

Grundlagen der Datenbanksysteme 2 (M-DB2)

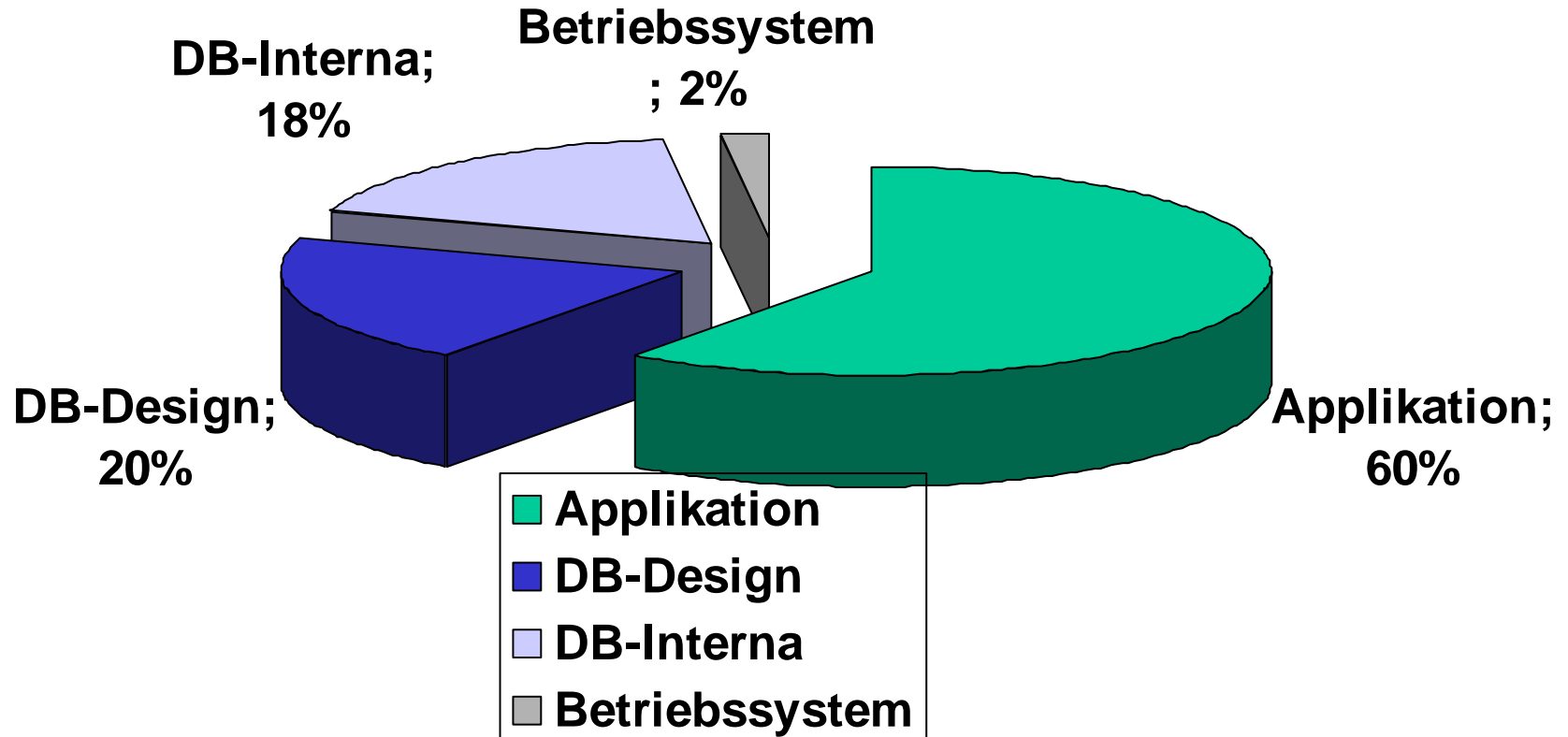
Dr. Karsten Tolle

Vorwissen und so ...

- SQL
- Umgang mit MySQL (Workbench)

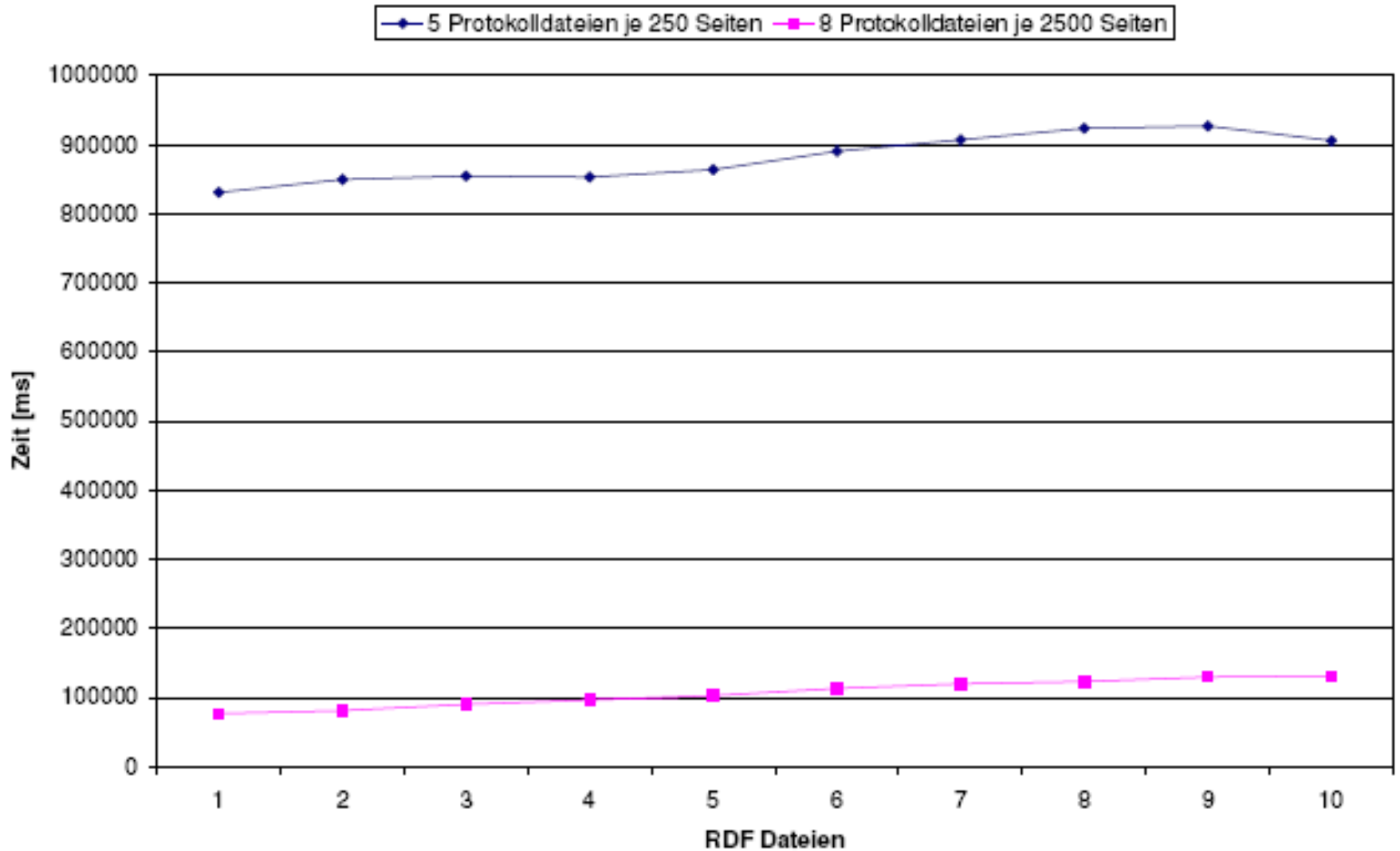
- Beispieldaten zum Spielen:
<http://download.geonames.org/export/dump/>

Tuningpotential



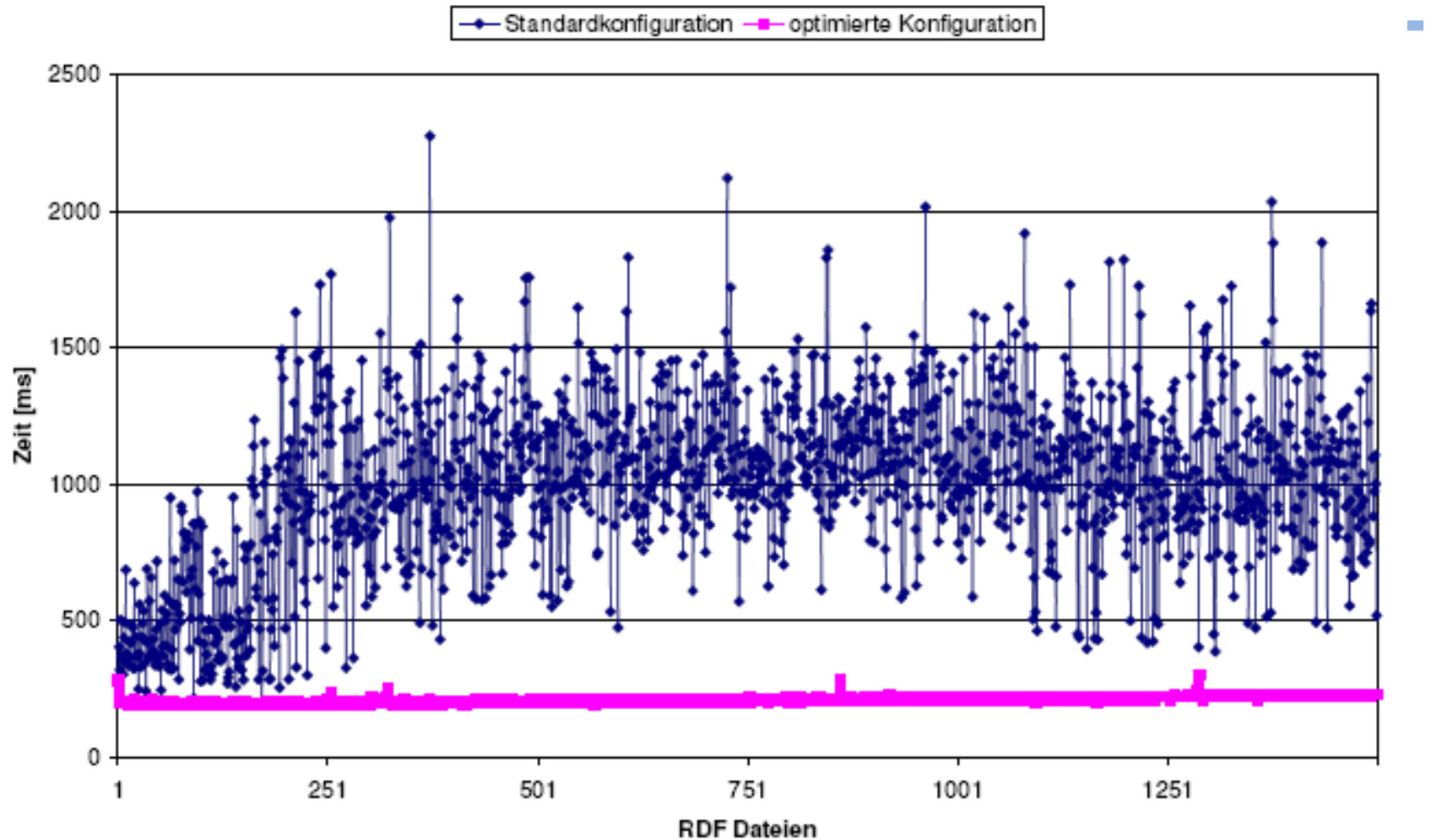
Werte aus: Jörg Frize u. Jürgen Marsch – Erfolgreiche Datenbankentwicklung mit SQL3
ISBN: 3-528-55210-7

Logging Einstellungen



Performance Optimierung der Nutzung von RDF-S3 Daten am Beispiel von IBMs DB2 v8.1
Diplomarbeit von Jashar Rexhepi (2005)

Beispiel für Optimierung



Performance Optimierung der Nutzung von RDF-S3 Daten am Beispiel von IBMs DB2 v8.1
Diplomarbeit von Jashar Rexhepi (2005)

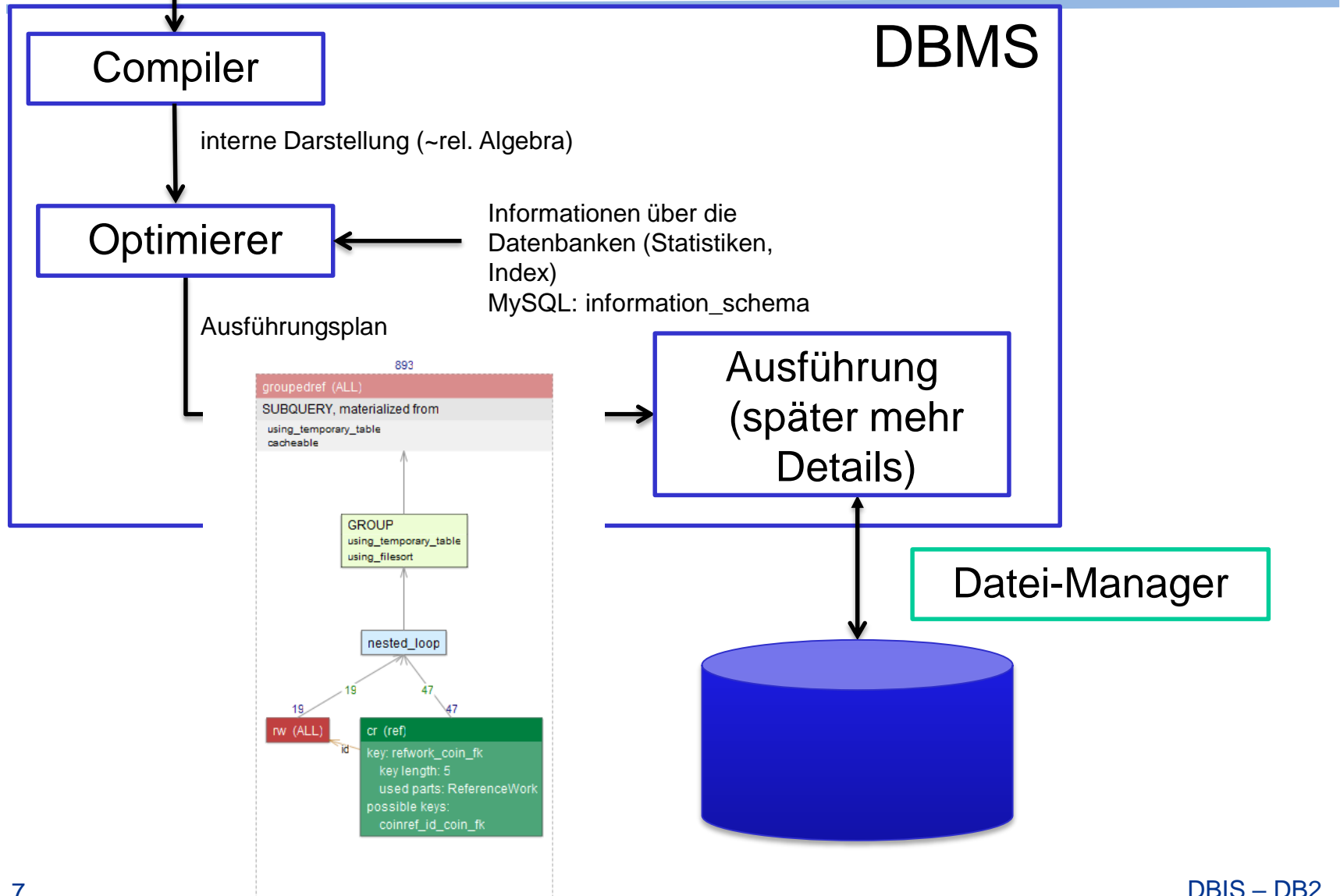
Was passiert hier eigentlich?

select count(*) from admindivision_4; → 44895

select * from admindivision_4;

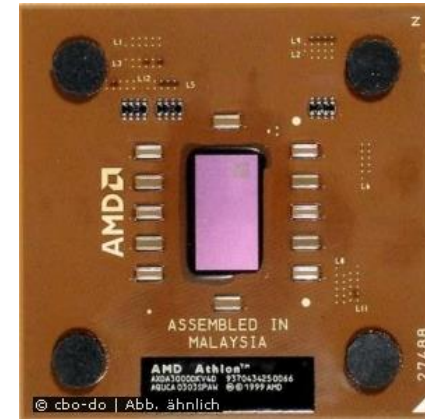
id	Name	Name_long	Description	Latitude	Longitude
DE_01_001_000_43257	Flensburg	Flensburg - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43258	"Flensburger Förde"	"Flensburger Förde" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43259	Tarup	Tarup - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43260	Sünderup	Sünderup - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43261	Wees	Wees - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43262	"Flensburg A"	"Flensburg A" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43263	"Flensburg B"	"Flensburg B" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43264	"Flensburg C"	"Flensburg C" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43265	"Flensburg D"	"Flensburg D" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43266	"Flensburg E"	"Flensburg E" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43267	"Flensburg F"	"Flensburg F" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43268	"Flensburg G"	"Flensburg G" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43269	"Flensburg H"	"Flensburg H" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43270	"Flensburg J"	"Flensburg J" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43271	"Flensburg K"	"Flensburg K" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43272	"Flensburg L"	"Flensburg L" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL
DE_01_001_000_43273	"Flensburg M"	"Flensburg M" - Flensburg - Flensburg, Stadt (DE)	NULL	NULL	NULL

SELECT * FROM afergk.groupedref;



CPU – z.B. AMD Geode 1750NX 1400 MHz

- **Hersteller** AMD
- **Bezeichnung** Athlon XP 1750+ Geode
- **Taktfrequenz** 1400 MHz
- **Level 1 Cache** 128 KB
- **Level 2 Cache** 256 KB
- **Bustakt** 266 MHz



• RAM – z.B. 1GB G-Skill PC3200/400 CL 3

- **Speichergeschwindigkeit** 400 MHz (PC3200)



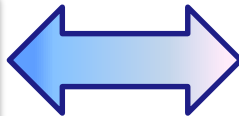
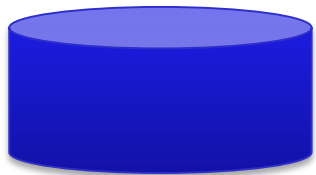
• Festplatte – z.B. WD 1TB SATA 3 8,9cm (3,5")

- **Spindelgeschwindigkeit** 7200 rpm
- **Datenübertragungsrate** 600 MBps
- **Cache-Größe** 32 MB
- **Durchschnittliche Latenzzeit** 4,20 ms (Nennwert)

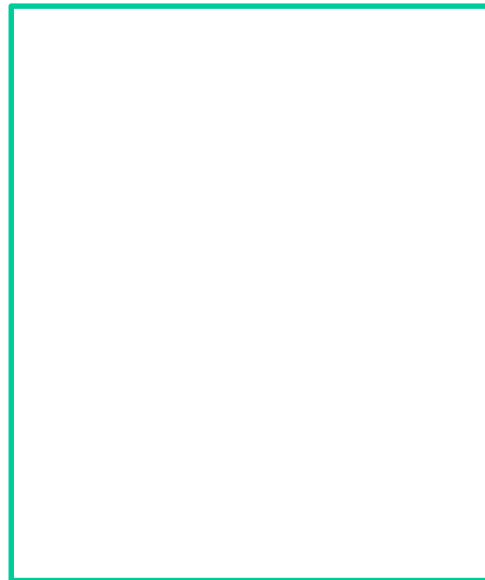




RAM



Austauscheinheit: ein Block



Adressen

- Der Hauptspeicher besteht aus einer festen Anzahl von **gleichgroßen Speicherzellen**, die jeweils Informationen speichern können.
- Die Speicherzellen sind **fortlaufend nummeriert**.
- Über die Nummer (**Adresse**) einer Speicherzelle kann deren **Inhalt** abgerufen oder verändert werden.

Adresse	Speicherzelle
0	00101101
1	01100001
2	00110010
...	...
...	...
n-1	10000101

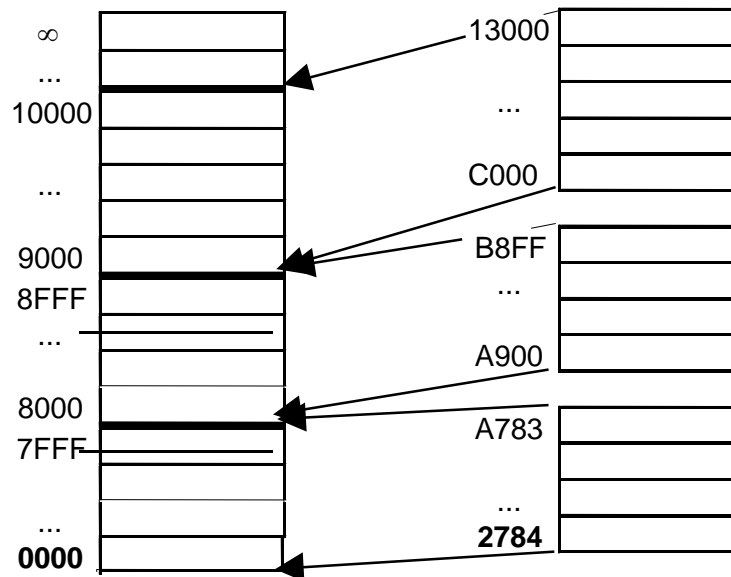
Virtueller Speicher

Forderung: *lineares Programmiermodell*

- Die Fragmente sollen **zusammenhängend** erscheinen.
- Es steht **beliebig viel** Speicherplatz zur Verfügung.

Virtueller Speicher:

- beliebig lang
- durchgehend



Physikalischer Speicher:

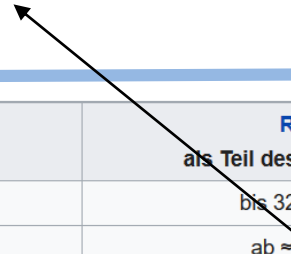
- fragmentiert
- endlich

→ Realisierung durch die **Memory Management Unit (MMU)**

	MLC-NAND-Flash-Laufwerk 1,0" bis 3,5"	RAM-Disk als Teil des Arbeitsspeichers	Festplatte 1,0" bis 3,5"
Größe (keine Raidlaufwerke)	bis 1 TB	bis 16 GB je Modul	bis 4 TB
Preis pro GB (Stand Mai 2013)	ab ≈ 0,55 € ^[8]	ab ≈ 4,5 € ^[69]	ab ≈ 0,044 € ^[9]
Anschluss	S-ATA, P-ATA, mSATA, PCIe	hauptsächlich DIMM-Connector	S-ATA, P-ATA, SCSI, SAS
Lesen (kein RAID)	bis 510 MB/s ^[70]	bis 51200 MB/s ^[71]	bis ca. 160 MB/s ^[72]
Schreiben (kein RAID)	bis 490 MB/s ^[70]	bis 51200 MB/s ^[71]	bis ca. 160 MB/s ^[72]
Mittlere Zugriffszeit lesen	0,2 ms	0,00002 ms	ab 3,5 ms
Mittlere Zugriffszeit schreiben	0,4 ms	0,00002 ms	ab 3,5 ms
Überschreibbar	3 bis 10 Tausend Mal (MLC)	„beliebig“	„beliebig“
Lagerbar bei	-45...85 °C	-25...85 °C	-40...70 °C
Stoßfestigkeit – Betrieb ^[73]	1500 g	?	60 g
Stoßfestigkeit – Lagerung	1500 g	?	350 g
Verbrauch – Ruhe	0,1...1,3 W	1 W pro SDRAM-Modul	4 W und höher
Verbrauch – Zugriff	0,5...5,8 W ^[74]	8 W pro SDRAM-Modul	6 W und höher
Verhalten beim PC-Ausschalten	problemlos	Datenverlust falls keine Sicherung auf SSD/Festplatte stattfindet	problemlos
Verhalten bei Stromausfall	mit Stützkondensator auf SSD-Ebene problemlos ^[75] ohne Stützkondensator Datenverlust möglich	Datenverlust	Datenverlust möglich
Lautlos	ja	ja	nein
Bemerkungen	unterstützen meistens S.M.A.R.T.	Größe begrenzt durch Hauptplatine oder zusätzlicher Adapter nötig,	unterstützen S.M.A.R.T.

... 2016

~4500 Euro in 2013



	MLC-NAND-Flash-Laufwerk 1,0" bis 3,5"	RAM-Disk als Teil des Arbeitsspeichers	Festplatte 1,0" bis 3,5"
Größe (keine Raidlaufwerke)	bis 16 TB	bis 32 GB je Modul	bis 12 TB
Preis pro TB (Stand Mai 2016)	ab ≈ 218 € ^[102]	ab ≈ 2.690 € ^[103]	ab ≈ 26 € ^[104]
Anschluss	IDE/(P)ATA, SATA, mSATA, PCIe, M.2	hauptsächlich DIMM-Connector	SCSI, IDE/(P)ATA, SATA, SAS
Lesen (kein RAID)	bis 510 MB/s ^[105]	bis 51.200 MB/s ^[106]	bis ca. 227 MB/s ^[107]
Schreiben (kein RAID)	bis 490 MB/s ^[105]	bis 51.200 MB/s ^[106]	bis ca. 160 MB/s ^[107]
Mittlere Zugriffszeit lesen	ab 0,031 ms ^[108]	0,000.02 ms	ab 3,5 ms
Mittlere Zugriffszeit schreiben	ab 0,023 ms ^[108]	0,000.02 ms	ab 3,5 ms
Überschreibbar (Zyklen)	3 bis 10 tausendmal (MLC)	> 10 ¹⁵ ^[109]	ca. 10 Mrd. (3 Jahre) ^[110]
Lagerbar bei	-45–85 °C	-25–85 °C	-40–70 °C
Stoßfestigkeit – Betrieb ^[111]	1.500 g	ca. 1.000 g (rüttelfest verlötet)	60 g
Stoßfestigkeit – Lagerung	1.500 g	ca. 1.000 g (ähnlich SSD)	350 g
Verbrauch – Ruhe	0,1–1,3 W	1 W pro SDRAM-Modul	4 W und höher
Verbrauch – Zugriff	0,5–5,8 W ^[112]	8 W pro SDRAM-Modul	6 W und höher
Verhalten beim Herunterfahren	problemlos	Datenverlust, falls keine Sicherung auf SSD/Festplatte stattfindet	problemlos
Verhalten bei Stromausfall	mit Stützkondensator auf SSD-Ebene problemlos ^[113] ohne Stützkondensator Datenverlust möglich	Datenverlust	Datenverlust möglich
Lautlos	Keine mechanischen Geräusche, bei Defekten sind Geräusche möglich ^{[114][115]}	ja	nein
Bemerkungen	unterstützen meistens S.M.A.R.T.	Größe begrenzt durch Hauptplatine oder zusätzlicher Adapter nötig, nicht bootfähig	unterstützen S.M.A.R.T.

Vergleich von Nano- vs Millisekunde

1 Nanosekunde (ns) = 1000 Pikosekunden = 0,000 000 001 Sekunden

1 Mikrosekunde (μ s) = 1000 Nanosekunden = 0,000 001 Sekunden

1 Millisekunde (ms) = 1000 Mikrosekunden = 0,001 Sekunden

In einer Nanosekunde (10^{-9} s) legt das Licht die Strecke von etwa 30 cm zurück.

In einer Mikrosekunde also $1000 * 30 \text{ cm} \rightarrow 30.000 \text{ cm} = 300 \text{ m}$.

In einer Millisekunde also $1000 * 300 \text{ m} \rightarrow 300.000 \text{ m} = 300 \text{ km}$.

Vergleich von Nano- vs Millisekunde (2)

1 Nanosekunde (ns) = 1000 Pikosekunden (ps) = 0,000 000 001 Sekunden

1 Mikrosekunde (μ s) = 1000 Nanosekunden = 0,000 001 Sekunden

1 Millisekunde (ms) = 1000 Mikrosekunden = 0,001 Sekunden

333 ps ($3,33 \cdot 10^{-10}$) – Taktzeit für einen Prozessor mit 3 GHz Taktfrequenz

→ 1 ns ~ 3 Takte

→ 1 ms ~ $1000 * 1000 * 3$ Takte = 3.000.000 Takte = 3 Millionen Takte

→ 20 ns ~ 60 Takte

→ 3,5 ms ~ 10,5 Millionen Takte

→ Für Datenbanksysteme ist meist die Anzahl der Blockzugriffe entscheidend.

Grundbegriffe

- **Block** – Organisationseinheit des Speichermediums (Festplatte)
- **Seite (Page)** – Organisationseinheit für das DBMS, Zuordnung zwischen Block und Seite mit festem Faktor: 1, 2, 4 oder 8.
z.B. SQLServer: „Ein Block umfasst acht zusammenhängende Seiten.“

<http://technet.microsoft.com/de-de/library/ms190969%28v=sql.105%29.aspx>

- Bei uns einfach: 1 Block = 1 Seite

Wie werden die Daten in einer Seite/Block abgelegt?

```
1 • SELECT * FROM star_trek.db2table;
```

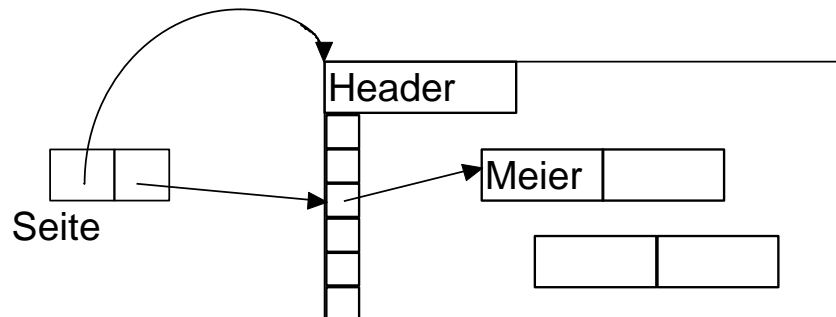
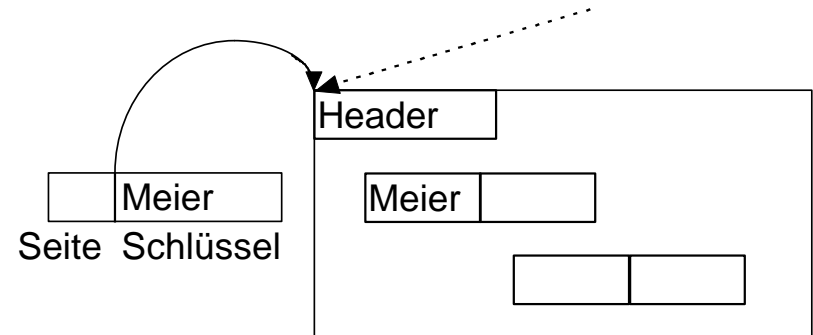
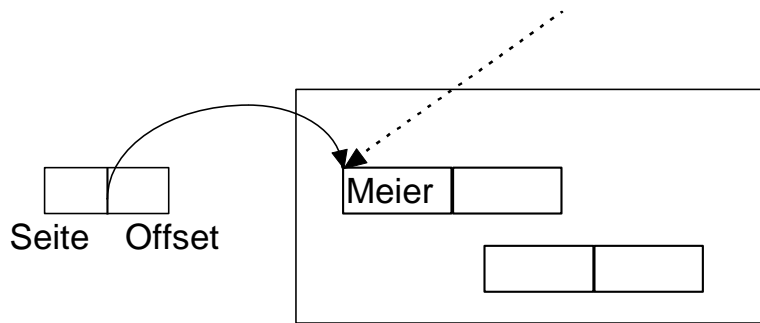
Result Set Filter: Edit

	attrint	attrvarchar
▶	1	Test1
	2	Test2
	3	Test3
	4	Test4
*	NULL	NULL

```
CREATE TABLE db2table  
(attrint int(11) NOT NULL,  
attrvarchar varchar(45),  
PRIMARY KEY (attrint)  
)
```

... und wer entscheidet darüber?

Wie wird auf Datensätze verwiesen?



Folgen?

Header

- Jede Seite (Block) umfasste einen Header! Dort stehen Meta-Informationen über die enthaltenen Daten. Was genau ist abhängig von der Speicherung. Beispiele sind:
 - Block-ID
 - # freier/belegter Plätze (Full/Empty-Bits)
 - Pointer zu den Datensätzen im Block
 - Deletion-Bits
 - ...

MySQL Storage Engines

<http://dev.mysql.com/doc/refman/5.6/en/storage-engines.html>

Table 15.1 Storage Engines Feature Summary

Feature	MyISAM	Memory	InnoDB	Archive	NDB
Storage limits	256TB	RAM	64TB	None	384EB
Transactions	No	No	Yes	No	Yes
Locking granularity	Table	Table	Row	Table	Row
MVCC	No	No	Yes	No	No
Geospatial data type support	Yes	No	Yes	Yes	Yes
Geospatial indexing support	Yes	No	Yes ^[a]	No	No
B-tree indexes	Yes	Yes	Yes	No	No
T-tree indexes	No	No	No	No	Yes
Hash indexes	No	Yes	No ^[b]	No	Yes
Full-text search indexes	Yes	No	Yes ^[c]	No	No
Clustered indexes	No	No	Yes	No	No
Data caches	No	N/A	Yes	No	Yes
Index caches	Yes	N/A	Yes	No	Yes
Compressed data	Yes ^[d]	No	Yes ^[e]	Yes	No
Encrypted data ^[f]	Yes	Yes	Yes	Yes	Yes
Cluster database support	No	No	No	No	Yes
Replication support ^[g]	Yes	Yes	Yes	Yes	Yes
Foreign key support	No	No	Yes	No	No
Backup / point-in-time recovery ^[h]	Yes	Yes	Yes	Yes	Yes
Query cache support	Yes	Yes	Yes	Yes	Yes
Update statistics for data dictionary	Yes	Yes	Yes	Yes	Yes

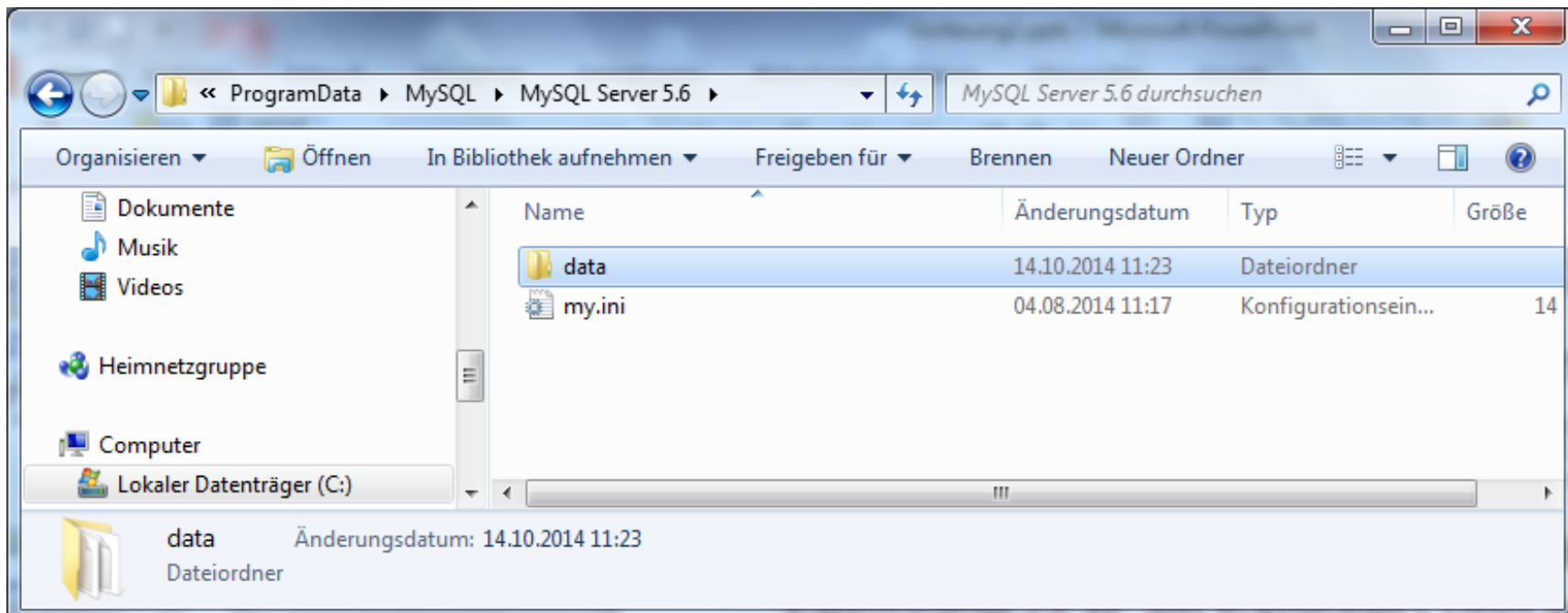
^[a] InnoDB support for geospatial indexing is available in MySQL 5.7.5 and higher.

^[b] InnoDB utilizes hash indexes internally for its Adaptive Hash Index feature.

^[c] InnoDB support for FULLTEXT indexes is available in MySQL 5.6.4 and higher.

Wie werden die verschiedenen Seiten einer Tabelle verwaltet?

- ... wo finde ich eigentlich die Daten auf der Festplatte (z.B. im Explorer) und wie sehen die aus?
- ... was ist die my.ini Datei?



Gemarkungen

select count(*) from admindivision_4; → 44895

select * from admindivision_4
where Name like 'Fra%';

id	Number	Parish	Name	Latitude	Longitude
7789	8847	09183113	Fraham	NULL	NULL
7790	8847	09183148	Fraham	NULL	NULL
35122	3812	07331026	Framersheim	NULL	NULL
5515	6171	09279122	Frammering	NULL	NULL
515	523	09677129	Frammersbach	NULL	NULL
516	522	09677131	Frammersbacher Forst	NULL	NULL
35562	4921	07335009	Frankelbach	NULL	NULL
1831	2048	09479166	Franken	NULL	NULL
33558	1135	07131077	Franken	NULL	NULL
37078	3905	14524290	Franken	NULL	NULL
13233	2221	12062092	Frankena	NULL	NULL
10875	2285	16076059	Frankenau	NULL	NULL
21712	1707	06635010	Frankenau	NULL	NULL
36393	7216	14522360	Frankenau	NULL	NULL

Gemarkungen

id	Number	Parish	Name	Latitude	Longitude
7789	8847	09183113	Fraham	NULL	NULL
7790	8847	09183148	Fraham	NULL	NULL
35122	3812	07331026	Framersheim	NULL	NULL
5515	6171	09279122	Frammering	NULL	NULL
515	523	09677129	Frammersbach	NULL	NULL
516	522	09677131	Frammersbacher Forst	NULL	NULL
35562	4921	07335009	Frankelbach	NULL	NULL
1831	2048	09479166	Franken	NULL	NULL
33558	1135	07131077	Franken	NULL	NULL
37078	3905	14524290	Franken	NULL	NULL
13233	2221	12062092	Frankena	NULL	NULL
10875	2285	16076059	Frankenau	NULL	NULL
21712	1707	06635010	Frankenau	NULL	NULL
36393	7216	14522360	Frankenau	NULL	NULL

~ 30 Byte pro DS

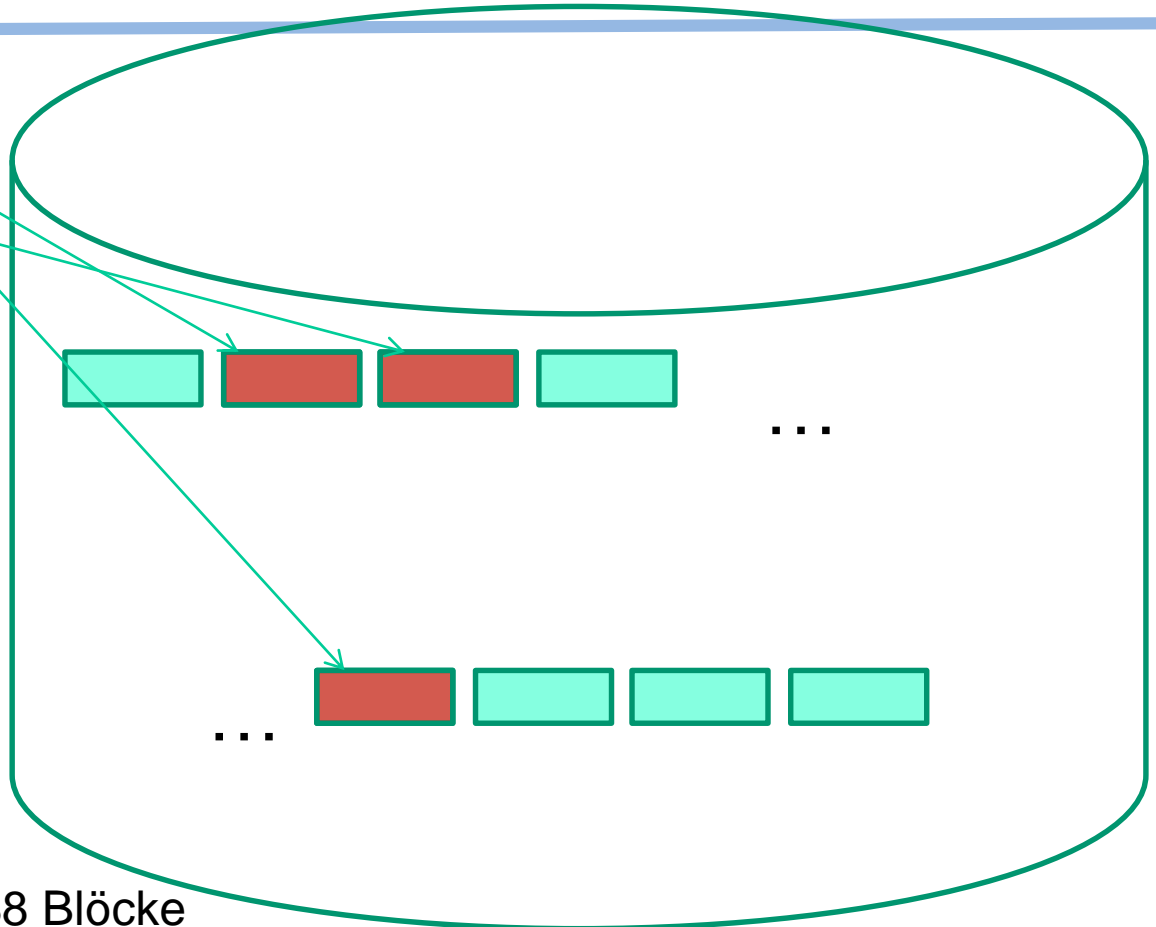
Block ~ 4000 Byte

→ ~ **133 DS pro Block**

44895 DS / 133 → mind. 338 Blöcke

`select * from admindivision_4 where Name like 'Fra%';`

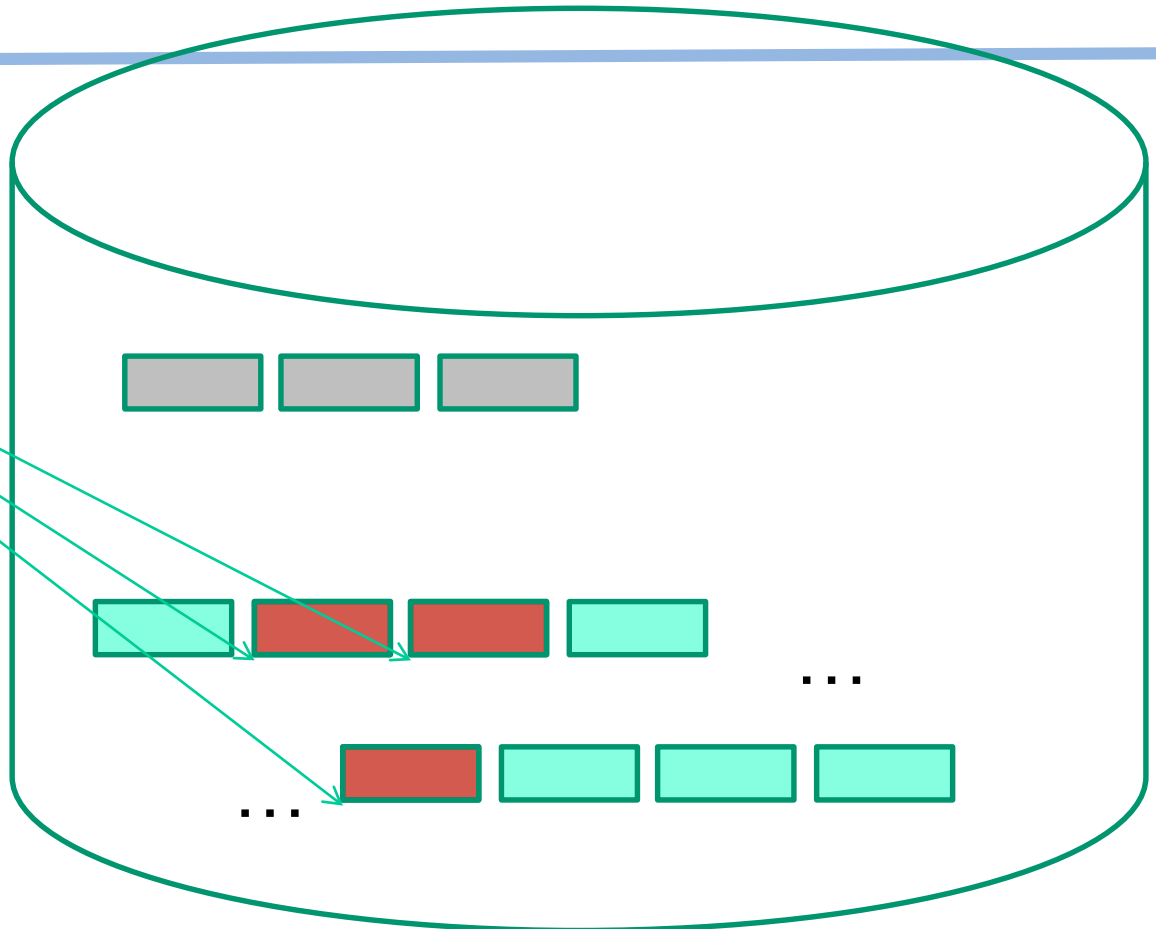
Anzahl zu lesender Blöcke (von HDD) ist mind. 338!



Gemarkungen mit Index

Index auf Name

BTree
üblich



id	Number	Parish	Name	Latitude	Longitude
7789	8847	09183113	Fraham	HOAL	HOAL
7790	8847	09183148	Fraham	HOAL	HOAL
35122	3812	07331026	Framersheim	HOAL	HOAL
5515	6171	09279122	Frammering	HOAL	HOAL
515	523	09677129	Frammersbach	HOAL	HOAL
516	522	09677131	Frammersbacher Forst	HOAL	HOAL
35562	4921	07335009	Frankelbach	HOAL	HOAL
1831	2048	09479166	Franken	HOAL	HOAL
33558	1135	07131077	Franken	HOAL	HOAL
37078	3905	14524290	Franken	HOAL	HOAL
13233	2221	12062092	Frankena	HOAL	HOAL
10875	2285	16076059	Frankenau	HOAL	HOAL
21712	1707	06635010	Frankenau	HOAL	HOAL
36393	7216	14522360	Frankenau	HOAL	HOAL

`select * from admindivision_4 where Name like 'Fra%';` 149 DS im Ergebnis
Anzahl zu lesender Blöcke ist maximal 149 (plus Index)!

Beispiel aus der Praxis

Tabelle 'myTable' mit:

Attributen: a, b und c

1.000.000 Einträgen

Index auf Attribut 'a'

Anfrage:

```
SELECT a,b,c FROM myTable  
WHERE a > 19850
```

Beispiel aus der Praxis

In einem Beispiel waren ca. 50% der Einträge unter Attribut 'a' mit dem Wert 19870 belegt.

→ Nutzung des Index nicht sinnvoll!

Ausführungszeit lag bei **10 Minuten!**

In Oracle können Hints in SQL Anfragen eingebaut werden:

```
SELECT /*+ full(myTable)*/ a, b, c FROM myTable  
WHERE a > 19850
```

Ausführungszeit ohne Index ~15sec