

## Systemprogrammierung 2, 2001

SysProg 2  
Lehner  
SS 2001

1. Phasen der Anfragevewrarbeitung (Zeichnung aus dem Skript)
2. SQL-Anfragen erstellen und Anfragegraph zeichnen.
3. Berechnung der Zugriffszeit bzw. der "Kosten" bei Anfragen
4.
  - Sperren: Dirty Read als SQL-Anfrage in Kurzform formulieren
  - Deadlock mit EINER Relation und zwei Transaktionen
  - Verfahren um Rekursives zurücksetzen abgeschlossener Transaktioen bei einem Fehler zu verhindern.
  - Wo stehen Projektion, Selektion und Vereinigung bzw. Join im Operatorgraph (=Anfragegraph)?
5.
  - Warum ist kein Undo notwendig, nach dem einspielen eines Datendumps?
  - Phasen des 2-Phasen Sperrprotokolls
6. Schattenspeicherkonzept: Sicherungspunkt erstellen bei geöffneter Relation mit geänderten Seiten. Beschreibe das Vorgehen!

Dies ist nur eine Stichpunktartige Liste. Sie erhebt keinen Anspruch auf Vollständigkeit. Alle Angaben ohne Gewähr.

Sysprog 2  
Lehner  
Juli 2002

### Bemerkungen zu Prüfung und Prüfer

- Naja nachdem ich eigentlich die 500 Seiten so mehr oder weniger auswendig im Kopf hatte, meinte ich optimal vorbereitet zu

sein. Aber wir warn dann alle doch ziemlich von der Klausur überrascht: Es war keine reine auswendig-gelerntes-niederschrieben-Klausur sondern der Hauptteil bestand aus richtigen Transfer-Aufgaben. Dass der Meyer-Wegener auf die verwegene Idee gekommen war Relationale Algebra und ER-Modell in seine Klausur einzubeziehen hätte ich nicht gedacht.

### Fragen

Die Klausur bestand aus 11 Aufgaben:

- Es war eine Reihenfolge von Zahlen gegeben, welche man in einen B-Baum einfügen sollte.
- Wieder war eine Reihe von Zahlen gegeben, welche man diesmal per Hashing in die jeweiligen Buckets verteilen sollte. Es waren 8 Primär-Buckets gegeben in die jeweils 3 Sätze reinpassten - es wurde eine feste Satzlänge angenommen. Als Hashfunktion sollte man  $h(x) = (x \bmod p) + 1$  verwenden.
- Hierbei gings um Ersetzungsstrategien. Eine Reihe von Seitenzahlen war gegeben, wobei einige fett markiert waren, was zu bedeuten hatte, dass diese auch nach dem Einlagern verändert wurden. Nun sollte die Ersetzungsstrategie LFU angewendet und dementsprechend eine Tabelle vervollständigen. Die Tabelle bestand dabei aus den Attributen: Zeit, (Nummer der eingelagerten Seite an Stelle 1 im Puffer und Anzahl wie oft diese referenziert wurde) ... (Nummer der eingelagerten Seite an der Stelle 10 im Puffer und Anzahl der Referenzen), Anzahl der E/A Operationen. So dass z.B. wenn die Seite schonmal eingelagert war es 0 E/A Operationen gab und beim Neu-einlagern mit gleichzeitigem Verdrängen einer veränderten Seite (zuvor fett gedruckt) hingegen 2.
- Man musste überprüfen ob zwei Ausdrücke der Relationalen Algebra äquivalent sind.
- Dann gab es noch eine Aufgabe mit SQL Statements. Es gab die Relationen Nation, Spieler und Platzverweis. Bei der dritten Unteraufgabe musste man umgangssprachlich Formulieren, was ein SQL-Statement zurückliefert und

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du a uf unserer Homepage.
---

bei der vierten sollte man ein SQL-Statement mit Hilfe der Relationalen Algebra ausdrücken.

- Ferner kam noch eine Aufgabe zum Entity-Relationship Modell. Es sollte dabei folgende Miniwelt nachmodelliert werden: Also es gibt verschiedene Arten von Wertpapieren, aber jedes Wertpapier besteht aus einer Wertpapiernummer und einem Namen. Ausserdem gibt es überall Wertpapierbörsen in denen zwar nicht jedes Wertpapier gehandelt wird aber mindestens 50. Jede Börse bestand aus den Attributen Wertpapiernummer und Ort.
- Eine Aufgabe hatte das Thema "Transaktionen". Zuerst galt es die Frage zu beantworten wofür TA überhaupts gut sind und dann eine Tabelle zu vervollständigen. Die vordeste Spalte bestand aus Redo, Undo und die horizontalen Attribute waren BOT,EOT,After-Image, Before-Image. Hierbei sollte man ein "+" eintragen falls die jeweilige Protokollinformation notwendig war, eine "0" für egal und ein "-" dafür dass dies Protokollinformation hierbei nicht vorkommen darf.
- Ferner galt es die Frage zu beantworten welche Gründe es geben könnte ein Programm in mehrere Transaktionen zu unterteilen. Im Folgenden war ein C-SQL-Programm gegeben, und man musste eine BOT-EOT-Klammer so setzen dass die logische Konsistenz gewahrt blieb. Konkret, es ging um so ein Buchrückgabe-Programm, für jedes Buch das der Nutzer zurückgab wurde der Eintrag von dessen Buchkonto gelöscht und eventuell entsprechende Kosten bei zu spätem zurückgeben berechnet usw. Die BOT-EOT-Klammer wurde genau von der der while-Schleife umklammert.
- Man musste diverse Fragen zu C++ beantworten, z.b. was für eine Bedeutung das "public" bei "AbgeleiteteKlasse : public VaterKlasse {...}" hat und welche Attribute noch möglich wären. Ferner wurde das Attribut "virtual" abgeprüft. Hierzu wurde eine Codesnippet gegeben und man musste wissen, welche Implementierung einer Methode in dem jeweiligen Fall ausgeführt wurde. Zum Schluss sollte die getCardinality Methode des relationalen Operators CSelect implementiert werden. Hierzu sollte man annehmen dass im Durchschnitt

15Prädikat erfüllten. Zusätzlich sollte noch eine CException geschmissen werden, falls der Operator zuvor nicht geöffnet wurde.

- Hierbei musste der dazugehörige Executionplan zu einem vorgegebenen Ausführungsplan erstellt werden. - genau wie bei der Rechnerübung.

So oder so ähnlich wars. Viel Glück

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage.
--