die Dokumentationen lokaler Radwandervereinigungen in die Beurteilung mit ein. Es wurden drei Zustandsklassen „gut“, „mittel“ und „schlecht“ gebildet. Da nicht über jeden Radweg eine Dokumentation angefertigt wurde, liegen die Daten nicht vollständig vor.
- der Belag: Der Belag ist vom Autor in sechs Belagsarten gegliedert worden, wobei kein Anspruch auf Vollständigkeit erhoben, sondern nur von den durch die oben genannten Dokumentationen der örtlichen Vereine genannten Arten ausgegangen wird. Obwohl es auch hier keine vollständigen Unterlagen gibt, sind die Belagsarten, so weit vorhanden, in Asphalt, feiner Schotter, Kopfsteinpflaster, Verbundsteine, unbefestigt und Beton eingeteilt worden.
- die Breite: Die Breite des Radwegs in Metern wurde auf Grundlage der Straßenverkehrsordnung und der damit einhergehenden Beschilderung (Radweg im 1-Richtungsverkehr, kombinierter Rad- und Gehweg, Radweg im 2-Richtungsverkehr) in drei Klassen eingeteilt (kleiner als 1,5 Meter, 1,5 bis 2,0 Meter und größer als 2,0 Meter), obwohl auch diese Daten nicht für alle Radwege des Landkreises vorliegen.

- die Gemeinde: Es gibt 24 Städte und Gemeinden im Landkreis Leipziger Land, in der das entsprechende Teilstück des Radweges liegen kann. Diese sind in alphabetischer Reihenfolge: Böhlen, Borna, Deutzen, Elstertrebnitz, Espenhain, Eulatal, Frohburg, Geithain, Groitzsch, Großlehna, Großpösna, Kitzen, Kitzscher, Kohren-Sahlis, Lobstädt, Markkleeberg, Markranstädt, Narsdorf, Neukieritzsch, Pegau, Regis-Breitungen, Rötha, Wyhratal und Zwenkau.
- die Planung: Radwege im Planungsprozess, also noch nicht gebaute Radwege, bilden ein weiteres wichtiges Merkmal für die Radwegdatenbank. Das Attribut soll als Boolean-Datentyp (ja/nein) gespeichert werden.
- die Straßenbenutzung: Es werden zwei Klassen „ja“ und „nein“ differenziert. Dabei werden Radwegsegmente in die Klasse „ja“ eingeordnet, wenn diese Teil der Straße sind. Ein Radwegsegment gehört in die Klasse „nein“, wenn der Radweg nicht auf der Straße, sondern separat verläuft.
- die Belastung: Die Verkehrsbelastung entsteht durch ein erhöhtes Kraftfahrzeugaufkommen, wenn der Radweg ein Teil der Straße ist. Auch hier sollen nur zwei Klassen (hohe Verkehrsbelastung? ja/nein) unterschieden werden. Da diese Daten nicht in Reinform vorliegen, wurde die Bedingung aufgestellt, dass die Verkehrsbelastung dann hoch ist, wenn ein Radweg Teil einer Bundes-, Staats- oder Kreisstraße ist. Das heißt, das Merkmal Belastung ist eine Teilmenge des Merkmals Straßenbenutzung und ergänzt dieses.

Es waren keine Daten im Landratsamt vorhanden oder beschaffbar, die den Eigentümer eines Radweges definieren können. Generell kann jedoch angemerkt werden, dass Radwege an Bundes- und Staatsstraßen in den Zuständigkeitsbereich des Straßenbauamtes fallen. Radwege an Kreisstraßen dagegen gehören nach Recherchen des Autors dem Landkreis, alle anderen Radwege sind entweder in privatem Eigentum oder Eigentum der Gemeinde, in der der Radweg liegt [SOMMER, mündliche Mitteilung, 2001].

Das Attribut der Neigung wurde fallen gelassen, da kaum größere Steigungen im gesamten Landkreis vorhanden sind.

Damit war der Inhalt des Konzeptes klar definiert. Die extrahierten Radwegemerkmale konnten aus vorhandenen Daten erfasst werden.

Da bei der praktischen Arbeit meist noch Fragen und Unklarheiten auftauchen, diese jedoch bei diesem Projekt zu großen Änderungen, großem Zeitverlust oder sogar zur vollkommenen Unbrauchbarkeit der Daten führen können, wurde zuerst eine Testphase vor der Erstellung der eigentlichen Radwegedatenbank eingeschoben. Dabei wurde der gesamte Arbeitsablauf von der Digitalisierung über die Erstellung der Datenbank, der Verknüpfung der Datenbank mit der Geometrie im GIS, der Simulation des Verhaltens von Avenue-Scripten bis zu Tests der Aktualisierung der Geometrien und der Datenbank anhand einiger Beispielradwege durchgeführt. Erst nach der erfolgreichen Testphase, bei der einige gravierende Probleme auftraten und gelöst werden konnten, wurde mit der eigentlichen Arbeit, die in den nachfolgenden Kapiteln beschrieben ist, begonnen. Diese Vorgehensweise zählt sich bei Projekten mit größeren Datensätzen aus und kann vom Autor nur empfohlen werden.

3. Überführung der analogen Daten in ein digitales Format

3.1. Vorüberlegungen zur Digitalisierung

Bevor eine Digitalisierung der Radwege vorgenommen werden konnte, mussten einige Vorüberlegungen zur Digitalisierung angestellt werden. Grundlage für diese Raster-Vektor-Konvertierung waren bereitgestellte georeferenzierte Topographische Karten im Maßstab 1:10.000. Diese großmaßstäbigen Karten wurden als Themen (Ebenen) in ein neues View „Radwege“ im GIS ArcView (siehe Kapitel 2.1.) eingefügt und lagen damit schon im Gauß-Krüger-Koordinatensystem vor. Deshalb brauchte keine Projektion für das View „Radwege“, in dem die Digitalisierung durchgeführt wurde, definiert werden, da diese automatisch voreingestellt wird. Die Kartier- und Abstandseinheiten waren auch noch nicht eingestellt, was an der fehlenden Anzeige des Maßstabs zu erkennen war. Da diese jedoch wichtig für Raumanalysefunktionen sowie die Entfernungsberechnung sind, mussten sie beide unter dem Menüpunkt „View“, Unterpunkt „Eigenschaften“ auf Meter gestellt werden. Das View „Radwege“ wird nun in Metern projiziert, und bei Pufferberechnungen um Elemente dieses Views wird auch die Einheit Meter benutzt.

Da die Ausrichtung der digitalisierten Radwege exakt sein muss, ist die Einstellung der allgemeinen Einpassungsumgebung durch Aktivierung des neuen Themas „Radwege“, in dem die Radwege digitalisiert werden sollen, notwendig. Dazu kann man, wie in Abbildung 10 gezeigt, unter dem Menüpunkt „Thema“, Unterpunkt „Eigenschaften“ im Blatt „Bearbeitung“ die allgemeine Einpassung aktivieren und die Toleranz, innerhalb derer die Radwegpunkte zusammengeführt werden sollen, eingeben.

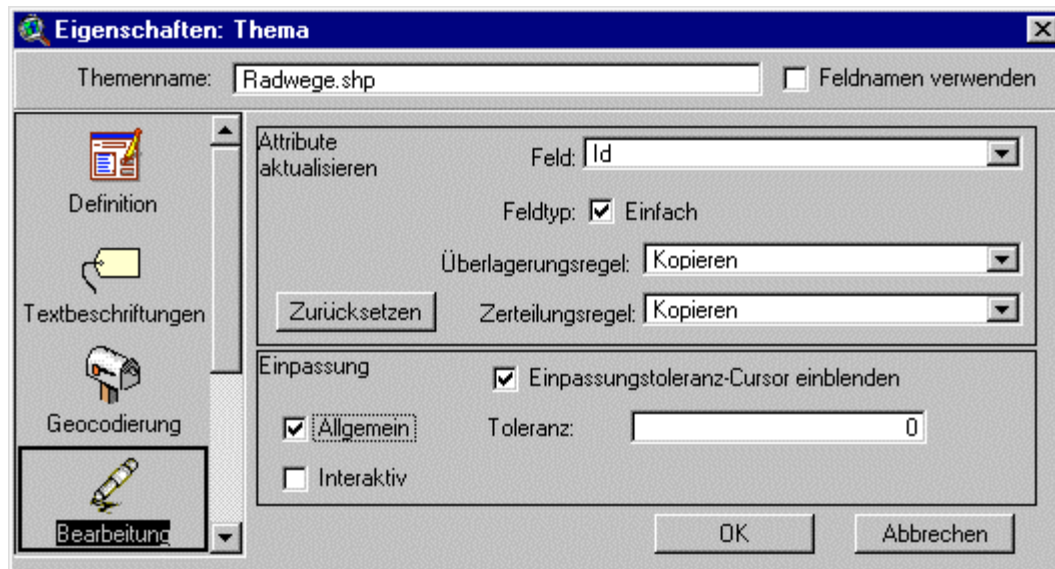


Abbildung 10: Aktivierung der allgemeinen Einpassung vor der Digitalisierung

Des Weiteren war es wichtig, Regeln zur Digitalisierung der Radweg-Stützpunkte festzulegen. Stützpunkte werden demnach an Punkten auf der Radroute gesetzt, wo:

- a) Kreis- oder Gemeindegrenzen verlaufen, da Gemeindegrenzen Planungseinheiten und unter gewissen Umständen Zuständigkeitseinheiten für die Instandsetzung und den Neubau sind;
- b) sich Planungshoheiten und Belastungssituationen ändern können (zum Beispiel dort, wo eine Bundesstraße zur Kreisstraße oder eine befestigte Straße zum unbefestigten Waldweg wird);
- c) Radwege Flüsse oder Eisenbahnschienen überqueren (auch von zum Teil ungenutzten Gleistrassen des ehemaligen Braunkohlebergbaus wie nördlich von Borna, da diese Trassen teilweise als Grundlage für neue Radwege genutzt werden);
- d) Kreuzungen auftreten, da diese logische Anknüpfungspunkte für Neutrassierungen und Streckenveränderungen sind (Zum Beispiel soll die Elsterradroute südlich der Ortslage Pegau direkt an die Elster verlegt werden, was einen Neubau und damit eine Streckenänderung zur Folge hätte.).

Die Durchführung der Digitalisierung unter diesen Voraussetzungen war Grundlage für eine effektive Segmentierung der Radwege.

3.2. Segmentierung

Die Segmentierung der Radwege wurde vorgenommen, da jedes Segment eines Radweges andere Attribute, beispielsweise unterschiedliche Belagsarten oder Nutzungen, haben kann. Eine Segmentierung im ArcView, das heißt eine Zerlegung der digitalisierten Linien an den Stützpunkten und eine Durchnummerierung dieser Segmente, ist daher für das Anhängen von Attributen unabdinglich.

Im Menüpunkt „Scripte“ im GIS ArcView (siehe auch Kapitel 2.1., Abbildung 2) kann ein leeres Scriptfenster mit dem Button „Neu“ erzeugt werden und eine Avenue-Scriptdatei (*.ave), die vorher in einem Texteditor bearbeitet werden kann, eingelesen und bearbeitet und das Script vom Benutzer kompiliert werden, um es auf Fehler zu überprüfen (Abbildung 11).



Abbildung 11: Der Button zum Kompilieren eines Scriptes

Anschließend kann es theoretisch ausgeführt werden (Abbildung 12). Meist sind jedoch gewisse Voraussetzungen nötig, um das Script ausführen zu können, da sonst Fehlermeldungen unvermeidlich sind.



Abbildung 12: Der Button zum Ausführen eines Scriptes

Genutzt wurde für die Segmentierung ein Script von Eugene Martin [MARTIN, 1998], welches vom Autor für die Zwecke des Landratsamtes modifiziert wurde (Anhang I). Voraussetzungen für die Ausführung des Scriptes waren:

- die Aktivierung des Views, in dem die digitalisierten Radwege enthalten sind,
 - die Aktivierung des Themas, in dem die digitalisierten Radwege enthalten sind
- sowie

- die Auswahl der Linien, die segmentiert werden sollen (keine zwanghafte Bedingung).

Außerdem wurden in der Attributtabelle des Themas „Radwege“ im Feld „ID“ die digitalisierten Linien durchnummeriert.

Wenn alle Voraussetzungen erfüllt sind, erzeugt dieses Script zuerst ein Ausgabethema als Shape-Datei und kontrolliert dann, ob Linien im aktiven Thema ausgewählt sind. Sind keine Linien markiert, werden alle Linien des aktiven Themas ausgewählt. Dann werden dort die Felder „Shape“ und „ID“, die als Standard in der Attributtabelle vorgegeben sind, gesucht sowie im Ausgabethema die Felder „Shape“, „SEGMENT_ID“ und „LINE“ erstellt. Nun zerlegt das Script die Linien an den Stützpunkten, nummeriert diese neu entstandenen Liniensegmente durch und fügt im Ausgabethema unter der Spalte „Shape“ die Einträge „Shape“ des aktiven Themas, unter der Spalte „SEGMENT_ID“ die Nummern der Liniensegmente und unter der Spalte „LINE“ die vorher im aktiven Thema „Radwege“ im Feld „ID“ eingetragenen Nummern der digitalisierten Linien ein. Jetzt kann der Benutzer wählen, in welches View er das neu erstellte Ausgabethema einfügen soll. Das Ausgabethema „Radw_seg“ hat nun eine Attributtabelle, in der die drei Spalten „Shape“, „SEGMENT_ID“ und „LINE“ vorhanden und mit Datensätzen gefüllt sind. Während die reine Geometrie im ArcView in einer Shape-Datei (*.shp) gespeichert wird, sind die Daten der Attributtabelle in einer dBase-Datei (*.dbf) enthalten.

Die Segmentierung und Erzeugung der dBase-Datei stellt den Schlüsselpunkt zwischen der Geometrie im ArcView und der Datenbank im Oracle dar und sichert in vorliegender Form eine Weiterentwicklung und Aktualisierung des Datenbankmanagementsystems.

4. Aufbau der relationalen Datenbank

4.1. Erstellung des physischen Konzepts

Nach der Erstellung eines logischen Entwurfs (Kapitel 2.4.) ist nun ein klares physisches Datenbankkonzept die Voraussetzung für den Aufbau der relationalen Datenbank. Beschreibt das logische Konzept nur den Inhalt der Datenbank (Radwegemerkmale), so müssen im physischen Konzept die Datentypen und Relationsarten sowie die Namen von Tabellen und Spalten enthalten sein.

Die Grundlage der Merkmalstabellen und Anbindungspunkt der GIS-Geometrie an diese Tabellen ist die Tabelle „SEGMENT“. Diese wird, wie im Kapitel 4.2. beschrieben, aus der Attributtabelle des Themas „Radw_seg“ (dBase-Datei) gewonnen und enthält die Spalten „SEGMENT_ID“ und „LINE“ aus derselben. Des Weiteren werden Tabellen der Merkmale im Oracle-SQL-Eingabefenster erstellt. Da die Tabellen erweiterbar sein und in der dritten Normalform (Kapitel 2.3.) vorliegen sollen, besitzen sie nur zwei Spalten, wobei eine davon den Primärschlüssel (die fortlaufende Nummer) enthält. Diese Merkmalstabellen haben ebenso wie die Tabelle „SEGMENT“ („SEGMENT_ID“) eine fortlaufende Nummer, die durch die Bezeichnung „*_ID“ erkennbar ist (Abbildung 13), während die zweite Spalte die eigentlichen Attribute der Radwege enthält.

Um nun die Merkmalstabellen mit der Geometrie im GIS (in der Datenbank durch die Tabelle „SEGMENT“ repräsentiert) verbinden zu können, müssen Referenztabellen erzeugt werden, die durch ein „REF_*“ gekennzeichnet sind (Abbildung 13). Diese enthalten in der Spalte „SEGMENT_ID“ den Primärschlüssel, wohingegen sich der Fremdschlüssel aus dem Primärschlüssel der Merkmalstabellen (zum Beispiel: DATE_ID) ergibt. Dabei existiert zwischen der Tabelle „SEGMENT“ und der Referenztable eine 1:1-Beziehung (Abbildung 13), denn jeder Identifikationsnummer in der Spalte „SEGMENT_ID“ der Tabelle „SEGMENT“ kann genau eine fortlaufende Nummer in der Spalte „SEGMENT_ID“ der Referenztable zugeordnet werden. Zwischen der Referenztable und der Merkmalstabelle tritt dagegen eine 1:n-Relation auf (Abbildung 13), da zwar jede Segment-ID nur einmal auftritt, jede

Merkmals-ID jedoch mehrfach. Diese Beschreibungen gelten für die Merkmale Name, Erfassungsdatum, Bearbeiter, Quelle, Nutzung, Zustand, Belag, Breite sowie Gemeinde. Da die Merkmale Planung, Belastung und Straßenbenutzung nur Boolean-Werte (ja/nein) enthalten, werden diese wie Referenztabellen behandelt. Das heißt, sie stehen in einer 1:1-Relation zur Ausgangstabelle „SEGMENT“ und bekommen als Primärschlüssel die Spalte „SEGMENT_ID“, so dass in der zweiten Spalte nur die Werte „ja“ und „nein“ eingetragen werden. Angaben zur Erstellung der Tabellen (default-Werte) runden das physische Konzept ab (Abbildung 13), welches dann wie folgt aussieht:



Abbildung 13: Das physische Radwege-Konzept der Oracle-Datenbank

Die Oracle-Datenbank enthält vier Datentypen:

Datentypen		
Abkürzung	Datentyp	Beschreibung
N	NUMBER (10, 0)	maximal 10 Stellen, keine Nachkommastellen
D	DATE	Datum
V	VARCHAR2 (50)	maximal 50 Buchstaben
V2	VARCHAR2 (4)	Boolean-Simulation: ja/nein

Da kein Datentyp Boolean im Oracle-Datenbankmanagementsystem existiert, muss mit dem Textformat VARCHAR2 (Variable Character String, Version 2) dieser Datentyp simuliert werden. Der Datentyp NUMBER wird für alle ID-Felder (und damit für alle Referenztabellen sowie die Tabelle „SEGMENT“) benutzt. Oracle speichert das Datum in dem speziellen Datumsformat DATE (TT-MMM-JJ, zum Beispiel: 02-FEB-01).

Somit stellt sich der Inhalt der neun Merkmalstabellen folgendermaßen dar (Datentyp in Klammern):

- Tabelle **ERF_DATE**:

DATE_ID (N)	ERF_DATUM (D)
1	02-FEB-01
2	20-MAR-01

- Tabelle **BEARB**:

BEARB_ID (N)	BEARB_NAME (V)
1	Schnabel

- Tabelle **QUELLE**:

QUELLE_ID (N)	DATENQUELLE (V)
1	Straßenbauamt
2	Landratsamt

- Tabelle **NAME**:

NAME_ID (N)	RADWEGNAME (V)
1	Elsterradroute
2	Südliche Schlachtfelder
3	Nord-Süd-Route
4	Altenburg-Colditz

- Tabelle **NUTZUNG**:

NUTZUNG_ID (N)	NUTZART (V)
1	Schüler- und Alltagsverkehr
2	Tourismus

- Tabelle **ZUSTAND**:

ZUSTAND_ID (N)	WEGZUSTAND (V)
1	gut
2	mittel
3	schlecht

- Tabelle **BREITE**:

BREITE_ID (N)	BREITE_M (V)
1	< 1,5
2	1,5 – 2,0
3	> 2,0

- Tabelle **BELAG**:

BELAG_ID (N)	BELAGSART (V)
1	Asphalt
2	feiner Schotter
3	Kopfsteinpflaster
4	Verbundsteine
5	unbefestigt
6	Beton

- Tabelle **GEMEINDE**:

GEMEINDE_ID (N)	GEM_NAME (V)
1	Böhlen
2	Borna
3	Deutzen
4	Elstertrebnitz
5	Espenhain
6	Eulatal
7	Frohburg
8	Geithain
9	Groitzsch
10	Großlehna
11	Großpösna
12	Kitzen
13	Kitzscher
14	Kohren-Sahlis
15	Lobstädt
16	Markkleeberg
17	Markranstädt
18	Narsdorf
19	Neukieritzsch
20	Pegau
21	Regis-Breitingen
22	Rötha
23	Wyhratal
24	Zwenkau

4.2. Erstellung der Datenbank

Bevor die im Kapitel 4.1. beschriebenen Merkmalstabellen sowie ihre Referenztabellen erstellt werden konnten, musste zuerst der Tablespace GIS eingerichtet werden. Der Tablespace ist der Platz eines Datenbankmanagementsystems, an dem alle Tabellen des jeweiligen Benutzers gespeichert werden und der vom Rest des Datenbankmanagementsystems abgetrennt ist [McCULLOUGH-DIETER, 1999, S. 181]. Dieser Tablespace wird normalerweise vom Systemadministrator des Datenbankmanagementsystems bei der Einrichtung der Benutzerrollen vergeben. Dabei werden im Programm „SQL Net Easy Configuration“ der Tablespace-Name (gis) unter dem Menüpunkt „add database alias“, die Serveradresse unter dem Menüpunkt „TCP/IP“ und die Instanz (gis) unter dem Menüpunkt „Instance“ angegeben. Nun können die Merkmalstabellen mit dem SQL-Befehl

```
create table <Tabellenname>  
(<Spaltenname der ersten Spalte> number (10, 0) not null primary key,  
 <Spaltenname der zweiten Spalte> <Datentyp der zweiten Spalte>);
```

Beispiel:

```
create table ERF_DATE  
(DATE_ID number(10,0) not null primary key,  
 ERF_DATUM date);
```

erstellt werden. SQL-Befehle werden immer mit einem Semikolon abgeschlossen. Die erste Spalte ist im vorliegenden Fall immer die ID-Spalte, daher darf diese nie den Wert „Null“ annehmen und ist als Primärschlüssel definiert. Da die zweite Spalte meist Text enthält, muss nur der Datentyp angegeben werden. Um nun diese erzeugten Tabellen zu füllen, werden die Daten mit dem SQL-Befehl

```
insert into <Tabellenname>
(<Spaltenname der ersten Spalte>, <Spaltenname der zweiten Spalte>)
values
(<Wert der ersten Spalte>, <Wert der zweiten Spalte>);
```

Beispiel 1:

```
insert into ERF_DATE
(DATE_ID, ERF_DATUM)
values
(1, '02-02-01');
```

Beispiel 2:

```
insert into QUELLE
(QUELLE_ID, DATENQUELLE)
values
(1, 'Straßenbauamt');
```

eingefügt. Dabei muss beachtet werden, dass die Einträge beim Einfügen von Text und Datum in Hochkommas stehen müssen. Die Datenbank mit den so erzeugten Merkmalstabellen wird nun um die Ausgangstabelle „SEGMENT“ ergänzt.

Die Attributtabelle der segmentierten Radwege (Thema „Radw_seg“) liegt im ArcView als dBase-Tabelle vor. Diese dBase-Datei wird im Tabellenkalkulationsprogramm Microsoft Excel in das PRN-Format (*.prn: formatierter Text, mit Leerzeichen getrennt) umgewandelt. Nun kann der Oracle-Datenbankadministrator diese Tabelle im Betriebssystem UNIX (weil Oracle auf dem UNIX-Server liegt) mit Hilfe des SQL*Loaders einlesen und importieren. Vor dem Import muss ein Controlfile (Kontrolldatei) angelegt werden, in dem das Format der zu erstellenden Tabelle im Oracle beschrieben wird. Dieses Controlfile kann im Texteditor geschrieben und als *.ctl – Datei gespeichert werden (Anlage IV).

Nun müssen beide Dateien (das Controlfile rad.ctl und die ins Oracle einzufügende PRN-Datei radw_seg.prn) in das Oracle-Unterverzeichnis /oracle/orahome/bin kopiert werden. Nachdem in der Instanz „gis“ mit den Befehlen

```
ORACLE_SID = gis
```

```
export ORACLE_SID
```

das Einloggen erfolgte, kann der Import der Datei radw_seg.prn vonstatten gehen, indem das Controlfile rad.ctl im SQL*Loader ausgeführt wird:

```
sqlldr userid = gis control = rad.ctl log = rad.log
```

Dabei erzeugt das Programm ein Logfile rad.log (Import-Statistik), ein Badfile (inkorrekte Datensätze), ein Discardfile (nicht ausgewählte Datensätze) und natürlich die Oracle-Tabelle „SEGMENT“.

Dieser Weg ist zwar kompliziert, erleichtert aber die Arbeit in der Oracle-Datenbank deutlich, da sonst alle (über 6900) Einträge per Hand hätten eingetragen werden müssen.

Auf der Grundlage dieser Tabelle „SEGMENT“ können die Referenztabellen, die die Verbindung zwischen der Tabelle „SEGMENT“ und den Merkmalstabellen herstellen, nach folgendem Schema erzeugt werden:

```
create table <Referenztabellenname>  
as select SEGMENT_ID from SEGMENT  
where SEGMENT_ID > 0;
```

```
alter table <Referenztabellenname> add  
primary key (SEGMENT_ID);
```

```
alter table <Referenztabellenname> add  
<Spaltenname der zweiten Spalte> number (10, 0) [default <Defaultwert>;
```

```
alter table <Referenztabellenname> add  
foreign key (<Spaltenname der zweiten Spalte>) references <Name der Merkmalstabelle>  
(<Spaltenname der Merkmalstabelle mit Primärschlüssel>;
```

Der erste Befehl markiert alle Einträge in der Ausgangstabelle „SEGMENT“ und erstellt eine neue Tabelle (Referenztable) nur mit der Spalte „SEGMENT_ID“. Diese wird mit dem zweiten Befehl als Primärschlüsselspalte festgelegt, das heißt, dass weder Null-Werte noch doppelte Nummern vorkommen dürfen. Damit ist die Ausgangstabelle mit der Referenztable verbunden. Der dritte SQL-Befehl fügt eine zweite Spalte hinzu, die immer den Datentyp „number“ besitzen muss, da die Einträge aus den Primärschlüsselspalten der Merkmalstabellen stammen. Soll die zweite Spalte der zu erzeugenden Referenztable Standardwerte besitzen (nicht zwingend), muss im dritten Befehl ein default angehängt werden. Der vierte Befehl legt nun diese zweite Spalte als Fremdschlüssel fest, es wird auf die Primärschlüsselspalte der entsprechenden Merkmalstabelle verwiesen. Damit ist diese mit der Referenztable verbunden und eine logische Verknüpfung der Merkmale mit der Geometrie (SEGMENT_ID) gegeben. Die drei Tabellen „PLANUNG“, „BELASTUNG“ und „SEPARAT“ werden nur mit den ersten drei Befehlen erzeugt. In zwei Beispielen sieht das folgendermaßen aus:

Beispiel 1:

```
create table REF_NUTZUNG
as select SEGMENT_ID from SEGMENT
where SEGMENT_ID > 0;
```

```
alter table REF_NUTZUNG add
primary key (SEGMENT_ID);
```

```
alter table REF_NUTZUNG add
NUTZUNG_ID number (10, 0);
```

```
alter table REF_NUTZUNG add
foreign key (NUTZUNG_ID) references NUTZUNG (NUTZUNG_ID);
```

Beispiel 2:

```
create table PLANUNG
as select SEGMENT_ID from SEGMENT
where SEGMENT_ID > 0;
```

```
alter table PLANUNG add
primary key SEGMENT_ID;
```

```
alter table PLANUNG add
GEPLANT varchar2 (4) default 'nein';
```

Um nun Werte in die zweite Spalte einzutragen oder Defaultwerte zu verändern, wird der Befehl zum Aktualisieren und Verändern der Tabelle benutzt:

```
update <Tabellenname>
set <Spaltenname der zweiten Spalte> = <Neuer Wert>
where
<Bedingung>;
```

Beispiel 1:

```
update REF_NUTZUNG
set NUTZUNG_ID = 2
where
SEGMENT_ID between 238 and 272;
```

Beispiel 2:

```
update PLANUNG
set GEPLANT = 'ja'
where
SEGMENT_ID between 230 and 237
or SEGMENT_ID = 4073
or SEGMENT_ID between 842 and 872;
```

Zu beachten ist, dass Bedingungen beliebig an den Befehl angehängt werden können. Allerdings muss die korrekte Syntax eingegeben werden. So würde ein „and“ anstelle des „or“ im Beispiel 2 kein Ergebnis bringen, da innerhalb der ersten Bedingung nach der zweiten gesucht werden würde. Es gibt aber keine Segment-ID mit dem Wert „4073“ innerhalb der Werte „230“ und „237“. Die

Syntax von „between“ bedeutet nicht nur innerhalb der Werte, sondern einschließlich dieser.

Alle Merkmale mussten aus der Fülle des vorhandenen Datenmaterials extrahiert und per SQL-Befehl eingegeben werden. Mit den neun Referenztabelle sowie zusätzlich den drei Merkmalstabellen „PLANUNG“, „BELASTUNG“ und „SEPARAT“ war daher eine große Menge an Befehlen verbunden. Deshalb ist es nötig, im Oracle eine Fehlerkontrolle, die auf einfachen Auswahlanweisungen beruht, durchzuführen. Dabei werden die Datensätze nach ihrer logischen Konsistenz überprüft. So kann zum Beispiel ein geplanter Radweg weder genutzt noch ein erhöhtes Verkehrsaufkommen haben. Liegt ein Radweg auf der Straße, kann er nicht separat zu dieser geführt sein. Ebenso darf ein geplanter Radweg weder eine Breite, noch Belag, Zustand, Name oder Belastung haben. Um solche logischen Fehler auszuschließen, werden zum Beispiel folgende SQL-Befehle in das Eingabefenster eingetragen:

Beispiel 1:

```
select * from BELASTUNG, PLANUNG
where
BELASTUNG.SEGMENT_ID = PLANUNG.SEGMENT_ID
and GEPLANT = VERKEHRBELAST_HOCH
and GEPLANT = 'ja';
```

Beispiel 2:

```
select * from REF_NUTZUNG, PLANUNG
where
REF_NUTZUNG.SEGMENT_ID = PLANUNG.SEGMENT_ID
and NUTZUNG_ID between 1 and 2
and GEPLANT = 'ja';
```

Beispiel 3:

```
select * from PLANUNG, SEPARAT
where
PLANUNG.SEGMENT_ID = SEPARAT.SEGMENT_ID
and GEPLANT = 'ja'
and STR_BENUTZUNG = 'ja';
```

Ergeben diese Abfragen positive Ergebnisse, das heißt es werden Datensätze angezeigt, ist deren Korrektur notwendig.

4.3. Anbindung der Datenbank an die Geometrie im GIS

Die logische Verknüpfung der Merkmalstabellen mit den Referenztabellen sowie der Referenztabellen mit der Tabelle „SEGMENT“ ermöglicht zwar die Zuordnung der Daten (Merkmale) zu den einzelnen Radwegsegmenten, stellt aber keine interaktive Verbindung zwischen der Datenbank im Oracle und der Geometrie in ArcView her. Das bedeutet, dass Veränderungen im ArcView, also Geometrieänderungen, keine Änderungen in der Datenbank und umgekehrt bewirken. Die zwei Programmteile sind vollkommen voneinander getrennt.

Um beide nun miteinander zu verbinden, kann eine einseitige Verbindung zwischen Oracle und ArcView über eine SQL-Verbindung im ArcView hergestellt werden (Abbildung 14). Dieses GIS verwendet die offene Datenbankschnittstelle ODBC (Open DataBase Connectivity) von Microsoft, um Verbindungen zu SQL-Datenbanken herzustellen. ODBC kann dabei auf Daten in relationalen, aber auch in nicht-relationalen Datenbankmanagementsystemen zugreifen. Nach dem Aufbau der Verbindung mit dem Datenbankserver wird eine SQL-Abfrage an die Datenbank geschickt. Die abgefragten Datensätze werden in einer neuen Tabelle gespeichert, die bei einem Abbruch der Verbindung bei gleichzeitiger Entfernung aller Abfragedaten wieder gelöscht werden [ESRI, 1999].

Wird keine Verbindung (Abbildung 14) „gis“ angezeigt (Datenbank, in der die Radwegtabellen liegen), muss erst der ODBC-Treiber für diese Datenbank installiert werden, um diese dann im ArcView verwenden zu können. Dazu wird das Programm "ODBC Administrator" benutzt. Die Datenbank kann nun mit dem Button „Hinzufügen“ im Menü „System-DSN“ unter Angabe des Datenbanknamens, des Passwortes, der Instanz und des Servers eingearbeitet werden. Um nun die SQL-Verbindung herzustellen, muss nur im ArcView auf den Menüpunkt „Projekt“ (wenn das Projektfenster aktiv ist) und den Unterpunkt „SQL-Verbindung...“ geklickt werden. Dadurch wird das Verbindungsfenster (Abbildung 14) aufgerufen.

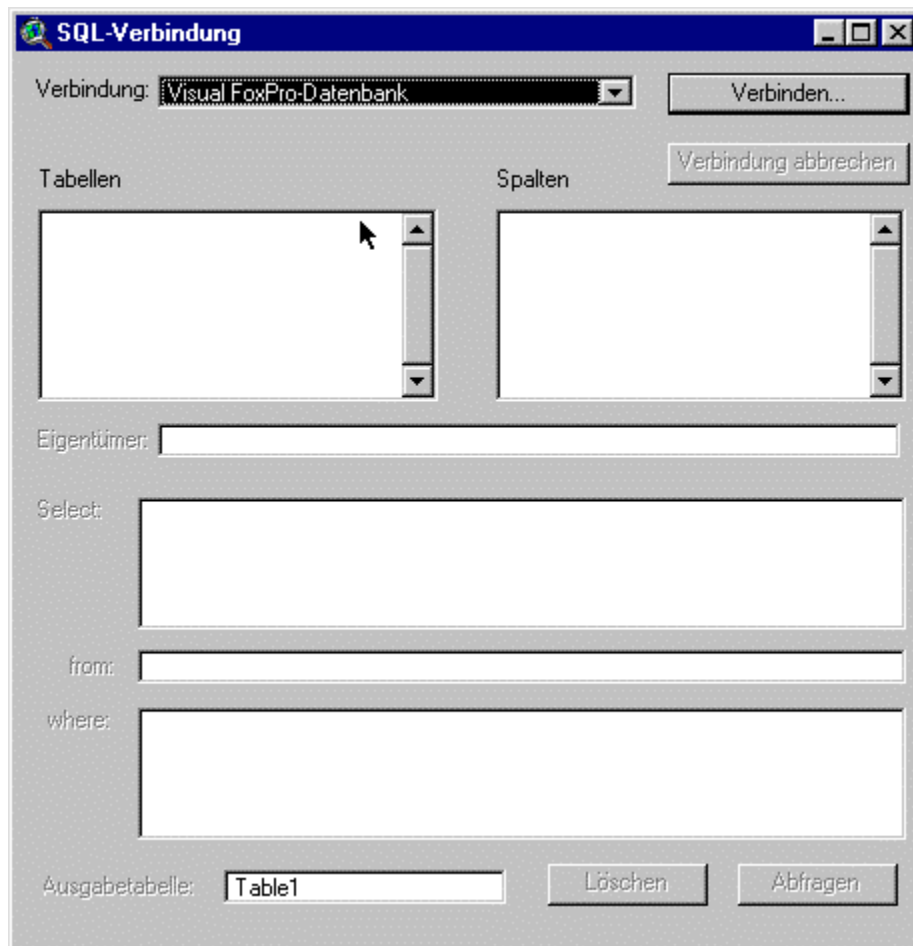


Abbildung 14: Herstellung einer SQL-Verbindung im ArcView

Ist ArcView mit der Datenbank („gis“) verbunden, werden automatisch im Fenster „Tabellen“ alle Tabellen der Datenbank angezeigt. Bei der Auswahl einer Tabelle (Doppelklick) erscheinen im Fenster „Spalten“ die Spalten der ausgewählten Tabelle. Außerdem wird der Tabellennamen in das Fenster „from:“ eingefügt. Die angezeigten Spalten können entweder einzeln (Doppelklick auf den Spaltennamen) oder alle (Doppelklick auf das Sternchen) ausgewählt werden, worauf sie im Fenster „Select:“ erscheinen. Nun kann noch der Name der Ausgabetabelle eingetragen werden. Dann wird die SQL-Abfrage mit dem Button „Abfragen“ an die Datenbank geschickt und das abgefragte Ergebnis in das GIS ArcView eingefügt.

Sind nun alle Merkmals- und Referenztabelle (außer der Tabelle „SEGMENT“) im ArcView verfügbar, können diese mit der Attributtabelle des Themas „Radw_seg“ über den Button „Verbinden“ verbunden werden. Dabei wird wie folgt vorgegangen: Beide zu verbindenden Tabellen (zum Beispiel

REF_QUELLE und die Attributtabelle) werden geöffnet und in jeder Tabelle die gemeinsame Spalte (zum Beispiel SEGMENT_ID) markiert. Es ist wichtig, dass das Fenster der Attributtabelle aktiv ist. Dadurch wird der Button „Verbinden“ aktiviert, der bei seiner Benutzung die zweite Spalte an die Attributtabelle anhängt (zum Beispiel QUELLE_ID). Der gleiche Vorgang kann mit der Merkmalstabelle durchgeführt werden (QUELLE_ID als Verbindungsspalte, Spalte DATENQUELLE wird eingefügt). Über den Menüpunkt „Tabelle“ Unterpunkt „Eigenschaften“ können dann die ID-Felder unsichtbar gemacht werden. Die Abfrage im View „Radwege“ mit dem Button „Identifizieren“ zeigt nach erfolgreicher Verbindung aller Merkmalstabellen mit der Attributtabelle ein Ergebnis wie in Abbildung 15.

Eine zweite Möglichkeit, die Oracletabellen mit der Geometrie im GIS zu verbinden, besteht in der Verknüpfung der Tabellen mit der Attributtabelle. Bei der Auswahl eines Datensatzes der Attributtabelle würden automatisch die entsprechenden zugehörigen Datensätze in den anderen verknüpften Tabellen markiert, jedoch nicht für die Abfrage verwendet werden [ESRI, 1999]. Daher ist diese Möglichkeit nicht sinnvoll.

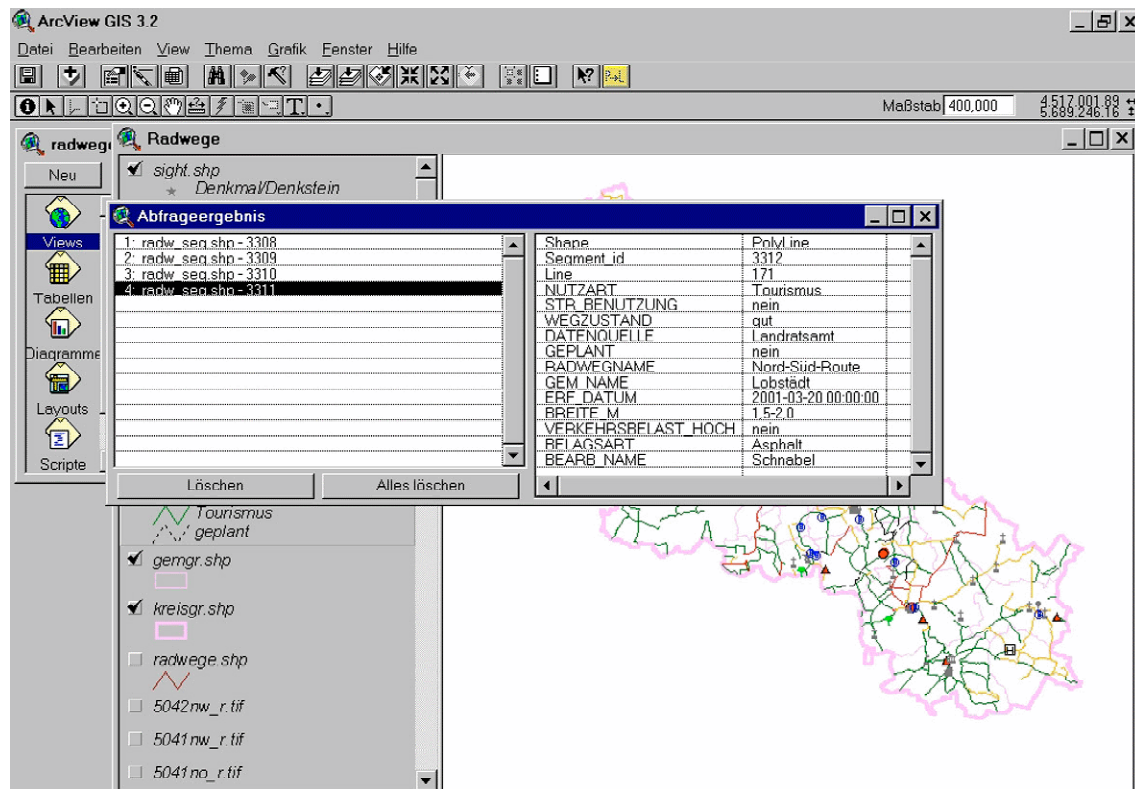


Abbildung 15: Abfrageergebnis nach erfolgreicher Verbindung aller Tabellen

Wird die Verbindung zur Datenbank im SQL-Verbindungs Fenster von ArcView abgebrochen, werden alle Daten (auch in die Attributtabelle eingefügte) sowie Datenbanktabellen aus dem GIS gelöscht. Bleibt die Verbindung bestehen und das Projekt wird bei laufender Verbindung gespeichert, bleiben alle eingefügten Tabellen und Daten erhalten und werden bei jedem Neustart von ArcView aktualisiert. Das heißt, dass Änderungen in der Oracle-Datenbank beim nächsten Öffnen von ArcView wirksam werden. Da dies aber nur in dieser einen Richtung (von Oracle zu ArcView) funktioniert, ist die Verbindung einseitig.

5. Möglichkeiten und Grenzen der Aktualisierung

Der im Kapitel 4.3. beschriebene Zustand der einseitigen Verbindung ist für den Anwender sehr unkomfortabel. Erstrebenswert wäre eine interaktive (beidseitige) Verbindung beider Programme. Trotz umfangreicher Bemühungen (Befragung von ArcView- und Oracle-Experten, Literaturrecherche) existiert derzeit keine akzeptable benutzerfreundliche Lösung [KAUFMANN, fernmündliche Mitteilung, 2001; ESRI, fernmündliche Mitteilung, 2001].

Das Problem wäre lösbar mit dem Programm Access von Microsoft, in dem Verbindungen zu Oracle und ArcView implementiert sind. Die Stabilität von Access in Bezug auf Abstürze ist jedoch deutlich niedriger als das auf dem UNIX-Server laufende Oracle. Es können dort aber die Attributtabelle der digitalisierten Themen mit den Daten in der Datenbank verknüpft werden. Daher ist für das Landratsamt Leipziger Land längerfristig folgende Verfahrensweise zu empfehlen:

- a) Übertragung der Oracle-Datenbank in Microsoft Access,
- b) Verknüpfung dieser Access-Datenbank mit ArcView,
- c) Verknüpfung von ArcView (Attributtabelle) mit der Access-Datenbank.

Danach ist ein interaktives Arbeiten (und damit eine Aktualisierung) in beide Richtungen (Access zu ArcView und ArcView zu Access) möglich. Leider war die Benutzung des Datenbankmanagementsystems Access, wie im Kapitel 2.2. erwähnt, aus lizenzrechtlichen Gründen nicht möglich. Deshalb mussten andere Wege zur Aktualisierung in beide Richtungen gefunden werden.

Eine Alternative der Aktualisierung von Oracle zu ArcView bietet das Microsoft Excel – Unterprogramm Microsoft Query. In dieses Programm können SQL-Tabellen eingelesen und bearbeitet werden. Dabei bietet Microsoft Query den unschätzbaren Vorteil, dass Tabellen der Oracle-Datenbank mit allen Datensätzen angezeigt werden können, das heißt, auch ungeübte Nutzer ohne SQL-Kenntnisse könnten die Oracle-Datenbank laufend halten. Dazu wird im Excel über den Menüpunkt „Daten“ der Unterpunkt „Daten importieren“ aufgerufen, was den Start von Microsoft Query zur Folge hat. Nun erscheint ein Fenster „Datenquelle auswählen“ (Abbildung 16).

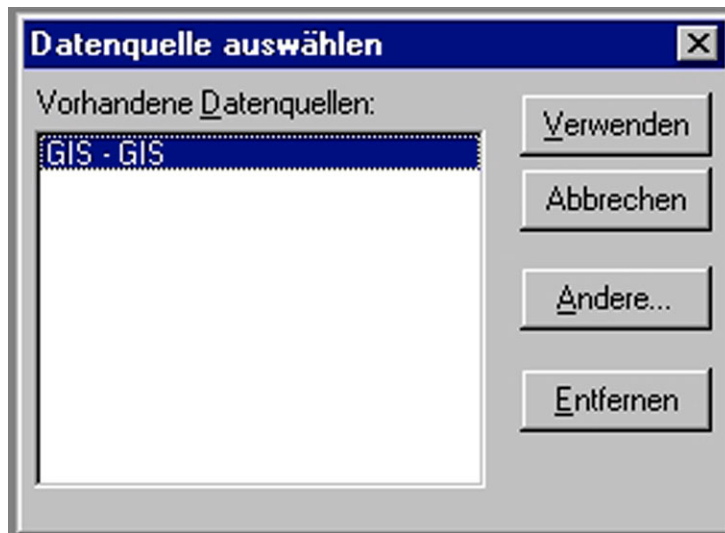


Abbildung 16: Auswählen der zu editierenden Datenbank im Microsoft Query

Dort kann die zu ändernde Datenbank (hier: gis, andere Schreibweise möglich) über den Button „Andere...“ ausgewählt und mit dem Button „Verwenden“ in Microsoft Query eingefügt werden. Dazu muss sich vorher in die Oracle-Datenbank eingeloggt werden (Abbildung 17):

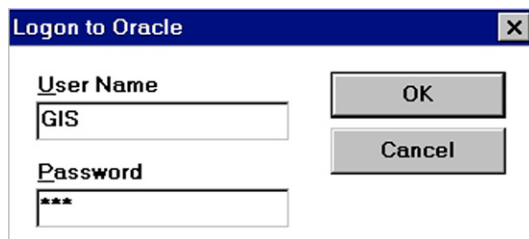


Abbildung 17: Einloggen in die Datenbank

Dann kann die zu editierende Tabelle aus der Oracle-Datenbank ausgewählt, der Abfrage hinzugefügt (Abbildung 18) und deren Spalten (Doppelklick auf den Stern) angezeigt werden (Abbildung 19).

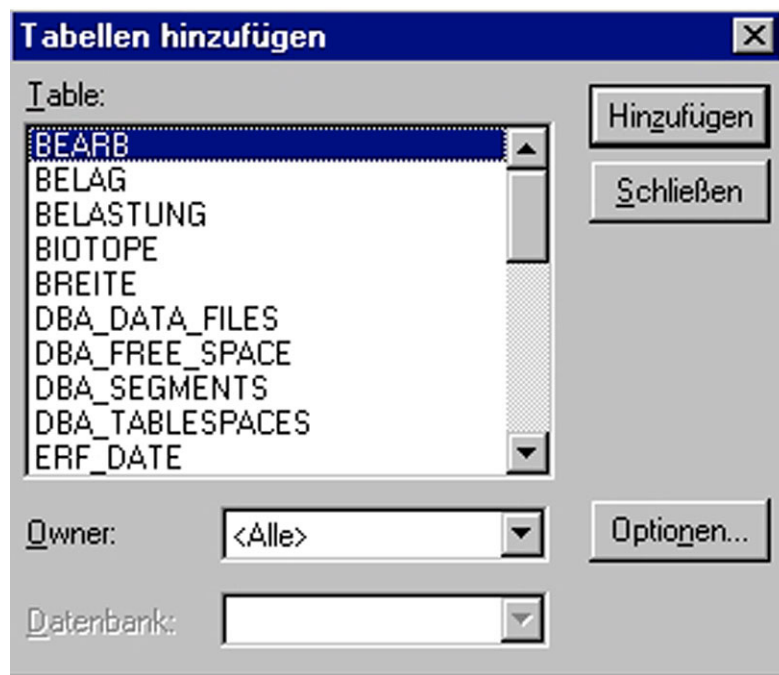


Abbildung 18: Einfügen der zu editierenden Tabelle in Microsoft Query



Abbildung 19: Anzeigen der zu editierenden Tabelle

Im Menü „Datensätze“ kann man den Untermenüpunkt „Bearbeiten ermöglichen“ wählen (Abbildung 20). Damit ist die Tabelle zur Bearbeitung bereit.

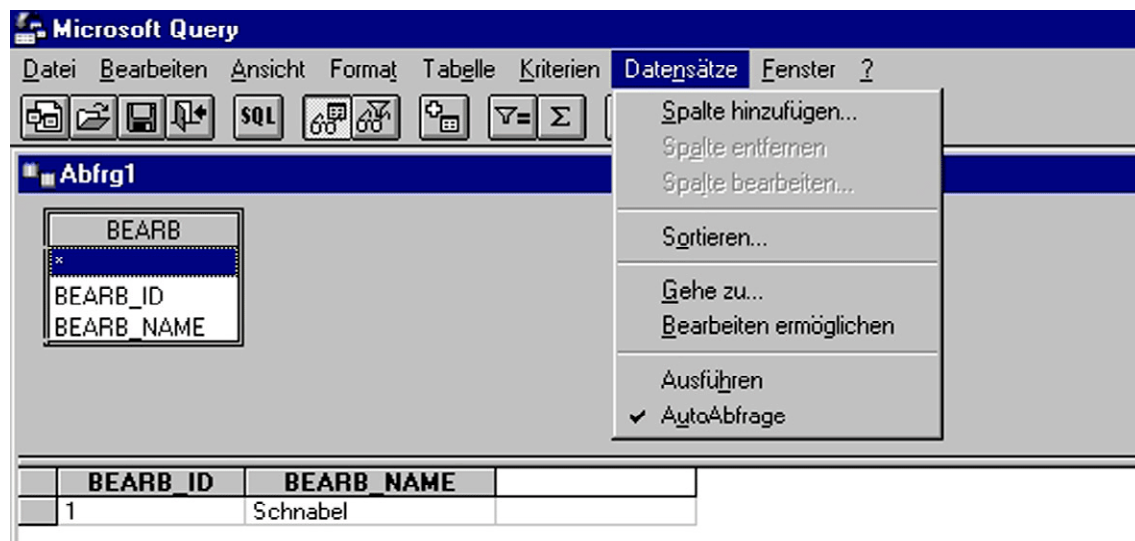


Abbildung 20: Anwählen von „Bearbeiten ermöglichen“ zur Tabelleneditierung

Es können neue Datensätze hinzugefügt und alte Daten verändert werden. Ist die Aktualisierung erfolgt, wird der Untermenüpunkt „Bearbeiten ermöglichen“ ausgeschaltet und die Tabelle unter „Datei“ und „Speichern unter“ gespeichert. Dazu muss wieder die Datenquelle ausgewählt (Datenbank „gis“) und der Tabellename noch einmal eingegeben werden (wie vorher, zum Beispiel BEARB), wobei die Groß- und Kleinschreibung zu beachten ist. Dann kann die Tabelle in die Oracle-Datenbank gespeichert werden, indem die Meldung „Tabelle besteht bereits. Ändern?“ bejaht wird. Eventuell auftretende Fehlermeldungen, die den Fremdschlüssel betreffen, sind zu ignorieren und mit „Ok“ zu beantworten. Danach können sowohl Microsoft Query als auch Excel beendet werden, ohne dass Abfragen gespeichert werden müssen.

Mit diesem recht einfachen Verfahren können Tabellen einer Oracle-Datenbank geändert werden, ohne das Datenbankmanagementsystem Oracle selbst öffnen und umfangreiche SQL-Kenntnisse haben zu müssen. Da sich bei einem Neustart von ArcView die darin eingefügten Oracle-Tabellen automatisch aktualisieren, kann die Aktualisierung von Oracle zu ArcView als gelungen angesehen werden.

Eine Aktualisierung von ArcView zu Oracle (etwa bei Änderungen der Geometrie wie Neudigitalisierungen) ist deutlich komplizierter. Dazu wird ein Aktualisierungsscript benutzt, welches in Zusammenarbeit zwischen dem Autor mit Herrn Hein [HEIN, mündliche Mitteilung, 2001] entstand (Anhang II). Wird eine Neudigitalisierung einer Radroute (zum Beispiel im Thema „Radwege“) vorgenommen, muss in der Attributtabelle des aktiven Themas in der Spalte „ID“ die Nummer der neuen Linie eingetragen (zum Beispiel: 328) und die Bearbeitung beendet werden. Nun wird die Attributtabelle des Zielthemas (zum Beispiel „Radw_seg“) geöffnet und die Bearbeitung gestartet, ohne jedoch wirklich etwas zu ändern. Dies ist nötig, da sonst bei der Ausführung des Aktualisierungsscriptes die Fehlermeldung „Schreibgeschütztes Feld“ erscheint. Jetzt wird der neu digitalisierte Radweg (im Thema „Radwege“) markiert (über die Attributtabelle möglich) und das Thema, welches ihn beinhaltet, durch Anklicken des View-Fensters und des Themas in der Legende aktiviert. Erst zu diesem Zeitpunkt kann das vor diesen Arbeitsschritten schon geöffnete und kompilierte Aktualisierungsscript (segm_akt.ave) ausgeführt werden. Es fordert den Benutzer auf, das Thema, in dem der neue Radweg enthalten ist („Radwege.shp“), sowie das Thema, in das die neuen Segmente eingefügt werden sollen („Radw_seg.shp“) auszuwählen. Es erscheint eine Meldung, welchen Wert die letzte Segment-ID der alten Radwege hat (zum Beispiel: 6951). Dann werden die Daten prozessiert und die neu digitalisierte Linie in Segmente zerlegt, die eine fortlaufende Nummer (ausgehend vom letzten Wert der alten Segment-ID) bekommen (zum Beispiel: 6952, ...). Die Linien-Ids werden aus der Attributtabelle des aktiven Themas („Radwege“) übernommen und eingefügt (zum Beispiel: 328). Dann kann die Bearbeitung in der noch offenen Attributtabelle des Zielthemas („Radw_seg“) beendet und die Änderungen gespeichert werden. Nun ist die Geometrie im GIS aktualisiert worden. Dieses Verfahren wird bei jeder Aktualisierung der Geometrie angewendet.

Die Änderungen müssen jetzt in die Oracle-Datenbanken eingetragen werden. Einerseits kann dazu der insert-Befehl (Kapitel 4.2.) im SQL-Eingabefenster von Oracle genutzt werden. Das bedeutet einen enormen Aufwand. Andererseits kann auch das Programm Microsoft Query (wie oben beschrieben) eingesetzt werden. Es müssen jedoch auch hier alle Referenztabellen verändert und die neuen Segment-ID eingetragen werden, was einen erheblichen Aufwand bedeutet.

Zusammenfassend gesehen ist eine Aktualisierung in beide Richtungen möglich, aber noch nicht optimal gelöst worden. Hier sind Ansätze für weitere Arbeiten zu finden.

6. Planungsrelevante Zusatzinformationen

Um alle Bereiche der Planung abzudecken, müssen noch zusätzliche planungsrelevante Zusatzinformationen in das GIS eingefügt werden. Dazu gehören Problempunkte von Radwegen, an denen zum Beispiel Brücken über Gewässerläufe fehlen. Ferner sind Gefahrenpunkte einzutragen, so zum Beispiel an besonders kurvenreichen Straßenabschnitten (wenn der Radweg auf der Straße verläuft) und an Kreuzungen mit vielbefahrenen Straßen. Diese Punkte sind wichtige Radwegeinformationen, die mit dem Ziel aufgenommen werden, diese nach der Erkennung systematisch zu entschärfen. Auch sind für die Planung Sehenswürdigkeiten und Ausflugsziele wichtig, da diese mit dem Rad erreichbar sein sollen. Daher ist die Darstellung von planungsrelevanten Zusatzinformationen zum Thema „Radwege“, die der Radwege- und Tourismuskonzeption entnommen und ausgewertet wurden, in einem View durchaus sinnvoll. Eine Verankerung in der Oracle-Datenbank scheint zum derzeitigen Zeitpunkt nicht sinnvoll zu sein, da Sehenswürdigkeiten nur fixe Objekte sind, eine Datenbank jedoch Veränderungen impliziert. Daher werden die Tabellen im ArcView selbst erstellt und verwaltet, wobei sich jedoch kaum Änderungen ergeben werden. Das Konzept für die Zusatzinformationen sieht somit folgendermaßen aus:

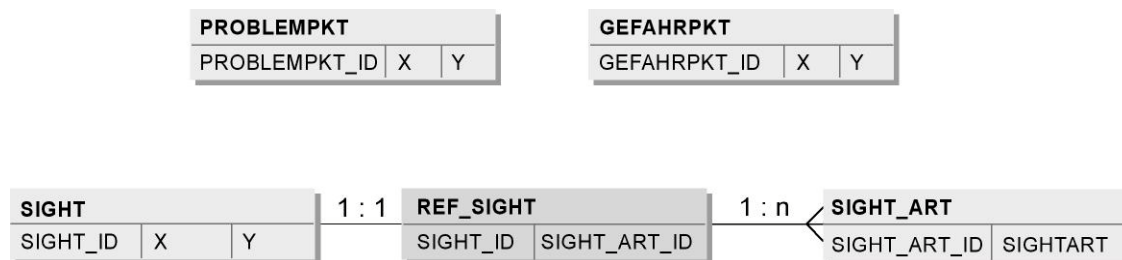


Abbildung 21: Konzept für die Tabellen mit Zusatzinformationen

Die Tabelle der Problempunkte sowie die der Gefahrenpunkte sind als Attributtabelle der Themen „Problempkt“ und „Gefahrpkt“ konzipiert, so dass diese auch nur im ArcView bearbeitet werden müssen. Diese Tabellen sowie die Tabelle „SIGHT“ (auch Attributtabelle) bekommen bei der Digitalisierung in einem jeweils neuen Thema im View „Radwege“ automatisch die Felder „Shape“ und „ID“. Nun wird nach der Digitalisierung der Punkthemen das von

Juan Bohorquez erstellte [BOHORQUEZ, 1999] und vom Autor veränderte Koordinatenscript (Anlage III) ausgeführt, welches in der Attributtabelle die Datensätze im Feld „ID“ durchnummeriert und eine neue Tabelle mit eben diesen Nummern und den Gauß-Krüger-Koordinaten (da das View durch die georeferenzierten Grundlagen in diesem Koordinatensystem projiziert wurde) erstellt.

Jetzt könnten die Tabellen verbunden werden. Aufgrund der einfacheren Handhabung und der geringen Zahl von Gefahrenpunkten, Problempunkten sowie Sehenswürdigkeiten ist es jedoch sinnvoll, anstelle zweier miteinander zu verbindender Tabellen die Koordinaten in der Attributtabelle zu speichern. Dazu wird bei der Aufforderung des Koordinatenscriptes, die neue Tabelle zu speichern, der Name der Attributtabelle (zum Beispiel: gefahrpkt.dbf) des entsprechenden Themas (zum Beispiel: Gefahrpkt) ausgewählt und überschrieben. Dann muss das gesamte Projekt gespeichert und geschlossen werden. Beim erneuten Öffnen des Projekts wird die Attributtabelle nun mit den Spalten „Shape“, „X“, „Y“ sowie der ID-Spalte angezeigt, wobei die Nummerierung der Datensätze noch per Hand zu erfolgen hat (die Datensätze sind schon in der richtigen Reihenfolge). Die Tabelle „SIGHT_ART“ enthält im vorliegenden Projekt folgende Elemente, ist jedoch wie alle Tabellen dieser Arbeit beliebig erweiterbar:

- Tabelle **SIGHT_ART**:

SIGHT_ART_ID	SIGHTART
1	Museum
2	Turm
3	Kirche
4	Windmühle
5	Schloss/Burg
6	Ruine
7	Jugendherberge
8	Gaststätte
9	Fahrradwerkstatt
10	Fahrradverleih
11	Freibad/Badesee
12	Gebäude
13	Denkmal/Denkstein
14	Botanischer Garten/Wildgehege
15	Naturdenkmal/Schutzgebiet

So wird im Projektfenster von ArcView unter dem Menüpunkt „Tabellen“ eine neue Tabelle hinzugefügt, im Menü „Bearbeiten“ der Unterpunkt „Feld hinzufügen“ ausgewählt, woraufhin die neue Spalte nach Name, Typ, Breite und Dezimalstellen definiert werden kann (zum Beispiel: Name: SIGHT_ART_ID, Typ: Zahl, Breite: 10, Dezimalstellen: 0). Die zweite neu zu erstellende Spalte würde zum Beispiel für die Tabelle „SIGHT_ART“ den Namen SIGHTART, den Typ Zeichenfolge und die Breite 50 erhalten. Nun können im Menü „Bearbeiten“ neue Datensätze hinzugefügt und bearbeitet werden. Ist die Bearbeitung abgeschlossen, kann sie im Menü „Tabelle“ beendet und die Tabelle gespeichert werden. Ebenso wird die Referenztabelle „REF_SIGHT“ erstellt. Damit können nun die Tabellen „SIGHT“, „REF_SIGHT“ und „SIGHT_ART“ über ihre ID-Spalten und den Button „Verbinden“ (Kapitel 4.3.) verbunden werden.

7. Grafische Gestaltung der Radwegekarte

Um die Radwege und Sehenswürdigkeiten sowie die Gefahren- und Problempunkte grafisch von den mehrfarbigen Grundlagenkarten abzuheben und dem Benutzer die Gelegenheit zu geben, die wichtigsten Radwegmerkmale auch optisch ohne vorherige Abfrage zu erkennen, wurden die nun vorliegenden Themen „Radw_seg“ (segmentierte Radwege, die mit den Oracle-Tabellen verbunden sind), „Sight“ (Sehenswürdigkeiten), „Gefahrpkt“ (Gefahrenpunkte) und „Problempkt“ (Problempunkte) grafisch gestaltet und an den Maßstab grafisch angepasst.

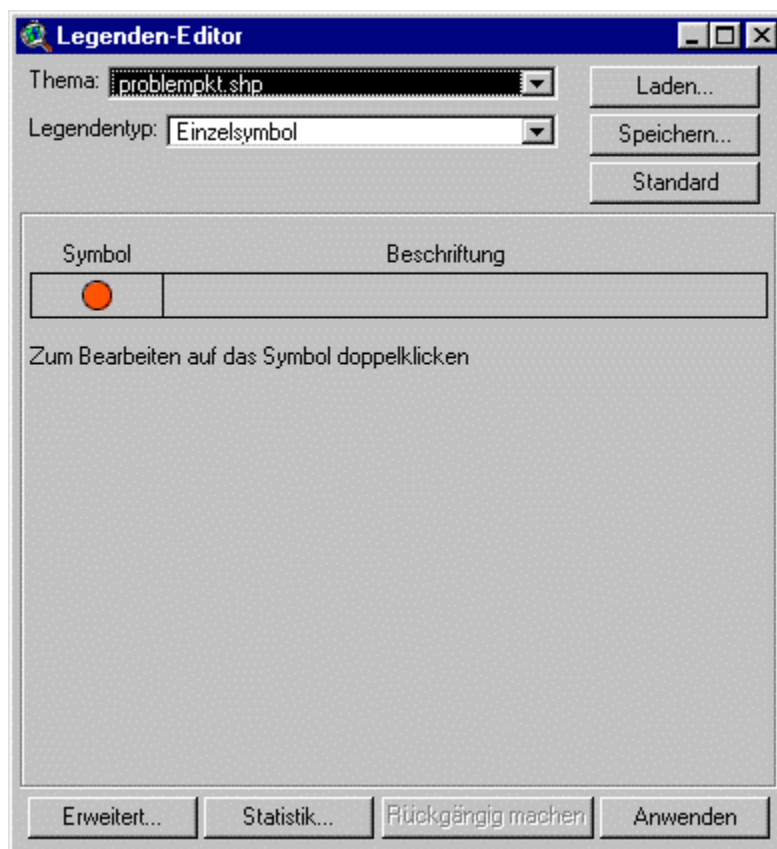


Abbildung 22: Der Legenden-Editor

Dazu wird die Legende des zu bearbeitenden Themas mit einem Doppelklick auf das Legendensymbol oder über das Menü „Thema“ Unterpunkt „Legende bearbeiten...“ geöffnet. Es erscheint der Legenden-Editor (Abbildung 22). Hier können die Symbole für die Problem- und Gefahrenpunkte geändert werden, indem auf das angezeigte Symbol doppelgeklickt wird. Dadurch öffnet sich das

Punktsymbolfenster (Abbildung 23), in dem die Symbolart und die Größe des Symbols eingestellt werden können. Die Farbe wird über den fünften Button von links ausgewählt, neue Legendenpaletten außer der Standardpalette können im Menü des sechsten Buttons von links geladen werden.

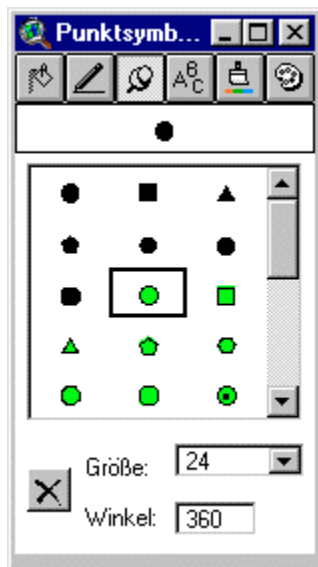


Abbildung 23: Das Punktsymbol-Fenster

Ist die Form, Farbe und Größe des Symbols ausgewählt, werden diese Änderungen im View angewendet, indem im Legenden-Editor der Button „Anwenden“ (Abbildung 22) benutzt wird. Das Symbol stellt sich allerdings immer in der eingestellten Größe, unabhängig vom benutzten Maßstab zur Darstellung der Themen dar. Um es an den Maßstab anzupassen, wird im Legenden-Editor (Abbildung 22) das Menü „Erweitert...“ aufgerufen (Abbildung 24). Dort kann der Referenzmaßstab bestimmt werden, in welchem das Symbol die eingestellte Größe haben soll. Die Symbolgröße korreliert nun mit dem veränderten Maßstab.



Abbildung 24: Die erweiterten Optionen zur Skalierung eines Symbols

Während für die Themen „Problempkt“ und „Gefahrpkt“ im Legenden-Editor der Legendentyp „Einzelsymbol“ eingestellt ist und alle Punkte die gleiche Signaturierung haben, wird für die Themen „Radw_seg“ und „Sight“ der Legendentyp „Einzelwert“ ausgewählt. Das hat den Vorteil, dass innerhalb der Themen noch Klassifizierungen nach bestimmten Spalten der Attributtabelle (also nach Attributen, zum Beispiel nach der Art der Sehenswürdigkeiten) vorgenommen werden können. Das Klassifizierungsattribut (eine Spalte der Attributtabelle, nach der die Daten zu klassifizieren sind) wird im Wertefeld ausgesucht. Dadurch zeigt die darunter liegende Tabelle die Werte an, nach denen die Daten nun unterteilt sind. Mit einem Doppelklick auf das Symbol eines bestimmten Wertes kann dieses wie oben beschrieben verändert werden. Durch geschickte Auswahl ist eine aussagekräftige Bildschirmkarte generierbar. Auch hier kann wieder die Skalierbarkeit der Punktsymbole und der Linienbreite der Radwege über den Button „Erweitert...“ eingestellt werden.

Die Sehenswürdigkeiten wurden mit der Spalte „Sightart“ klassifiziert, die mit der Attributtabelle des Themas „Sight“ verbunden ist. Dadurch entstanden fünfzehn Klassen, von denen zehn bisher verwendet und daher mit eigenen Symbolen versehen worden sind.

Die Merkmale der Radwege wurden mit in die Attributtabelle des Themas „Radw_seg“ aufgenommen. Da die Klassifizierung der Daten eines Themas nur nach einer Spalte (das heißt nach einem Merkmal) erfolgen kann, war eine Unterteilung der Daten nach der Spalte „Nutzung“ die sinnvollste Möglichkeit. Radwege, die weder eine touristische noch eine Alltagsnutzung aufwiesen, mussten notwendigerweise noch in Planung sein. Daher konnte im Legenden-Editor der mit der Zahl „0“ gekennzeichnete Button eingesetzt werden, der eine

dritte Klasse „Keine Daten“ hinzufügte, die in der Spalte „Beschriftung“ im Legenden-Editor in „Planung“ umbenannt wurde. Die Abbildung 25 zeigt das Endergebnis der grafischen Gestaltung (ohne die im Hintergrund liegenden topographischen Karten).

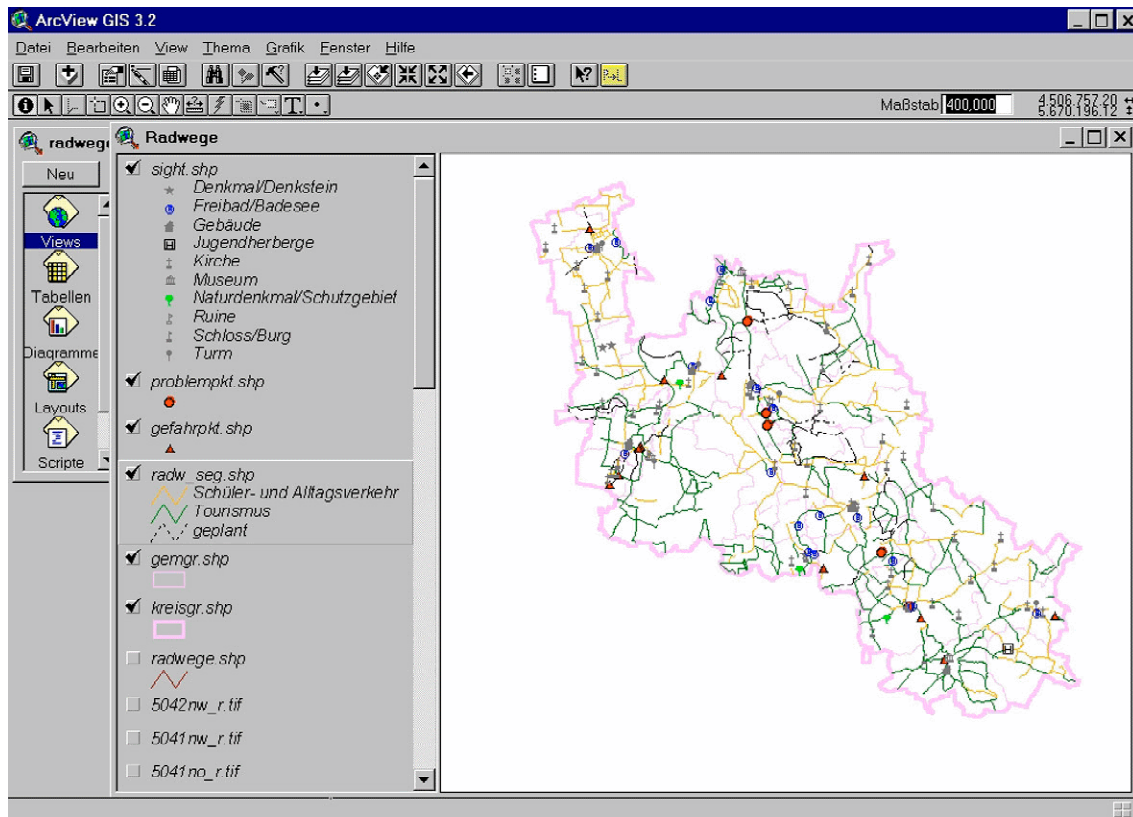


Abbildung 25: Das Endergebnis (noch ohne Symbol-Skalierung)

8. Zusammenfassung

Ziel dieser Studienarbeit, deren Aufgabenstellung selbst gewählt wurde, war die Entwicklung einer Konzeption für eine Radwege-Datenbank im Oracle 7.3.4 sowie die Erstellung von Radwege-Themen für den gesamten Landkreis Leipziger Land im GIS ArcView 3.2. Eine wesentliche Grundlage dieses Projektes stellte die Verbindung der beiden Programme miteinander dar, um raumbezogene Abfragen (auch themenübergreifend) zu Radwegen nach Sachdaten aus der Datenbank ausführen zu können.

Zur Sicherung einer optimalen Datenbasis mussten vor der Digitalisierung der Radwege des Landkreises Leipziger Land im GIS ArcView umfangreiche Vorüberlegungen zu den Datenbankinhalten, der Datenstruktur und der Art und Weise der Datenerfassung angestellt und umgesetzt werden. Die darauf basierende Segmentierung der Radwege erfolgte über ein Avenue-Script.

Der Aufbau der Radwege-Datenbank erforderte die Erstellung eines Konzeptes, in dem alle Grundvoraussetzungen einer relationalen Datenbank (Relationen, dritte Normalform usw.) implementiert sind. ArcView kann auf die erstellten Oracle-Daten über eine SQL-Verbindung zugreifen, da diese Verbindung über die ODBC-Schnittstelle in der GIS-Software enthalten ist. So können Abfragen an die Datenbank entweder ganze Tabellen oder nur bestimmte Datensätze aus diesen Tabellen an ArcView zurückliefern. Wird eine Tabelle der Datenbank editiert, ist eine Aktualisierung in dieser Richtung automatisch möglich, da in der ArcView-Projektdatei nur die SQL-Abfrage an die Datenbank gespeichert wird und bei jedem Neustart der GIS-Software die abgefragten Tabellen sowie deren Verbindungen erneuert werden.

Die Aktualisierung in Gegenrichtung (von ArcView zu Oracle) ist problematisch, aber über externe Programme möglich. Eine mit einer Geometrieänderung im ArcView verbundene Aktualisierung der Segmentierung kann über ein vom Autor erstelltes Avenue-Script problemlos erfolgen. Die automatische Übernahme dieser Änderungen in das Datenbankmanagementsystem ist derzeit nicht ohne einen extrem hohen Programmieraufwand möglich. Eine Laufendhaltung der Datenbank erfordert entweder die Anwendung von SQL-Befehlen im Oracle oder die Editierung im externen Programm Microsoft Query.

Ungeübte Nutzer ohne SQL-Kenntnisse sollten die bessere grafische Benutzeroberfläche von Microsoft Query nutzen.

Eine Erhöhung der Effizienz der Verbindung zwischen der GIS-Software und dem Datenbankmanagementsystem, wenn auch nicht ihrer Stabilität, ist durch ein auf dem Betriebssystem Windows basierendes Datenbankmanagementsystem möglich. Hier wäre Access von Microsoft zu empfehlen, welches nicht nur Vereinfachungen bei der Aktualisierung des GIS ermöglicht, sondern auch eine bedienerfreundlichere Oberfläche besitzt. Auch eine neuere Version von Oracle würde deutliche Vorteile bringen. So können längerfristig alle Datenbanken mit dem Utility „WebDB“ ins Internet oder Intranet gestellt werden und würden damit auch Anwendern außerhalb des Landratsamtes die Möglichkeit des Zugriffs auf die Daten geben.

Prinzipiell sind die erstellten GIS-Themen sowie die aufgebaute Datenbank als nutzbringende und sinnvolle Erweiterungen des schon bestehenden GIS des Landratsamtes Leipziger Land anzusehen. Eine zeitgemäße digitale Planung der Radwege ist mit dieser Arbeit möglich geworden und sollte in Zukunft noch ausgebaut werden.

HINWEIS:

Zum Zeitpunkt der Bearbeitung lagen zwar aktuelle Daten der Gemeinde Heuersdorf vor, waren aber für diese Arbeit noch nicht verfügbar. Die Gemeinde Heuersdorf muss daher noch als 25. Gemeinde des Landkreises Leipziger Land der Gemeindeliste hinzugefügt werden.

9. Danksagung

Der Autor möchte folgenden Mitarbeitern des Landratsamtes Leipziger Land für die Bereitstellung der Daten und Arbeitsmaterialien sowie für Hintergrundinformationen und Hinweise danken: Herrn Fehse (GIS-Betreuer des Bauplanungs- und Wirtschaftsförderungsamtes), Herrn Müller (Datenbank-Betreuer), Frau Sommer (Sachgebietsleiterin des Bauplanungs- und Wirtschaftsförderungsamtes), Herrn Mai, Frau Hagen sowie Frau Kuhne (Mitarbeiter des Bauplanungs- und Wirtschaftsförderungsamtes).

Mein besonderer Dank gilt Herrn Hein (Institut für Länderkunde Leipzig) für die hilfreiche Unterstützung und Beratung bei Problemen mit Avenue und ArcView.

10. Literatur- und Quellenverzeichnis

- Achilles, Dr. A.: SQL: Standardisierte Datenbanksprache vom PC bis zum Mainframe. R. Oldenbourg Verlag, München/Wien 1991.
- Bohorquez, J.: Koordinatenscript unter www.esri.com - ArcScripts, 1999 (Zugriff: 02.02.2001).
- ESRI: ArcView-Handbuch, 1999.
- ESRI: Avenue-Handbuch, 1999.
- ESRI-Kundenbetreuung: fernmündliche Mitteilung, Leipzig 11.04.2001.
- Fehse, J.: mündliche Mitteilung, Leipzig 2000.
- Gillenson, M.: Datenbank-Konzepte: Datenbanken – Netzwerke – Expertensysteme. Sybex-Verlag GmbH, Düsseldorf 1990.
- Grüner, M.: Möglichkeiten eines objektbezogenen Zugriffs von ArcView auf eine Oracle-Datenbank über den Geodatenserver ArcSDE. Diplomarbeit, Technische Universität, Dresden 2000.
- Hein, U.: mündliche Mitteilung, Leipzig 2001.
- Heinrich, L.: Einführung eines Geographischen Informationssystems im Landratsamt Leipziger Land. Diplomarbeit, Hochschule für Technik und Wirtschaft, Dresden 2000.
- Kaufmann, K.-U.: fernmündliche Mitteilung, 11.04.2001.
- Lockman, D.: Oracle 8 – Datenbankentwicklung in 21 Tagen. SAMS, Haar bei München 1998.

- Martin, E.: Segmentierscript unter www.esri.com - ArcScripts, 1998 (Zugriff: 02.02.2001).
- McCullough-Dieter, C.: Oracle 8i für Dummies. MITP-Verlag, Bonn 1999.
- Sommer, R.: mündliche Mitteilung, Leipzig 2001.
- Ufer, F.: Konzeption eines Radwanderatlases für Dresden und Umgebung. Großer Beleg, Technische Universität, Dresden 1996.

11. Abbildungsverzeichnis

	Seite
• Abbildung 1:	 Titelblatt
• Abbildung 2:	Das ArcView - Projektfenster 8
• Abbildung 3:	Das Oracle SQL*Plus - Eingabefenster 12
• Abbildung 4:	Die Tabelle RADWEG1 verstößt gegen die erste Normalform. 15
• Abbildung 5:	Die Tabellen RADWEG1 und PERSONENDATEN erfüllen die erste Normalform. 15
• Abbildung 6:	Die Tabelle RADWEG2 verstößt gegen die zweite Normalform. 16
• Abbildung 7:	Beide Tabellen entsprechen der zweiten Nor- malform. 17
• Abbildung 8:	Die Tabelle RADWEG3 verstößt gegen die dritte Normalform. 17
• Abbildung 9:	Die Tabellen in dritter Normalform 18
• Abbildung 10:	Aktivierung der allgemeinen Einpassung vor der Digitalisierung 24
• Abbildung 11:	Der Button zum Kompilieren eines Scriptes 25
• Abbildung 12:	Der Button zum Ausführen eines Scriptes 25
• Abbildung 13:	Das physische Radwege-Konzept der Oracle-Datenbank 28
• Abbildung 14:	Herstellung einer SQL-Verbindung im ArcView 40
• Abbildung 15:	Abfrageergebnis nach erfolgreicher Verbin- dung aller Tabellen 42
• Abbildung 16:	Auswählen der zu editierenden Datenbank im Microsoft Query 44
• Abbildung 17:	Einloggen in die Datenbank 44
• Abbildung 18:	Einfügen der zu editierenden Tabelle in Micro- soft Query 45
• Abbildung 19:	Anzeigen der zu editierenden Tabelle 45
• Abbildung 20:	Anwählen von „Bearbeiten ermöglichen“

	zur Tabelleneditierung	46
• Abbildung 21:	Konzept für die Tabellen mit Zusatzin- formationen	49
• Abbildung 22:	Der Legenden-Editor	52
• Abbildung 23:	Das Punktsymbol-Fenster	53
• Abbildung 24:	Die erweiterten Optionen zur Skalierung eines Symbols	54
• Abbildung 25:	Das Endergebnis (noch ohne Symbol- skalierung)	55
• Abbildung 26:	Die Symbole der bisher vorhandenen Sehenswürdigkeiten	Anhang V.

Anlagen

- I. Segmentierscript
- II. Segmentierscript zur Aktualisierung
- III. Koordinatenscript
- IV. Controlfile
- V. Anleitung zur Aktualisierung

I. Segmentierscript

' Name: segment
' Titel: Zerlegung von Linien in Liniensegmente
' Beschreibung: Zerlegt Linien und Polygone in ihre Segmente und erzeugt ein neues
' Shapefile. Wenn keine speziellen Merkmale ausgewählt sind, werden
' alle Daten prozessiert.
' Voraussetzungen: Ein View muss aktiv sein, ein Polyline-Thema muss aktiv sein.
' funktioniert in ArcView 3.x
' Autor: Eugene Martin, 1998, ewmartin@u.washington.edu'
' Modifizierung: Olaf Schnabel, 2001, oschnabel@gmx.de

```
theView = av.GetActiveDoc  
thmThemeIn = theView.GetActiveThemes.Get(0)
```

```
' Erzeuge das Ausgabe-Shapefile...  
fnDefault = FileName.Make("$HOME").MakeTmp("shape","shp")  
fnOutput = FileDialog.Put( fnDefault,"*.shp","Ausgabe-Shapefile" )  
if (fnOutput = nil) then exit end  
fnOutput.SetExtension("shp")  
ftbOutput = FTab.MakeNew( fnOutput, PolyLine )  
ftbOutput.AddFields({Field.Make("SEGMENT_ID", #FIELD_LONG, 8, 0)})  
ftbOutput.AddFields({Field.Make("LINE", #FIELD_LONG, 8, 0)})
```

```
' Kontrolliere, ob Linien ausgewählt sind, ansonsten benutze das gesamte Thema...  
if (thmThemeIn.GetFTab.GetSelection.Count > 0) then  
    lineToProcess = thmThemeIn.GetFTab.GetSelection  
else  
    lineToProcess = thmThemeIn.GetFTab  
end
```

```
' Suche im aktiven Thema und im Ausgabe-Thema die Felder...  
fldShapeIn = thmThemeIn.GetFTab.FindField("shape")  
fldShapeOut = ftbOutput.FindField("shape")  
fldlineIDin = thmThemeIn.GetFTab.FindField("ID")  
fldsegmentIDOut = ftbOutput.FindField("SEGMENT_ID")  
fldlineIDOut = ftbOutput.FindField("LINE")
```

```

' Beginne die Prozessierung...
nCount=0
for each r in lineToProcess
    linein = thmThemeIn.GetFTab.ReturnValue(fldShapeIn,r)
    lineID = thmThemeIn.GetFTab.ReturnValue(fldlineIDin,r)
    mpoints=linein.asmultipoint
    linelist=linein.split(mpoints)
    for each one in linelist
        nCount=nCount+1
        nRecNew = ftbOutput.AddRecord
        ftbOutput.SetValue(fldShapeOut,nRecNew,one)
        ftbOutput.SetValue(fldlineIDOut,nRecNew,lineID)
        ftbOutput.SetValue(fldsegmentIDOut,nRecNew,nCount)
    end
end

if (MsgBox.YesNo("Shapefile als Thema dem View hinzufügen?",
    "Zerlegung der Polylines in Segmente", true).Not) then
    exit
end

' Erstelle eine Liste der Views und erlaube dem Benutzer, welchem View er das neue
' Thema hinzufügen soll...
lstViews = {}
for each d in av.GetProject.GetDocs
    if (d.Is(View)) then
        lstViews.Add( d )
    end
end

lstViews.Add("<New View>")
vweAddTo = MsgBox.ListAsString( lstViews,"Füge Thema hinzu zu:",
    "Zerlegung der Polylines in Segmente" )

```

' Bekomme das spezielle View, erstelle das Thema und füge es hinzu...

if (vweAddTo <> nil) then

 if (vweAddTo = "<New View>") then

 vweAddTo = View.Make

 vweAddTo.GetWin.Open

 end

 thmNew = FTheme.Make(ftbOutput)

 vweAddTo.AddTheme(thmNew)

 vweAddTo.GetWin.Activate

end

II. Segmentierscript zur Aktualisierung

```
' Name: segm_akt
' Titel: Zerlegung von Linien in Liniensegmente
' Beschreibung: Zerlegt Linien und Polygone in ihre Segmente und aktualisiert
' ein vorhandenes Shapefile.
' Voraussetzungen: Ein View und ein Thema müssen aktiv sein, die neuen Linien
' müssen selektiert sein.
' funktioniert in ArcView 3.x
' Autor: Olaf Schnabel, 2001, oschnabel@gmx.de

' altes Segment-Thema und neues Digitalisierthema wählen...
theView = av.GetActiveDoc
theActThemes = theView.GetThemes
if (nil = theActThemes) then
    exit
end

if (theActThemes.Count < 2) then
    MsgBox.Error("Es müssen wenigstens zwei Themen im View sein!","Falsche
    Themenanzahl")
    exit
end

ThemeList = theView.GetThemes
FirstTheme = MsgBox.ChoiceAsString(ThemeList,"Welches Thema enthält die alten
    Radwegesegmente?","Erstes Thema auswählen")
if (FirstTheme = nil) then
    exit
end
SecondTheme=MsgBox.ChoiceAsString(ThemeList,"Welches Thema enthält die neu
    digitalisierten Linien?","Zweites Thema auswählen")
if (SecondTheme = nil) then
    exit
end

' Attributtabellen (FTabs) greifen...
FirstFtab = FirstTheme.GetFTab
```

SecondFtab = SecondTheme.GetFTab

' initialisieren...

theSegIDField=FirstFTab.FindField("SEGMENT_ID")

oCount = 0

' bislang größte Segment-ID feststellen...

for each i in FirstFTab

 SegIDWert=FirstFTab.ReturnValue(theSegIDField, i)

 if (SegIDWert > oCount) then

 oCount=SegIDWert

 end

end

' und diese als Text ausgeben...

MsgBox.Info("Größe auftretende alte Segment-ID:"+oCount.AsString, "Maximum")

' initialisieren...

theShapeField = FirstTheme.GetFTab.FindField("Shape")

theLineField = FirstTheme.GetFTab.FindField("LINE")

' Kontrolliere, ob neu digitalisierte Linien im zweiten Thema ausgewählt sind, ansonsten benutze die gesamte Thematabelle...

if (SecondTheme.GetFTab.GetSelection.Count > 0) then

 LineToProcess = SecondTheme.GetFTab.GetSelection

else

 LineToProcess = SecondTheme.GetFTab

end

' Prozessierung der neuen Daten...

fldSecondShape = SecondTheme.GetFTab.FindField("shape")

fldSecondID = SecondTheme.GetFTab.FindField("ID")

' Hänge neue Daten an...

nCount=oCount

for each r in LineToProcess

 lineIn = SecondTheme.GetFTab.ReturnValue(fldSecondShape,r)

 lineID = SecondTheme.GetFTab.ReturnValue(fldSecondID,r)

```

mpoints=lineIn.AsMultiPoint
linelist=lineIn.split(mpoints)
for each one in linelist
    nCount=nCount+1
    nRecNew = FirstTheme.GetFTab.AddRecord
    FirstFTab.SetValue(theShapeField,nRecNew,one)
    FirstFTab.SetValue(theLineField,nRecNew,lineID)
    FirstFTab.SetValue(theSegIDField,nRecNew,nCount)
end
end

' Lösche die neu eingefügten Datensätze, die Segment_id = 0 haben...
for each t in FirstTheme.GetFTab
    if (theSegIDField = 0) then
        FirstFTab.RemoveRecord(t)
    end
end
end

```

III. Koordinatenscript

' Name: koord
' Titel: Anhängen von 2 Koordinatenspalten X, Y
' Beschreibung: Koordinatenermittlung von Punkten und Linien (ID, x, y)
' Voraussetzungen: ein Thema muss markiert sein
' Autor: Juan Bohorquez, 1999, jbohorqu@prosis.com
' Modifizierung: Olaf Schnabel, 2001, oschnabel@gmx.de

```
myView = av.GetActiveDoc
myTheme = myView.GetActiveThemes.Get(0)
myTheme.ClearSelection
myTable = myTheme.GetFtab
FeldID = myTable.FindField("ID")

' Wenn ID-Feld von Ausgangstabelle nicht existiert, erstelle es...
If (FeldID = nil) then
    myTable.SetEditable(true)
    FeldID = Field.Make("ID",#FIELD_DOUBLE,4,0)
    myTable.AddFields({FeldID})
    myTable.SetEditable(false)
end

Seleccion = myTable.GetSelection
zahl = Seleccion.GetSize
myTable.SetEditable(true)
Feld = myTable.FindField("ID")

' Selektierte Datensätze durchnummerieren...
zahl = zahl - 1
for each i in 0..zahl
    o = i + 1
    myTable.SetValue(Feld,i,o)
end

' Feld Shape merken...
myTable.SetEditable(false)
myshp = myTable.FindField("Shape")
```

```
ListFelds = myTable.GetFields
```

```
myArchiv = FileDialog.Put("koord.dbf".AsFileName,"*. *", "Speichern")
```

```
' Neue Attributtabelle erstellen...
```

```
avtab = Vtab.MakeNew(myArchiv,dbase)
```

```
NewTable = Table.Make(avtab)
```

```
Feld1 = Field.Make("SIGHT_ID",#FIELD_DOUBLE,10,0)
```

```
Feld2 = Field.Make("X",#FIELD_FLOAT,15,0)
```

```
Feld3 = Field.Make("Y",#FIELD_FLOAT,15,0)
```

```
avtab.AddFields({Feld1,Feld2,Feld3})
```

```
av.ShowStopButton
```

```
' Koordinatenabfrage und Durchnummerierung der Datensätze...
```

```
for each p in myTable
```

```
    poly1 = myTable.ReturnValue(myshp,p).AsMultiPoint
```

```
    my = poly1.AsString
```

```
    Punktzahl = my.Extract(1).AsNumber
```

```
    WertID = myTable.ReturnValueNumber(Feld,p)
```

```
    for each Num in 2 .. (Punktzahl + 1)
```

```
        rec = avtab.AddRecord
```

```
        xy = my.Extract(Num)
```

```
        listaxy = xy.AsTokens(",")
```

```
        WertX = listaxy.Get(0).AsNumber
```

```
        WertY = listaxy.Get(1).AsNumber
```

```
        avtab.SetValue(Feld1,rec,WertID)
```

```
        avtab.SetValue(Feld2,rec,WertX)
```

```
        avtab.SetValue(Feld3,rec,WertY)
```

```
    end
```

```
    progreso = (p/(myTable.GetNumRecords)) * 100
```

```
    demora = av.SetStatus(progreso)
```

```
    if (not demora) then
```

```
        break
```

```
    end
```

```
end
```

```
avtab.SetEditable(False)
```

```
av.ClearStatus
```

IV. Controlfile

-- Controlfile rad.ctl zum Einladen der Datei radw_seg.prn in eine Oracle-Tabelle

load data

infile 'radw_seg.prn'

into table SEGMENT

(SEGMENT_ID position (01:11) integer external,

LINE position (12:22) integer external)

V. Anleitung zur Aktualisierung

A) Änderung vorhandener Merkmale in existierenden Radwegen (zum Beispiel: Tourismusnutzung wird zu Alltagsnutzung)

- ArcView: betreffende Segmente heraussuchen

Segment-ID unbekannt:

1. Thema „Radw_seg“ aktivieren (Mausklick auf Thema in Legende)
2. Button  aktivieren
3. auf Radwegsegment im View klicken und Segment-ID ablesen (weiter bei 7.)

bei Anzeige mehrerer Segment-ID:

4. Auswahl aufheben mit Button 
5. Maßstab vergrößern
6. zurück zu 1.

Segment-ID bekannt:

1. Thema „Radw_seg“ aktivieren (Mausklick auf Thema in Legende)
2. Button  (Abfrage-Manager) aktivieren
3. Doppelklick auf „Segment_id“
4. Doppelklick auf „=“
5. Segment-ID eingeben
6. „Neue Auswahl“ drücken (im View wird das ausgewählte Segment markiert dargestellt; auch mit mehreren Segmenten möglich)

- Oracle:

```
7. update <Referenztabellenname>
   set <Spaltenname der zweiten Spalte> = <Neuer Wert>
   where
     <Bedingung>;
```

zum Beispiel: Segment 1200 ändert sich von Tourismusnutzung (NUTZUNG_ID = 2) zu Alltagsnutzung (NUTZUNG_ID = 1):

```
update REF_NUTZUNG
set NUTZUNG_ID = 1
where
  SEGMENT_ID = 1200;
```

- ODER MS Query statt Oracle:

7. Excel öffnen
8. Daten/Daten importieren (MS Query wird aufgerufen)
9. im MS Query: Datenquelle wählen: GIS
10. wenn nicht angezeigt, dann Button „Andere“ aktivieren und GIS auswählen
11. Button „Verwenden“ drücken
12. Einloggen in die Oracle-Datenbank (User_name + Passwort)
13. Tabelle hinzufügen: Tabelle auswählen und Button „Hinzufügen“ aktivieren
14. Doppelklick auf Sternchen in Tabelle (alle Datensätze sichtbar)
15. Datensätze/Bearbeiten ermöglichen (ein Klick)
16. entsprechende Segment-ID suchen, in zweiter Spalte Wert (ID) ändern
17. ist Wert (ID) unbekannt, siehe Merkmalstabellen im Kapitel 4.1.
18. nach Bearbeitung: Datensätze/Bearbeiten ermöglichen ausschalten
19. Datei/Speichern unter
20. Datenquelle (Datenbank gis) auswählen
21. Tabellennamen eingeben (zum Beispiel: REF_NUTZUNG)
22. dabei Groß- und Kleinschreibung beachten
23. Button „Speichern“
24. Meldung „Tabelle besteht bereits. Ändern?“ bejahen
25. Fehlermeldungen mit „Ok“ wegklicken (ist trotzdem gespeichert!)
26. MS Query und Excel schließen, ohne die Abfragen zu speichern

- ArcView: neu starten (Änderungen werden wirksam)

B) Hinzufügen neuer Merkmale in vorhandene Radwege (zum Beispiel: Belag ändert sich von Beton zu bisher noch nicht vorhandenem Ziegelstein)

- ArcView: betreffende Segmente heraussuchen (siehe Punkt A, 1. bis 6.)

- Oracle: 7. insert into <Tabellenname>
 (<Spaltenname der ersten Spalte>, <Spaltenname der zweiten Spalte>)
 values
 (<Wert der ersten Spalte>, <Wert der zweiten Spalte>);

zum Beispiel: Merkmal Ziegelstein kommt hinzu:

```
insert into BELAG
(BELAG_ID, BELAGSART)
values
(7, 'Ziegelstein');
```

```
8. update <Referenztabellenname>
set <Spaltenname der zweiten Spalte> = <Neuer Wert>
where
<Bedingung>;
```

zum Beispiel: Segment 1300 ändert sich von Beton (Belagsart = 6) zu Ziegelstein (Belagsart = 7):

```
update REF_BELAG
set BELAG_ID = 7
where
SEGMENT_ID = 1300;
```

- ODER MS Query statt Oracle:

7. wie Punkt A, 7. bis 15.
8. in zweiter Spalte letzten Wert anklicken
9. Tabulatortaste drücken (neue Zeile erscheint)
10. neue Werte für beide Spalten angeben (zum Beispiel: 7, Ziegelstein)
11. wie Punkt A, 18. bis 25.
12. Datei/Neue Abfrage
13. bei eventuellen Fragen: „Soll Abfrage gespeichert werden?“ verneinen

14. wie Punkt A, 9. bis 26.

- ArcView: neu starten (Änderungen werden wirksam)

C) Hinzufügen einer neuen Merkmalstabelle (zum Beispiel: Neigung)

- Oracle:

1. create table <Tabellenname>

(<Spaltenname der ersten Spalte> number (10, 0) not null primary key,
<Spaltenname der zweiten Spalte> <Datentyp der zweiten Spalte>);

zum Beispiel: die Tabelle Neigung mit den Spalten Neigung-ID und Grad:

```
create table NEIGUNG  
(NEIGUNG_ID number(10,0) not null primary key,  
GRAD varchar2 (50));
```

2. insert into <Tabellenname>

(<Spaltenname der ersten Spalte>, <Spaltenname der zweiten Spalte>)
values
(<Wert der ersten Spalte>, <Wert der zweiten Spalte>);

zum Beispiel: in der Tabelle Neigung der Grad „stark“:

```
insert into NEIGUNG  
(NEIGUNG_ID, GRAD)  
values  
(1, 'stark');
```

3. create table <Referenztabellenname>

as select SEGMENT_ID from SEGMENT
where SEGMENT_ID > 0;

zum Beispiel: die Referenztabelle Ref_Neigung erstellen:

```
create table REF_NEIGUNG  
as select SEGMENT_ID from SEGMENT  
where SEGMENT_ID > 0;
```

4. alter table <Referenztabellenname> add

primary key (SEGMENT_ID);

zum Beispiel: Primärschlüssel für die Tabelle Ref_Neigung hinzufügen:

```
alter table REF_NEIGUNG add  
primary key (SEGMENT_ID);
```

5. alter table <Referenztabellenname> add
<Spaltenname der zweiten Spalte> number (10, 0) [<default> <Defaultwert>];

zum Beispiel: Spalte Neigung-ID zur Referenztabelle Ref_Neigung hinzufügen:

```
alter table REF_NEIGUNG add  
NEIGUNG_ID number (10, 0);
```

6. alter table <Referenztabellenname> add
foreign key (<Spaltenname der zweiten Spalte>) references <Name der
Merkmalstabelle> (<Spaltenname der Merkmalstabelle mit Primärschlüssel>);

zum Beispiel: Fremdschlüssel zur Referenztabelle Ref_Neigung hinzufügen:

```
alter table REF_NEIGUNG add  
foreign key (NEIGUNG_ID) references NEIGUNG (NEIGUNG_ID);
```

7. update <Referenztabellenname>
set <Spaltenname der zweiten Spalte> = <Neuer Wert>
where
<Bedingung>;

zum Beispiel: Segmente 1300 bis 2000, Segment 2010 und Segmente 2200 bis 2300
sollen starke Neigung (Neigung-ID = 1) erhalten:

```
update REF_NEIGUNG  
set NEIGUNG_ID = 1  
where  
SEGMENT_ID between 1300 and 2000  
or SEGMENT_ID = 2010  
or SEGMENT_ID between 2200 and 2300;
```

- ODER MS Query statt Oracle:

7. wie Punkt A, 7. bis 26.

- ArcView: Verbindung von Oracle-Tabelle mit ArcView-Attributtabelle (zum Beispiel: Tabellen
Ref_Neigung und Neigung in ArcView mit Attributtabelle von Thema „Radw_seg“
verbinden):

8. Projekt/SQL-Verbindung (Projektfenster muss aktiv sein)
9. unter Verbindung „gis“ suchen (alle Tabellen werden angezeigt)
10. Auswahl der gesuchten Referenztabelle (Doppelklick) (zum Beispiel: REF_NEIGUNG)
11. alle Spalten wählen (Doppelklick auf Sternchen)
12. Name der Ausgabetabelle eingeben (zum Beispiel: REF_NEIGUNG)
13. Button „Abfragen“ drücken
14. 8. bis 13. für Merkmalstabelle wiederholen
15. im Projektfenster: Tabellen/Attributtabelle von Thema öffnen (zum Beispiel: Attributtabelle von „Radw_seg“) oder im View (zum Beispiel „Radwege“) Button  drücken
16. im Projektfenster: Tabellen/Merkmalstabelle und Referenztabelle öffnen (zum Beispiel: REF_NEIGUNG, NEIGUNG)
17. in Referenztabelle Spaltenkopf „SEGMENT_ID“ markieren
18. in Attributtabelle Spaltenkopf „SEGMENT_ID“ markieren
19. Button  (Verbinden) drücken (Referenztabellenspalten werden an Attributtabelle angefügt)
20. in Merkmalstabelle Spaltenkopf (ID) markieren (zum Beispiel: NEIGUNG_ID)
21. in Attributtabelle die neu angefügte ID-Spalte markieren (zum Beispiel: NEIGUNG_ID)
22. Button  (Verbinden) drücken (Merkmalstabellenspalten werden an Attributtabelle angefügt)
23. Tabelle/Eigenschaften: alle ID-Felder unsichtbar machen (Haken wegklicken)

D) Löschung von Segmenten eines Radweges (Geometrieänderungen - zum Beispiel: Elsterradroute)

- ArcView: Radweg löschen:

1. Thema „Radw_seg“ aktivieren
2. Thema/Bearbeitung starten
3. Radweg markieren oder im Abfrage-Manager: Radwegname = „Elsterradroute“ eingeben und „Neue Auswahl“ drücken
4. Entfernen-Taste drücken

- Oracle: Merkmale der entsprechenden Segmente löschen (nicht unbedingt notwendig!):

5. delete from <Tabellenname>
where
<Bedingung>;

zum Beispiel: Segmente 700 bis 1000 fallen weg:

```
delete from SEGMENT  
where  
SEGMENT_ID between 700 and 1000;
```

6. 5. für alle Referenztabellen wiederholen

- ODER MS Query statt Oracle:

5. wie Punkt A, 7. bis 15.
6. zu löschende Datensätze markieren und löschen
7. wie Punkt A, 18. bis 26.
8. 5. bis 7. für alle Referenztabellen wiederholen

E) Hinzufügen eines neuen Radweges (Geometrieänderungen - zum Beispiel: bei Streckenänderungen und Neubau)

- ArcView: Digitalisierung

1. Thema „Radwege“ aktivieren und Thema/Bearbeitung starten
2. Button  benutzen und neuen Radweg im Maßstab 1 : 10.000 digitalisieren
3. Thema/Bearbeitung beenden
4. Änderungen speichern (neuer Radweg bleibt markiert)
5. Attributtabelle des Themas „Radw_seg“ öffnen
6. Tabelle/Alle Verbindungen lösen
7. Tabelle/Bearbeitung starten
8. letzte Nummer in der Spalte „Line“ merken (zum Beispiel: 327)
9. Attributtabelle des Themas „Radwege“ öffnen
10. Tabelle/Bearbeitung starten (neuer Radweg ist markiert)
11. in Spalte „Line“ in das markierte Feld die nächsthöhere Nummer (zum Beispiel: 328) eintragen
12. Tabelle/Bearbeitung beenden
13. Änderungen speichern, Attributtabelle „Radwege“ schließen
14. Scripte (im Projektfenster)/„segm_akt“ öffnen (weiter mit 18.)
15. wenn „segm_akt“ nicht vorhanden, Scripte/Neu
16. Script/Textdatei laden
17. Datei „segm_akt.ave“ laden
18. Kompilieren (siehe Kompilierbutton im Kapitel 3.2.)
18. im View „Radwege“ Thema „Radwege“ aktivieren
19. Fenster/segm_akt (zu Script wechseln; View-Fenster muss zuletzt aktiv gewesen sein!)
20. Button „Ausführen“ (siehe Kapitel 3.2.) drücken
21. Thema mit den alten Radwegsegmenten auswählen („Radw_seg“)
22. Thema mit der neu digitalisierten Linie auswählen („Radwege“)
23. Meldung über die Zahl der letzten Segment-ID zur Kenntnis nehmen und „Ok“ drücken
24. Attributtabelle „Radw_seg“: Tabelle/Bearbeitung beenden
25. Änderungen speichern, Tabelle schließen

- Oracle: neue Segment-ID eingeben und Merkmale hinzufügen:

26. insert into <Tabellenname>

```
(<Spaltenname der ersten Spalte>, <Spaltenname der zweiten Spalte>)
values
(<Wert der ersten Spalte>, <Wert der zweiten Spalte>);
```

zum Beispiel: Segmente 6940 bis 6950 der Linie 328 kommen hinzu:

```
insert into SEGMENT
(SEGMENT_ID, LINE)
values
(6940, 328);
```

27. 26. für alle Segmente wiederholen

28. 26. und 27. für alle Referenztabellen wiederholen

29. update <Referenztabellenname>

```
set <Spaltenname der zweiten Spalte> = <Neuer Wert>
where
<Bedingung>;
```

zum Beispiel: die neuen Segmente bekommen das Attribut Tourismusnutzung:

```
update REF_NUTZUNG
set NUTZUNG_ID = 2
where
SEGMENT_ID between 6940 and 6950;
```

- ODER MS Query statt Oracle:

26. wie Punkt A, 7. bis 15.

27. in zweiter Spalte letzten Wert anklicken (zum Beispiel: Tabelle
SEGMENT)

28. Tabulatortaste drücken (neue Zeile erscheint)

29. alle neuen Werte für beide Spalten angeben (zum Beispiel: 6940, 328)

30. wie Punkt A, 18. bis 25.

31. Datei/Neue Abfrage

32. bei eventuellen Fragen: „Soll Abfrage gespeichert werden?“
verneinen

33. 26. bis 32. für alle Referenzdateien wiederholen

34. MS Query und Excel schließen, ohne die Abfragen zu speichern

F) Hinzufügen einer Sehenswürdigkeit (Geometrieänderungen - gilt auch für Gefahren- und Problempunkte)

- ArcView: zum Beispiel ein Museum hinzufügen

1. Attributtabelle von Thema „Sight“ öffnen
2. Tabelle/Alle Verbindungen lösen (nur bei „Sight“ nötig)
3. View aktivieren, Thema/Bearbeitung starten
4. mit Button  neue Sehenswürdigkeit im Maßstab 1 : 10.000 digitalisieren
5. Thema/Bearbeitung beenden
6. Änderungen speichern (Punkt ist noch markiert)
7. im Projektfenster Scripte/„sight_koord“ öffnen (weiter mit 11.)
(bei Gefahrenpunkten: „gpkt_koord“, bei Problempunkten: „ppkt_koord“ öffnen)
8. wenn Script nicht vorhanden ist, Scripte/Neu
9. Script/Textdatei laden
10. Datei „sight_koord.ave“ laden
11. Kompilieren (siehe Kompilierbutton im Kapitel 3.2.)
12. im View „Radwege“ Thema „Sight“ aktivieren
13. Fenster/sight_koord (zu Script wechseln; View-Fenster muss zuletzt aktiv gewesen sein!)
14. Button „Ausführen“ (siehe Kapitel 3.2.) drücken
15. als Tabelle „sight.dbf“ speichern (alte Datei ersetzen)
16. Projekt speichern, schließen und neu öffnen
17. Attributtabelle des Themas „Sight“ öffnen
18. Tabelle/Bearbeitung starten
19. in Spalte „Sight_id“ alle Datensätze von oben nach unten durchnummerieren
20. Tabelle/Bearbeitung beenden, Änderungen speichern
21. Tabelle „ref_sight“ öffnen und Bearbeitung starten (ab hier nur für Thema „Sight“)
22. Bearbeiten/Datensatz hinzufügen
23. Änderungen eintragen (zum Beispiel: Sight_id = 101, Sight_art_id = 1)
24. Tabelle/Bearbeitung beenden, Änderungen speichern
25. wie Punkt C, 15. bis 23.

G) Grafische Attribute

- ArcView: Änderungen der Grafik siehe Kapitel 7.

- ArcView: Einstellungen, bezogen auf den Maßstab 1 : 10.000:

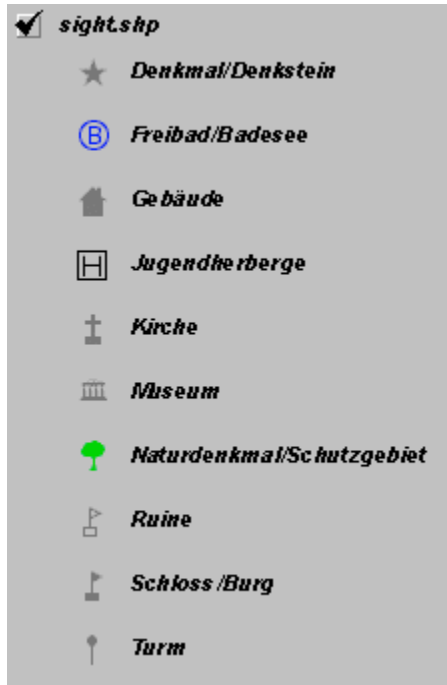


Abbildung 26: Die Symbole der bisher vorhandenen Sehenswürdigkeiten

Thema „Kreisgr“:

Umrissbreite: 2

Farbe: blasslila

Thema „Gemgr“:

Umrissbreite: 1

Farbe: blasslila

Thema „Radw_seg“ - Tourismus:

durchgezogene Linie

Breite: 2

Farbe: dunkelgrün

Thema „Radw_seg“ - Schüler- und Alltagsverkehr:

durchgezogene Linie

Breite: 2

Farbe: dunkelgelb

Thema „Radw_seg“ - geplant:

gestrichelte Linie

Breite: 2

Farbe: schwarz

Thema „Gefahrpkt“:

aufgerichtetes Dreieck

Größe: 24

Farbe Umrandung: schwarz

Farbe Füllung: rot

Thema „Problempkt“:

Kreis

Größe: 24

Farbe Umrandung: schwarz

Farbe Füllung: rot

Thema „Sight“:

Symbole aus den Zusatzpaletten von ArcView (Denkmal/Denkstein:

Stern, Freibad/Badesee: „B“ in Kreis, Gebäude: Haus,

Jugendherberge: „H“ in Quadrat, Kirche: Kreuz, Museum:

Säulentempel, Naturdenkmal/Schutzgebiet: Laubbaum, Ruine: Wimpel,

Schloss/Burg: gefüllter Wimpel, Turm: gefüllter Kreis mit Linie darunter)

Größe: 24

Farbe: für Denkmal, Gebäude, Kirche, Museum, Ruine, Schloss, Turm

– grau

für Freibad – blau

für Jugendherberge – schwarz

für Naturdenkmal – hellgrün

Alle anderen Klassen (Windmühle, Gaststätte, Fahrradwerkstatt,

Fahrradverleih, Botanischer Garten/Wildgehege) waren nicht

vorhanden, es wurden deshalb noch keine Signaturen vergeben!