Fünf Dinge „aus der Informatik“, die ein Digital Humanist kennen sollte

Dipl.-Math. Martin Sievers

Trier Center for Digital Humanities

DH Orientierungstage WiSe 2016 / 2017

Einführung / Motivation

- Was sind die „Digital Humanities“?
- Was tut ein „Digital Humanist“?
- Interdisziplinäres Team am TCDH
- Abgrenzung (manchmal „Kampfsprache“):
  - Geisteswissenschaftler vs. Informatiker
  - Anwender vs. Programmierer
  - Fachwissenschaftler vs. Techniker
- Ziel: Diskussion auf Augenhöhe
- Themenauswahl subjektiv, aber nicht willkürlich
- Kein Anspruch auf Vollständigkeit

Gliederung

Gliederung

Idee / Konzept

Ein regulärer Ausdruck ermöglicht es, einen Text nach einem bestimmten Muster (mit Wildcards) zu durchsuchen und dieses Muster (oder Teile davon) zu ersetzen.
Pattern Matching (Musterabgleich): Suche alle Vorkommen einer Zeichenkette im Text

- Abkürzungen: RegExp oder Regex
- Unterstützung in Skript- und Programmiersprachen sowie in Texteditoren
- Verschiedene Standards:
  - Perl Compatible Regular Expressions (PCRE)
  - Portable Operating System Interface (POSIX.2)

  (Unterscheidung zwischen Basic RegExp und Extented RegExp)

Zeichenvorrat

- Einzelne Zeichen: `a` `b` `0`
- Zeichenauswahl:

  `[abc]` Menge einzelner Buchstaben

  `[0-5]` Zeichenbereich (kann wiederholt werden: `[0-9a-f]`)

  `[^a]` Ausschluss von Zeichen
- Vordefinierte Zeichenklassen, u. a.:
  - `\d` $\hat{=}$ `[0-9]`
  - `\w` $\hat{=}$ `[a-zA-Z_0-9]` (und ggf. Umlaute)
  - `\s` Whitespace-Character (Leerzeichen, Tabs, Zeilenumbruch, …)
  - Komplimentärmengen `\D`, `\W` und `\S`
- `.` steht für ein beliebiges Zeichen (`\.` sucht das Zeichen .)
- `^` markiert Anfang, `$` Ende einer Zeichenkette bzw. Zeile

Operationen

Verkettung Aneinanderreihung einzelner Ausdrücke: `Test`

Alternative `|` für boolesche ODER-Verknüpfung: `Test|Seite`

Wiederholung Angabe von „Quantoren“ für den voranstehenden Ausdruck:

- `{min,max}` Vorkommen mindestens `min`-mal und höchstens `max`-mal
  Varianten: `{n}`, `{min,}`, `{0,max}`
- `?` Vorkommen optional (≙ `{0,1}`)
- `+` Vorkommen mindestens einmal (≙ `{1,}`)
- `*` Vorkommen beliebig oft (≙ `{0,}`)

Ggf. ist eine Klammerung nötig. Vgl. `ab|c` vs. `a(b|c)`

Besonderheiten

- Gierig vs. faul
  - Quantoren sind greedy, d. h. sie liefern immer die größtmögliche Übereinstimmung zurück
  - Non-greedy oder lazy durch zusätzliches `?`
  - wird nicht von allen Implementierungen unterstützt
  - Beispiel: `A.*B` bzw. `A.*?B` für Zeichenfolge ABCDEB
- Gruppierung und Rückwärtsreferenzierung
  - Ausdrücke können durch `(...)` gruppiert (und gespeichert) werden (Capturing group)
  - späterer Zugriff (beim Ersetzen) durch `\1` (oder `$1`)
  - Erzeugung von Rückwärtsreferenzen kostet Laufzeit und Speicher
  - Non-capturing group `(?:...)`

Anwendungsbeispiel I

Aufgabe

Ersetze alle normalen Leerzeichen bei Seitenangaben der Form S.␣X durch geschützte Leerzeichen (in LaTeX durch ~)

- Definiere präzises Suchmuster:
  - S ist einfaches Zeichen (Character)
  - . ist Sonderzeichen (Escaped character)
  - Leerzeichen kann als Klasse ausgedrückt werden (Whitespace)
  - Seitenangabe ist ein Bereich mit Wildcard +
  - Seitenangabe muss man sich merken (Capturing group)

  $\Rightarrow$ Suche nach `S\.\s([0-9]+)`
- Ersetzungsausdruck `S.~$1`
- Tests z. B. über `http://regexr.com/`

Anwendungsbeispiel II

Aufgabe

Ein Datum im Format `MM/DD/YYYY` soll in das Format `YYYY-MM-DD` überführt werden.

- Monat ein- oder zweistellig: erste Ziffer 0 oder 1, zweite 0 bis 9
- Tag ein- oder zweistellig: erste Ziffer 0 bis 3, zweite 0 bis 9
- Jahreszahl muss vierstellig sein: Ziffern jeweils 0 bis 9
- Gruppierung der Einzelteile
- / muss je nach Anwendung maskiert werden

⇒ Suche nach `([0-1]?[0-9])\/([0-3]?[0-9])\/([0-9]{4})`

- Ersetzungsausdruck: `\3-\1-\2`

Gliederung

Idee / Konzept

Eine Auszeichnungssprache (markup language) ist eine maschinenlesbare Sprache für die Gliederung und Formatierung von Texten und anderen Daten.

- Begriff ursprünglich aus der Druckersprache
- Typographische Bedeutung
- Trennung von Inhalt und Form führt zu Kennzeichnung von Elementen gleicher Bedeutung (semantische Auszeichnung)
- SGML (Standard Generalized Markup Language) 1986 und XML (Extensible Markup Language) 1998 als Metasprachen zur Definition von Auszeichnungssprachen
- Wichtigste Anwendung in den DH: TEI P5 (Text Encoding Initiative)

Aufbau einer XML-Datei

- Aufteilung in Kopf (<head>) und Hauptteil (<body>)
- Umschließe Teile (Elemente) in <. . .> (Tags)
- Tags können Attribute enthalten:

```
<body> <!-- vgl. http://teibyexample.org/examples/TBED07v00.htm -->
 <div>
  <lg>
   <l n="1"> <app>
    <rdg wit="#H201 #L1894 #LL">Faith</rdg>
    <rdg wit="#P1891 #CP">FAITH</rdg>
   </app> is a fine invention</l>
```

- Schema zur Validierung eines Dokuments (DTD, RelaxNG)
- Vielfältige Analyse-, Umwandlungs- und Weiterverarbeitungsmöglichkeiten (X-Technologien)

Spezialfall LaTeX

I hope to die before I have to use Microsoft Word.

Donald E. Knuth

- LaTeX als Markup-Sprache basierend auf TeX
- Turing-vollständige Programmiersprache
- Idee: Textsatz als Optimierungsproblem
- Erste Veröffentlichung 1978 (TeX78)
- Open Source
- Breite Unterstützung des wissenschaftlichen Textsatzes

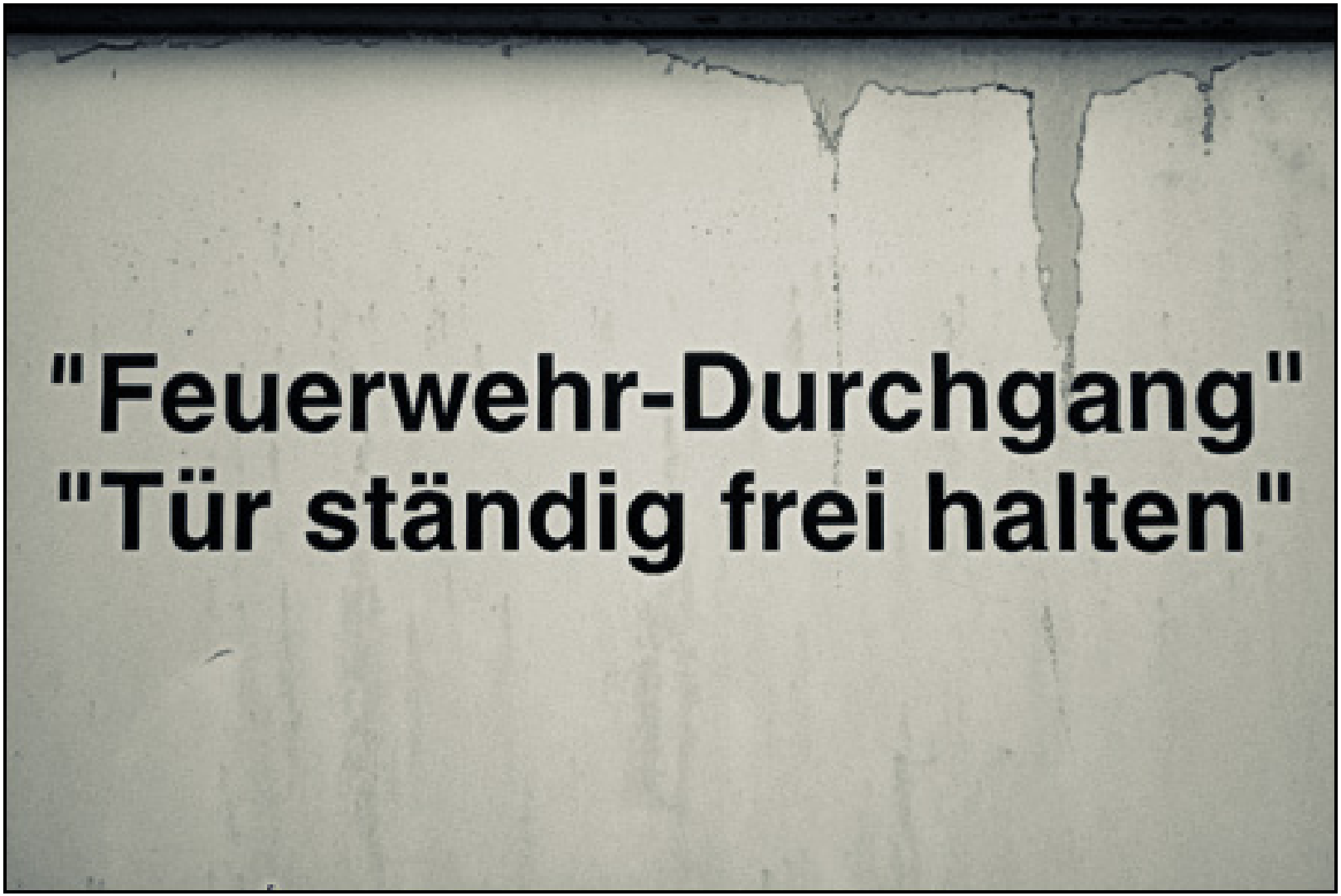
Gliederung

Idee / Konzept

Typographie bezieht sich klassischerweise auf die Kunst und das Handwerk des Druckens mit beweglichen Lettern (Gutenberg).

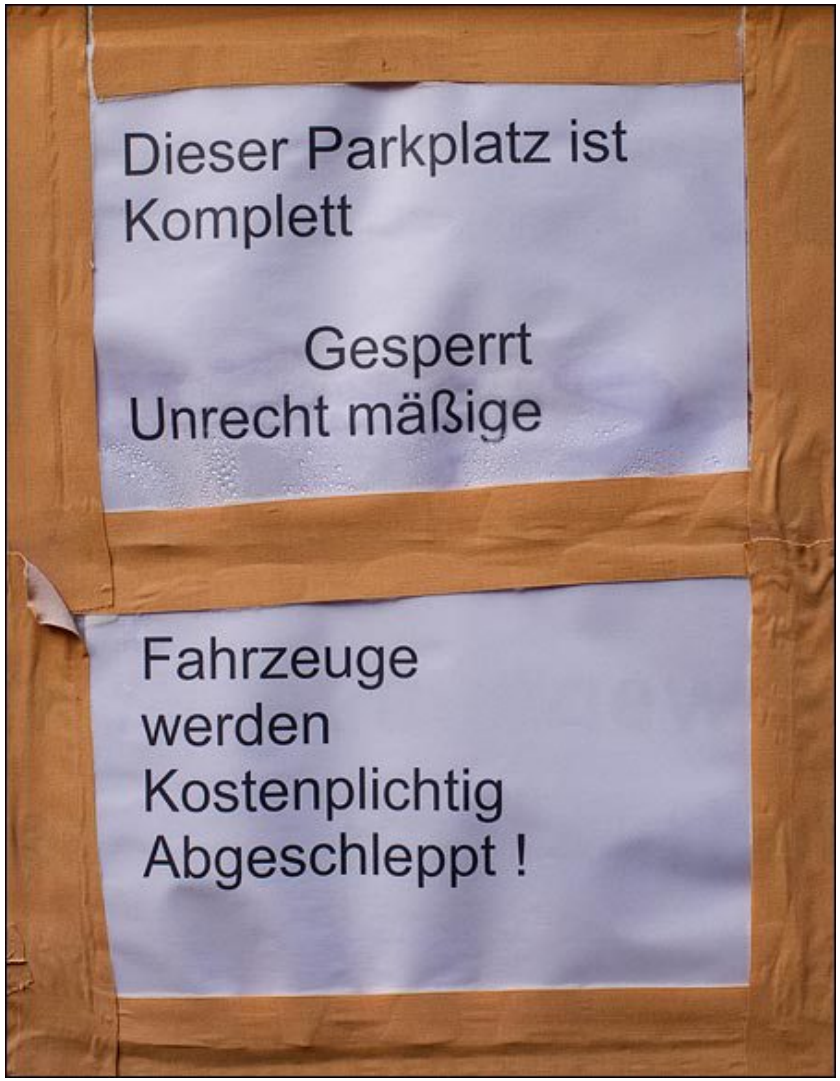
- Heutzutage oft Oberbegriff für den Gestaltungsprozess, auch für elektronische Medien
- Mehrere Ebenen:
  - Makrotypographie beschreibt die Gesamtheit einer Druckseite oder Webpage (Seitenformat, Satzspiegel, Platzierung von Bildern und Tabellen, …)
  - Mikrotypographie (oder Detailtypographie) umfasst Feinheiten des Schriftsatzes (Schriftart, Kapitälchen, Ligaturen, Laufweite, …)

Beispiele aus dem Alltag (I)



Quelle: `https://butschinsky.wordpress.com/2011/09/14/spitze-finger-ironie-dialog/`

Beispiele aus dem Alltag (II)



Quelle: `https://butschinsky.wordpress.com/2011/09/10/ohne-titel/`

Beispiele aus dem Alltag (III)

Quelle: `https://butschinsky.wordpress.com/2011/09/10/der-apostroph/`

Konsequenzen für Publikationen

- Geeignete Software nutzen („Office“ ist nur bedingt geeignet)
- Verwendung (hochwertiger) OpenType-Schriften (Kapitälchen, Ligaturen, Mediävalziffern, …)
- Trennung von Inhalt / Struktur und Layout
- Verwendung korrekter Zeichen (Unicode) und Abstände:
  - – vs. -
  - „ “ vs. " " oder auch ’ vs. ´
  - 5 % vs. 5% / 5 %
  - …
- Absatz vs. Zeilenumbruch
- Web: Nutzung von CSS (Cascading Style Sheets)

Gliederung

Idee / Konzept

Eine Versionsverwaltung ist ein System zur Erfassung von Änderungen an Dokumenten oder Dateien

- Alle Versionen in Archiv (Repository)
- Zeitstempel und Nutzerkennung
- Typische Nutzung für Quelltexte in der Softwareentwicklung bzw. in CMS
- Anwendung auf Binärdaten (Grafiken, proprietäre Formate etc.) eingeschränkt möglich (kein Merge)
- Nutzung über eigene Server / NAS / externe Festplatten oder über Plattformen wie GitHub, Sourceforge und BitBucket bzw. Ticketsysteme wie Redmine

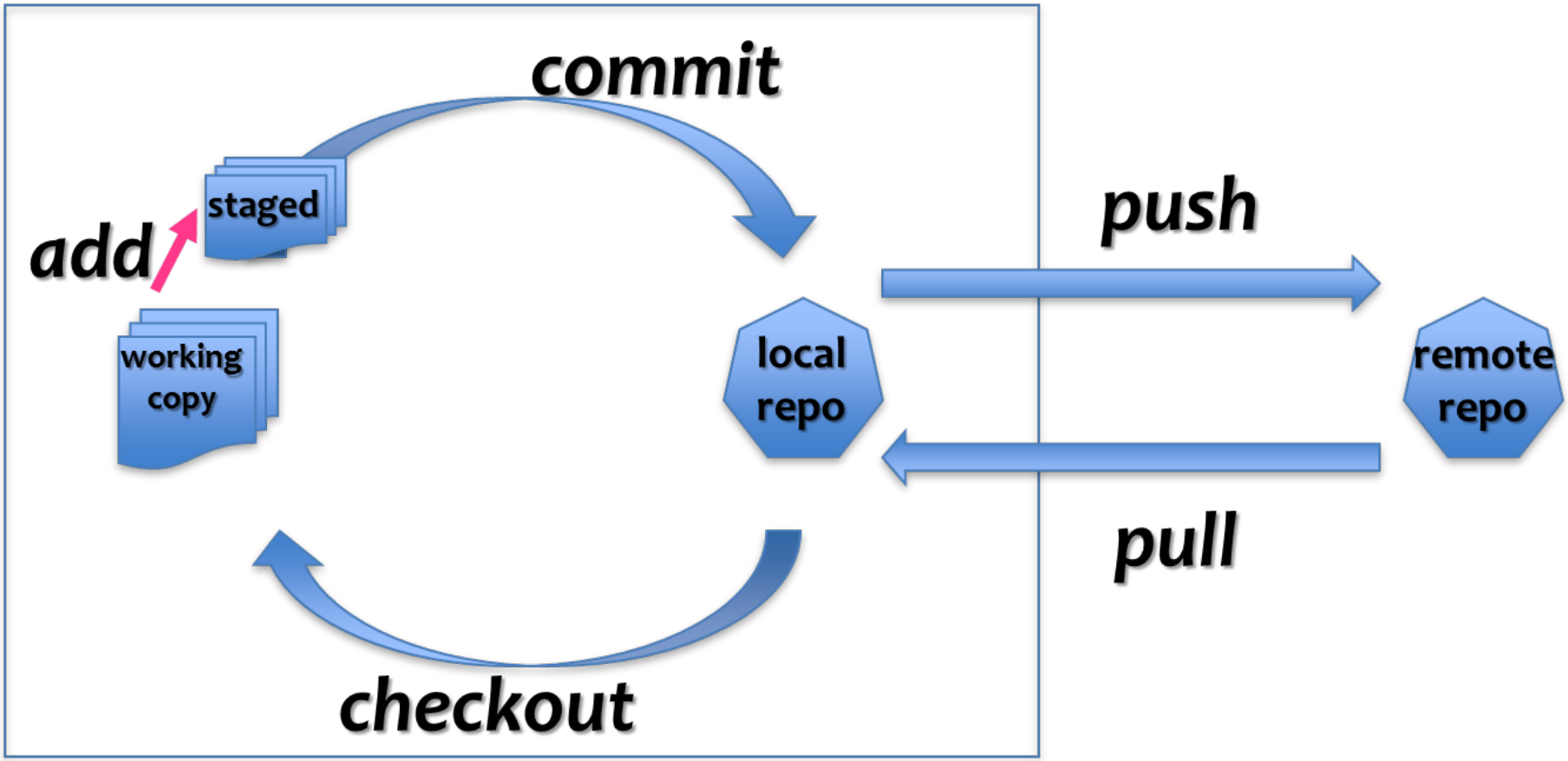
Funktionen einer Versionsverwaltung

- Protokollierungen der Änderungen:
  Wer hat wann was geändert
- Wiederherstellung alter Fassungen einzelner Dateien:
  Versehentliche Änderungen jederzeit rückgängig machen
- Archivierung einzelner Fassungen: jederzeit Zugriff auf alle Versionen
- Koordinierung des gemeinsamen Zugriffs mehrerer Entwickler
- Gleichzeitige Entwicklung durch mehrere Zweige (Branches)

Arbeitsweise von Versionsverwaltungen

- Variante 1: Zentrales Repository
  - Hinzufügen mit `commit`
  - Aktualisierung mit `update`
  - Beispiel: Subversion (SVN)
- Variante 2: Lokales und zentrales Repository
  - Indexierung per `add`, hinzufügen per `commit` bzw. `push` (lokal bzw. zentral)
  - Aktualisierung per `push` bzw. `checkout`; dabei evtl. `merge`
  - Beispiele: Git, Mercurial

Ablauf am Beispiel Git



 Brigitte Jellinek, entnommen `https://web-development.github.io/git/basic-git/`

Gliederung

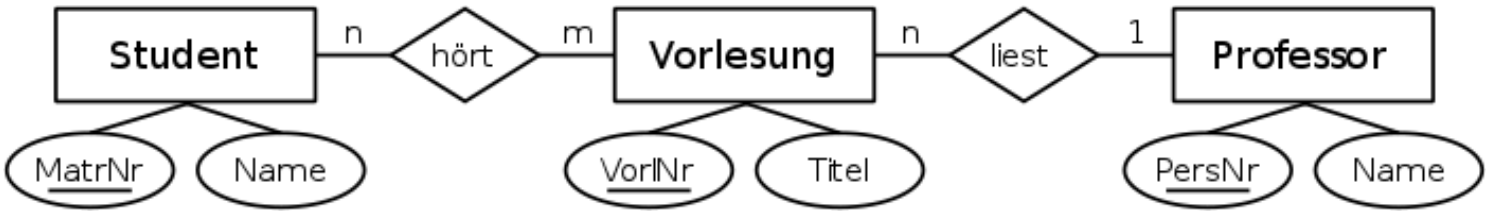
Idee / Konzept

Eine Datenbank speichert große Datenmengen effizient, widerspruchsfrei und dauerhaft. Sie stellt benötigte Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformen für Benutzer und Anwendungsprogramme bereit

- Unterschiedliche Datenbankmodelle
- Relationale Datenbanken (seit den 1970er-Jahren): MySQL
  - Relation als Tabelle: Attribute in Spalten, Werte in Zeilen
  - Arbeitssprache SQL (Structured Query Language)
  - am weitesten verbreiteter Datenbanktyp

Beispiel

Entity-Relationship-Modell (ER-Modell):



 Nils Boßung, entnommen `https://de.wikipedia.org/wiki/SQL`

Relationen:

Student

MatrNr	Name
26120	Fichte
25403	Jonas
27103	Fauler

hört

MatrNr	VorlNr
25403	5001
26120	5001
26120	5045

Vorlesung

VorlNr	Titel	PersNr
5001	ET	15
5022	IT	12
5045	DB	12

Professor

PersNr	Name
12	Wirth
15	Tesla
20	Urlauber

Beispiel für Relationen: `(26120,Fichte)`, `(5022,IT,12)`

Beispielabfragen

- **SELECT** VorlNr, Titel **FROM** Vorlesung;

VorlNr	Titel
5001	ET
5022	IT
5045	DB

- **SELECT** VorlNr, Titel **FROM** Vorlesung **WHERE** Titel = 'ET';

VorlNr	Titel
5001	ET

- **SELECT** Vorlesung.Titel, Professor.Name **FROM** Professor,

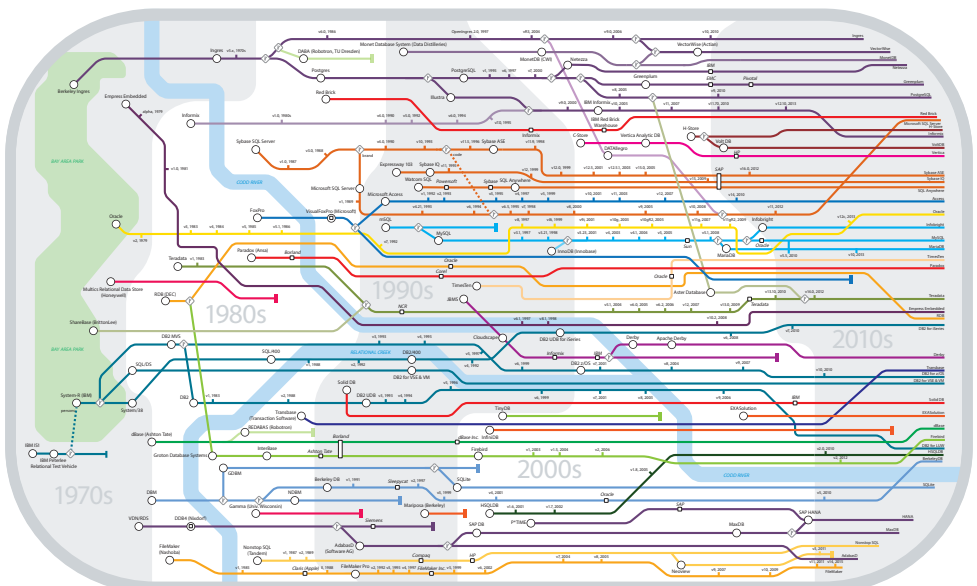
  Vorlesung **WHERE** Professor.PersNr = Vorlesung.PersNr;

Titel	Name
ET	Tesla
IT	Wirth
DB	Wirth

Alternativen zu SQL

- Probleme z. B. bei der Indexierung großer Datenmengen oder auch Websites mit hohen Lastaufkommen
- Alternative: NoSQL (Not only SQL, seit etwa 2000), u. a.
  - Dokumentorientierte Datenbanken (XML-Datenbanken): eXistDB
  - Graphdatenbanken: Neo4j

Genealogy of Relational Database Management Systems



https://hpi.de/naumann/projects/rdbms-genealogy.html

Fazit / Abschluss

- Informatik ist wesentlicher Bestandteil der DH, nicht notwendiges Übel
- Diskussionen auf Augenhöhe nur mit Vorkenntnissen möglich

Vielen Dank für Ihre Aufmerksamkeit

- Fragen?
- Anmerkungen?


sievers@uni-trier.de


TeX4Publication


`http://kompetenzzentrum.uni-trier.de`