

Interdisziplinäre Anwendungen Raumbezogener Informationstechnik

Organisatorisches

- Alle Zeitangaben unter Vorbehalt -

Organisatorisches

- Kennenlernen
 - Martin Unold
 - Wissenschaftlicher Mitarbeiter am i3mainz
 - Büro in Raum C0.04
 - E-Mail: martin.unold@hs-mainz.de
 - Prof. Kai-Christian Bruhn
 - Im November
- Unterlagen
 - <http://unold.net/idarit>

Organisatorisches

- Modulziele – Was lernen Sie hier?
 - In interdisziplinärer Zusammenarbeit Aufgabenstellungen und Lösungsmöglichkeiten zur Dokumentation raum- und zeitbezogener Informationen aus den Geisteswissenschaften zu formulieren und entsprechende Projekte zu planen
 - Geoinformationssysteme für die genannten Fachgebiete selbst anzuwenden
 - Anforderungen an digitale Bestände von Forschungsdaten hinsichtlich standardisierter Datenhaltung und Langzeitverfügbarkeit zu verstehen und umzusetzen

Organisatorisches

- **Modulinhalte – Was sind die Themen?**
 - Bedeutung raumbezogener Information in ausgewählten geisteswissenschaftlichen Fachgebieten
 - Zeit- und raumbezogene Problemstellungen
 - Modellierung für die informationstechnische Bearbeitung
 - Design von Datenbanken und GIS
 - Dateneingabe in ein GIS
 - Analysefunktionen eines GIS
 - Präsentationsmöglichkeiten eines GIS
 - Austausch von Datenbeständen auf Grundlage standardisierter Formate
 - Beispielprojekt

Organisatorisches

- Termine

- 2 SWS Vorlesung – Do, 14 - 16 Uhr (B1.09)
- 2 SWS Übung – Do, 16 - 18 Uhr (C0.06)
- 1 SWS Seminar – Blockveranstaltung

- 5.10., 12.10., 19.10.

Einführung in Semantic Web bei Martin Unold

- 26.10.

Klausur zum Thema Semantic Web + Vergabe Web-Projekt

- Ab 2.11.

GIS-Projekt bei Prof. Bruhn

Organisatorisches

- Termine (werden noch genauer spezifiziert)
 - Blockveranstaltung Anfang Dezember
Präsentation Web-Projekt
 - Ende Dezember
Abgabe GIS-Projekt
 - Anfang Januar
Abgabe Web-Projekt
 - Ende Januar
Mündliche Prüfung

Organisatorisches

- Anforderungen – Berufsbegleitender Master Geoinformatik
 - Studienleistung
 - Anwesenheit und Mitarbeit in den Übungen
 - Abgabe GIS-Projekt
 - Prüfungsleistung (benotet)
 - Klausur Semantic Web (40%)
 - GIS-Projekt (60%)

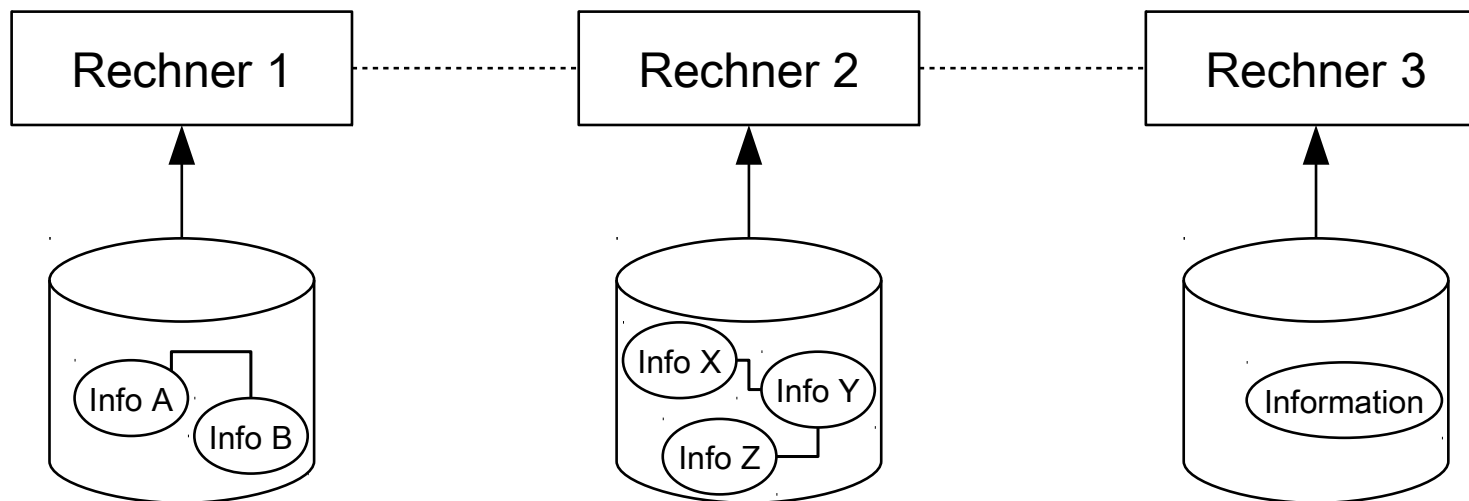
Organisatorisches

- Anforderungen – Konsekutiver Master Geoinformatik und Vermessung
 - Studienleistung
 - Anwesenheit und Mitarbeit in den Übungen
 - Abgabe GIS-Projekt
 - Prüfungsleistung (benotet)
 - Klausur Semantic Web (40%)
 - Web-Projekt (60%)
 - besteht aus Präsentation, Abgabe und mündlicher Prüfung

Semantic Web

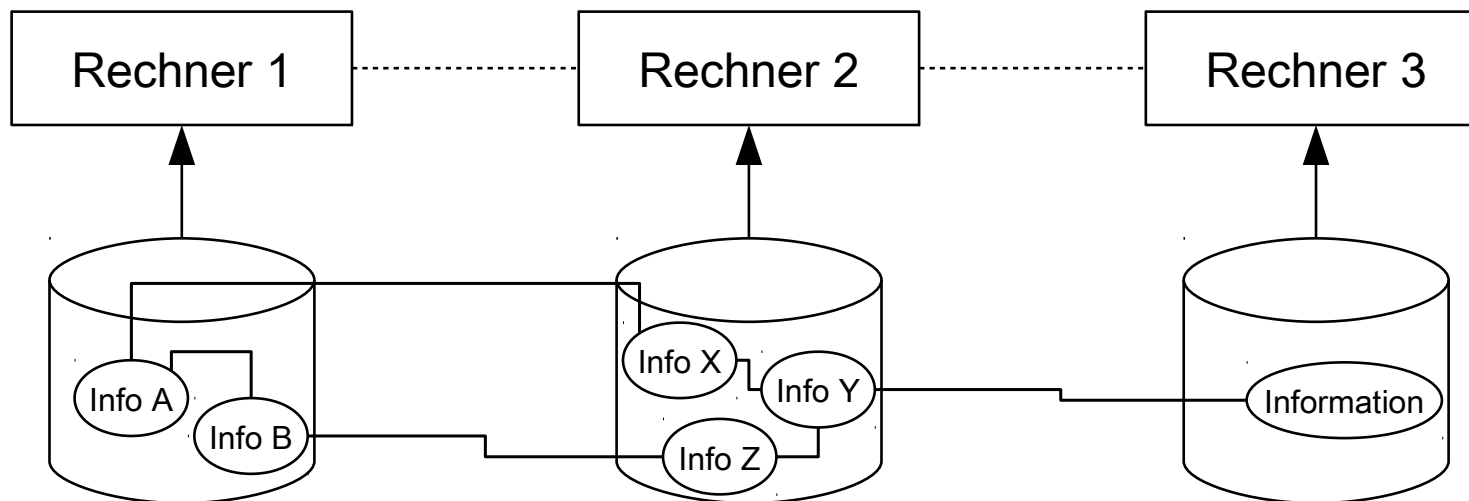
Semantic Web

- Die Idee ist, Informationen zwischen Rechnern über das Web zu verbinden, dadurch austausch- und verwertbarer zu machen



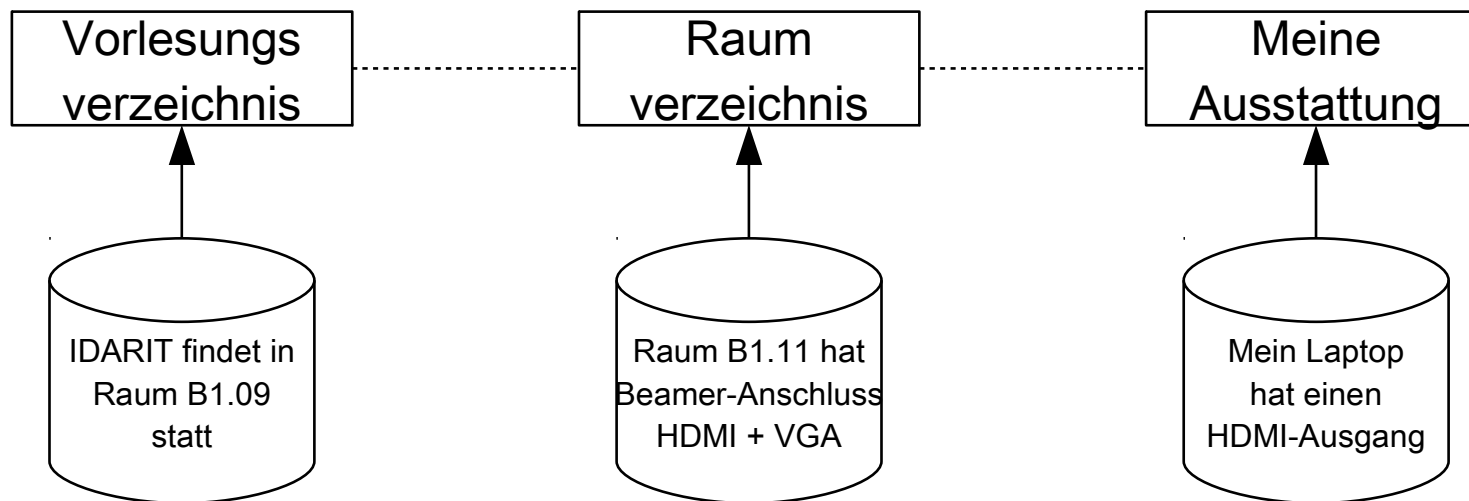
Semantic Web

- Die Idee ist, Informationen zwischen Rechnern über das Web zu verbinden, dadurch austausch- und verwertbarer zu machen
- Ziel ist die Erstellung eines „gigantischen globalen Graphs“, in dem möglichst viele Daten weltweit miteinander vernetzt sind



Semantic Web

- Beispiel
 - Kann ich meinen Laptop für die Präsentation in der IDARIT-Vorlesung verwenden?



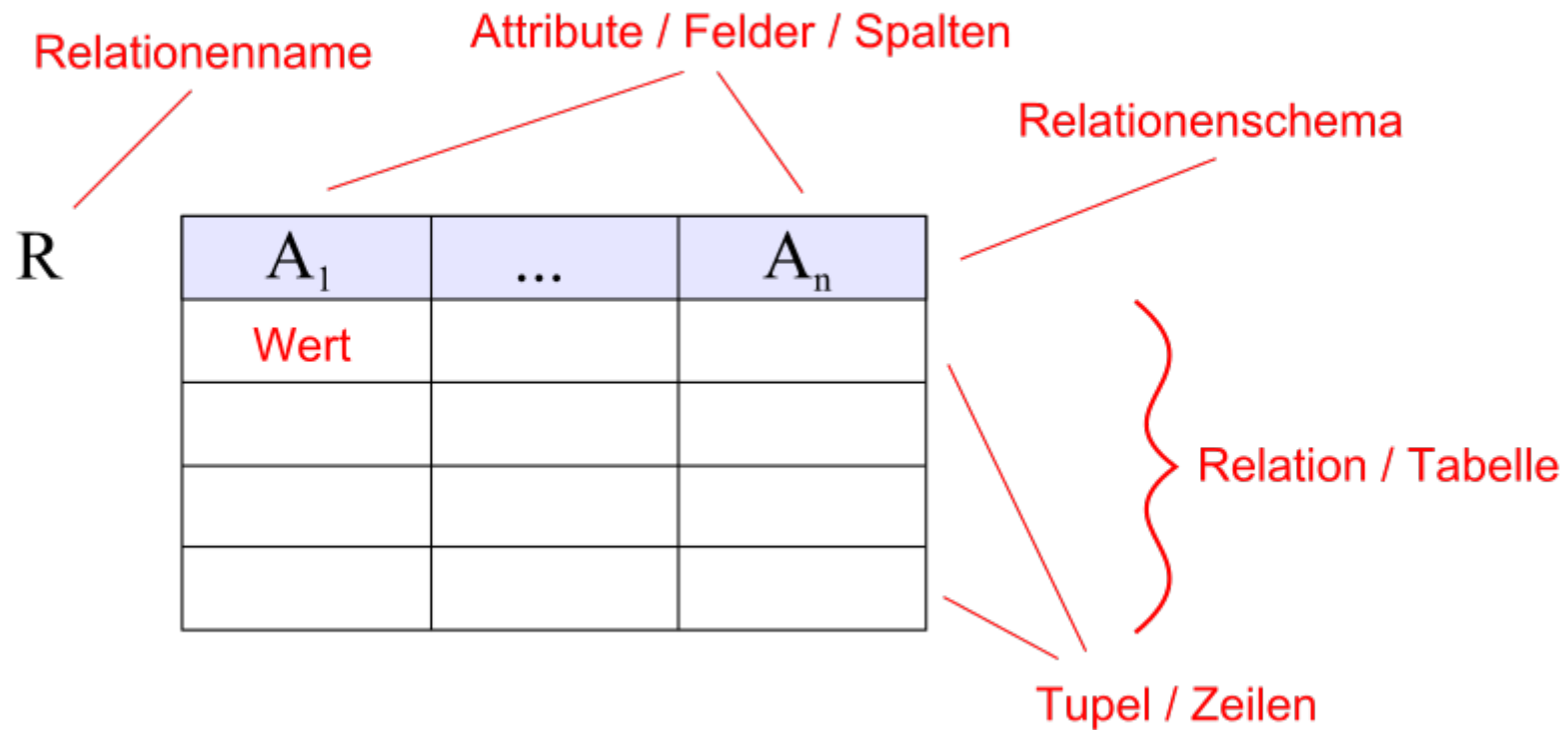
Datenbanken

Relationale Datenbanken

- Am weitesten verbreitetes Datenbankmodell
- Grundlegendes Konzept
 - Tabellen (Relationen)
 - Der Kopf der Tabelle ist das Relationenschema
 - Jede Zeile einer Tabelle (Tupel) ist ein Datensatz
 - Jedes Tupel enthält gemäß dem Relationenschema der Tabelle gewisse Werte in jeder Spalte (Attribute)
 - Beziehungen zwischen Tabellen können über Schlüssel-Attribute ausgedrückt werden

Relationale Datenbanken

- Relation (allgemein)



Relationale Datenbanken

- Relation Buch

Beispiel einer Relation „Buch“:

Buch-ID	Autor	Verlag	Verlagsjahr	Titel	Datum
1	Hans Vielschreiber	Musterverlag	2007	Wir lernen SQL	13.01.2007
2	J. Gutenberg	Gutenberg und Co.	1452	Drucken leicht gemacht	01.01.1452
3	G. I. Caesar	Handschriftverlag	-44	Mein Leben mit Asterix	16.03.-44
5	Galileo Galilei	Inquisition International	1640	Eppur si muove	1641
6	Charles Darwin	Vatikan-Verlag	1860	Adam und Eva	1862

Relationale Datenbanken

- Beispiel:
Datenbank einer
Bibliothek

Beispiel einer Relation „Buch“:

Buch-ID	Autor	Verlag	Verlagsjahr	Titel	Datum
1	Hans Vielschreiber	Musterverlag	2007	Wir lernen SQL	13.01.2007
2	J. Gutenberg	Gutenberg und Co.	1452	Drucken leicht gemacht	01.01.1452
3	G. I. Caesar	Handschriftverlag	-44	Mein Leben mit Asterix	16.03.-44
5	Galileo Galilei	Inquisition International	1640	Eppur si muove	1641
6		Vatikan-Verlag	1860	Adam und Eva	1862

 Relation
„Entliehen“

Nutzer-ID	Buch-ID
10	1
10	2
10	3
12	5
12	6

Relation „Nutzer“

Nutzer-ID	Vorname	Nachname
10	Hans	Vielschreiber
11	Jens	Mittelleser
12	Erich	Wenigleser

Relationale Datenbanken

- Beziehungen zwischen Relationen

Beispiel einer Relation „Buch“:

Buch-ID	Autor	Verlag	Verlagsjahr	Titel	Datum
1	Hans Vielschreiber	Musterverlag	2007	Wir lernen SQL	13.01.2007
2	J. Gutenberg	Gutenberg und Co.	1452	Drucken leicht gemacht	01.01.1452
3	G. I. Caesar	Handschriftverlag	-44	Mein Leben mit Asterix	16.03.-44
5	Galileo Galilei	Inquisition International	1640	Eppur si muove	1641
6		Vatikan-Verlag	1860	Adam und Eva	1862

 Relation
 „Entliehen“

Nutzer-ID	Buch-ID
10	1
10	2
10	3
12	5
12	6

Relation „Nutzer“

Nutzer-ID	Vorname	Nachname
10	Hans	Vielschreiber
11	Jens	Mittelleser
12	Erich	Wenigleser



Relationale Datenbanken

- Anforderungen
 - Atomarität
 - Folge von Operationen wird komplett durchgeführt oder gar nicht
 - Konsistenz
 - Primär- und Fremdschlüssel des Datenbestands müssen in sich stimmig sein
 - Durch Normalisierung werden Redundanzen vermieden
 - Isolation
 - Gleichzeitige Operationen dürfen sich nicht gegenseitig beeinflussen
 - Dauerhaftigkeit
 - Daten müssen auch bei Systemausfall reproduzierbar sein

Relationale Datenbanken

- Nachteile von Relationalen Datenbanken
 - Schlechte Performanz
bei großem Datenbestand und vielen Zugriffen
 - Künstliche Schlüsselattribute manchmal notwendig
 - Datensätze müssen exakt ins Relationenschema passen
 - Daten aus verschiedenen Datenbanken können nur mühevoll
zusammengebracht werden

NoSQL-Datenbanken

Graphdatenbanken

Triple-Stores

Graphentheorie

Ein einfacher gerichteter Graph ist ein Tupel $G=(V, E)$

bestehend aus

- Einer Menge an Knoten (vertices) $V = \{v_1, v_2, \dots, v_N\}$
- Einer Menge an Kanten (edges) $E \subset V \times V$

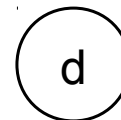
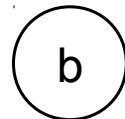
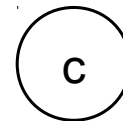
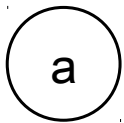
Graphentheorie

- Beispiel
 - $G=(V, E)$

Graphentheorie

- Beispiel

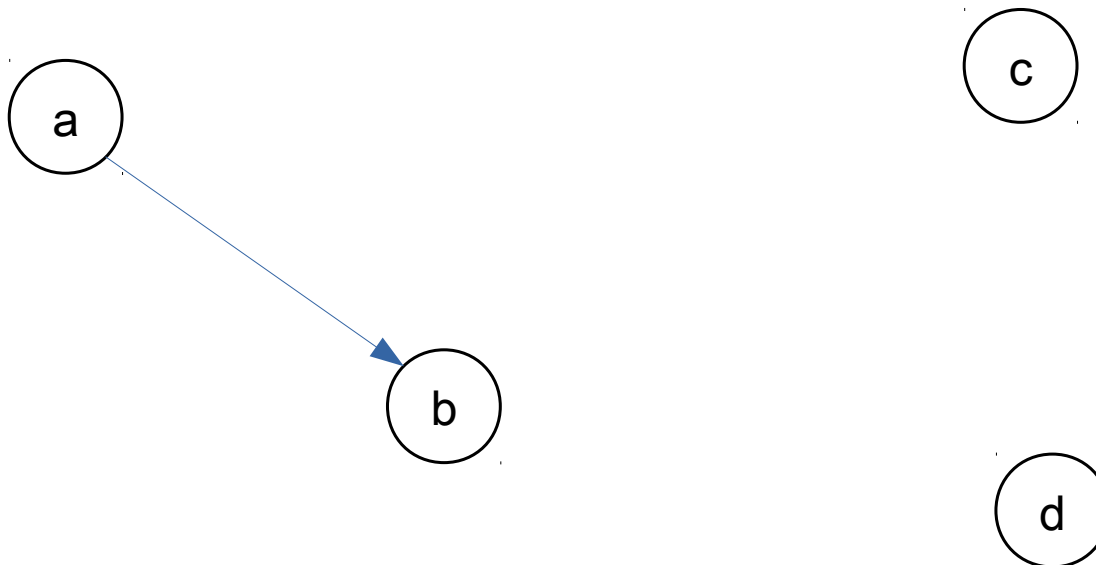
- $G=(V, E)$
- $V=\{a, b, c, d\}$



Graphentheorie

- Beispiel

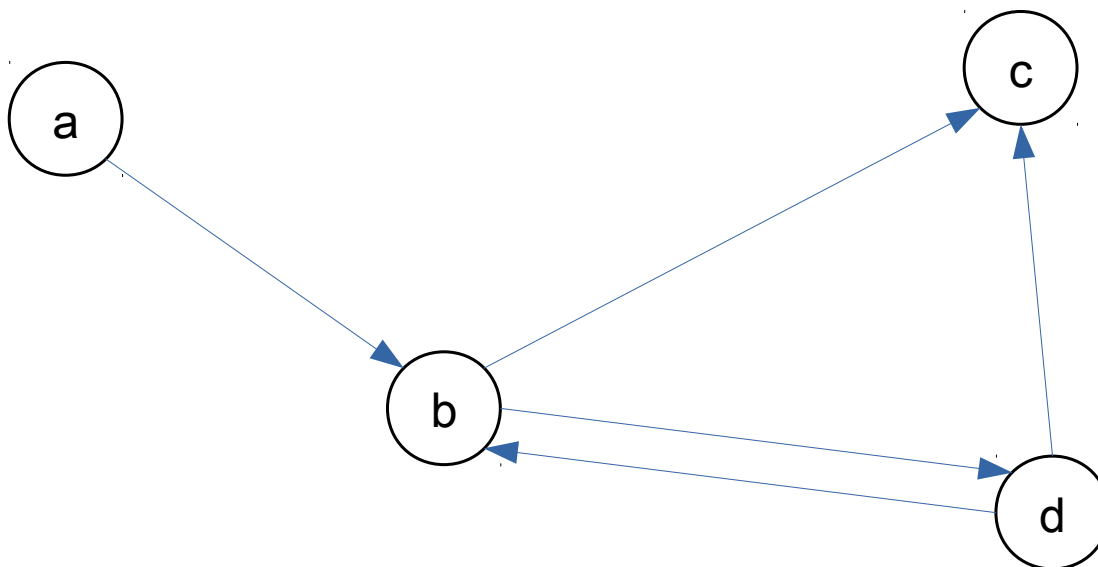
- $G=(V, E)$
- $V=\{a, b, c, d\}$
- $E=\{(a, b)\}$



Graphentheorie

- Beispiel

- $G=(V, E)$
- $V=\{a, b, c, d\}$
- $E=\{(a, b), (b, c), (b, d), (d, b), (d, c)\}$



Graphentheorie

- Ein kantengefärbter gerichteter Graph ist ein Tupel $G = (V, E, f)$

bestehend aus

- Einer Menge an Knoten (vertices) $V = \{v_1, v_2, \dots, v_N\}$
- Einer Menge an Kanten (edges) $E \subset V \times V$
- Einer Färbefunktion, die jeder Kante eine Farbe zuordnet $f : E \rightarrow C$
sowie eine Menge an möglichen Farben $C = \{c_1, c_2, \dots, c_M\}$

Graphentheorie

- Beispiel

- $G=(V, E, f)$

- $V=\{a, b, c, d\}$

- $E=\{(a, b), (b, c), (b, d), (d, b), (d, c)\}$

$$f((b, c))=rot$$

$$f((d, b))=rot$$

$$f((a, b))=blau$$

$$f((b, d))=blau$$

$$f((d, c))=blau$$

Graphentheorie

• Beispiel

- $G = (V, E, f)$
- $V = \{a, b, c, d\}$
- $E = \{(a, b), (b, c), (b, d), (d, b), (d, c)\}$

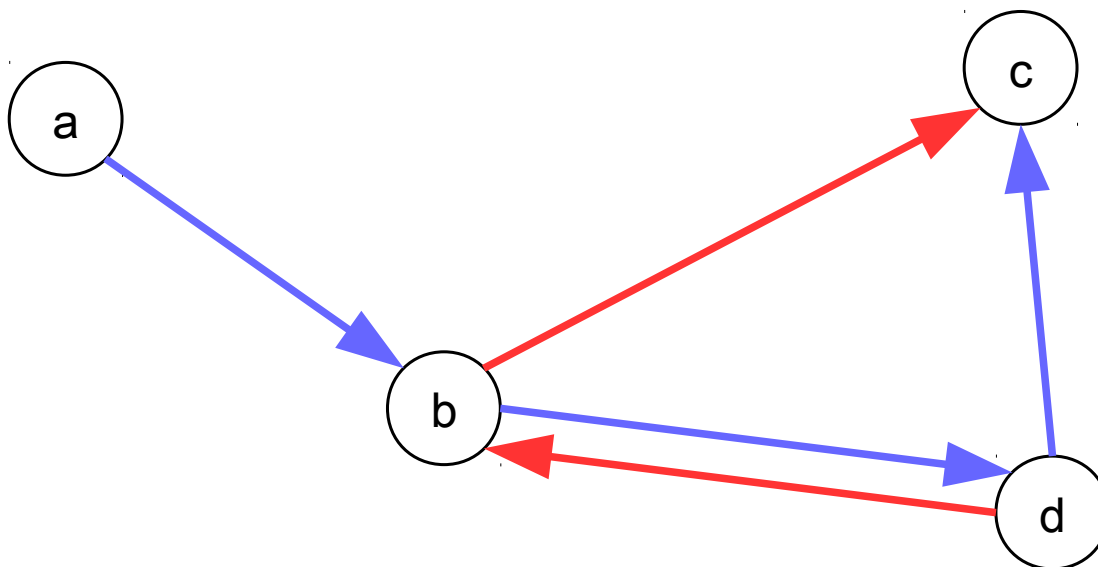
$$f((b, c)) = \text{rot}$$

$$f((d, b)) = \text{rot}$$

$$f((a, b)) = \text{blau}$$

$$f((b, d)) = \text{blau}$$

$$f((d, c)) = \text{blau}$$



Graphdatenbanken

und

Triple-Stores

Relationale Datenbank

- Beispiel: Bibliothek

Beispiel einer Relation „Buch“:

Buch-ID	Autor	Verlag	Verlagsjahr	Titel	Datum
1	Hans Vielschreiber	Musterverlag	2007	Wir lernen SQL	13.01.2007
2	J. Gutenberg	Gutenberg und Co.	1452	Drucken leicht gemacht	01.01.1452
3	G. I. Caesar	Handschriftverlag	-44	Mein Leben mit Asterix	16.03.-44
5	Galileo Galilei	Inquisition International	1640	Eppur si muove	1641
6	Charles Darwin	Vatikan-Verlag	1860	Adam und Eva	1862

Relation „Nutzer“

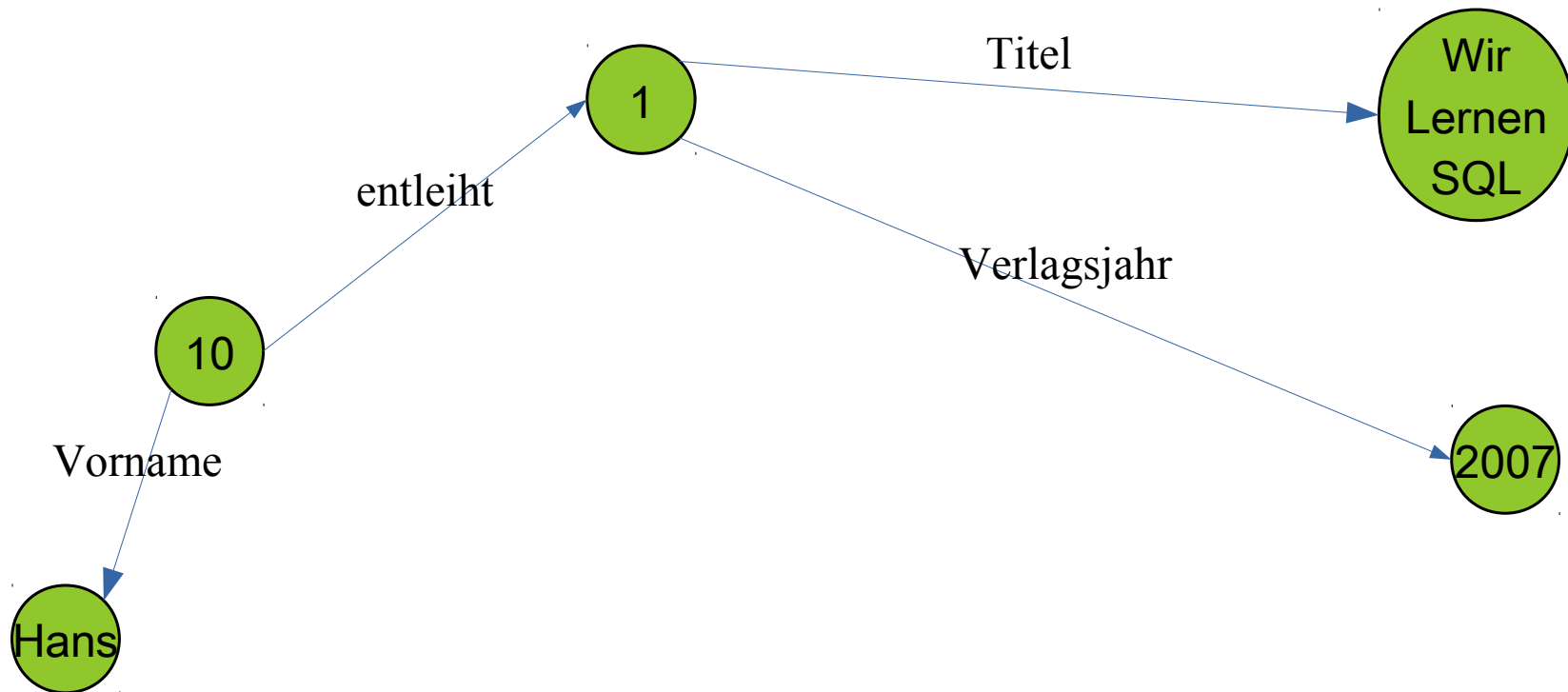
Nutzer-ID	Vorname	Nachname
10	Hans	Vielschreiber
11	Jens	Mittelleser
12	Erich	Wenigleser

Relation „Entliehen“

Nutzer-ID	Buch-ID
10	1
10	2
10	3
12	5
12	6

Graphdatenbank

- Beispiel: Bibliothek (Ausschnitt)



- Uniform Resource Identifier
 - Eindeutiger Bezeichner für eine Ressource
- Beispiele für Ressourcen
 - Personen
 - Smartphones
 - Bücher
 - Autos
- Jede URL (Uniform Resource Locator) ist eine URI
 - Eine URL ist eine Web-Adresse (http)

RDF Turtle Syntax

- **Subjekt**
Ausgehender Knoten
- **Prädikat**
Beschriftung der Kante
- **Objekt**
Eingehender Knoten
- **Punkt**

RDF Turtle Syntax

- URIs mit spitzen Klammern `< ... >`
- Subjekte und Prädikate (ausgehender Knoten und Kantenbeschriftung) müssen URIs sein
- Literale gemäß Datentyp, zum Beispiel:
 - Integer `123`
 - Double `12.3`
 - String `"bla"`

Triple Store

- Beispiel: Bibliothek (mit URI)



Triple Store

- Beispiel: Bibliothek (RDF Turtle Syntax)

<http://www.bibliothek.de/10>

<http://www.bibliothek.de/Vorname>

„Hans“

```
<http://www.bibliothek.de/10>
```

```
<http://www.bibliothek.de/Vorname> "Hans" .
```

Namespace

- Organisation von Bezeichnern
 - Ressourcen erhalten Namespace + eigentlichen Namen
- Wo werden Namespaces verwendet?
 - Dateisysteme (Ordner + Dateinamen)
 - Programmiersprachen (Pakete + Klassen + Variablen/Methoden)
 - Häuserbezeichnung (Postleitzahl + Straßename + Hausnummer)
 - ...

Triple Store

- Beispiel: Bibliothek (Namespace)

```
@prefix bib: <http://www.bibliothek.de/> .
```

<http://www.bibliothek.de/10>

<http://www.bibliothek.de/Vorname>

„Hans“

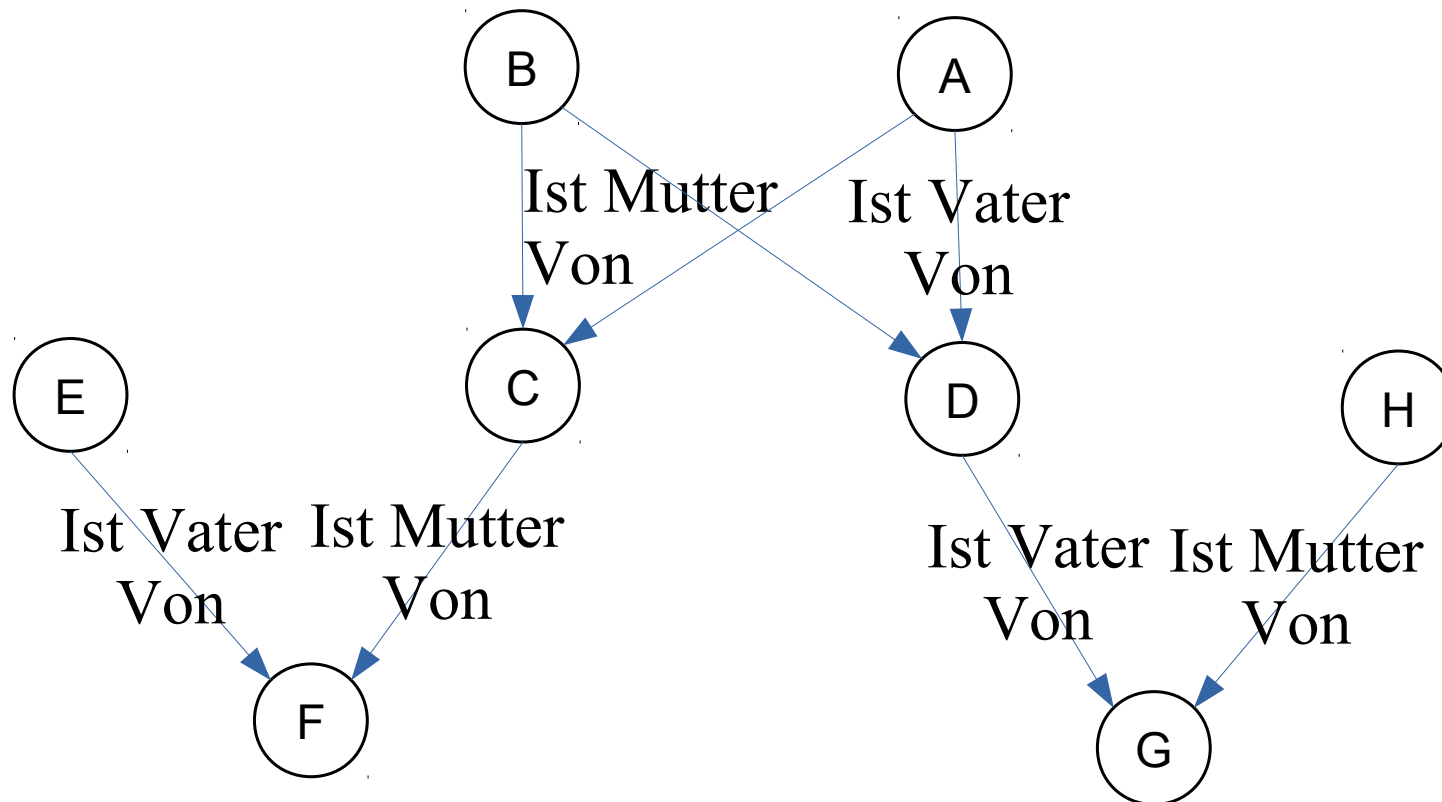
```
bib:10 bib:Vorname "Hans" .
```


Kurze Pause

Graphdatenbank

- Ein anderes Beispiel für Daten, die als Graph modellierbar sind
 - **Anton** Ist Vater Von **Christine**
 - **Berta** Ist Mutter Von **Christine**
 - **Anton** Ist Vater Von **Daniel**
 - **Berta** Ist Mutter Von **Daniel**
 - **Emil** Ist Vater Von **Friedrich**
 - **Christine** Ist Mutter Von **Friedrich**
 - **Daniel** Ist Vater Von **Gustav**
 - **Heike** Ist Mutter Von **Gustav**

Graphdatenbank



Triple-Store

- Turtle Syntax für den Stammbaum

<https://www.w3.org/TR/turtle/>

```
@prefix idarit: <http://idarit.hs-mainz.de/> .
```

```
idarit:Anton      idarit:Vater      idarit:Christine .  
idarit:Berta      idarit:Mutter     idarit:Christine .  
idarit:Anton      idarit:Vater      idarit:Daniel .  
idarit:Berta      idarit:Mutter     idarit:Daniel .  
idarit:Emil       idarit:Vater      idarit:Friedrich .  
idarit:Christine idarit:Mutter     idarit:Friedrich .  
idarit:Daniel     idarit:Vater      idarit:Gustav .  
idarit:Heike      idarit:Mutter     idarit:Gustav .
```

Datenbankabfragen

- SPARQL

<https://www.w3.org/TR/rdf-sparql-query/>

```
SELECT ... WHERE {
```

```
    Variable: ?...
```

```
    URI: <...>
```

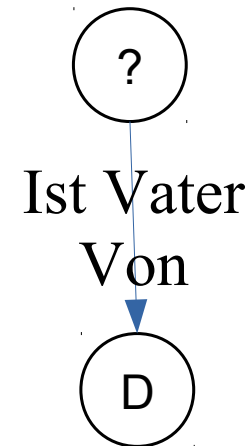
```
}
```

Datenbankabfragen

- Beispiel:
Wer ist Vater von Daniel?

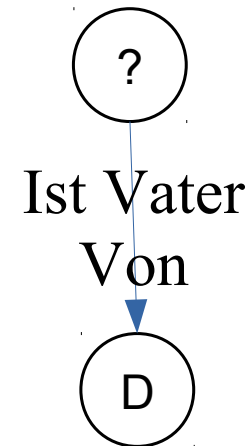
Datenbankabfragen

- Beispiel:
Wer ist Vater von Daniel?



Datenbankabfragen

- Beispiel:
Wer ist Vater von Daniel?



```
PREFIX idarit: <http://idarit.hs-mainz.de/>
```

```
SELECT ?person WHERE {
```

```
    ?person idarit:Vater idarit:Daniel .
```

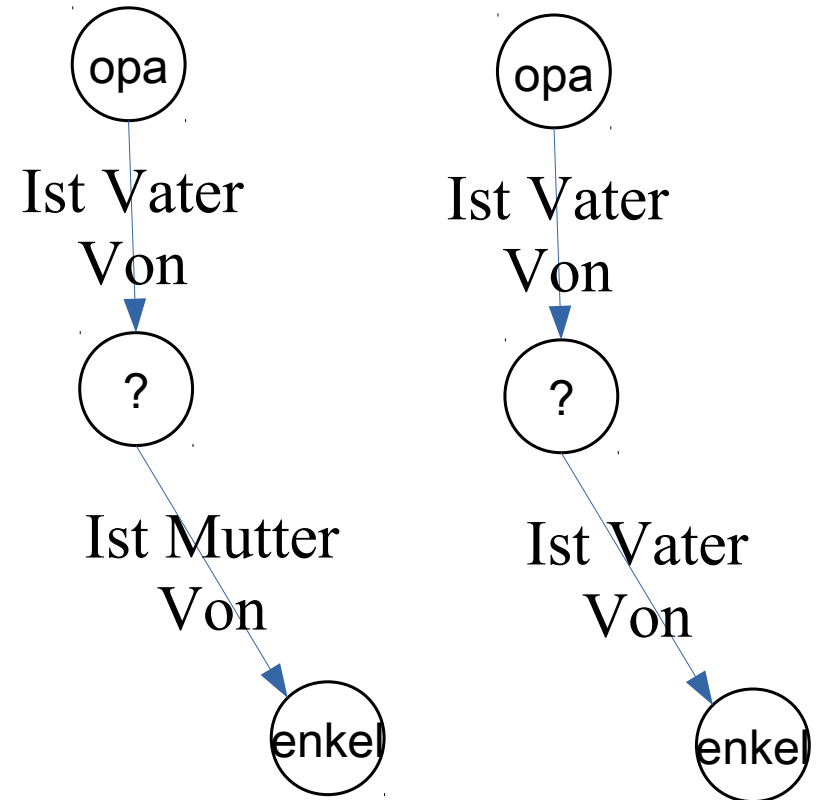
```
}
```


Datenbankabfragen

- Beispiel:
Wer ist Opa von wem?

Datenbankabfragen

- Beispiel:
Wer ist Opa von wem?



Datenbankabfragen

- Beispiel:

Wer ist Opa von wem?

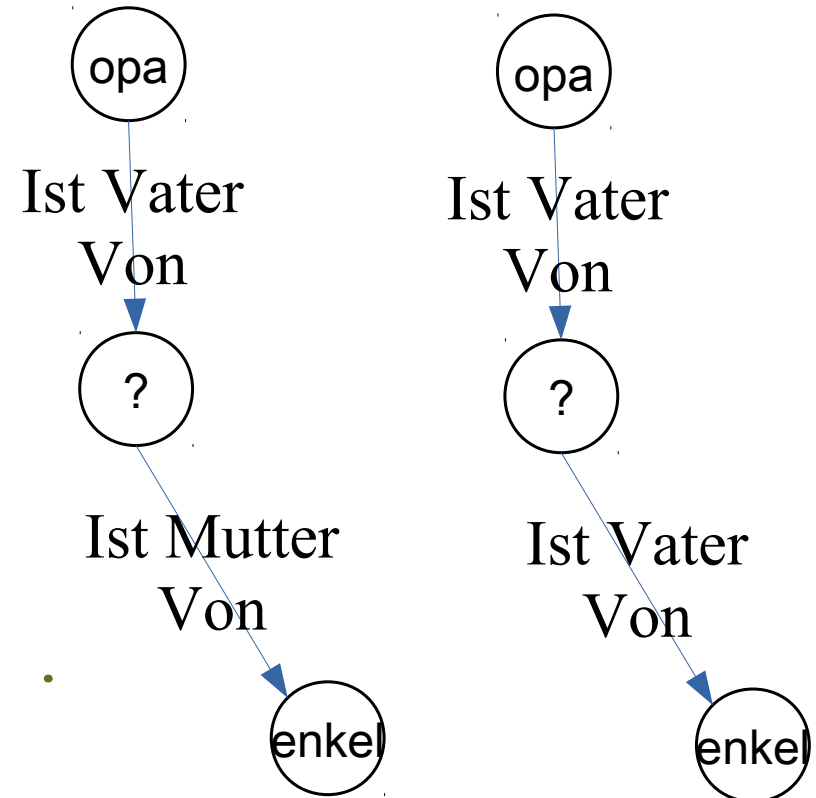
PREFIX ...

```
SELECT ?opa ?enkel WHERE {
```

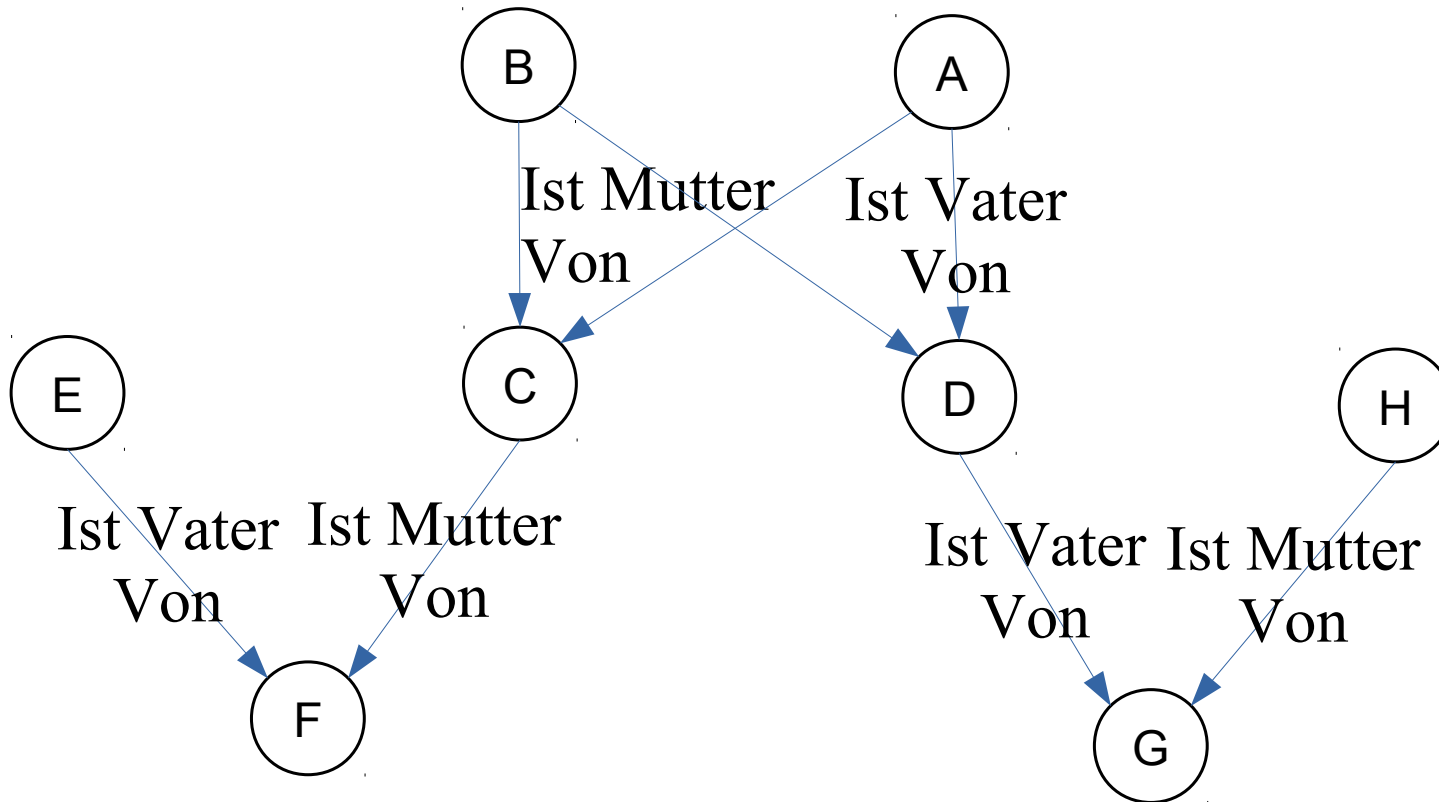
```
{
  ?opa idarit:Vater ?x .
  ?x idarit:Mutter ?enkel .
}
```

```
UNION
```

```
{
  ?opa idarit:Vater ?x .
  ?x idarit:Vater ?enkel .
}
```



Weitere Abfragen ?



SPARQL Filter

- Numerische Filter
 - Beispiel: Der Wert der Variablen x soll größer als 100 sein
`FILTER (?x > 100)`
- Reguläre Ausdrücke
 - Beispiel: Der String der Variablen x soll ein a enthalten
`FILTER (regex (?x, "a"))`
- Negation
 - Beispiel: x soll nicht Vater von Daniel sein
`FILTER NOT EXISTS { ?x idarit:Vater
idarit:Daniel . }`

Vielen Dank
für die
Aufmerksamkeit