



Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg

Studiengang Geoinformatik

Datenbanken in der Geoinformation

Bachelorarbeit

vorgelegt von: *Michél Schwandner*

Zum Erlangen des akademischen Grades

„Bachelor of Engineering“ (B.Eng.)

Erstprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig

Zweitprüfer: Prof. Dr.-Ing. Ernst Heil

Eingereicht am 03.09.2012

URN: urn:nbn:de:gbv:519-thesis2012-0643-5

Inhaltsverzeichnis

1 Einführung in die Thematik	5
2 Geoinformationssysteme	7
2.1 Was sind Geoinformationssystem	7
2.2 Aufbau und Anwendungen	8
2.3 Erweiterung zum offenen Standard	9
3 Modellierung von Geodaten.....	11
3.1 Eigenschaften und Standardisierung von Geodaten	11
3.2 Simple – Feature – Modell.....	13
3.3 SQL / MM Spatial	15
3.4 ISO 19107 – Spatial Schema	18
3.5 Räumliche Bezugssysteme	21
4 Datenbanksysteme	23
4.1 Allgemeiner Aufbau von Datenbanksystemen.....	23
4.2 Relationale Datenbanksysteme.....	24
4.3 Objektorientierte Datenbanksysteme	25
4.4 Objektrelationale Datenbanksysteme.....	27
5 Geodatenbanksysteme.....	28
5.1 Anforderungen Geodatenbanksysteme.....	30
5.2 Open-Source Geodatenbanksysteme	31
5.2.1 PostgreSQL / PostGIS	31
5.2.2 MySQL	34

5.2.3 SQLite / SpatialLite	38
5.3 Kommerzielle Geodatenbanksysteme	40
5.3.1 Oracle Spatial.....	40
5.3.2 Informix	45
5.3.3 DB2	48
5.4 Weitere Systeme	51
6 Bearbeitung von Standardoperationen	56
7 Zusammenfassung der Ergebnisse	68
7.1 Vergleich der Open – Source Geodatenbanksysteme	68
7.2 Vergleich der kommerziellen Geodatenbanksysteme	71
7.3 Vergleich der kommerziellen und Open – Source Geodatenbanksysteme	73
8 Ausblick und aktuelle Trends.....	76
8.1 Mobile Geodatenbanksysteme	76
8.2 Spatio – temporale Datenbanken	77
8.3 Spatial Data Mining	78
8.4 Sensornetzwerke und Datenströme	80
Literatur- und Quellenverzeichnis	82
Abbildungsverzeichnis.....	87
Tabellenverzeichnis.....	88

Kurzfassung

Diese Bachelorarbeit beschäftigt sich mit dem Thema der Datenbanken im Bereich der Geoinformation. Hier rücken besonders die sogenannten Geodatenbanken in den Vordergrund. Die einzelnen proprietären und freien Geodatenbanksysteme werden anhand ausgewählter Vergleichsmerkmale miteinander verglichen. Zu den wichtigsten Vergleichskriterien zählen hier die veröffentlichten Standards und Normen der OGC sowie der ISO. Anhand einfacher Standardoperationen für Datenbanken werden die Unterschiede in der Abfragebearbeitung versucht zu erläutern.

Abstract

This bachelor thesis deals with the issue of databases in the field of spatial information. Especially the so-called spatial databases come to the fore. The various proprietary and open spatial database systems are compared each other on the basis of selected characteristics. The main criteria for comparison here are the published standards and norms of the OGC as well as the ISO. The differences in the query processing of the various systems try to explain with the help of simple standard operations for databases.

1 Einführung in die Thematik

In den letzten Jahren nahm der Einsatz von Geodatenbanken stetig zu. Dies hat vielseitige Gründe. Zum einen sind Geodatenbanken heutzutage eine wesentliche Komponente von Geoinformationssystemen (GIS) und anderen Anwendungen zur Verarbeitung von räumlichen Daten. Im Zuge der Weiterentwicklung zu offenen Geoinformationssystemen wurden die standardmäßigen Datenbanken immer mehr von Geodatenbanken abgelöst. Zuvor wurden die Geodaten in externen Dateien oder in GIS gehalten. Erst durch den Einsatz von Geodaten in immer neuere Anwendungsbereiche verloren GIS ihre führende Rolle im Bereich der Geoinformatik oder der Katasterverwaltung. Deshalb wird seit Mitte der 1990er Jahre von offenen GIS gesprochen. Dieser Vorgang beschreibt die mangelnde Interoperabilität zwischen Systemen verschiedenster Anwendungsbereiche mit Hilfe von Normen und Standards zu beheben. Durch diese Entwicklung wurde eine einheitliche Datenhaltung, Datenverarbeitung und Datenaustausch ermöglicht. Als Ergebnis dieser Entwicklung stehen vereinheitlichte und standardisierte Geodateninfrastrukturen (GDI) zur Verfügung. Zum anderen eroberten die Geodatenbanken schnell den IT – Markt. Zur Anwendung kamen diese zum Beispiel in Google Maps und Google Earth, Navigationssystemen oder im Tourismussektor. Durch die Verschmelzung von Geoinformationsanwendungen und Webapplikationen wurden immer mehr Datenbankhersteller aufmerksam, sich diesem Thema anzunehmen. Erster Datenbankhersteller, der das Thema Geodatenbanken aufgriff, war Informix mit der räumlichen Erweiterung DataBlade. Das derzeit beliebteste und mächtigste Datenbanktool zu Verarbeitung von Geodaten ist Spatial aus dem Hause Oracle. Weiterhin gibt es zu den beiden erst genannten ebenfalls aus dem Hause IBM den DB2 Spatial Extender und den Microsoft SQL Server 2008. Das Datenmodell aller Systeme unterstützt die Forderungen des Simple – Feature Modells. Somit können räumliche Anfragen gestellt und räumliche Indexstrukturen erstellt werden. Neben diesen beiden proprietären Herstellern setzten sich auch vereinzelt Open – Source Datenbanksysteme durch. Zu diesen gehören PostgreSQL mit der Erweiterung PostGIS und MySQL. Die Unterschiede dieser beiden Datenbanken liegen im Funktionsumfang und in den daraus resultierenden unterschiedlichen Anwendungsbereichen. Diese Arbeit soll die Unterschiede zwischen den Open - Source Datenbanken untereinander sowie mit den proprietären

Systemen vergleichend darstellen. Zum anderen sollen alle Systeme mit den wichtigsten Standards für Geodatenbanksysteme schematisch gegenübergestellt werden. Zu Beginn werden jedoch die Geodatenbankstandards ISO und der OGC kurz vorgestellt. Danach folgt eine genauere Untersuchung der vorhandenen Funktionen der einzelnen Systeme in Bezug auf diese Geodatenbankenstandards. Mittlerweile ermöglichen objektrelationale Geodatenbanksysteme sogar die Speicherung von dreidimensionalen Geodaten, die Verarbeitung von XML, KML oder GML Dateien und können topologische Datenmodelle bereitstellen. Das letzte Kapitel dieser Arbeit gibt einen kleinen Ausblick in welche Richtung sich die aktuellen Trends im Bereich der Geodatenbanken bewegen können.

2 Geoinformationssysteme

2.1 Was sind Geoinformationssystem

Anfang der 1960er Jahre tauchte zum ersten Mal der Begriff des Geoinformationssystemes in Kanada auf. In den kanadischen Bundes- und Provinzialverwaltungen wurden Informationen über die jeweiligen Landnutzungen der einzelnen Regionen gesammelt. Als Grundlage dafür dienten analoge Karten, die die aktuelle und potenzielle Landnutzung veranschaulichten. Aufgrund der Größe des Landes konnte man manuell nur die einzelnen Kartenblätter zusammenfügen. Im Falle Kanadas waren das mehr als 3000. Um diese ungeheure Masse an Daten verwalten zu können benötigte man eine bekannte Struktur oder ein Bezugssystem zur weiteren Verarbeitung der Daten. So wurde als erste bekannte Struktur das Informationssystem eingeführt. Ein Informationssystem (IS) dient der rechnergestützten Erfassung, Speicherung, Verarbeitung, Pflege, Analyse, Benutzung, Verbreitung, Übertragung und Anzeige von Informationen bzw. Daten. Ein rechnergestütztes System besteht es aus Hardware, Software, Daten und deren Anwendungen [Bil10].

Somit wurde schon damals ein Frage-Antwort-System in seiner einfachsten Form erstellt. Nun war es möglich die Daten zu analysieren, aufzubereiten, zu präsentieren und wiederzugeben. Wenn es sich wie in dem Beispiel aus Kanada auch noch um Daten mit Raumbezug handelt, spricht man in diesem Falle von einem Geoinformationssystem. Ein Geoinformationssystem ist ein rechnergestütztes System, das aus Hardware, Software und Daten besteht und mit dem sich raumbezogene Problemstellungen in unterschiedlichen Anwendungsgebieten modellieren und bearbeiten lassen. Die dafür benötigten raumbezogenen Daten / Informationen können digital erfasst werden und redigiert, verwaltet und reorganisiert, analysiert sowie alphanumerisch und grafisch präsentiert werden [Bil10].

Die Spezialisierung dieses speziellen Informationssystems liegt dabei auf Geodaten.

2.2 Aufbau und Anwendungen

Durch die Spezialisierung auf Geodaten verändern sich auch die Aufgabenbereiche und Anwendungsbereiche herkömmlicher Informationssysteme. Derzeit finden GI-Systeme Einsatz in einer Vielzahl von Unternehmen, Behörden und Organisationen. Die nachfolgende Liste zeigt dabei nur einen kleinen Ausschnitt aus den tatsächlich verfügbaren Anwendungsbereichen auf.

- Vermessungswesen (Erstellung topographischer Karten)
- Katasterwesen (automatisierte Liegenschaftskarten)
- Facility Management (Verwaltung von Gebäuden)
- Amtliche Statistik (Bevölkerungsstatistik)
- Raum und Bebauungsplanung (Bodenordnung)
- Leitungsdokumentation (Netzinformationssysteme)
- Telekommunikation (Mobilfunkstationen)
- Verkehr (Navigationssysteme)
- Marketing (Standortplanung)
- Umweltschutz (Umweltinformationssystem)
- Versicherungswirtschaft (Hochwassereinschätzung)
- Geologie (Bewertung von Bodenschätzen)
- Logistik (Fuhrparkkontrolle)

[Bri08]

In den nächsten Jahren wird diese Liste auch weiterhin ansteigen. Dies liegt zum einem an der immer größer werdenden GIS – Anwendergemeinde und zum anderen werden auch immer neue Anwendungsbereiche und Technologien erschlossen. Von mobilen Endgeräten Servern bis hin zu Desktoparchitekturen lassen sich heute GI-Systeme finden. Dabei folgen alle Geräte den selben Aufbau wie auch alle anderen rechnergestützten Informationssysteme. Dieser setzt sich aus den 4 Komponenten Erfassung (Input), Verwaltung (Management), Analyse (Analysis) und Präsentation (Presentation) zusammen. Dieses 4 Komponentenmodell kann auch als EVAP (IMAP) abgekürzt werden. Die nachfolgende Abbildung zeigt das Zusammenspiel der einzelnen Komponenten.

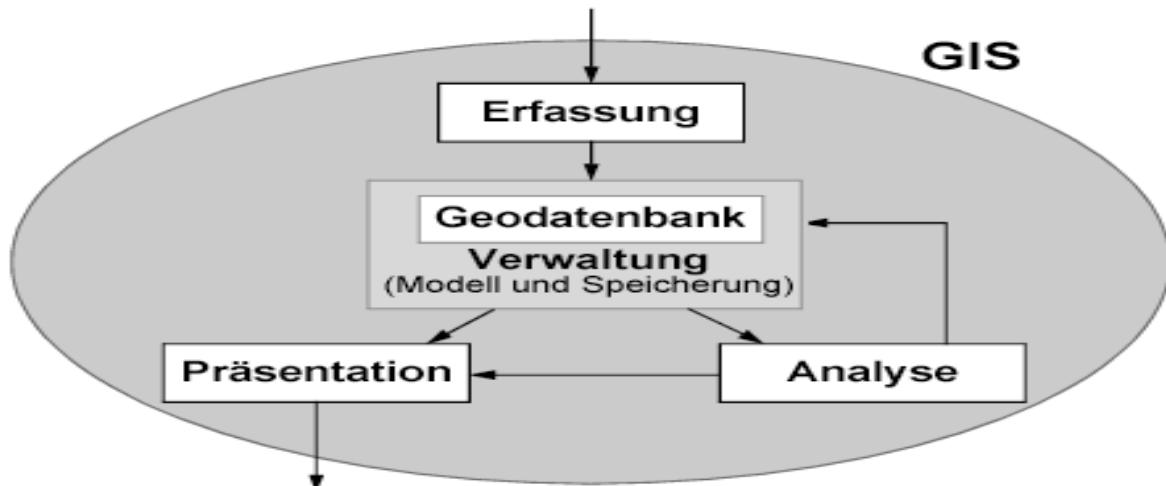


Abbildung 1: 4 Komponentenmodell GIS [JU012]

Unter Eingabe versteht man die Datenerfassung sowie die Erhebung von Geodaten durch zu Hilfenahmen von vermessungstechnischen Methoden. Erfasst werden können mit einem GIS zum Beispiel Vektordaten, Sachdaten oder Rasterdaten. Die Verwaltungskomponente beschreibt die Verwaltung, Verarbeitung und Manipulation der in digitaler Form vorliegenden Geodaten. Das Kernstück bildet hierbei die Geodatenbank mit dem dazugehörigen Datenbankmanagementsystem. Der Analyseteil zeichnet sich durch seinen großen Funktionsumfang zur Datenauswertung aus. Dadurch können Informationen zur Entscheidungsgewinnung gewonnen werden. Des Weiteren können geometrische, logische, relationale und statistische Verfahren auf die Daten angewendet werden. Der letzte Teil der Verarbeitungskette ist die Ausgabekomponente. Diese hat die Aufgabe die Ergebnisse zu visualisieren, weshalb sie auch heutzutage als Geovisualisierung bezeichnet wird. Der Vorteil des GIS gegenüber den älteren Ausgabemethoden liegt in der Verwendung von Methoden aus der Kartografie und der Computergrafik.

2.3 Erweiterung zum offenen Standard

Die Entwicklung im Bereich der Geoinformatik nahm in den letzten Jahren eine drastische Entwicklung. In den letzten Jahren wurden vor allem technische und funktionale Eigenentwicklungen vorangetrieben. Man verpasste es jedoch die bereits die vorhandenen Komponenten und Systeme weiterzuentwickeln. Jeder Hersteller hat bis circa Mitte der 1990er Jahre seine eigenen Produkte für die

Datenmodellierung und Datenspeicherung forciert. Dies lag grundsätzlich an den Nachteilen der vorhandenen Datenbanksysteme. Zu erwähnen sind hier unter anderem die fehlende Standardisierung und der mangelnde Entwicklungsaufwand in diesem Bereich. Ersteres hat zu Folge, dass ein erhöhter Schulungsaufwand für Mitarbeiter besteht. Zum anderen besteht die Gefahr, dass die spezifischen Datenbanksysteme der GIS - Hersteller über einen längeren Zeitraum gesehen, hinter den Entwicklungen der IT – Branche zurückbleiben. Diese vermeintlichen Nachteile wurden im Laufe der Zeit jedoch ausgemerzt, da sich die GI – Technologien in immer mehr Anwendungsfeldern, Organisationen und Firmen durchgesetzt haben. Dadurch verlieren die GIS ihre Vorrangstellung im vermessungstechnischen Bereich und in der Energieversorgung. Vielmehr mussten die GIS in die bestehen Infrastrukturen der verschiedensten neuen Bereiche integriert werden. Durch diesen Prozess wurde Mitte der 1990er Jahre der Begriff vom „Open GIS“ geprägt. Viele GIS – Hersteller, GIS – Anwender und Forschungseinrichtungen haben sich 1994 zum „Open Geospatial Consortium“ (OGC) zusammengeschlossen, um die fehlende Interoperabilität zwischen den GIS und den Standardanwendungen zu beseitigen. Unter Interoperabilität versteht man die Fähigkeit zweier oder mehr Systeme, verzahnt miteinander zu arbeiten. Dafür müssen geeignete Geodatenformate, -modelle und -schnittstellen spezifiziert und durch standardisierte Metadaten beschrieben werden [Bri09].

Nicht nur das OGC, sondern auch durch Normungen wurde die Erweiterung zum offenen Standard vorangetrieben. UML als einheitliche Modellierungssprache findet man einerseits in der ISO – Normung und andererseits im AAA – Konzept wieder. Auf europäischer Ebene gibt es das Normungsprogramm INSPIRE, das die Entwicklung im Bereich von Geodateninfrastrukturen forciert. Ein wesentlicher Bestandteil der Bemühungen GIS für andere Anwendungen zu öffnen, stellt die Verwendung von handelsüblichen Datenbanksystemen und standardisierten Datenmodellen dar, so dass beliebige Anwendungen Geodaten aus der Datenbank lesen bzw. in der Datenbank ablegen können [Bri08]. Die kommerziellen GIS – Anwendungen besitzen heutzutage bereits eine solche Funktionalität.

3 Modellierung von Geodaten

3.1 Eigenschaften und Standardisierung von Geodaten

Informationen die mit Hilfe von vermessungstechnischen Verfahren aufgenommen wurden, können nur mit einem rechnergestützten System weiterverarbeitet werden, wenn diese vorher in einem Modell abgebildet worden sind. Dabei muss jedoch berücksichtigt werden, dass man die Komplexität der Informationen meist nicht einfach in das neue System übernehmen kann. Deshalb muss zunächst eine Vereinfachung vorgenommen werden. Mit der Hilfe von sogenannten Datenmodellen können die jeweiligen Informationen in verschiedene logische Ebenen eingeordnet und dargestellt werden. Dabei beschreibt ein Datenmodell die grundlegenden Eigenschaften die für alle Erscheinungen einer bestimmten Sicht auf die Wirklichkeit eine einheitliche Abbildung erleichtern. Es bestimmt die grundsätzlichen Strukturen, die Beziehungen, die prinzipiell möglich sind, und die Eigenschaften, die zugeordnet werden können [Bil10].

Man muss sich also im Klaren sein, dass durch diesen Prozess Informationen verfälscht oder verloren gehen können, da nur ein vereinfachtes Datenmodell in einer Datenbank abgebildet werden kann. Bereits bei der Erfassung der Daten im Feld werden diese mit thematischen Eigenschaften versehen. Dies kann zum Beispiel die Postleitzahl oder der Name einer Gemeinde sein. Da diese Informationen hingegen nicht ausreichen um die Menge an Daten später im Rechner zu unterscheiden, wurden weitere Unterscheidungskriterien eingeführt. Zu diesen zählen die geometrischen, topologischen und temporalen Eigenschaften von Geodaten. Diese werden im Folgenden kurz erläutert.

Die thematischen Eigenschaften lassen sich derzeit ganz einfach mit einer Datenbank nachbilden. Thematische Eigenschaften dienen der weiteren Beschreibung von nicht-geometrischen Eigenschaften. Zu diesen zählen nominale Eigenschaften, qualitative Eigenschaften und quantitative Eigenschaften. Bei der Zuordnung von thematischen Eigenschaften wird zwischen objektbasierten Datenmodellen und raumbasierten Datenmodellen unterschieden. Wichtig ist jedoch das diesen Daten ein geeigneter ein- oder zweidimensionaler Raum zugewiesen

wird. Das Geobjekt beziehungsweise der Datenraum selbst stehen im Mittelpunkt der objektbasierten beziehungsweise der raumbasierten Ansätze. Zur Bildung von Layern kann es kommen, wenn es mehrere thematische Attribute oder verschiedene Funktionen zu einem Geobjekt gibt.

Geometrische Eigenschaften von Geodaten dienen zur Beschreibung der Lage und Ausdehnung im Raum [Stö11]. Zur Darstellung der geometrischen Eigenschaften gibt es 2 verschiedene Modellansätze. Zum einen das Vektormodell und das Rastermodell. Beim Rastermodell bestehen alle Daten aus gleichförmigen Flächen. Die Aufteilung erfolgt dann meist in quadratischen Rasterzellen (Pixel). Dazu kann es zu einem Verlust der Genauigkeit führen. Vektordaten hingegen setzen sich aus Punkten und Linien zusammen. Aus diesen beiden Grundgeometrien können weitere komplexe Strukturen, wie Polygone oder Polylinien zusammengesetzt werden. Über Koordinaten wird die Lage der einzelnen Punkte beschrieben.

Die topologischen Eigenschaften können einerseits direkt in einem topologischen Datenmodell dargestellt werden oder indirekt aus der Geometrie der Geodaten abgeleitet werden. Die Topologie selbst beschäftigt sich mit den nichtmetrischen, räumlichen und strukturellen Beziehungen beliebiger Elemente in abstrakten Räumen [Bil10]. Demnach dienen die topologischen Eigenschaften von Geodaten zur Beschreibung der relativen räumlichen Beziehungen von Objekten zueinander, wobei von der Geometrie abstrahiert wird [Stö11]. Als Beispiel für die Topologie kann die Darstellung eines Liniennetzes dienen.

Die temporalen Eigenschaften dienen zur Beschreibung eines Zeitpunktes oder eines Zeitraumes von Geobjekten. Von besonderem Interesse kann auch die Betrachtung der Eigenschaften für mehrere aufeinanderfolgende Zeitpunkte oder Zeiträume sein. Liegen diese nämlich zeitlich versetzt kann die Dynamik eines Geobjektes beschrieben werden. Werden bei der Betrachtung nicht nur die zeitlichen sondern auch die räumlichen Veränderungen betrachtet, spricht man auch von einer räumlich – zeitlichen Modellierung der Daten (Spatio-Temporal Data Modeling). Diese können aber nur mit Hilfe von GIS oder Geodatenbanksystemen umgesetzt oder dargestellt werden.

Ein erster wichtiger Schritt zu Öffnung von Geoinformationssystemen war die Erarbeitung von Normen und Standards für den Datenaustausch. Dazu gehört auch das Ablegen von raumbezogenen Daten in einer Geodatenbank. Mitte der 1990er drängten sich 2 Organisationen in den Vordergrund. Zum einem war dies, die OGC (Open Geospatial Consortium) und zum anderen das Technische Komitee 211 „Geographic Information / Geomatics“ der International Organisation for Standardization. Aus Sicht der Datenbank ist auch noch das ISO / IEC – Komitee zur Entwicklung der Norm SQL / MM Spatial zu erwähnen. Zwei Arten bei der Spezifikation werden bei der Standardisierung von Geodaten unterschieden. Dabei werden die abstrakte Spezifikation und die Implementierungsspezifikation näher betrachtet.

Abstrakte Spezifikationen bilden implementierungsabhängige Rahmen an dem sich technische Spezifikationen ausrichten sollen. Implementierungsspezifikationen bieten Entwicklern eine Basis, auf der Software implementiert werden kann. Die Spezifikation muss dabei so konkret seien, dass die Software in der Lage ist mit einer Software, die die gleichen Spezifikationen besitzt, kommunizieren zu können [Kud07].

In einem Abkommen zwischen der OGC und der ISO / TC 211 von 1998 wird die zukünftige Zusammenarbeit geregelt. Die abstrakten Spezifikationen der OGC werden durch neu erarbeitete Normen der ISO / TC 211 ersetzt. Als Normvorschläge werden bei der ISO die neu erarbeiteten Implementierungsspezifikationen der OGC eingereicht.

3.2 Simple – Feature – Modell

Eines der wohl wichtigsten Datenmodelle für Geodatenbanksysteme ist das Simple – Feature – Modell das ursprünglich von der OGC entwickelt wurde. Das OGC wurde 1994 gegründet und umfasst heute rund 330 Mitglieder. Das Hauptaugenmerk legt die OGC auf Spezifikationen für Schnittstellen und Codierungsregeln. Das Modell beruht auf einer Untermenge aus der ISO – Norm 19107 Spatial Schema und dient zur Beschreibung zweidimensionaler Vektorgeometrien. Seit 2004 findet sich das Modell in 2 ISO – Normen wieder. Einmal wird das allgemeine Modell vorgestellt

(ISO19125-1: Simple Feature Access – Common Architecture) und auf der anderen Seite die Umsetzung in einem SQL – Datenbankschema beschrieben (ISO19125-1: Simple Feature Access – SQL – Option). Bei Simple Features handelt es sich um Geometrien im zweidimensionalen Raum, deren Stützpunkte geradlinig miteinander verbunden sind [Bri08]. Die folgende Abbildung verdeutlicht nochmals bei welchen Geometrietypen es sich um Simple Features handelt und bei welchen nicht.

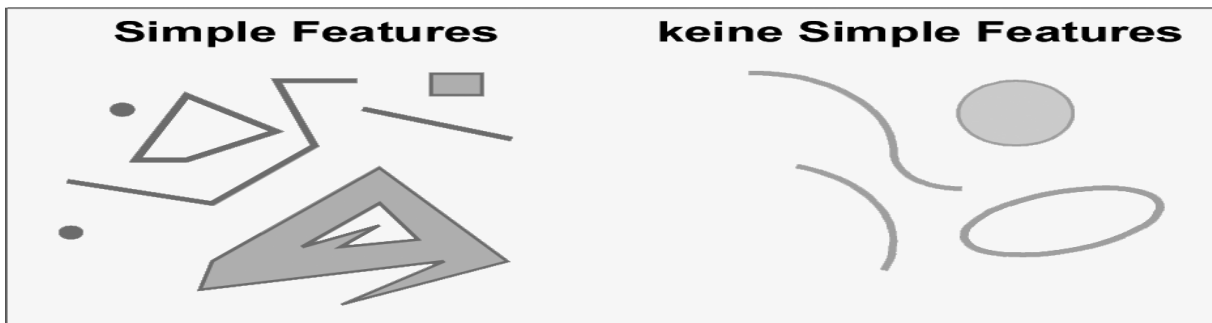


Abbildung 2: Darstellung Simple Features [JUO12]

In einem UML – Klassendiagramm werden die verschiedenen Geometrietypen veranschaulicht dargestellt. Die Klasse „Geometrie“ stellt die Oberklasse dar von der alle anderen Klassen erben. Auch kann jeder Geometrie ein räumliches Bezugssystem zugewiesen werden. Die Klassen Punkt, Linie, Fläche und Geometriesammlung leiten sich von der Oberklasse Geometrie ab. Die nachfolgende Abbildung 3 zeigt das gesamte UML – Klassenmodell der Simple- Features.

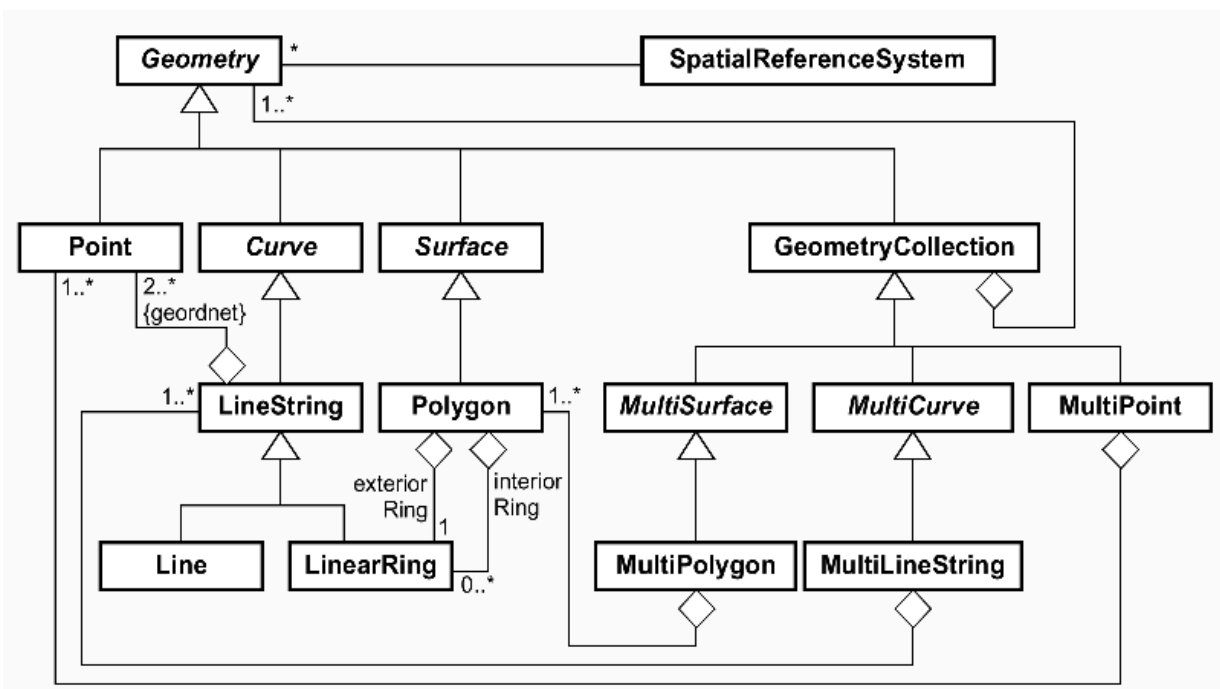


Abbildung 3: Geometrieschema Simple Features [JUO12]

In den Geometrieklassen werden die verschiedenen Objekte näher definiert. So setzt sich zum Beispiel ein Punktobjekt aus einer bestimmten Position im Raum zusammen. Über bestimmte Methoden kann auf die Koordinaten zugegriffen werden. Ähnliches trifft auch auf die Objekte der Klassen Linie und Fläche zu. Nur lassen sich hier noch weitere Methoden finden, die über diese Klassen aufgerufen werden können.

Weiterhin werden im Simple – Feature – Modell noch verschiedene Repräsentationsformen unterschieden und topologische Prädikate eingeführt. Die Simple – Features können in 2 verschiedenen Varianten dargestellt werden. Zum einen gibt es die Textpräsentation als Well – known Text (WKT) und zum anderen die Binärpräsentation als Well – known Binary (WKB).

Die Textpräsentation dient zur alphanummerischen Darstellung der Geometrien. Die Darstellung eines Punktes mit den Koordinaten 1,1 könnte wie folgt aussehen: Point(1 1). Die Binärpräsentation hingegen stellt die Geometrien als kontinuierliche Bytekette dar. Als Datentyp dient hier ein Integerwert ohne Vorzeichen. Die Darstellung für dieselbe Koordinate sieht wie folgt aus: 01010000000000000000F03F000000000000F03F.

Topologische Prädikate geben Aufschluss über die topologischen Beziehungen zwischen Geometrien mit Hilfe der booleschen Algebra. Zur Beschreibung dieser Beziehungen wird das 9 – Intersection – Modell genutzt.

Auch werden im Simple – Feature Modell verschiedene geometrische Funktionen bereitgestellt. Zu diesen Funktionen gehören unter anderem die Berechnung der Konvexen Hülle, eines Buffers, der Distanz zwischen 2 Punkten und verschiedenen Verschneidungsfunktionen wie Union, Differenz oder Intersection. All diese Funktionen befinden sich in der Klasse „Geometry“.

3.3 SQL / MM Spatial

Datenbanken sind heute anerkannte Technologien in Geoinformationssystemen. Viele der großen Datenbankhersteller unterstützen die standardisierte Datenbankabfragesprache SQL. Durch die leicht erlernbare Syntax, die auch auf fast

jedem System läuft, macht SQL zu der bedeutendsten Abfragesprache im GIS Bereich. In der Standardurversion konnten einfache Abfragen über Sachdaten erfolgen. Doch viele GIS Produkte speichern nicht nur Sachdaten sondern auch Geometriedaten in relationalen Datenbanken ab und können deshalb auch auf SQL zurückgreifen. Im Rahmen der Standardisierung im GIS – Sektor in den 1990er Jahren wurde auch SQL im Jahre 1999 zum ersten Mal genormt. Im Laufe der Jahre konnte der SQL – Standard nicht mit allen Neuentwicklungen auf dem GIS – Markt Schritt halten. Von daher wurden immer wieder genormte SQL – Erweiterungen als Standard auf den Markt gebracht. Einer der wichtigsten Standards im Bereich der Geoinformatik ist der SQL / MM Spatial der ISO. MM steht dabei für Multimedia. Neben dem MM Spatial untergliedert sich dieser Standard noch in MM Full – Text für die Unterstützung komplexer Dokumente, MM Still Image zur Speicherung von Rasterbildern und MM Data Mining zur Unterstützung von Suchanfragen. MM Spatial dient vor allem als Datenbankschema für Geodaten. Dieser liegt in 3 Versionen vor, wobei die letzte im Jahre 2006 veröffentlicht wurde. Die nachfolgende Abbildung 4 zeigt das vollständige Geometriemodell dieses Standards. Dieses ähnelt in den Grundzügen dem Simple – Feature – Modell.

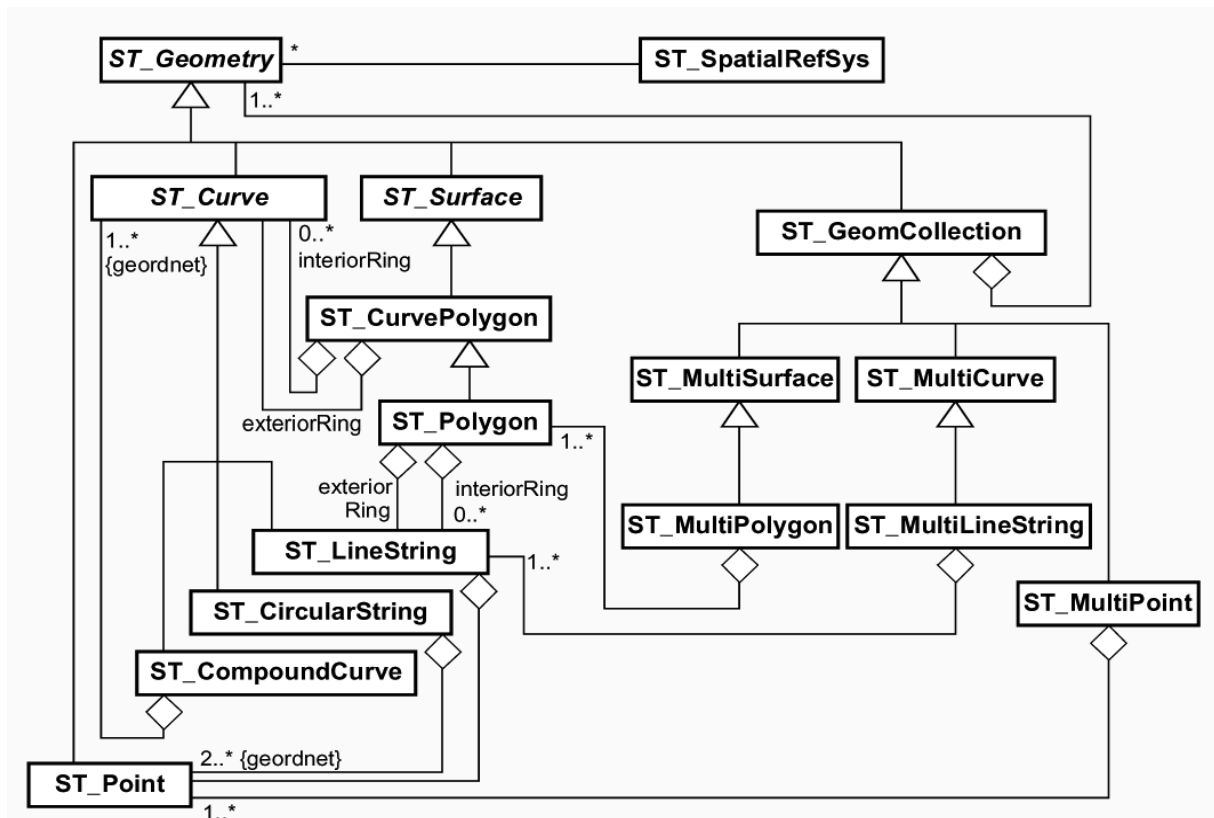


Abbildung 4: Geometrieschema SQL / MM Spatial [JUO12]

Im Grunde genommen wurde versucht in der Ursprungsversion das Simple – Feature Modell nachzubilden und später sogar mit einigen neuen Funktionen zu erweitern. Neben diesen Klassen sind ebenfalls Definitionen für die Darstellung von räumlichen Bezugssystemen, Winkeln und Richtungen vorhanden. Auch in diesem Modell stellt die abstrakte Klasse ST_Geometry die Oberklasse dar. Der einzige Unterschied zum bestehenden Simple – Feature – Modell ist, dass der Funktionsumfang der einzelnen Klassen um einiges höher ist. So sind zum Beispiel folgende Funktionen im Simple – Feature – Modell nicht implementiert worden.

Name der Funktion	Aufgabe
ST_CoordDim()	Liefert die Dimension der Koordinaten, die die Geometrie im Datenraum beschreiben
ST_Transform()	Transformiert die Geometrie in ein anderes räumliches Bezugssystem
ST_IsValid()	Boolesche Funktion validiert die Geometrie
ST_Is3D	Boolesche Funktion prüft, ob Z-Koordinatenwerte vorliegen
ST_AsGML()	Gibt die Geometrie als XML – Fragment der GML zurück

Tabelle 1: nicht definierte Funktionen Simple – Feature – Modell [Bri08]

Auch in diesem Modell können Geometrien einem räumlichen Bezugssystem zugeordnet werden. Dies kann mit Hilfe eines Verweises auf die Klasse ST_SpatialRefSys geschehen.

Auch in diesem Modell lassen sich verschiedene Repräsentationsformen für Geometrien finden, ähnlich dem Simple – Feature – Modell. Auch hier wird zwischen WKB und WKT unterschieden. Der syntaktische Aufbau unterscheidet sich jedoch nicht von den Repräsentationsformen im Simple – Feature – Modell.

Zudem enthält das SQL/MM Spatial in der 3. Edition zusätzlich zum Simple – Feature Modell ein Datenmodell für Netzwerke und Topologien.

3.4 ISO 19107 – Spatial Schema

Die Aktivitäten im Bereich der Standardisierung nahm im GIS – Sektor Mitte der 1990er Jahre stetig zu. Dabei sind Normungen auf nationaler und internationaler Ebene zu unterscheiden. In den letzten Jahren werden Standardisierungen zu raumbezogenen Informationsbeschreibung von der ISO durchgeführt. Die International Organization for Standardization wurde 1947 gegründet und ging aus der Union der nationalen Standardisierungsinstitutionen hervor. Das technische Komitee TC 211 „Geoinformation / Geomatik befasst sich damit Standards für digitale raumbezogene Informationen zu entwerfen. Eine der wichtigsten Normen in Bezug auf Datenbanken und Geodaten ist die ISO Norm 19107 „Geographic Information – Spatial Schema“ aus dem Jahre 2003. Das darin enthaltende konzeptionelle Datenmodell enthält Beschreibungen für räumliche Beschreibungen von Geodaten und definiert gleichzeitig auch räumliche Standardoperationen für Zugriff, Anfrage, Verwaltung, Verarbeitung und Austausch von Geoobjekten. Als Datenmodell wird das objektbasierte Vektordatenmodell verwendet. Das Modell der Feature - Geometry Spezifikation setzt sich aus 2 Hauptpaketen zusammen. Diese sind hier aufgelistet.

- Paket „Geometry“ (geometrische Eigenschaften der Geoobjekte)
- Paket „Topology“ (topologische Eigenschaften der Geoobjekte)

Das Paket „Topology“ baut dabei auf dem Paket „Geometry“ auf und verwendet nebenbei noch andere Pakete aus anderen ISO Normen.

- Paket „Coordinates“ aus der ISO-Norm 19111 Spatial Referencing by Coordinates

- Paket „Basic Types“ aus der ISO-Norm 19103 Conceptual Schema Language

Die folgende Abbildung soll den Zusammenhang der einzelnen Pakete nochmals näher erläutern.

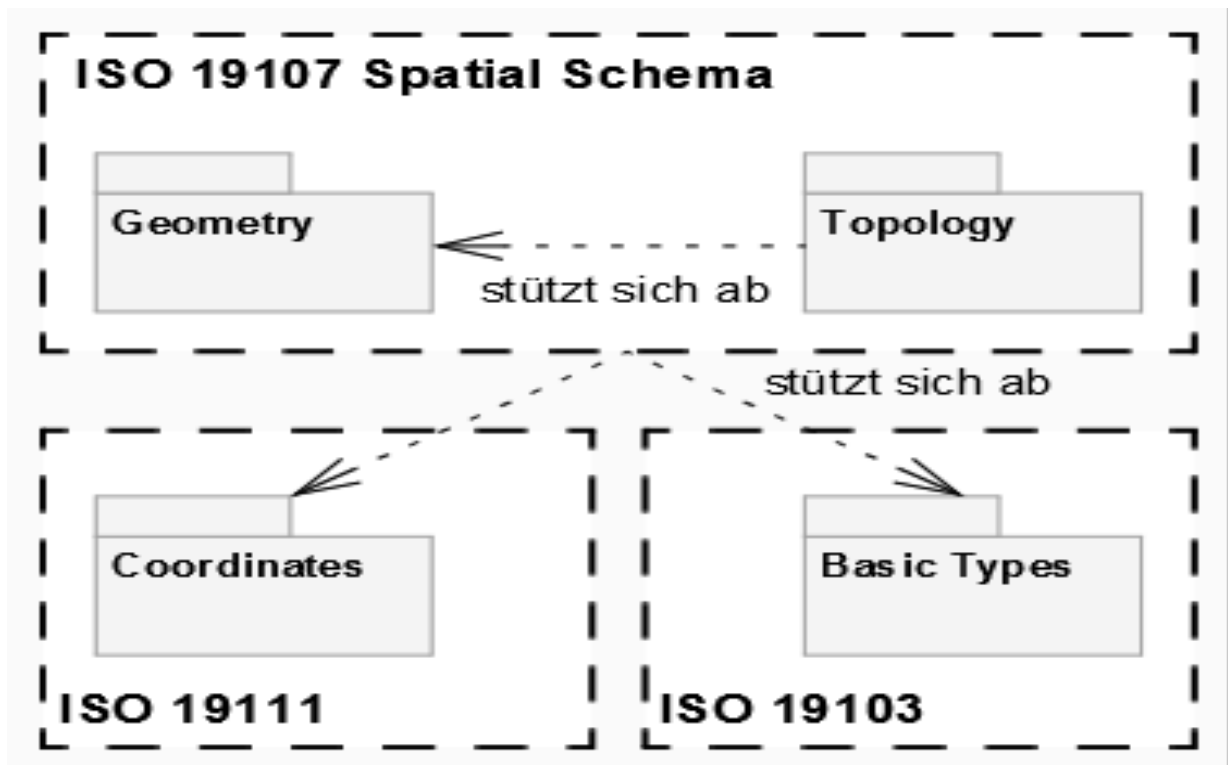


Abbildung 5: Pakete der ISO 19107 Norm [JU012]

Als nächstes legen wir den Fokus auf das „Geometry“ Paket, das auch in allen anderen vorgestellten Normen und Standards auftaucht. Dieses Paket besteht aus 5 Teilpaketen und beschreibt die geometrischen Informationen von Geodaten. Zu den 5 Teilpaketen gehören:

- Geometry root enthält eine allgemeine Oberklasse für alle Geometrien
- Geometry primitive stellt die geometrischen Primitive zur Verfügung

- Coordinate Geometry enthält Klassen, die für die Definition von Geometrie mittels Koordinaten (Stützpunkte) benötigt werden.
- Geometry aggregate erlaubt die Zusammenfassung von mehreren Geometrien in losen Geometrieaggregationen
- Geometry complex definiert hingegen die Zusammenfassung von mehreren geometrischen Primitiven, um eine komplexe Geometrie zu beschreiben

[Bri08]

Die nachfolgende Abbildung zeigt nochmals die wichtigsten Klassen und Beziehungen aus den verschiedenen Teilpaketen. In der Klasse GM_Object befinden sich alle Attribute und Methoden, die ebenfalls allen Klassen zugehörig sind.

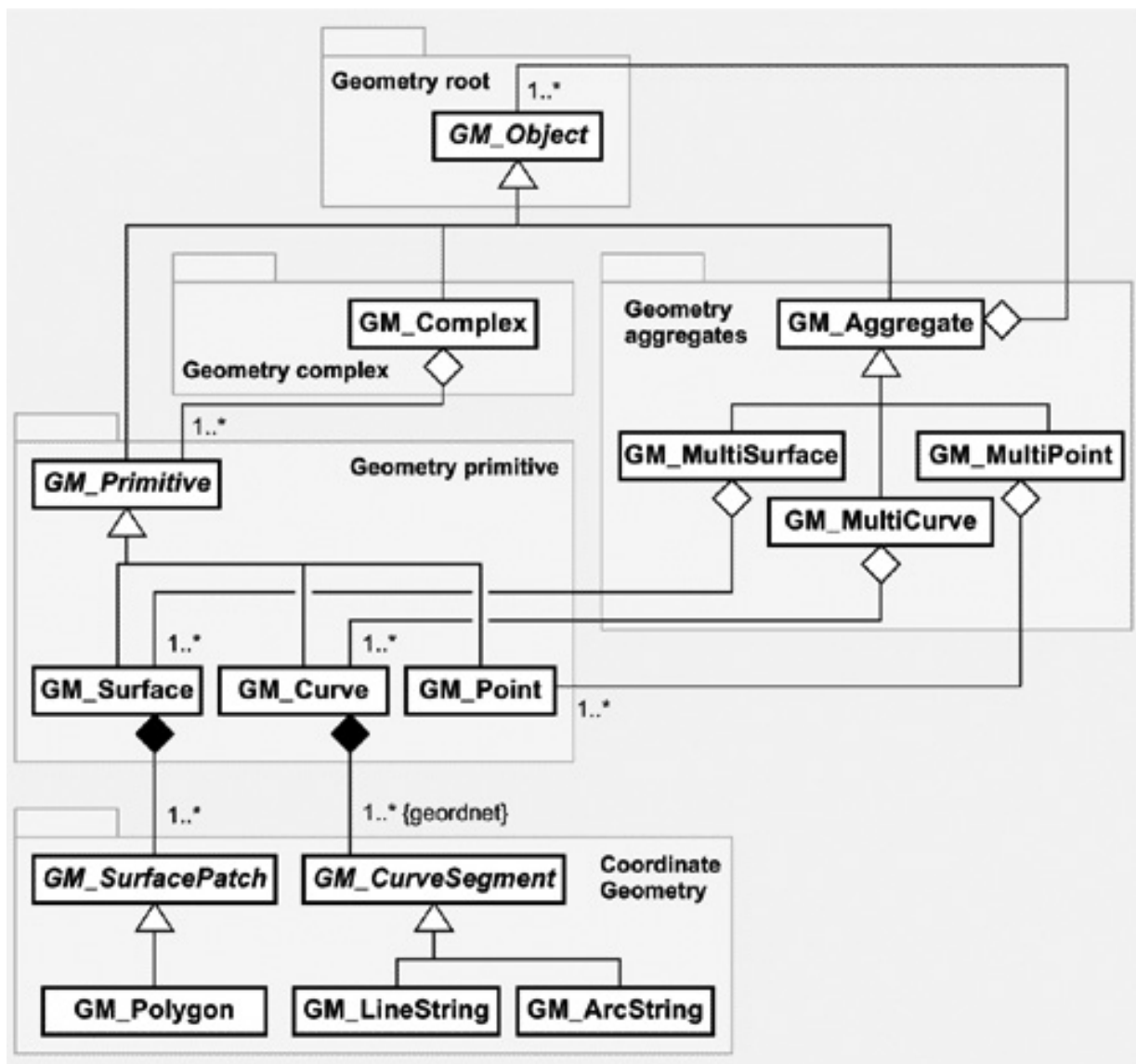


Abbildung 6: Geometrieschema Feature – Geometry Modell [JUO12]

3.5 Räumliche Bezugssysteme

Räumliche Bezugssysteme erlauben die Interpretation der gespeicherten Koordinaten als Beschreibung von Lage- und Ausdehnungsinformationen in einem realen Datenraum [Bri08]. Dies ist dann zum Beispiel von Nöten, wenn Längen- oder Flächeninformationen aus einem Objekt abgeleitet werden müssen. Man unterscheidet dabei georeferenzierte Koordinatensysteme, die einen Bezug zu einer bestimmten Position auf der Erdoberfläche besitzen und lokale Koordinaten. Diese werden durch eine Position unabhängig von der Erdoberfläche charakterisiert und kommen vorrangig in CAD – Systemen zum Einsatz.

Bei georeferenzierten Koordinatensystemen kann man die Koordinaten in 2 verschiedene Typen unterteilen, zum einen in geographische Koordinatensystemen und projizierte Systeme. Erstere beschreiben ein Ellipsoid, dass sich möglichst gut dem mathematischen Gesamtbild der Erde anpasst. Das WGS-84 – Ellipsoid ist neben dem Bessel und Krassowski – Ellipsoid das bekannteste und das meist verwendete in Deutschland. Das Koordinatensystem hat seinen Ursprung im Geozentrum und als Rotationsachse wird die z – Achse festgelegt. Die Koordinaten werden über geographische Koordinaten festgelegt. Bezugspunkte sind der Äquator und der Nullmeridian. Durch geographische Länge und Breite sowie einer eventuellen Höhenangaben kann ein Punkt exakt auf der Erde bestimmt werden.

Projizierte Koordinatensysteme werden dazu genutzt die Erde mathematische genau in einer Kartenprojektion abzubilden. Durch Flächentreue, Winkeltreue oder Linientreue kann es in projizierten Koordinatensystemen zu Verzerrungen kommen. Deshalb finden diese Systeme vor allem in Kartenanwendungen ihren Einsatz.

Zur Beschreibung von Koordinaten wurden deshalb die schon angesprochenen WKT und WKB Repräsentationsformen von der OGC und der ISO eingeführt. Eine weitere Möglichkeit zur Darstellung stellt die im Jahre 2005 gegründete European Petroleum Survey Group (EPSG) dar. Diese entwickelte eine Datenbank zur Beschreibung von räumlichen Bezugssystemen. Diese Datenbank ist im Netz frei zugänglich und enthält über 500 verschiedene Datumsangaben und über 50 verschiedene

Ellipsoidbeschreibungen. Des Weiteren stehen noch 1000 räumliche, georeferenzierte Bezugssysteme zur Verfügung.

Ein lineares Bezugssystem erlaubt die Identifizierung von Punkten auf einer Linie durch Entfernungsangaben zu einem ausgezeichneten Anfangspunkt [Bri08]. Gerade in Netzwerken findet diese Art der Anwendung großen Zuspruch.

4 Datenbanksysteme

4.1 Allgemeiner Aufbau von Datenbanksystemen

Im Laufe der Entwicklung des Computers wurde die Datenverwaltung in Dateisystemen zunehmend durch Datenbanken abgelöst. Datenbanksysteme dienen der persistenten Speicherung von Daten. Wichtige Aufgaben dabei sind die Beschreibung der Daten durch ein Datenmodell, die Verwaltung der Daten und der Zugriff auf die Daten über eine Datenbanksprache [Bri08]. Die heutigen modernen Datenbanksysteme bestehen zumeist aus den folgenden Bestandteilen, der eigentlichen Datenbank und dem Datenbankmanagementsystem. Die nächste Abbildung erläutert diesen Sachverhalt nochmals genauer.

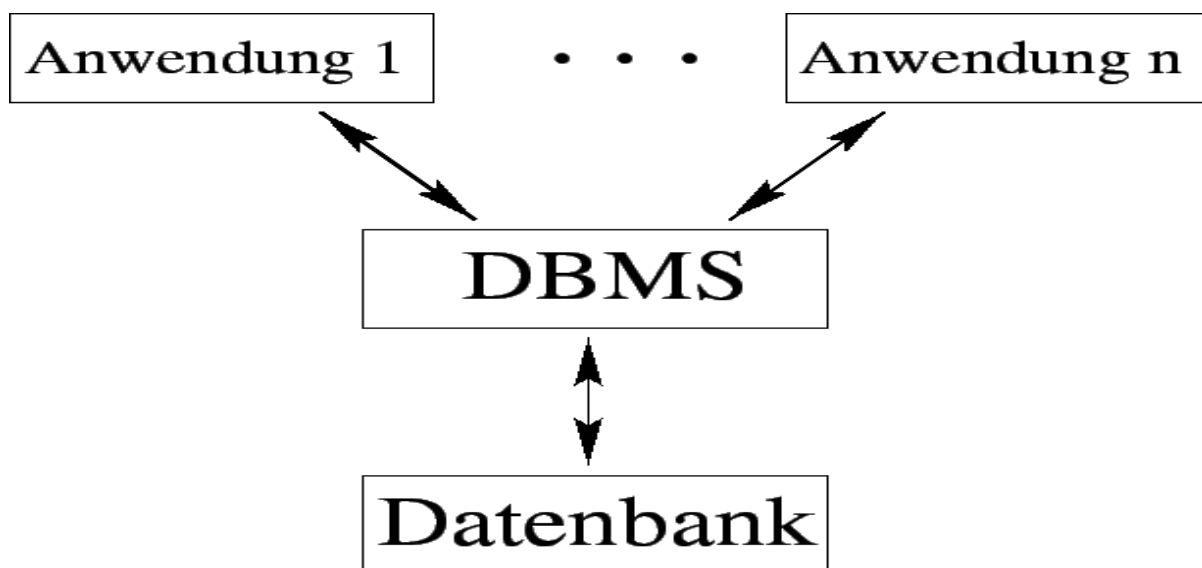


Abbildung 7: allgemeiner Aufbau Datenbanksysteme [UHH11]

Eine Datenbank beschreibt eine systematische Sammlung von Daten. Diese Daten müssen einheitlich beschrieben sein und sind zur dauerhaften Speicherung vorgesehen. Die Datenstrukturen sind von den Anwendungen in einer Datenbank völlig losgelöst. Das Datenbankmanagementsystem bildet das Herzstück einer Datenbank. Es trägt zum Datenschutz und zur Datenkonsistenz bei. Auf der anderen Seite ermöglicht es ebenfalls eine einheitliche Beschreibung, die Verwaltung und die Abfrage der Daten aus der Datenbank.

4.2 Relationale Datenbanksysteme

Die ersten konkreten Datenbanken beruhen auf dem relationalen Ansatz. Diesem Ansatz liegt das relationale Datenbankmodell von Edgar F. Codd aus dem Jahre 1970 zu Grunde. Dabei dient das Datenmodell einerseits der formalen Beschreibung der Daten. Auf der anderen Seite sollen ebenfalls die Beziehungen der Daten untereinander erläutert werden. Das relationale Modell sieht vor, dass die Daten in Tabellen organisiert werden. Sollten die Daten nur kurzfristig gespeichert werden, verwendet man den Begriff Relationen anstatt Tabellen. Ein Datensatz wird auch Tupel genannt und entspricht einer Zeile in der Tabelle. In den einzelnen Spalten der Tabellen, die auch Attribute genannt werden, werden die Eigenschaften der Daten (Attributwerte) abgelegt. Dadurch wird die gleiche Struktur aller Datensätze in der Datenbank gewährleistet. Die nachfolgende Abbildung zeigt nochmal die Zusammenhänge der verschiedenen Begriffe.

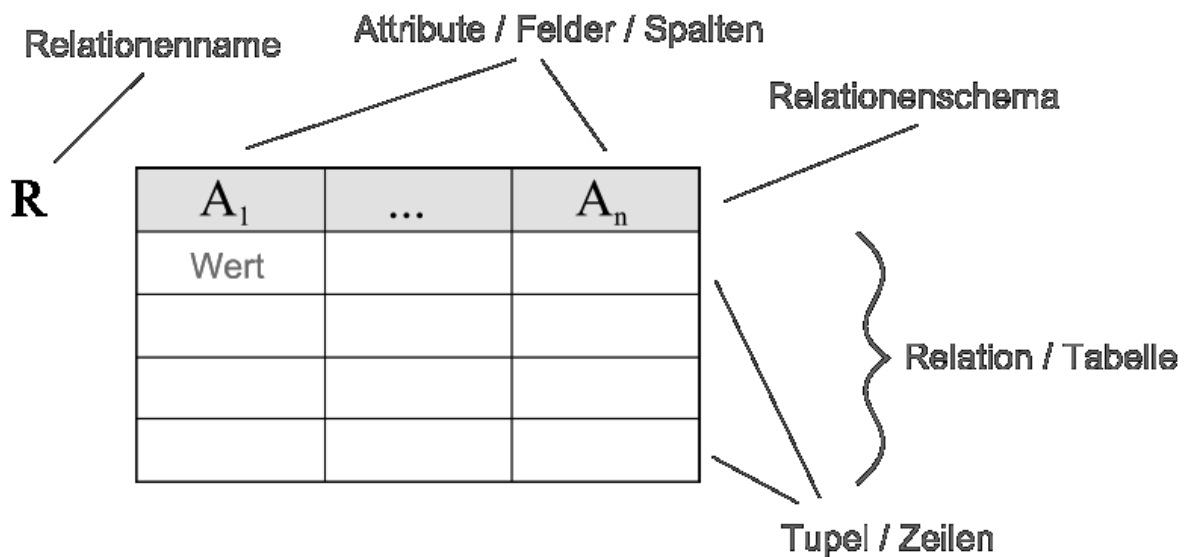


Abbildung 8: Aufbau des relationalen Datenbankmodells [Wik12]

Das wichtigste Attribut in einer Datenbank ist der Schlüssel. Dieser kann sich aus mehreren oder einem Attribut zusammensetzen. Die Aufgabe der Schlüssel ist die eindeutige Identifizierung eines Datensatzes in der Tabelle.

Wichtige Formen zur Überführung eines Modells in das relationale Datenmodell sind die sogenannten Normalformen. Unter Normalisierung des Datenschemas versteht man die schrittweise Zerlegung von Relationen. Dies hat das Ziel spätere

Redundanzen innerhalb des Datenschemas zu vermeiden. Die Normalisierung kommt vor allem beim Design einer relationalen Datenbank zum Einsatz. Die ersten drei Normalformen von insgesamt fünf sind jedoch die gebräuchlichsten. Mit Hilfe von Schlüsselattributen können dann auch Beziehungen über den Relationen erstellt werden. Dies geschieht genauer gesagt mit Fremdschlüsseln. Beziehungen können sowohl zwischen einer oder mehreren Tabellen erstellt werden. So kann später zwischen drei verschiedenen Typen von Beziehungen unterschieden werden:

- 1 – zu – 1 – Beziehungen
- 1 – zu – n – Beziehungen
- m – zu – n – Beziehungen

Mit Hilfe von SQL können Informationen aus der Datenbank abgefragt werden. Durch die Verbindung der Relationen durch Beziehungen können auch Informationen aus mehreren Tabellen in einer Abfrage kombiniert werden.

Relationale Systeme sind heutzutage von der verwendeten Technik und Umsetzung sehr gut ausgereift. Zu den weiteren Stärken dieses Modells gehört die effiziente und schnelle Verarbeitung von großen Datenmengen. Nachteilig daran ist, dass nur einfach strukturierte Datentypen verwendet werden können. Dies wirkt sich meist auch negativ auf den Datenbankentwurf aus, wenn man keine komplexen Datentypen zur Verfügung stehen hat. Auch wird die Abfrage über mehrere Tabellen dadurch komplexer. Weiterhin ist es ziemlich schwierig das relationale Datenmodell in eine objektorientierte Datenbank zu überführen. Dies liegt zu einem daran, dass es keine ausgereiften Tools zur Überführung gibt und zum anderen der reine Export der Daten nicht möglich ist.

4.3 Objektorientierte Datenbanksysteme

Objektorientierte Datenbanksysteme (OODBS) sind seit den 1980er Jahren auf dem Markt verfügbar und zählen somit zu den neueren Datenbankkonzepten. Jedoch nehmen sie heutzutage keine dominierende Rolle auf dem Datenbankmarkt ein. Mit diesem Konzept versucht man die heutigen komplexeren Probleme in ein Datenbankmodell zu überführen. Jedoch liegt dem OODBS kein einheitliches Datenmodell zu Grunde. So wurde versucht für Entwickler und Anwender einen

einheitlichen Standard zu schaffen wie zum Beispiel SQL für relationale DBS. Dieser Aufgabe nahm sich das Object Database Management Group (ODMG) an. Ziel war es also ein einheitliches Modell für objektorientierte Datenbanken zu schaffen. Dieser Standard umfasst sowohl neben dem Objektmodell, eine Objektdefinitionssprache (ODL), eine Objektabfragesprache (OQL), sowie Sprachanbindungen an C++ und Smalltalk. Als zentrale Einheit wird das Objekt angesehen. Objekte können Attribute, Methoden und Verweise auf andere Objekte enthalten. Die Definition der Objekte erfolgt über Klassen. Die Aufgabe der ODL ist die Beschreibung der Typen des Datenmodells. Neben der Kapselung, Zusammenfassung von Daten und Methoden, lassen sich weitere OO – Konzepte finden. Implementiert wurden unter anderem die Konzepte der Vererbung, des Überladens oder des automatischen Bindens. Auch komplexe Datentypen wie Array oder Listen sind in diesem Modell implementiert worden. Die OQL soll dabei die Aufgaben von SQL übernehmen. Beispiele für objektorientierte Datenbanken sind unter anderen:

- db40
- SmallSQL
- Zope Objekt Database
- JDataStore
- O2
- ObjectStore
- Objectivity/DB

Zu den Stärken dieses Systems gehört die Möglichkeit zur Schaffung eigener Datentypen, Methoden und Klassen. Diese können sowohl vom Standarddatentyp oder vom komplexen Datentyp seien. Ein weiterer Vorteil lässt sich aus der Objektidentität ableiten. Dadurch ist es jederzeit möglich die Attributwerte eines Objektes zu ändern. Aufgrund dieses Mechanismus kann ein Objekt nur zu einer einzigen Klasse gehören. Zu den Schwächen der Systeme gehört, dass die verwendeten Entwurfsmodelle und Entwurfsansätze noch nicht ausgereift sind. Dies liegt vor allem an der fehlenden Etablierung auf dem Datenbankmarkt. Ebenfalls sind OODBS nicht für große oder einfach strukturierte Daten geeignet. In diesem Falle würden Performanceeinbußen gegenüber relationalen Datenbanken zu Buche schlagen.

4.4 Objektrelationale Datenbanksysteme

Mitte der 1990er Jahre setzten sich immer mehr OO – Programmiersprachen durch. Durch diesen Erfolg konzipierte man ebenfalls auch OO – Datenbanksysteme, jedoch mit bescheidenerem Erfolg. Als Alternative dazu versuchte man, das relationale Datenmodell mit OO – Konzepten zu ergänzen. So wurde das objektrelationale Datenbankmodell geboren. Wie schon in anderen Kapiteln erwähnt fand diese Entwicklung ebenfalls im Bereich von SQL statt. Dieses Konzept versucht die Vorteile des OODBS zu nutzen und die Nachteile der RDBS auszumerzen. Somit sind objektrelationale Datenbanken keine wissenschaftliche Neuheit sondern vielmehr ein Produkt verschiedener Datenbankhersteller. Dabei liegt das einfach strukturierte Modell der Relationen zu Grunde. Dieses wird mit den komplexen Strukturen der Objektorientierung versucht zu kombinieren. So werden im objektrelationalen Modell versucht, komplexe Objekte in Tabellen abzuspeichern. Über Referenzen können auch in diesem Modell Objekte miteinander verbunden werden. Um komplexe Sachverhalte nachzubilden, müssen auch entsprechenden Konzepte zur Beschreibungen für das Modell und die Struktur einer Datenbank vorhanden seien. Dabei bilden die einfach strukturierten Datentypen der relationalen Datenbank die Grundlage. Aufbauend darauf werden komplexe Datentypen und Typkonstruktoren integriert. Das Schlüsselkonzept objektrelationaler Datenmodelle sind strukturierte Typen. Dadurch können die Prinzipien der Vererbung und der Kapselung realisiert werden. Aktuelle Systeme sind IBM DB2, Intersystems Cache oder PostgreSQL.

5 Geodatenbanksysteme

Erst durch die zunehmende Entwicklung von GI – Systemen in den 1990er Jahren und der Schaffung von einer offenen GIS – Welt wurde der Weg für Geodatenbanken geschaffen. Eine Geodatenbank ist ein Datenbanksystem, das die Speicherung von Geodaten und die Bearbeitung räumlicher Anfragen in hinreichender Weise unterstützt [Bri08].

Mit der Entwicklung von objektrelationalen DBMS und der Möglichkeit zur Definition neuer Datentypen erkannte man in der GIS – Branche, dass sich diese Datenbanksysteme besonders gut zur Speicherung von Geodaten nutzen lassen. Somit kam man zur Einsicht, dass die Verwendung objektrelationaler DBMS zur Verwaltung von Geodaten in Geoinformationssystemen die beste Lösung sei. Zu den heute bekanntesten Lösungen zählen im kommerziellen Bereich IBM Informix Spatial Data Blade, IBM DB2 Spatial Extender, Oracle Spatial und aktuell auch MS SQL Server 2008. Im Bereich der Open Source Produkte findet man folgende Datenbanklösungen, PostgreSQL / PostGIS und MySQL. Mit MySQL taucht in dieser Liste auch ein relationales Datenbanksystem zur Speicherung von Geodaten auf. Diese relationalen Systeme verwenden verschiedene Mechanismen zu Speicherung der Informationen. Die 3 standardmäßigen Techniken zum Ablegen von Geodaten werden nun im Näheren kurz erläutert.

Die Zerlegung der geometrischen Attribute stellte den ersten Ansatz da. Dabei wird versucht die verschiedenen Informationen auf mehrere Tabellen aufzuteilen. So kann es vorkommen, dass eigentlich zusammengehörige Informationen auf mehrere Tabellen aufgeteilt werden. Problematischer wird es noch wenn größere Datenbestände oder komplexere Datenmodelle zur Anwendung kommen. Die Umsetzung in SQL lässt sich dementsprechend auch schwieriger oder sogar gar nicht gestalten. Somit lässt sich resümieren, dass die Aufteilung sich weder funktional noch aus ökonomischer Sicht rechtfertigen lässt.

In älteren Geoinformationssystemen lässt sich noch eine andere Möglichkeit finden Geodaten in der Datenbank abzulegen. Diese sieht eine getrennte Speicherung von geometrischen und alphanummerischen Attributen vor. Die alphanummerischen Daten sollen dabei in der relationalen Datenbank abgelegt werden, wo hingegen die

geometrischen Attribute in einer GIS – spezifischen Datei abgelegt wird. Die Verbindung der beiden verschiedenen Datensätze erfolgt ebenfalls über einen Schlüssel. Gegen diese Umsetzung spricht jedoch, dass die Geodaten nur mit Hilfe eines bestimmten GIS verwendet werden können. Auch gehen so alle Vorteile von relationalen Datenbanksystemen verloren. Auch können erhöhte Betriebskosten in diesem Falle auftreten.

Die letzte Variante stellt die Verwendung von sogenannten BLOBs (Binary Large Objects) dar. In diesem Datentyp können beliebige Binärdaten abgelegt werden. Somit besteht die Möglichkeit Geometrien in binärer Form abzuspeichern. Von allen 3 vorgestellten Lösungen bietet diese Variante die meisten Vorteile. Alle Daten werden in einem Objekt oder einer Datei gehalten und auch die Vorteile eines Datenbanksystems, wie Zugriffskontrolle oder Mehrbenutzerbetrieb bleiben erhalten. Aus Sicht des Datenbanksystems ist es schwierig die Binärinformationen richtig zu interpretieren, denn für das DBS stellt das BLOB nur eine Folge von Nullen und Einsen dar. So können keine effizienten Algorithmen zur Anfragebearbeitung bereitgestellt werden. Auch geometrische Operationen lassen sich aus diesen Gründen nicht umsetzen. Auch widerspricht die Verwendung von BLOBs den offenen GIS – Standard, da externe Programme auf die Nutzung eines GI – Systems angewiesen ist.

Somit lässt sich zusammenfassen, dass auch relationale Datenbanken für den Einsatz zur Speicherung von Geodaten geeignet sind. Jedoch sind damit ein erhöhter Aufwand und Kosten, verminderte Effizienz und eine erschwerte Abfragebearbeitung verbunden. Damit verbleibt als einzige Lösung nur die Verwendung eines objektrelationalen Datenbanksystems. Diese Systeme bieten entsprechende Methoden und Datentypen an und ermöglichen einen einfachen Nutzerbetrieb. Jedoch sind für eine optimale und effiziente Anfragebearbeitung spezielle Erweiterungskomponenten notwendig. Diese Komponenten sind jedoch von Hersteller zu Hersteller unterschiedlich. Dabei begibt man sich jedoch in eine bestimmte Abhängigkeit von einem bestimmten Datenbankhersteller. Jedoch wird dieses Problem ein wenig durch die offenen und allgemeinen Standards aufgelockert.

5.1 Anforderungen Geodatenbanksysteme

Wie in der Definition schon festgehalten wurde, müssen Geodatenbanken entsprechende Datentypen für Geodaten bereitstellen und diese auch unterstützen können. Aus dieser Forderung ergeben sich noch weitere verschiedene Anforderungen an Geodatenbanken. Eine Geodatenbank muss geometrische und topologische Datentypen für den Anwender bereitstellen. Diese müssen ebenfalls dazu geeignet sein Geometrie- und Topologieattribute repräsentieren zu können. Dadurch ist es dann später möglich Linien, Punkte, Polygone und Körper in der Datenbank abzulegen. Aus dieser Forderung ergibt sich zwangsläufig auch die Nachfolgende. Das Geodatenbanksystem muss für die angebotenen Datentypen entsprechende Methoden bereitstellen. Zu diesen Methoden gehören zum Beispiel die Berechnung der Länge eines Streckenzuges, der Schnitt zwischen 2 Flächen oder das Prüfen ob bestimmte Geometrien in einer anderen enthalten sind. Auf diese Methoden sollte der Nutzer dann auch in der Anfragesprache des Datenbanksystems zugreifen können. Sollte eine Anfrage eine oder mehrere Operationen mit Raumbezug enthalten, wird die Anfrage auf eine oder eine Folge von Basisanfragen zurückgeführt. Beispiele dafür sind die Punkteabfrage oder die Rechteckabfrage (Clipping). Die Verarbeitung dieser einfachen Basisanfragen oder anderer geometrischer Grundoperationen sollte auch in einer entsprechenden Geschwindigkeit erfolgen. Dies kann aber nur funktionieren wenn entsprechende Algorithmen oder Datenstrukturen zur Verfügung stehen. Die Forderung kann teilweise durch die Verwendung von Indizes für das Suchen von bestimmten Geodaten erreicht werden und zum anderen sollten schnelle Algorithmen in das DBMS integriert und implementiert sein. Alle in das System eingebundenen Datentypen, Methoden und Algorithmen müssen den Forderungen der verschiedenen GIS - Standards genügen Dies ermöglicht eine Interoperabilität zwischen den verschiedensten Applikationen.

5.2 Open-Source Geodatenbanksysteme

In immer mehr mobilen Endgeräten werden heutzutage Geodaten verarbeitet. Egal ob Smartphone, Handy, Navigationsgeräte oder Digitalkameras, all diese technischen Geräte verfügen meist schon über eingebaute GPS – Empfänger. Mit zusätzlicher Internetverbindung können diese Geräte auch auf Webdienste im Internet zugreifen und so die gewonnenen Informationen für die Nutzer entsprechend aufwerten. Um diesen Service bereit zu stellen, werden meist Geodatenbanken eingesetzt. Hier kommen meist jedoch keine kommerziellen Produkte sondern frei zugängliche Systeme zum Einsatz. Nachfolgend werden die 3 bekanntesten Open-Source Geodatenbanksysteme vorgestellt und im nächsten Kapitel miteinander verglichen. Zu diesen gehören PostgreSQL, MySQL und Spatialite.

5.2.1 PostgreSQL / PostGIS

PostgreSQL ist ein objektrelationales Open – Source Datenbankmanagementsystem. Die Entwicklung begann bereits in den 1980er Jahren und ab 1997 wurde die Entwicklung völlig der Open – Source Community überlassen. Heutzutage gehört dieses Datenbanksystem neben MySQL zu den bedeutendsten im Open – Source Bereich.

Mit PostGIS begann dann auch die Entwicklung von Open-Source Geodatenbanksystemen. PostGIS ist eine Erweiterung der PostgreSQL Datenbank und wurde von der Refractions Research entwickelt. Schon schnell bemerkte man bei der Entwicklung, dass die vorhandenen Datentypen für die verschiedensten Geometrien nicht ausreichen würden. Wie schon im vorherigen Kapitel beschrieben waren diese beiden Techniken nicht die Effizientesten. Dadurch ergab sich dann die Entwicklung von neuen Geometriedatentypen. Diese neu entwickelten Methoden und Datentypen wurden in der GEOS-Bibliothek zusammengefasst. Diese Bibliothek wurde in C++ geschrieben und ist eine Portierung der JAVA JTS Topology Suite. Diese enthält ein räumliches Objektmodell und die dazu nötigen geometrischen Funktionen und Datentypen. Die GEOS-Bibliothek kann auch von anderen Datenbanksystemen verwendet werden. Zu diesen gehört unter anderem auch Spatialite. In der im Jahre 2003 erschienenen Version 0.8 wurden auch das Simple

– Features – Access (ISO 19125) implementiert. In den kommenden Jahren wurde immer wieder das Datenmodell beziehungsweise das Datenformat neu überarbeitet. Im Jahre 2006 wurde der SQL/MM – Standard implementiert. Dieser fand jedoch nicht den Einzug in die GEOS – Bibliothek. Der Vorteil dieses Standards gegenüber der ISO 19125 ist die größere Menge an geometrischen Objekten. Die aktuelle PostGIS – Version ist im Moment die Version 3.3.4.

PostgreSQL besitzt ein eigenes Geometriemodell, das nicht konform ist mit denen der OGC oder der ISO. So bietet das Postgre – Datenmodell die eigenen Datentypen Point, Line oder Box an, die in ihrer Funktionalität noch beschränkt sind. In der Erweiterung PostGIS sind alle wichtigen Standards der OGC, ISO und die neusten Versionen von SQL implementiert. Auch das PostGIS - Klassenmodell ist konform zu allen zuvor genannten Standards.

Der Simple – Feature Access – Standard ist dabei komplett im Datenbanksystem integriert. Nur teilweise implementiert ist der SQL / MM-Standard. Des Weiteren sind zusätzlich noch selbstprogrammierte Funktionen der Entwickler hinzugefügt worden. Beispielhaft zu nennen sind die Funktionen zum Konvertieren in das SVG oder KML Format oder aber auch die Bildung von Polygonen wird dadurch ermöglicht.

Bei Postgre setzt man auf einen R – Baum als räumlichen Index. In der Erweiterung PostGIS wird hingegen der R – Baum um den Generalized Search Tree (GiST) erweitert. Dies liegt zum einem an der besseren Resistenz gegenüber Nullwerten. Und zum andern ergibt sich daraus das nur Teile eines Objektes als Indexwert gespeichert werden. Die räumlichen Indizes kommen nur zum Einsatz für Funktionen, die mit den MBRs rechnen. Nächste Nachbarn – Anfragen werden jedoch nicht von PostGIS unterstützt.

PostGIS unterstützt ebenfalls die Verwendung von räumlichen Bezugssystemen, allen voran die EPSG – Bezugssysteme. Diese sind in einer internen Systemtabelle „SPATIAL_REF_SYS“ abgelegt. Somit ist es möglich mit Hilfe von Funktionen Längen und Flächen über Koordinaten zu berechnen. Umrechnungen von Gradangaben in metrische Längenangaben sind bisher in direkter Form noch nicht möglich. Dafür gibt es die beiden Funktionen DISTANCE_SPHEROID, zur Berechnung des Abstandes zwischen 2 Punkten und LENGTH_SPHEORID zur

Bestimmung der Länge einer linearen Geometrie. Für die Flächenbestimmung stehen jedoch keine solcher Hilfsfunktionen zur Verfügung.

Aggregatfunktionen werden durch das normale SQL bereitgestellt. Diese lassen sich jedoch nicht auf geometrische Datentypen anwenden. Auch in den vorgestellten Standards lassen sich die geometrischen Aggregate nicht finden. Deshalb mussten die Entwickler von PostGIS sich selber Gedanken um die Entwicklung solcher Funktionen machen. Dabei ging man einen komplett anderen Weg als die Entwickler von Oracle oder MySQL. Die Funktion Collect übernimmt dabei mehrere Aufgaben. Mit ihrer Hilfe ist es möglich aus verschiedenen Beziehungen von Geometrien eine Geometriekollektion zu erzeugen. Gespeichert wird in dieser Kollektion die Geometrie der Beziehung. Dadurch können jetzt alle geometrischen Operationen auf dieser Kollektion angewendet werden.

Wie erwähnt setzten sich Rasterdaten aus Pixel zusammen, die meist in Form von Luftbildern oder analogen Karten vorliegen. Erstmals finden in der Version 2.0 alle Funktionen zur Unterstützung von Rasterdaten Einzug in das Hauptprogramm. Zuvor stand nur die Erweiterung WKTRaster als einzige Erweiterungskomponente zu Verfügung. Damit wird ermöglicht einige Operationen zur Analyse der Vektordaten ebenfalls auf Raster- oder gemischte Datensätze anzuwenden. Zum Laden von Bilddaten benötigt PostGIS die freie Programmibliothek GDAL. Im Vordergrund dieses Projektes steht deshalb die Konvertierung von verschiedenen Formaten sowie Operationen für Rasterdaten. Mit der Hilfe des Skriptes „raster2pgsql.py“ können die Rasterdaten nach PostGIS importiert werden. Demnach ist es auch notwendig Python auf seinem Rechner installiert zu haben. Alle Rasterdaten werden in einem einzigen Datentyp namens „Raster“ gehalten. All dies macht PostGIS im Rasterbereich von GDAL und Python abhängig.

Die Verarbeitung von großen Datenmengen ist in GIS ein wichtiger Aufgabenbereich. Hier stößt PostGIS zum ersten Mal in seinen Grenzbereich. Nach langer Recherche ließ sich herausfinden, dass die Verarbeitung von riesigen Datenmengen nur relativ langsam bis gar nicht vonstattengeht. GIS – Analyseoperationen von Daten mit einer Speicherkapazität von bis zu 4GB und mehr sind kaum in einer akzeptablen Zeit zu erreichen. Jedoch lässt sich mit genügender Rechenleistung die Verarbeitungszeit

um einiges senken. Laut [ObH11] ist PostGIS ein Versuch Geoinformationssysteme durch räumliche Datenbanken zu ersetzen.

PostGIS wird häufig im Zusammenhang mit OGC getreuen Geowebdiensten eingesetzt. Dies führt häufig zum Einsatz in Geodateninfrastrukturen in Zusammenarbeit mit einem Webmap – Server.

5.2.2 MySQL

MySQL nimmt heute auf dem relationalen Datenbankmarkt eine der bedeutendsten Positionen ein. Die Entwicklung dieses Produktes begann jedoch schon im Jahre 1994 bevor es 1997 der Öffentlichkeit vorgestellt wurde. Von Beginn an wurde Wert auf die Verarbeitung von großen Datenmengen und auf schnelle Performance gelegt. Seit dem Jahr 2010 steht MySQL unter der Riege von Oracle, die die Firma Sun Microsystems aufgekauft hatte. Erst mit der Bereitstellung der Version 4.1 fanden räumliche Erweiterungen Einzug ins System. Diese Erweiterung war jedoch nur gültig für Tabellen, die die MyISAM-Storage Engine verwendeten. Auch wurde hier schon der Simple – Feature Access – Standard implementiert. Jedoch fehlen hier einige wichtige Pakete. Ab der Version 5 werden nun auch durch die Subspeichersysteme (Storage-Engines) InnoDB, NDB, BDB und ARCHIVE Spalten für das Ablegen von Geodaten bereitgestellt. Topologische Operationen fanden dann in den späteren Versionen Einzug. Diese werden mit dem Präfix ST_ versehen und arbeiten jetzt mit den exakten mathematischen und geometrischen Formen. Vorher wurden die Daten nur mit den Hüllgeometrien verarbeitet.

Die in MySQL zur Verfügung stehenden Datentypen sind die gleichen die durch das Simple – Feature Modell bereitgestellt werden. Dabei werden Typen unterschieden die nur einzelne Geometriewerte abspeichern können wie

- GEOMETRY
- POINT
- LINESTRING
- POLYGON

und Datentypen, die eine Sammlung von Werten aufnehmen können:

- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION.

GEOMETRY und GEOMETRYCOLLECTION können verschiedene Werte oder mehrere Objekte beliebiger Typen aufnehmen. Die anderen Einzelwerttypen beziehungsweise Collection – Typen können nur Werte eines bestimmten Typen speichern.

Um jetzt die Daten intern zu verwalten stehen die 2 Repräsentationsformen WKT und WKB ebenfalls in MySQL zur Verfügung. Diese beiden Werte werden mit MySQL – spezifischen Funktionen ergänzt. Intern verwendet MySQL jedoch ein Schema das keinem der beiden Formate ähnelt. MySQL bietet mehrere Funktionen an, die das WKT - Format in eine Geometrie umwandeln. Dabei gibt es Funktionen, die als Eingabeparameter die WKT – Darstellung erwarten. Andere nutzen dafür aber auch optional einen Georeferenzsystembezeichner (SRID). Eine dieser Funktionen ist zum Beispiel `GeomFromText()`. Im Simple – Feature Modell sind auch Funktionen integriert, die MySQL nicht implementiert hat. Diese optionalen Funktionen stellen Polygon- oder MultiPolygon – Werte bereit. Aus den WKT Präsentationen von Linienzügen oder Ringen werden diese Geometrien erzeugt. Die Verarbeitung von WKB unterscheidet sich ein wenig im Vergleich zur Darstellung im WKT Format. Die Funktion hier erwartet als Übergabeparameter ein BLOB. Auch im Falle der WKB kann optional wieder ein Georeferenzsystembezeichner übergeben werden. Ansonsten gibt es keine weiteren Unterschiede zwischen den beiden Repräsentationsformen in der internen Verarbeitung. Jedoch liefert MySQL einige nicht dem Standard der OGC entsprechenden Funktionen. Diese erstellen WKB – Darstellungen aus Geometriewerten im BLOB – Format ohne eine SRID zu enthalten. Eine dieser Funktionen ist zum Beispiel `GeomFromWKB()`.

Wenn die Datenbank mit Daten gefüllt ist, können diese mit Hilfe von SQL abgefragt werden. Dafür stehen in MySQL diverse Funktionen bereit. Die Funktionen lassen in 4 Gruppen untergliedern. Die Gruppen werden nach den Operationen, die ausgeführt werden, gebildet.

- Funktionen, die Geometrien zwischen verschiedenen Formaten konvertieren
- Funktionen, die Zugriff auf qualitative oder quantitative Eigenschaften einer Geometrie zulassen
- Funktionen, die Beziehungen zwischen zwei Geometrien beschreiben
- Funktionen, die aus vorhandenen Geometrien neue erzeugen

[nach MyS12]

Analysefunktionen für Geodaten können vielseitig eingesetzt werden. Zum einen können sie in interaktiven SQL – Programmen oder in Anwendungsprogrammen zum Einsatz kommen.

Suchoperationen nehmen eine besondere Stellung ein, da diese intern anders arbeiten. Normalerweise werden die Suchen in Datenbanken über bestimmte Indizes erleichtert. Dieses Prinzip lässt sich auch auf Geodatenbanken anwenden. Bisher wurden verschiedenen Methoden für eine Suche mit Indizes entworfen. Zu den bekanntesten Funktionen gehören die Punkt- und die Bereichsabfrage. Die Punktabfrage sucht nach allen Objekten, die einen bestimmten Punkt enthalten. Die Bereichsabfrage hingegen sucht nach allen Objekten, die einen bestimmten Bereich überdecken. In MySQL kommen R – Trees zum Einsatz. Diese sind zusätzlich mit einem quadratischen Split ausgestattet um Spalten in Geodatenbanken besser zu indizieren. MySQL lässt die raumbezogenen Indizes mit Hilfe von MBRs bilden. Das MBR ist für die Geometrie, das kleinste umschließende Rechteck. Die Verwendung dieser Funktion bringt Effizienzvorteile gegenüber den Operationen, die mit den exakten Geometrien rechnen. Es wird den Nutzern ebenfalls angeboten normale Indizes auf Spalten anzulegen. Die R – Trees finden jedoch nur Anwendung in der Speicher – Engine MyISAM. Bei allen anderen Engines ist die Unterstützung von räumlichen Indizes eingeschränkt.

Jedoch gibt es in MySQL auch einige Funktionen und Operationen, die noch nicht vollends oder gar nicht umgesetzt werden. Beispielsweise werden keine zusätzlichen Metadaten – Views von Seiten MySQL`'s angeboten. Es werden jedoch von der OGC andere Views empfohlen. Ein anderes Problem taucht im Zusammenhang mit der Funktion Length() auf. Diese Funktion berechnet die Länge von LineStrings und MultiLineStrings, kann aber in MySQL nur mit dem Namen GLength() verwendet werden. Dies liegt daran, dass es schon eine SQL – Funktion mit diesem Namen gibt. Auch stehen in MySQL keine Funktionen zur Berechnung metrischer Längen bereit. Weiterhin lassen sich auch keine Flächen über geographische Koordinaten berechnen.

MySQL setzt generell auf den Simple Feature Access – Standard. Dieser wurde jedoch nicht komplett implementiert, sondern man setzt nur auf eine Untermenge dieses Schemas. So fehlen zum Beispiel die Verschneidungsoperation Union() und Intersection(). Als geometrische Operationen stehen nur Buffer und Konvexe Hülle zur Auswahl. Auch die Testfunktionen IsRing und IsSimple sind nicht einsatzfähig. Topologische Prädikate können auch nur grundlegend umgesetzt werden. Auf der Grundlage der MBRs werden die topologischen Prädikate geprüft. Dadurch mangelt es jedoch an der Exaktheit der Ergebnisse. Aufgrund dessen können bei Selektions- oder Verbundanfragen nur die Filterschritte ausgeführt werden. Auch bei den Berechnungen mit Koordinatensystemen gibt es einiges zu beachten. So wird die SRID in der Datenbank abgelegt, jedoch wird diese bei Berechnungen im mehrdimensionalen Raum nicht berücksichtigt. Die SRID legt die Eigenschaften eines Koordinatensystems fest. Bei Berechnungen wird zur Vereinfachung ein ebenes zweidimensionales Koordinatensystem verwendet.

Für MySQL steht ebenfalls die GDAL – Bibliothek zur Verfügung. Dadurch ist es wie in PostGIS möglich, Rasterdaten zu öffnen und danach weiterzuverarbeiten. Damit stehen auch Verarbeitungsroutinen zur Verfügung.

Für den Einsatz von großen Datenmengen eignet sich MySQL besonders gut, da dies ein Schwerpunkt bei der Entwicklung war. Jedoch funktioniert das nur bei einfach strukturierten Daten, wie zum Beispiel XML – Daten, relativ gut. Bei

Analyseoperationen von Datenmengen im GB – Bereich stößt es ebenfalls wie PostGIS an seine Grenzen. Deshalb sollte bei der Verarbeitung von großen Datenmengen von vorn herein klar sein, welche Prozesse intern ablaufen müssen, damit die Aufgabe erfüllt werden kann.

Das Haupteinsatzgebiet der MySQL – Datenbank sind räumliche Webanwendungen.

5.2.3 SQLite / SpatiaLite

SQLite geht einen komplett anderen Weg als die vorangegangenen beschriebenen Datenbanksysteme. SQLite ist eine freie Programmbibliothek und ein Geoinformationssystem zugleich. In diesen ist ein serverloses relationales Datenbanksystem integriert, das auf einer SQL – Engine aufbaut. Aufgrund dessen kann nicht der volle Funktionsumfang von SQL genutzt werden. Letzteres ist der entscheidende Unterschied gegenüber den anderen Datenbanksystemen. Die erste Version dieses Datenbanksystems erschien im März 2008. In der Nachfolgerversion wurde auch die schon in PostGIS eingesetzte Programmbibliothek GEOS in die Datenbank implementiert. Damit genügt die Datenbank vollends dem Simple – Feature Access – Standard. Ergänzend wurde auch noch die PROJ.4 Bibliothek ins System eingearbeitet. Dadurch ist es möglich verschiedene Projektionen aus der Kartografie nachzubilden. Im Laufe der Weiterentwicklung fanden dann auch das SQL – Math sowie eine Verarbeitungsroutine für EXIF – GPS Daten Einzug in die Datenbank. In den letzten Versionen wurden auch Konvertierungsfunktion für GML und KML implementiert.

Auch SpatiaLite bietet einen großen Umfang an Funktionen für Geodaten. Jedoch fehlen hier einige Funktionen aus dem SQL – Standard. Dies liegt daran, dass die gesamte Datenbank in nur einer einzelnen Datei abgelegt ist. Aufgrund dessen schlägt die Anwendung der Funktionen GRANT und REVOKE fehl. Hier werden nur die Zugriffsrechte der Datenbankdatei geltend gemacht. Es gibt auch noch weitere Funktionen, die nicht wie gewohnt arbeiten. Zu dieser gehört ALTER TABLE. Diese Funktion erlaubt nur RENAME und ADD COLUMN. Des Weiteren lassen sich DELETE, INSERT und UPDATE nur auf Views anwenden. Ganz und gar fehlt der FOR EACH STATEMENT – Trigger. Da in SpatiaLite die gleiche Bibliothek zur

Verarbeitung von Geodaten genutzt wird wie in PostGIS, stehen auch hier alle Funktionen und Typen des Simple – Feature Access – Standards zur Verfügung. Auch die PROJ.4 – Bibliothek steht neben der GEOS – Bibliothek zur Verwendung bereit. Somit ist auch die Funktion TRANSFORM enthalten mit der es möglich ist, ein Koordinatensystem in ein anderes zu transformieren. Diese Funktion wird durch die SRID gesteuert. Ein weiteres Feature von SpatiaLite ist die Bereitstellung von sogenannten „Virtual Tables“. Damit ist es möglich verschiedene Module zu entwickeln, die auf verschiedene Datenquellen zugreifen können. Diese „Virtual Tables“ werden als normale Tabellen im System hinterlegt und können dann in jedem beliebigen SQL – Statement aufgegriffen werden. Einige solcher Module stehen schon beim ersten Start der Datenbank zur Verfügung. Zu diesen gehören VirtualShape und VirtualText. VirtualShape dient zum Zugriff auf Shape – Dateien. VirtualText erlaubt den Zugriff auf CSV – Tabellen. Dadurch wird dem Nutzer das Importieren dieser Daten erspart. Auch eine Routing – Funktion namens VirtualNetwork steht bereit. Diese setzt den Dijkstra – Algorithmus zur Berechnung der kürzesten Wege in einem Graphen um.

Mit RasterLite steht für SQLite / SpatiaLite eine einfache C – Erweiterung zur Verarbeitung von Rasterdaten zur Verfügung. Aufbauend auf die GDAL Version 1.7.0 erlaubt der RasterLite Treiber das Lesen und Erstellen von rasterorientierten Datenformaten. In der Standardbibliothek sind 5 nützliche Tools enthalten. All diese sind einfache CLI – Werkzeuge (Command Line Interface). Zur Benutzung müssen diese über die Kommando – Shell über spezielle Befehle aufgerufen werden.

- rasterlite_load
- rasterlite_pyramid
- rasterlite_topmost
- rasterlite_tool
- rasterlite_grid

Die RasterLite Bibliothek implementiert ebenfalls eine einfache C - API. Dies bedeutet, dass RasterLite in eigene C oder C++ Programme eingebettet werden kann. Durch geeignete Bindings ist es auch möglich es mit anderen Programmiersprachen wie zum Beispiel Python zu verbinden.

5.3 Kommerzielle Geodatenbanksysteme

Heutzutage ist es sehr schwer zwischen kommerziellen und freien Produkten zu unterscheiden. Alle nachfolgenden Datenbanksysteme gehören zu kommerziellen Produkten, jedoch sind die Spatial Erweiterungen kostenfrei. Die Spatial Erweiterungen können meist frei heruntergeladen werden und ermöglichen das Visualisieren der Geodaten mit Hilfe eines Viewers. Eine Verwaltung in einer Datenbank kann dadurch jedoch nicht realisiert werden.

5.3.1 Oracle Spatial

Im Jahre 1978 erschien die erste Version des relationalen Datenbanksystems Oracle der Firma Relational Software Inc. unter dem Namen Oracle V.2. Im Laufe der Jahre wurden das Datenbanksystem für Unix – Plattformen und Mehrprozessorarchitekturen bereitgestellt. Mit der Version 8.0 erschien erstmals eine Oracle Datenbank mit einem objektrelationalen Ansatz. Fortan lag das Hauptaugenmerk auf der Unterstützung von Internetanwendungen. Solche Versionen erkennt man an dem Zusatzkürzel i im Namen. Die zuletzt veröffentlichte Version ist seit 2007 unter dem Namen Oracle 11g verfügbar. Für die Verarbeitung von Geodaten wurde in den 1990er Jahren Produktumgebungen mit einer solchen Spezialisierung veröffentlicht. Seit der Veröffentlichung der Oracle Version 8.1.5 erhielt die Erweiterung den Zusatz Spatial, so wie sie auch noch heute genannt wird. Seit dem Erscheinen dieser Version wird mit dem Spatial Zusatz der objektrelationale Ansatz verfolgt. In den nächsten Jahren wurden immer wieder Erweiterungen oder Erneuerungen bisher entwickelter Komponenten in Oracle Spatial integriert. So zum Beispiel in der Version 9, die die Verwendung geographischer Koordinatensysteme unterstützt, in der Version 10, die Speicherung der Topologie ermöglicht und georeferenzierte Rasterdaten unterstützt. Die Neuerungen in der Version 11 beziehen sich alle auf die Verarbeitung von 3D Geodaten [Bri08].

In Oracle stehen verschiedene Cartridges zur Verfügung, die die Funktionalität des Datenbankservers erweitern. Diese Software – Pakete bestehen zumeist aus benutzerdefinierten Funktionen und Typen. Zu den bekanntesten Erweiterungen gehören

- interMedia Option (Text, Audio, Image, Video, Locator)
- Spatial
- Time Series
- Visual Information Retrieval Options

Der Spatial Zusatz besteht aus zwei verschiedenen Paketen. Zum einem aus dem sogenannten Kern, der zu allen anderen Vorgängerversionen kompatibel ist und zum anderen aus einem zusätzlich erhältlichen Paket (Spatial Option). Letzteres stellt zusätzliche Funktionen bereit. Letzteres spielt in den folgenden Betrachtungen keine Rolle mehr. Das Hauptpaket besteht aus den vier Komponenten

- Schema, zur Beschreibung der Speicherung, Syntax und Semantik der unterstützten Datentypen
- Räumliche Indizierungsmechanismen
- Sammlung von Operatoren und Funktionen um Verschneidungsoperationen auszuführen
- Administrative Tools

In Spatial werden zwei verschiedene Geometriemodelle unterschieden. Zum einen gibt es das objektrelationale Konzept und zum anderen den relationalen Ansatz. Beide Modelle unterstützen dabei die von der OGC und ISO geforderten Standards und Normen. Das eigentliche Datenmodell hat einen hierarchischen Aufbau. Die Geodaten lassen sich dabei in Elemente, Geometrien und Layer unterscheiden. Geometrien setzen sich aus einem oder mehreren Elementen zusammen. Ein Element wird entweder aus einer Linie, einem Punkt oder einer Fläche gebildet. Diese vier Primitiven werden mit einer eindeutigen ID (Geometrie_id) gekennzeichnet. Layer gruppieren Geometrien mit gleichartigen Attributen. Das Modell erlaubt es auch räumliche Indizes für Geodaten anzulegen. Dabei unterstützt Spatial zwei verschiedene Arten von Indizes. So stehen der Quad – Tree und der R – Tree zur Auswahl. Der R – Tree umhüllt die Geometrie mit dem MBR. In einem Layer werden die MBRs mehrerer Geometrien hierarchisch geordnet. Die Koordinaten eines jeden einzelnen Elementes werden als eine Gleitkommazahl abgelegt. Damit

die Koordinaten vom Datenbanksystem richtig gedeutet werden können, ist eine Verbindung zu einem Koordinatensystem erforderlich. Ist kein Koordinatensystem vom Nutzer explizit angegeben, wird vom Datenbanksystem ein kartesisches Koordinatensystem standardmäßig ausgewählt. Für diesen Fall ist es nicht möglich mit metrischen Einheiten zu rechnen. Aus diesem Grund ist besser ein Koordinatensystem anzugeben. Vordefinierte Systeme findet man in der Tabelle MDSYS.CS_SRS, die über eine SRID – Nummer ausgewählt werden.

Die Indizes werden in speziellen Indextabellen (SDO_INDEX_TABLE) abgelegt. Beim Quad – Tree Index wird das Koordinatensystem, in dem die Geometrien eines Layers dargestellt werden, durch einen Algorithmus gleichmäßig aufgeteilt. Wie das Koordinatensystem unterteilt werden soll, kann durch den Nutzer persönlich eingestellt werden. Dadurch wird auch der Aufbau der Indextabelle ein wenig komplizierter gestaltet. Spatial erlaubt es unter anderem auch beide Indextypen gemeinsam zu verwenden. Damit muss der Nutzer selbst entscheiden welcher Indextyp für welche Anwendung und für welches System am besten geeignet ist.

In Oracle werden zwei verschiedene Modellansätze verfolgt. Zu den beiden Modellen gehört zum einen das objektrelationale Modell und zum anderen das relationale Modell. Das relationale Modell hat in der Praxis jedoch keine relevante Bedeutung und wird deshalb nicht näher beleuchtet.

Das objektrelationale Modell setzt sich aus verschiedenen Typen von Objekten zusammen. Zum einen gibt es einen Index und zum anderen Funktionen, die auf diese Objekte zugreifen. Alle in die Datenbank aufgenommenen Geometrien werden als Objekte der Klasse SDO_GEOMETRY in einer Tabellenspalte abgelegt. Diese Geometrieklasse setzt sich aus 5 verschiedenen Attributen zusammen. Die nachfolgende Tabelle zeigt nochmals übersichtlich alle 5 Geometrieattribute.

Bezeichnung	Datentyp	Beschreibung
SDO_GTYPE	Number	bestimmt den Typ des Objektes, z.B. Punkt, Linestring

SDO_SRID	Number	reserviert für zukünftige Erweiterungen
SDO_POINT	SDO_POINT_TYPE	für Speicherung von Punktdaten; beinhaltet die Koordinaten der Punkte (nur wenn nächsten 2 Elemente Null sind, ansonsten ignoriert)
SDO_ELEM_INFO	MDSYS.SDO_ELEM_INFO_ARRAY	enthält Triplets an Attributen, wie die Werte in SDO_ORDINATES interpretiert werden sollen; besteht Objekt nur aus einem Element, sind hier nur 3 Zahlen gespeichert
SDO_ORDINATES	MDSYS.SDO_ORDINATE_ARRAY	enthält die Koordinaten des Objektes in X,Y bzw. X,Y,Z-Paaren

Tabelle 2: Geometrieattribute der Klasse SDO_GEOMETRY [Bri08]

Mit Hilfe der Data Definition Language (DDL) können Datenstrukturen angelegt, geändert, beschrieben oder gelöscht werden. Die Data Manipulation Language (DML) hingegen bezieht sich auf die Inhalte der verwendeten Datenstrukturen. Neben der Klasse SDO_GEOMETRY gibt es noch eine weitere Geometrieklasse, die sich ebenfalls der SQL / MM Spatial Norm unterordnet. Die Klasse ist vom Aufbau einfacher gehalten, denn es enthält nur das Attribut „geom“ vom Typ SDO_GEOMETRY. Diese Klasse wird auch als Wrapperklasse bezeichnet. Die Klasse kann ebenfalls auch als Datentyp für Geometrieattribute genutzt werden. Weiterhin wird die Klasse aus Sicht von Oracle auch dazu genutzt um das Simple – Feature Modell umzusetzen.

Auf die Klasse SDO_GEOMETRY lassen sich natürlich auch verschiedenste Funktionen und Operationen anwenden. Die meisten Funktionen stammen aus dem Paket SDO_GEOM. Alle dieser Funktionen besitzen eine gewisse Toleranz als Parameterwert. Dieser Wert beschreibt die Genauigkeit der Ergebnisse. Es kann beim Aufruf von Funktionen auch eine Maßeinheit definiert werden. Jedoch muss zuvor ein entsprechendes Koordinatensystem (SRID) definiert sein. Die meisten Operationen bieten 2 Übergabeparameter für Geometrien an. Die Anordnung dieser Parameter beeinflusst die Geschwindigkeit der Abfrage. Oracle setzt den Index meist an die erste Stelle. Aus diesem Grunde sollte man aus Performancegründen immer die gesuchte Größe als erstes Argument verwenden.

Alle Operatoren in Spatial dienen dem Test von topologischen Beziehungen und Entfernungen zwischen den Geodaten. Keine der Operatoren ist einem bestimmten Paket zugehörig. Als Rückgabewert liefern die meisten Operatoren einen booleschen Wert. Standardmäßig wird die Zeichenkette true zurückgegeben.

Natürlich gibt es noch viele weitere Einsatzmöglichkeiten von Oracle Spatial. Das Paket SDO_SAM hält Funktionen für das Data Mining bereit, so zum Beispiel für das Bilden von Clustern. Auch für das Geocoding stehen verschiedene Varianten für die Umwandlung von Koordinaten in Adressen und umgekehrt zur Verfügung. Weiterhin können auch Körper, Texturen oder Routen berechnet werden. Außerdem gibt es noch Werkzeuge wie den MapBuilder oder den MapViewer.

Für die Verarbeitung von Rasterdaten sieht Oracle den Oracle Spatial GeoRaster vor. Dieser beugt seit der Version 10 dem Trend entgegen, Rasterdaten in BLOB's abzulegen. Dadurch war es bisher nicht möglich die Inhalte einzelner Raster abzufragen. Dies war zuvor dem Geoinformationssystem überlassen worden. Der GeoRaster hingegen erlaubt die Speicherung, Indexierung, Anfrage und Analyse von georeferenzierten Rasterobjekten. Dazu werden objektrelationale Datentypen und Funktionen bereitgestellt [Bri08]. Alle Funktionen, Attribute und Methoden werden durch die Klasse SDO_GEOASTER bereitgestellt.

Für die Verarbeitung von großen Datenmengen macht sich Oracle der Idee der Parallel Query zu nutzen. Die Verwendung dieser Technologie fand bereits Einzug in der Version 7.3 des Datenbanksystems. Die Grundidee von Parallel Query ist es, mehrere Unterprozesse zu starten, die jeder für sich eine Teilmenge der Daten bearbeiten und das Zwischenresultat dann zur Ermittlung des Endergebnisses an den zentralen Benutzerprozess zurückmelden. Durch dieses Prinzip werden die zur Verfügung stehenden Rechnerressourcen besser ausgenutzt und eine zum Teil erhebliche Verbesserung der SQL Antwortzeiten bei Abfragen großer Tabellen erreicht [OTN12].

5.3.2 Informix

Informix war eines der bedeutendsten Datenbanksysteme unter dem Unix – Betriebssystem in den 1990er Jahren. Die Firma Informix entstand 1996 aus der Übernahme der Firma Illustra Information Technologies, die sich vorrangig auf objektrelationale Datenbanksysteme spezialisierte. Das Besondere an dieser Datenbank war die Erweiterbarkeit durch sogenannte DataBlades, die dann auch in den Informix Dynamic Server (IDS) übernommen wurde. IBM übernahm die Firma Informix im Jahre 2001, nachdem diese es versäumten, einen konkurrenzfähigen Nachfolger für ihre Entwicklungsumgebung Informix 4GL auf den Markt zu bringen. Nachdem Kauf versuchte man Informix mit dem hauseigenen Datenbanksystem DB2 zusammenzuführen. Um die bestehenden Informix Kunden nicht zu verunsichern, entschied man sich beide Produkte parallel weiterzuentwickeln. Man setzte den Fokus auf die erneute Etablierung des Informix Dynamic Servers auf dem Absatzmarkt.

Auch um die Datenbank für die räumlichen Anwendungen zu erweitern, wurden die angesprochenen DataBlades verwendet. Diese Softwarepakete bestehen aus SQL-Anweisungen und Code – Anweisungen in der Stored Procedure Language (SPL) oder einer anderen Programmiersprache. Diese Anweisungen können Zugriffsmethoden enthalten oder Funktionen für die Verwaltung von Indizes beschreiben. Außerdem stehen verschiedene Funktionen zur Optimierung von Anfragen bereit. IBM hat verschiedene DataBlades zur Verfügung gestellt. Zu diesen gehören:

- Excalibur Text Search DataBlades
- Image DataBlades
- Spatial DataBlades
- Time Series DataBlades
- Geodetic DataBlades
- Video DataBlades
- Web DataBlades
- NAG (Numerical Algorithm Group) DataBlades.

Somit wurden als Erweiterungen speziell für Geoanwendungen der Spatial und der Geodetic DataBlade veröffentlicht. Erstere stellt geometrische Datentypen und Methoden zur Verfügung, die den Vorgaben des Open Geospatial Consortium entsprechen [Bri08]. Jedoch stehen auch zusätzliche Werkzeuge außerhalb dieses Standards zu Verfügung. So können GML oder ESRI Shapefiles verarbeitet werden oder auch Generalisierungen und Transformationen zum Einsatz kommen. Das Geodetic DataBlade ist dagegen völlig konträr. Es unterstützt keinen der OGC Standards und stellt verschiedene geografische Koordinatensysteme zur Verfügung. Einzige Gemeinsamkeit ist, dass beide Erweiterungen räumliche Indexe bei Anfrageoperationen einsetzen.

Der Spatial DataBlade Modul implementiert die Standards der OGC und SQL3 Spezifikation für abstrakte Datentypen (ADT). Diese Daten können räumliche Datentypen aufnehmen. Die Spatial – Erweiterung erfüllt auch die Simple – Features

Spezifikation. Es können Module sowohl für räumliche und nicht räumliche Daten integriert werden, auf die mit SQL zugegriffen werden kann.

Wenn das DataBlade – Modul installiert ist und im Datenbanksystem registriert ist, können Tabellen erstellt werden, die räumliche Attribute enthalten können. Das Modul kann mit Java Anwendungen, durch den JDBC - Treiber, mit C – Anwendungen mit Hilfe des ODBC – Treiber kommunizieren und direkt auf ESQL / C – Module, sowie auf die eigentliche Datenbank zugreifen. Der Zugriff erfolgt jeweils mit der Abfragesprache SQL. Die nachfolgende Abbildung zeigt nochmals vereinfacht die Architektur des Systems.

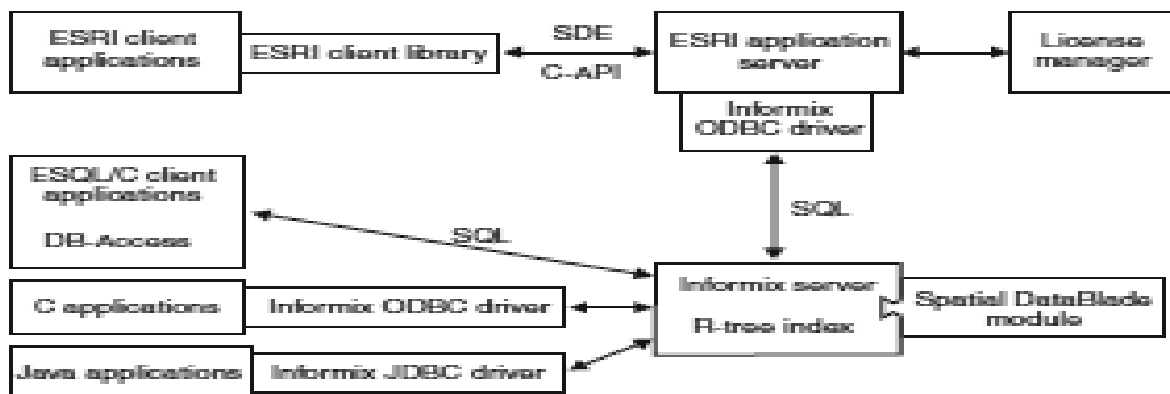


Abbildung 9: Systemarchitektur [SDB12]

Als Indizes bietet die Erweiterung einen R – Tree an. Dieser dient speziell für den schnelleren, effizienteren Zugang auf Geodaten. Um die Geodaten überhaupt nutzen zu können, muss das Erweiterungsmodul in der Tabelle spatial_references registriert werden. In der Tabelle werden alle Kartenprojektionen gespeichert. Damit kann später zwischen den verschiedenen geographischen oder ebenen kartesischen Koordinaten umgerechnet werden. Diese Zuordnung ergibt das räumliche Bezugsreferenzsystem. Die SRID ist der eindeutige Schlüssel für jeden Datensatz in der Tabelle. Alle benutzten Referenzsysteme müssen sich in dieser Tabelle befinden. Alle verwendeten Geometrien in einer Spalte müssen dasselbe Referenzsystem verwenden.

Das Datentypmodell von Informix ist stark angelegt an das Simple – Feature Modell der OGC. Die Grundbasistypen sind auch hier Punkte, Linien und Flächen und besitzen die gleichen Eigenschaften wie das Simple – Feature Modell. Die

nachfolgende Abbildung zeigt nochmals die verfügbaren Datentypen in der Spatial – Erweiterung.

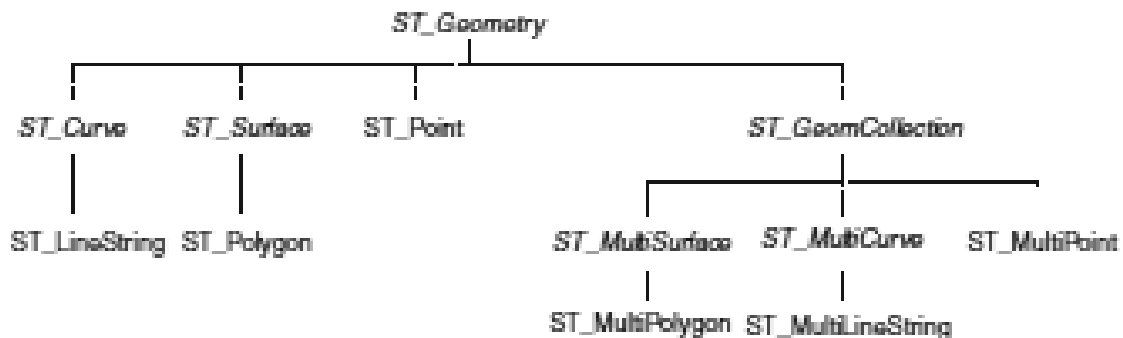


Abbildung 10: Geometriemodell Informix Spatial [SDB12]

Der Datentyp `ST_Geometry` ist die abstrakte Superklasse, die keine instanziierten Datentypen anbietet. Von allen anderen Unterklassen können Objekte erstellt werden. Das bedeutet, es können Tabellenspalten mit den Datentypen, wie in Abbildung 10 zu sehen, übernommen werden.

IBM bietet auch eine spezielle Schnittstelle an, die es den Nutzern ermöglicht eigene DataBlades zu erstellen. Das DataBlade Developers Kit bietet diese Möglichkeiten an. Zu den Werkzeugen gehören unter anderem ein graphisches Entwurfs- und ein Debuggingtool. Dieses DataBlade enthält noch andere verschiedene Pakete, die für die Entwicklung nötig sind. Die meisten DataBlades werden von speziellen Drittanbietern entwickelt.

5.3.3 DB2

Die DB2 Datenbank ist das hauseigene Datenbanksystem der Firma IBM. Dieses setzt im Gegensatz zum aufgekauften Datenbanksystem Informix auf den relationalen Ansatz, der noch auf die Grundlagen von E. F. Codd zurückzuführen ist. Für die Verwaltung von Geodaten steht als Erweiterung der Spatial Extender zur Verfügung. Ebenso wie der Spatial DataBlade stellt auch der Spatial Extender geometrische Datentypen und Methoden bereit, die ebenfalls den OGC – Bestimmungen entsprechen. Räumliche Anfragen werden durch eine hierarchische Gitterstruktur unterstützt. In Analogie zum Geodetic DataBlade und in Ergänzung zum Spatial Extender bietet IBM DB2 das Geodetic Data Management Feature an,

das Voronoi – Diagramme als räumliche Indexe einsetzt [Bri08]. Weiterhin werden alle wichtigen Standards der ISO oder OGC unterstützt. Alle die vom Spatial Extender unterstützten Geometrien, sind im OpenGIS Simple – Feature Specification for SQL definiert. Alle zur Verfügung gestellten Funktionen unterliegen der OGC - Spezifikation Simple Features for SQL sowie dem ISO – Standard SQL/MM Part 3: Spatial.

Die Extender – Pakete sind ebenfalls Softwarekomponenten zur Funktionalitätserweiterung des eigentlichen Datenbanksystems. Wie auch alle anderen Pakete setzt sich dieses aus verschiedenen Funktionen und Datentypen zusammen. Zu den bekanntesten Erweiterungen gehören:

- Image Extender
- Audio Extender
- Video Extender
- Text Extender
- Spatial Extender
- XML Extender
- Net Search Extender
- EcoWin Time Series Extender

Im Bereich der Geodaten ist der Spatial Extender die wichtigste Erweiterung. Dieser wird als Teilpaket des DataJoiner ausgeliefert. Mit dem Spatial Extender können raumbezogene Daten verarbeitet werden und diese Daten genutzt werden um geographische Objekte zu erzeugen und zu analysieren. Zu den Basiskonzepten, die den meisten Operationen zu Grunde liegen, gehören die geografischen Objekte, räumliche Informationen, Daten sowie die Geometrien. Die einfachste Zusammensetzung eines Datenelementes besteht aus einem Koordinatenpaar, das die Position in geographischen Koordinaten ausdrückt. Ein etwas komplexeres Element besteht aus mehreren Koordinatenpaaren, das zum Beispiel ein Linienzug oder Polygon sein kann. Jedes dieser Elemente lässt sich zu einem räumlichen Datentyp zuordnen. Ein geografisches Objekt kann durch ein oder mehrere Datenelemente dargestellt werden, z. B. durch die Datenelemente in einer Zeile einer Tabelle. Objekte werden allgemein durch Symbole abgebildet, die gleichzeitig auch

als Geometrie angesehen werden. Geometrien setzen sich aus einem optischen und einem mathematischen Teil zusammen. Der optische Teil beschreibt im Grunde nur die Form der Geometrie, also ob es sich um einen Punkt, eine Linie oder ein Polygon handelt. Die mathematischen Eigenschaften hingegen beschreiben die Lageparameter der Geometrie. Diese fassen alle Koordinaten eines Objektes zusammen.

Geodetic Data Management Feature ist eine geodätische Erweiterung für den Spatial Extender. Dieses Paket enthält ebenfalls alle räumlichen Datentypen und Funktionen wie das Spatial Produkt. Einziger Unterschied ist das jetzt auch Abfragen möglich sind, die die Polregionen einschließen oder über den 180. Meridian hinausgehen. Durch die Verwendung von Längen- und Breitengraden auf einem ellipsoidischen Erdmodell können exakte Positionen auf der Erdoberfläche bestimmt werden.

Mit räumlichen Indizes lassen sich räumliche Abfragen beschleunigen. Folgende Indextypen stehen im Spatial Extender und im Geodetic Data Management Feature. Im Spatial Extender wird mit der Rasterindexierung gearbeitet. Dieser Index wird vor allem in mehrdimensionalen räumlichen Daten eingesetzt. Jedoch müssen diese Daten auch über räumliche Indexspalten verfügen. Den größten Nutzen kann der Spatial Extender aus Verarbeitung zweidimensionaler Daten ziehen. Besonders bei der Verarbeitung von Flachprojektionen eignet sich diese Methode sehr gut. Das Geodetic Data Management Feature hingegen setzt als Indextyp auf Geodätische Voronoi Indizes. Dieser eignet sich besser, da er die Gestalt der Erde als Idealbild ansieht. Es werden somit keine Verzerrungen oder Verformungen berücksichtigt. Für alle Geometrien wird der minimal einschließende Kreis berechnet. Mit dieser Information werden die Daten in einer Zellenstruktur angeordnet. Der Voronoi – Index kann dadurch schneller die Objekte auf einen bestimmten Bereich überprüfen. Danach können dann an den gefilterten Objekten die Operationen durchgeführt werden. Die Verwendung dieses Index führt damit zur Leistungsverbesserung, da alle nicht beachteten Objekte vernachlässigt werden können.

5.4 Weitere Systeme

In diesem Kapitel sollen weitere Datenbanksysteme vorgestellt werden die auch im Bereich der Geoinformatik eingesetzt werden, aber den Schwerpunkt nicht auf die Speicherung oder Verarbeitung von Geodaten legen. Zu diesen gehören der Apache Xindice, der Rasdaman Rasterserver und der Tamino XML Server, die im Folgenden kurz näher vorgestellt werden.

Apache Xindice:

Der Apache Xindice ist ein Open-Source Datenbanksystem, dass zur Speicherung und Verarbeitung von XML – Daten verwendet werden kann. Xindice ist die Fortführung des dbXML – Core – Projekts. Diese Projektgruppe verkaufte im Dezember 2001 ihren Quellcode an Apache. Die komplette Datenbank ist in JAVA implementiert worden und benutzt dabei als Anfragesprache XPath. Als Sprache zur Aktualisierung der Datenbank und der Datensätze wird XML:DB XUpdate bereitgestellt. XML:DB ist vergleichbar mit der JDBC. Mit dem mitgelieferten XML – RPC – Plugin ist es möglich auf Xindice mit anderen Programmiersprachen zu zugreifen. Xindice ist nicht auf das Speichern von riesigen Datenmengen und großen Dateien ausgelegt. Vielmehr liegt der Schwerpunkt auf der Speicherung von XML – Datenpaketen. Die Form dieser Speicherung bietet einige Vorteile. So müssen die XML – Dateien nicht mehr in andere Strukturen umgewandelt werden, bevor sie in der Datenbank verarbeitet werden können. Dadurch werden die Daten in XML geliefert und auch als Ergebnis zurückgeliefert. Den Nutzern soll damit mehr Flexibilität ermöglicht werden. Über Java- oder Kommandozeilenprogramme können Daten in die Datenbank eingepflegt werden.

Rasdaman Rasterserver:

Der Rasdaman Rasterserver ist ein Produkt der deutschen Firma Rasdaman mit Sitz in Berlin. Herzstück der Firma ist die Rasdaman – Technologie. Mit Hilfe dieser können auf einem Server riesige mehrdimensionale Rasterbilder abgespeichert werden. Dabei spielt die Größe der Daten jedoch keine Rolle. Die Daten werden in einer normalen Standarddatenbank vorgehalten. Der Rasterserver ermöglicht die

Navigation, Extraktion und Analyse von Sensordaten, Bildern und statistischen Daten in unbegrenzter Größe im Bereich von 1D bis 4D. Es werden alle gängigen Rasterdatentypen unterstützt. Auch Orthofotos, thematische Karten, Radaraufnahmen, Hyperspektralbilder oder selbstdefinierte Typen können verarbeitet werden.

Rein technisch gesehen, kann der Rasdaman Rasterserver als Middleware bezeichnet werden, die die standardmäßig eingesetzte relationale Datenbank um Abfragemöglichkeiten für große mehrdimensionale Arrays erweitert. Zu diesem Zweck wird von der Firma eine Definitionssprache namens „rasdl“ angeboten um Array – Typen zu definieren. Mit „rasql“ steht eine SQL – basierte Abfragesprache bereit. Somit kann auch weiterhin mit „select“ oder „insert“ auch auf eine Rasterdatendatenbank zugegriffen werden. Auf der Serverseite werden die Anfragen optimiert und dann auf die entsprechenden Objekte in der Datenbank angewendet. Die Rasterdaten sind in der Datenbank als BLOBs gelistet und zudem noch partitioniert. Auf der Server Seite sind die Arrays in einer Standarddatenbank gespeichert. Zu diesem Zweck werden Arrays in Teilfelder aufgeteilt. Jedes dieser Teilfelder wird in einem BLOB gespeichert. Dies ermöglicht es herkömmlichen relationalen Datenbanksystemen, Arrays von unbegrenzter Größe zu verwalten. Eine API sowohl für C++ als auch für Java stehen bereit. Die Client – Entwicklung wird durch die C – API, „raslib“, der Java – API, „rasj“ und dem ODMG Standard unterstützt. Die Kommunikation mit der Datenbank ist dann relativ einfach. Eine Verbindung zur Datenbank wird geöffnet, danach der Abfrage – String übergeben und schließlich wird die Ergebnismenge zurückgeliefert. Iteratoren ermöglichen bequemen Zugang zu den Abfrageergebnissen. Durch ein Servlet-Paket wird ein Layer für OGC-Schnittstellen eingefügt. Dadurch wird der Rasterserver sogar zu einem Geo-Rasterserver. Vorrangig werden nur die OGC Webservices optimiert für den Rasterserver eingesetzt.

Eine Serverinstallation kann aus einer beliebigen Anzahl von Serverprozessen bestehen. Ein dynamischer Scheduler, der „rasmgr“, empfängt eingehende Verbindungen und weist diesen einen kostenlosen Serverprozess zu. Dieser Serverprozess wird dann einem bestimmten Client gewidmet, bis die Verbindung

beendet wird. Dies ermöglicht einen zeitgleichen Zugriff. Gleichzeitig wird dadurch die Sicherheit der Clients, sich gegeneinander zu isolieren erhöht.

Das Server Frontend erhält WCPS Abfragen im String oder XML Format. Diese werden analysiert und auf ihre syntaktische und semantische Richtigkeit geprüft, die in den Metadaten zu finden ist. Danach kann eine entsprechende Abfrage in der Rasdaman Abfragesprache, rasql erzeugt werden. Der Server ist das eigentliche Arbeitsgerät, der die Anfrage erstellt. Die Clients dienen zur Darstellung oder Visualisierung von 1D – Zeitreihen, 2D – Bildern und 3D – Voxelwürfeln. Die Abbildung zeigt nochmal die Systemarchitektur des Rasterservers.

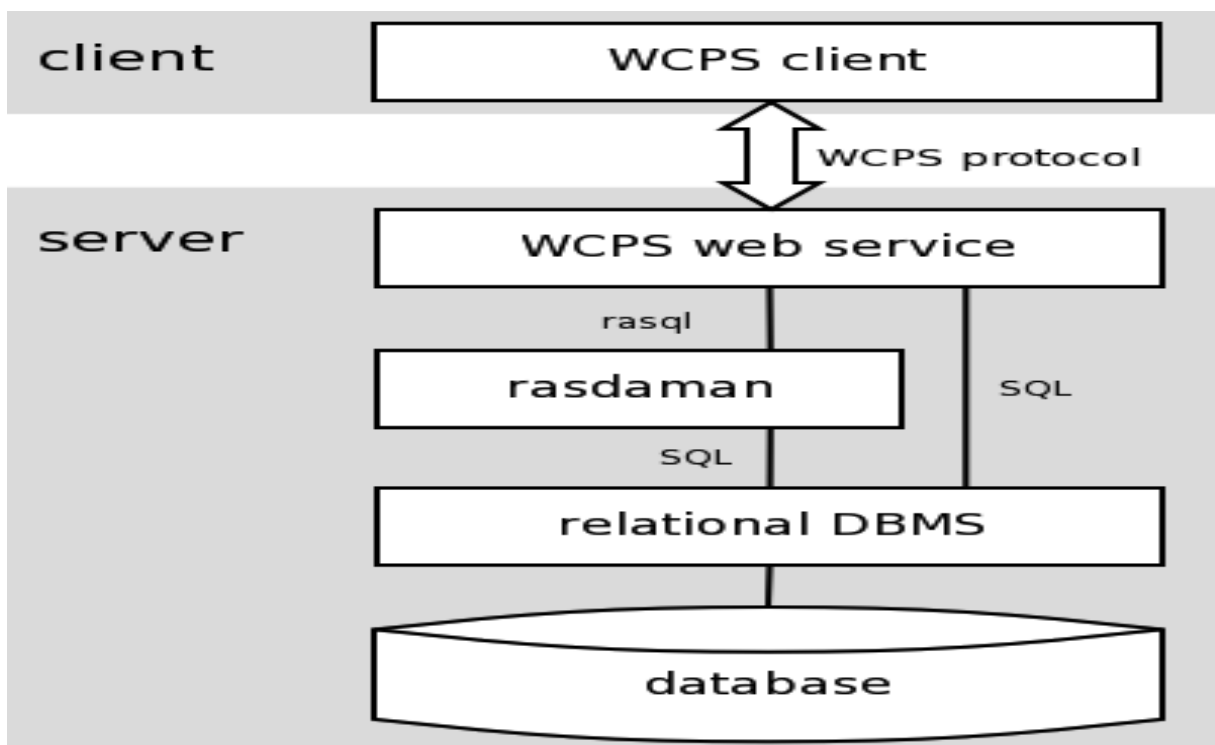


Abbildung 11: Systemarchitektur Rasdaman Rasterserver [Ras12]

Den Rasterserver gibt es sowohl in einer freien Version als auch in einer käuflich erwerblichen Fassung.

Tamino XML Server:

Der Tamino XML Server ist ebenfalls ein XML – Datenbankserver geht aber einen anderen Weg als der Apache Xindice. Vertrieben wird das System von der Software AG. Zur Verfügung steht allerdings nur eine käuflich erwerbliche Lizenz. Grundlage

des Systems sind die vorhandenen offenen Internet Standards und Technologien. Der Server bietet die Möglichkeit Daten von verschiedenen Quellen aufzunehmen und in der Datenbank abzulegen. Danach können die Daten in verschiedenen Formaten umgewandelt und später ausgegeben werden. Der größte Vorteil dieser Anwendung liegt in der nativen Abspeicherung aller vorhandenen XML – Daten. Dies bedeutet, dass der Server auf alle gängigen Web und XML Standards zugreifen kann. Die nachfolgende Abbildung zeigt die Architektur des Tamino XML Server. Man erkennt dass der Server aus mehreren Hauptkomponenten zusammengesetzt ist. Zu diesen gehören der eigentliche Serverteil und die einzelnen Produktbestandteile. Letztere können auch als Standalone – Komponenten eingesetzt werden.

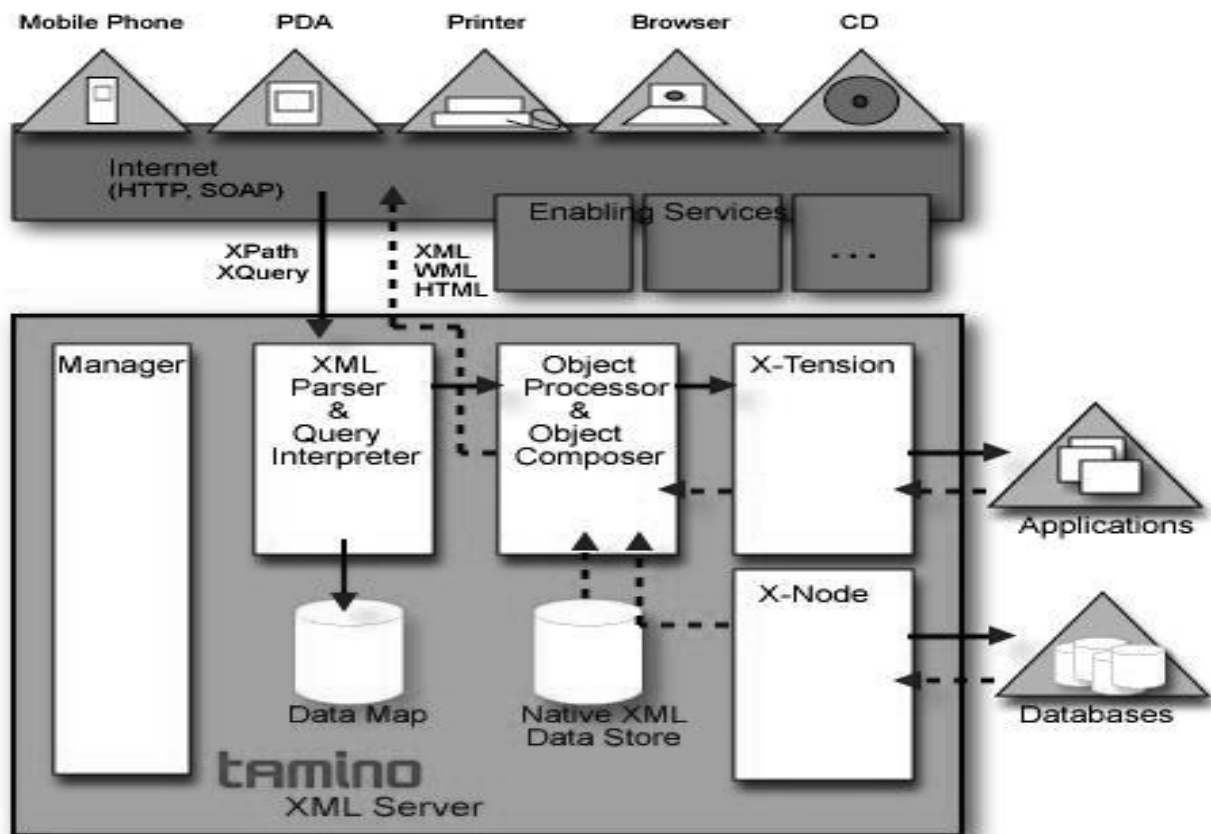


Abbildung 12: Systemarchitektur Tamino XML Server [FHK12]

Wie sich aus der Abbildung erkennen lässt, setzt sich der XML Server aus 5 wesentlichen Komponenten zusammen. Diese sind

- Native XML Data Store, welcher die XML-Engine enthält
- Data Map

- X - Node
- X - Tension
- Tamino Manager

Der Nativer XML Data Store mit der XML – Engine ist die zentrale Einheit des Systems, da sie als Grundlage für die meisten der vielen Funktionalitäten dient. Dazu zählen unter anderem XQuery oder eine Volltextsuche. Diese Komponenten stützen sich auf die im XML Data Store integrierten Funktionen (XML Parser, Query – Interpreter, XML Datenspeicher). In der Data Map werden alle Metadaten gesammelt, die vorher in den Schemata deklariert werden. Weiterhin sind dort Informationen gespeichert, die einige verwendete Funktionen benötigen. Einer dieser Funktionen ist die Validierung oder das Mapping der Daten. Mit X – Node wird eine Komponente bereitgestellt, die es ermöglicht externe Speichersysteme in die Architektur miteinzubinden. Dadurch wird es möglich verschiedene Datenbanksysteme an das System zu binden. Die relationale Struktur der Datenbanken wird dabei in eine XML konforme Struktur umgewandelt. Danach können die Daten leichter weiterverarbeitet werden. X-Tension ermöglicht es vom Benutzer erstellte Funktionen aufzurufen. Dabei spielt es keine Rolle in welcher Programmiersprache die Funktionen geschrieben sind. Der Aufruf erfolgt über diese Komponente. Der Tamino Manager ist die zentrale Verwaltungskomponente, die nach dem Client – Server – Prinzip entwickelt wurde. Dem Nutzer wird auch eine graphische Benutzeroberfläche bereitgestellt, die im Webbrowser aufgerufen wird. Mit diesem Tool können die Datenbanken aufgebaut werden oder Backups erstellt werden. Weiterhin besteht die Möglichkeit auf die Data Map zuzugreifen.

6 Bearbeitung von Standardoperationen

In diesem Kapitel soll anhand von 3 Standardoperationen die Anfrageverarbeitung in den einzelnen Datenbanksystemen untersucht werden. Als Referenzoperationen werden folgende Funktionen näher beleuchtet:

- Puffer – Funktion
- Verschneidungsfunktionen
- Prüfung topologischer Prädikate

Beim Buffering wird um eine bestehende Geometrie eine neue Geometrie generiert. Es können sowohl Puffer für Punkte, Linien oder Flächen erzeugt werden. Ausgehend von der bestehenden Geometrie wird in einem festen Abstand eine neue Fläche gebildet. Bei der Erzeugung dieser Flächen wird die gesamte Geometrie umschlossen. Der Abstand selbst kann vom Anwender später selber festgelegt werden. Es besteht auch die Möglichkeit mehrere Puffer um ein Objekt zu legen. Die ursprünglichen Geometrien bleiben von dieser Funktion komplett unbeeinträchtigt. Vielmehr entsteht mit der Pufferfläche eine neue Geometrie. Somit sind Puffer neue Objekte, die sich zur Weiterverarbeitung eignen. Besonders zur Verarbeitung mit verschiedenen Verschneidungsoperationen lassen sich die Pufferobjekte verwenden. Unter Verschneidung versteht man die Überlagerung von verschiedenen Layern oder Objektklassen. Bei der Verschneidung werden 4 Operationen unterschieden. Zu diesen gehören der Schnitt (Intersection), die Vereinigung (Union), die Differenz (Difference) und die symmetrische Differenz (Symmetric Difference). Die gängigste Operation ist im GIS – Bereich die Vereinigung und wird im weiteren Verlauf dieses Kapitels an den verschiedenen Datenbanksystemen getestet. Bei allen Operationen wird mit Hilfe von booleschen Operationen eine neue Ausgangsmenge erstellt. Auch hier entstehen als Ergebnis jeder Operation wieder neue Objekte, die Ausgangsebenen bleiben davon jedoch unverändert. Als letztes sollen in diesem Kapitel die topologischen Verknüpfungen von Geodaten im Einsatz getestet werden. Topologische Prädikate sind boolesche Aussagen über topologische Beziehungen zwischen zwei Geometrien. Um die Semantik solcher Prädikate exakt festlegen zu können, bedient man sich des 9 – Intersection – Modells (9IM). Dazu wird jede Geometrie in den Rand, das Innere und das Äußere der Geometrie aufgeteilt. Diese drei Bereiche sind paarweise disjunkt [Kud07]. Anhand der folgenden Abbildung soll

aufgezeigt werden, welche Beziehungen zwischen den Geometrien geprüft werden können.

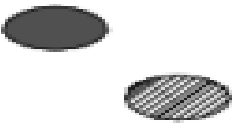

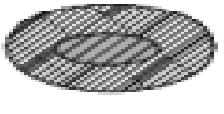




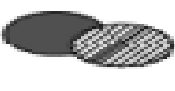
			
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ <p>disjoint</p>	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ <p>contains</p>	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ <p>inside</p>	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ <p>equal</p>
			
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ <p>meet</p>	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ <p>covers</p>	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ <p>coveredBy</p>	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ <p>overlap</p>

Abbildung 13: 9 – Intersection – Modell [GIS12]

Als nächstes soll an den vorgestellten Geodatenbanksystemen die Funktionsfähigkeit dieser 3 Standardoperationen veranschaulicht werden. Dazu mussten zuerst alle Datenbanksysteme lauffähig gemacht werden.

PostgreSQL / PostGIS

PostGIS unterstützt sowohl den OpenGIS als auch den SQL / MM Standard. Aufgrund der höheren Anzahl an bereitgestellten Funktionen bietet letzterer natürlich einige Vorteile. Zunächst sollen diese 3 oben genannten Funktionen in PostGIS untersucht werden.

a) Buffer

Die Buffer – Funktion fand mit der Version 1.5 Einzug in das Datenbanksystem. Der allgemeine Aufruf der Funktionen aus beiden Standards sieht folgendermaßen aus.

geometry ST_Buffer(geometry g1, float radius_of_buffer);

Buffer (geometry, distance [, amoseg]) : POLYGON

In einem praktischen Beispiel sieht die Umsetzung wie folgt aus. Getestet wurde die PostGIS – Version 1. auf der OSGeo – Live CD in der Version 6.0. Die aktuelle Version 2.0.1 ist jedoch die aktuellste. Somit war auch die Visualisierung der Ergebnisse mit einem geeigneten GIS – System möglich. Im Test wurde OpenJump als Visualisierungswerkzeug verwendet. Die Datensätze stammen aus einer freien Testdatenbank für das Datenbanksystem SpatiLite. Alle getätigten Abfragen sind nur als beispielhaft anzusehen.

```
SELECT ST_Buffer(Towns.Geometry, 10)  
from Towns  
where Towns.Name = 'Firenze';
```

```
SELECT Buffer(Towns.Geometry, 10)  
from Towns  
where Towns.Name = 'Firenze';
```

b) Verschneidung

Als typische Verschneidungsoperation gilt die Vereinigungsfunktion. In PostGIS gibt es 3 verschiedene Varianten dieser. Zum einen gibt es die klassische Variante, die zu den Testzwecken herangezogen wird. Andererseits gibt es eine Union – Funktion, die weniger speicherintensiv ist und wiederum eine Funktion, die für das Handling von Komponenten der Klasse Geometriekollektion zuständig ist. Standardmäßig sieht der Aufruf so aus:

```
geometry ST_Union(geometry g1, geometry g2);
```

```
Union ( geom1, geom2 ) : GEOMETRY oder EMPTY
```

In einem praktischen Beispiel sieht die beispielhafte Umsetzung wie folgt aus.

```
SELECT ST_Union(Regions.Geometry, Towns.Geometry)  
  
from Towns, Regions;  
  
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

```
SELECT Union(Regions.Geometry, Towns.Geometry)
from Towns, Regions;
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

c) Topologische Prädikate

Wie aus dem einleitenden Abschnitt hervorgeht, gibt es eine Vielzahl von Operationen zur Bestimmung der topologischen Beziehung. Die Funktion `Overlaps()` soll hierbei schemenhaft herangezogen werden und ausgeführt werden. Als Ergebnismenge kommen nur boolesche Werte in Frage.

```
boolean ST_Overlaps(geometry A, geometry B);
Overlaps ( geom1, geom2 ) : Boolean
```

In einem praktischen Beispiel sieht das dann folgendermaßen aus.

```
SELECT ST_Overlaps(Regions.Geometry, Towns.Geometry)
from Towns, Regions;
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

```
SELECT Overlaps(Regions.Geometry, Towns.Geometry)
from Towns, Regions;
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

Zusammenfassend kann man sagen, dass PostGIS alle zu testenden Funktionen bereitgestellt hat und diese auch programmtechnisch umsetzbar sind. Vor allem durch die Unterstützung der beiden Standards gibt es verschiedene Varianten beim Aufruf der Funktionen. Somit lässt sich PostGIS als Geodatenbank auch für komplexere Anwendungen einsetzen.

MySQL

Wie schon aus Kapitel 5 hervorging, ist der Funktionsumfang von MySQL im Bereich GIS doch sehr eingeschränkt. Die aktuelle GIS – Implementierung von MySQL

unterstützt allerdings nur Berechnungen, die sich auf die Bounding – Box einer Geometrie stützen. Dadurch können die aus GIS – Systemen bekannten Funktionen wie Buffer, Distance, Intersection, ConvexHull oder SymDifference nicht verwendet werden. Es existieren bereits Patches, die die Situation verbessern, es ist aber unklar, wann sie in eine offizielle Version von MySQL aufgenommen werden [KoÖ09]. Im Handbuch zur Version 5.1 lassen sich Informationen finden, die schon eine konkrete Implementierung ansprechen. Jedoch fehlen in den beiden letzten veröffentlichten Handbüchern jegliche Informationen zu MySQL und Geodaten. Dennoch gibt es auch Funktion, die in MySQL umgesetzt worden sind. Zu diesem gehören unter anderem die Operationen zur Bestimmung der topologischen Beziehungen. Diese Operationen arbeiten zum einen mit der Bounding – Box und zum anderen ohne diese. Auch hier wird zu Testzwecken die Funktion Overlaps() herangezogen. Der formale Aufbau der Funktion sieht folgendermaßen aus. Getestet wurde die MySQL Version 5.5.25 aus dem XAMPP in der Version 1.8.0.

OVERLAPS(G1, G2) oder MBROVERLAPS(G1, G2)

In einem Beispiel könnte dies dann folgendermaßen aussehen.

```
SELECT Overlaps(Regions.Geometry, Towns.Geometry)
from Towns, Regions;
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

An dieser Stelle lässt sich resümierend festhalten, dass MySQL nur wenig Funktion für die Verarbeitung von Geodaten bereitstellt. Einige Analysefunktionen fehlen bisher komplett. Gerade diese stellen einen wichtigen Teil der Anforderungen an Geodatenbanken dar. Auch beziehen sich die meisten bereitgestellten Funktionen nur auf geometrische ebene Objekte und lassen somit komplexe Projekte nicht zu.

SpatLite

Zum Ausführen von SQL – Anweisungen eignet sich besonders gut die bereitgestellte grafische Oberfläche. In dieser können die Anweisungen ganz einfach ausgeführt werden. Auch wird eine Testdatenbank mit Geodaten bereits bereitgestellt. Als erste zu testende Funktion wird der Puffer herangezogen.

a) *Buffer*

SpatialLite nutzt die GEOS – Bibliothek um die Performance der Funktionen zu verbessern. Diese Funktion wird vom OpenGIS – Standard definiert. Die Buffer – Funktion liefert als Rückgabewert ein Objekt vom Typ GEOMETRY. Es kann auch vorkommen, dass die Ergebnismenge leer ist. Der Buffer kann auf jede Art von Geometrie angewendet werden. Der Aufruf der Funktion sieht folgendermaßen aus

$$geom2 = Buffer (geom1 , radius)$$

Der Radius wird in der derselben Einheit angegeben, wie er in der Ausgangsgeometrie zu finden ist. Zu Testzwecken wurde die Graphische Oberfläche in der Version 1.5.0 auf der OSGeo – Live CD verwendet.

```
SELECT Buffer(Towns.Geometry, 10)
from Towns
where Towns.Name = 'Firenze';
```

b) *Verschneidung*

Auch für die Verschneidungsoperationen steht wieder die GEOS – Bibliothek aus Performancegründen zur Verfügung. Hier konzentrieren wir uns auf die Vereinigungsfunktion. Im OpenGIS – Standard ist eine Funktion Union() vordefiniert. Jedoch ist Union in der SQLite Syntax ein reserviertes Schlüsselwort, so dass die in SpatialLite die Funktion GUnion() genannt wird. Auch das Ergebnis dieser Funktion liefert wieder ein Geometrieobjekt. Der allgemeine Aufruf der Funktion sieht so aus:

$$g3 = GUnion(g1, g2)$$

An einem Beispiel sieht das Ganze dann folgendermaßen aus.

```
SELECT GUnion(Regions.Geometry, Towns.Geometry)
from Towns, Regions;
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

c) Topologische Prädikate

Zur Bewertung der Topologischen Beziehungen wird ebenfalls wieder die GEOS – Bibliothek verwendet. Als Ergebniswert aller topologischen Funktionen kommen die Werte 1 (= True) oder 0 (= False) in Frage. In Ausnahmefällen kann auch -1 als Rückgabewert erfolgen, jedoch nur wenn ein Fehler auftreten sollte. Als Beispielfunktion dient in diesem Falle Overlaps(). Allgemein sieht der Aufbau der Funktion so aus.

Overlaps (geom1 , geom2)

In einem Beispiel könnte die Funktion so angewendet werden.

```
SELECT Overlaps(Regions.Geometry, Towns.Geometry)
from Towns, Regions;
where Towns.Name = 'Firenze' and Region.Name = 'Firenze'
```

Zusammenfassend kann man sagen, dass SpatiaLite durch seine einfache Installation und Handhabung zu überzeugen weiß. Auch wurden alle zu testenden Funktionen einsatzbereit zur Verfügung gestellt.

Oracle

Oracle stellt für alle 3 zu testenden Standardoperationen entsprechende Funktionen zur Verfügung. Dabei muss ebenfalls beachtet werden, dass sich für jede bereitgestellte Norm oder Standard der Name der Funktion ändert. Die Ergebnisse bleiben jedoch die Gleichen. Alle entnommenen Beispiele und Funktionsaufrufe finden sich in [DGO12] wieder. Erläutert werden darin alle Funktionen und Beispiele aus der Version 11.

a) Buffer

Die erste zu untersuchende Funktion ist der Buffer. Der Funktionsname aus der generellen Geometrieklasse SDO_GEOMETRY lautet SD_GEOM.SDO_BUFFER, die SQL / MM Spatial Funktion trägt den Namen ST_GEOMETRY.ST_BUFFER und BUFFER heißt der Funktionsname aus dem Simple – Feature Modell. Alle 3 Funktionen nutzen den objektrelationalen Ansatz und unterscheiden sich in der

Anzahl der Übergabeparameter. Der generelle Aufbau der einzelnen Funktionen sieht so aus:

```
SDO_GEOM.SDO_BUFFER(  
    geom IN MDSYS.SDO_GEOMETRY,  
    dist IN NUMBER  
    tolerance IN NUMBER  
    [, params IN VARCHAR2]  
) RETURN MDSYS.SDO_GEOMETRY;
```

```
ST_GEOMETRY.ST_BUFFER(  
    distance IN NUMBER  
) RETURN ST_GEOMETRY;
```

```
BUFFER(  
    geom IN GEOMETRY  
    distance IN NUMBER  
) RETURN GEOMETRY;
```

Alle Übergabeparameter sind selbsterklärend und verpflichtend anzugeben. In einer konkreten Abfrage an einem Beispiel soll der Einsatz kurz aufgezeigt werden.

```
SELECT g2.name, SDO_GEOM.SDO_BUFFER(g2.shape, 20, 1)  
FROM cola_markets c, user_sdo_geom_metadata m  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =  
'SHAPE' AND c.name = 'cola_a';
```

```
SELECT g2.name, ST_GEOMETRY.ST_BUFFER (20)  
FROM cola_markets c, user_sdo_geom_metadata m  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =
```

```
'SHAPE' AND c.name = 'cola_a';
```

```
SELECT g2.name, BUFFER(g2.shape, 20)
FROM cola_markets c, user_sdo_geom_metadata m
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =
'SHAPE' AND c.name = 'cola_a';
```

Hieran lässt sich erkennen, dass sich die SQL – Anweisung nur im Aufruf der Funktion und sich in der Anzahl der Übergabeparameter unterscheiden. Die letztere Variante dient jedoch zur besseren Übersichtlichkeit und zur besseren Verständlichkeit der Abfrage.

b) Verschneidung

Auch für die Verschneidungsoperation Union stehen in Oracle wieder 3 verschiedene Möglichkeiten zum Aufruf bereit. Der generelle Aufbau der Funktionen ähnelt auch denen des Buffers. Die Funktionen sind folgendermaßen aufgebaut.

```
SDO_GEOM.SDO_UNION(
    geom1 IN MDSYS.SDO_GEOMETRY,
    geom2 IN MDSYS.SDO_GEOMETRY,
    tol IN NUMBER
) RETURN MDSYS.SDO_GEOMETRY;
```

```
ST_GEOMETRY.ST_UNION(
    geom2 IN ST_GEOMETRY,
) RETURN ST_GEOMETRY;
```

```
UNION(
    geom1 IN GEOMETRY,
    geom2 IN GEOMETRY,
```


) *RETURN GEOMETRY*;

Hier muss beachtet werden, dass in der zweiten Aufrufvariante, die Funktion auf ein Geometrieobjekt angewendet werden muss, da als Übergabeparameter nur die zweite zu verarbeitende Geometrie angegeben wird.

```
SELECT SDO_GEOM.SDO_UNION(g1.shape, g2.shape, 10)  
FROM cola_markets g1, cola_markets g2,  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =  
'SHAPE' AND g1.name = 'cola_a' AND g2.name = 'cola_c';
```

```
SELECT geom1.shape.ST_UNION(g2.shape, 10)  
FROM cola_markets g1, cola_markets g2,  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =  
'SHAPE' AND g1.name = 'cola_a' AND g2.name = 'cola_c';
```

```
SELECT UNION(g1.shape, g2.shape)  
FROM cola_markets g1, cola_markets g2,  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =  
'SHAPE' AND g1.name = 'cola_a' AND g2.name = 'cola_c';
```

Auch hier wird wieder deutlich, dass die letzte Abfrage die Übersichtlichste darstellt. Die Ergebnismenge bleibt bei allen gegebenen Funktionen die Gleiche.

c) *Topologische Prädikate*

Hier soll stellvertretend die Funktion *Overlaps* als Referenz herangezogen werden. Von Haus aus liefert Oracle keine eigenen Funktionen zur Bestimmung topologischer Prädikate. Es gibt nur die Funktion *SDO_GEOM.RELATE()*, die prüft ob eine und welche Art der topologischen Beziehung vorliegt. Durch Übergabe spezieller Übergabeparameter kann die Funktion auch auf bestimmte topologische Beziehungen testen. Dafür stehen spezielle Schlüsselwörter zur Verfügung. Zur Prüfung der eigentlichen Beziehungen dienen die Objekte aus der Klasse *ST_GEOMETRY*. Außerdem stehen auch die Funktionen aus dem Simple – Feature

Modell zur Verfügung. Der Rückgabewert dieser Funktionen ist immer vom Typ NUMBER. Somit kommen als Ergebniswert 0 (= False) und 1 (=True) in Frage.

```
ST_GEOMETRY.ST_OOVLAPS(  
    geom2 IN ST_GEOMETRY,  
)  
RETURN NUMBER;
```

```
OVERLAPS(  
    geom1 IN GEOMETRY,  
    geom2 IN GEOMETRY,  
)  
RETURN NUMBER;
```

Hier muss ebenfalls im ersten Beispiel, die Funktion auf ein Geometrieobjekt angewendet werden.

```
SELECT g1.shape.ST_OVERLAPS(g2.shape)  
FROM cola_markets g1, cola_markets g2,  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =  
'SHAPE' AND g1.name = 'cola_a' AND g2.name = 'cola_c';
```

```
SELECT OVERLAPS(g1.shape, g2.shape)  
FROM cola_markets g1, cola_markets g2,  
WHERE m.table_name = 'COLA_MARKETS' AND m.column_name =  
'SHAPE' AND g1.name = 'cola_a' AND g2.name = 'cola_c';
```

Zusammenfassend kann man sagen, dass Oracle für alle 3 getesteten Operationen Funktionen bereitstellt. Man hat dabei die Wahl zwischen verschiedenen Aufrufvarianten für die Funktionen. Somit bleibt festzuhalten, dass die Spatial Erweiterung von Oracle wohl die umfangreichste Ausstattung bietet.

IBM Informix und DB2

Diese beiden proprietären Systeme konnten sich dem Test nicht unterziehen. Aufgrund der hohen finanziellen Kosten und der nicht Verfügbarkeit eines freien Tools, kann an dieser Stelle nur auf die Handbücher und die entsprechenden Developers Guides hingewiesen werden. Da beide Datenbanksysteme alle Normen und Standards erfüllen, werden somit alle in diesem Test unterzogenen Operationen von beiden Produkten unterstützt.

7 Zusammenfassung der Ergebnisse

7.1 Vergleich der Open – Source Geodatenbanksysteme

In der nachfolgenden Tabelle sind nochmal kurz alle wichtigen Ergebnisse der Recherche zusammengefasst. In der Tabelle werden alle gängigen Funktionen der verschiedenen Normen und Standards aufgelistet und gezeigt ob diese in den einzelnen Datenbanksystemen vorhanden ist.

Funktion	PostGIS	MySQL	Spatilite
Datenmodell	objektrelational	relational	relational
Zugriff auf Basiseigenschaften	ja	ja	ja
Test von Geometrieeigenschaften	ja	eingeschränkt	ja
Messfunktionen	ja	ja	ja
Geometrische Funktionen auf einer Geometrie	ja	eingeschränkt	ja
Verschneidungsoperationen	ja	nein	ja

Topologische Prädikate (9-Intersection Matrix)	ja	eingeschränkt	ja
Konvertierung in Well-Known-Text (WKT) Well-Known Binary (WKB)	ja	ja	ja
Konvertierung in Geography Markup Language (GML)	ja	nein	nein
Konvertierung in ScalableVector Graphics (SVG) und Keyhole Markup Language (KML)	ja	nein	ja
Validierung von Geometrien	ja	nein	ja
Koordinatentransformationen	ja	nein	ja
Approximation von Kurvensegmenten durch Streckenzüge	nein	nein	nein
Gesonderte geometrische Funktionen und Verschneidungen auf achsenparallelen Rechtecken	ja	nein	nein

Generalisierung und Polygonisierung von Linien	ja	nein	nein
Geometrische Aggregatfunktionen	ja	nein	nein
Verarbeitung von Rasterdaten	ja, aber nur durch zusätzliche Erweiterungen	ja, aber nur durch zusätzliche Erweiterungen	ja, aber nur durch zusätzliche Erweiterungen
Verarbeitung von großen Datenmengen	Sehr langsam	nur einfach strukturierte Daten	keine Angaben

Tabelle 3: Vergleich Open – Source Geodatenbanken [Bri07]

Als Resultat dieser Zusammenstellung erkennt man, dass zweifelsohne PostGIS den größten Funktionsumfang für die Verarbeitung von Geodaten bereitstellt. Danach folgt mit Abstrichen Spatialite. Im Vergleich zu den beiden anderen vorgestellten Datenbanksystemen stellt MySQL den kleinsten Funktionsumfang bereit. Somit ist es für komplexe Berechnungen und Bearbeitungen völlig ungeeignet. Jedoch bietet Spatialite dank seiner einfachen Installation viele Vorteile für Einsteiger. Wer also auf wenige Einschränkungen im Bereich der geometrischen Operationen und der Rechtevergabe verzichten kann, findet mit Spatialite das perfekte Geodatenbanksystem. Einziger Schwachpunkt ist das fehlende Benutzerhandbuch oder andere Hilfsmittel für den Anwender.

7.2 Vergleich der kommerziellen Geodatenbanksysteme

Auch der Vergleich zwischen den kommerziellen Systemen lässt sich schwer realisieren. Teils sind viele Informationen über die interne Umsetzung einiger Funktionen nicht publik für Außenstehende. Dadurch ist es schwer zu entscheiden, wie das System die Daten verarbeitet und welche Komponenten angesprochen werden. Auch lassen sich keine Testumgebungen auf dem eigenen Rechner ausprobieren, da es sich um proprietäre Systeme handelt. Auch setzen die meisten Hersteller auf die Client - Server Technologie, was einen Test ebenfalls unmöglich macht. Einige Hersteller bieten jedoch Produkte zur Visualisierung an. Abfragen auf Basis einer räumlichen Datenbank sind damit aber nicht möglich. Somit lässt sich ein Vergleich nur anhand der Handbücher und Developers Guides vollziehen. In der nachfolgenden Tabelle sollen nochmals Schemenhaft die 3 vorgestellten Systeme verglichen werden.

Leistungsmerkmale	Oracle Spatial	Informix Spatial DataBlade	DB2 Spatial Extender
Datenbankmodell	Relationales und objektrelationales Modell	objektrelationales Modell	relationales Modell
Verwendung des Simple Feature – Modells	volle Unterstützung	volle Unterstützung	volle Unterstützung
Verwendung des SQL / MM Spatial	volle Unterstützung	volle Unterstützung	volle Unterstützung

Räumlicher Index	R – Tree und Quad – Tree	R – Tree	Rasterindexierung und Geodätischer Voronoi – Index
Anwendungsbereich	überall, aufgrund des Preises nicht für private Anwender geeignet	überall, aufgrund des Preises nicht für private Anwender geeignet	überall, aufgrund des Preises nicht für private Anwender geeignet
Limitierung bei der Geometriegröße	unbegrenzt	2TB	4GB
Unterstützung des XML – Formates	nein	nein	ja
Verarbeitung von Rasterdaten	Oracle GeoRaster	Nur durch Import verschiedener Rasterdatenbanken	keine Informationen
Lizenz	proprietär	proprietär	proprietär

Tabelle 4: Vergleich kommerzielle Geodatenbanksysteme

Aus dieser Übersicht lässt sich nur schwer ableiten, welches dieser Datenbanksysteme sich am besten für die Verarbeitung von Geodaten eignet. Alle Systeme unterstützen die Standards der ISO und der OGC vollends. Allerdings stellt

Oracle noch einige andere Pakete und Erweiterungen für Spatial bereit. Somit stellt es den größten Funktionsumfang im Vergleich zu den anderen Systemen dar. Diese können zwar auch durch zusätzliche Pakete erweitert werden, jedoch werden diese meist von Drittanbietern bereitgestellt und nicht von IBM selbst. Auch hängt das System DB2 hinter den anderen beiden zurück, da es nur auf einem relationalen Datenbankmodell beruht. Wie aus den vorherigen Kapiteln hervorgeht bietet dieses gegenüber dem objektrelationalen Modell viele Nachteile. Demnach eignen sich Oracle Spatial und Informix Spatial DataBlade am besten im Einsatz. Deshalb sollte man sich genau entscheiden für welches Einsatzgebiet man die Erweiterung benötigt. Je vielseitiger die Einsatzmöglichkeiten sein sollen umso eher greift man zum Produkt von Oracle. Aufgrund der hohen Anschaffungspreise für die einzelnen Systeme eignen sich diese für private Anwendungen nicht.

7.3 Vergleich der kommerziellen und Open – Source Geodatenbanksysteme

Dieser Vergleich zwischen den beiden verschiedenen Arten hinkt ein wenig, obwohl die Open – Source Systeme immer weiter aufholen und den kommerziellen Produkten immer mehr zu Konkurrenz werden. Allen voran PostGIS kann vom Leistungsumfang mit den kommerziellen Produkten einigermaßen Stand halten. Auch MySQL kann im Bereich der einfach gestrickten Webanwendungen durchaus überzeugen. Jedoch weisen die Open – Source Produkte auch einige Schwächen auf. Allenfalls die neu entwickelten oder neu implementierten Standards finden meist erst relativ spät Einzug in die Produktumgebung. Dies liegt vor allem an der freien Entwicklergemeinde. Aus diesem Grunde fehlen unter anderem Netzwerk- und Topologiemodelle. Auch die Unterstützung von Rasterdaten und 3D Geodaten fehlen oder sind nur teilweise umgesetzt. Die größte Schwäche liegt im Bereich der Nutzung von räumlichen Bezugssystemen. Diese können in MySQL gar nicht und PostGIS nur teilweise eingesetzt werden. Weiterhin fehlen Pakete aus dem SQL / MM Spatial Standard. So gibt es Probleme bei der Verwendung von Anfragebearbeitungen zum Beispiel bei der Nächste – Nachbar – Abfrage. Erschwerend kommt hinzu, dass die meisten größeren Geoinformationssystemhersteller die Open – Source Datenbanksysteme nicht

unterstützen. Zur Verwendung müssen erst entsprechende Softwareprogramme eingesetzt werden.

Resümierend kann man zusammenfassen, dass als beste Lösung für die Verwaltung von Geodaten ein kommerzielles Produkt anzusehen ist. Diese bieten vollen Support durch den Hersteller. Des Weiteren sind alle wichtigen Standards komplett implementiert und können eingesetzt werden. Große Datenmengen lassen sich ebenfalls gut verarbeiten. Für kleine Heim- und Netzwerkanwendungen sind jedoch meist freie Produkte die bessere Wahl. Die nachfolgenden Tabellen zeigen nochmals vergleichend die Stärken und Schwächen der beiden Systeme auf.

	Kommerzielle Datenbanken	Open – Source - Datenbanken
Umsetzung Simple Feature – Modell	komplett	nur teilweise umgesetzt
Umsetzung SQL / MM Spatial	komplett	nur teilweise umgesetzt
Verwendung räumlicher Indizes	ja	ja
Preis	käufliche Lizenzen, aber auch frei verfügbare Versionen für Heimanwender mit eingeschränktem Funktionsumfang	frei verfügbar, jedoch bieten einige Hersteller auch kaufpflichtige Produktumgebungen an

Verwaltung von großen Datenmengen	große und kleine Datenmengen lassen sich verarbeiten	die Mehrheit der Produkte eignet sich eher für kleinere Geodatenmengen
Anwendungsbereich	lassen sich überall einsetzen, außer für den Heimgebrauch nicht die beste Lösung	Einsatz fast überall möglich, vorrangig eignen sich aber Heim- und Webanwendungen
Hardwareanspruch	hohe Hardwareanforderungen und im Dauerbetrieb werden hohe Rechenleistungen erwartet	Geringe Anforderungen an Hardware und geringe Speicherbelastung im Betrieb
Verarbeitung von Rasterdaten	meist entsprechende Erweiterungen mit entsprechendem Funktionsumfang	nur durch Erweiterungen nutzbar, eingeschränkter Funktionsumfang

Tabelle 5: Vergleich zwischen Open – Source und kommerziellen Systemen

8 Ausblick und aktuelle Trends

Das letzte Kapitel gibt einen abschließenden Ausblick über die weiteren Entwicklungen im Bereich der Geodatenbanken. Hier sollen kurz aktuelle Trends vorgestellt werden, die sich noch in der Entwicklung, der Testphase oder im Einsatz erprobt haben. Natürlich bieten die nachfolgenden Abschnitte keinen vollständigen Überblick aber geben immerhin Aufschluss über die wichtigsten Trends, die es in den nächsten Jahren sicher genauer zu betrachten gilt.

8.1 Mobile Geodatenbanksysteme

Mit der immer größeren Verbreitung von mobilen Endgeräten, wie Notebooks oder Smartphones werden auch mobile Geodatenbanken oder die Location – Based Services immer wichtiger. Dabei sind mobile Daten jedoch differenzierter zu betrachten wie normale Geodaten. Mobile Daten befinden sich auf mehreren mobilen Geräten. Dabei können die Daten gleichzeitig gelesen, geändert und geschrieben werden obwohl keine Netzwerkverbindung zwischen den Geräten oder zu irgendeiner anderen Quelle im Netz besteht. Dabei stellt die Synchronisation dieser Daten eines der Hauptaufgaben von mobilen Datenbanksystemen dar. Zur Realisierung dieser Aufgabe existieren verschiedene Herangehensweisen. Das größte Problem dabei besteht darin die Datenintegrität zu gewährleisten. Zu den bekanntesten Ansätzen gehören die Client / Server – Architektur, das System der verteilten Datenbanken, mobil verteilte Datenbanken und ein erweitertes Client / Server – Prinzip, wobei sich letztere heutzutage durchgesetzt hat. Im Grunde genommen beschreibt diese Architektur nur ein Anfrage – Antwort – Protokoll zwischen einem Client und einem Datenbankserver. Prinzipiell beginnt die Verarbeitung mit einer Anfrage des Clients an den Server, der die Anfrage bearbeitet und das Resultat dieser Anfrage sofort an den Client zurück schickt. Innerhalb dieser Architektur können die Clients und die Server nach der Zugehörigkeit verschiedener Schichten eingeteilt werden. Wie eine solche Verteilung aussehen kann zeigt die nachfolgende Abbildung.

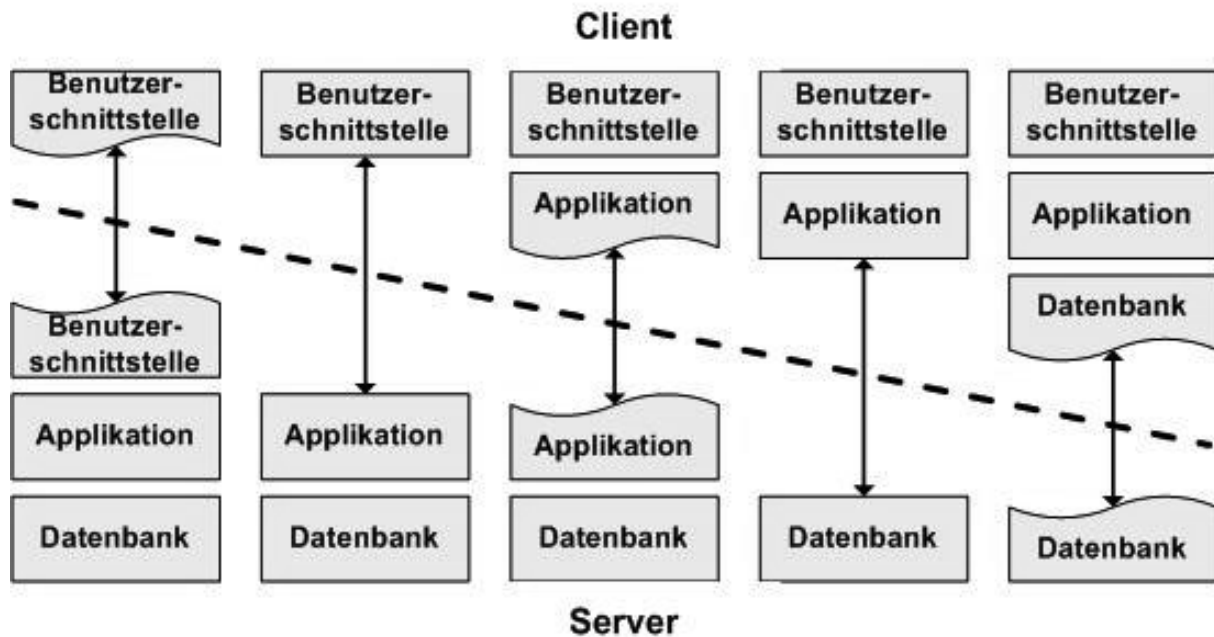


Abbildung 14: Schichtung Client – Server Architektur [UHH05]

In der Regel werden die mobilen Geräte mit weniger Schichten bedacht. Doch mit der zunehmenden Entwicklung von immer leistungsfähigeren Geräten wird diese heutzutage dazu genutzt Daten abzuspeichern und die eigentliche Applikation an sich auf mobiler Seite laufen zu lassen. Die erweiterte Client – Server – Architektur ergänzt die klassische Variante um den Ansatz der verteilten Datenbanken. Dies setzt somit den Standard für mobile Datenbanksysteme. Die mobilen Geräte sind mit Replikaten der Daten ausgestattet. Replikation bezeichnet die mehrfache Speicherung derselben Daten an meist mehreren verschiedenen Standorten und die Synchronisation dieser Datenquellen (Wikipedia, 2012). Mit diesen Replikaten kann dann auf den Clients lokal gearbeitet werden. Ein Austausch der Daten zwischen den Clients erfolgt mit Hilfe eines Datenbank – Servers. Ein Zugriff auf die Daten ist nur möglich durch die Bereitstellung eines Replikations – Server. Dieser kann als Middleware angesehen werden und dient gleichzeitig zur Umsetzung der Synchronisation. Die Synchronisation beschreibt das Abgleichen der Daten.

8.2 Spatio – temporale Datenbanken

Wie aus den anderen vorherigen Kapiteln hervorging, eignen sich Geodatenbanken zu effektiven Verwaltung der verschiedenen Eigenschaften von Geodaten besonders gut. Besonders wichtig sind bei Geodaten die Beobachtungen über einen

bestimmten Zeitraum ohne dabei den Raumbezug zu verlieren. Somit sind der Raumbezug und der Zeitbezug eng miteinander verzahnt. Sowohl räumliche als auch temporale Datenbanken sind damit nicht geeignet, bestimmte Operationen und Datenstrukturen bereit zu stellen. Dies führte in letzten Jahren zu der Entwicklung von spatio – temporalen Datenmodellen sowie dann später auch zu spatio – temporalen Datenbanken. Daraus entwickelte sich später der Begriff 4D – GIS.

Die Komponente der Zeit lässt sich nicht so einfach in ein bekanntes Datenbankschema integrieren, denn die Zeit kann nicht als zusätzliche räumliche Dimension ins Modell aufgenommen werden. Die Verbindung von Raum und Zeit weist nämlich andere Eigenschaften auf. So muss zum Beispiel zwischen diskreten und kontinuierlichen Veränderungen bei der Modellierung unterschieden werden. Wohingegen diskrete Veränderungen noch gut in bekannten Modellen nachbilden lassen, wird das bei kontinuierlichen Veränderungen schon weitaus schwieriger. Geobjekte oder auch Personen oder Fahrzeuge bewegen sich meist kontinuierlich und verändern dabei ihre Position beziehungsweise ihre Gestalt. Ist nur die Positionsveränderung von Bedeutung so spricht man von einem „Moving Point“. Datentypen für die Verwaltung von Zeitangaben lassen sich in der ISO 1908 „Geographic Information“ Norm finden. In diesem Standard sind 2 verschiedene Primitive zur Festlegung von Zeiten verankert. So können Angaben zum Zeitpunkt und zum Zeitraum gemacht werden. Auch stehen Operationen zur Verarbeitung dieser Primitiven zur Verfügung.

8.3 Spatial Data Mining

Unter Data Mining werden verschiedene Techniken verstanden, die es erlauben, in Datenbeständen bisher unerkannte Informationen und Zusammenhänge zu erkennen [Bri08]. Der Begriff lässt sich sogar bis in die 1950er Jahre zurück zu verfolgen. Der Begriff steht eng im Zusammenhang mit „Knowledge Discovery in Databases“ (KDD). In manchen Kreisen besitzen beide Begriffe dieselbe Bedeutung, obwohl das Data Mining als Analyse im KDD eingesetzt wird. Der Prozess des Data Mining setzt sich aus den folgenden Schritten zusammen

- Datenintegration und Bereinigung,

- Datenselektion und Transformation,
- Suche von Muster und Trends,
- Wissensfindung und Konstruktion,
- Auslieferung.

Im Unterschied zum allgemeinen Data Mining wird beim Spatial Data Mining in räumlichen Informationen nach gewissen Zusammenhängen gesucht. Hier kommen besonders die topologischen Eigenschaften von Geodaten und die Entfernung zwischen zwei Objekten zum Tragen. So können vor allem räumliche Cluster identifiziert werden. Mit dem Clustering möchte man versuchen die Daten in bestimmte Gruppierungen aufzuteilen. Alle Objekte die sich in einem oder mehreren Merkmalen ähneln, bilden eine zusammengehörige Gruppe. Die Verteilung der Objekte ergibt sich aus der Autokorrelation. Das Clustering kann im einfachsten Fall durch eine Abstandsfunktion bestimmt werden. Jedoch gibt es auch kompliziertere Algorithmen zur Berechnung.

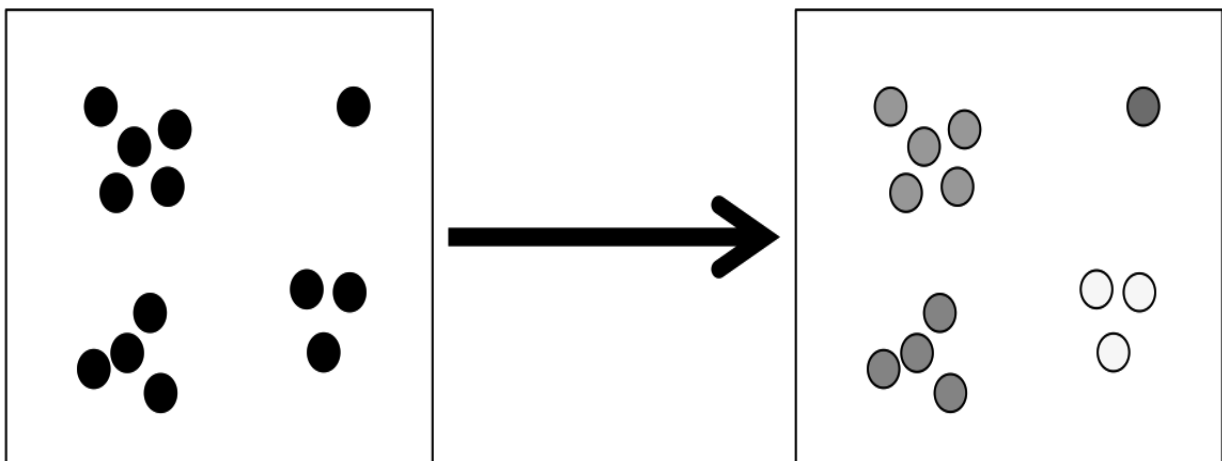


Abbildung 15: Beispiel Clustering [UOI11]

Eine weitere Aufgabe des Spatial Data Mining ist das Auffinden von räumlichen aber auch spatio – temporalen Abhängigkeiten. Zur Bestimmung räumlicher Abhängigkeiten werden vorrangig das Variogramm oder die „Moving Window“ – Statistik eingesetzt. Für spatio – temporalen Abhängigkeiten gibt es bisher noch keine gültigen Auswertemethoden.

Auch zur Charakterisierung von Geodaten kann das Spatial Data Mining genutzt werden. Hier können Aussagen über Gestalt der Geometrie oder über die

Komplexität der Objekte getroffen werden. Als weitere Zielsetzung kann die Bestimmung räumlicher Trends angesehen werden. Hier sollen Ausbreitungs- und Bewegungsmuster von Objekten untersucht werden. Man versucht hierbei die Änderung beziehungsweise die Trends in den Daten entlang einer oder mehrerer Dimensionen im Raum zu bestimmen. Danach werden die Tendenzen in den räumlichen Attributen gesucht. Dabei kann zwischen globalen und lokalen Trends unterschieden werden.

Im Datenbanksystem Oracle gibt es ein Paket, das räumliche Methoden des Spatial Data Mining bereitstellt. Seit der Version 10 befinden sich diese im Paket SDO_SAM. Mit der Funktion SPATIAL_CLUSTERS lassen sich räumliche Cluster bestimmen. Zusätzlich zu dieser Funktion lassen sich mit Hilfe der Funktion BIN_GEOMETRY verschiedene Geometrien den Clustern zuordnen.

8.4 Sensornetzwerke und Datenströme

Sensoren dienen dazu, Messwerte automatisiert zu erfassen [Bri08]. So lassen sich dann Temperaturen, Lautstärken oder Abstände messen. Durch die immer schnellere Entwicklung in der Mikrotechnologie ist es heutzutage möglich Sensoren oder Prozessoren auf kleinsten Mikrochips unterzubringen. Bei einer Einheit von bis zu vier Sensoren, einer Funkeinheit und einer Stromversorgungseinheit spricht man von einem Sensorknoten. Mehrere dieser Sensorknoten können ein Sensornetzwerk bilden. Damit das Sensornetz einwandfrei arbeitet, werden mindestens vier Sensorknoten, eine Basisstation zum Empfang der Daten und eine spezielle Software benötigt, die für die Verarbeitung und Sicherheit der einzelnen Bestandteile des Netzes sorgt. Je mehr Knoten in einem Netz vorhanden sind umso leichter sind Beobachtungen aus der realen Welt zu realisieren und auch die Genauigkeit der Messungen nimmt dadurch zu. Anwendung finden diese Netzwerke zum Beispiel in der medizinischen Überwachung, im Facility Management, in der Lagerverwaltung oder auch im Natur- und Umweltschutz.

Die Sensorknoten besitzen zu dem eine Position im zwei oder dreidimensionalen Raum. Für manche Anwendungen ist die Bestimmung der exakten Position von größter Bedeutung. Demnach nimmt ein solcher Sensorknoten also auch Geodaten

auf. Auch kann die Zeit in einigen Applikationen eine besondere Rolle spielen. Sollte diese nicht simultan übertragen werden, ist es ebenfalls noch notwendig einen Zeitstempel zu übergeben. Von bewegten Objekten an denen der Sensor befestigt ist, können aus der Position Größen wie Geschwindigkeit, Beschleunigung oder die Bewegungsrichtung des Objektes abgeleitet werden. Bei der kontinuierlichen Datenübertragung von Messdaten an Zielknoten entsteht ein so genannter Datenstrom. Um solche Datenströme in einer Datenbank abzulegen benötigt man spezielle Verwaltungssoftware, da Datenströme ein großes Datenvolumen aufweisen und eine hohe Änderungsrate aufzeigen. Diese Programme zur Verwaltung und zur Anfragebearbeitung nennen sich Datenstrommanagementsysteme (DSMS). Die meisten Abfragen des Datenstromes beziehen sich auf längere Abschnitte des Stromes. Dies nennt man auch „Continuous Query“. Solche speziellen Abfragen werden sofort im DSMS registriert. Danach werden kontinuierlich die Ergebnisse der Anfrage an ein Client – Programm übertragen.

Dieses vorhandene System lässt sich auch mit Datenbanken realisieren. Dazu muss der Sensorknoten mit Datenbankfunktionalitäten ausgerüstet werden. Dadurch wird dem Client – Programm ermöglicht, Abfragen an das Sensornetzwerk zu stellen. Die Abfrage wird dann an die einzelnen Sensorknoten weitergeleitet, die Ergebnisse ermittelt und dann zurück an die Datenbank übertragen werden. Somit kann das Sensornetzwerk als Teil eines Geoinformationssystem aufgefasst werden.

Literatur- und Quellenverzeichnis

Bücher und Zeitschriftenartikel:

- [Bil10] **Bill, Ralf:** Grundlagen der Geo-Informationssysteme; Berlin : Wichmann, 2010.
- [Kud07] **Kudraß, Thomas:** Tachenbuch für Datenbanken; Carl Hanser Verlag GmbH & CO. KG, 2007.
- [Bri08] **Brinkhoff, Thomas:** Geodatenbankensysteme in Theorie und Praxis; München : Wichmann Verlag, 2008.
- [Bri07] **Brinkhoff, Thomas:** Datenbank – Spektrum; Heft 22, dpunkt-Verlag, August 2007.
- [Stö11] **Störl, Uta:** Geodatenbanksysteme - Kapitel 2: Modellierung von Geodaten; Darmstadt : Hochschule Darmstadt, 2011.
- [Bri06] **Brinkhoff, Thomas:** Photogrammetrie, Fernerkundung, Geoinformation (PFG); Heft 5/2006, 2006.
- [Gie03] **Giesen, Klaus:** SQL – Das bhv Taschenbuch; Bonn: verlag moderne industrie Buch AG & Co. KG, 2003.
- [KoÖ09] **Kofler, Michael; Öggel Bernd:** PHP 5.3 & MySQL 5.4: Programmierung, Administration, Praxisprojekte; Berlin : Addison-Wesley Verlag, 2009.
- [ObH11] **Obe, Regina O.; Hsu, Leo S:** PostGIS in Action; Stanford: Manning Publication Co, 2011.

Internetquellen:

- [UOs12] <http://www.informatik.uni-osnabrueck.de/um/95/95.1/oodb/oodb.html>
- Universität Osnabrück, Fachbereich Informatik / Zugriff am 16.07.2012
- [ITW12] <http://www.itwissen.info/definition/lexikon/Objektorientierte-Datenbank-ODB-object-oriented-database.html>
- Zugriff am 16.07.2012
- [WiL12] http://de.wikipedia.org/wiki/Liste_der_Datenbankmanagementsysteme
- Zugriff am 16.07.2012
- [WiO12] <http://de.wikipedia.org/wiki/Objektdatenbank>
- Zugriff am 16.07.2012
- [FHK12] http://wikis.gm.fh-koeln.de/wiki_db/Onlinelexikon/Startseite
- Fachhochschule Köln, Campus Gummersbach / Zugriff am 7.07.2012
- [IWO12] <http://www.itwissen.info/definition/lexikon/Objektrelationale-Datenbank-object-relational-database.html>
- Zugriff am 18.07.2012
- [JUO12] <http://www.fergi.uni-osnabrueck.de/module/geodstd/>
- Jade Universität Osnabrück / Zugriff am 18.07.2012
- [Gol12] <http://www.golem.de/0203/19055.html>
- Zugriff 23.07.2012
- [XML12] <http://xml.apache.org/xindice/>
- Zugriff 23.07.2012
- [Ras12] <http://rasdaman.eecs.jacobs-university.de/trac/rasdaman/wiki>
- Zugriff 23.07.2012

- [FHK12] http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Tamino
- Fachhochschule Köln, Campus Gummersbach / Zugriff 25.07.2012
- [ITW12] http://www.imn.htwk-leipzig.de/~kudrass/Lehrmaterial/Oberseminar/2010/10-Spatial_Data_Mining-Gaebler.pdf
- Hochschule für Technik, Wirtschaft und Kultur Leipzig / Zugriff 26.07.2012
- [ITW12] http://www.knowscore.de/javacd/doc/javainasel4/javainasel_20_017.htm#Rxx365java20017040008431F032103
- Zugriff 23.07.2012
- [MyS12] <http://dev.mysql.com/doc/refman/5.1/de/spatial-extensions.html>
- Zugriff 31.07.2012
- [ETH12] http://www.dbs.ethz.ch/education/ordb/WS_01/fohlen/8-Erweiterbarkeit.pdf
- ETH Zürich Computer Science Department / Zugriff 29.07.2012
- [DMS12] <http://database-management-systems.findthebest.com/compare/9-17/DB2-vs-Informix-Dynamic-Server>
- Zugriff 30.07.2012
- [INB12] <http://database-management-systems.findthebest.com/compare/9-36/DB2-vs-Oracle>
- Zugriff 30.07.2012
- [DGO12] http://docs.oracle.com/cd/E16338_01/appdev.112/e11830.pdf
- Zugriff 27.07.2012
- [SDB12] <http://publibfp.dhe.ibm.com/epubs/pdf/22964053.pdf>
- IBM Informix Spatial DataBlade Module User's Guide / Zugriff 29.07.2012

- [UAU12] http://www.informatik.uni-augsburg.de/lehrstuehle/dbis/db/lectures/ss11/mobileGIS/themen/Thema5_Ausarbeitung_Winter.pdf
- Universität Augsburg, Fachbereich Informatik / Zugriff 27.07.2012
- [UAu12] http://www.informatik.uni-augsburg.de/lehrstuehle/dbis/db/lectures/ss11/mobileGIS/themen/Thema20_Ausarbeitung_Raddatz.pdf
- Universität Augsburg, Fachbereich Informatik / Zugriff 27.07.2012
- [USI12] http://pi.informatik.uni-siegen.de/lehre/2000w/2000w_proseminar/ausarbeitungen/ORDBMS.pdf.gz
- Universität Siegen, Fachbereich Informatik / Zugriff 23.07.2012
- [FHO12] http://iapg.jade-hs.de/events/fergi_geodbs_2007/Brinkhoff_AktuelleEntwicklungen.pdf
- FH Oldenburg, Institut für Angewandte Photogrammetrie und Geoinformatik / Zugriff 01.08.2012
- [DiA02] http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/93810/diplomarbeit_smallworld_be.pdf?command=downloadContent&filename=diplomarbeit_smallworld_be.pdf
- Hochschule Neubrandenburg, Diplomarbeit / Zugriff 27.07.2012
- [FHT12] http://public.fh-trier.de/~weigla/study/oraws_ausarbeitung.pdf
- Fachhochschule Trier, Hausarbeit / Zugriff 28.07.2012
- [DB212] <ftp://ftp.software.ibm.com/ps/products/db2/info/vr9/pdf/letter/nlv/DE/db2sbg90.pdf>
- Spatial Extender und Geodetic Data Management Feature Benutzerhandbuch / Zugriff 29.07.2012

- [HTK10] <http://www.imn.htwk-leipzig.de/~kudrass/Lehrmaterial/Oberseminar/2000-01/SpatialDB.pdf>
- Zugriff 30.07.2012
- [UHH05] <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2005/thome/abstract.pdf>
- Zugriff 26.07.2012
- [UOI11] <http://www.informatik.uni-oldenburg.de/~iug11/ge/website/images/clustering.png>
- Universität Oldenburg, Fachbereich Informatik / Zugriff 02.08.2012
- [Wik12] http://de.wikipedia.org/wiki/Relationale_Datenbank
- Wikipedia / Zugriff 11.08.2012
- [UHH11] <http://print-www.informatik.uni-hamburg.de/Dokumentation/mysql/DSP/>
- Universität Hamburg / Zugriff 28.07.2012
- [GIS12] <http://gisisit.wikispaces.com/>
- Zugriff 11.08.2012
- [Tut12] <http://www.gaia-gis.it/gaia-sins/spatialite-tutorial-2.3.1.html#t4.3>
- Tutorial SpatiaLite / Zugriff 12.08.2012
- [OTN12] http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/parallel_query/index.html
- Zugriff am 24.08.2012

Abbildungsverzeichnis

Abbildung 1: 4 Komponentenmodell GIS [JUO12].....	9
Abbildung 2: Darstellung Simple Features [JUO12]	14
Abbildung 3: Geometrieschema Simple Features [JUO12]	14
Abbildung 4: Geometrieschema SQL / MM Spatial [JUO12]	16
Abbildung 5: Pakete der ISO 19107 Norm [JUO12]	19
Abbildung 6: Geometrieschema Feature – Geometry Modell [JUO12].....	20
Abbildung 7: Allgemeiner Aufbau Datenbanksysteme [UHH11]	23
Abbildung 8: Aufbau des relationalen Datenbankmodells [Wik12].....	24
Abbildung 9: Systemarchitektur [SDB12].....	47
Abbildung 10: Geometriemodell Informix Spatial [SDB12].....	48
Abbildung 11: Systemarchitektur Rasdaman Rasterserver [Ras12].....	53
Abbildung 12: Systemarchitektur Tamino XML Server [FHK12]	54
Abbildung 13: 9 – Intersection – Modell [GIS12].....	57
Abbildung 14: Schichtung Client – Server Architektur [UHH05].....	77
Abbildung 15: Beispiel Clustering [UOI11].....	79

Tabellenverzeichnis

Tabelle 1: nicht definierte Funktionen Simple – Feature – Modell [Bri08].....	17
Tabelle 2: Geometriattribute der Klasse SDO_GEOMETRY [Bri08].....	43
Tabelle 3: Vergleich Open – Source Geodatenbanken [Bri07]	70
Tabelle 4: Vergleich kommerzielle Geodatenbanksysteme	72
Tabelle 5: Vergleich zwischen Open – Source und kommerziellen Systemen	75

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Neubrandenburg, den

Unterschrift

Danksagung

Abschließend möchte ich mich bei allen bedanken, die mich bei der Erstellung meiner Bachelorarbeit mit Korrekturlesen, Hinweisen und Diskussionen unterstützt haben. Ich bedanke mich herzlich bei Herrn Prof. Dr. Andreas Wehrenpfennig und Herrn Prof. Dr. Ernst Heil für die Betreuung meiner Bachelorarbeit und das kritische Lesen der Arbeit während der Entstehung. Ein großer Dank gilt meinen Eltern, die mir dieses Studium ermöglicht haben.