Grundlagen der Datenbanktechnik

Martin Klier

Klug GmbH integrierte Systeme, Teunz

22.02.2010

Referent

- Martin Klier, 30
- Datenbankadministrator für ORACLE
- Fachliche Schwerpunkte:
  - Performanceoptimierung / Tuning
  - hochverfügbare Systeme
  - Cluster und Replikation
- Linux seit 1997
- Oracle Database seit 2003
- Kontakt: martin.klier@unix.net
- Weblog: http://www.usn-it.de

Unternehmen

- Klug GmbH integrierte Systeme
  Lindenweg 13
  92552 Teunz
- Führendes und erfolgreiches Unternehmen der Software- und Steuerungssysteme für die Intralogistik
- Konzeption, Beratung, Softwareerstellung, Elektronik, Kommissioniertechnik, Hardware, ...
- 210 Mitarbeiter
- 50 Auszubildende

ORACLE PARTNER

DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Was ist eine Datenbank?

Was ist eine Datenbank?

„Die wesentliche Aufgabe eines Datenbanksystems ist es, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern und benötigte Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformen für Benutzer und Anwendungsprogramme bereitzustellen.“ (Wikipedia)

Anwendung - OLTP

OLTP – OnLine Transaction Processing
Echtzeitbetrieb
viele kleine Transaktionen

- Kunden-, Adress- und Artikeldatenbank, (Online-)Shop
- Website-backend (dynamische Webseiten)
- Finanzbuchhaltung
- Automatisiertes Lager / Logistik / Warenwirtschaft

Response time, response time, response time!

Anwendung - OLAP

OLAP – OnLine Analytcal Processing

DataWarehouse

wenige große Transaktionen, viel Leseaktivität

- Auswertungen zur Wirtschaftlichkeit, Controlling
- Auswertungen zur Güterverteilung
- Kundenprofil-Erstellung
- Rasterfahndung (BKA/LKA)

„Erkaufen“ von Zeit - durch Aufwand von Speicherplatz

Was kann eine Datenbank?

- Relationen schützen!
- Daten konsistent lesen!
- Daten konsistent schreiben!
- **Trotzdem effektiv/performant arbeiten?**

DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

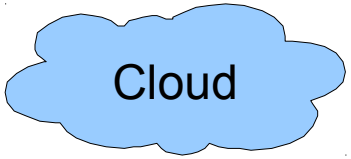
Geschichte I

- 1970 Theoretische Grundlagen von Edgar F. Codd (IBM Almaden)
- 1974 Ingres RDBMS (QUEL)
- 1977 IBM System R
  Sonderanfertigung für Pratt&Whitney (SEQUEL)

- 1977 SDL (Vorläufer von Oracle) erstellt SEQUEL-RDMBS für C.I.A.
- 1979 RSI (ex SDL) bringt „Oracle RDBMS 2.0“ auf den Markt (SQL)
- 1984 Erstes RDMBS mit Lesekonsistenz

Geschichte II

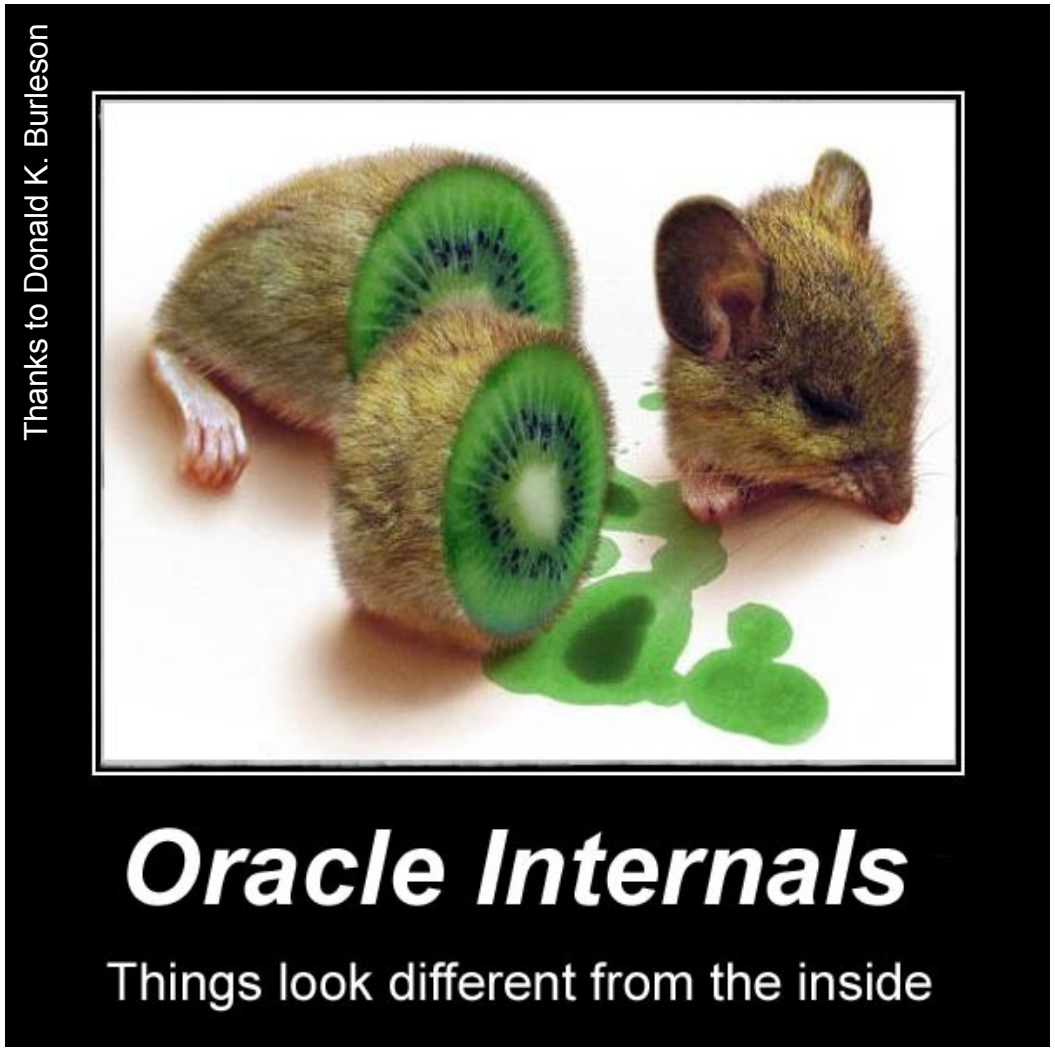
- 1987 RDBM-Systeme für UNIX kommen auf den Markt
- 1988 Prozedurale Programmierung möglich (PL/SQL)
- 1995 64bit-RDBMS erscheinen
- 1997 Datenbanken werden für Internetapplikationen optimiert
- 1999 XML-Unterstützung 

- 2003 Datenbank-Server-Cluster mit voller Cachekohärenz
- 2005 Ableger der großen Enterprise-RDBMS kostenlos verfügbar
- 2008 „Selbstverwaltende Datenbank“ macht Fortschritte
- 2010 Database 

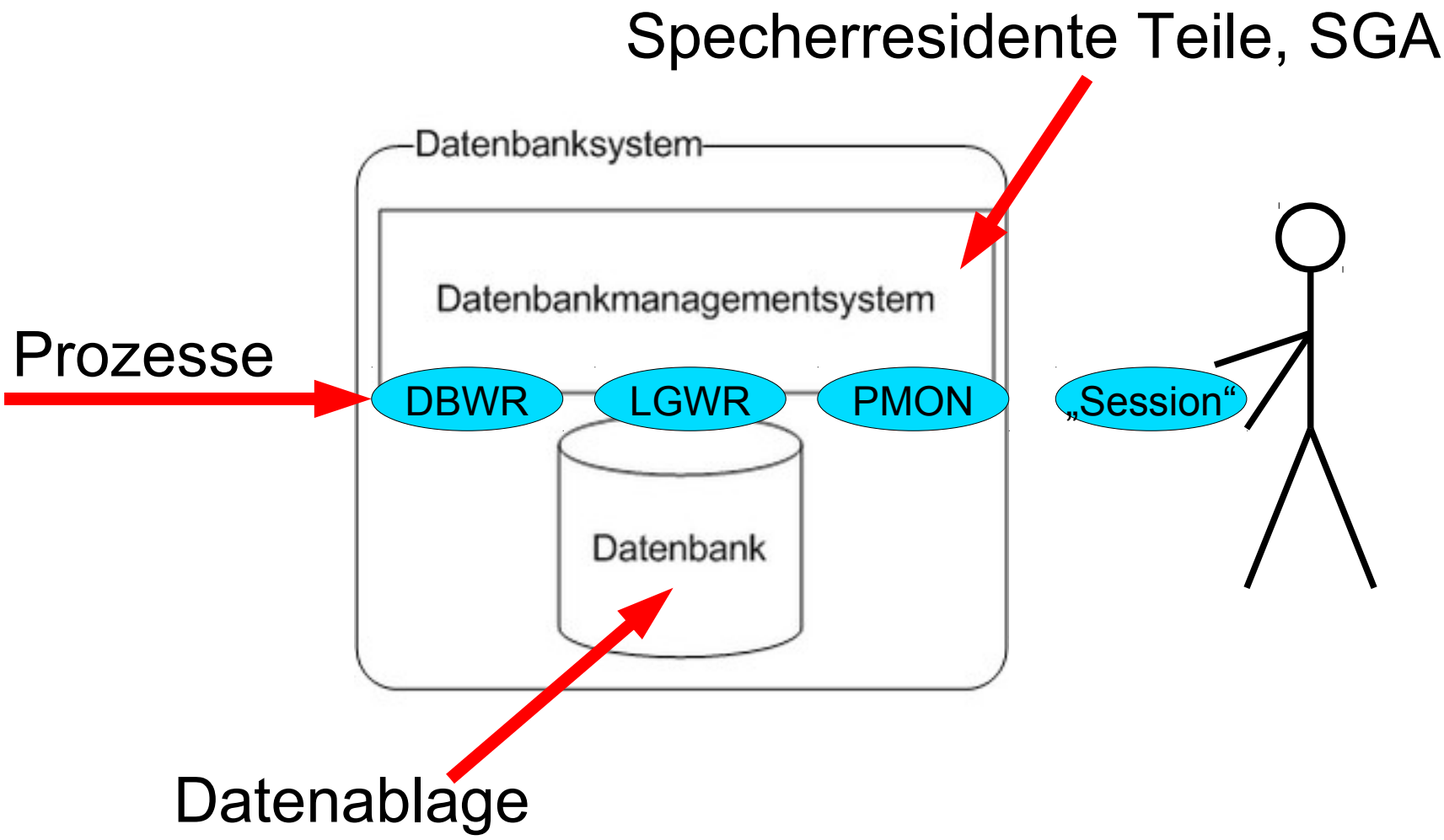
DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Am Rande ...


Thanks to Donald K. Burleson
Oracle Internals
Things look different from the inside

Aufbau des RDBMS

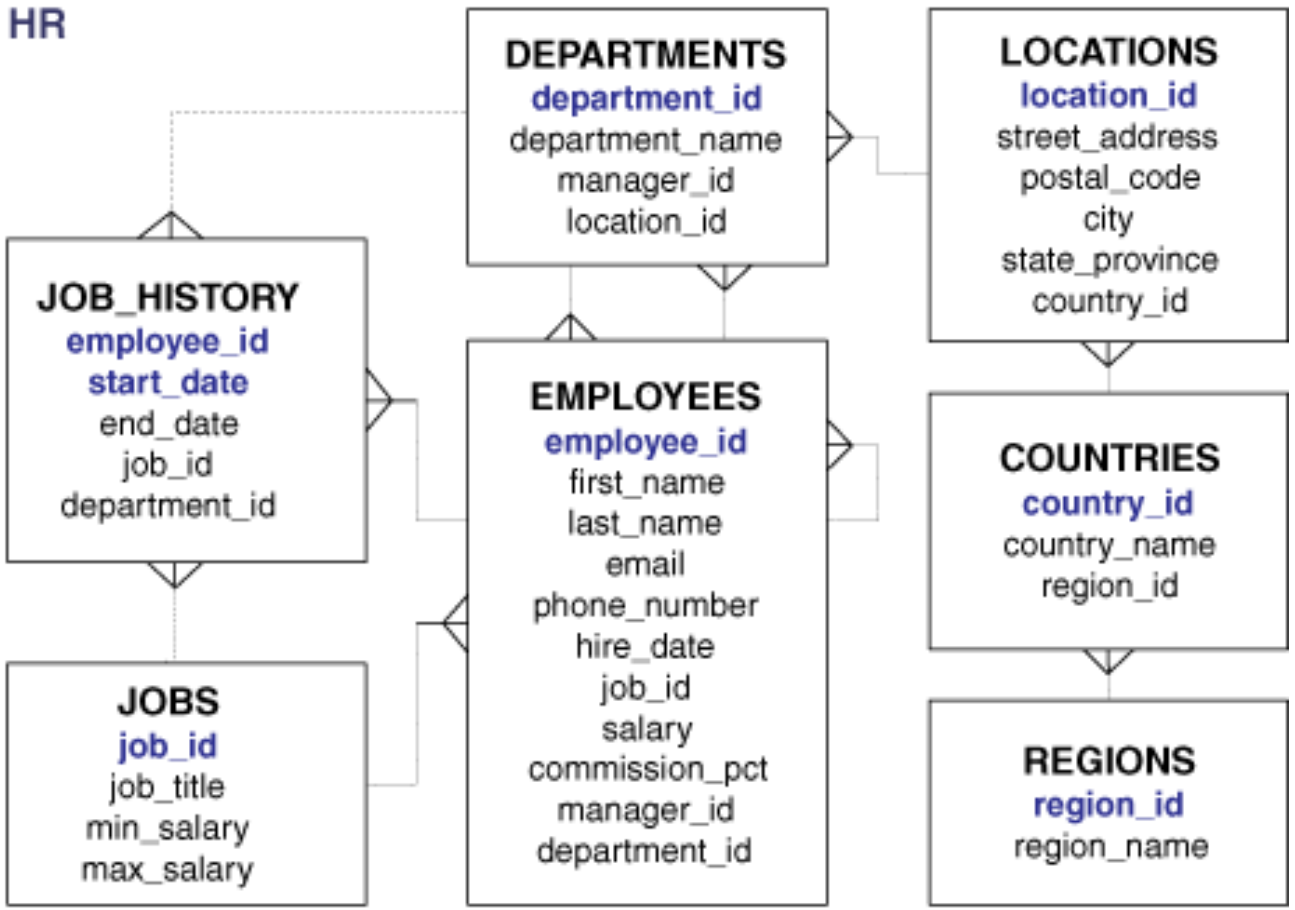


DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Relationen

ER-Diagramm



FK constraint violation

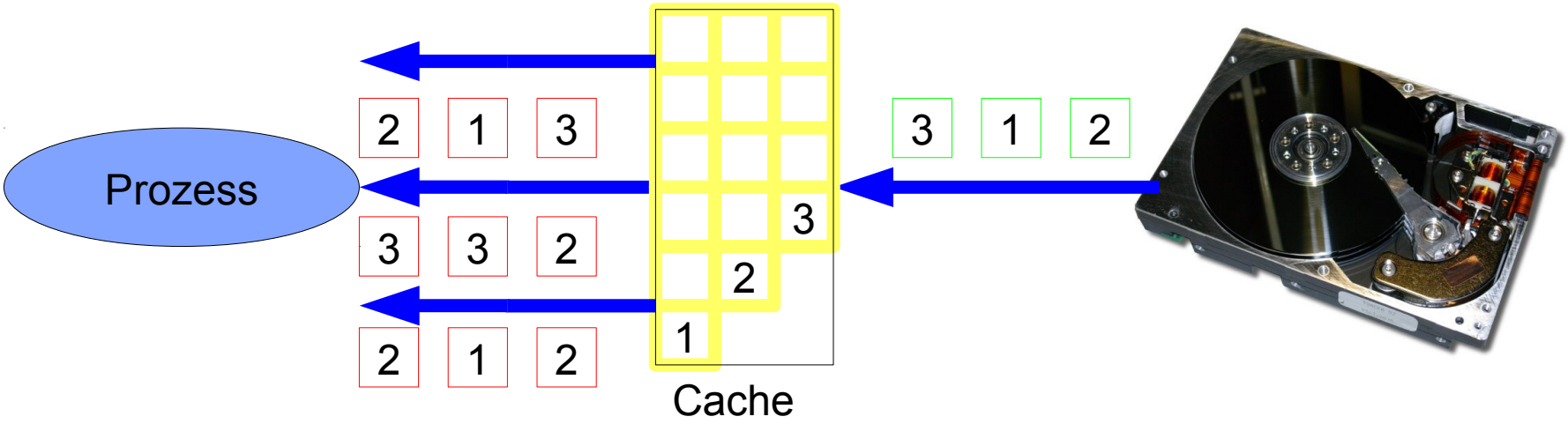
```
SQL> delete from JOB_HISTORY where EMPLOYEE_ID=4711;


SQL-Fehler: ORA-02292: Integritäts-Constraint (HR.R_213)
       verletzt - untergeordneter Datensatz gefunden
02292. 00000 - "integrity constraint (%s.%s)
       violated - child record found"

*Cause:    attempted to delete a parent key value that had a
             foreign dependency.

*Action:   delete dependencies first then parent or
             disable constraint.
```

DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Cache / Absicherung

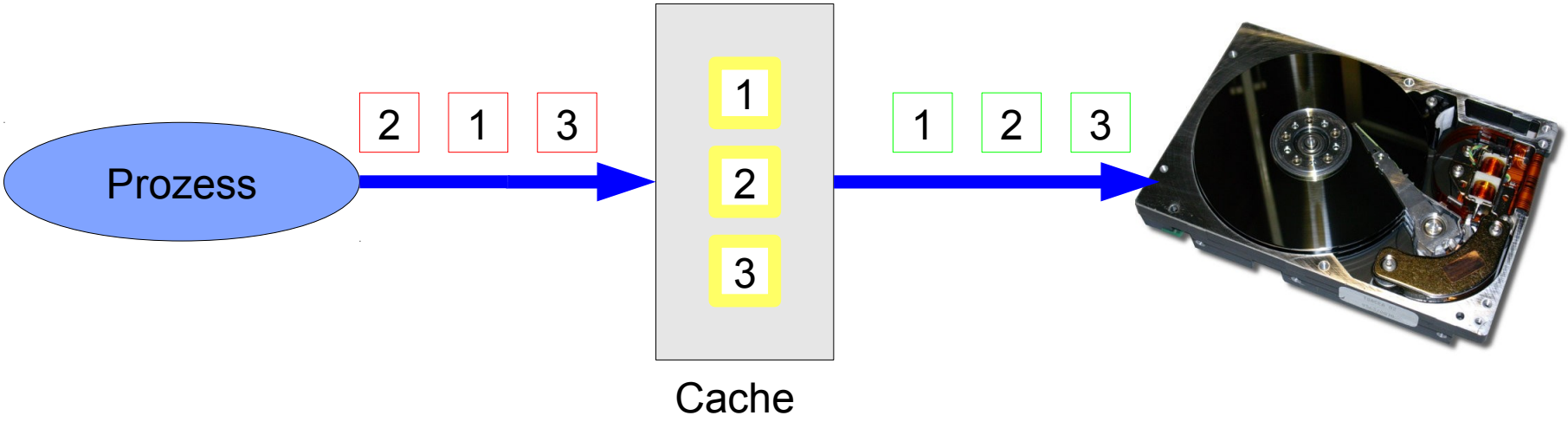
Lese-Cache (RAM nach Disk) ...



... macht Lesezugriff zu RAM-Zugriff.

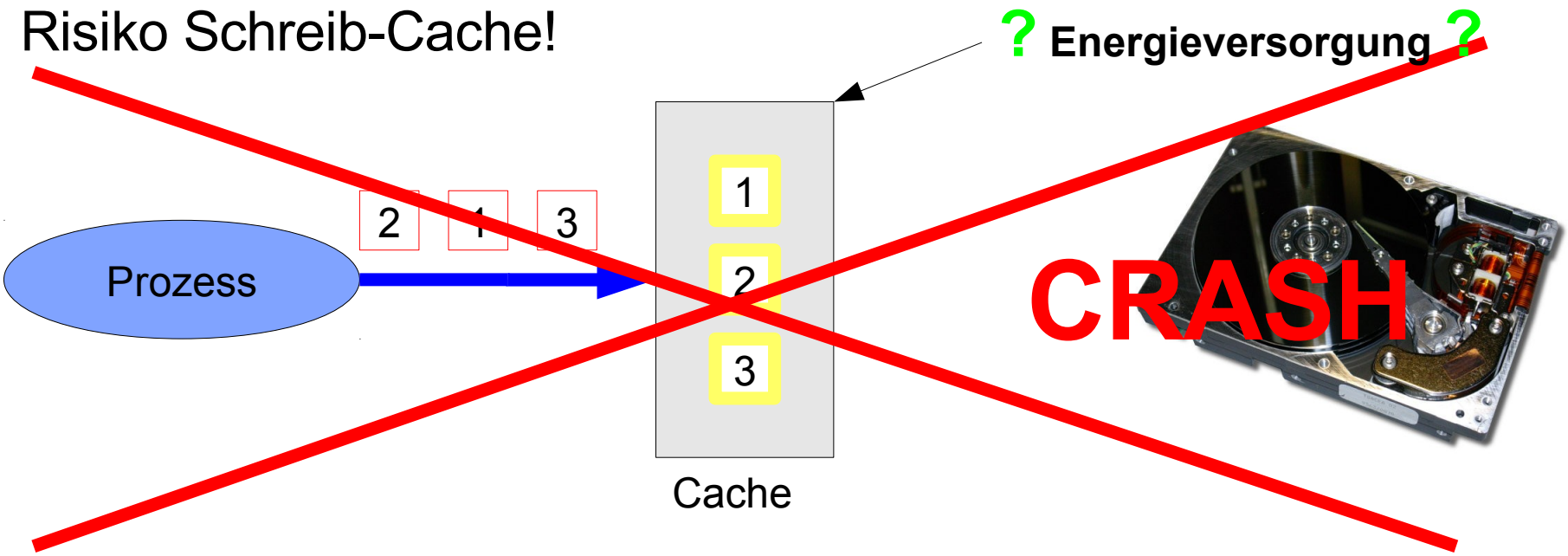
Cache / Absicherung

Schreib-Cache (RAM vor Disk) ...



... serialisiert.

Cache / Absicherung



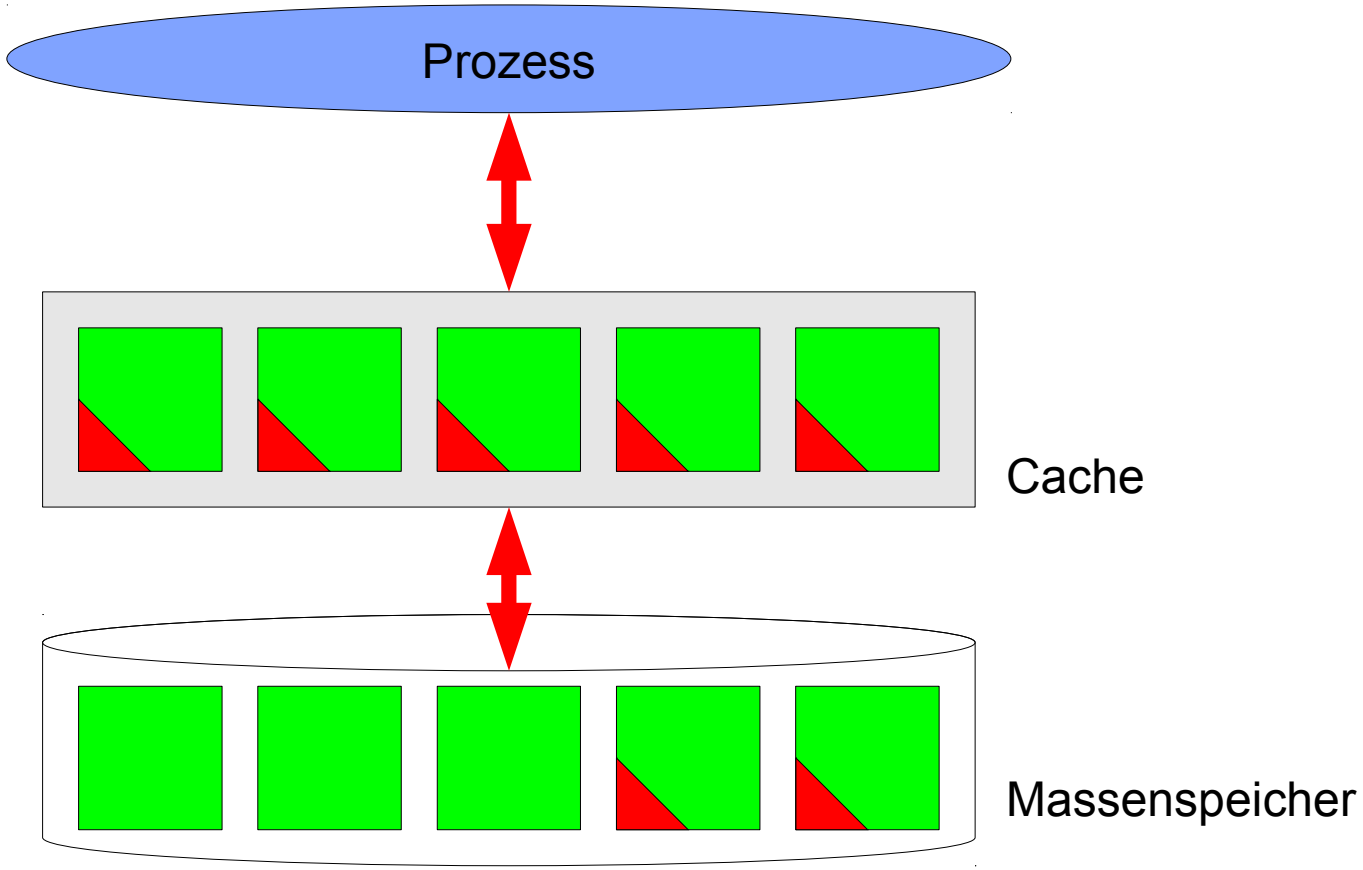
Cache / Absicherung

- Cache beschleunigt Zugriffe
- Aber: RAM => Totalverlust bei Spannungsausfall
- Cache erfordert Absicherung!
- Kompromiss notwendig
- Schreiben von „Delta“-Werten auf schnelle HDD

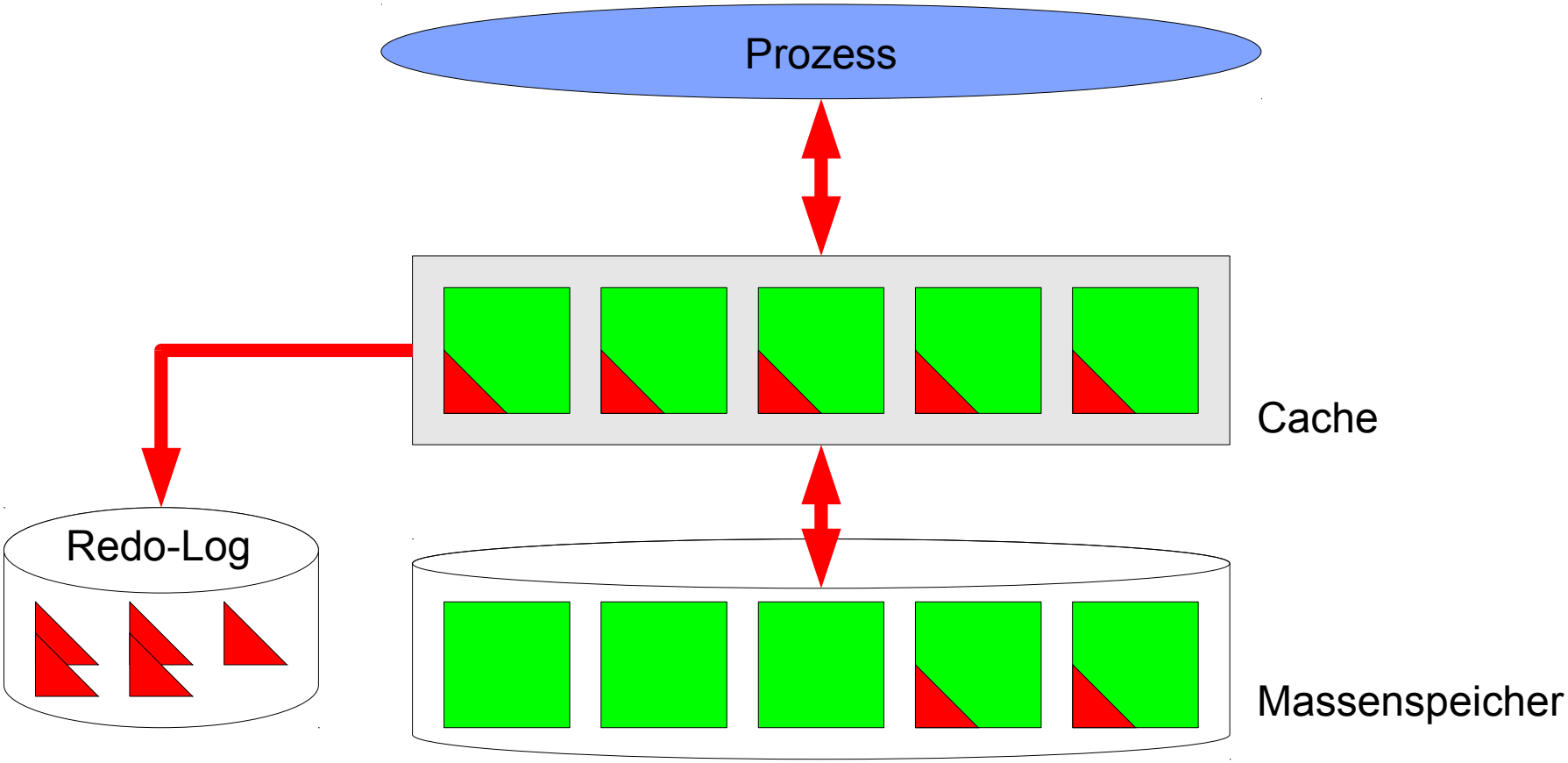
Cache / Absicherung

CRASH

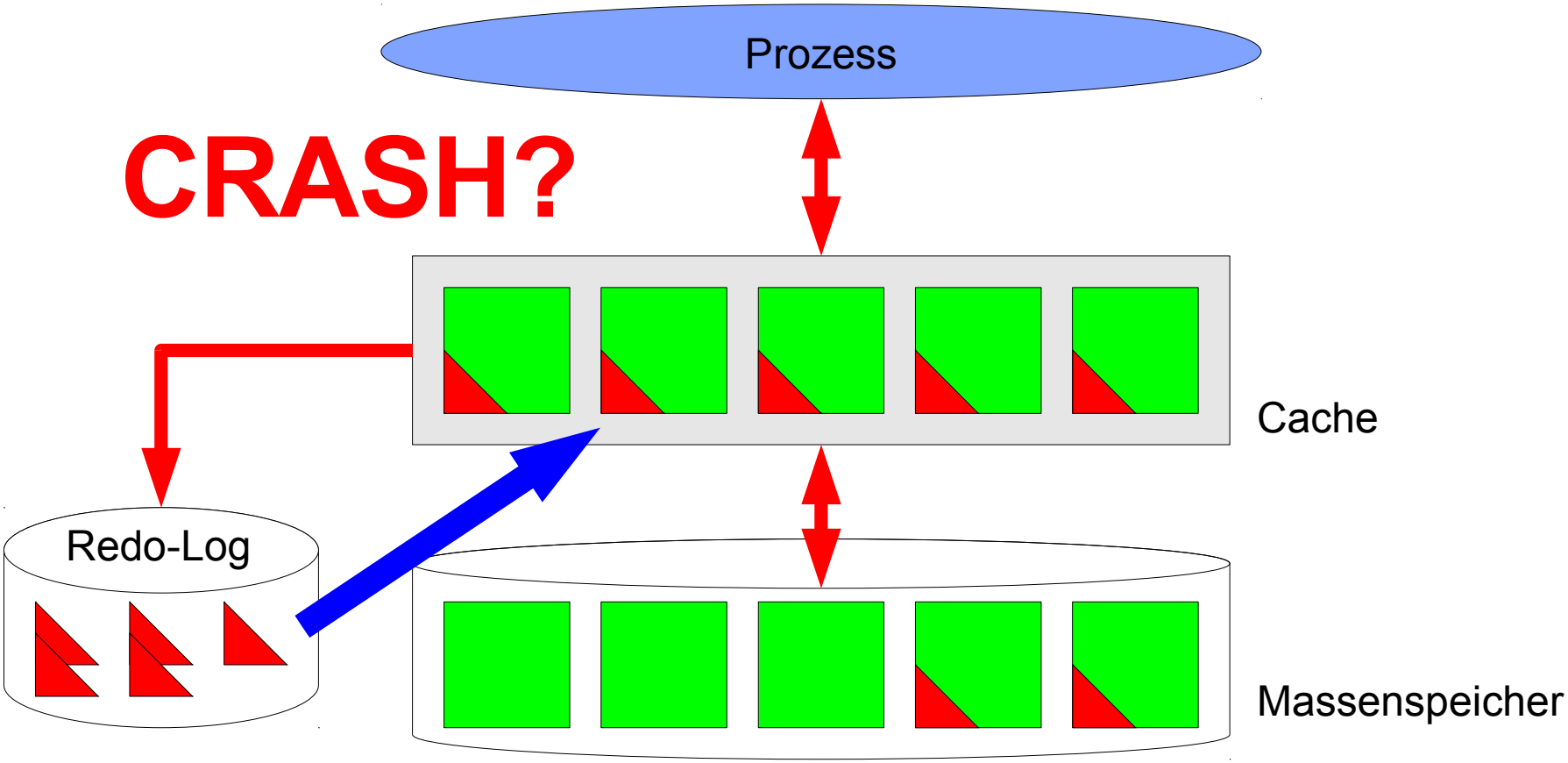
??



Cache / Absicherung



Cache / Absicherung



DB-Grundlagen - Agenda

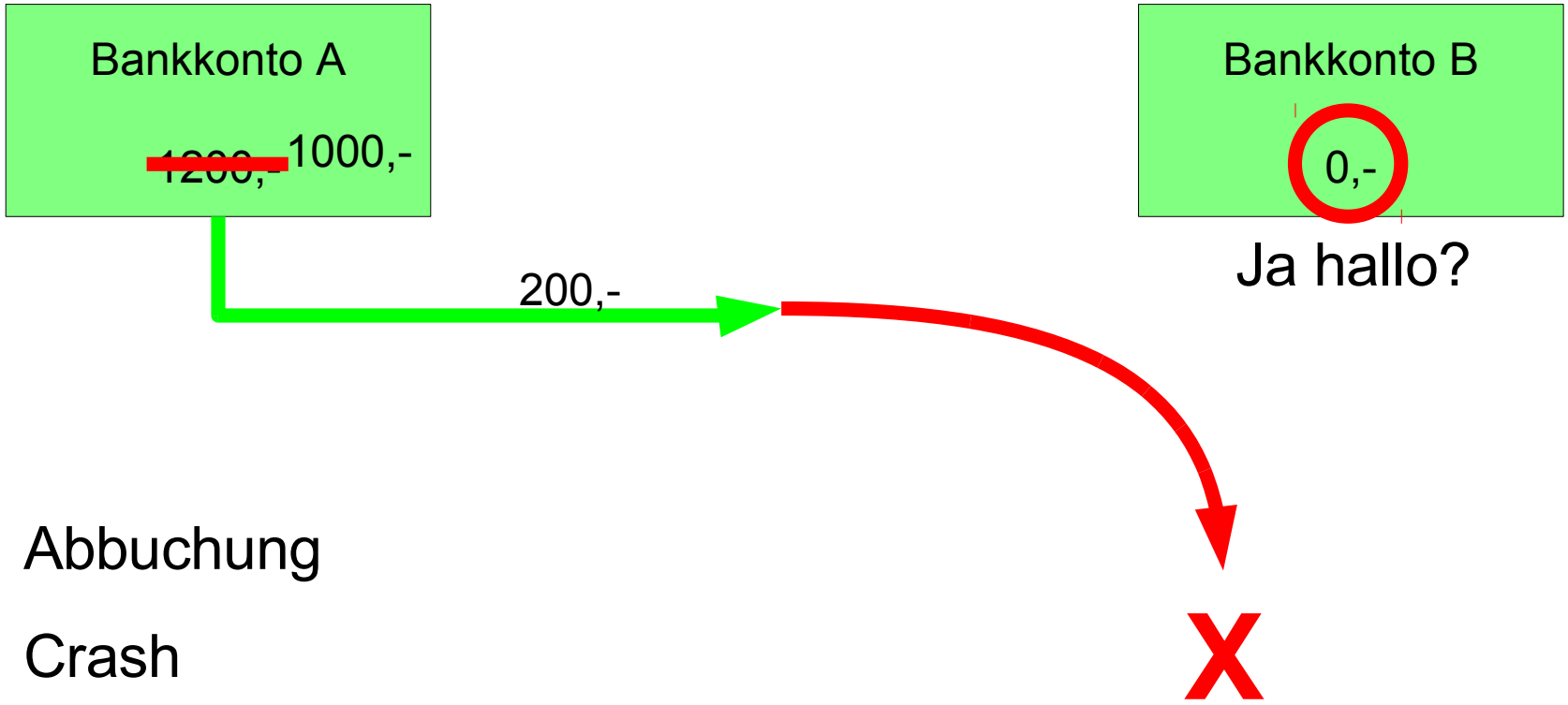
- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Transaktionsprinzip

„Eine Transaktion fasst in sich mehrere logisch zusammengehörige, jedoch nicht zwangsweise technisch voneinander abhängige Arbeitsschritte zusammen“

Transaktionsprinzip

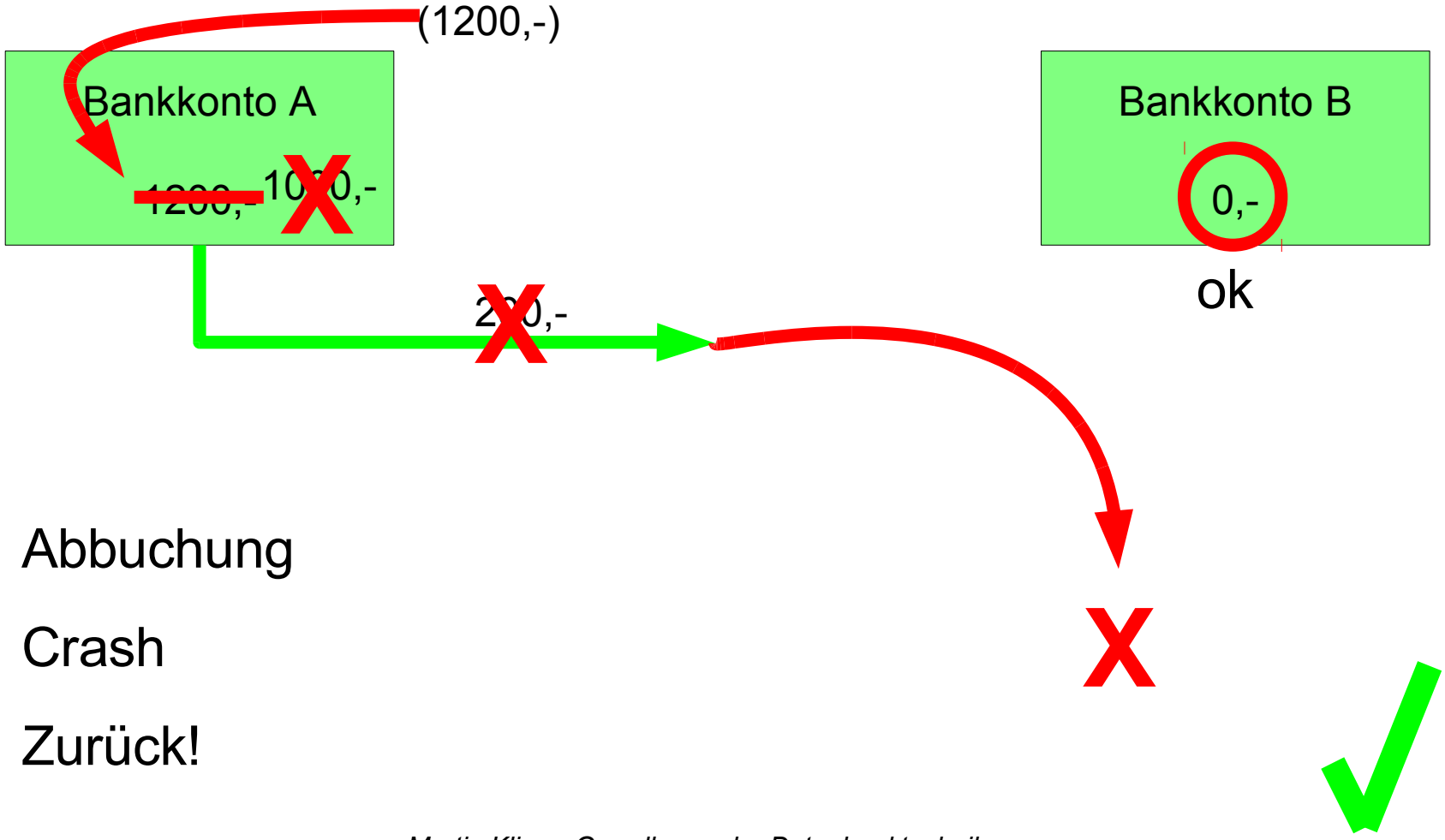
Beispiel „Überweisung“



- ➔ Abbuchung
- ➔ Crash
- ➔ Was nun?

Transaktionsprinzip

Beispiel „Überweisung“

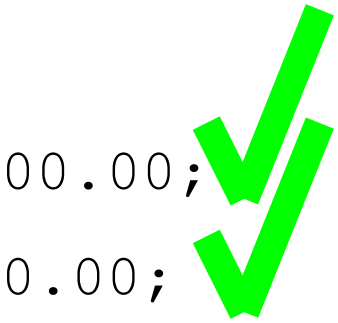


- ➔ Abbuchung
- ➔ Crash
- ➔ Zurück!

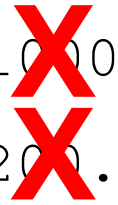
Transaktionsprinzip

Commit / Rollback

```
update KONTEN where KONTO_NR=A set SALDO=1000.00;
update KONTEN where KONTO_NR=B set SALDO=200.00;
commit;
```



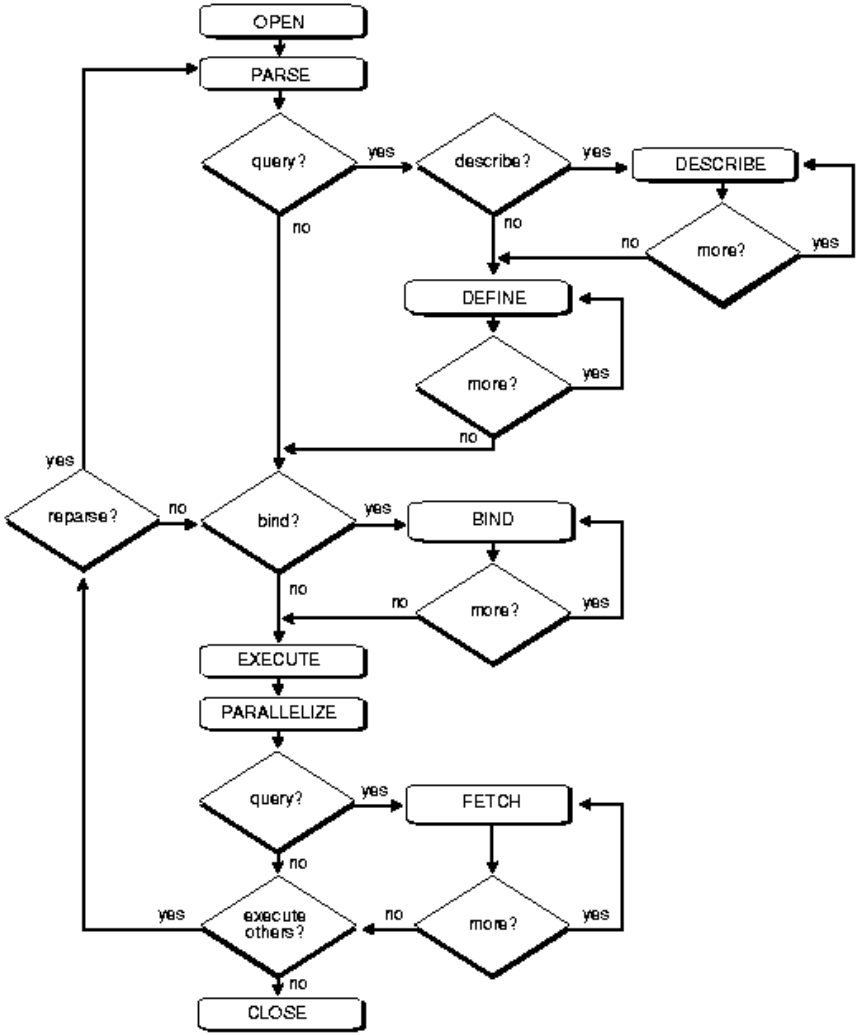
```
update KONTEN where KONTO_NR=A set SALDO=1000.00;
update KONTEN where KONTO_NR=B set SALDO=200.00;
####### update ERROR 890 - USER NOT SANE ######
rollback;
```



DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - JOIN
- Zusammenfassung

Behandlung von SQL



Behandlung von SQL

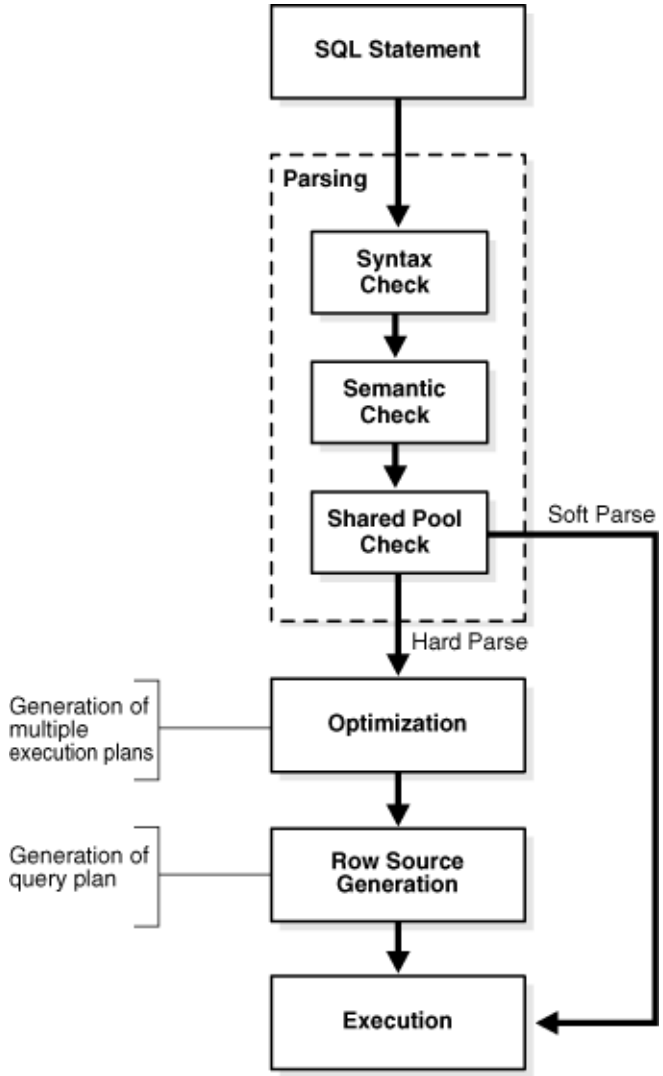
Anforderungen:

- Häufige Ausführung
- Performance / Antwortzeiten

Probleme:

- Endliche, d. h. begrenzte Systemleistung
- Unbekannte Mengenverhältnisse
  und Hellsehen ist bestenfalls schwierig
- Unberechenbare „User“

Behandlung von SQL



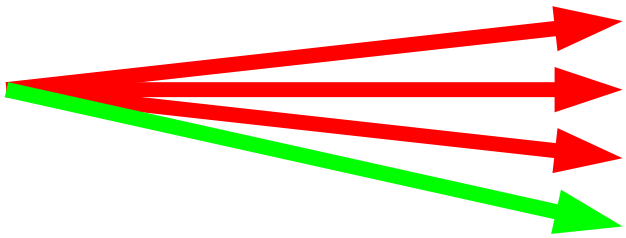
Beispiel!

Join

```
select b.employee_id
    from JOBS a, EMPLOYEES b
    where a.JOB_ID=b.JOB_ID
            and a.JOB_ID=20;
```

JOBS

JOB_ID	JOB_TITLE
100	Bäcker
20	Metzger
...	
13	Maler



EMPLOYEES

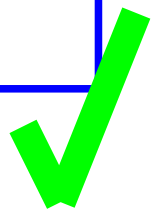
EM_ID	JOB_ID	NAME
5233	97	Berger
5253	13	Wolf
....		
5243	20	Schulz

DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Update

```
update EMPLOYEES a
    set a.salary = 1200
    where a.JOB_ID = 20;
```



EMPLOYEES

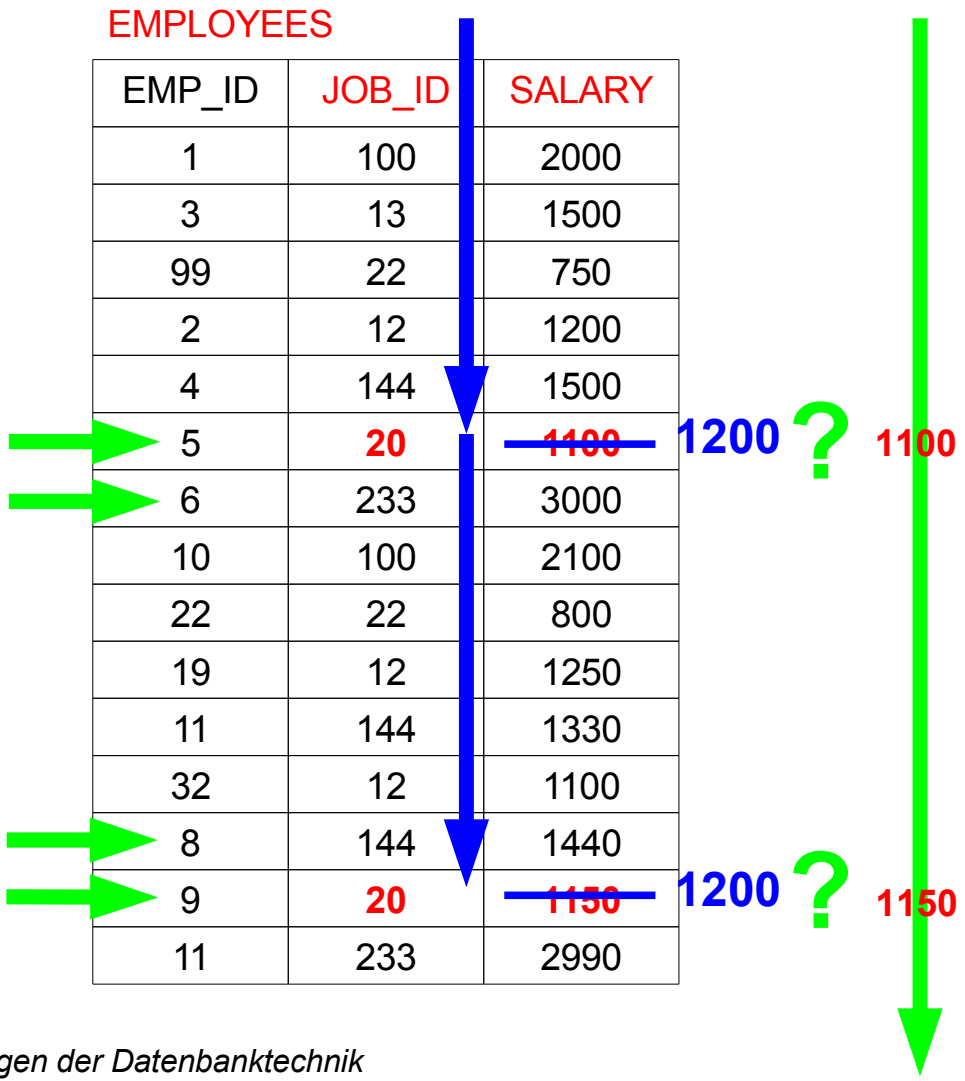
EMP_ID	JOB_ID	SALARY
1	100	2000
3	13	1500
99	22	750
2	12	1200
4	144	1500
5	**20**	~~1100~~ **1200**
6	233	3000
10	100	2100
22	22	800
19	12	1250
11	144	1330
32	12	1100
8	144	1440
9	**20**	~~1150~~ **1200**
11	233	2990

commit

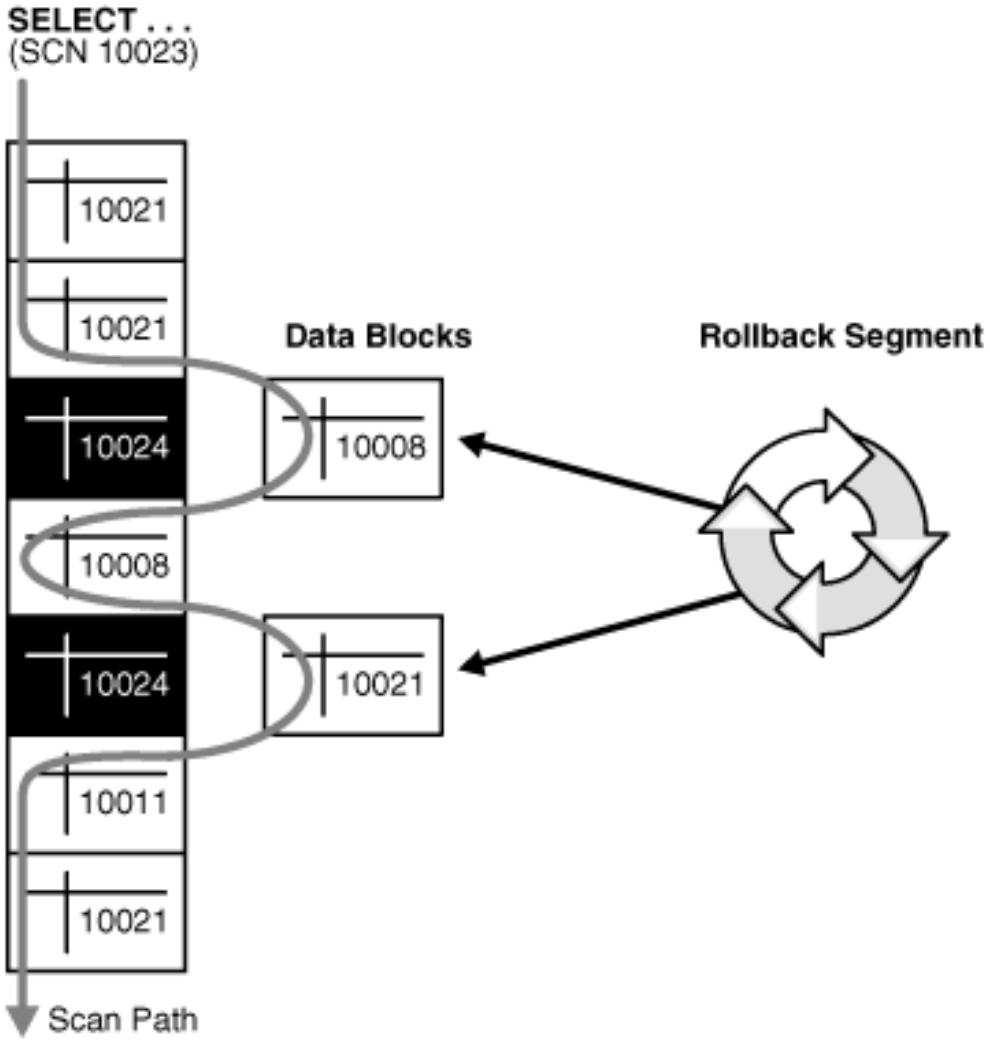
Update: Konsistentes Lesen

```
update EMPLOYEES a
    set a.salary = 1200
    where a.JOB_ID = 20;
```

```
select SALARY
from EMPLOYEES
where EMP_ID > 4
    and EMP_ID < 10;
```



Update: Konsistentes Lesen



Update: Konkurrenz

```
update EMPLOYEES a
    set a.salary = 1200
    where a.JOB_ID = 20;
```

```
update EMPLOYEES a
    set a.salary = 1500
    where a.EMP_ID = 5;
```

Lock

EMPLOYEES

EMP_ID	JOB_ID	SALARY
1	100	2000
3	13	1500
99	22	750
2	12	1200
4	144	1500
5	**20**	~~1100~~ 1200 ?
6	233	3000
10	100	2100
22	22	800
19	12	1250
11	144	1330
32	12	1100
8	144	1440
9	20	~~1150~~ 1200
11	233	2990

Update

```
update EMPLOYEES a
    set a.salary = 1200
    where a.JOB_ID = 20;
```

```
update EMPLOYEES a
    set a.salary = 1500
    where a.EMP_ID = 5;
```

Lock

EMPLOYEES

EMP_ID	JOB_ID	SALARY
1	100	2000
3	13	1500
99	22	750
2	12	1200
4	144	1500
5	20	~~1100~~ 1200 1500
6	233	3000
10	100	2100
22	22	800
19	12	1250
11	144	1330
32	12	1100
8	144	1440
9	20	~~1150~~ 1200
11	233	2990

commit

Einschub: TCL

Wie funktioniert TCL (Transaction Control Language)?

- COMMIT
- ROLLBACK

DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Index

Aufgaben:

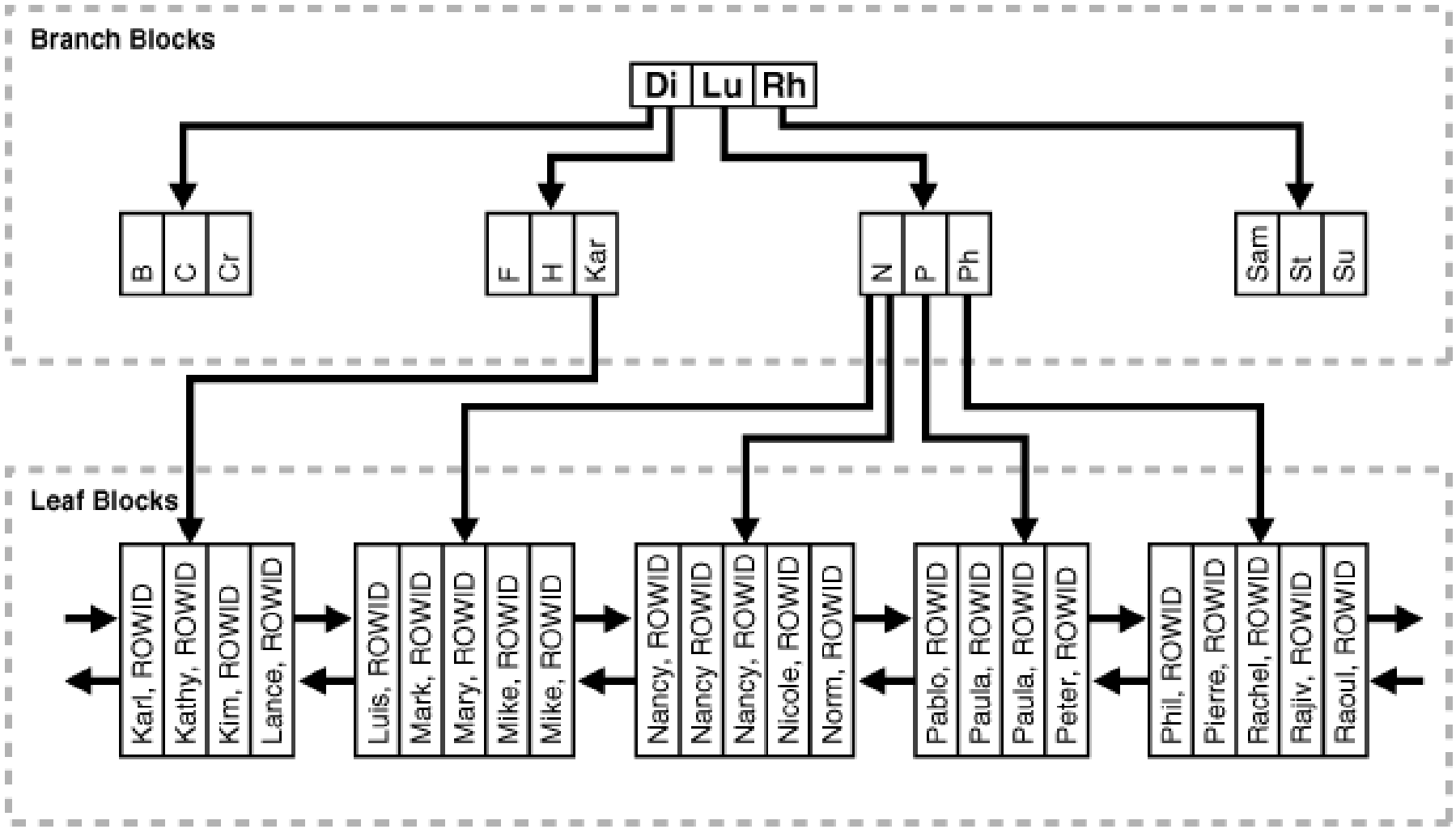
- Beschleunigung des Zugriffs
- Einsparen von Sortieroperationen
- Einsparen von gelesener Datenmenge
- Technische Realisierung des UNIQUE-Constraints

Index

Wie ist ein Index aufgebaut?

- Einfachster Fall: Numerische Werte
- Wachstum und Balance

Index: Alphanumerisch



Index

Aufgaben:

- Beschleunigung des Zugriffs
- Einsparen von Sortieroperationen
- Einsparen von gelesener Datenmenge
- Technische Realisierung des UNIQUE-Constraints

Am Rande ...

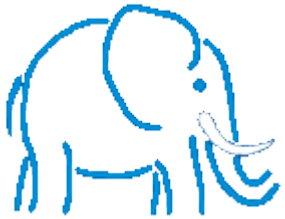

Thanks to Donald K. Burleson
Fully Automated SQL Tuning
Sometimes you need a human expert

DB-Grundlagen - Agenda

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung

Klassifikaktion I

- **Tabellenkalkulation**
  - Excel
  - OpenOffice Calc
- **Workstation-Level**
  - Access
  - BerkeleyDB
- **Mittelklasse**
  - MySQL
  - PostegreSQL



Klassifikation II

- **Enterprise-Niveau**
  - Oracle
  - IBM DB2
  - Microsoft SQL-Server
  - IBM Informix
  - Sybase
  - (SAP-DB)

Bewertung der Klassen

Produktklasse	Vorteile	Nachteile
Tabellenkalkulation	schnell erlernbar wenig Aufwand	kein SQL Strukturschwächen
Workstation-Niveau	geringe Größe wenig Administration	keine Mehrbenutzerfähigkeiten fehlende Absicherung (Cache usw.)
Mittelklasse	kostenlos viele Fähigkeiten	schwache Transaktionssicherheit Sperr- bzw. Konsistenzprobleme
Enterprise Level	fast alle Features exzellenter Support	sehr teuer (~30.000 EUR / CPU) Admin: lange Einarbeitungszeit

Know your needs!

Zusammenfassung

- Vorstellung
- Geschichte
- Aufbau und Architektur
- Relationen
- Interne Abläufe
  - Cache/Absicherung
  - Transaktionsprinzip
  - SELECT
  - UPDATE
  - Index
- Zusammenfassung


Q & A

Abschluß

Quellen:

- http://www.oracle.com/technology
- http://wikipedia.org
- http://www.usn-it.de (eigenes Blog)

Vielen Dank für Ihre Aufmerksamkeit!