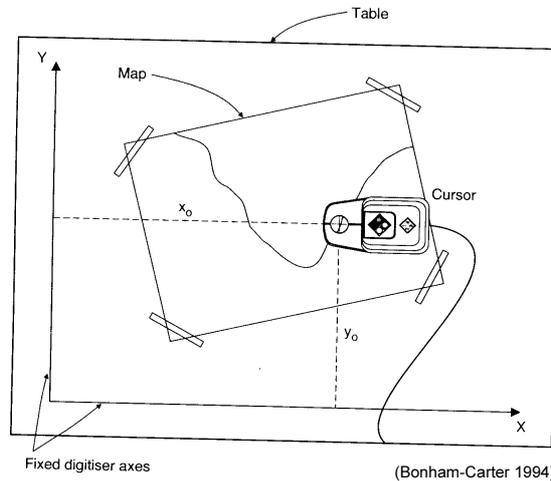


(Vektor-)Datenhaltung I (relational)

Herkunft von Vektordaten

- Vermessung
- Digitalisierung analoger Karten
- Verschneidung vorhandener digitaler Karten

Digitalisierung

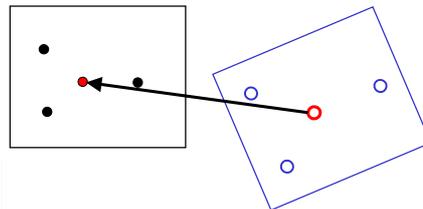


Digitalisierung: Passpunkte

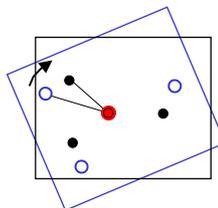
= Punkte, deren Positionen (jeweils in lokalen Koordinaten) auf beiden Karten bekannt sind

Abgleich:

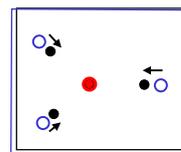
1. Angleichung der Schwerpunkte



2. Rotation

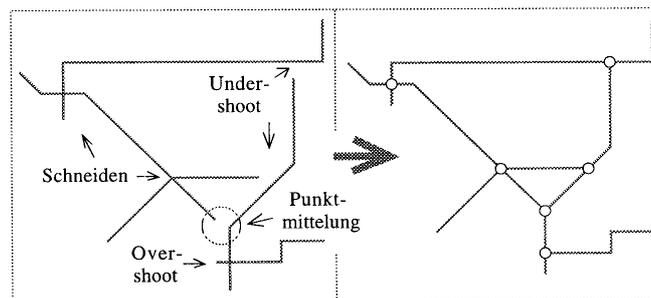


3. Abgleich der Entfernungen



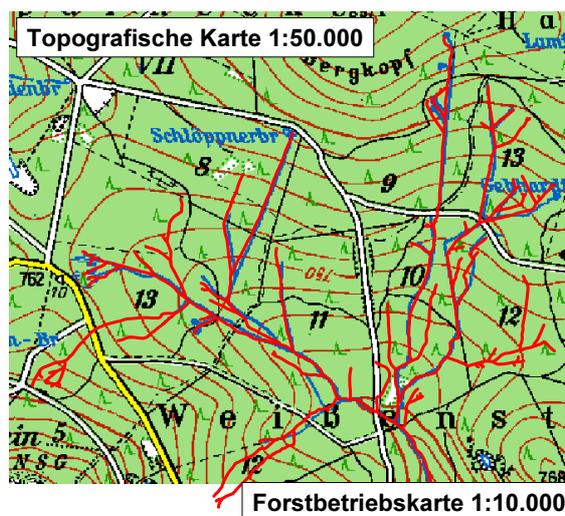
Digitalisierung

- Spaghetti: topologisch nicht strukturierte Linien
- Pizza: topologisch nicht strukturierte Flächen



(Bartelme 2000)

Verschneidung von Karten



E(A)R-Konzept

Drei Arten von Bestandteilen von Geodatenätzen:

1. Entität:

- = eindeutig identifizierbares Ganzes
- Entitäten (= **Instanzen**) können zu Entitätsklassen (= **Typen**) mit best. Eigenschaften zusammengefasst werden, z.B.: Schule, Garage, Hütte
→ Gebäude

2. Relation (Beziehung):

- zwischen Entitäten bzw. Entitätsklassen
- **Grad** einer Beziehung: Anzahl der "Beziehungspartner"
- **Kardinalität** einer Beziehung (Eindeutigkeit), z.B. *1:1*, *1:m*, *m:n*

3. Attribut (Eigenschaft):

- für Entitäten, z.B. Größe, Hausnummer, Einwohnerzahl
- für Relationen, z.B. "liegt in", "ist Hauptstadt von", "grenzt an"

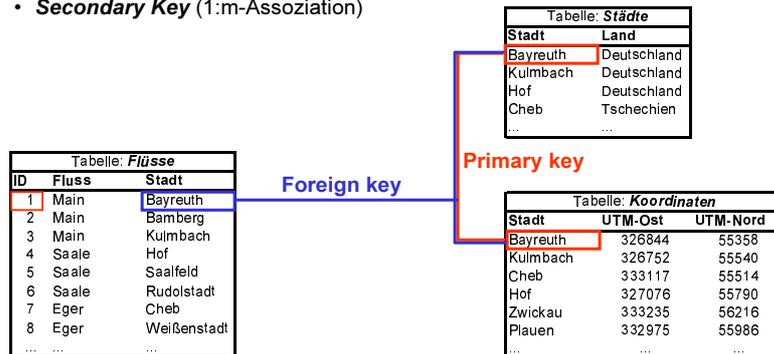
Relationale Datenbank (RDB)

- **Verknüpfungen** (*relations*) zwischen **Objekten** (*entities*) und ihren **Eigenschaften** (*attributes*)
- Speicherung der Daten in Tabellen (*relations*):
 - Zeilen (*Tupel*): Einzelobjekte (Entities)
 - Spalten (*Domänen*): Eigenschaften (Attribute);
Anzahl der Spalten = Grad der Relation
 - Einzeleinträge (Elemente) = Segmente

Stadt	UTM-Ost	UTM-Nord
Bayreuth	326844	55358
Kulmbach	326752	55540
Cheb	333117	55514
Hof	327076	55790
Zwickau	333235	56216
Plauen	332975	55986
...

Relationale Datenbank (RDB)

- **Verknüpfungen** zwischen verschiedenen Tabellen anhand von **Schlüsseln** (= **Keys**)
- **Key** = Eigenschaft (Segment), die eine eindeutige Identifizierung des Objektes ermöglicht
 - **Primary Key** (m:1-Assoziation)
 - **Foreign Key** (Segment dient als Primary Key für anderen Entitätstyp)
 - **Secondary Key** (1:m-Assoziation)



Eigenschaften

- Organisation in Tabellen (entsprechend der intuitiven Organisation von Daten)
- Atomare Struktur: jeweils maximal 1 Eintrag pro Segment
- Minimale Redundanz der Daten (z.B. keine doppelten Tupel)
- Willkürliche Anordnung der Spalten und Zeilen
- Interne Organisation einer Tabelle unabhängig von der anderer Tabellen
- Formulierung von Abfragen unabhängig von der Art, wie die Abfragen intern behandelt werden (*nichtprozedurale Kommunikation*)
- Tabellen nachträglich leicht erweiterbar (Einfügen zusätzlicher Zeilen oder Spalten)
- Möglichkeit der Erstellung benutzerspezifischer Ansichten (*views*)

Eigenschaften

- Strukturierung der Daten durch Einteilung in Klassen gleichartiger Entitäten (Städte, Länder, Flüsse,)
- Einfache Regeln (= Normalformen) ermöglichen eine minimale (redundanzfreie) Datenspeicherung und stabile Datenstruktur
- Vereinfachung der Wartung und Konsistenzerhaltung durch Integritätsbedingungen => stabile Datenbankstruktur (= *Normalform*):
 - Veränderungen der Daten ziehen nicht unerwünschte Nebeneffekte und Fernwirkungen nach sich
 - Unvollständige oder unrichtige Abfragen werden toleriert

1. Normalform

Drei Bedingungen:

1. Verknüpfung der Daten über logische Verweise, nicht über physische Adressen (bei RDB automatisch gegeben)
2. Primärschlüssel für jeden Entitätstyp
3. Eindeutige Namen für jedes Segment innerhalb einer Entität => maximal ein Segment der Entität pro Tupel

Gegenbeispiel

Land	Stadt1	Stadt2	Stadt3	...
------	--------	--------	--------	-----

=> würde variable Spaltenzahl erfordern

2. und 3. Normalform

2. Normalform: zusätzlich zu den Bedingungen der 1. Normalform

- Jedes Segment einer Entität hängt funktional voll von einem Schlüssel ab

3. Normalform: zusätzlich zu den Bedingungen der 2. Normalform

- Keine transitive Abhängigkeit eines Segments von einem Schlüssel (*Abhängigkeit eines Segmentes von einem anderen Segment, das wiederum vom Schlüssel abhängt*)

Integritätsbedingungen

Ziel:

- Aufrechterhaltung der Datenkonsistenz
- Ermöglichung automatisierter Abfragen

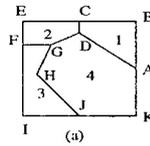
Bsp. für Integritätsbedingungen:

- **Wertebereiche** für Attribute
- **Kardinalität** von Beziehungen
- **topologische Konsistenz**, Bsp.: *Flüsse kreuzen sich nicht, Grundstücke überlappen sich nicht, etc.*
- **Key Constraint:** jeder Eintrag wird anhand eines *primary key* identifiziert
- **Entity Integrity:** kein Eintrag ohne *primary key*
- **Referential Integrity:** *foreign key* einer Tabelle muss identisch sein mit dem *primary key* einer anderen Tabelle

Speichern von Vektordaten

Spaghetti structure

Polygon	Line segment
1	AB, BC, CD, DA
2	CD, CE, EF, FG, GD
3	FG, FI, IJ, JH, HG
4	JK, KA, DA, GD, HG, JH



Arc node structure

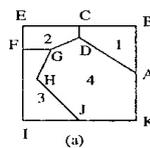
Arc	Start node	End node	Left area	Right area
AB	A	B	1	X
BC	B	C	1	X
CD	C	D	1	2
DA	D	A	1	4
GD	G	D	2	4
		...		

Doubly connected edge list (DECL)

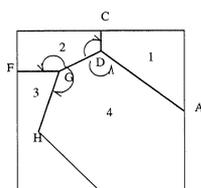
Arc	Start Node	End node	Left area	Right area	Prev. arc	Following arc
AB	A	B	1	X	KA	BC
BC	B	C	1	X	AB	GD
CD	C	D	1	2	BC	DA
DA	D	A	1	4	CD	AB
GD	G	D	2	4	FG	DC
		...				

(Adam und Gangopadhyay 1997)

Speichern von Vektordaten

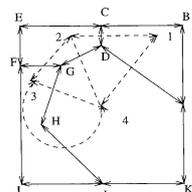


Winged-edge structure



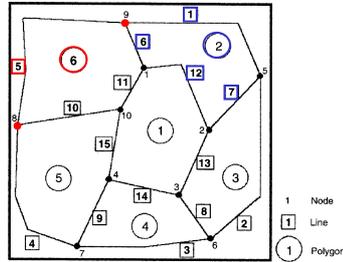
Start Node: G
 End Node: D
 Left face: 2
 Right face: 4
 Edge incident clockwise at G: HG
 Edge incident counterclockwise at G: FG
 Edge incident clockwise at D: DC
 Edge incident counterclockwise at D: DA

Quad-edge structure



(Adam und Gangopadhyay 1997)

Vektordaten: Relationale Datenbank



Node list		
ID	X	Y
1	95	115
2	112	94
3	107	72
4	83	77
5	134	112
6	117	56
7	72	54
8	53	95
9	89	130
10	87	101

Line list				
ID	FROM NODE	TO NODE	LEFT POLYGON	RIGHT POLYGON
1	5	9	2	3
2	5	6	2	3
3	6	7	4	5
4	7	8	5	6
5	8	9	6	7
6	9	10	6	5
7	2	5	2	3
8	3	6	3	4
9	4	7	4	5
10	8	10	6	5
11	1	10	1	6
12	1	2	2	1
13	2	3	3	1
14	3	4	4	1
15	4	10	5	1

Segment list		
Line ID	X	Y Pairs
1	122	130
2	134	72
3	95	84
4	56	52, 71

Polygon list	
ID	Line list
1	11 12 13 14 15
2	1 7 12 6
3	2 6 13 7
4	3 9 14 8
5	4 10 15 9
6	5 8 11 10

(Bramm 2003)