

Relationale Datenbanken 1

Aufgabenblatt 11

Jan Germann - Matrikelnummer 4131658
Christian Ratz - Matrikelnummer 123456

22. Oktober 2012

11.2

a) Datenbank

Eine Sammlung von Daten.

b) DBMS

Ein DBMS ist eine Sammlung von Programmen zur Verwaltung von einer Datenbank. Zuständigkeiten sind unter anderem Physische Abbildung, Strukturierung, Manipulation der Daten,

c) Filesysteme

Filesysteme sind physikalische Interfaces, die es erlauben auf Daten zu über einen Pfad zuzugreifen. Das Filesystem erlaubt nur den Zugriff über den Pfad, kümmert sich allerdings nicht um die Konsistenz oder Backups dieser Daten. Filesysteme machen also nur die physische Verwaltung der Daten.

Ein DBMS hingegen stellt logische Interfaces, zum Zugriff auf die in einer Datenbank gespeicherten Daten. Es kümmert sich im Gegensatz zum Filesystem auch um die physikalische Konsistenz und logische Kohärenz der gespeicherten Daten. Desweiteren erlaubt ein solches System auch die Redundanzverwaltung sowie Backup und Recovery. Der auffälligste Unterschied von einem DBMS zu einem Filesystem ist allerdings der Zugriff auf Daten aufgrund von Datensemantik und nicht durch einen spezifizierten Pfad.

d)

Ein DBMS ist im zum Beispiel Bankengeschäft sinnvoll. Hier wurde auch schon in der Vergangenheit die Datenverwaltung in Filesystemen durch DBMS ersetzt. Diese technologische Transitionen war durch das stetige Wachstum der Banken bedingt. Mit mehreren tausend Kunden ist das finden einen Datensatzes im Filesystem schwierig, die Erstellungen von Statistiken über alle Kunden eher schwierig durchzuführen.

Gerade im Bankgeschäft ist die Inkonsistenz oder gar der Verlust von Daten nicht zu tolerieren. Durch das Transaktionskonzept wurde bei dieser riesigen Menge von Datensätze auch das erste mal möglich die Konsistenz der Daten auch bei hohen Zugriffsvolumina gewährleistet.

Mit der Einführung von DBMS wurde der Grundstein für die Einführung von Geldautomaten gelegt. Erst durch diese wurde erst sicherer simultaner Zugriff auf gleiche Datenbestände ermöglicht. Man stelle sich vor, dass ein Kunde in Frankfurt und ein Kunde der in Hamburg bei der selben Bank abheben wollen. Diese müssten bei einem sequenziellen Zugriff auf den gleichen Datenbestand aufeinander warten. Datenbankmanagementsysteme erlauben die Konsistenzhaltung auch bei dieser Starken Belastung.

11.3

a)

Redundanz bezeichnet das mehrfache Vorhandensein von gleichen Entitäten (in diesem Fall »Daten«).

b)

Bei unkontrollierter Redundanz kann es zur Inkonsistenz der redundanten Daten kommen. Sowie zur Verschwendung von Speicherplatz aufgrund von überflüssiger Redundanz.

c)

Kontrollierte Redundanz kann den Vorteil haben, dass die Daten zwar redundant (sicherheitshalber auch an verschiedenen Orten) gespeichert werden, aus Sicht der zugreifenden Applikation allerdings nur einmal vorhanden sind. Hierbei kümmert sich das DBMS um die Konsistenz der mehrfach vorhandenen Daten. Hierbei kann Redundanz auch eingesetzt werden, um

die Geschwindigkeit von Zugriffen zu erhöhen, weil die Daten zum Beispiel auf verschiedenen Festplatten gespeichert werden, oder durch materialisierte Views angesprochen werden. Applikationen greifen hierbei optimalerweise über ein einzelnes Interface auf diese Daten zu.

11.4

a)

Views erlauben unterschiedliche Perspektiven auf die Datenbank. Ein View kann daher eine Teilmenge der Daten darstellen oder auch virtuelle Daten enthalten welche dynamische auf realen erstellt werden. Es muss zwischen zwei Arten von Views unterschieden werden, den dynamischen welche bei jeder Abfrage erneut zusammengestellt werden und den materialisierten welche von DBMS auch ohne eine direkt Abfrage aktualisiert werden. Für Applikationen gibt es keinen Unterschied zwischen der Benutzung einer Tabelle und einem View.

b)

Eine Transaktion bezeichnet eine Serie von Datenbankoperationen die logisch als eine einzelne behandelt werden. So wird die Konsistenz der Daten, bei voneinander abhängigen Abfragen, gewährleistet. Ein entsprechendes DBMS kann bei der Benutzung von Transaktionen auch die Nebenläufigkeit verwalten und somit bei Erhaltung der Konsistenz die Geschwindigkeit der Zugriffe optimieren. Transaktionen müssen atomar und von einander isoliert sein.

c)

Eine deklarative Abfrage spezifiziert nur, welche Daten gewollt sind, allerdings nicht woher sie kommen sollen. Diese Art von Abfrage ist also abgespalten von der physikalischen Positionen, Organization und Speicherung der abgefragten Daten.

Beispiel `SELECT first_name FROM data WHERE first_name='Smith'`