

# Prüfungsprotokoll Praktische Informatik

Prüfer: Prof. Seidl

Datum: April 2005

Themen: Introduction to Databases, Indexstrukturen für Datenbanken, Betriebssysteme

Note: 1,3

## **Datenbanken (ca. 20 Minuten):**

Q: Ich würde gerne mit Datenbanken anfangen, ist das ok?

A: Ja.

Q: Um nicht immer in der gleichen Reihenfolge zu fragen: Was ist eine Transaktion?

A: mehrere DB-Operationen zu einer atomaren Einheit zusammengefasst.

Q: Was ist ACID?

A: Atomicity, Consistency, Isolation(hier hab ich aus versehen Independency gesagt, aber er meinte das wäre nicht schlimm, würde die Aussage eigentlich noch besser beschreiben), Durability und alle 4 kurz erklärt.

Q: Was muss man machen wenn eine Transaktion fehl schlägt?

A: die bisherigen Änderungen rückgängig machen

Q: Welche DB-Modelle gibt es so?

A: Relationales, Objekt-relationales, Objektorientiertes, Deduktives und noch ein paar ältere, die nicht mehr verwendet werden...

Q: Was haben die den alle gemeinsam?

A: (Wusste ich nicht so richtig) Relationen? (Obwohl OO-DB arbeiten doch mit Objekten?!? Ich weiß immer noch nicht was er eigentlich hören wollte...)

Q: Was ist denn eine Relation?

A: Das Kreuzprodukt von Attributen...

Q: Hier wurde nachgebohrt: Hhm, wie groß kann denn so eine Relation werden, wenn Sie sagen, dass ist ein Kreuzprodukt?

A:?!? Ich wusste nicht genau worauf er hinaus wollte, bis wir/ich irgendwann drauf gekommen bin, dass eine Relation natürlich nur eine (endliche) Teilmenge des Kreuzproduktes ist!

Q: Was ist denn ein Schlüssel?

A: Ein Attribut, dass ein Tupel einer Relation eindeutig identifiziert...

Q: (Hier war er natürlich nicht ganz zufrieden mit. Ich weiß nicht mehr genau wie das genau war, irgendwas hat er von künstlichen Schlüsseln erzählt, dass man die nicht immer hat...) Hhm, was ist denn wenn man beispielsweise eine Person mit Name, Adresse, Telefonnummer hat

A: Ja dann muss man fast alle Attribute als Schlüssel nehmen,...

Q: Ok. Wir haben jetzt eine Relation gegeben (VorNr, Titel, MatrNr, Name). Ist das gut?

A: Nein, nicht besonders. Hier können Anomalien auftreten.

Q: Welche Anomalien gibt es denn?

A: Einfüge-, Lösch-, Update-Anomalien

Q: Was wäre denn eine Lösch-Anomalien unserem Beispiel?

A: Wenn man alle Studenten löscht, würde die Vorlesung auch wegfallen, bzw. müsste man NULL-Werte beim Studenten einfügen.

Q: Wie kann man denn diese Anomalien vermeiden?

A: Durch Normalisierung.

Q: Wir wollen jetzt unsere Relation in 3.NF haben, da gibt es den Synthese-Algorithmus. Wie funktioniert der?

A: Kanonische Überdeckung,...

Q: Was ist den die Kanonische Überdeckung? Wie berechnet man sie?

A: Minimale äquivalente Menge von FDs. Links-, Rechtsreduktion, Regeln mit leerer rechten Seite eliminieren, gleiche linke Seiten zusammenfassen.

Q: Was ist denn z.B. die Rechtsreduktion?

A: Man überprüft in einer FD, ob man ein Attribut der rechten Seite weglassen kann und dieses durch andere FDs erzeugen kann...

Q: Wie geht man dann beim Synthese-Algorithmus weiter vor?

A: Man teilt Relationen gemäß der FDs auf und ordnet die FDs zu. (Am Beispiel vorgemacht). Und wenn wir dann in keiner der Relationenschemata einen Kandidatenschlüssel haben, müssen wir noch eine weitere Relation erzeugen nur mit den Attributen eines Kandidatenschlüssels und ohne FDs.

Q: Wir wollen jetzt eine Anfrage machen, alle Matrnr von Studenten, die 'DB' hören. Wie geht das in SQL.

A: SELECT MatrNr from Vorlesungen, hören WHERE Titel='DB'

Q: Und was ist die Entsprechung in Relationaler Algebra?

A: Also SELECT ist die Projektion, Where die Selektion,...

Q: Schreiben sie doch für dieses Beispiel auch hin.

A: \*Hingeschrieben\*

Q:(Hier weiß ich nicht mehr genau die Fragen, wir kamen irgendwie aufs Domänenkalkül) Dann schreiben sie mal die Anfrage mal im Domänenkalkül...

A:(Das ging etwas stockend, da ich etwas mit dem Tupelkalkül durcheinander kam, deswegen wollte er mir helfen:)

Q: Worauf arbeiten denn Tupel- und Domänenkalkül? (Das hat mir nicht geholfen, im Gegenteil)

A: Tupel und Domänen?!? Obwohl eigentlich arbeitet das Domänenkalkül eher mit Tupeln... (hier war ich etwas verwirrt, Prof. Seidl fing an etwas von Mathematischer Logik zu erzählen, ich wusste wirklich nicht was er hören wollte, dieser Teil war einer von Zweien, die die Note abgeschwächt haben. Im Endeffekt wollte er hören dass es TupelVARIABLEN und DomänenVariablen sind und nicht Tupel und Domänen... Die Anfrage im Domänenkalkül hab ich ihn dann aber doch noch hinschreiben können und er freute sich, weil ich sie direkt optimiert habe, also für 'DB' nicht noch extra eine Variable definiert habe)

Q: So, jetzt haben wir schon etwas überzogen, kommen wir zu Indexstrukturen...

## **Indexstrukturen (ca. 15 Min):**

Q: Brauch man denn Indexstrukturen überhaupt?

A: Nicht unbedingt. Man könnte auch die gesamte Datenbank mit einem linearen Scan durchsuchen. Bei manchen Anwendungen... oder allgemein... eigentlich möchte man immer schnellere Antworten bekommen, deswegen benutzt man Indexstrukturen.

Q: Wenn wir jetzt Beispielsweise Matrikelnummern oder Namen speichern wollen, wie können wir das machen?

A: Mit Hashing beispielsweise oder Bitmaps...

Q: Was sind denn Bitmaps? Machen Bitmaps hier Sinn?

A: Nicht so, man verwendet diese normalerweise für Attribute mit kleinen Wertebereichen.

Für jedes Attribut gibt es einen Bitvektor und ein Tupel besteht dann aus einer Menge von Bits.

Q: (Hier war er auch nicht ganz zufrieden, wir haben dann noch drüber diskutiert, wie das genau aussieht. Das war der zweite Teil, der die Note leicht abgeschwächt hat...)

Q: Es gibt da einen Baum, der in den letzten 30 Jahren für ziemlich viel Aufsehen gesorgt hat...

A: Meinen Sie den B-Baum?

Q: Ja.

A: Also der B-Baum... (ausführlich erklärt: dynamisch, höhenbalanciert,  $k-2k$  Knoten,  $n+1$  Söhne, etc.)

Q: Was gibt es denn wenn wir ausgedehnte Objekte speichern sollen?

A: R-Baum, Erklärt mit MURs, Teilbäumen, Überlappungen...

Q: Wie ist denn der Suchaufwand im Vergleich zu B-Bäumen?

A: Schlechter, wegen den Überlappungen

Q: Wie wäre dann die Laufzeit?

A: Hhm, ich hatte erst gedacht zusätzlich zum lesen aller Blätter  $O(n)$  würde noch etwas logarithmisches zum Baumdurchlauf dazu kommen also  $O(n * \log n)$ , aber dann sind wir zusammen drauf gekommen, dass innerhalb des Baumes auch etwa genauso viele Elemente besucht werden wie auf Blattebene, also insgesamt  $2*n$ , was aber wieder  $O(n)$  ist.

Q: (Dann kamen wir irgendwie zur Punkttransformation)

A: Erklärt wie Intervalle als Punkte gespeichert werden, mit Punkt- und Bereichsanfrage

Q: Wie veranschaulicht man denn Intervalle? (ich wusste nicht worauf er hinaus wollte, auf Nachfrage meinte er, dass er es nur ganz einfach haben wollte)

A: Ich habe dann eine Achse gemalt und darüber Intervalle...

Q: Ja genau, es sind also ausgedehnte Objekte im 1D...

A: Ja.

Q: Dann muss ich natürlich noch auf den Relationalen Intervallbaum zu sprechen kommen

A: Ja natürlich :-). Also, Primärstruktur wird nicht materialisiert, wir müssen uns nur die Wurzel merken, die ein Wert  $2^{(n-1)}$  ist und einen Bereich von  $(2^n)-1$  abdeckt. An den Knoten werden die Intervalle als 2 Listen der Anfangs- und Endpunkte gespeichert, also Speicherbedarf von  $O(n)$ ...

Q: Werden da wirklich Listen gespeichert?

A: Man könnte vielleicht auch besser Bäume nehmen zum Abspeichern

Q: Ja, so macht man das auch, man nimmt dann einfach die sowieso vorhandenen B(\*)-Bäume

## **Betriebssysteme:**

Q: Was ist ein Prozess?

A: Programm in Ausführung mit Befehlszähler, Registerinhalte, Variablenbelegungen, etc.

Q: Wie hängt das BS mit Prozessen zusammen?

A: Prozesse sind das Grundlegende Konzept von Betriebssystemen.

Q: Und was macht das BS mit Prozessen?

A: Es erzeugt, schedult und beendet sie. (Irgendwie kamen wir dann noch drauf, dass es alle Betriebsmittel zuweist.

Q: Was ist ein Deadlock?

A: Wenn in einer Gruppe von Prozessen...(wissen wir doch alle, oder?)

Q: Welche Bedingungen gibt es für Deadlocks?

A: Die vier aufgezählt...

Q: Die CPU ist ja auch ein Betriebsmittel, kann es da auch zu Deadlocks kommen?

A: Nein, die ist entziehbar.

Q: Und beim Speicher?

A: Jein. Speicherseiten kann man auslagern. Irgendwann könnte dann natürlich die Auslagerungsdatei voll sein, aber bei den großen Kapazitäten heutzutage....

Q: Wie funktioniert denn das mit dem Speicher?

A: Heutzutage verwendet man virtuellen Speicher mit Paging... (ausführlich erklärt: Seiten, Frames, virtuelle/physische Adresse, MMU, etc.)

Q: Wie lagert man die Seiten denn aus?

A: Auslagerungsstrategien (angefangen ausführlich zu erklären: optimaler Algorithmus, NRU, FIFO, Second Chance, Clock, LRU, NFU, Aging,...)

Q: Ok, das reicht, wir wollen ja nicht überziehen, dann dürfen wir sie mal eben bitten raus zu gehen...