Datenbanksysteme II:
Implementation of Database Systems

Ulf Leser

- Slides in English, Vortrag auf Deutsch
- Much input from
  - Prof. J-C Freytag, HU Berlin
  - Prof. K-U Sattler, TU Illmenau
  - Prof. A Kemper, Dr. Eickler, TU München
- AGNES
- Prof Freytag / Prof Leser
- What you should (must) know to follow this course
- What do you study?

Some Motivation

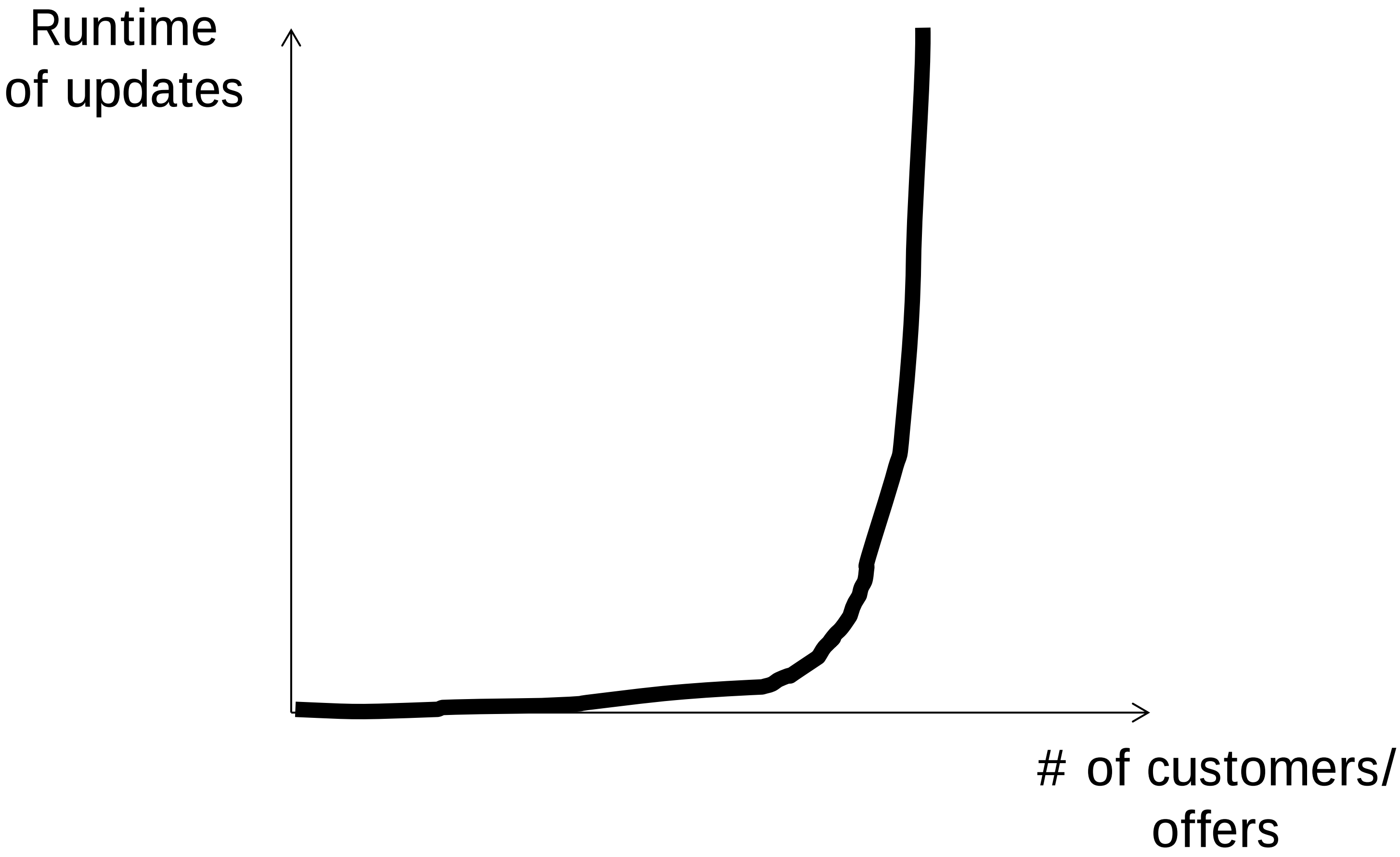
A classical first contact with database

- Company: “I need to track my customers”
  - Name, address, profession, prior contracts
- Naïve engineer: “No problem”
  - ~1984: Turbo Pascal 4.0, Schneider CPC646, 512 KB main memory
  - Each customer one record / line in file
  - Load customers from disk into memory
  - Repeat until “Q”
    - (S)earch and list customers
    - (E)dit customer
    - (D)elete customer
    - (I)nsert customer
    - (Q)uit
  - Write customers to disk
- Invoice: ... DM

Story part 2

- Company: “I need to track my offers”
  - Customers have projects and call for bids, company makes offers
  - Many customers have many projects over time
- Naïve engineer: “No problem”
  - Reuse existing architecture
  - Load offers from disk into memory
  - Repeat until “Q”
    - (S)earch and list offers
    - (E)dit offer
    - (D)elete offer
    - (I)nsert offer
    - (Q)uit
  - Write offers to disk
- Invoice: more DM

Story part 3



Part 3

- Disaster!
  - ~ 500 customers
  - ~ 40 offers per customer
  - ~ 2KB per offer
  - Gives 500*40*2.000 = 40 MEGABYTE
    - This was the size of a hard disc at that time
  - No way to load and hold all data at startup
- Wrong architecture
- Solution: Buy a RDBMS

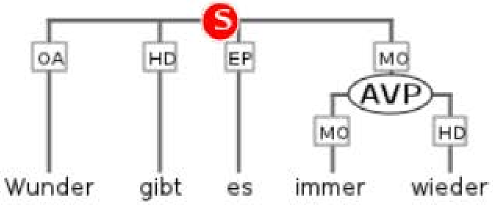
Lessons learned

- Scalability in data management is an issue
  - Scalable: Graceful runtime degradation with increasing data volumes
  - Not scalable: Works fine for small datasets, breaks down for large datasets
- Data is business-critical
  - If offers-file corrupted – company goes out of business
    - Which will affect engineer too!
- Think before you start programming
  - Project 100% over budget (database license)
  - Project 300% over time (6 months instead of 2)

Second Example: Linguistic Databases

(Victor Rosenfeld, 2013)

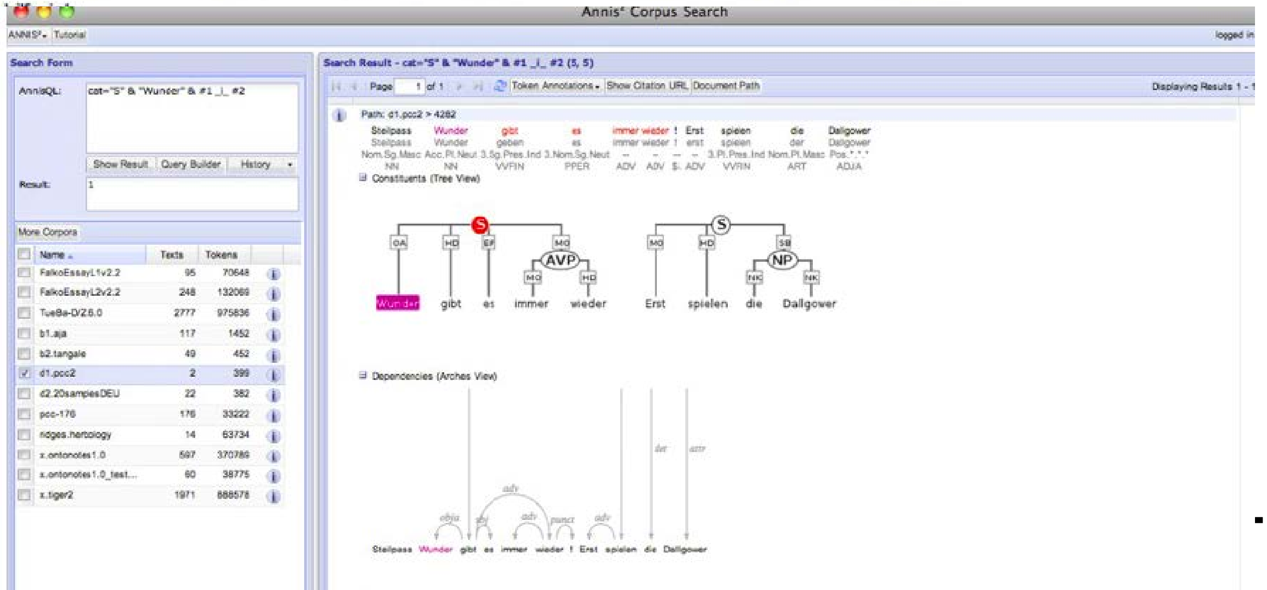
Wunder	gibt	es	immer wieder	!
Wunder	geben	es	immer wieder	!
Acc.Pl.Neut	3.Sg.Pres.Ind	3.Nom.Sg.Neut	-- --	--
NN	VVFIN	PPER	ADV ADV	$.



Semantic types

Steilpass Wunder gibt es immer wieder ! Erst spielen die Dallgower Gemeindevertreter so statisch und verzagt wie die deutsche Abwehrreihe der Fußballkicker . Und dann kommt aus der Tiefe solch ein fulminanter Steilpass , von dem man hofft , dass die Seeburger oder Groß-Glienicker Mitspieler ihn aufnehmen können . Ein Befreiungsschlag ist es allerdings nicht , weil es vorerst keine Gefahr fürs Dallgower Tor gab . Die Seeburger und einige Groß-Glienicker haben den Ball erst zurückgespielt und dann um so drängender wieder gefordert . Nun sollen sie zeigen , wie sie die Chance verwerten . Eine Diskussion , wo künftig die Trainerkabine stehen soll , wäre in der jetzigen Spielsituation verheerend . Und eine Parallele zu den deutschen Grotten-Kickern gibt es immer noch . Auch wenn die Spieler aus den verschiedenen Vereinen zusammengewürfelt sind , sie müssen sich daran gewöhnen , dass sie nun in einer Mannschaft " Döberitzer Heide " spielen . Und das heißt gemeinsam und nicht gegeneinander . Ermahnungen von der Seitenlinie , miteinander fair umzugehen und sich nicht beim kleinsten Schubser gegenseitig zu zerfleischen , sind normalerweise überflüssig . Vorerst allerdings

Cross-sentence links



Annis² Corpus Search

ANNIS²▾ Tutorial — logged in

Search Form

AnnisQL: cat="S" & "Wunder" & #1 _i_ #2

Show Result | Query Builder | History ▾

Result: 1

More Corpora

□	Name	Texts	Tokens	
□	FalkoEssayL1v2.2	95	70648	i
□	FalkoEssayL2v2.2	248	132069	i
□	TueBa-D/Z6.0	2777	975836	i
□	b1.aja	117	1452	i
□	b2.tangale	49	452	i
☑	d1.pcc2	2	399	i
□	d2.20samplesDEU	22	382	i
□	pcc-176	176	33222	i
□	ridges.herbology	14	63734	i
□	x.ontonotes1.0	597	370789	i
□	x.ontonotes1.0_test...	60	38775	i
□	x.tiger2	1971	888578	i

Search Result - cat="S" & "Wunder" & #1 _i_ #2 (5, 5)

Page 1 of 1 — Token Annotations ▾ | Show Citation URL | Document Path — Displaying Results 1 - 1

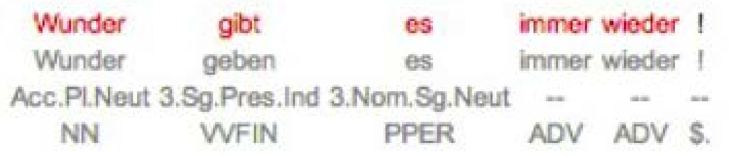
Path: d1.pcc2 > 4282

Steilpass	Wunder	gibt	es	immer wieder	!	Erst	spielen	die	Dallgower
Steilpass	Wunder	geben	es	immer wieder	!	erst	spielen	der	Dallgower
Nom.Sg.Masc	Acc.Pl.Neut	3.Sg.Pres.Ind	3.Nom.Sg.Neut	-- --	--	--	3.Pl.Pres.Ind	Nom.Pl.Masc	Pos.*.*.*
NN	NN	VVFIN	PPER	ADV ADV	$.	ADV	VVFIN	ART	ADJA

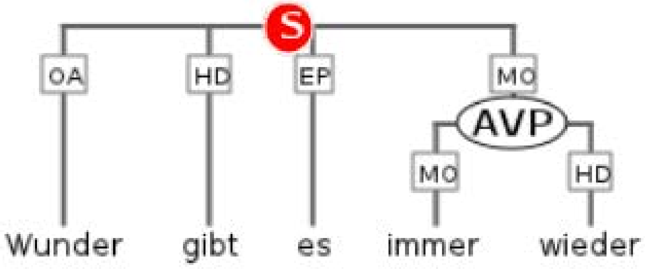
⊟ Constituents (Tree View)

⊟ Dependencies (Arches View)

AQL Queries

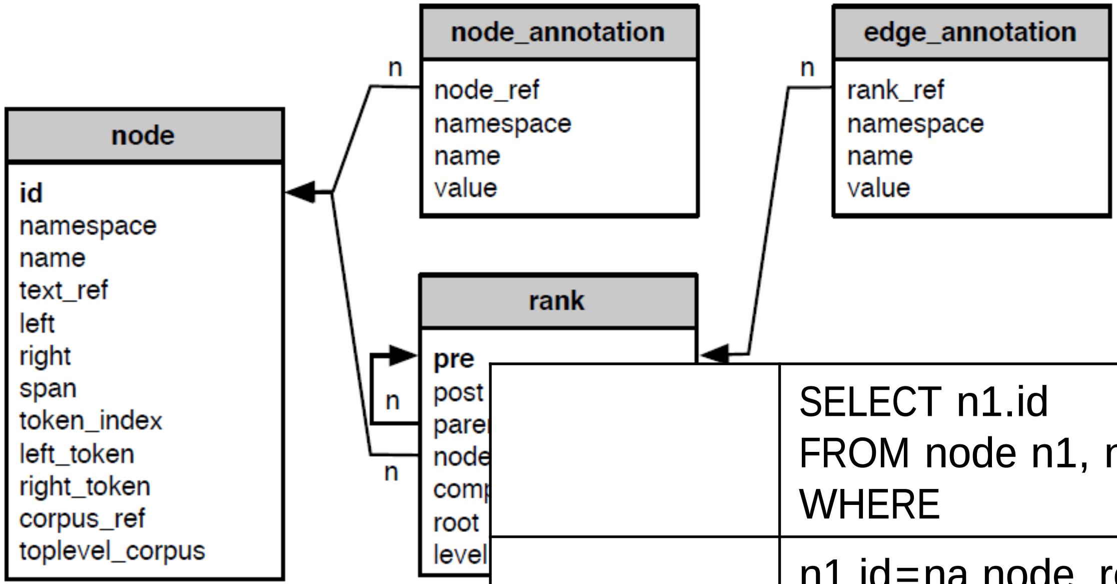


Wunder	gibt	es	immer	wieder	!
Wunder	geben	es	immer	wieder	!
Acc.Pl.Neut	3.Sg.Pres.Ind	3.Nom.Sg.Neut	--	--	--
NN	VVFIN	PPER	ADV	ADV	$.



cat=„S“ &	Find all sentences; bind to variable #1
„wunder“ &	Find all token „Wunder“; bind to variable #2
#1 _i_ #2	Join: remove #1 which do not include #2

Let‘s do it right - PostGreSQL

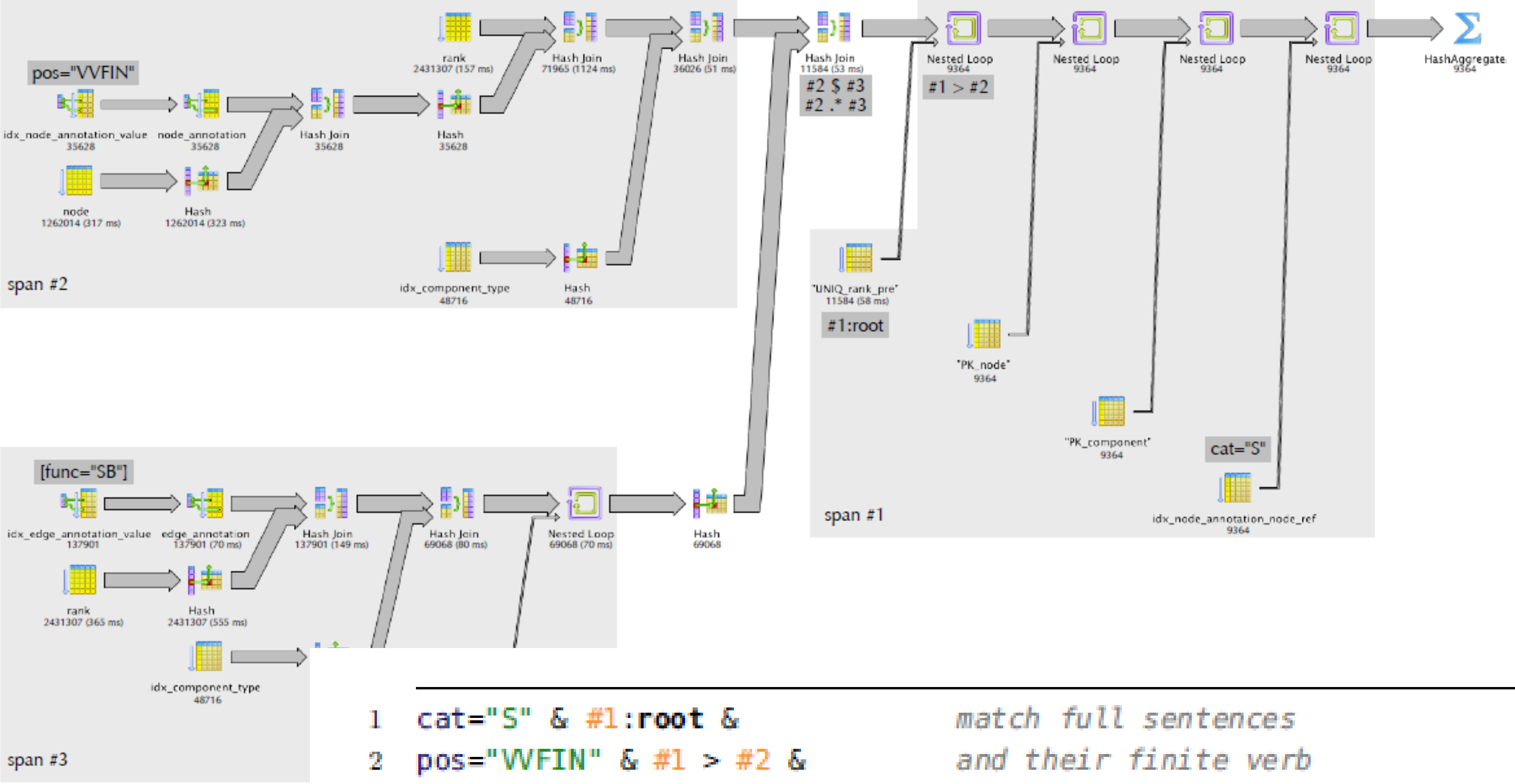


	SELECT n1.id
	FROM node n1, node n2, node_annotation na
	WHERE

cat=„S“ &	n1.id=na.node_ref AND
	na.name=„cat“ AND
	na.value=„S“ AND

„wunder“ &	n2.span=„Wunder“ AND
#1 _i_ #2	n1.text_ref=n2.text_ref AND n2.right<=n1.right AND n2.left>=n1.left

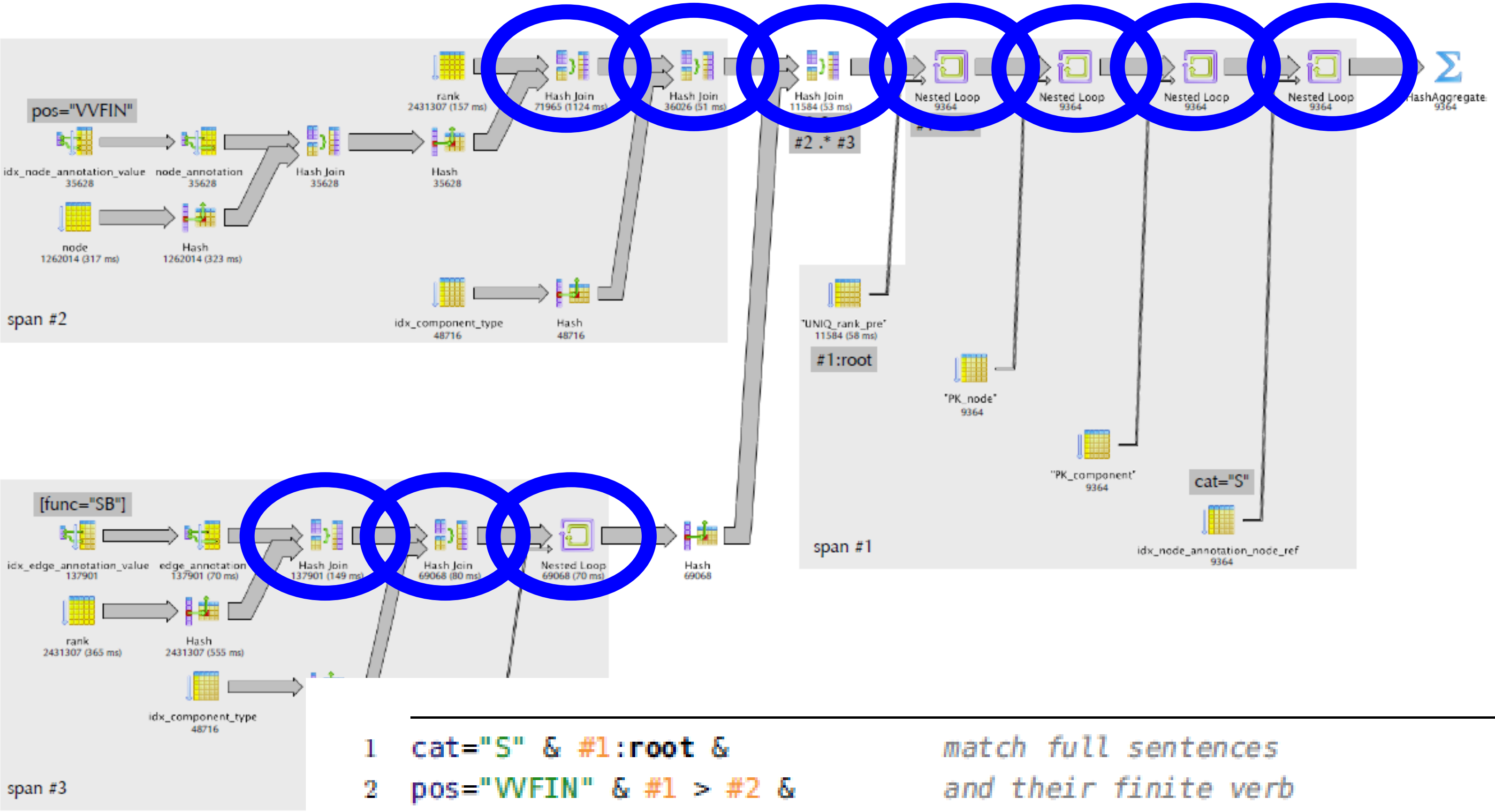
More Complicated Queries



```
cat="S" & #1:root &                  match full sentences
pos="VVFIN" & #1 > #2 &              and their finite verb
node & #1 >[func="SB"] #3 &          and their subject
#2 .* #3 &                           where the verb precedes the subject
meta::Genre="Politik"                only consider documents of the genre Politik
```

Listing 1: Annis query matching sentences in which the subject follows the verb.

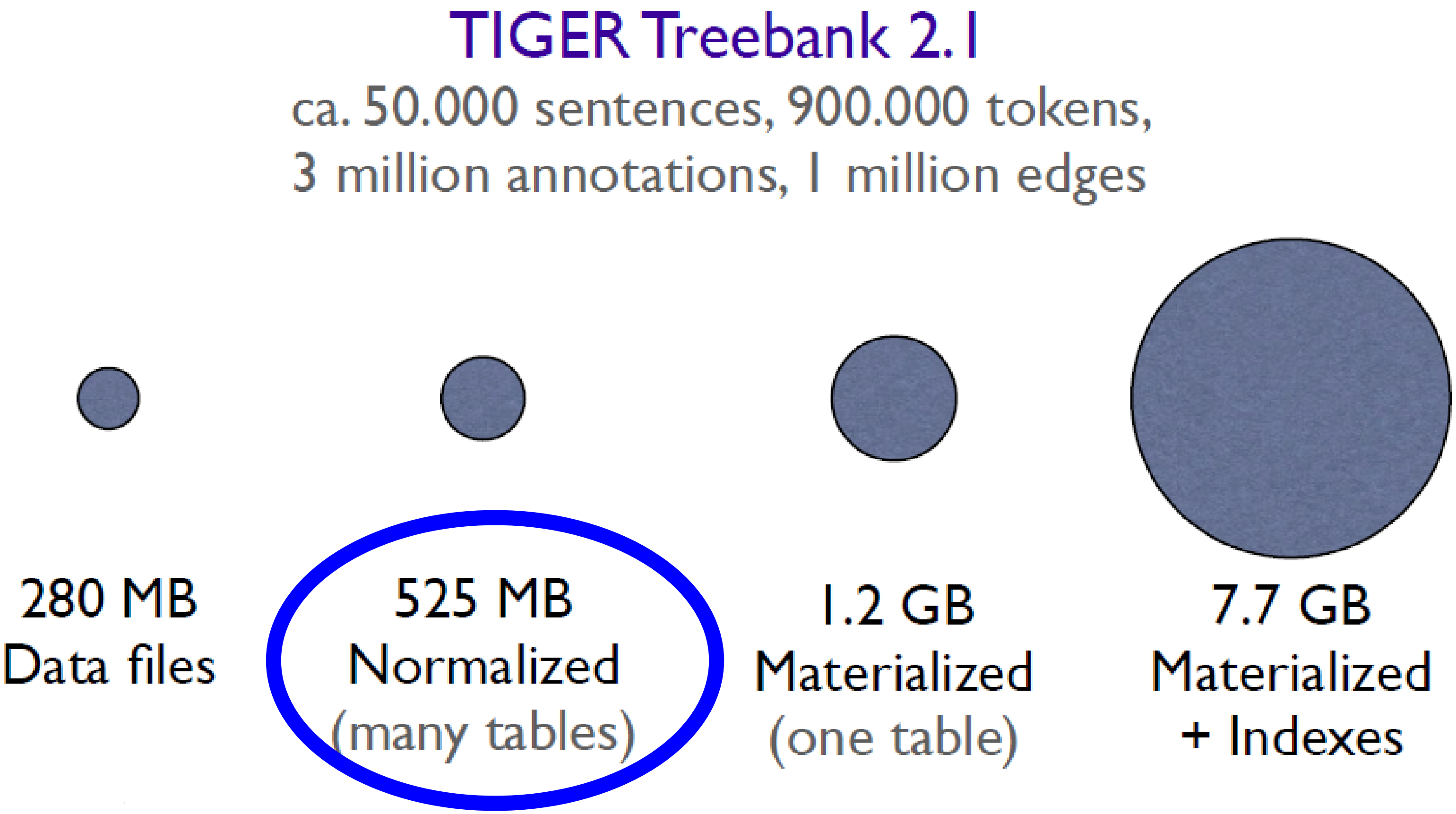
Did we do it right?



```
cat="S" & #1:root &                   match full sentences
pos="VVFIN" & #1 > #2 &               and their finite verb
node & #1 >[func="SB"] #3 &           and their subject
#2 .* #3 &                            where the verb precedes the subject
meta::Genre="Politik"                 only consider documents of the genre Politik
```

Listing 1: Annis query matching sentences in which the subject follows the verb.

RDBMS Feature: Indexes, Materialized Views



TIGER Treebank 2.1

ca. 50.000 sentences, 900.000 tokens,
3 million annotations, 1 million edges

280 MB
Data files

525 MB
Normalized
(many tables)

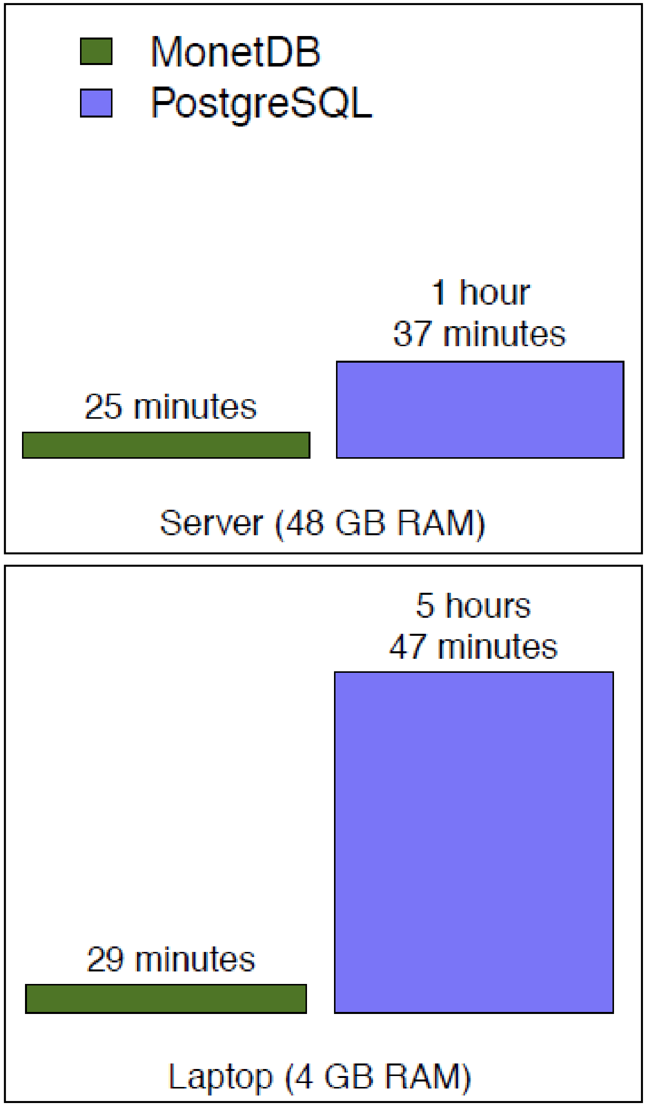
1.2 GB
Materialized
(one table)

7.7 GB
Materialized
+ Indexes

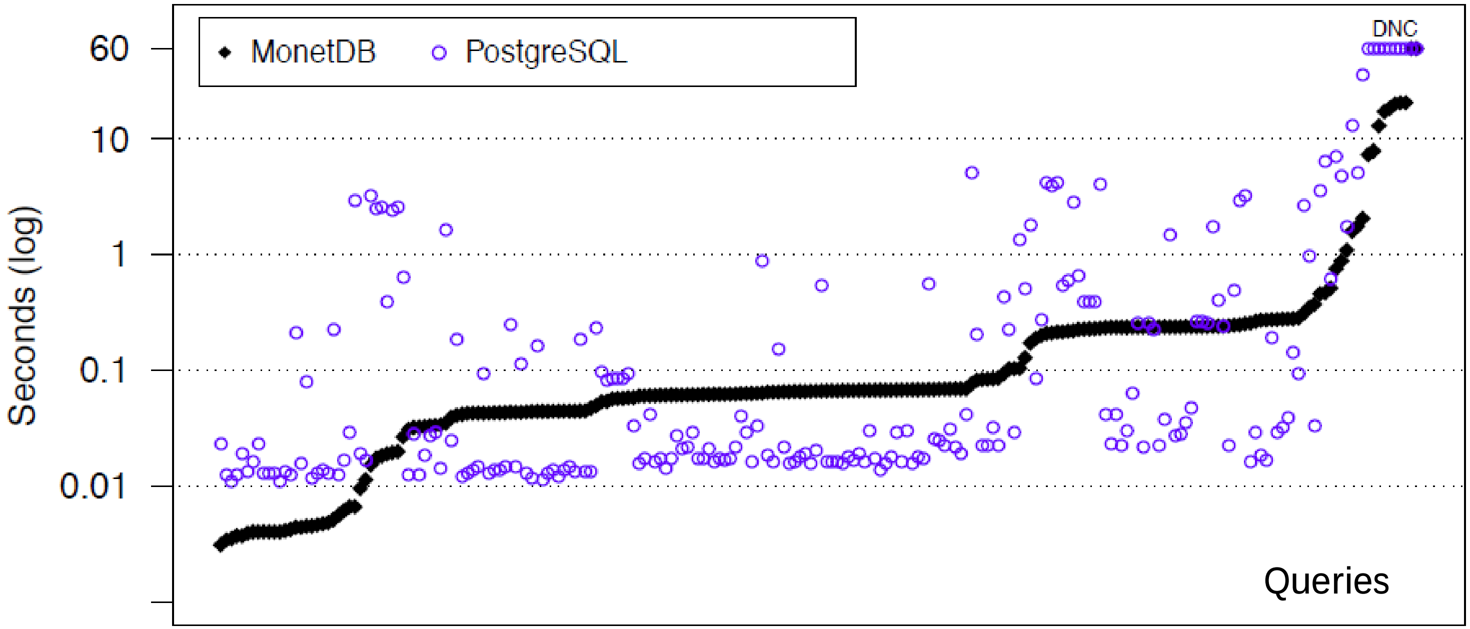
It is 2016 – we can keep this in memory!

MonetDB: A Main-Memory Column Store

- Workload: 330 real-life queries
- MonetDB is a RDBMS, but
  - All data kept in main memory
  - No indexes – all scans
  - Column store: Keep column values together, not tuples
- Advantages
  - No IO, buffering, caching, ...
  - Much better cache utilization for scans (outweighs missing indexes)
  - Column compression (memory, faster scans)
- Still relational: Many joins

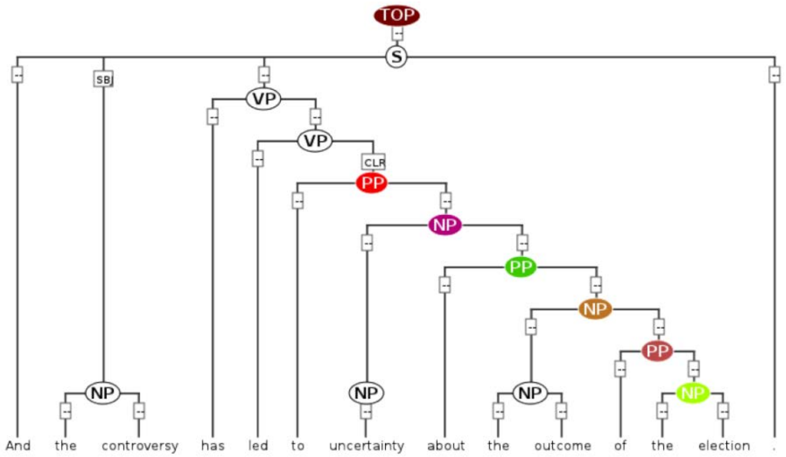


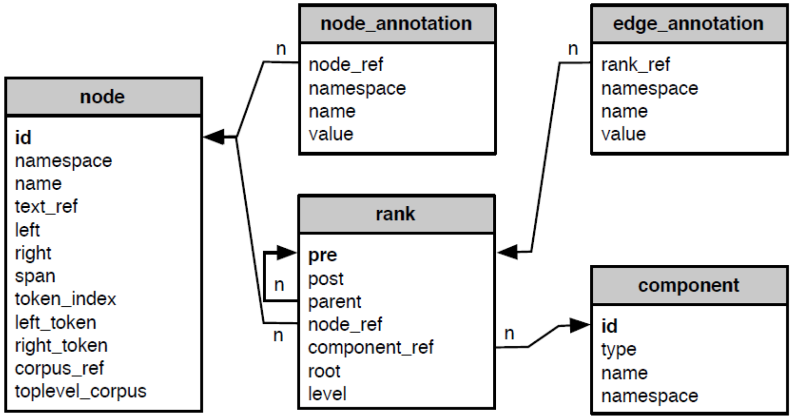
Query Optimization is Difficult to Predict



- PostGreSQL is faster for many queries despite IO
- But if it is slower, it is much slower (log scale)

Even Better: A Graph Store





node
id
namespace
name
text_ref
left
right
span
token_index
left_token
right_token
corpus_ref
toplevel_corpus

node_annotation
node_ref
namespace
name
value

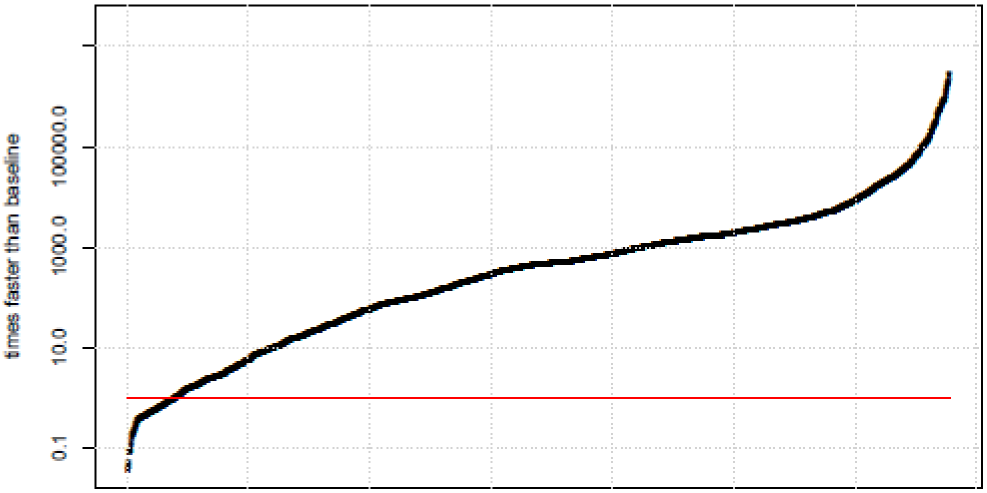
edge_annotation
rank_ref
namespace
name
value

rank
pre
post
parent
node_ref
component_ref
root
level

component
id
type
name
namespace

- AQL queries navigate through graphs
- Relational: One join for (almost) every edge traversal
- graphANNIS: AQL on a main-memory graph data model
  - No joins, but following pointers
    - Implemented as indexes into arrays
  - Indexes to find the right nodes quickly (to start traversals)

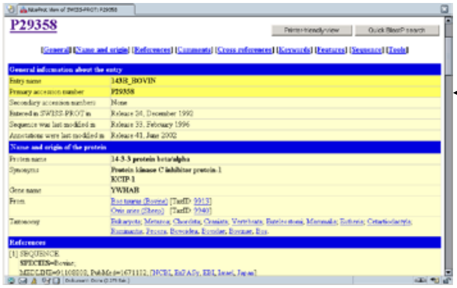
Thomas Krause (submitted)



- Workload: ~3300 real-life queries against ~20 corpora
- graphANNIS versus PostGreSQL
  - ~40 times faster on the entire workload
  - Faster for 97% of all queries
  - Not much slower for the remaining 3%

Databases are Infrastructure

Classical Three-Tier Architecture



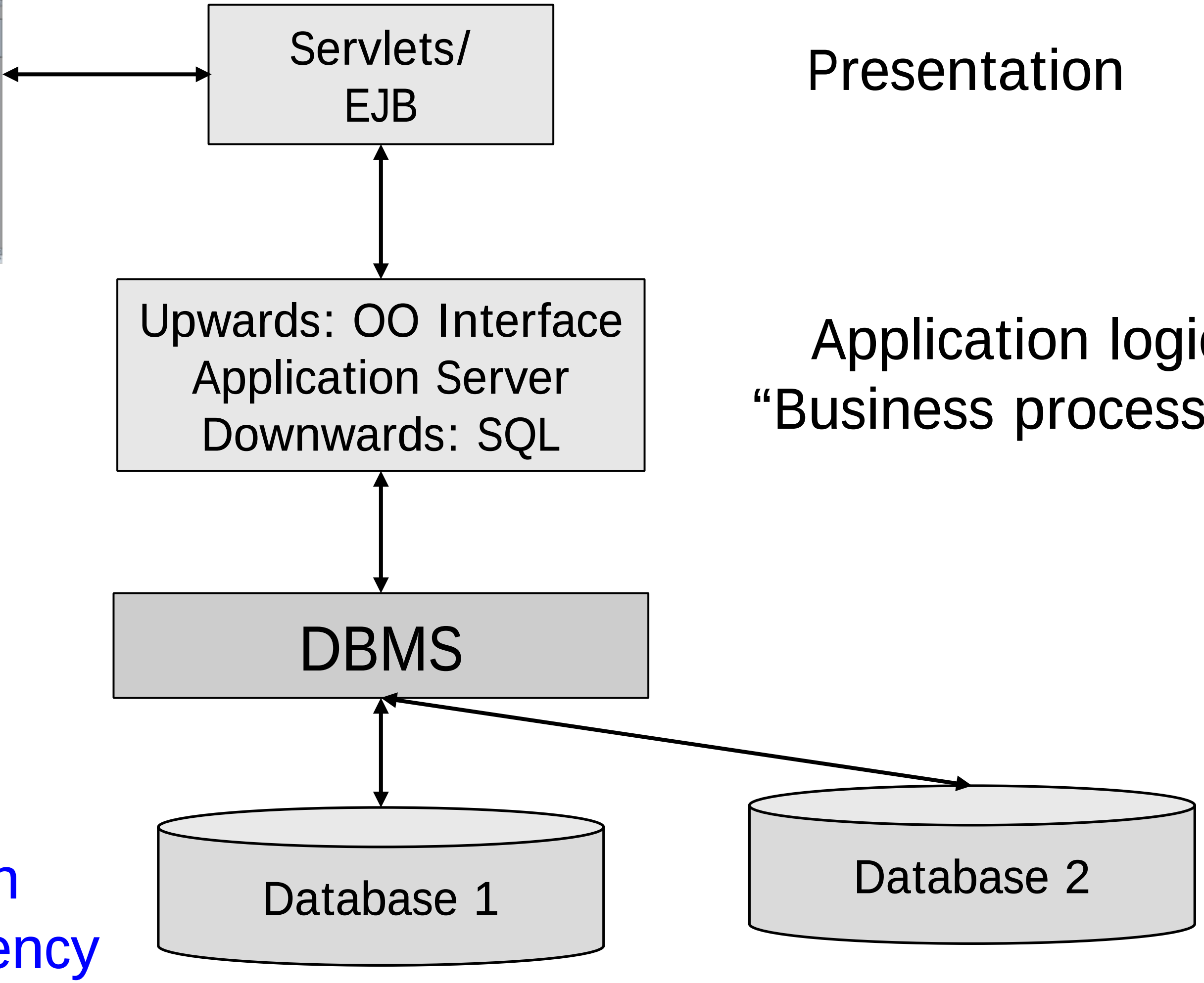
P29358

[General] [Name and origin] [References] [Comments] [Cross-references] [Keywords] [Features] [Sequence] [Tools]

General information about the entry	
Entry name	143B_BOVIN
Primary accession number	P29358
Secondary accession numbers	None
Entered in SWISS-PROT in	Release 24, December 1992
Sequence was last modified in	Release 33, February 1996
Annotations were last modified in	Release 41, June 2002
Name and origin of the protein	
Protein name	**14-3-3 protein beta/alpha**
Synonyms	**Protein kinase C inhibitor protein-1** **KCIP-1**
Gene name	**YWHAB**
From	Bos taurus (Bovine) [TaxID: 9913] Ovis aries (Sheep) [TaxID: 9940]
Taxonomy	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Cetartiodactyla; Ruminantia; Pecora; Bovoidea; Bovidae; Bovinae; Bos.

References

[1] SEQUENCE
SPECIES=Bovine;
MEDLINE=91108000; PubMed=1671112; [NCBI, ExPASy, EBI, Israel, Japan]



Presentation

Application logic
“Business processes”

“State”
Storage, Search
Recovery, Consistency

Today’s Database Systems

- RDBMS are essential parts of enterprise infrastructures
  - More important than OS
  - Long-running, expensive, essential investment
  - Holds the most important business assets: Data, information
- Database administrator is a well-paid profession
  - Developers write SQL & business logic
  - Admins make SQL run fast
  - Many programmers, fewer DB developers, very few DB admins
  - A skills much demanded in industry
- RDMS became an often “invisible” piece of software
  - “So nützlich wie fließendes Wasser” (G. Weikum, MPI Saarbrücken)

Main Features

- Data needs to be stored
  - Disk access (or cache utilization) is the main bottleneck
  - Hence: Minimize access time -> minimize disk access
- Data is manipulated by many clients
  - Concurrent access quickly screws up data
  - Hence: Synchronize access
- Data is used by many apps with different requirements
  - No good to design application specific “optimal” data structures
  - Hence: Use application independent languages and models
- Systems crash
  - Crashes cannot be avoided
  - Hence: Protect consistency by logging, constraint enforcement, ...

DBS2: Implementation of Database Systems

- Lecture 4 SWS
  - Wednesday, 11 – 13 , 3.113
  - Thursday, 11 – 13 , 3.113

- Contact

  Ulf Leser

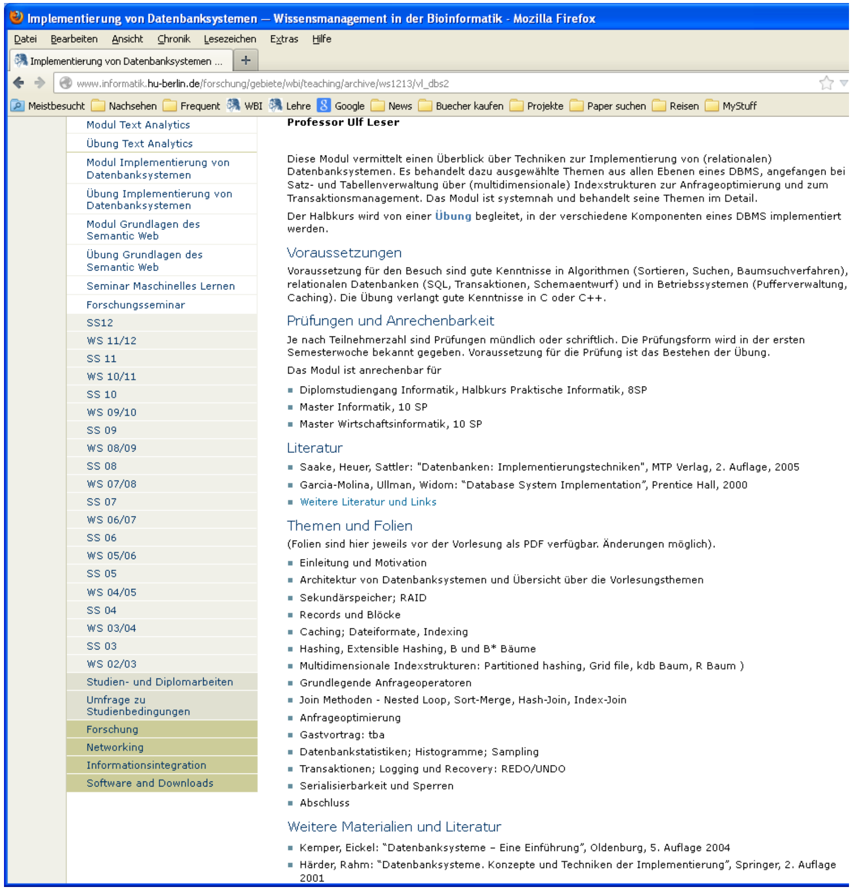
Room:	IV.401
Tel:	(030) 2093 – 3902
Mail:	leser (at) informatik.hu-berlin.de

Exercises and Examination

- Exercises run by Jörg Bachmann (DBIS)
  - Presence & commitment are necessary
  - Implementation of file-/ buffer-/ index manager in C++
  - Wednesday, 13-15, 3.113
  - Starts today (probably not!)
- Examination
  - Oral or written?
  - Oral exam dates will be set mid-January
  - Admission: Passing the exercises

Slides

- Slides are available shortly after the lecture
- Please send me any errors
- Slides are
  - not a script
  - no substitution for listening to the lecture
  - no substitution for reading a book


Implementierung von Datenbanksystemen — Wissensmanagement in der Bioinformatik - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Implementierung von Datenbanksystemen ...
www.informatik.hu-berlin.de/forschung/gebiete/wbi/teaching/archive/ws1213/vl_dbs2
Meistbesucht Nachsehen Frequent WBI Lehre Google News Buecher kaufen Projekte Paper suchen Reisen MyStuff
Modul Text Analytics
Übung Text Analytics
Modul Implementierung von Datenbanksystemen
Übung Implementierung von Datenbanksystemen
Modul Grundlagen des Semantic Web
Übung Grundlagen des Semantic Web
Seminar Maschinelles Lernen
Forschungsseminar
SS12
WS 11/12
SS 11
WS 10/11
SS 10
WS 09/10
SS 09
WS 08/09
SS 08
WS 07/08
SS 07
WS 06/07
SS 06
WS 05/06
SS 05
WS 04/05
SS 04
WS 03/04
SS 03
WS 02/03
Studien- und Diplomarbeiten
Umfrage zu Studienbedingungen
Forschung
Networking
Informationsintegration
Software and Downloads

Professor Ulf Leser

Diese Modul vermittelt einen Überblick über Techniken zur Implementierung von (relationalen) Datenbanksystemen. Es behandelt dazu ausgewählte Themen aus allen Ebenen eines DBMS, angefangen bei Satz- und Tabellenverwaltung über (multidimensionale) Indexstrukturen zur Anfrageoptimierung und zum Transaktionsmanagement. Das Modul ist systemnah und behandelt seine Themen im Detail.

Der Halbkurs wird von einer **Übung** begleitet, in der verschiedene Komponenten eines DBMS implementiert werden.

Voraussetzungen

Voraussetzung für den Besuch sind gute Kenntnisse in Algorithmen (Sortieren, Suchen, Baumsuchverfahren), relationalen Datenbanken (SQL, Transaktionen, Schemaentwurf) und in Betriebssystemen (Pufferverwaltung, Caching). Die Übung verlangt gute Kenntnisse in C oder C++.

Prüfungen und Anrechenbarkeit

Je nach Teilnehmerzahl sind Prüfungen mündlich oder schriftlich. Die Prüfungsform wird in der ersten Semesterwoche bekannt gegeben. Voraussetzung für die Prüfung ist das Bestehen der Übung.

Das Modul ist anrechenbar für

- Diplomstudiengang Informatik, Halbkurs Praktische Informatik, 8SP
- Master Informatik, 10 SP
- Master Wirtschaftsinformatik, 10 SP

Literatur

- Saake, Heuer, Sattler: "Datenbanken: Implementierungstechniken", MTP Verlag, 2. Auflage, 2005
- Garcia-Molina, Ullman, Widom: "Database System Implementation", Prentice Hall, 2000
- Weitere Literatur und Links

Themen und Folien

(Folien sind hier jeweils vor der Vorlesung als PDF verfügbar. Änderungen möglich).

- Einleitung und Motivation
- Architektur von Datenbanksystemen und Übersicht über die Vorlesungsthemen
- Sekundärspeicher; RAID
- Records und Blöcke
- Caching; Dateiformate, Indexing
- Hashing, Extensible Hashing, B und B* Bäume
- Multidimensionale Indexstrukturen: Partitioned hashing, Grid file, kdb Baum, R Baum)
- Grundlegende Anfrageoperatoren
- Join Methoden - Nested Loop, Sort-Merge, Hash-Join, Index-Join
- Anfrageoptimierung
- Gastvortrag: tba
- Datenbankstatistiken; Histogramme; Sampling
- Transaktionen; Logging und Recovery: REDO/UNDO
- Serialisierbarkeit und Sperren
- Abschluss

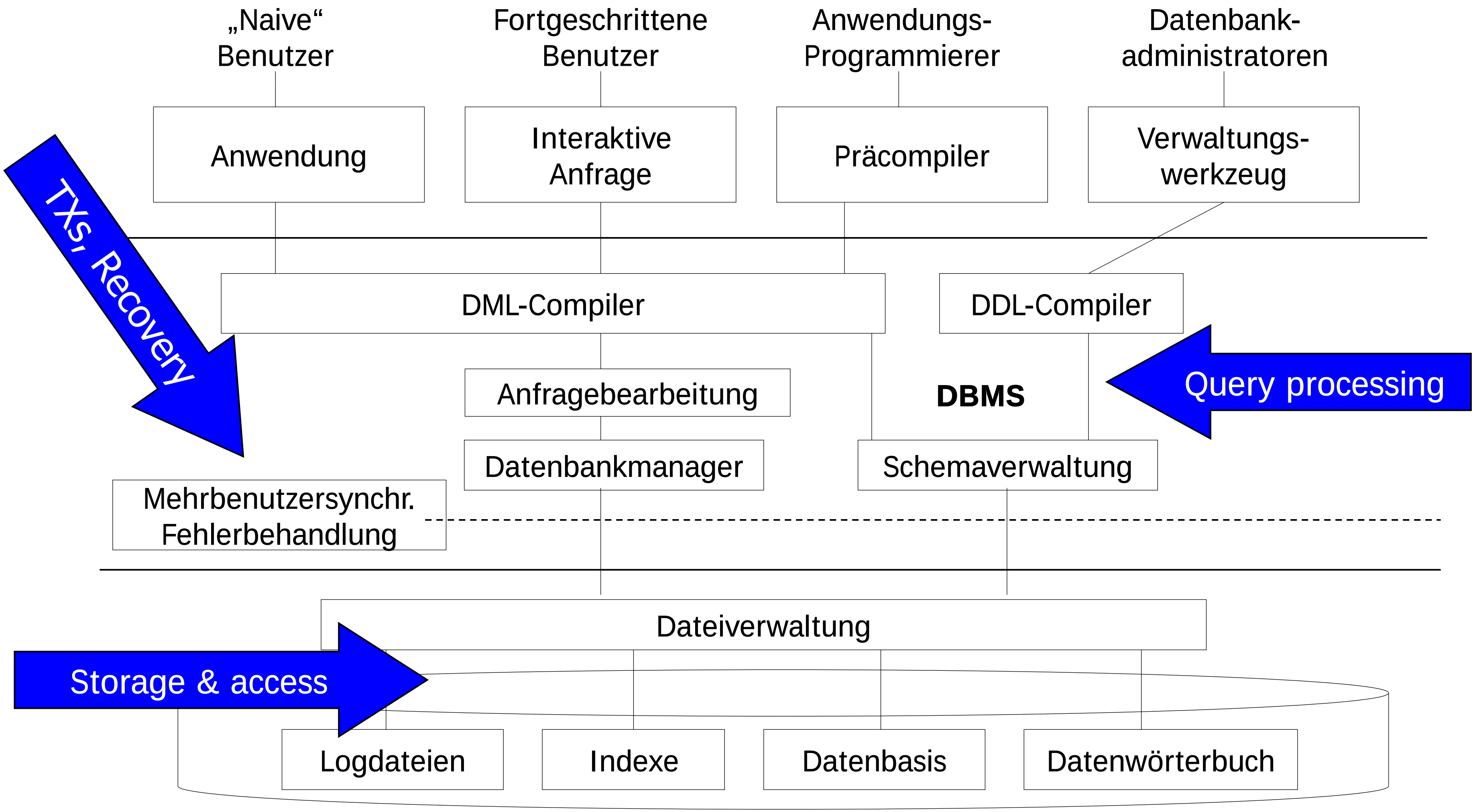
Weitere Materialien und Literatur

- Kemper, Eickel: "Datenbanksysteme – Eine Einführung", Oldenburg, 5. Auflage 2004
- Härder, Rahm: "Datenbanksysteme. Konzepte und Techniken der Implementierung", Springer, 2. Auflage 2001

Literature

- Primary
  - Saake, Heuer, Sattler “Datenbanken: Implementierungstechniken”, mitp Verlag, 2005 (2. Auflage)
  - Garcia-Molina, Ullman, Widom: “Database System Implementation”, Prentice Hall, 2000
- Other
  - Kemper, Eickel: “Datenbanksysteme – Eine Einführung”, Oldenburg, 5. Auflage 2004
  - Härder, Rahm: “Datenbanksysteme. Konzepte und Techniken der Implementierung”, Springer, 2. Auflage 2001
  - R. Elmasri und S.B. Navathe: Fundamentals of Database Systems, Benjamin Cummings
    - Deutsche Übersetzung: „Grundlagen von Datenbanksystemen“, Pearson, 2002

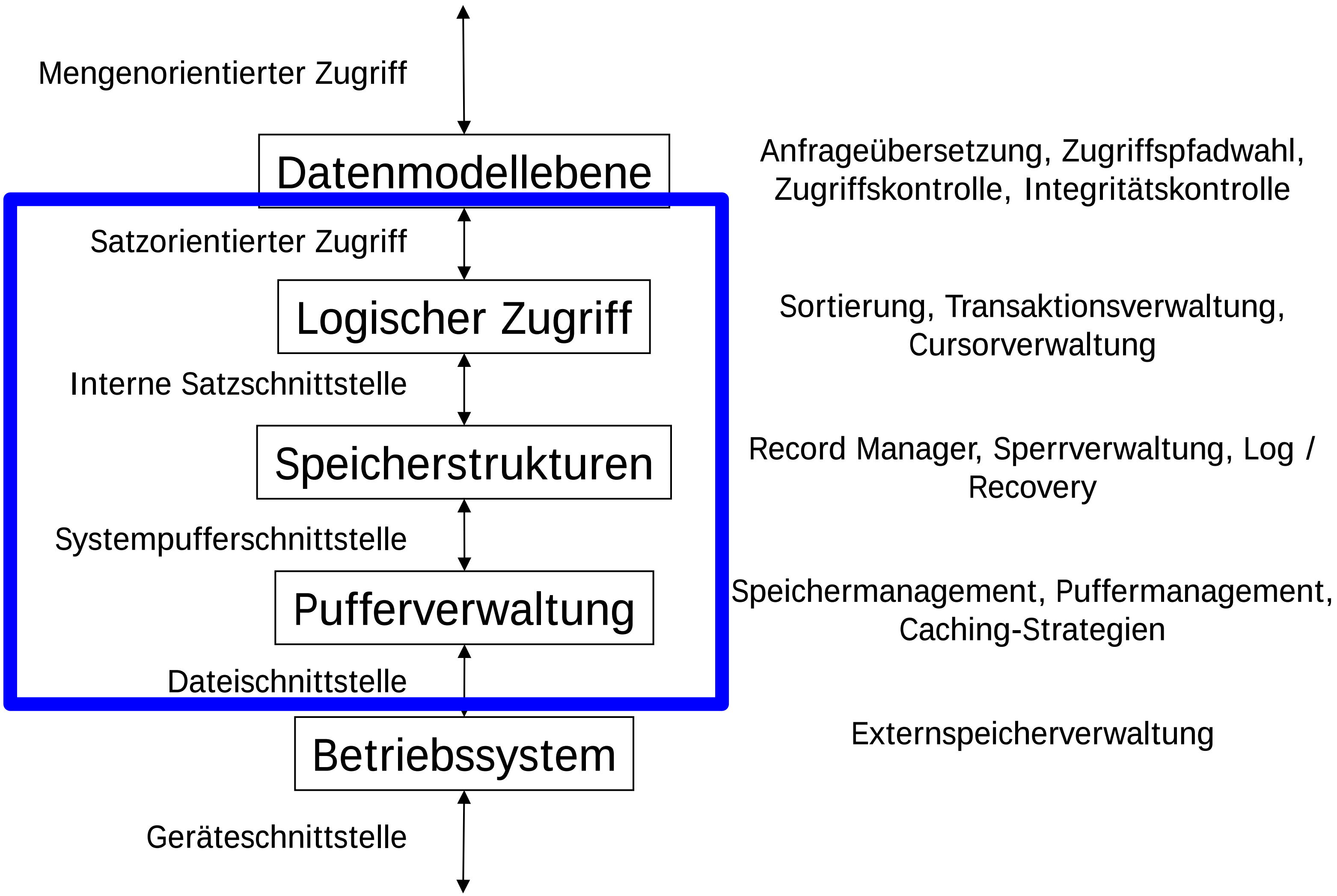
Überblick


„Naive“ Benutzer
Fortgeschrittene Benutzer
Anwendungs-Programmierer
Datenbank-administratoren
Anwendung
Interaktive Anfrage
Präcompiler
Verwaltungs-werkzeug
TXs, Recovery
DML-Compiler
DDL-Compiler
Query processing
Anfragebearbeitung
DBMS
Datenbankmanager
Schemaverwaltung
Mehrbenutzersynchr. Fehlerbehandlung
Dateiverwaltung
Storage & access
Logdateien
Indexe
Datenbasis
Datenwörterbuch

Contents

- Introduction
- Overview and architecture
- Storage and access methods
  - B*-Trees, Extensible hashing, index-sequential files ...
  - Multidimensional indexing: Grid-files, kd-Trees, R-Trees ...
- Query processing and optimization
  - Physical relational operators
  - Cost-based optimization
- Recovery
- Transactions and concurrency control

5 Schichten Architektur



Mengenorientierter Zugriff

Datenmodellebene

Anfrageübersetzung, Zugriffspfadwahl, Zugriffskontrolle, Integritätskontrolle

Satzorientierter Zugriff

Logischer Zugriff

Sortierung, Transaktionsverwaltung, Cursorverwaltung

Interne Satzschnittstelle

Speicherstrukturen

Record Manager, Sperrverwaltung, Log / Recovery

Systempufferschnittstelle

Pufferverwaltung

Speichermanagement, Puffermanagement, Caching-Strategien

Dateischnittstelle

Betriebssystem

Externspeicherverwaltung

Geräteschnittstelle

Guests

- TBA ...

Feedback 2012 / 2013

Alter	Geschlecht	Gefehlt	Teilnehmerzahl	Warum kommen?	Studiengang	Fachsemester	Freundlich	Fragen	Sprache	Präsentation	Beispiele	Konzeption	Überblick	Viel neues	Kritische Auseinandersetzung	Nützlich	Lernziele	Materialien	Tempo	Schwierigkeit	Arbeitsaufwand	Dozent	Vorlesung	Abweichung vom Optimum
21	M	1	3	2	BA	5	6	5	6	5	6	5	5	6	4	6	5	5	3	3	3			**8**
28	M	0	3	2	?	5	6	6	4	6	6	6	6	6	5			5	3	3	3	1	1	**4**
35	M	4	3	2	DI	8	6	6	6	4	6	6	6	6	5	6	6		3	3	4	1	1	**4**
25	M	5	3	2	DI	11	6	6	6	4	6	5	5	6	4	5	5	5	3	3	3	1	1	**9**
35	M	2	3	2	?	6	6	6	6	6	6	6	6	5	6	5	5		2	2	3	1	1	**5**
24	M	0	3	1,2,3	DI	9	6	6	6	6	6	6	6	6	6	6	6	6	3	3	3	1	1	**0**
25	M	1	3	2,3	MW	3	6	6		5	6	5	6	6	6	6	6	5	3	3	3	1	1	**3**
22	M	4	3	2	MI	1	6	5	6	5	5	6	6	4	4	5			3	4	3	1	2	**10**
24		5	2	2	MI	1	6	6	6	5	5	5	6	6	5	5	6		3	4	3	1	1	**6**
22	M	0	2	2	MI	1	6	6	5	6	6	6	6	5		6	6	4	3	4	3	1	1	**5**
24	M	0	3	2	MI	1	6	5	6	5	5	6	6	5	4	5	6	4	2	3	3	1	2	**11**
25,9		**2,0**	**2,9**			**4,6**	**6,00**	**5,73**	**5,70**	**5,18**	**5,67**	**5,67**	**5,82**	**5,55**	**5,14**	**5,44**	**5,63**	**4,86**	**2,82**	**3,18**	**3,09**	**1,00**	**1,20**	
			3,0				**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**6,00**	**3,00**	**3,00**	**3,00**	**1,00**	**1,00**	
			0,10				**0,00**	**0,27**	**0,30**	**0,82**	**0,33**	**0,33**	**0,18**	**0,45**	**0,86**	**0,56**	**0,38**	**1,14**	**0,18**	**-0,18**	**-0,09**	**0,00**	**-0,20**	
							0,00	0,27	0,30	0,82	0,33	0,33	0,18	0,45	0,86	0,56	0,38	1,14	0,18	0,18	0,09	0,00	0,20	6,28

Free Text

- **Besonders gut**
- 5 Auftreten Dozent
- 4 Small Talk am Anfang
- 5 Beispiele
- Deutsche Vorlesung
- Engagement Dozent
- 2 Gastvorträge
- 2 Konkreter Anwendungsbezug
- Übung
- Tempo

- **Verbesserung**
- Struktur Folien
- Tafelbild zu unklar
- Warum nur Oracle/MySQL
- Liegt zeitlich schlecht
- Fehlt: PL/SQL, UDF, ...
- Entweder nur vormittags oder nur nachmittags
- Praktikum nicht schwer, aber aufwändig
- Übung enger mit Vorlesung verbinden

Datenbanken@Informatik

- A predefined focus area in our Master

- Datenbanken 1: Grundlagen (BA)
- Information Retrieval (BA)
- Datenbanken 2: Implementierung
- Data Warehousing und Data Mining
- Informationsintegration (inkl. verteilter Anfrageoptimierung)
- Neue Konzepte und Techniken für Datenbanksysteme
- Techniken und Konzepte zum Schutz der Privatsphäre
- Datenbanktheorie
- Frequent seminars