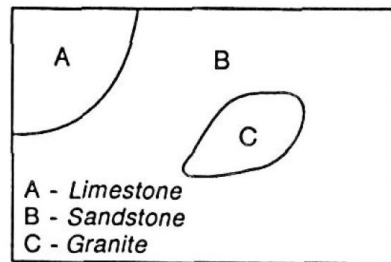


Topologische Datenstrukturen: Darstellung von “Nachbarschafts”beziehungen zwischen Objekten (geometrische Eigenschaften und Distanzen spielen keine grosse Rolle, sondern nur die relative Lage und Beziehung zueinander)

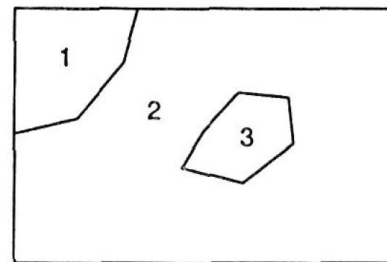
- *Grund-Elementtypen:* Punkte/Vertex, Kante (verbindet zwei Punkte), Fläche (begrenzt durch Kanten), Volumen (begrenzt durch Flächen). Alle Objekte von Dimension $d > 0$ sind begrenzt durch Elemente von Dimension $d-1$
- Grund-Elementtypen erlauben Definition weiterer *topologischer Elemente:*
 - **Linie/Linienzug/Bogen (line/arc):** geordnete, zusammenhängende Menge von Kanten mit einem End- und einem Anfangsknoten
 - **Knoten (Node):** Punkt an dem eine Linie beginnt/endet oder an dem sich zwei oder mehrere Linien treffen
 - **Kette/Bogen (chain/arc):** Linien, die Teil der Grenze eines Flächenobjektes sind
 - **Ring:** Grenze eines Flächenelements, welche aus mehreren Ketten/Bögen besteht
 - **Polygon:** Flächenelement begrenzt durch ein oder mehrere Ringe (mindestens ein äußerer Ring)
 - Einfaches Polygon: nur ein äußerer Ring
 - Komplexes Polygon: ein oder mehrere innere Ringe, zusätzlich zu einem äußeren Ring

Topologische Datenstrukturen: Darstellung von "Nachbarschafts"beziehungen zwischen Objekten (geometrische Eigenschaften und Distanzen spielen keine grosse Rolle, sondern nur die relative Lage und Beziehung zueinander)

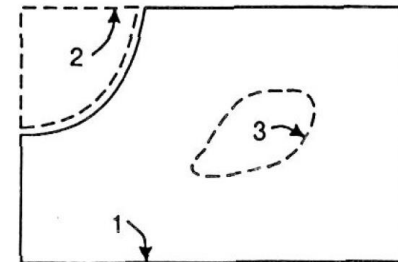
- Grund-Elementtypen (Kanten), Volumen durch Elemente von
- Grund-Elementtypen
 - **Linie/Linienelemente:** Kanten mit einer Richtung
 - **Knoten (Nodes):** mehrere Linien treffen
 - **Kette/Bogen:** mehrere Knoten
 - **Ring:** Grenze eines Objektes
 - **Polygon:** Fläche
 - Einfach
 - Komplex



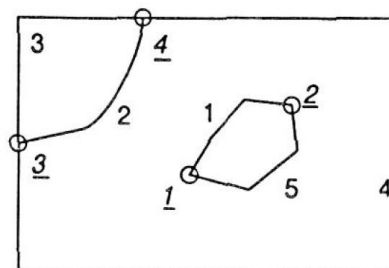
A) Map



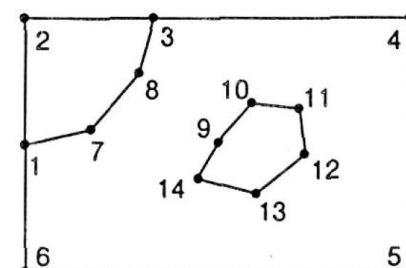
B) Polygons



C) Rings



D) Chains and nodes

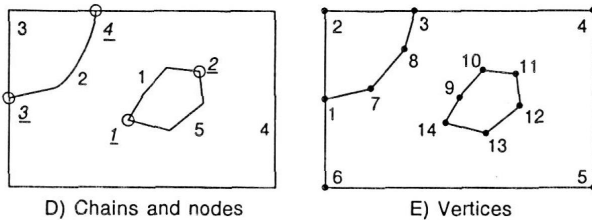
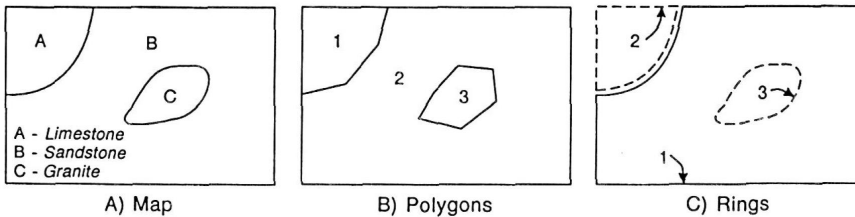


E) Vertices

Darstellung der topologischen Struktur mittels Listen/Tabellen (z.B. van-Roessel-Topologie):

- Polygon Topologie Tabelle
- Ring Topologie Tabelle
- Ketten Topologie Tabelle
- Knoten-zu-Punkt Tabelle
- Kette-zu-Punkt Tabelle
- Koordinatentabelle

Darstellung der topologischen Struktur mittels Listen/Tabellen (z.B. van-Roessel-Topologie):



A. Polygon topology table

Polygon #	Ring #	Ring Sequence #
1	2	1
2	1	1
2	3	2
3	3	1

B. Ring topology table

Ring #	Chain #	Chain Sequence #
2	3	1
2	2	2
1	2	1
1	4	2
3	1	1
3	5	2

C. Chain topology table

Chain #	Start Node	Stop Node
1	1	2
2	3	4
3	4	3
4	4	3
5	1	2

F. Coordinates of vertices table (part only)

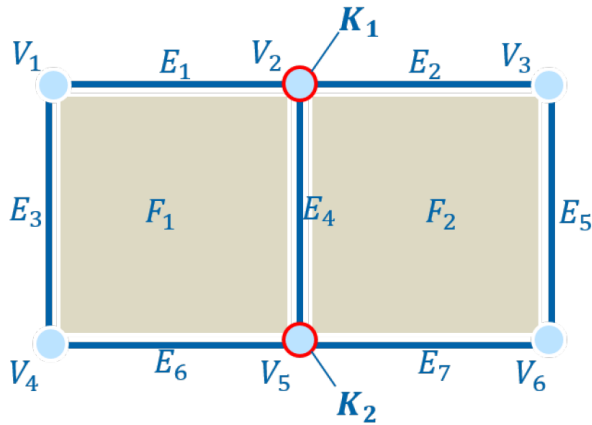
Vertex #	X	Y
1	x_1	y_1
2	x_2	y_2
.	.	.
14	x_{14}	y_{14}

D. Node-to-vertex table

Node #	Vertex #
1	14
2	11
3	1
4	3

E. Chain-to-vertex table

Chain #	Vertex #	Vertex Sequence #
1	14	1
1	9	2
1	10	3
1	11	4
2	1	1
2	7	2
2	8	3
2	3	4
3	3	1
3	2	2
3	1	3
4	3	1
4	4	2
4	5	3
4	6	4
4	1	5
5	14	1
5	13	2
5	12	3
5	11	4



Linien/Ketten:

$$L_1 = \{E_1, E_3, E_6\}$$

$$L_2 = \{E_4\}$$

$$L_3 = \{E_2, E_5, E_7\}$$

Ringe:

$$R_1 = \{L_1, L_2\}$$

$$R_2 = \{L_2, L_3\}$$

Inzidenz- und Adjazenzmetrizen als Möglichkeit zur mathematischen Beschreibung der Nachbarschaftsbeziehungen

$$I_{VE} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad A_V = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Vektormodell

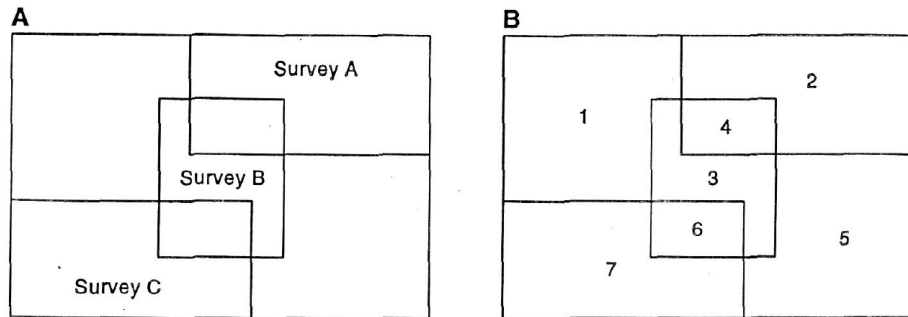
Für welche (beispielhaften) Abfragen könnte solch ein topologische Struktur sinnvoll/hilfreich sein?

- Finde alle Grenzschichten zwischen Granit und Kalkstein
- Entferne alle Kanten zwischen Polygonen mit gleichen Attributeigenschaften
- Finde alle Punkte an denen sich mehr als zwei Kontinentalplatten treffen

Vektormodell

Ist jede Realisierung eines Vektormodells für solche Abfragen sinnvoll?

Es sollte nach Möglichkeit keine sich überlappenden Polygone geben. Erreichbar z.B. durch **Planar Enforcement** (oder allgemeiner, planare Vermaschungen)



POLYGON	SURVEY			
	NONE	A	B	C
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	1	1	0
5	1	0	0	0
6	0	0	1	1
7	0	0	0	1

Abschliessender Vergleich:

- Rastermodell sehr gut geeignet zur Überlagerung von Layern und zur Bildverarbeitung (aber oft speicherintensiv)
- Vektormodell mit Sphagettistruktur: gut geeignet und effizient zur Visualisierung von Grenzen, z.B. in der Kartographie
- Vektormodell mit topologischer Struktur: sehr gut geeignet für räumliche Abfragen und Verbindungsinformationen (aber oft aufwendig anzulegen)

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten gleichen Typs weisen den gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
 - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
 - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
 - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

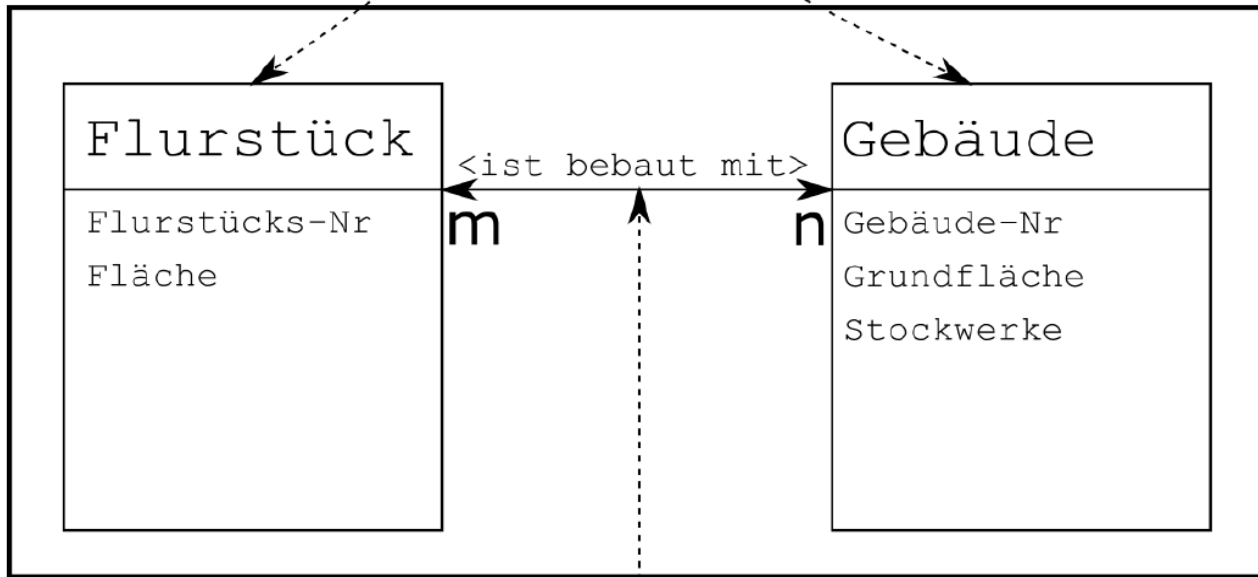
Entitäten-Relationen-Modell:

Entitätsmengen

nt wird

vom Typ B

den



Relation

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten gleichen Typs weisen den gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
 - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
 - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
 - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Graphen-basierte Modelle:

- Beziehungen zwischen Entitäten werden in gerichteter Form dargestellt
- hierarchische und vernetzte Darstellung möglich

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten Gleichen Typs weisen den Gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
 - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
 - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
 - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Graphen-basierte Modelle:

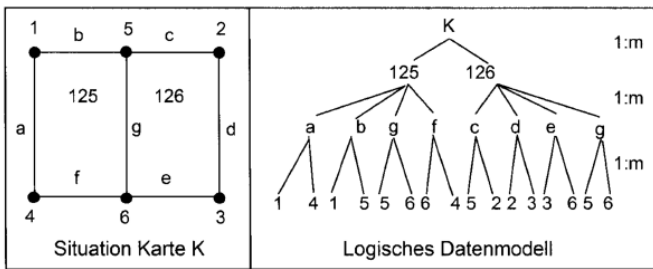


Abbildung 6.43: Hierarchisches Datenmodell.

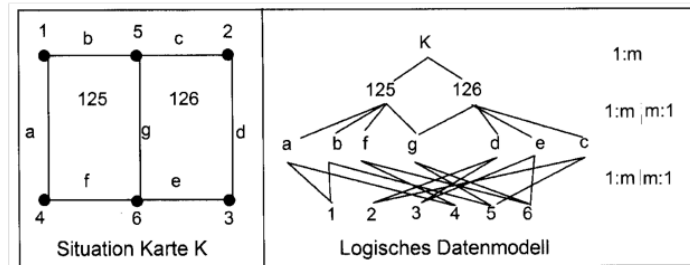
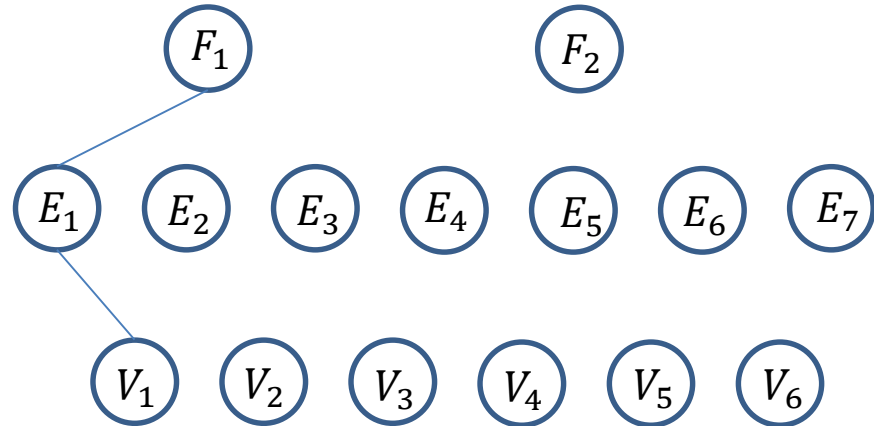
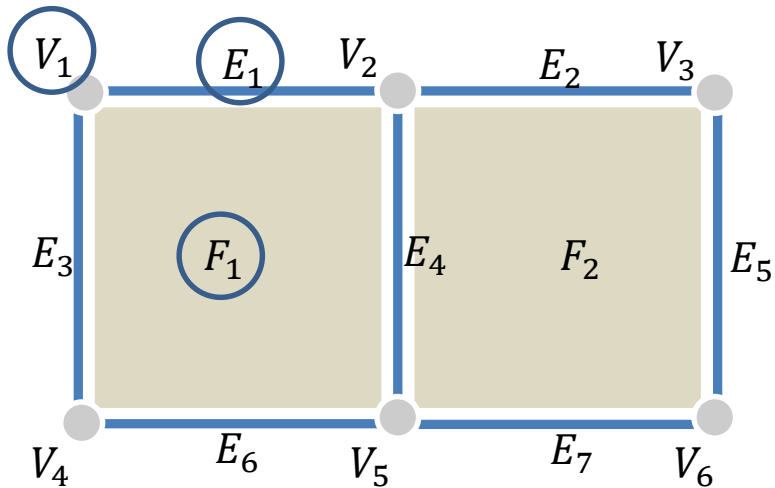


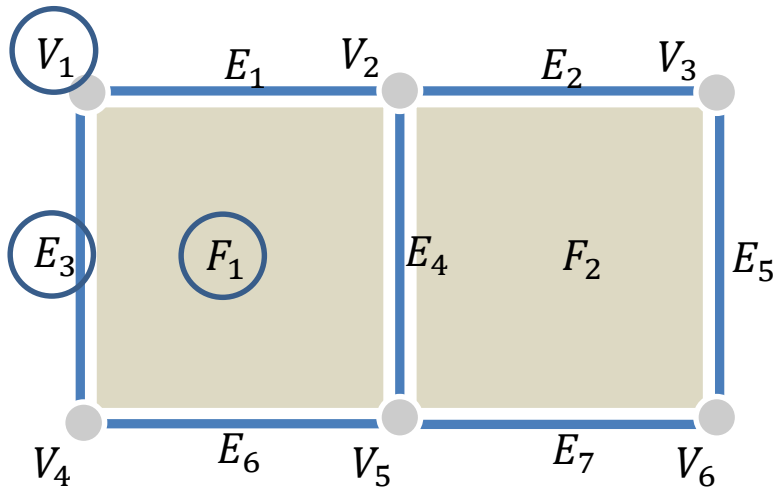
Abbildung 6.44: Netzwerkartiges Datenmodell.

Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell

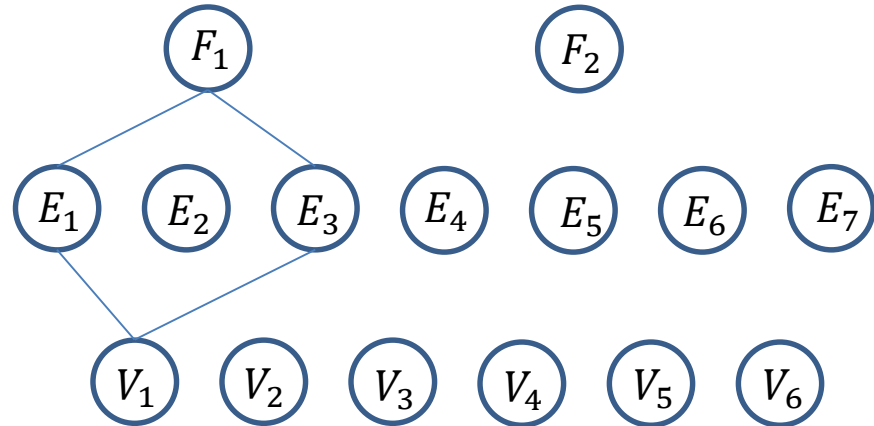


$\{V_1, E_1, F_1\}$

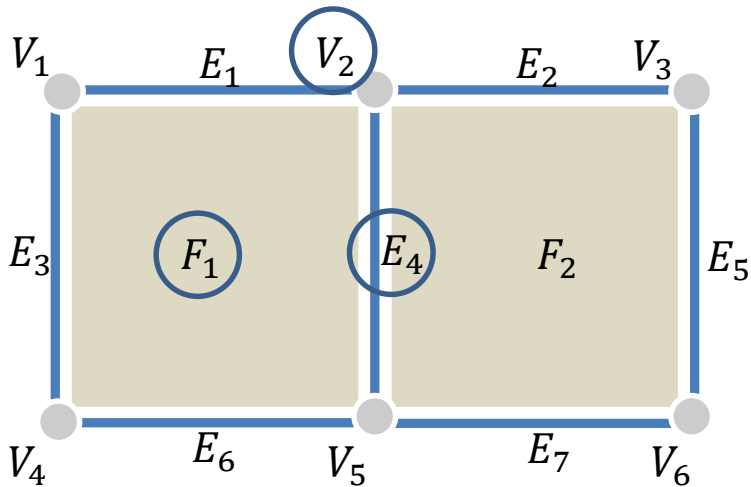
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



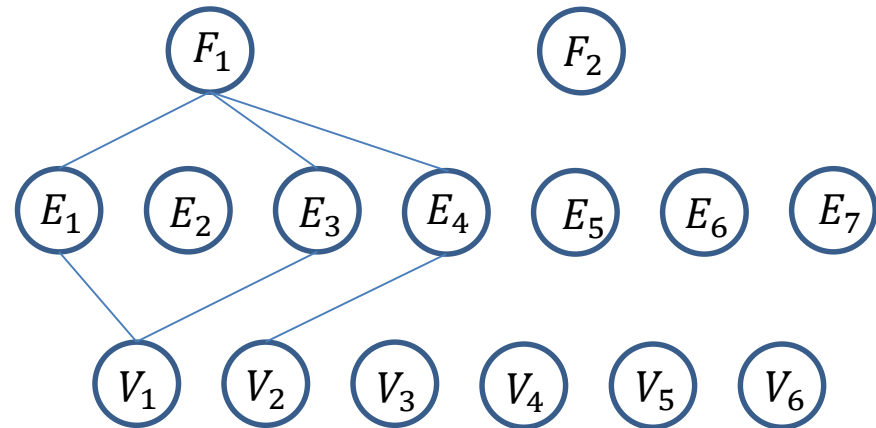
$\{V_1, E_3, F_1\}$



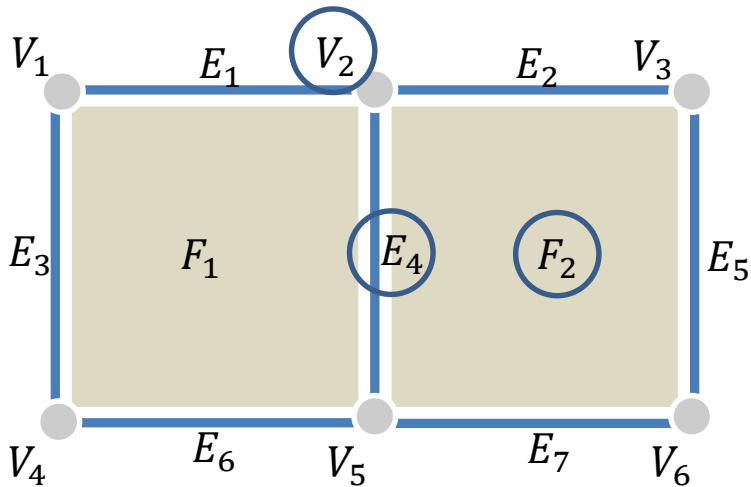
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



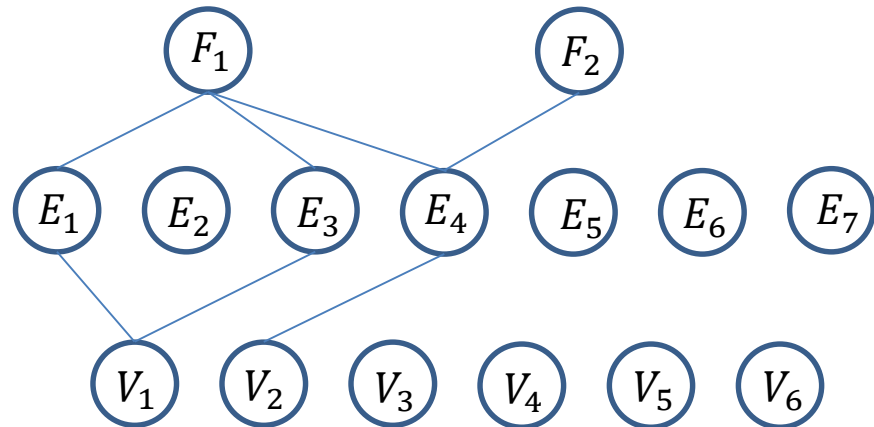
$\{V_2, E_4, F_1\}$



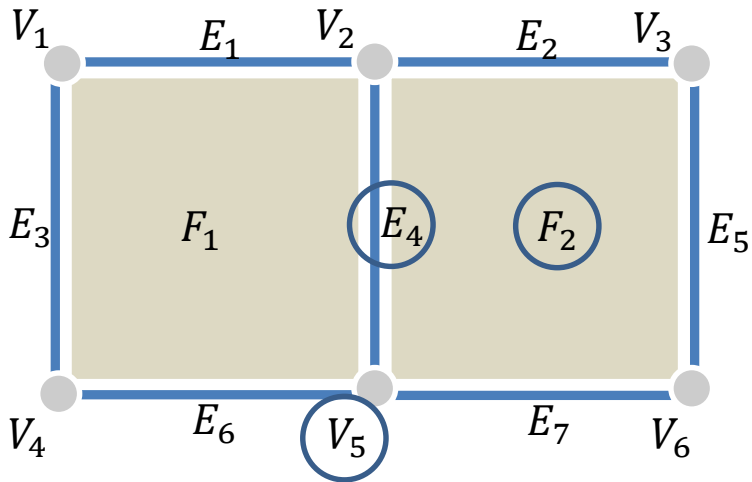
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



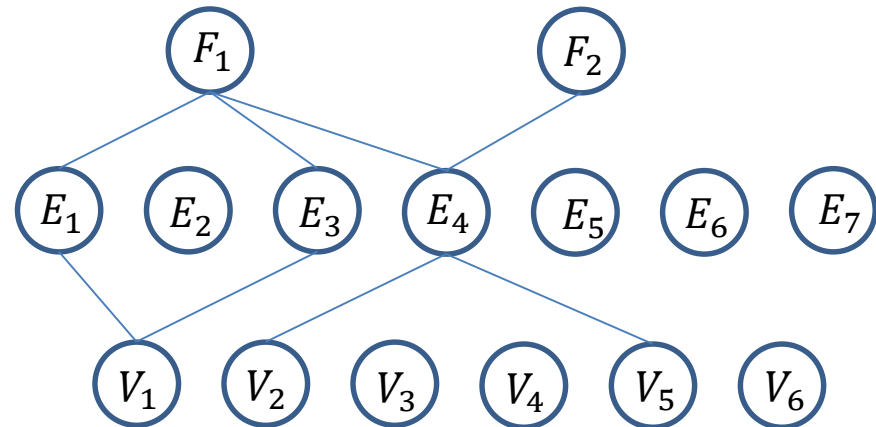
$\{V_2, E_4, F_2\}$



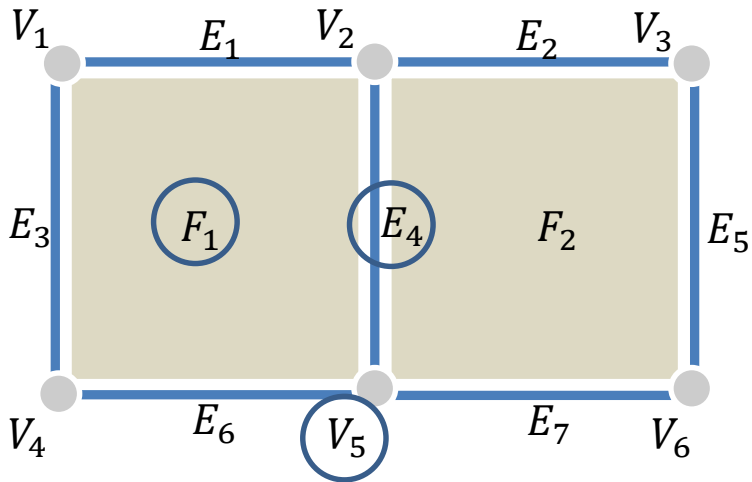
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



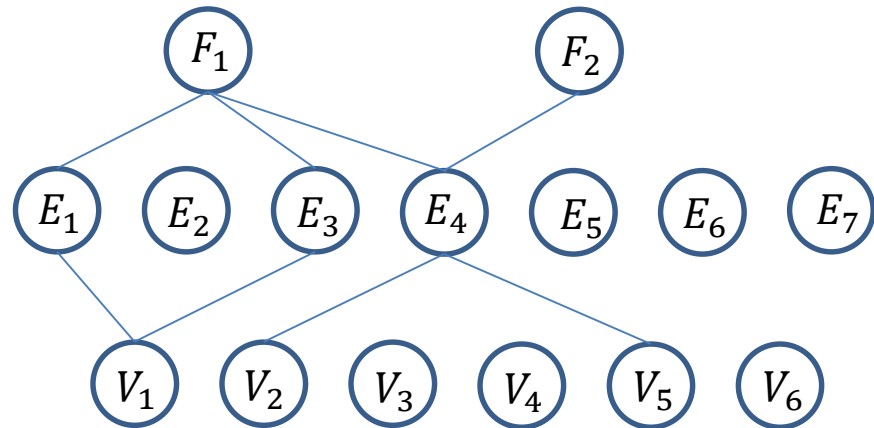
$\{V_5, E_4, F_2\}$



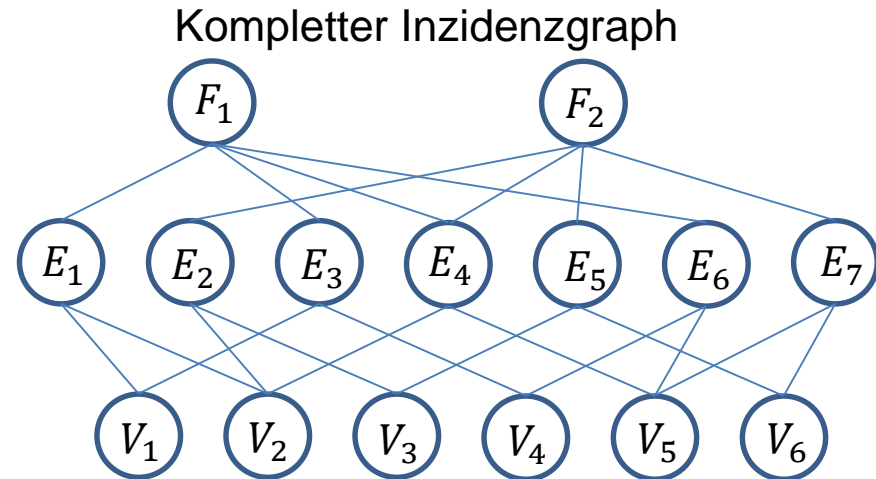
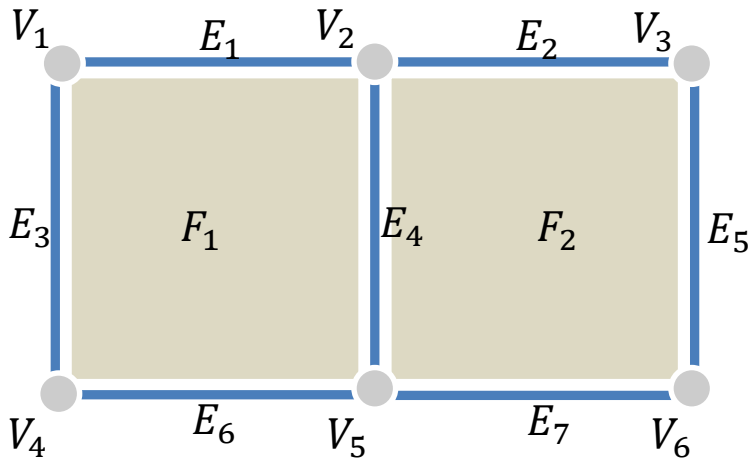
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



$\{V_5, E_4, F_1\}$



Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



⇒ alternatives Datenmodell für topologische Beziehungen

Relationales Datenmodell:

- Entitäten als auch deren Beziehungen werden in Tabellen (sogenannten *Relationen*) hinterlegt
- Jede Zeile der Relation entspricht einem *Tupel*, jede Spalte einem *Attribut*
- Attribute, die zur Identifizierung einzelner Tupel dienen, heissen *Schlüssel-Attribute (keys)*; jedes Tupel einer Relation ist eindeutig durch einen *Primärschlüssel* identifiziert
- *Fremdschlüssel* verweisen auf Tupel einer anderen Relation und können mehrfach auftauchen
- Tupel dürfen sich in einer Relation *nicht wiederholen*. Bedeutung einer Relation darf *nicht von der Reihenfolge* der Attribute oder Tupel *abhängen*.

Relationales Datenmodell:

- Entitäten als auch deren Beziehungen werden in Tabellen (sogenannten *Relationen*) hinterlegt
- Jede Zeile der Relation entspricht einem *Tupel*, jede Spalte einem *Attribut*
- Attribute die zur Identifizierung einzelner Tupel dienen heissen *Schlüssel-Attribute (keys)*;

Polygone				
Polygon #	Formation	Lithologie	Alter	Zeitspanne
1	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ
2	Grit Fm.	Sandstein	Oberkarbon	323.2 – 298.9 MJ
3	Slab Fm.	Schiefer	Oberkarbon	323.2 – 298.9 MJ
4	Mount Fm.	Granit	Kreide	145 – 66 MJ
5	Mount Fm.	Granit	Kreide	145 – 66 MJ
6	Volcano Fm.	Tuff	Trias	251.9 – 201.3 MJ
7	Mount Fm.	Granit	Kreide	145 – 66 MJ
8	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ
9	Slab Fm.	Schiefer	Oberkarbon	323.2 – 298.9 MJ
10	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ

hrfach
f nicht

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- Numerische Codes für textbasierte Attribute einführen

Polygon #	Formation #	Formation	Polygone		Alter #	Alter	Zeitspanne
			Lithologie #	Lithologie			
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 – 298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 – 298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 – 201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 – 298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- Numerische Codes für textbasierte Attribute einführen
- **1. Normalform:**
 - alle Attributwerte müssen *atomar* sein, d.h. sie bestehen nur aus einem einzigen Wert
 - die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)

Polygone								
Polygon #	Formation #	Formation	Lithologie #	Lithologie	Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 MJ	298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 MJ	201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- Numerische Codes für textbasierte Attribute einführen
- **1. Normalform:**
 - alle Attributwerte müssen *atomar* sein, d.h. sie bestehen nur aus einem einzigen Wert
 - die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)
- **2. Normalform:**
 - die Relation liegt bereits in 1. Normalform vor
 - jedes Nicht-Schlüsselattribut muss von *allen* Primärschlüsseln abhängen (voll funktional abhängig)

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- **3. Normalform:**
 - die Relation liegt bereits in 2. Normalform vor
 - Alle Nicht-Schlüsselattribut sind untereinander unabhängig (d.h. sie müssen *nicht-transitiv* bzgl. dem Primärschlüssel sein)



Polygone								
Polygon #	Formation #	Formation	Lithologie #	Lithologie	Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 MJ	298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 MJ	201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Formation ist nicht-transitiv von *Polygon* abhängig, aber *Lithologie* nicht \Rightarrow Relation aufteilen

Weitere Datenmodelle

Polygone	
Polygon #	Formation #
1	2
2	3
3	4
4	1
5	1
6	5
7	1
8	2
9	4
8	2

Formation			
Formation #	Formation	Lithologie #	Alter #
1	Mount Fm.	2	8
2	Shelly Fm.	7	5
3	Grit Fm.	6	5
4	Slab Fm.	5	5
5	Volcano Fm.	3	7

Lithologie	
Lithologie #	Lithologie
2	Granit
3	Tuff
5	Schiefer
6	Sandstein
7	Kalkstein

Alter			
Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
5	Oberkarbon	323.2 MJ	298.9 MJ
7	Trias	251.9 MJ	201.3 MJ
8	Kreide	145 MJ	66 MJ

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- **3. Normalform:**
 - die Relation liegt bereits in 2. Normalform vor
 - Alle Nicht-Schlüsselattribut sind untereinander unabhängig (d.h. sie müssen *nicht-transitiv* bzgl. dem Primärschlüssel sein)
- Weiter Normalformen können hinzugefügt werden, aber die 3. Normalform reicht aus, um sinnvoll Operationen der relationalen Algebra wie **Vereinigungen**, **Differenzen**, **Durchschnitte**, **karthetische Produkte**, **Projektionen**, **Selektion** von Relationen zu definieren.
- Änderungen und Ergänzungen können ebenfalls einfach vorgenommen werden

Weitere Datenmodelle

Table 2-6. Editing the relation shown in Table 2-4 B before normalization. It has been discovered that the Mount Fm. is a gabbro not a granite (!). Even in this very trivial example it can be seen that the change must be repeated exactly the same in three places in the unnormalized table. Change lithology # and lithology each time the Mount Fm. occurs, see items in bold italics.

Polygon #	Formation #	Formation Name	Lithology #	Lithology	Age #	Age
1	2	Shelly Fm.	7	Limestone	5	Pennsylvanian
2	3	Grit Fm.	6	Sandstone	5	Pennsylvanian
3	4	Slab Fm.	5	Shale	5	Pennsylvanian
4	<u>1</u>	Mount Fm.	<u>4</u>	<u>Gabbro</u>	8	Cretaceous
5	<u>1</u>	Mount Fm.	<u>4</u>	<u>Gabbro</u>	8	Cretaceous
6	5	Volcano Fm.	3	Tuff	7	Triassic
7	<u>1</u>	Mount Fm.	<u>4</u>	<u>Gabbro</u>	8	Cretaceous
8	2	Shelly Fm.	7	Limestone	5	Pennsylvanian
9	4	Slab Fm.	5	Shale	1	Pennsylvanian
10	2	Shelly Fm.	7	Limestone	5	Pennsylvanian

Table 2-7. After normalization, the POLYGON and AGE relations remain unchanged, and two small changes are needed in the FORMATION and LITHOLOGY relations. Suppose this change is made in a real situation for a geological map containing several thousand polygons. The editing changes would still be the same as shown here, because the POLYGON relation requires no modification.

FORMATION relation (1 change shown in italics)

Formation #	Formation Name	Lithology #	Age #
1	Mount Fm.	<u>4</u>	8
2	Shelly Fm.	7	5
3	Grit Fm.	6	5
4	Slab Fm.	5	5
5	Volcano Fm.	3	7

LITHOLOGY relation
(1 insertion shown in italics)

Lithology #	Lithology
2	Granite
3	Tuff
<u>4</u>	<u>Gabbro</u>
5	Shale
6	Sandstone
7	Limestone