

Prüfer : Prof. Jarke

Termin : 12.1.96

Fächer : Einführung - Implementierung DB

Betriebssystem von Silberschatz

Anfangen hat es wohl mit dem Betriebssystem, jedoch gab er die Frage nicht getrennt jensech die Thema sondern gemischt.

- Bakery-Algorithmus :
Das Problemstellung, die Idee des Algorithmus erklären. Jedoch wollte er nicht explizite Angabe des Algo.

- Semaphore :
Explizit aufschreiben und erklären; mit Busywaiting und die Vermeidung

- Deadlock :
Die Bedingungen und die Vermeidung, Verhindern, Erkennen und beseitigen.
Eine Vorlesung halten. Er stellt einige kleine Frage dabei.

- In DB-System :
Wie wird Deadlock in DB-System verhindert? In anderen Protokolle und Transaktionsverwaltung schauen.

- Short-term Scheduler :
Methode aufzählen und besonders mit dem Bild (siehe im Buch von Silberschatz) erklären, wie die Vorgang ist.

- 2PL-Scheduler :
Was es ist, welche Probleme da gibt's (Deadlock), Bild malen, in welchen Phase auftritt, ein Bsp von Schemale mit Deadlock angeben, die Lösungsweg... Hier muss man viel wissen.

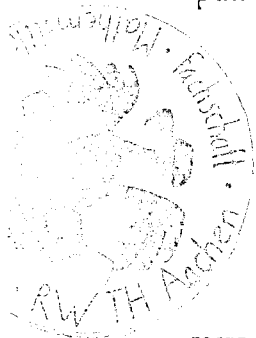
- B*-Baum :
Wie sieht er aus, die Kosten,wie übliche

- relationale Algebra :
alle sorte von Operatoren angeben

- Dekomposition :
zu beachten : Verlustlosigkeit, Abhängigkeitserhaltung. Die genaue Definition und
erklärung
Die Test-Methode.

Viele sagen, dass der Professor sein Stil geändert hat. Aber bleibt die Frage immer in selben Thema, finde ich. Ich habe kein einzige Frage gehabt, die nicht in Prüfungsprotokoll stand. Nur muss man ein bisschen tiefer kennen. viel glück.

Seiteneretzungsstrategie
'Pre Redung',





eine Frage ein?

B(eisitzer): Nöh.

J zu B: Sie haben doch auch Prüfung in Betriebssysteme gemacht. Fällt Ihnen da wirklich nichts ein?

B: Nöh, mir fällt wirklich nichts ein..

Eine Minute angestrengtes Nachdenken. Ich fühle mich

immer besser, da ich gehört hatte, daß Jarke nur mit dem

Beisitzer scherzt, wenn die Prüfung gut läuft. War mir bis

dahin nicht klar.

J: So was wie Scheduling gibt es beim Paging ja auch.

Welche Strategien gibt es denn, Seiten einzuladen.

T: Pure Paging, Prefetching. (+ Erklären)

J: Und wenn der Speicher dann mal voll ist, welche Stra-

tegien gibt es, Seiten zu verwerfen?

T: FIFO. (Wollte gerade erklären und dann weitere aufzählen,

da..)

J: Wie implementiert man denn FIFO, billig.

T: Mit rundaufendem Zeiger, der immer auf den nächsten

Kandidaten zeigt, der dann rausgeworfen wird.

J: Welche Algs. kennen sie noch.

T: Opt: Tut's aber nicht, da Zukunft nicht bekannt. (Den

Kram mit Vorhersage mittels alten Werten habe ich mir ge-

spart, da es den Eindruck machte, als ob Jarke fertig wer-

den wollte.) LRU, teuer wegen Verwaltungsaufwand. Habe

dann noch second chance erklärt. Nach der Erklärung fiel

mir dann ein: Das nennt man Second Chance.

J: Danke das war's.

(Es waren jetzt erst 45 Minuten um!)



T: INSERT/UPDATE/DELETE-Anomalien erklärt. Auch einige Zeilen für die Relation angegeben.
J: Wie verhindern Sie das?
T: Zwei Relationen hingeschrieben...
J: Was kann denn da sonst noch so passieren?
T: Er wollte auf nicht verlustfrei und nicht abhängigkeits-erhaltend raus.
J: Wie kann man verlustfreiheit nachweisen?
T: Tableaumethode. Irgendein Beispiel aus den Fingern gesogen. hat auch nicht weiter gestört, daß ich die FDs noch ein wenig angepaßt habe, damit mein Beispiel verlustfrei wird ;-)
J: Wie findet man raus, ob eine Zerlegung abhängigkeits-erhaltend ist.
T: Abschlußberechnung. Habe keinen formalen Algorithmus hingeschrieben, sondern nur prosa erzählt. Mit ein paar Zwischenfragen, die mir sehr weitergeholfen haben, hat er mich immer wieder auf den rechten Weg zurück gebracht.
Hier habe ich besonders viel gestockt.
J: ja dann müssen wir jetzt noch Betriebssysteme prüfen.
Nicht mehr zu Einführung? Na gut ist jetzt auch egal.

4 Betriebssysteme

J: Was frage ich sie denn jetzt. Nach Semaphoren habe ich die beiden anderen schon gefragt, CPU-Scheduling. Malen sie mir mal die Zustände von so einem Prozeß hin. Mit Übergängen.
T: Hingemalt, Übergänge erklärt.



J: Für das CPU-Scheduling gibt es ja zwei Ansätze. . .
T: Preemptive und Non-Preemptive Scheduling
J: Ja, beim preemptiven (sp?) Scheduling. Welches ist da der beste Ansatz.
T: SJF.
J: Hinsichtlich welches Kriteriums?
T: Hinsichtlich kürzester Wartezeit.
J: *Durchschnittlicher* kürzester Wartezeit. Kann man das beweisen?
T: Ja.
J: Ja dann tun Sie daß mal.
T: (Überleg.) Da kann man konstruktiv zeigen. (Hab dann so'n Bildchen gemalt wie im Silberschatz) Wenn irgendein längerer Prozeß weiter vorne wäre, würden mehr Prozessanfänge nach hinten rutschen. . .
J: Das kann man durch Widerspruch zeigen. Und welcher Alg. ist beim non-preemptive Scheduling am besten?
T: (Keine Ahnung. Habe wohl ziemlich doof geguckt. Überleg. . .) Auch SJF.
J: (War wohl nicht zufrieden. Stimmt ja auch nicht.) Über legen Sie mal. Sie haben einen Job mit 5 Sekunden, einen mit 7 und einen mit 10. Die führen Sie in dieser Reihenfolge aus (SJF). Während der erste Job läuft, kommt einer mit drei Sekunden. Unterbrechen Sie jetzt, oder nicht?
T: (Jetzt kam die große Raterie. Hab wohl nichts richtiges gesagt. Er hat es mir noch zwei dreimal erklärt bis bei mir der Groschen fiel) Shortest Remaining Time First.
J: Hmmm, jetzt muß ich Ihnen noch eine Frage zu Betriebssystemsystemen stellen (Zum Beisitzer:) Fällt Ihnen noch

delltheoretische Semantik. (Hoffentlich fragt er nicht, was das ist, hoffentlich fragt er nicht, was das ist, hoffentlich fragt er nicht, was das ist. . .)

J: Erklären Sie mal wie das geht.

T: Tja, da hab' ich mir ganz schön einen einen abgestottert. Vonwegen Interpretation der Fakten und Regeln und so. Bin irgendwie beim alten Herbrand gelandet. Er hat da ganz schön geböhrt. Wollte irgendwie drauf hinaus, daß es ein *minimales Modell* gibt und daß das auch noch endlich ist. Alles in allem hatte ich hier ein ganz schön schlechtes Gefühl. Naja, war wohl unbegründet.

J: (Irgendwann war's J. dann wohl genug.) Sie haben eben gesagt, daß es drei Möglichkeiten gibt. . .

T: Da gibt es noch die *algorithmische Semantik* (keine Ahnung, ob das wirklich so heißt). Habe erklärt, wie man aus Fakten und Regeln neue Fakten generieren kann. Leider habe ich mich dabei ein wenig verplappert. habe ihm erzählt, daß dabei die gleiche Menge wie bei der modelltheoretischen Semantik herauskäme.

J: Da wollen wir mal nicht hoffen, daß sie da recht haben. Man will doch nicht bei jeder Anfrage alle generierbaren Fakten herausbekommen.

T: (Habe *ich* das gesagt? Oh ohh.) Ihm irgendwas davon erzählt, wie das genauer abgeht. Anfrage besteht aus einem Prädikat, das mit Konstanten oder Variablen instanziiert wird. Zuerst versucht, das Prädikat mit einem bekannten Fakt zu matchen, klappt das nicht. . . (Forward-Chaining erklärt bin dann drauf gekommen, daß sobald ein Match da ist keine weiteren Fakten erzeugt werden. Irgendwann





habe ich dann noch was von einer Heuristik erzählt, die auf Prädikatsabhängigkeitsgraphen beruht — siehe auch Elmasri-Navathe Datalogkapitel. Jarke *liebt* Datalog.)
 J: Und das nennt man dann Forward-Chaining. Da war aber noch eine Möglichkeit...
 T: (Schluck, wie hieß die denn noch) Die *syntaktische Semantik* (noch weniger Ahnung, ob das so stimmt). Habe noch irgendwas erzählt, war aber nicht so toll, glaube ich. Dann gehen wir mal über zur Einführung in Datenbanken...
 Hier mußte ich dann schlucken. Es waren laut Jarke Uhr erst 20 Minuten vergangen.

3 Einführung Datenbanken

Einführung in Datenbanken war das Fach, wo ich am besten vorbereitet war. J: Sie haben da einen Satz Relationen und einige funktionale Abhängigkeiten. Wie kann man die Relationen schöner machen?
 T: Über erste und zweite in dritte NF bzw. BCNF bringen.
 J: Schreiben Sie mir mal eine Relation auf, die nicht in 2NF ist.

T: EmpDepHours (SSN, Name, Dept, Hours)
 Abhängigkeiten:



- SSN → Name
- SSN, Dept → Hours

J: Was kann da denn sonst so passieren?

Gibt es Gründe auch aus der 1:N Beziehung eine Relation zu machen?

Regeln erläutert, durchgeführt, u.a. 1:N Beziehung mit foreign key auf der N-Seite

Modell.

Zurück zu den klassischen Modellen. Transformieren sie das ER-Diagramm in ein relationales

drei Arten erläutert: ad hoc p., inclusion p., bound p.

Was ist Polymorphie?

Spezialisierung, Verfeinerung vom Op., Mehrfachvererbung

b) Keine Operationen auf Daten (OO: strukturelle Repräsentation + Verhaltensspezifikation)
 c) keine Generalisierung/Spezialisierung, sollte diese Begriffe genau erläutern, Arten von

Und was noch?

EER, am Diagramm erklärt

Wie kann man dieses doch mit ER-Diagrammen modellieren?

a) keine Aggregation, Verteilung von phy. Obj. auf mehrere Entitäten (Segmentation)

Was sind die Kritikpunkte dieser Modellierung aus der objektorientierten Gemeinde?

cardinality ratio bei Beziehungen, participation constraints (total, partial)

Was für Constraints gibt es?

Entitäten def., Beziehungstypen (1:1, 1:n, M:N), Attribute beider, usw.

(Zeichnet ein ER-Diagramm) Erklären sie die Komponenten.

ten Modellierung)

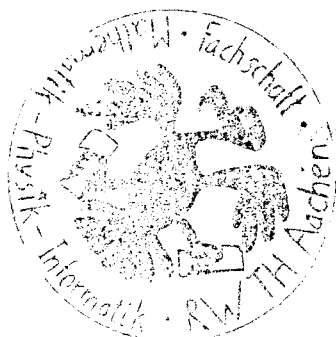
Entity-Relationship Diagramme (und evtl. SA zur Datenfluspez., OMT zur objektorientier-

Wie entwirft man eine Datenbank? Welche Modellierungssprachen kennen sie?

Entf.: DB/OODBS

Fach:	Vertiefungsgebiet: Datenbanken
Datum:	Oktober 1994
Dauer:	ca. 45 Minuten
Vorlesungen:	Einführung in Datenbanken (Jarke) Implementierung von Datenbanken (Kemper) Objekt-orientierte Inf.verwaltung f. ingenieurwis. Anwendungen (Kemper)
Bemerkungen:	Angenehmes Prüfungsgespräch; viele Zwischenfragen, die nicht alle im Protokoll mit aufgenommen werden konnten; die Fragen scheinen sich aus dem Gespräch zu ergeben: Prüfer greift vom Prüfling gegebene Stichwörter auf und errät genauere Erläuterungen

Gedächtnisprotokoll



No. (Leider doch: wenn auf nur wenige Entitäten der N-Seite eine Beziehung zu der I-Seite besitzen, um die vielen NULL-Werte zu vermeiden)

(1:N Fall, dessen Entitäten bisher nur die Bezeichner A und C besaßen, werden – auch zur Erläuterung des obigen Problems – in Gefängnis und Person umgetauft.)

Wie formuliert man auf den Relationen Gefängnis und Person Anfragen in SQL?

`SELECT FROM WHERE` erläutert, einfacher Join und eine Select-Bedingung angeben

Bestimmen sie das Durchschnittsalter der Insassen der einzelnen Gefängnisse.

`AVG()` Fkt. und `GROUPED BY` hinzugefügt.

Implementierung

Wie würde eine solche Anfrage in einer Datenbank mit Schichtenarchitektur ausgewertet?

5 Schichten und Schnittstellen (6) erläutert, ext. Satzchnittstelle wird evtl. nicht benötigt.

Warum eigentlich eine Segmentchnittstelle, wenn es doch schon im BS eine virtuelle Speicher-verwaltung gibt?

Portabilität, eigene Informations- und Log-Strukturen sowie Strategien für Seitenersetzung, Transaktions- und Recovery-Unterstützung

Was für Seitenersetzungsstrategien kennen sie?

`LRU`, `CLOCK`, `GCLOCK`, `LRD`

Kennen sie LRU-K?

Nein.

Nicht schlimm. Ganz neue Technik, die gerade den DB-Markt revolutioniert. :-)

Was für Arten von Recovery gibt es?

R1 – R4 definiert und kurz erläutert

Erläutern sie die Variante R1 oder R2 genauer.

R2: Unterscheidung physischer (after images) und logischer Protokollierungsverfahren (Nachvollziehen der in LOG gescheicherten DML-Befehle), Kombination beider Verf. im Schattenspeicherkonzept

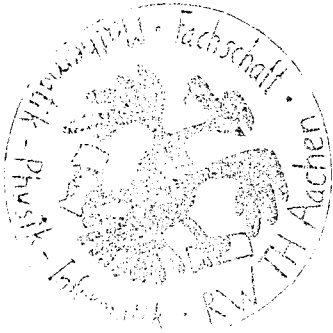
Wie implementiert man einen Join?

a) Nested Loop

b) unter Zuhilfenahme einer Zugriffsstruktur für Join-Attribut

c) Sort-Merge Join

d) Hash Join



Verteilte Datenbanken:
Was kommt beim Entwurf einer verteilten DB im Vergleich zu einer zentr. DB hinzu?

Fragmentierung, Allokation

Was ist Fragmentierung? Was ist Allokation?

Wie verläuft die Anfrageauswertung in verteilten DB?

Wie kann man eine Anfrage in vert. DB optimieren?

Semi-join

Wie geht man bei der Fragmentierung vor?

Mintermalalgorithmus

Wie geht man bei der Allokation vor?

Optimierungsproblem

Kosten-Nutzen-Analyse, vor allem bei der Frage, ob Fragmente

redundant abgespeichert werden sollen.

Konzeptuelle Modellierung:

Was ist ein ER-Modell? Welche Komponenten hat es?

Welche Vorteile hat es im Vergleich zum relationalen Datenmodell?

Welche Ausdrucksmöglichkeiten kommen bei objektorientierter Modellierung in

Telos hinzu?

Is-a- und In-Kanten

Jarke gibt ein Telos-Modell vor: Person mit zwei Attributen PA-Nr. und Name;

Empl. mit Attributen E# und Gehalt; Is-a-Kante von Empl. nach Person. Wie

kann man dies in ein relationales Datenmodell umwandeln?

Zwei Tabellen hingedeichnet, eine für Person mit zwei Attributen,

eine für Empl. mit vier Attributen

Wie kann man diese normalisieren?

1.-3. Normalform

Danach sieht man, daß Tabelle für Empl. mit drei Attributen

(PA-Nr., E#, Gehalt) auskommt.



(Jarke gab 2 Relationen vor : Angestellte, Bereiche)

In SQL eine Abfrage angeben, die alle Angestellten aus dem Bereich "Aachen" realisiert !

Erklären wie Selektion, wie Projektion, wie Join in SQL formuliert !

Gibt es bestimmte Bedingungen für die mengentheor. Operatoren ?
Join : identische Attributmengen

Kann der mengenth. Operator Vereinigung auf 2 bel. Relationen angewandt werden ?

Nein

Erklärung des relationalen Kalküls !

Tupelkalkül; Tupelvariable, Bedingungen, wohlgeformte Formeln

Wie wird eine SQL-Anfrage implementiert ?

Sortier- und Mischformen; Indexverwaltung

Aufgaben des Process Managements !

Was ist ein Prozeß ?

Wie wird Prozeßwechsel durchgeführt ?

Was ist ein Deadlock ?

Wie lauten die 4 notwendigen Bedingungen für einen Deadlock ?

Wie verhindert oder beugt man Deadlocks vor ?

Welche Arten von Speicher-Management gibt es ?

Welches ist die am häufigsten verwandte Methode ?
demand paging

Erklärung der verschiedenen Seitenauswahlstrategien !

Anmerkung :

Es war ein Nachteil, die Vorlesung Datenbanken nicht bei Prof. Jarke zu hören, denn er kannte die Inhalte der Walter-Vorlesung nicht und bezog sich mehr auf seine Vorlesung bei der Fragestellung.

Kennen Sie ein Betriebssystem, wo ein bestimmtes (welches?) angewandt

Welche Seitenersatzstrategien gibt es? Erklärung.

Demand Paging

Wie wird es realisiert?

Was ist Virtual Memory?

Noch was zu Betriebssystemen

wird?

Wie werden Sperren gesetzt, damit nicht der gesamte Baum blockiert

B-Bäume

Bei welchen Speicherungsstrukturen wird das gemacht?

vermeiden?

Was kann man beim 2-Phasen-Sperreprotokoll tun, um Deadlocks zu

Welches sind die vier Bedingungen für einen Deadlock? Erläuterung.

werden können?

Was ist wenn nach der Freigabe von Sperren wieder Sperren angefordert

Wieso werden Probleme dadurch gelöst?

Sperren setzen und 2-Phasen-Sperreprotokoll.

Womit erreicht man das?

gleiches Ergebnis wie bei serieller Abfolge.

Was ist serialisierbar?

Welche Probleme treten bei parallel laufenden Transaktionen auf?

Wie wird der Join implementiert?

Überleitung zur Implementierung

Wie kann man diese Anfrage optimieren?

Die Anfrage in relationaler Algebra hinschreiben.

Welche SQL-Statements entsprechen welchen algebraischen Operatoren?

Welche Arten von Antragesprachen gibt es?

Alle Angestellten, die in Aachen wohnen und in der Forschung arbeiten.

Folgende Anfrage in SQL hinschreiben:

Synthese-Algorithmus kurz erklären.

Jetzt wird es etwas chaotisch, da Prof. Jarke seinen Spaß am Detail zu entdecken scheint. Was er hören will und wir dann "gemeinsam entwickeln" ist folgendes:

Nachdem ein Hardwareinterrupt (z.B. Timer- oder I/O-Interrupt) erfolgte - solche muß es geben, damit ein Prozeß nicht die ganze Zeit die CPU belegt - wird von einem

Hardwaremechanismus der aktuelle Inhalt des Befehlszählerregisters gerettet, dann der Befehlszähler auf die Startadresse einer Interruptroutine gesetzt, die dafür sorgt, daß der

gerettete Befehlszähler sowie die Registerinhalte bzgl. des unterbrochenen Prozesses in dessen PCB gespeichert werden. Dabei darf die Interrupt-Routine natürlich nicht die Register vor der

Rettung deren Inhalte überschreiben. Schließlich bestimmt der Scheduler, welcher Prozeß als nächster die CPU erhält. Die Register werden mit den im PCB gespeicherten Werten

geladen und der Befehlszähler mit der Adresse geladen, an der der Prozeß unterbrochen wurde (auch im PCB enthalten).

Prof. Jarke fragt, ob neben den Register- und dem Befehlszählerinhalt nicht noch etwas zu retten ist. Mir fällt dazu nichts ein, und er sagt schließlich: der Page-Table-Inhalt.

Prof. Jarke setzt nun zum Finale an und sagt:

Frage: *Mit welcher Scheduling-Politik erreicht man denn garantiert die kürzesten Wartezeiten in der ready-Schlange?*

Glücklicherweise fällt mir hier Abb. 4.12 ein und ich sage: Shortest Job First.

Alle Bemerkungen zur Prüfung:

Die Prüfung war ziemlich fair, wenn auch Prof. Jarke, nachdem er gemerkt haben muß, daß ich B-Stern-Bäume nicht besonders gut beherrsche, eher damit hätte Schluß machen können. Die Prüfungsatmosphäre war ruhig, Prof. Jarke wurde nie hektisch oder ungeduldig. Die Benotung empfand ich als ok.

Gedächtnisprotokoll Diplomprüfung in Praktischer Informatik

bei: Prof. Jorke

am: 21. April 1993

Dauer: exakt 3/4 Std.

Fächer: Einführung in Datenbanken

Implementierung von Datenbanken (Vorlesung Kemper)

Betriebssysteme (Buch Silberschatz ...)

Fragen:

Erklärung des Entity-Relationship-Modells.

Beispiel aufmalen für:

Datei Angestellte mit Name und Adresse,

Datei Abteilung mit Nummer und Titel

und Beziehung gehört-zu mit Einstellungsdatum

Jeder Angestellte gehört eindeutig zu einer Abteilung

Umsetzung dieses Beispiels in das relationale Datenbankmodell.

Wie kann man dies noch vereinfachen?

Relationen mit gleichen Schlüsseln zusammenfassen.

Jetzt betrachten wir das Modell mal als n:m - Beziehung.

Welche Probleme treten auf?

Anomalien, Redundanzen

Welche Normalform haben wir hier vorliegen?

Warum? Welche Abhängigkeit stört?

Was kann man nun dagegen machen?

Was ist Dekomposition?

Was soll denn bei der Normalisierung erhalten bleiben?

Erklärung von *verlustlos* und *abhängigkeitserhaltend*.

Wie kann man dies beweisen?

Tableaumethode für Verlustlosigkeit (mit Erklärung)

Abschlussberechnung für Abhängigkeitserhaltung

Wie berechnet man den Abschluss?

mit den Armstrong Axiomen

Prüfungsprotokoll Praktische Informatik

bei Prof. Jarke

Fächer: **Einführung in Datenbanken** (Vorlesung im SS 93 ???), weitere Literatur: Elmarsi/Navathe - siehe Lit.verzeichnis, empfehlenswert, Elmarsi/Navathe - siehe Lit.verz., empfehlenswert, auf deutsch, bei Netzwerkmodell will Jarke aber mehr als bei Vossen steht, besser Elmarsi dafür man glaubt es nicht - die Vorlesungskopien mit einer Menge guten Willen **Betriebssysteme** - Silberschatz: Operating Systems Concepts oder so. 4. Auflage, das mit den Dinosauriern, Kapitel 1-9 3. oder 4. Auflage ist egal, Jarke fragt nur das, was er kennt und das scheint nicht die 4. Auflage zu sein (ohne Gewähr!) **Expertensysteme** - Gottlob, ..., Neidl: „Expertensysteme“ **Achtung:** Jarke weiß so genau nicht, was in dem Buch so steht und fragt deshalb lieber PROLOG und seine Grundlagen, Herbrandtheorie usw. ab. Wenn man vorher Logikprogrammierung gehört/gelernt hat, ist das wohl ok, aber sonst sollte man aber über fundiertes Wissen aus Mathe. Logik verfügen. Das ist natürlich meine persönliche Meinung, aber bis jetzt drängt sich dieser Eindruck auf.

Dauer: ca. 45 min

Datum: 4.8.1995

Note: 1,3

Allgemein:

Die Antworten sind Stichworte und geben nicht immer das wieder, was ich gesagt habe, sondern das, was gefragt war (soweit ich das weiß). Bei Expertensysteme habe ich mich bemüht, die Fragen mit meinen Antworten aufzuschreiben. Die richtigen Antworten kann man meist den Fragen von J. entnehmen.

Betriebssysteme:

J: *Prozesssynchronisation, welche Probleme gibt es?*

-Möglichkeiten der Prozesssynchronisation (gemeinsamer Speicher, Briefkasten), bei gemeinsamen Speicher Probleme des gegenseitigen Ausschlusses. J will noch etwas hören, aber was???

J: *Da fehlt was, eine Folge von dem, was sie gesagt haben?*

Deadlocks, er will Deadlocks hören, kann er haben!

J: *Was sind Deadlocks, was kann man gegen sie tun?*

siehe Silberschatz, quasi alles nachzählen, er mag Deadlock-Prevention, also kein Bankers Alg., sondern eine der 4 Bedingungen darf nicht zutreffen.

J: *Wie sieht die Speicherverwaltung bei modernen BS aus?*

Paging & Demand Paging

J: *Erklären Sie Demand Paging genauer.*

Habe Bildchen aus Silberschatz gemalt und erklärt. Bei Demand Paging gibt es 2 Probleme: Seitenersetzungsstrategien und Frame Allocation

J: *Was gibt es denn für Verfahren bei Frame Allocation?*

Working Set und Pagefault Frequency erklärt

Einführung in DB:

J: Was hat Paging mit DBs zu tun?
 Indexstrukturen & Blöcke sollen Framegröße haben. Weil J. immer auf B*-Bäume raus will, sie als Beispiel für Indexstruktur verwendet.
J: Erklären Sie B-Bäume!*
 Wiederholen der Schlüssel in Blättern, Mindestknotengröße, Mindestzahl an Zeigern in Knoten
J: Kosten bei Zugriff?
 Nach Umdrehen der Ungleichheitszeichen durch den Beisitzer:
 $\log_2 N \geq C \Rightarrow \log_2 N; N = \# \text{Elemente}$
J: Was sind das für Kosten? Kosten bei Einfügen?
 Kosten, um einen Verweis auf einen Datensatz zu finden, Einfügen, äh, wenn das Blatt nicht voll ist, die Suchkosten, die hat man immer plus in Knoten einfügen = $O(1)$, also C von oben.
J: Und wenn das Blatt voll ist?
 Genau das wollte ich nicht beantworten, stoter & stammel, da muß das Blatt aufgeteilt werden, das kann sich rekursiv nach oben fortsetzen und eine neue Ebene einühren, also: Suchkosten + Blatt aufspalten für alle C Ebenen + 1 neue Ebene dazu + neuen Blatteintrag durch alle C+1 Ebenen runterreichen, macht so in etwa $3C$.
J: Naja, so in etwa. (Das relativiert den Wert meiner Aussage doch ziemlich stark, dieselbe Frage ist übrigens in einer späteren Prüfung noch mal gekommen)
Was ist ein ER-Diagramm?
 Attribute, Entitäten, Beziehungen, Kardinalitäten, Verwendungszweck ...
J: Ein Beispiel. Der Beisitzer muß sich ein Beispiel aus dem Kreuz leihen: Ein Hospital hat mehrere Ärzte, die mehrere Patienten behandeln. Die Patienten liegen in einem Zimmer und leiden an mehreren Krankheiten. Eine Krankenschwester überwacht mehrere Zimmer.
 Machen sie daraus ein ER-Diagramm!
 Alle sind Entitäten, die max. in 1:n Beziehung stehen. Es gab mal eine Übung mit einem erweiterten Hospitalbsp., siehe dort
J: Machen sie daraus ein Netzwerkmodell!
 Gesagt, getan. Hier zahlte es sich aus, daß nur 1:n Beziehungen vorkamen.
J: Und jetzt dazu eine Anfrage. Geben Sie alle Namen der Ärzte aus, die Patienten behandeln, die Cholera haben und vor 1930 geboren worden sind.
 Voll erwischt! Keine Ahnung von Netzansprachen, fasel irgendetwas von get first, get next (Jarke: genau!). Schreibe einen Ausdruck, der stark an das Domänenkalkül erinnert, hin und übersehe meine eigenes Netzwerkmodell völlig. (Krankheiten sind auf einmal Attribute von Patienten)
J: Sehr deklarativ hingeschrieben! Schreiben Sie die Armstrongaxiome auf!
 Gerne, die kenne ich wenigstens
J: Wie berechnet man lossless-join?
 Tableau-Methode erklärt.

Expertensysteme:
J: Was hat die Tableau-Methode mit Expertensystemen zu tun?
 Was will er denn jetzt? T.-Meth. erinnert an Forward Chaining, aber das ist wohl zu naheliegend und einfach, ich rede mir selbst ein, daß Bachward Chaining gemeint sein müßte und erzähle es J
J: Falsch! Ich wollte Forward Chaining hören (Tralalala.. Ich kam mir reichlich doof vor).
 Was ist ein EFRS?
 alles erklärt, keine Negation, Funktionssymbole usw.
J: Semantik von EFRS?
 Herbrandmodell erklärt, Cons(s), Achtung, das Buch ist hier sehr ungenau, es empfiehlt sich über Herbrand und seine Theorie mehr zu wissen, siehe z.B. Mathe. Logik oder den Schöningh
J: Wie berechnet man Cons(s)?
 Forward Chaining erklären

Gedächtnis-Protokoll einer Diplompriifung Praktische Informatik

Prüfer: Prof. Jarke

Termin: 15. Februar 1993

Fächer und Referenzen: Betriebssysteme (Silberschatz, Peterson, Galvin: Operating System Concepts), Datenbanken (Frei nach Jarke), Wissensrepräsentation (Geneserth, Nilsson: Logische Grundlagen der KI)

Dauer: ca. 55 min.

Note: 1,0

Datenbanken

Jarke: Definition der Prädiikatenlogik erster Stufe?

PL1!

Ich: Definition in pseudo-EBNF hingeschrieben.

Jarke: Wo spielt die PL1 bei den Datenbanken eine Rolle?

Ich: Relationales Kalkül: Tupel- und Domänenvarianter erwähnt.

Jarke: Wie kann man das relationale Kalkül definieren?

Ich: Induktive Definition von Anfragen im Tupel-Kalkül aufgeschrieben.

Jarke: Wie unterscheiden sich Tupel- und Domänenkalkül bezüglich der PL1?

Ich: In den Wertebereichen der Variablen.

Jarke: Wie kann man Anfragen des relationalen Kalküls durchführen?

Ich: Übertragung in relationale Algebra.

Jarke: Wie ist die definiert?

Ich: Definition der Grundoperationen sowie einiger darauf aufbauender Operationen aufgeschrieben.

Jarke: Und wie sieht die Übertragung des Kalküls in diese Algebra aus?

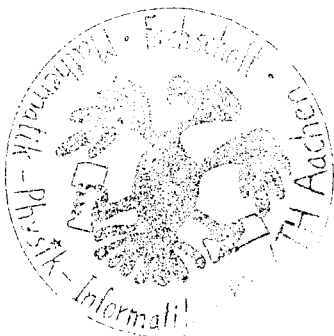
Ich: (Geriet etwas ins Schwitzen.)

Jarke: Nur exemplarisch: also stellen Sie sich vor... (murmelte etwas von „Existenzquantoren sollten schon drin sein“) und malte zwei Relationenschemata mit einem gemeinsamen Attribut auf. Wie sieht die Anfrage (irgendeine einfache Anfrage mit einem Join) aus? Erstmals in SQL.

Ich: Hingeschrieben.

Jarke: Und wie formuliert man das im Tupelkalkül?

Ich: Auch hingeschrieben.



PL1

Jarke: Wann möchte man beim Datenbankanwurf unter Umständen die Dekomposition vermeiden?

Ich: Denk, denk, aber: Weiß nicht.

Jarke: Viele Anfragen, wenige Änderungen: teure Join-Operationen werden vermieden.

Ich: (Ja, was soll ich dazu sagen?)

Wissensrepräsentation

Jarke: Wo liegt die Schwierigkeit bei Wissensbanken im Gegensatz zu Datenbanken?

Ich: (Erster Versuch mit Nichtmonotonie abgewürgt; dann mit vielen Tipps: Auch die Fakten und Regeln selbst liegen in PL1 vor.)

Jarke: Was ist nichtmonotones Schließen?

Ich: (Nach ein paar warmen Worten zum Warum & Wieso) CWA, Prädikatenvervollständigung und Zirkumskription aufgeführt. Zirkumskription genauer definiert (Abschlussformel).

Jarke: Wie sieht das konkret aus?

Ich: Trivialbeispiel aufgeschrieben (Vogel können fliegen vervollständigt zu Flieger können fliegen oder so).

Betriebssysteme

Jarke: Welche Arten von Prozessstatus gibt es: wie arbeiten sie zusammen?

Ich: Diagrammchen mit running-, waiting- und ready-Knödeln gemalt.

Jarke: Wie koordiniert das Betriebssystem das?

Ich: Kurzreferat über System Calls, Interrupts, verschiedene Queues usw. gehalten. Ich hatte das Gefühl, daß er eigentlich etwas anderes hören wollte, aber er wurde jetzt ziemlich hektisch (es waren schon etwa 50 Minuten gelaufen) und ich kam vor lauter Fragen kaum noch zum Antworten.

Jarke: Was ist denn ein Deadlock? (Die Frage scheint er gerne zu stellen.)

Ich: (Versuche etwas allgemeiner anzusetzen)

Jarke: Da gibt es doch Bedingungen ...

Ich: Die vier Bedingungen erwähnt (zum Erklären ließ er mir schon keine Zeit mehr).



Jarke: Und wie übertragt man diese Anfrage in die Algebra?

Ich: Grob skizziert: Konjunktion von Existenzquantoren in kartesischer Produkte, Bedingungen an existenzquantifizierte Variablen in Selektion, Bedingungen an die Anfragevariable in Projektionen etc. Resultat nochmal hingeschrieben.

Jarke: Wo spielt die PLI bei den Datenbanken noch eine Rolle?

Ich: (Grübel, grübel) Bei den Constraints?

Jarke: Ja, Constraints kann man als PLI-Formeln auffassen, aber wie formuliert man Constraints bei beim Datenbankenwert?

Ich: (Mit einigen Hilfestellungen und Tips) Mit funktionalen Abhängigkeiten.

Jarke: Wie ist die funktionale Abhängigkeit definiert?

Ich: Definition hingeschrieben.

Jarke: Wo treten die Abhängigkeiten beim Entwurf auf? Also, welche Eigenschaften sind für einen Entwurf wünschenswert?

Ich: Verlustlosen Join und Erhaltung der funktionalen Abhängigkeiten definiert: Gewährleistung von Redundanzfreiheit durch Normalformen erwähnt.

Jarke: Wie ist der Abschluß (den ich bei der Definition der Abhängigkeitserhaltung erwähnt hatte) definiert?

Ich: Als Abschluß unter den Armstrong-Axiomen.

Jarke: Wie lauten die?

Ich: Erstes Axiom hingeschrieben.

Jarke: (Unterbricht:) Na, dann glaube ich Ihnen mal, daß Sie die anderen auch können. Wie beweist man die Axiome?

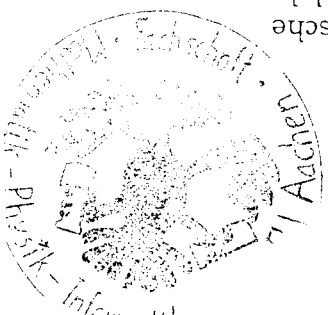
Ich: Direkt mit der Definition der funktionalen Abhängigkeit.

Jarke: Welche Verfahren zum Datenbankenwert gibt es?

Ich: (Nachdem ein Ansatz mit Normalformen abgewürgt wurde:) Komposition und Dekomposition von Relationenschemata.

Jarke: Wie verhalten sich die Verfahren bezüglich der erwünschten Eigenschaften?

Ich: Die Komposition erhält die funktionalen Abhängigkeiten: die Dekomposition den verlustlosen Join.



Prüfungsprotokoll

Praktische Informatik bei Prof. Jarke

vom 20.01.'94

Themengebiete:

- Compilerbau: Als Grundlage galt das Drachenbuch (Aho/Sethi/Ullmann), jedoch ohne Thema Codeoptimierung.
- Betriebssysteme: Als Grundlage galt die Vorlesung von Professor Haupt (WS91/92, SS92), obwohl ich eher den Eindruck hatte, daß er nach dem Buch von Silberschatz geprüft hat.
- Datenbanken: Hier galt Jarke's Skript als Grundlage. Es empfiehlt sich jedoch, sich mit zusätzlicher Literatur zu versorgen (Date, Teorey/Frey, Ullmann)

Organisatorisches:

- Prüfungsdauer: so zwischen 35 und 40 min, auf keinen Fall länger.
- Note: 1,0

Fragen:

(Die Fragen sind aus naheliegenden Gründen natürlich nicht wörtlich zitiert. Die Ausschweifungen, die bei einigen Fragen vorkommen, habe ich mir jedoch nicht selbst ausgedacht, die hat Jarke wirklich gebracht.)

Jarke: Sie haben also als Prüfungsgebiete die Bereiche Datenbanken, Betriebssysteme und Compilerbau?

Ich: Ja.

Jarke: Mit welchem Themengebiet möchten sie denn gerne beginnen?

Ich: Datenbanken.

Jarke: Betrachten wir als erstes ein Beispiel aus dem Supermarkt. Nehmen wir an, wir hätten eine Menge von Kunden, die im Supermarkt Waren einkaufen, wobei Kunden, die mehrfach einkaufen, verschiedene Nummern pro Einkauf erhalten. Kunden werden identifiziert durch Nummern, Waren ebenso. Zu Waren soll zusätzlich noch eine Gruppe (Gemüse, Obst ...) und der Preis mit abgespeichert werden. Kunden kaufen Mengen von Waren ein. Das Schema einer diese Informationen verwaltenden Datenbank soll mit einem Entity-Relationship-Modell dargestellt werden.

Ich: (Zeichne das ER-Diagramm auf)

Jarke: Wandeln sie dieses in ein relationales Schema um!



- Ich: (Mache dies)
- Jarke: Vergessen wir von nun an die Relation Kunde, die nur durch die Kundennummer repräsentiert wird. Stellen sie die verbliebenen Relationen im hierarchischen Modell dar.
- Ich: (Zuerst etwas verdutzt, zeichne den Baum mit zwei Knoten hin)
- Jarke: OK, zurück zum relationalen Modell. Wir wollen jetzt, um evtl. Ladendiebe aufspüren zu können, feststellen, welche Waren ein Kunde gekauft hat. Stellen Sie diese Anfrage in SQL oder so.
- Ich: (Anfrage hingeschrieben)
- Jarke: Noch eine Anfrage. Um am Ende eines Tages feststellen zu können, welche Waren noch vorhanden sind und welche nachbestellt werden müssen, wäre es sinnvoll, einen Überblick über alle Waren und der verkauften Menge (insgesamt, nicht pro Kunde) zu erhalten. Formulieren Sie auch diese Anfrage.
- Ich: (Etwas überlegen, da diesmal mit der Group-by-Klausel und der eingebauten Funktion Sum agiert werden muß. Jarke legt anscheinend keinen so großen Wert auf Syntax.)
- Jarke: Nun noch ein wenig Normalisierungstheorie. Jeder Kunde erhält eine Rechnung mit Kundennummer, Name (glaub ich zumindest), Gesamtbetrag, sowie für jede gekaufte Ware einen Eintrag für Warennummer, Menge, Einzelpreis, Gesamtpreis. Stellen Sie das als Relation auf und wandeln sie es in 3. Normalform um.
- Ich: (Relation ist schon 1. NF, habe sie dann in 2. NF umgewandelt, dann haben wir noch diskutiert, ob sie denn auch schon 3. NF ist, da ich meinte, daß in der Relation Ware mit den Attributen Kundennummer, Warennummer, Menge und Gesamtpreis noch eine transitive Abhängigkeit verborgen sein könnte. War aber im Endeffekt egal. Jarke wollte die Relation nicht noch weiter zerstückeln.)
- Jarke: Das sollte zu Datenbanken reichen. Beschäftigen wir uns mit Betriebssystemen. Was sind die Vorteile von Multiprogramming im Gegensatz zu Singleprogramming?
- Ich: Auslastung der Maschine (Betriebsmittel), höherer Durchsatz ...
- Jarke: Was ist der Unterschied zwischen Multiprogramming und Time-Sharing?
- Ich: (Habe nicht ganz genau verstanden, worauf er mit Unterschied hinaus will und habe ihm die beiden Begriffe erklärt, ohne explizit den Unterschied herauszuarbeiten. Hat ihm ausgereicht.)
- Jarke: Nennen Sie doch mal einige Zuteilungsverfahren für Betriebsmittel und erklären Sie kurz die Vor- und Nachteile.
- Ich: FIFO-Warteschlange, mehrere Warteschlangen mit unterschiedlichen Prioritäten, Zeitscheibe, sich dynamisch ändernde Prioritäten.
- Jarke: In welchen Zuständen kann eigentlich ein Prozeß sein? Und zeichnen Sie bitte ein Zustands-Übergangsdiagramm auf.
- Ich: Running, waiting, blocked.
- Jarke: Zu was für Anlässen treten die Übergänge auf und wie wird das realisiert (Datenstrukturen usw.)

III. Betriebssysteme

Frage: Was können Sie zur Hauptspeicherverwaltung sagen ?

Ich erkläre ihm, daß es einmal möglich ist, den Speicher in nur zwei Teile einzuteilen, den Bereich für das Betriebssystem und den für einen User (single partition allocation), daß es aber bei mehreren Benutzern nötig ist, den Speicher auf mehrere Prozesse aufzuteilen (multiple partition allocation). Dazu zähle ich als Möglichkeiten Paging, Segmentation und Paged Segmentation auf.

Frage: Wie funktioniert das denn beim Paging ?

Ich erzähle ihm das Grundprinzip mit der Page Table usw. und daß die noch nicht belegten Seiten (Frames) vom Betriebssystem zu verwalten sind. Falls ein Prozeß in den Hauptspeicher will, dafür aber nicht mehr genug Platz ist, kann er entweder nicht aufgenommen werden oder ein anderer Prozeß muß herausgeswap werden. Irgendwie kommen wir dann - von mir forciert - auf virtuellen Speicher zu sprechen.

Frage: Was passiert bei einem Seitenfehler ?

Nachdem mit Hilfe der Page Table festgestellt wurde, daß die angesprochene Seite nicht im Hauptspeicher ist, wird sie aus dem Swap-Bereich der Platte in den Hauptspeicher geholt. Ist dort ein Frame frei, so wird dieser benutzt, ansonsten muß eine anderer Frame freigemacht werden.

Frage: Was kann denn beim dauernden Paging passieren ?

Hier ist das *Thrashing*-Phänomen angesagt. Ich erkläre ihm, was es ist.

Frage: Wie kann man dem begegnen ?

Ich bringe das Working-Set-Modell ins Spiel, womit er zufrieden ist.

Frage: Sie sprachen eben von mehreren Prozessen. Wie wird das denn geregelt ?

Zunächst erkläre ich, was ein Prozeß ist und in welchem Zustand er sein kann (ready, running, waiting) bzw. worauf ein Prozeß alles warten kann (s. Abb. 4.8 im Buch). Dann erwähne ich die ready-Schlange, in der Prozesse auf die Zuteilung des Prozessors warten.

Frage: Welcher Prozeß wird denn nun gewählt ?

Ich erwähne Scheduler und einige Möglichkeiten des Schedulings

Frage: Welche Kriterien gibt es denn, diese zu beurteilen ?

Ich fange an aufzuzählen: Wartezeit in der ready-queue, komme aber nur bis hier, da er wohl nur das hören wollte.

Frage: Wie geschieht denn genau ein Prozeßwechsel ?

Literatur:

Betriebssysteme:

- Silberschatz/Galvin: „Operating System Concepts“
- Die ersten 9 Kapitel plus das über UNIX bildeten den vereinbarten Prüfungsumfang.
- Tanenbaum: „Betriebssysteme“

Das Buch ist im englischen Original vermutlich ziemlich gut. Die deutsche Übersetzung ist allerdings fürchterlich!

Datenbanken:

- Elmasri/Navathe: „Fundamentals of Database Systems“
- Kann ich nur jedem weiterempfehlen: ausführlich und sehr gut verständlich. Besonders gut geeignet: die Kapitel zur Normalisierungstheorie und der Vergleich von hierarchischem, Netzwerk- und relationalem Modell sowie zur Transformation von ER in die verschiedenen Modelle.
- Korth/Silberschatz: „Database System Concepts“
- ebenfalls empfehlenswert, allerdings weniger tiefgehend als der Elmasri/Navathe
- Folienkopien der Jarke-Vorlesung
- ziemlichher Schrott, aber was hilft's ...

Datenbanken/Expertensysteme:

- J. D. Ullman: „Database and Knowledge Base Systems“, Band I
- Da steht insbes. alles über DATALOG drin, was man so braucht.

Expertensysteme:

- G. Gottlob et al.: „Expertensysteme“

bildete die Grundlage für den Prüfungsteil über Expertensysteme. Reicht hierzu auch völlig aus ... bis eben auf die PROLOG- und DATALOG-Grundkenntnisse, die man zusätzlich benötigt.

Viel Erfolg!

André

Zugriffsstrukturen Hier konnte ich mich auf die bloße Aufzählung der einzelnen Verfahren und die Erläuterung des Unterschiedes zwischen Primär- und Sekundärzugriffungsverfahren beschränken. Er wollte dann noch etwas über sequentielle Verarbeitung wissen, ich erzählte von physisch sequentieller und verketteter sequentieller Speicherung, außerdem sollte ich noch sagen, wie man die einzelnen Blöcke auf Platte ablegt, um die Zugriffszeiten bei sequentieller Verarbeitung zu minimieren.

Sortieren großer Dateien Einmal direktes Mischen (siehe Vorlesung) und noch eine andere Variante aus dem Kemper/Eickler, die ich noch zufällig im Gedächtnis hatte.

Einfluß der Pufferverwaltung auf die Recovery Hier packte ich dann aus über *steal*- und *force*-Eigenschaften, wann Redo und wann Undo nötig ist.

Transaktionen Charakterisierung von Transaktionen durch die ACID-Eigenschaften, genau erklärt, was die einzelnen Buchstaben bedeuten. Er wollte auch noch wissen, welche Buchstaben denn die Recovery betreffen (A + D).

Serialisierbarkeit Hier sollte ich einfach mal erzählen, ich fing an mit FSR, VSR, meinte, daß VSR eigentlich ein ganz gutes Kriterium sei, aber daß der Testalgorithmus dummerweise exponentiell ist, und daß man deshalb CSR verwendet. Jarke meinte, es gebe noch einen weiteren Grund, nach einer kurzen Denkpause fiel mir schließlich noch PCA ein. VSR ist nicht praxis-commit-abgeschlossen, CSR aber schon (siehe Skript oder Vossen).

Scheduling-Protokolle Hier konnte ich mich durch selbständigen Vortrag vor einer detaillierten Wiedergabe schützen, ich nannte alles, was es so gibt: 2PL, S2PL, MGL, TL, Zeitmarkenverfahren, Serialisationsgraphen-Tester, optimistische Verfahren. Außerdem motivierte ich S2PL (Striktheit des Erzeugnisses) und MGL (Transaktion bestimmt Sperrgranularität selbst).

Vergleich: objektorientiertes Datenmodell \rightarrow relationales Datenmodell

Schließlich kam der große Vergleich zwischen den beiden Datenmodellen. Der Vergleich sollte auf Herrn Jarke Wunsch entlang der üblichen Dimensionen stattfinden:

- Strukturen,
- Operationen,
- Constraints.

Vossen: (der Vossen), 2. Aufl. Der Vossen ist an vielen Stellen unnötig formal. Sehr brauchbar sind allerdings die Abschnitte über DBMS-Architektur und Transaktionsverwaltung. Das Kapitel über Antrageoptimierung bietet eine gute Übersicht und hilft, etwas Orientierung in den entsprechenden Folienkopien zu finden.

Elmasri/Navathe: Fundamentals of Database-Systems, 2. Aufl. Eine sehr detaillierte und praktisch ausgerichtete Quelle. Hervorragend geeignet für die Themen Indexstrukturen und Normalisierung.

Kemper/Moerkotte: Object-Oriented Database Management Sehr leicht zu lesen. Wer bereits eine objektorientierte Programmiersprache wie Java oder C++ beherrscht, sollte gar keine Probleme haben. Besonders einprägen sollte man sich das Kapitel über die Schwächen des relationalen Modells sowie den Abschnitt über Transaktionskonzepte für Designanwendungen. Das ist ungefähr schon die halbe Miete, was diesen Prüfungsteil anbelangt.

Viel Erfolg!

• Also formulieren sie mal eine Relation in DATALOG, die alle Beziehungen zwischen einem Wissenschaftler und den von ihm beeinflussten Wissenschaftlern enthält. Ein Wissenschaftler X beeinflusst einen anderen Y, falls Y den X zitiert – und das pflanzt sich dann fort.
 Ah ja, sie möchten eine transitive Hülle haben. Also:

2 Nicht-Standard Datenbanken

```

Select p.Name, Count(v.VNr)
From publiziert p, Veröffentlichung v, Paper pap
Where p.VNr = v.VNr and pap.VNr = v.VNr
Group by p.Name
Order by Count(v.VNr) desc
    
```

• Ah, brauche ich gar nicht (peinlich). Also:

• Warum brauchen sie Wissenschaftler ?

Tja, ich wollte erstmal Wissenschaftler, publiziert, veröffentlicht und Paper joinen.

• Ja.

In SQL ? (Dumme Frage, wie will man das wohl in RA/TRK/DRK machen ??)

• Gut. Jetzt hätte ich gerne ein Ranking der Wissenschaftler nach der Anzahl ihrer Veröffentlichungen in Zeitschriften. Also eine Tabelle der Form *Ranking(Name, #Veröffentlichungen)* mit absteigender Anzahl.

```

Wissenschaftler(Name, Adresse)
publiziert(Name, VNr)
Veröffentlichung(VNr, Name)
zitiert(VNr_zitiert, VNr_wirdzitiert)
Buch(ISBN, VNr)
Zeitschrift(Titel, Vol, VNr)
Paper(Anz_Seiten, VNr, Zeitschr_VNr)
    
```

Schema heraus:

Ah ja, man führt einen künstlichen Schlüssel (VNr) für Veröffentlichung ein. (Argerlich das ich da nicht gleich von selbst draufgekommen bin). Also es kommt folgendes DB-

• Was macht man da allgemein ?

Veröffentlichung ja nicht eindeutig.

Danke. Hm, jetzt kriegen wir aber Probleme mit den Schlüsseln. Der Name ist für

• Die kann man einfach als 1:1 Beziehung darstellen.

Ich fing also an Relationenschemata zu erstellen. Was ist jetzt mit der Isa-Beziehung ?

ma.

• OK, dann übertragen Sie mal das ganze in ein relationales Datenbanksche-





I'm a loser baby... so why don't ya...

- **Schade. Hätte ich von Ihnen eigentlich erwartet.**
Haha. OK, blackout. Ich male irgendwelche magic Prädikate hin und muss dann gestehen, daß ich das nicht draufhabe.
 - **Jaja, aber wandeln sie doch mal die beiden Regeln um, wenn ich z.B. wissen will, wen Jarke alles so beeinflusst ?**
Also ich erzähle etwas von Top-Down Analyse der Regelmenge, Rewriting für jedes mögliche Adornment, effiziente Bottom-Up Auswertung, da nur relevante Relationen betrachtet werden. Das ganze natürlich stockend.
Und das auch nur halbherzig, weil ich dachte Jarke würde danach niemals fragen. Grober Fehler.
Schluck!!! Magic-Set hatte ich mir irgendwann mal vor ein paar Wochen angeschaut. Und das auch nur halbherzig, weil ich dachte Jarke würde danach niemals fragen. Grober Fehler.
 - **OK, da gibt es ja noch eine andere Methode, nämlich Magic-Set. Wie sieht die Auswertung hier aus ?**
Obige DATALOG-Klauselmenge ist linear, also können wir die (effizientere) Variante der semi-naiven Auswertung nehmen. Der Vorteil ist, daß bei jedem Schritt nur das Delta der letzten beiden Schritte berücksichtigt werden muss. Die intensionalen Relationen müssen bei der naiven Methode im Gegensatz hierzu immer neu berechnet werden.
 - **Was ist der Unterschied bei der Semi-naiven Methode ?**
zitiert \cup zitiert \bowtie beeinflusst
 - **Wie lautet der Ausdruck ?**
Also eval(beeinflusst,zitiert,beeinflusst) wird per RA Ausdruck ausgewertet. Und zwar so lange bis ein Fixpunkt für beeinflusst erreicht wird.
 - **Schreiben sie mal hin!**
Die intensionale Relation beeinflusst wird zu Beginn mit \emptyset materialisiert.
 - **Reicht. Erzählen sie doch mal wie man das in einer RelDB auswerten würde.**
Da gibt es einmal die naive und die bessere semi-naive Auswertungsmethode. Ich habe die Methoden also verbal beschrieben.
- Jetzt haben wir allerdings erst mal nur die VNummern und noch nicht die Namen der zugehörigen Wissenschaftler. Müsste man noch zwei Joins realisieren. Oder reicht das so ?
- beeinflusst(X,Y) :- zitiert(X,Y).
beeinflusst(X,Y) :- zitiert(X,Z), beeinflusst(Z,Y).

Prüfungsprotokoll Praktische Informatik

Prüfer : Prof. Dr. M. Jarke
Fächer : Einf. in DB
Objekt orientierte DB (Kemper/Morkotte)
Betriebssysteme (Silberschatz)
Datum : 20.08.96

- Einf. in DB :
- Im Internet gibt es ein Buchladen, wo man Bücher als Kunde bestellen kann. Ein Kunde kann auch gleichzeitig als Autor sein. Situation in ER-Modell darstellen.
 - ER-Modell in relationalem Modell Überführen.
 - Sind die Relation in Normalformen ?
 - Anfrage in SQL formulieren, um alle Kunde herauszufinden, die gleichzeitig Autor sind.
 - ER-Modell in Netzwerkmodell überführen.
 - ER-Modell in Netzwerkmodell überführen.
 - Wie sehen die Datensätze in Netzwerkmodell aus ?

OODB:

- Wie sehen die Modellierung von obige ER-Modell in OO-Modell aus ?
- Welche Vorteile hat man in OODB im Vergleich zu relationalen DB ?
- Was muß man bei Vererbung von Operationen beachten ?

Betriebssysteme:

- Beim Drücken von einem Dokument werden die Daten in Seiten zum Druckerpuffer geschickt, welche Situation tritt da auf ?
- Wie sehen die Lösung von Consumer-Producer Problem mit Semaphore aus ?
- Welche Probleme können da auftreten ?
- Was sind die Deadlock-Bedingungen ?
- Bei welchen konkreten Betriebsmitteln kann kein Deadlock auftreten ? warum ?
- Bei welchen konkreten Betriebsmitteln kann Deadlock auftreten ?
- Was ist Paging-System ?
- Welche Seitensetzungsalgorithmus werden dann verwendet ?



Prüfer: Prof. Dr. Matthias Jarke
Datum: 11.9.96
Note: 2,3
Fächer:

Einführung in Datenbanken (sehr zu empfehlen: „Fundamentals of database systems“ von Elmasri/Navathe, sehr gut zu lesen, reicht eigentlich zur Vorbereitung völlig aus; Vossen ist meiner Meinung nach viel zu theoretisch);

Betriebssysteme (Sibberschatz Kapitel 1-12 + UNIX);

Expertensysteme (Gottlob, Frühwirth, Horn „Expertensysteme“ - nicht sehr zu empfehlen, da viele Sachen nur angenssen werden, die in der Prüfung ausführlicher verlangt werden, etwa Prolog; ich weiß allerdings nicht, ob das neue Buch von Norvig/Russell unbedingt besser ist);

Ich hatte eigentlich zuerst den Eindruck, daß Jarke an diesem Tag ganz gut gelaunt war, was sich nachher allerdings als falsch herausstellte...alle Leute bekamen schlechtere Noten, als sie es erwartet hatten - von seiner angeblich guten Notengebung war nichts zu merken. Vielleicht lag es auch daran, daß es für uns alle ein Freiversuch war und er das Prüfungsniveau deshalb etwas angezogen hat.

Bis auf Betriebssysteme lief die Prüfung bei mir eigentlich ziemlich untypisch, da er fast alles neue Fragen stellte, die in anderen Protokollen noch nicht vorkamen. Außerdem ließ er mich viele Fragen gar nicht zuende beantworten; ich bekam von ihm auch nicht viel Feedback, deswegen kann ich kaum sagen, ob meine Antworten überhaupt richtig waren.

Betriebssysteme

Erzählen Sie mir mal etwas über die Geschichte der Hauptspeicherverwaltung, wie das so anfang und welche Strategien heute meistens verandt werden.
Anfangen bei Single-Partition-Allocation, Multiple-Partition-Allocation - ermöglicht Multiprogramming, bis zum heute angewandten Demand Paging.

Sie haben gesagt, durch Multiple-Partition-Allocation wird Multiprogramming ermöglicht. Welche weiteren Hardwarevoraussetzungen gibt es denn für Multiprogramming? Es muß möglich sein, die CPU zu unterbrechen (Interrupts und Preemption).

Jetzt gibt es ja neben Demand Paging noch andere Strategien, Seiten in den Hauptspeicher zu lesen...

Pure Demand Paging und Prepaging.

Gut, Kommen wir zu den verschiedenen Strategien, Seiten auszulagern.
FIFO, Optimaler Algorithmus (nur zu Vergleichszwecken), Approximation: LRU (hier wollte er noch genau wissen, wie das implementiert ist: Stack oder Counter), Approximationen von LRU: second chance und enhanced second chance.



Nein, aber ich meine formal, so richtig mit Formeln.
Ich habe es dann mit einer Formel versucht, die ein bisschen nach Relationenentpupelkalkül aussah, war aber auch nicht gemeint. Diese gesuchte Formel sollte angeblich im Skript stehen, konnte ich aber bis jetzt nicht finden.

Na gut, nehmen wir doch für eine SQL-Anfrage mal unser Standardbeispiel Angestellter(Eno, Name, Adresse, Dept). Hier besteht ja eine funktionale Abhängigkeit zwischen Eno und Dept. Was bedeutet das konkret?
Daß kein Angestellter in zwei verschiedenen Departments arbeiten kann.

Gut. Dann stellen sie doch mal eine SQL-Anfrage nach dieser funktionalen Abhängigkeit, d. h. fragen sie nach den Angestellten, die in zwei verschiedenen Abteilungen arbeiten.
Hier habe ich dann etwas länger gebraucht, Jarke hat mir auch irgendwie nicht mehr viel Zeit zum Überlegen gelassen. Meine Lösung sah nachher ungefähr so aus:

```
SELECT Eno FROM Angestellter x, Angestellter y  
WHERE x.Eno = y.Eno  
AND x.Dept != y.Dept
```

Hieraus eine Anfrage in Relationenalgebra zu machen, wäre wahrscheinlich etwas zu kompliziert. Aber sagen Sie doch mal, welchen Konstrukten der Algebra die einzelnen Anfragezeile entsprechen!
SELECT entspricht der Projektion, FROM dem kartesischen Produkt, obwohl daraus nachher sowieso meistens ein Join wird, und WHERE der Selektion.

Ija, dann brauchen wir noch eine schöne Frage zum Schluß. Man will ja bei der Erstellung von Datenbanken möglichst alle Relationen in 3. NF haben. Geben Sie einen abhängigkeitserhaltenden Algorithmus zur Dekomposition in die 3. NF an.
Ich habe dann den Algorithmus angegeben, wie er in der Übung WS 95/96 beschrieben war, also erstmal minimale Überdeckung bestimmen, dann gleiche linke Seiten zusammenfassen.

Der Algorithmus, den Sie beschreiben, ist doch eher ein Syntheselgorithmus!
Habe dann gesagt, daß es genau so in der Übung beschrieben ist, und zwar geht man dabei von einer Universalrelation aus, die man durch diesen Algorithmus dekomponiert.

Viel Glück für Eure Prüfung!



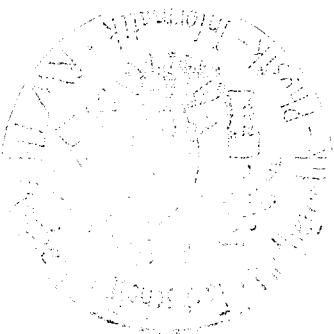
Praktische Informatik – Gedächtnis-Protokoll

Datum der Prüfung: 29.09.95

Prüfer: Prof. Dr. M. Jarke

Einführung in/Implementierung von Datenbanken (Jarke-Vorlesung)
Betriebsysteme (Silberschatz-Buch)

Note: 1,3



Betriebsysteme

Jarke: Man hat ja schon immer versucht, Rechner besonders effektiv zu nutzen; dazu hat man sich diverse Hilfsmittel einfallen lassen. Können Sie mal eine kurze historische Übersicht von zehn Sekunden geben, es darf auch eine halbe Minute werden!

ich: Ziel war, die CPU-Auslastung zu erhöhen, ich erzähle schnell was über Monitore, off-line Operation, Buffering, Spooling und Multiprogramm/Timesharing. Stets wird versucht, Wartezeiten (IO) sinnvoll zu nutzen.

Jarke: Ja, wenn ein Prozeß auf eine IO-Operation wartet, was passiert denn da, da gibt es ja diverse Warteschlangen, malen Sie das doch mal genau auf!

ich: Prozeß-Zustandsdiagramm gemalt, Warteschlangen fuer IO-Geräte und *ready queue* erklärt.

Jarke: OK, wenn nun ein Prozeß vom *running* in den *waiting* Zustand wechselt, was passiert denn dann?

ich: Dann wird er vom Betriebssystem in eine Queue für ein Gerät eingereiht, und der Scheduler wählt einen Prozeß aus der *ready queue* aus, der als nächster laufen darf.

Jarke: Aber das Betriebssystem ist doch selbst auch ein Prozeß, wie geht denn das?

ich: (Software-) Interrupt, Prozessor geht in Supervisor-Modus

Jarke: Und was passiert dabei alles?

ich: Also, die CPU rettet beim Interrupt erst einmal den PC und den Status, anschliessend kann das Betriebssystem CPU-Register und systemabhängige Dinge retten.

Jarke: Was für systemabhängige Dinge zum Beispiel?

ich: Bei Paging Systemen z.B. Page Table Base Pointer.

Jarke: Und dann wählt das Betriebssystem einen neuen Prozeß aus der *ready queue* aus, was passiert dabei?

ich: Das geht umgekehrt wie oben, also Register zurück, in User-Modus und dann in den jetzt zu aktivierenden Prozeß springen.

Jarke: Und wie geht das genau, das Springen; welche Instruktion wird dabei vom Betriebssystem zuletzt ausgeführt?

ich: Das Laden des Programm-Zeigers.

Jarke: OK, Sie erwähnten eben *pagng*, also Speicherverwaltung. Wie machen denn das moderne Systeme so, wenn man nicht unbedingt zur Microsoft-Welt gehören will?

ich: (hatte ihn falsch verstanden: "..., wenn man zur Microsoft Welt gehört?") Also, von Microsoft hab' ich ja gar keine Ahnung...

Jarke: (wiederholt seine Frage nochmal)

ich: Moderne Systeme verwenden heute meist *demand pagng*. Ich erzähle, was *pagng* ist, und daß mit "demand" die virtuelle Speicherverwaltung hinzugefügt. Dabei werden Frames ausgelagert, wenn für neue Pages kein Platz mehr ist.