

**Modul Nr. 13193 + 13194,
M.Sc. Bau: Geodäsie und GIS
Teil GIS – Kapitel 4:
Datenbanken**



Univ.-Prof. Dr.-Ing. Wolfgang Reinhardt
AGIS / Inst. Für Angewandte Informatik (INF4)
Universität der Bundeswehr München
Wolfgang.Reinhardt@unibw.de
www.agis.unibw.de

Überblick – Kap. 4

- Einführung Datenbanken
- Arten von Datenbanken
- Eigenschaften von Datenbanken
- Relationale Datenbanken
- SQL
- Geodatenbanken

⇒ Praktische Vertiefung in den Übungen

Literatur

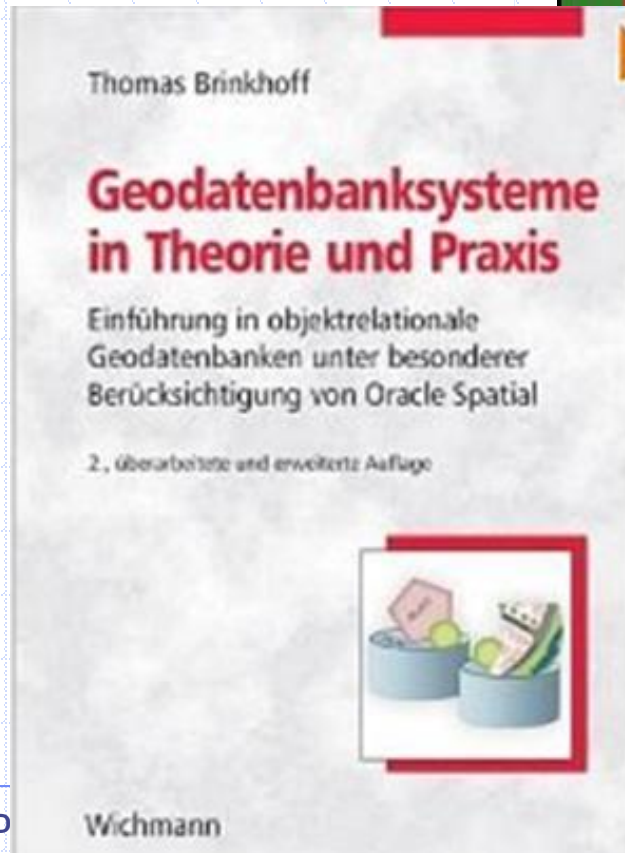
Zu Datenbanken große Auswahl!

Ein Beispiel:

Zu Geodatenbanken:



D



Allgemeine Datenbankeigenschaften

Datenbank - Wieso? Weshalb? Warum?

- Daten unabhängig von Anwendungsprogrammen
- Selbstbeschreibend (Schema als Metadaten)
- Simultanzugriff (Viele Nutzer teilen gleiche Daten)
- Verschiedene Sichten (der untersch. Benutzer)
- Zugriffskontrolle (Benutzer)
- Fehlertoleranz / Transaktionen
- Datenkonsistenz (Kontrolle, Sicherstellung)
- Datensicherheit (Wiederherstellbarkeit)

Allgemeine Datenbankeigenschaften

Weitere Vorteile

- Nutzung von Standards (Zugriff, Bedienung ..)
- Effizienter Datenzugriff
- Grosser Funktionsumfang
- Hohe Flexibilität (Nutzeranforderungen)
- Hohe Verfügbarkeit (s. z.B. banking)
- Gute Wirtschaftlichkeit

Allgemeine Datenbankeigenschaften

Arten von Datenbanken

- Relationale Datenbanken - RDB
im weiteren primär betrachtet
- Objektorientierte Datenbanken
- Mischformen (z.B. Objektrelationale)
- Weitere (z.B. hierarchische)
- Vgl. auch Folien zu Geodatenbanken

Allgemeines zu Datenbanken

Relationale Datenbanken –

Beispiele für Software Produkte / Hersteller:

- Oracle
- DB2 (IBM)
- MySQL (open source)
- SQL-Server (MS)
- PostgreSQL (open source)
- MS Access
- U.v.m.

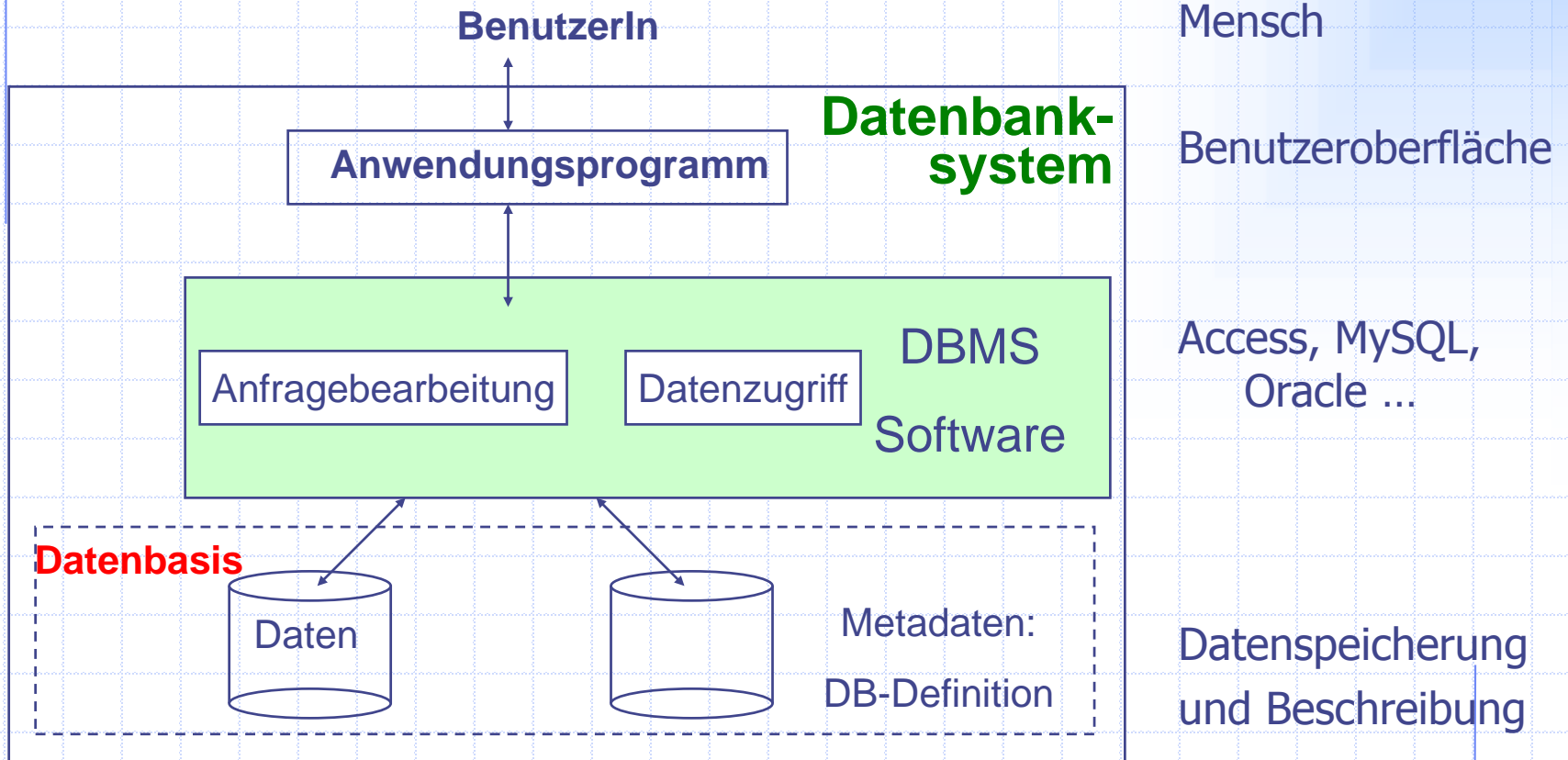
Architektur

Allgemeine Datenbankeigenschaften

Geodatenbanken

Räumlicher Index

Räumliche Operatoren



Mensch

Benutzeroberfläche

Access, MySQL, Oracle ...

Datenspeicherung und Beschreibung

Überblick

Allgemeine Datenbankeigenschaften

Geodatenbanken

Räumlicher Index

Räumliche Operatoren

Anforderungsanalyse

▶ Text



Konzeptionelles Schema

▶ ER-Schema (o. UML)

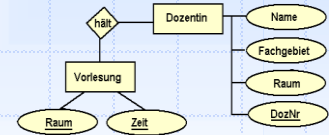
Logisches Schema

▶ DB-Schema

Implementierung Schema

Anwendung/Nutzung

Wartung und Pflege



```
CREATE TABLE studentIn  
(MatrNr INTEGER PRIMARY KEY,  
Nam  
Stuc  
CREATE TABLE Vorlesung  
(VorNr CHAR(10) PRIMARY KEY,  
Titel CHAR(40),  
Raum CHAR(10),  
CREATE TABLE DozentIn  
(DozNr INTEGER PRIMARY KEY,  
Name CHAR(60) NOT NULL,  
Raum CHAR(10),  
Fachgebiet CHAR(80));
```

SQL

```
mysql> create database semesterplan;
```

```
mysql> create table vorlesung  
(vorlnr char(10) not null,  
titel char(40) not null,  
raum char(10) not null,  
wtag char(10) not null,  
zeit datetime,  
PRIMARY KEY (vorlnr)  
);
```

```
mysql> insert into vorlesung (vorlnr, titel)  
values  
( 'Inf 02',  
'Theorie der Datenbanken');
```

Definition Datei

Eine Datei...

...Eine Datei ist ein strukturierter Bestand inhaltlich zusammengehöriger Daten, die auf einem beliebigen Datenträger oder Speichermedium abgelegt bzw. gespeichert werden kann. Diese Daten existieren über die Laufzeit eines Programms hinaus und werden als nicht flüchtig oder **persistent** (Gegensatz: transient) bezeichnet. ...

Definition Datenbank

Eine Datenbank besteht aus ...

... der Datenbasis, d. h. einer Sammlung strukturierter Daten mit ihrem **Datenschema** und einem **Datenbankmanagementsystem** zur Verwaltung der Datenbasis.

Ein Datenbankmanagementsystem (DBMS) ...

... ist ein Softwaresystem, das die Daten in der Datenbasis entsprechend vorgegebener Beschreibungen effizient verwaltet (Speichern, Auffinden, weitere Operationen). Typischerweise enthält ein DBMS Routinen zur **Dateneingabe, Datenüberprüfung, Speicherung, Abfrage, Kombination und Analyse.**

RDB - Datenbankgrundlagen

Datenbankdesign:

Anforderungsanalyse, was wird benötigt!

Beschreibung mit ER (Entity Relationship) – Diagramm
bzw. auch Entity Relationship Modell (ERM) genannt

Bzw. alternativ mit UML

Entity = etwas, das existiert

- Bsp: Studentin, Dozentin, Vorlesung

Attribute = Eigenschaften eines Entity

- Bsp: Name, Studiengang, Titel, Raum, Zeit

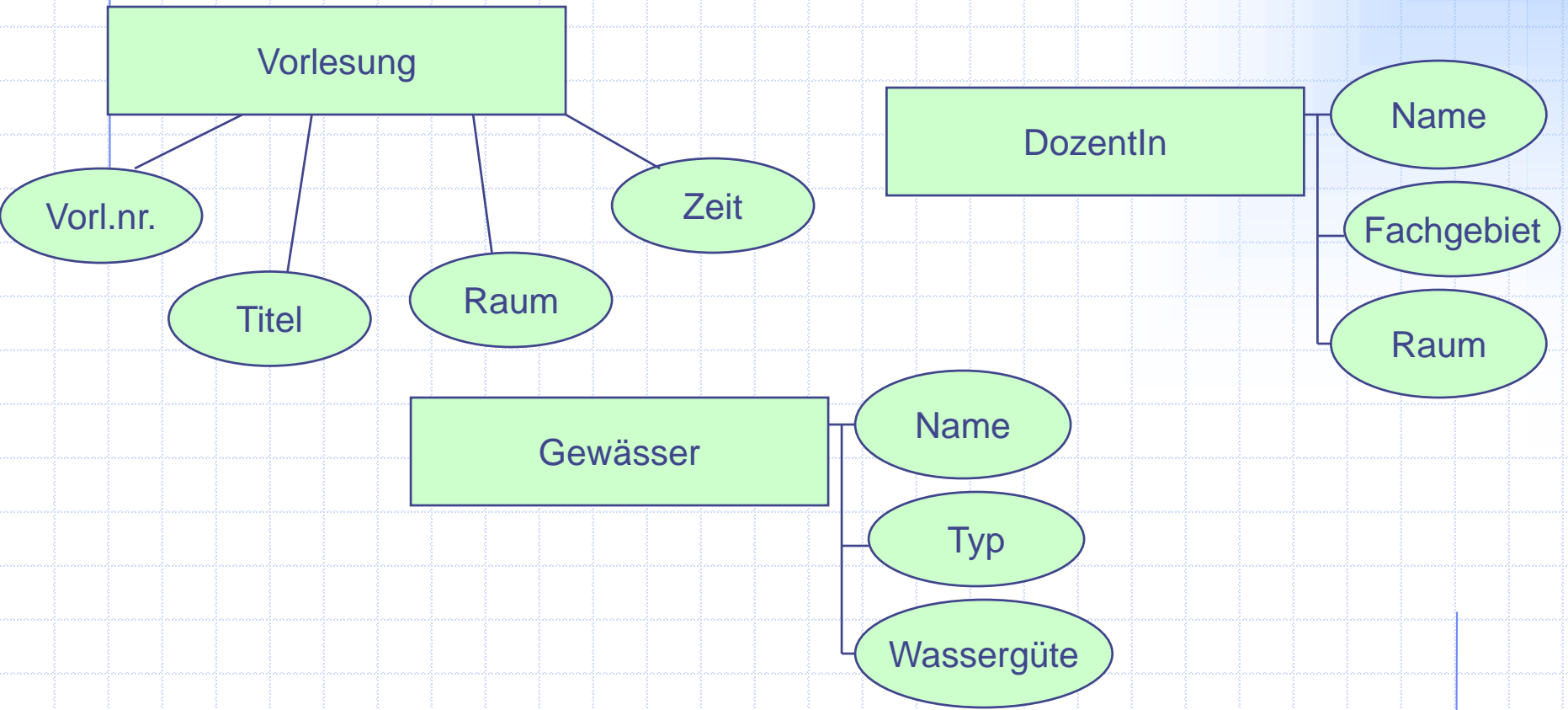
ERM

Geodatenbanken

Räumlicher Index

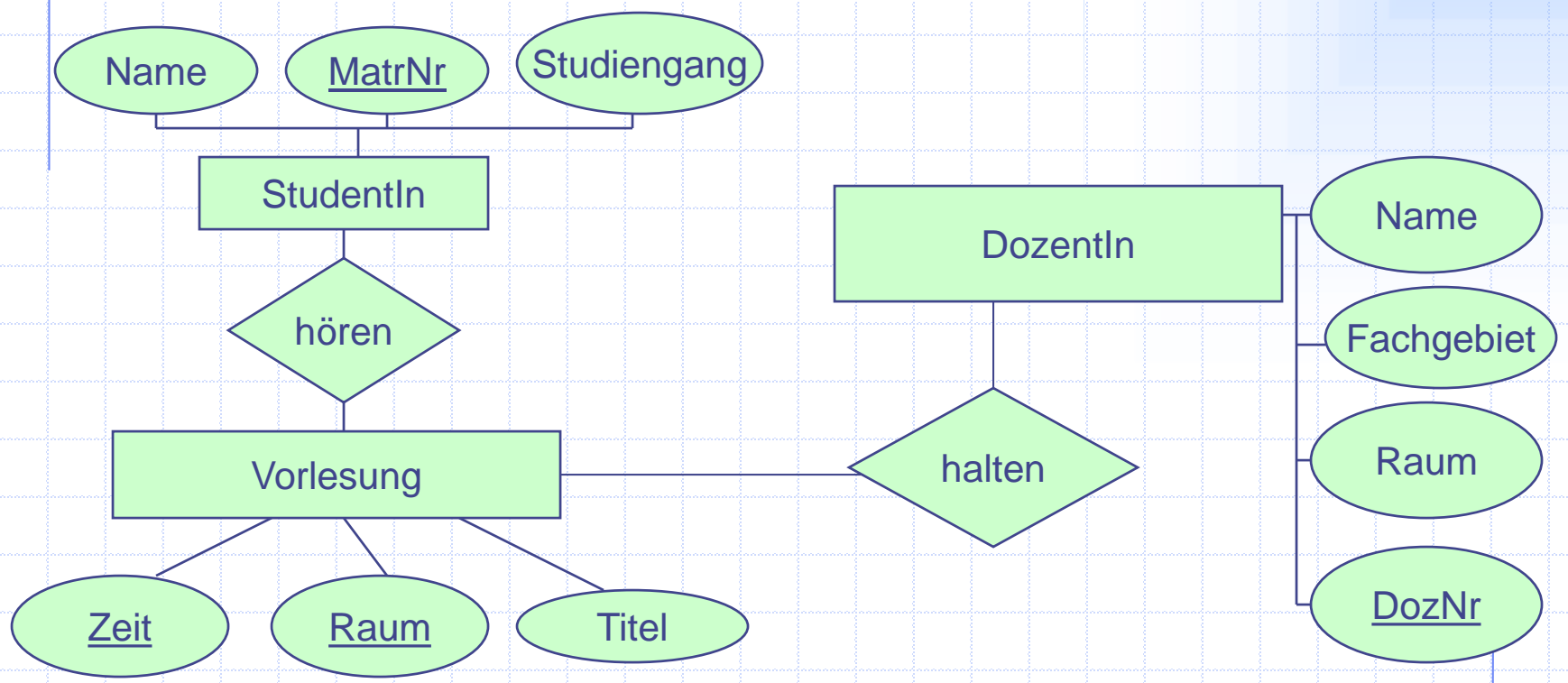
Räumliche Operatoren

Beispiele – Entites, Attribute:



ERM

Beispiele –ER-Modell:



Relationales Modell (1)

Math. Hintergrund der RDB (math. Definition von Operationen etc.)

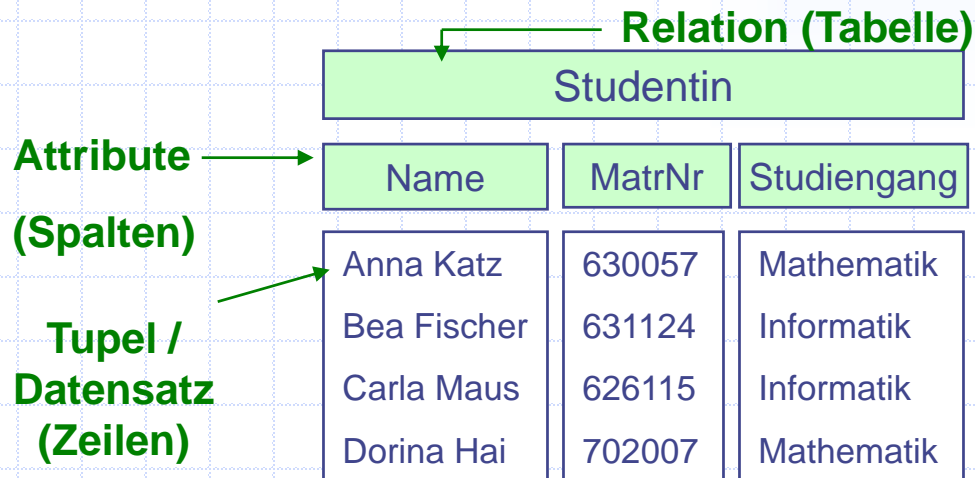
- Relationale Algebra
 - ◆ Mathematische Struktur
- Relationales Schema
 - ◆ Rel.-name (Attribut₁, ..., Attribut_n)
- Relation
 - ◆ Menge von Tupeln (auch: „n-Tupel“ oder verkürzt „Tupel“ bezeichnet eine Sammlung mit einer beliebigen Anzahl n von Attributen)
- Mengen plus Operationen
 - ◆ Projektion, Selektion, Durchschnitt, Differenz,...
- Wertebereiche für Tupel

Relationales Modell (2)

ER-Modell leicht in Relationales Modell (RM) und RDB transformierbar (s. Folie 22)

Relationale Datenbank (RDB)

Grundstruktur:



RDB - Datenbankgrundlagen

Implementierung in Tabellen:

Vorlesung				
VorlNr	Titel	Raum	Wtag	Zeit
Inf 04	Geoinformationssysteme	341107	Di	13.00
Win 12	Einführung in Datenbanksysteme	343101	Do	8.00
Bau 121	Statik	342205	Mi	14.00

RDB - Datenbankgrundlagen

Datenbank Schlüssel, dient in einer Relationalen Datenbank dazu, die Tupel einer Relation eindeutig zu identifizieren,

ist also:

- Eindeutig
- Besteht aus ein oder mehreren Attributen
- Beispiel:
 - ◆ Matrikelnummer (StudentIn)
- Welcher Schlüssel für „Vorlesung“?

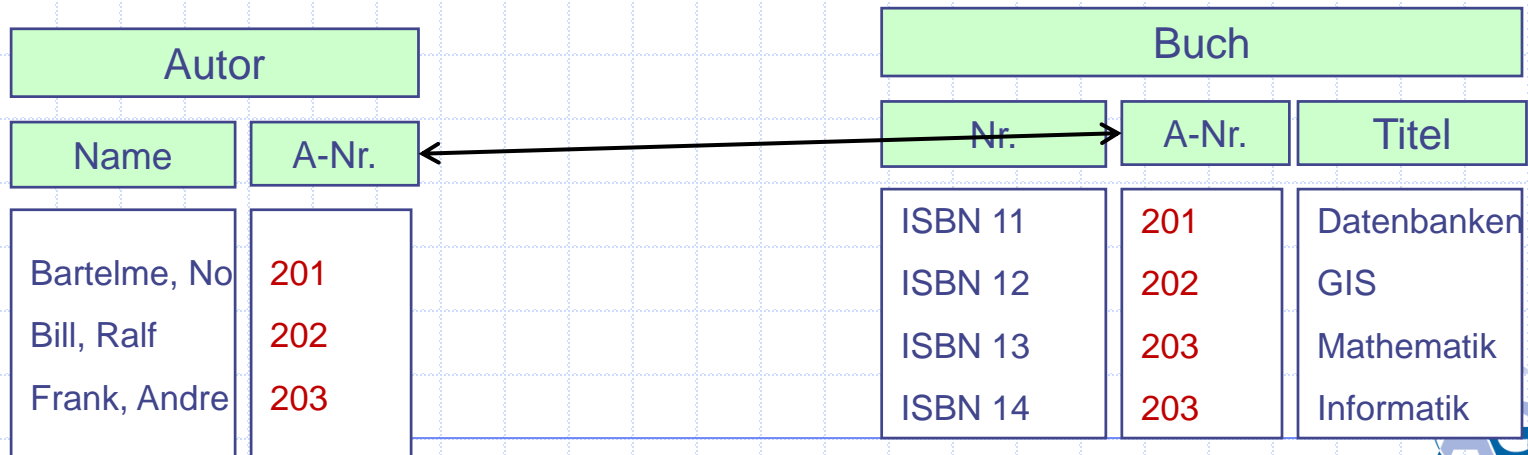
RDB - Datenbankgrundlagen

Ein Fremdschlüssel

ist ein Attribut einer Relation, das in einer anderen Relation Primärschlüssel oder Schlüsselkandidat ist.

Er dient als Verweis zwischen zwei Relationen, d. h. er zeigt an, welche Tupel der Relationen inhaltlich miteinander in Verbindung stehen.

Beispiel:



Vergleich von Begriffen / Konzepten

Implementierung	Math. Grundlagen	Modellierung	
Relationale Datenbank (RDB)	Relationales-Modell (RM)	Entity-Relationship-Modell (ERM)	Unified Modeling Language (UML)
Tabelle	Relation	Entität	Objekt-Klasse
Spaltenüberschrift	Attribut	Attribut	Attribut
Zeile	Tupel	Entität	Objekt, Instanz
Zelle	Attributwert	Attributwert	Attributwert
(Fremd)schlüssel	(Fremd)schlüssel	Relationship	Assoziation

Wiederholung:

- ERM / UML zur Modellierung der Anforderungen (-> konzeptionelles Modell)
- RM bildet math. Grundlage für RDB



Structured Query Language (-> Standard!)

- Anlegen der DB
- Füllen mit Daten / Änderungen / Löschen
- Zugriffsrechte
- Anfragen an Datenbank

Anlegen einer Datenbank

```
Sql> create database semesterplan
```

```
Sql> Create table vorlesung
```

```
(VorlNr char(10) not null,
```

```
Titel char(40) not null,
```

```
Raum char(10) not null,
```

```
Wtag char(10) not null,
```

```
Zeit datetime,
```

```
PRIMARY KEY (VorlNr)
```

```
);
```

```
Sql> insert into Vorlesung (VorlNr, Titel, Raum, Wtag, Zeit)
```

```
values
```

```
(„Inf 04“, „Geoinformationssysteme“, ..... );
```

Vorlesung				
VorlNr	Titel	Raum	Wtag	Zeit
Inf 04	Geoinformationssysteme	341107	Di	13.00
Win 12	Einführung in Datenbanksysteme	343101	Do	8.00
Bau 121	Statik	342205	Mi	14.00

Anlegen einer Datenbank

Ein Beispiel, wie es ein unerfahrener Nutzer machen könnte
(durch Überführung seiner „Papierstruktur“):

AuftrNr.	Datum	Kunde	ArtikelNr.	Bezeichnung	Menge
1	1.1.99	1 Schmitt, Bonn	134	Coxorange	4 Kisten
1	1.1.99	1 Schmitt, Bonn	135	Kiwi	4 Kisten
2	1.1.99	2 Müller, Köln	140	Butterbirne	2 Kisten
2	1.1.99	2 Müller, Köln	160	Kürbis, rot	2 Stück
2	1.1.99	2 Müller, Köln	160	Kürbis, gelb	10 Stück
3	2.2.99	1 Schmitz, Bonn	103	Johannisbeeren	5 Kilo
3	2.2.99	1 Schmitz, Bonn	134	Coxorange	12 Kisten
3	2.2.99	1 Schmitz, Bonn	135	Kiwi	2 Kisten
4	2.2.99	45 Lehmann, Jülich	30	Bananen	12 Kilo
4	2.2.99	45 Lehmann, Jülich	27	Ananas	60 Stück

Bewertung?



Anlegen einer Datenbank

Bewertung:

- Viele gleiche Einträge (Redundanz) sorgen nach ein paar Monaten dafür, dass die Datenbank unnötig groß wird.
- Tippfehler (letzte Zeile: Lehmann) machen das Auffinden aller Lieferungen an Lehmann, Jülich unmöglich.
- In der Spalte Kunde sind sowohl Kundennummer, Name und Ort zugleich eingetragen. Herr Krämer muß sich stets selber die Kundennummer merken und alle Daten bei jedem neuen Auftrag stets neu eintippen.
- Artikelnummer und Bezeichnung könnten als Synonyme verwendet werden, sind aber hier jeweils für sich in getrennten Spalten gespeichert.
- Die Farbe des Kürbis gibt eine andere Art an, die eine neue Artikelnummer erfordert.
- Die Spalte Menge enthält sowohl die Stückzahl als auch die Einheit der Ware (Kiste, Stück, Kilo). Die Einheit ist aber stets an die Art der Ware gekoppelt.

Anlegen einer Datenbank

Die **Normalisierung** eines relationalen Datenschemas

Überführt es in eine Form, die keine vermeidbaren Redundanzen enthält (Details zur Normalisierung / den Normalformen s. Literatur)

Das Beispiel: in der 2. Normalform

1. Tabelle:

AuftrNr.	Datum	KundenNr.	Name	Ort
1	1.1.99	1	Schmitt	Bonn

Bewertung?

Nicht wirklich zusammengehörende Attribute in einer Relation!

2. Tabelle:

AuftrNr.	ArtNr.	Bez.	Menge
1	134	Cox Orange	4

Anlegen einer Datenbank

Das Beispiel, vollständig normalisiert:

1. Tabelle:

ArtNr.	Bez.
134	Cox Orange

Bewertung?

2. Tabelle:

AuftrNr.	Menge	ArtNr.
1	4	134

3. Tabelle:

Auftr.Nr.	KundenNr.	Datum
1	1	1.1.99

4. Tabelle:

KundenNr.	Name	Ort (Anschrift...)
1	Schmitt	Bonn



Anfragen (Queries) an DB

```
SELECT * FROM tabelle WHERE bedingung;
```

Beispiel: Wie heißt die Vorlesung mit der Nummer Inf 04?

```
SELECT titel FROM Vorlesung WHERE VorlNr = Inf 04;
```

Weitere Beispiele!

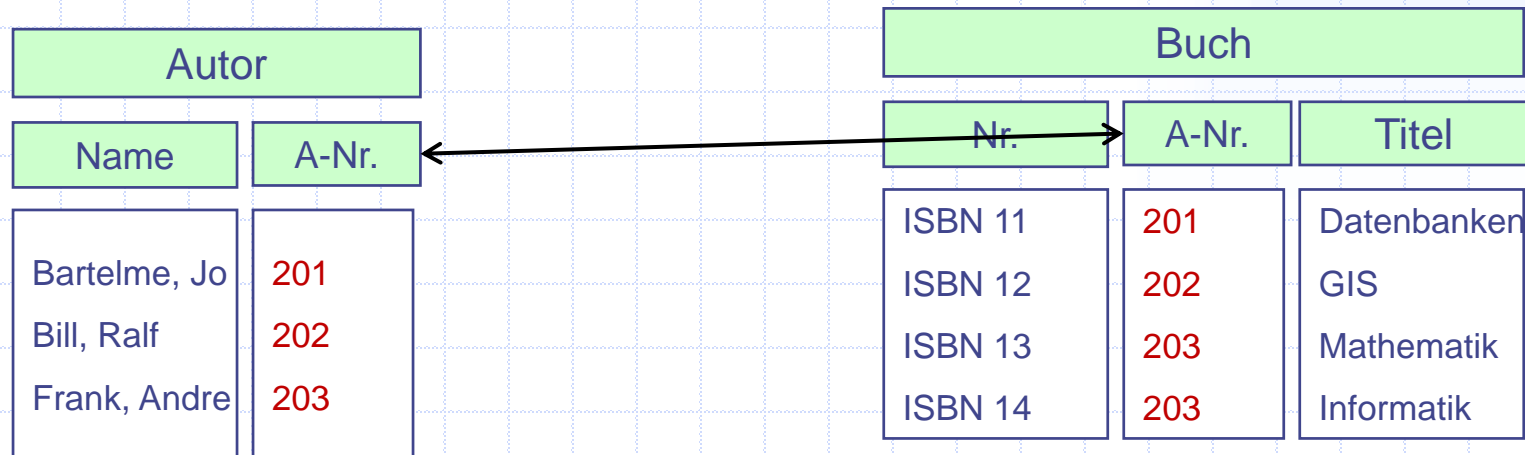
Anfragen über mehrere Tabellen

Sinn: alle möglichen Kombinationen der Zeilen ausgeben (ohne Bedingung)

⊗ ergibt das "Kreuz-Produkt" oder "kartesische Produkt",

Tabellen verbinden -> Auch „Tabellen-JOIN“ genannt, hier: „cross JOIN“ genannt

Beispiel von oben:

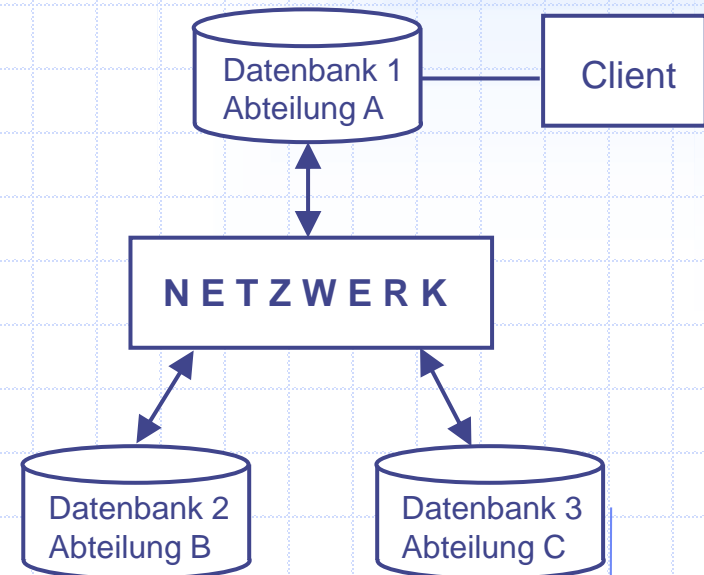


```
select autor.name, buch.titel
from autor, buch
where autor.A-Nr. = buch.A-Nr.
```

Allgemeine Datenbankeigenschaften (1a)

Verteilte Datenhaltung / verteilte Datenbanken

Teile einer Datenbasis befinden sich auf unterschiedlichen Rechnern (z. B. in einem Netzwerk), jeweils verwaltet durch das DBMS. Das DBMS führt diese zusammen und unterstützt den Zugriff auf die Datenbasen. Die verteilte Datenhaltung verläuft aus Sicht der Benutzer unbemerkt (Transparenz).



Beispiel: geographisch verteilte Datenhaltung

Allgemeine Datenbankeigenschaften (1b)

Vorteile:

- Lokalität der Daten (dezentrale Speicherung)
- Leistungssteigerung durch Parallelisierung (bei Anfrageberechnungen)

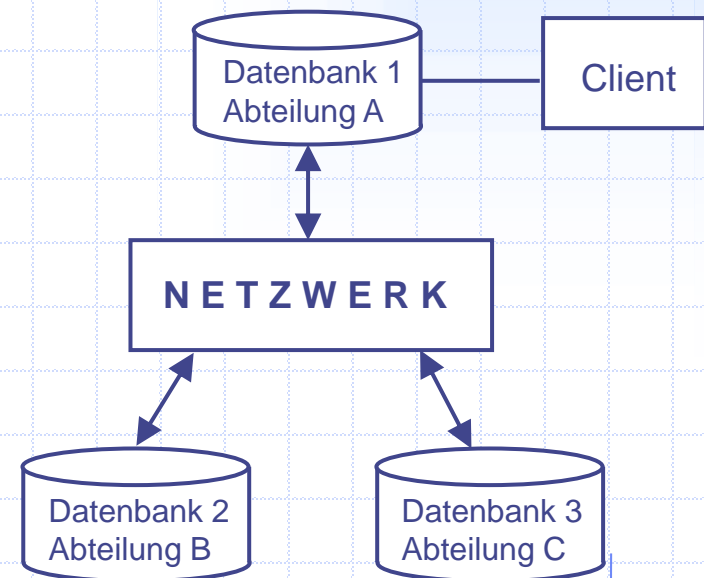
Probleme:

- Netzausfall (keine Verbindung)
- langsames, Teures Netz
- bei redundanter Speicherung: evtl. Widersprüche bei Zusammenführung der Daten

Anwendung:

Unterteilung nach:

- Zuständigkeit,
- thematisch,
- geographisch



Beispiel: geographisch verteilte Datenhaltung

Allgemeine Datenbankeigenschaften (2a)

Geodatenbanken

Räumlicher Index

Räumliche Operatoren

Mehrbenutzerbetrieb / Transaktionskonzept:

Mehrere Benutzer arbeiten gleichzeitig mit einer Datenbank (auch verteilt). Das DBMS hat dabei die Aufgabe, die Benutzer so zu verwalten, dass sie sich nicht gegenseitig behindern (unterschiedliche Benutzergruppen mit unterschiedlichen Benutzerrechten).

Wenn mehrere Benutzer gleichzeitig an einer Datenbank arbeiten, können sich die Aktionen gegenseitig beeinflussen bzw. behindern (mehrere mit Schreibrechten!!!). Nimmt mehr als ein Benutzer Änderungen vor, kann es zu Konflikten kommen. Viele Lösungen sind möglich (abhängig von der Semantik), z. B.:

Allgemeine Datenbankeigenschaften (2b)

Geodatenbanken

Räumlicher Index

Räumliche Operatoren

Optimistischer Ansatz:

Es werden keine Vorkehrungen gegen das Entstehen von Konflikten getroffen. Treten diese auf, sind sie von den Benutzern oder dem DBMS zu beseitigen. Aufdecken und beseitigen durch Benutzer problematisch, daher Ansätze durch DBMS:

- Benachrichtigung: die von einem Konflikt Betroffenen werden benachrichtigt.
- Semantische Konfliktlösung: Automatische Bereinigung durch das System (Wiederherstellen eines konsistenten Zustands). Dazu ist Wissen über die Semantik der Datenbasis notwendig.

Pessimistischer Ansatz:

Lesen für Änderungen nur für einen Benutzer erlaubt.

Allgemeine Datenbankeigenschaften (3)

Replikation

Mehrfaches Vorhalten von Datenbeständen (redundant) - auf die replizierten Daten kann von unterschiedlichen Benutzern gleichzeitig zugegriffen werden.

Ziel:

Verbessern der Performance bei Netzverbindungen mit schmalen Bandbreiten Erhöhung der Verfügbarkeit bei unterbrochener Verbindung (s. o).

Um Probleme bei der Konsistenthaltung / Aktuellhaltung der Datenbasen zu vermeiden werden folgende Methoden benötigt:

- Schreibsperrern
- Transaktionsmechanismen, Unterscheidung:
 - lange Transaktionen
 - kurze Transaktionen
- Konfliktauflösung (vgl. Mehrbenutzerbetrieb)

Allgemeine Datenbankeigenschaften (4)

Transaktionen

Sind zu Paketen zusammengefasste Anweisungen an die Datenbank (z. B. beliebige Änderungsanweisungen). Ein Paket von Anweisungen wird als Transaktion bezeichnet, wenn es die ACID-Eigenschaften besitzt:

A = Atomicity: Transaktionen sind atomar – entweder alle einzelnen Schritte (Anweisungen) werden durchgeführt oder keine.

C = Consistency: Transaktionen bewahren die Konsistenz einer Datenbank, überführen die Datenbank also von einem konsistenten Zustand in einen neuen (konsistenten) Zustand.

I = Isolation: Transaktionen sind voneinander isoliert. D. h. auch wenn viele Transaktionen gleichzeitig ablaufen, sind die Änderungen innerhalb einer Transaktion für andere nicht sichtbar, bis die Transaktion beendet ist.

D = Durability: Ist eine Transaktion beendet, sind die durchgeführten Änderungen dauerhaft. Auch ein nachfolgender Systemabsturz kann die Änderungen nicht mehr rückgängig machen.

Allgemeine Datenbankeigenschaften (6)

Benutzerrechte:

- Einzelne Benutzer oder Gruppen
- Rechte nach thematischen oder räumlichen Kriterien, z. B.:
 - Leserechte, Schreibrechte, Änderungsrechte, Löschrchte
 - Administrationsrechte
 - ...

Versionsmanagement

Version = Inhalt (Zustand) der Datenbank zu einem bestimmten Zeitpunkt

- stichpunktsbezogen / diskret
- kontinuierlich abrufbar (temporale Datenbank)
- Versionen, z.B. für Historienbildung
- Vollversionierung durch Speicherung der gesamten DB zu einem Stichpunkt
- inkrementelle Versionierung durch Speicherung der Änderungen gegenüber einem bestimmten Stichpunkt

Geodatenbanken

Warum spezielle Geodatenbanken?

Überblick Themen:

- Datentypen in Geodatenbanken
- Organisationsformen von Geodatenbanken
- Gegenüberstellung objektorientiert - relational
- Objektrelationales GeoDBMS

Geodatenbanken

- Warum spezielle Geodatenbanken?

Anforderungen an Geodatenbanken:

- Raumbezogene Objekte (komplexe Datentypen)
 - Verwaltung großer Datenmengen mit Raumbezug
 - Abfragen hinsichtlich der Existenz, Position und den Eigenschaften / Beziehungen von raumbezogenen Objekten (interaktiv)
- ⇒ Interaktivität erfordert spez. Zugriffsstrukturen (räumliche Indizierung)

Eine Geodatenbank ...

... ist eine Datenbank, deren Datenbasis neben den Standard- Datentypen (Zeichen, Zahlen usw.) auch Objekte mit Raumbezug verwalten kann. Der Zugriff auf die Daten und die Verwaltung der gesamten Datenbasis (z.B. auch Indexdaten und Metadaten) erfolgt über ein Geo-Datenbankmanagementsystem.

Ein Geo-Datenbank-Management-System (GeoDBMS) ...

... ist ein vollständiges DBMS mit zusätzlichen Möglichkeiten zur Verwaltung (Repräsentation, Abfrage, Manipulation + weitere Operatoren) von Objekten mit Raumbezug.

Datentypen in Geodatenbanken

Ein Datentyp legt eine Menge von Werten und eine Menge von darauf zugeschnittenen Operationen fest.

INTEGER	
Wertebereich (Ganze Zahlen)	von -32768 bis 32768
Operationen	Addition, Subtraktion, Multiplikation, Division, ...

Beispiel: Datentyp Integer

Unterscheidung zwischen:

Basisdatentyp:
im System enthalten, z.B. Integer

benutzerdefinierten Datentyp:
durch Benutzer festgelegt



nichtraumbezogene Datentypen:
Datentypen, die keine raumbezogenen Eigenschaften eines Objekts definieren, z.B. Integer, Character ...
oder komplexe Datentypen, wie Foto, Ton, Video ...

raumbezogene Datentypen:
definieren die räumlichen Eigenschaften eines Objekts (geometrische und topologische Eigenschaften).



Organisationsformen von Geodatenbanken

Relationales GeoDBMS

Alle Entitäten werden in Tabellen bzw. Relationen gehalten. Anfragen an die Datenbank über SQL.

Zeile = „Objekt“

Spalte = Attribut

Duales GeoDBMS

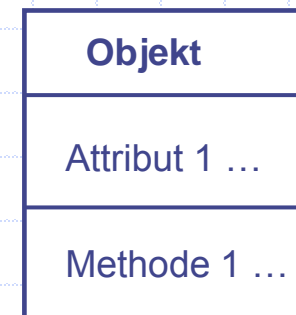
Es gibt zwei Datenbasen:

- Nicht räumliche Entitäten werden in Tabellen bzw. Relationen gehalten (relationales Datenmodell)
- Raumbezogene Entitäten in proprietären Strukturen
- Verknüpfungsmöglichkeiten / Objektbildung durch spezielle SW-Funktionen



Objektorientiertes GeoDBMS

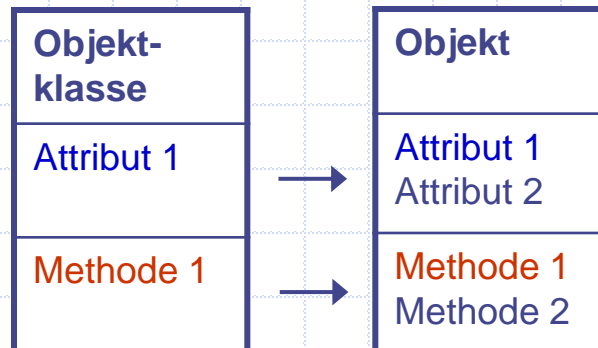
- Abstrakte und vom Benutzer definierbare Datentypen
- Objektbildung, - Klassenbildung, gekoppelt mit Vererbung von elementaren und strukturierten Attributen
- Kapselung von Objekteigenschaften, -bedingungen, -operationen
- Anfragen an die Datenbank über objektorientierte query language



Gegenüberstellung: objektorientiert - relational

	objektorientiert	relational
Grundelemente	Klassen, Objekte, Attribute, Methoden	Tabellen (Relationen), Zeilen, Spalten
Datentypen	komplexe Datentypen	einfache Datentypen
Eigenschaften und Begriffe	Object Query Language (OQL), Kapselung, Vererbung, Identität, Polymorphismus, Aggregation	Strukturierte Anfragesprache (z.B. SQL), Abfrageverarbeitung und -optimierung, Primärschlüssel, Fremdschlüssel

Allg. DB-Konzepte!



Vererbung: Objekte einer Unterklasse erben Eigenschaften und Methoden der Oberklasse

STUDENT				
Nummer	PLZ
111	80686
112	80141
113	10899
114	10561

Primärschlüssel

Fremdschlüssel

ORT		
PLZ	Ort	...
80686	München	...
80141	München	...
10899	Berlin	...
10561	Berlin	...

Beziehungen
zwischen relationalen Tabellen

Objektrelationales GeoDBMS

Geodatenbanken

Räumlicher Index

Konsequente Erweiterung der relationalen GeoDBMS:

- Erweiterung der relationalen Datenbanken um objektorientierte Eigenschaften
- Komplexe Datentypen, benutzerdefiniert, in einer Spalte einer Tabelle:
 - Objekte mit eigenen Attributen und Methoden
 - erweiterte Anfragesprache
 - Methoden zur effizienten Speicherung und Indizierung dieser Datentypen

ID	Nutzung	Fläche	Geometrie	...
88	Acker	17593,81	Polygon	...
...

Sachdaten und Geometriedaten als Attribut! Polygon -> komplexer Datentyp

Indexstrukturen in Datenbanken

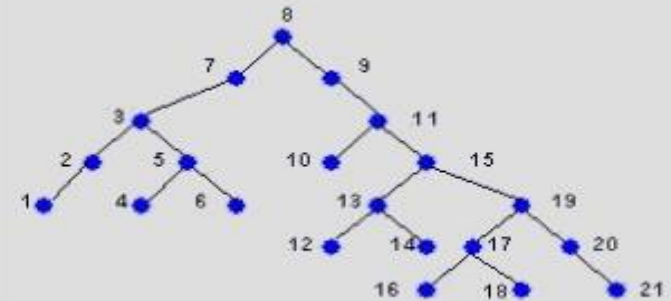
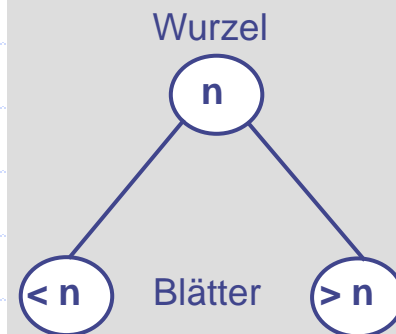
Um Anfragen an die Datenbank zu beschleunigen, verwendet man Indexe (gespeicherte, zusätzliche Informationen über die Daten in der Datenbasis).

...	Kunden-Nr.
...	92
...	1215
...	10301

Sortierung
nach Größe

Index Beispiel:
Standarddatentyp Integer

Bei Anfragen mit sehr großen Datenmengen werden als Standard-Indexstrukturen Binär-Bäume verwendet.



Standarddatentypen lassen sich linear sortieren. Für Komplexe Datentypen (z.B. Geometrien) sind Anfragen wie “In welchem Polygon liegt Punkt P” zu lösen. Für mehrdimensionale Daten wurden deshalb eigene Indexstrukturen entwickelt.

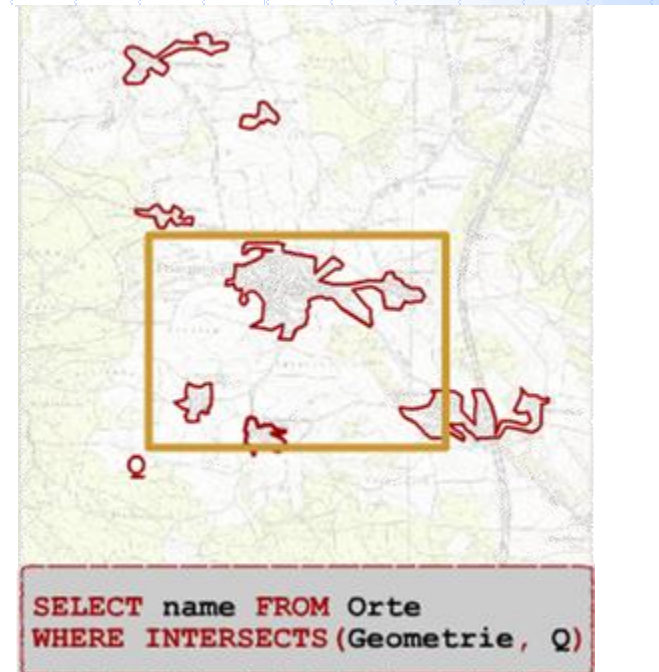
Räumliche Anfragen

Gegeben:

Menge von flächenhaften
Geoobjekten
z.B.: Orte (ID, Name, Einwohner,
Geometrie)

Gewünschte Anfragetypen:

- *Punktanfrage*: In welchem Ort (Polygon) liegt der Punkt P?
- *Bereichsanfrage*: Welche Orte (Polygone) liegen innerhalb des achsenparallelen Rechtecks Q?



Animation: 3x klicken

Motivation

Naive Lösung:

Sequentieller Test aller Polygone

Probleme:

1. Die naive Berechnung ist sehr aufwendig
2. Suchzeit linear von der Anzahl der Polygone abhängig - Nicht vertretbar für große Mengen!

Lösungsansätze:

- zu 1): Filterschritt: Zuerst das minimal umschließende, achsenparallele Rechteck (minimum Bounding Box = BB) testen
- zu 2): Spezielle Indexstrukturen für BBs aufbauen



Animation: 3x klicken

Indexstrukturen - Einführung

Räumliche Indexstrukturen beruhen auf einer (geographischen) Einschränkung des Suchraums bzw. einer Unterstützung der Suche durch Baumstrukturen

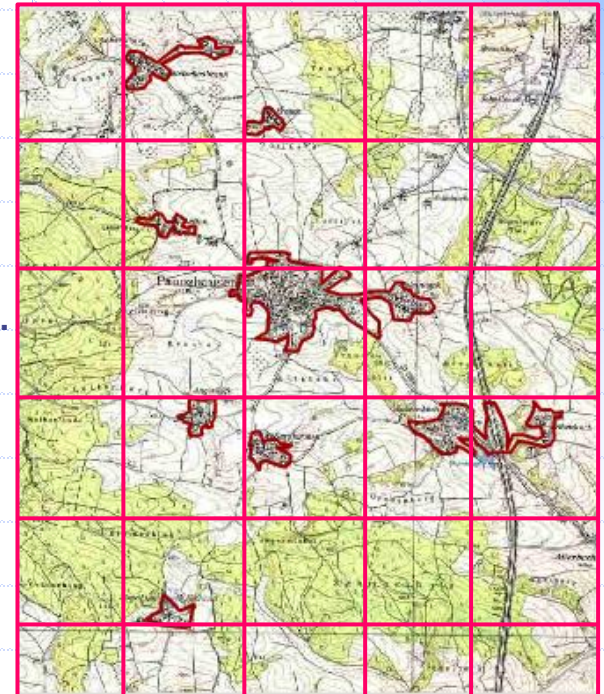
Eine einfache, denkbare Lösung wäre eine Aufteilung des Raumes in gleich große Bereiche (s. Skizze) mit einer entsprechenden Verwaltung.

Probleme:

1. Die Datendichte ist bei Geodaten i.d.R. nicht homogen
2. Zuordnung von Objekten zu einem Bereich mit Standardverfahren nicht möglich (s. Skizze)

Viele Lösungsansätze vorhanden:

- Ein Beispiel: R-Baum (R-Tree)
- ...



Ein Beispiel: R-Baum

Definition:

dynamische Zugriffsmethode oder Indexstruktur in der Datenhaltung für mehrdimensionale Räume

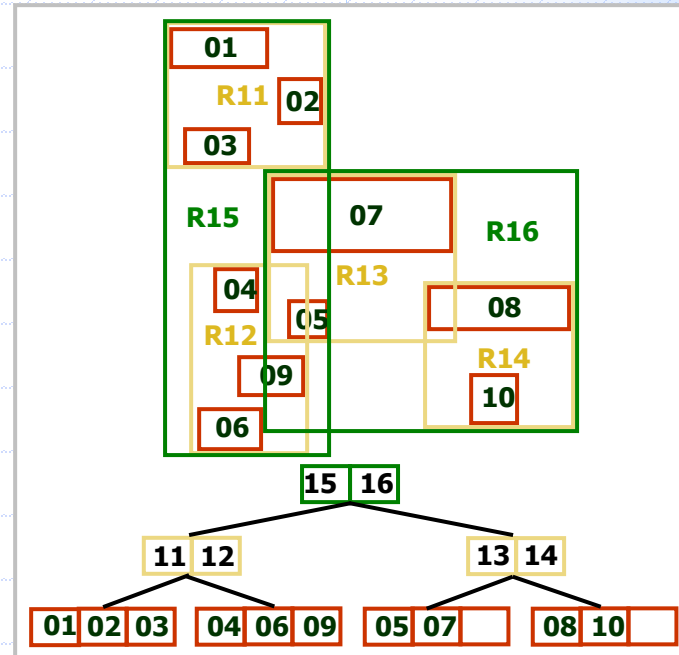
Struktur entspricht einem balancierten Baum:

Jede Baumebene gruppiert und aggregiert die Elemente der nachfolgenden Ebene.

In jedem Blatt werden Referenzen auf die eigentlichen Geometrien, sowie deren Bounding Boxes gespeichert. Jeder Vaterknoten speichert für jeden seiner Nachfolger wieder die Bounding Box über alle seine Teilgeometrien.

Beispiel: innere Knoten zeigen auf einen Teilbaum, welcher von einer BB begrenzt wird.

Weiteres: Literatur



Legende:

Animation: 3x klicken

- 1 Innere Knoten
- 5 Blattknoten
- 1 Blattknoten
- 0 Blattknoten
- 1



Vielen Dank für die Aufmerksamkeit!

Weitere Fragen?

Kontakt

Keine festen Sprechstunden, einfach vorbeikommen, oder besser vorher anrufen, bzw. noch besser: email schreiben

**Univ.-Prof. Dr.-Ing. Wolfgang Reinhardt
AGIS / Inst. Für Angewandte Informatik (INF4)
Universität der Bundeswehr München
D-85577 Neubiberg
Telefon +49 (0)89 6004-2450
Telefax +49 (0)89 6004-3906**

Wolfgang.Reinhardt@unibw.de

www.agis.unibw.de

Geb 37/200 Raum 0208