

Zentralübung ERDB

Programm heute:

- Übersicht Prüfungsstoff
- Fragerunde – Aber auch gerne während der Übersicht

Allgemeine Voraussetzungen

- Grundkenntnisse wichtig!
 - Bsp. Cube Operator
 - Bsp. Skyline in SQL
 - Bsp. Datalog in SQL ausdrücken
- Casts zwischen Datentypen nicht wichtig
- Folien mixen teilweise DB2 SQL Syntax und Postgre SQL Syntax → Beide sind in der Prüfung gültig, innerhalb einer Aufgabe muss aber konsistent sein

- Universitätsschema muss nicht auswendig gelernt werden
- Schema ist immer angegeben
- Teilw. auch Beispieldaten

Überblick Prüfungsstoff

Disclaimer: Werde heute hauptsächlich besprechen welche Themen besonders wichtig sind. Falls etwas auf den Folien nicht erwähnt ist, aber in den Übungen / Vorlesung besprochen wurde, kann darauß nicht geschlossen werden, dass es nicht in der Prüfung drankommt. Auf den Folien dargestellte Übungen können Fehler enthalten

Recovery - **Alles wichtig, besonderer Fokus auf:**

- ACID
- Force / Steal 
- Write Ahead Logging
- Log Struktur
- Physical / Logical logging
- Redo / Undo
 1. Analyse
 2. Redo aller Transaktion
 3. Undo der Losertransaktionen
- Kompensationseinträge

Frage: Wie sähen die physischen Logeinträge aus?

$$A = 900, C = 1000, B = 2000$$

Schritt	T_1	T_2	Log
			[LSN, TA, PageID, Redo, Undo, PrevLSN]
1.	BOT		[#1, T_1 , BOT , 0]
2.	$r(A, a_1)$		
3.		BOT	[#2, T_2 , BOT , 0]
4.		$r(C, c_2)$	
5.	$a_1 := a_1 - 50$		$A = 850, A = 900$ [#3, T_1 , $P_A, A-50, A+=50, \#1$]
6.	$w(A, a_1)$		
7.		$c_2 := c_2 + 100$	
8.		$w(C, c_2)$	[#4, $T_2, P_C, C+=100, C=100, \#2$]
9.	$r(B, b_1)$		$C = 1100, C = 1000$ $B = 2050, B = 2000$
10.	$b_1 := b_1 + 50$		[#5, $T_1, P_B, B+=50, B=50, \#3$]
11.	$w(B, b_1)$		[#6, T_1 , commit , #5]
12.	commit		
13.		$r(A, a_2)$	
14.		$a_2 := a_2 - 100$	$A = 750, A = 850$
15.		$w(A, a_2)$	[#7, $T_2, P_A, A-=100, A+=100, \#4$]
16.		commit	[#8, T_2 , commit , #7]

Frage: Force / Steal und Logging

*Redo
Undo*

	Force	No Force
No Steal	?? <i>kein Redo/ Undo</i>	?? <i>Redo kein Undo</i>
Steal	?? <i>kein Redo Undo</i>	?? <i>Redo Undo</i>



Frage: Welche Kombination für Hauptspeicherdatenbanken besonders relevant?

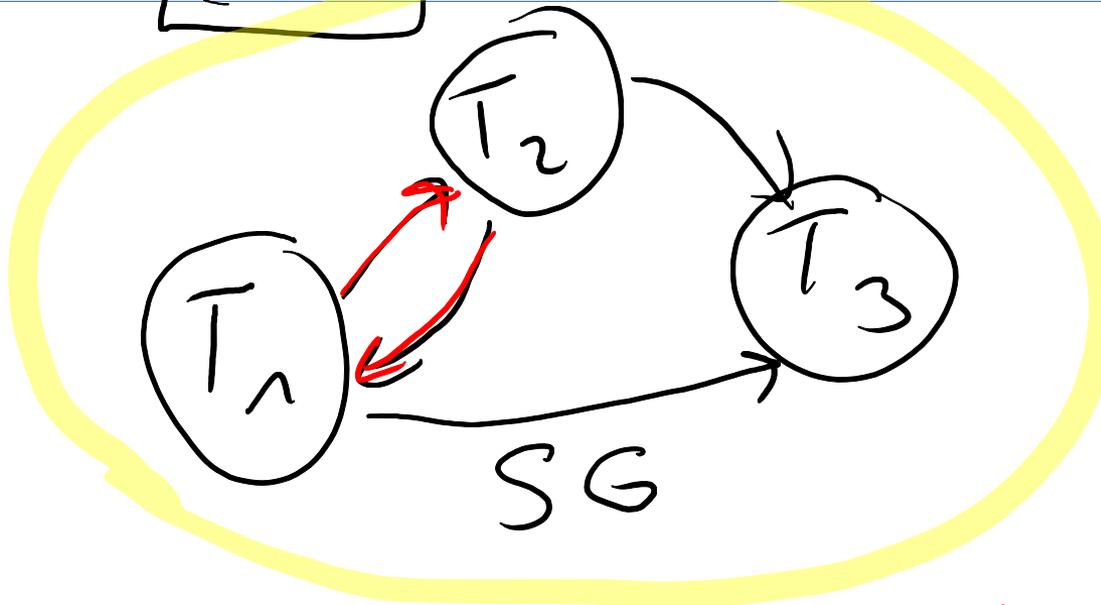
Multiuser - **Alles wichtig mit Fokus auf**

- Welche Probleme können entstehen
- Serialisierbarkeit / Historien (auch die **Historienklassen und Zuordnung**)
- Deadlocks (Gründe / Erkennung / Vermeidung)
- 2PC / TS / Optimistic Concurrency Control
- ...
- Nicht wichtig:
 - Isolation Level in SQL
 - Synchronisation von Bäumen

Serialisierbarkeit einer Historie:

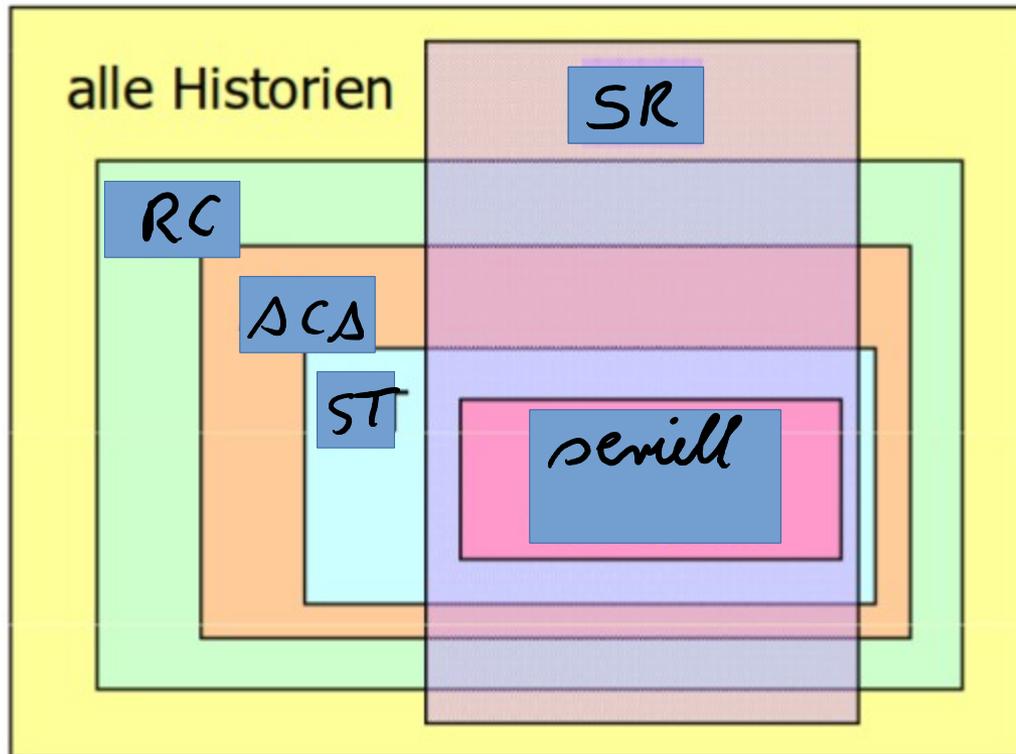
- Aufspannen des Serialisierbarkeitsgraphen bezüglich der Konfliktoperationen (read vor write, write vor read, write vor write)
- $w_1(A) \rightarrow r_3(A)$ der Historie H führt zur Kante $T_1 \rightarrow T_3$ des SG

Beispiel $H = w_1[x] w_2[x] w_2[y] c_2 w_1[y] w_3[x] w_3[y] c_3 w_1[z] c_1$



nicht serialisierbar

Frage: Füllen sie in folgendem Schaubild die verschiedenen Historienklassen ein



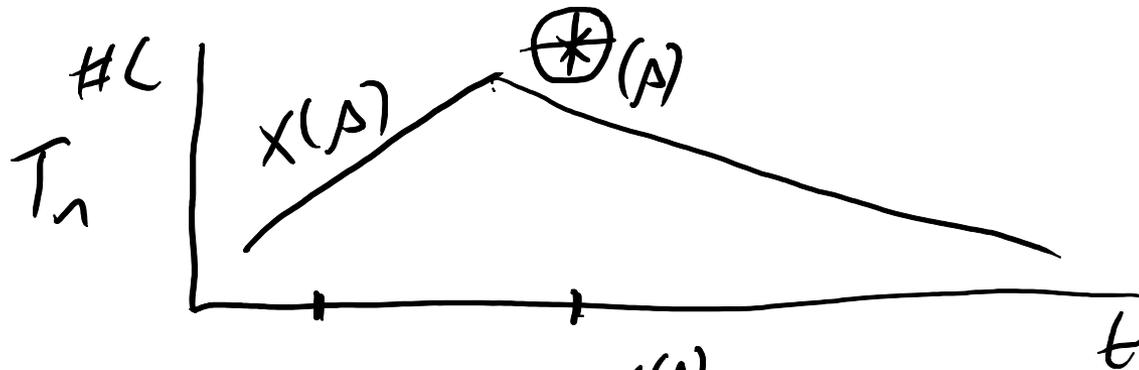
Frage: Welche Art von Sperren gibt es und wie sind sie zueinander kompatibel

exclusive (X)
 shared (S)

	NC	X	S
S	X	-	X
X	X	-	-

Frage: Kann es bei 2PL nicht rücksetzbaren Historien kommen?

Frage: Kann es bei 2PL zu nicht-serialisierbaren Schedules kommen?



Fokus auf Stoff in den Übungsblättern

- Angriffsarten
- k-Anonymität
- RSA (Verschlüsseln & Entschlüsseln, welche Voraussetzungen für Sicherheit?)
- SQL Injection
- TLS

- **Wurde in der Vorlesung nicht besprochen**
→ **Nicht prüfungsrelevant**

- **Wurde in der Vorlesung nicht besprochen**
→ **Nicht prüfungsrelevant**

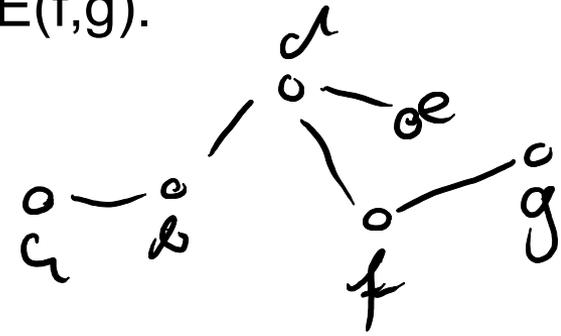
- Datalog Theorie
 - Wann ist Programm sicher?
 - Stratifizierbar?
 - ...
- **Datalog Programme! Wichtig!**
 - Definition neuer Regeln
 - \neq , not(...), +
 - Einfache Regeln zu SQL übersetzen
 - Rekursion
 - **Keine Aggregationsfunktionen!**

- Beispiel, Erreichbarkeit im Graphen

EDB: $E(a,b). E(b,d). E(d,e). E(d,f). E(f,g).$

IDB: $(1) R(X) :- E(a,X).$

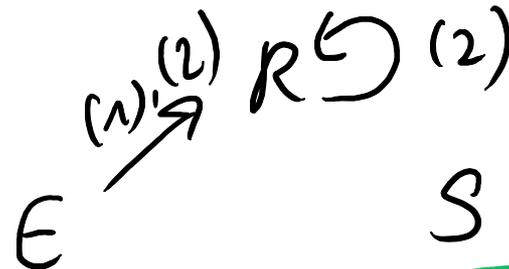
$(2) R(X) :- R(Z), E(Z,X).$



- Frage: Sind die Regeln sicher?

Ja

- Ist das Programm stratifizierbar?



Stratifizierbar!

- Beispiel, Erreichbarkeit im Graphen

EDB: $E(a,b). E(b,d). E(d,e). E(d,f). E(f,g).$

IDB: $R(X) :- E(a,X).$

$R(X) :- R(Z), E(Z,X).$

- Frage: Naive Auswertung

- Beispiel, Erreichbarkeit im Graphen

EDB: $E(a,b). E(b,d). E(d,e). E(d,f). E(f,g).$

IDB: $R(X) :- E(a,X).$

$R(X) :- R(Z), E(Z,X).$

- Frage: Naive Auswertung

- Beispiel, Erreichbarkeit im Graphen

EDB: $E(a,b). E(b,d). E(d,e). E(d,f). E(f,g).$

IDB: $R(X) :- E(a,X).$

$R(X) :- R(Z), E(Z,X).$

- Frage: Semi-Naive Auswertung

- Beispiel, Erreichbarkeit im Graphen

EDB: $E(a,b). E(b,d). E(d,e). E(d,f). E(f,g).$

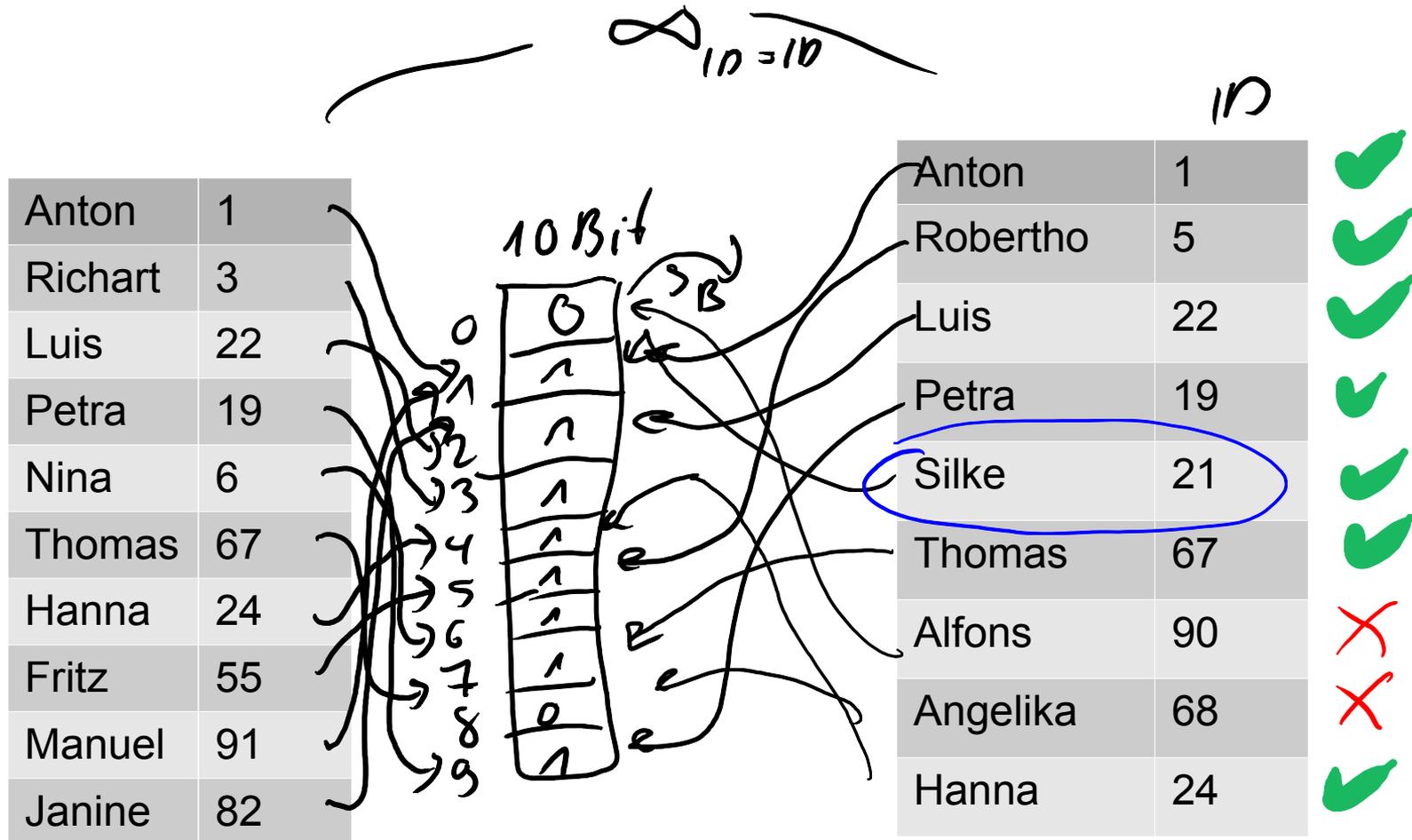
IDB: $R(X) :- E(a,X).$

$R(X) :- R(Z), E(Z,X).$

- Frage: Semi-Naive Auswertung

- **Fragmentierung**
 - Vertikal / Horizontal
 - Korrektheit
 - Rekonstruktion
 - Join Auswertung
- Bloom Filter
- **Chord**
- **Synchronisation: 2PC, Quorum,, ...**

- Semi-Join / Bloom Filter**



S_A

$h(x) = x \text{ mod } 10$

τ Hashfunktion

S_B

- ***Cube Operator***
 - Bezug zu Group By
- **Top-K Anfragen**
 - Threshold / NRA
 - Skyline (in SQL mit und ohne „skyline“ operator)
- **Klassifikation**
 - Entscheidungsbaum
 - **Assoziationsregeln** (Support / Konfidenz)

- **Speicherhierarchie**
- **Row vs Column Store**
- Indexstrukturen
 - ART

- Grundkenntnisse XML Schema
- **XPath**
 - Achsen
- **Xquery**
 - FLOWR
 - Falls Funktionen zur Lösung der Aufgaben benötigt werden, sind sie angegeben!

**Schwierigkeit der Übungsblätter kann als
Orientierung für die Prüfung genutzt werden**