Data Engineering

Markus Stumptner

Inhalt

1. Einführung in Datenbanksysteme
2. Konzeptuelle Modellierung
3. Das Relationenmodell
4. Logische (relationale) Modellierung
5. Mehrbenutzerkontrolle
6. Recovery
7. Physische Datenorganisation

Datenbanksystem

besteht aus *Datenbank + Datenbankmanagementsystem*

Datenbank (DB)

- Schema
- Ausprägung

Datenbankmanagementsystem (DBMS)

- Software zur Gewährleistung der Datenbank-funktionalität (Wiederanlauf, Mehrbenutzerbetrieb, Datenschutz, Hintergrundspeicherverwaltung, etc.)

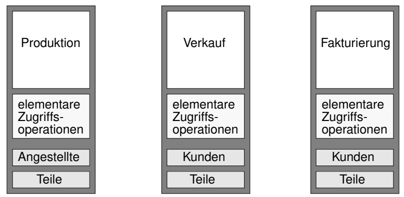
“Datenbank” wird oft auch als Kurzform für “Datenbank-system” (DBS) verwendet

Der Weg zum Datenbanksystem

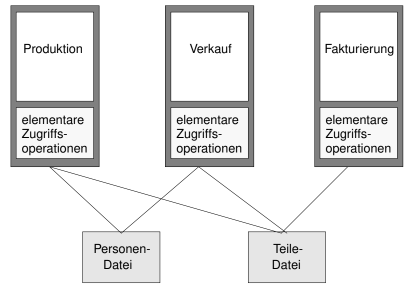
- Programme mit privaten Dateien (Einzellösungen)
- Integrierte Datenverarbeitung
- Datenbanksystem
- Objektorientiertes Datenbanksystem
- Verteiltes Datenbanksystem

Data Engineering -> Ansätze und Methoden zur Modellierung und Beschreibung von Daten (in Datenbanksystemen). Begriff entwickelt in Anlehnung an “Software Engineering”

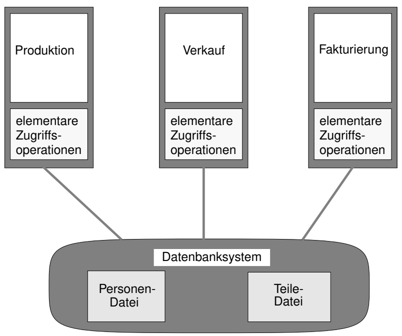
Einzellösungen



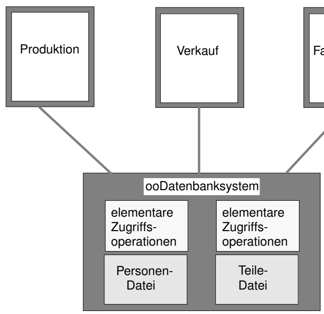
Integrierte Datenverarbeitung



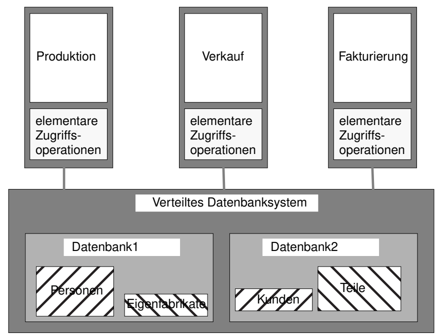
Datenbanksystem



objektorientiertes DBS



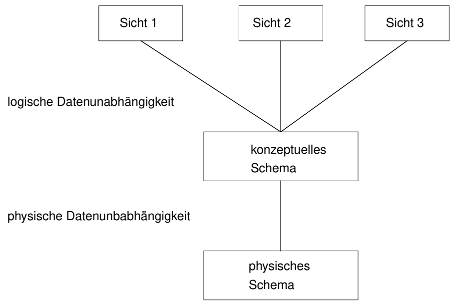
Verteilte Datenbank



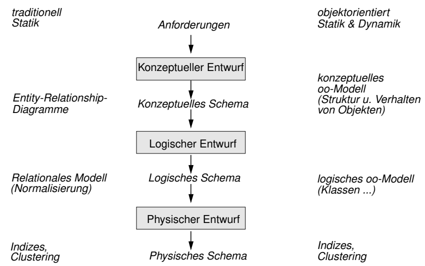
Datenbankkonzepte

1. persistente Datenhaltung
2. Hintergrundspeicherverwaltung
3. Wiederanlauf
4. Mehrbenutzerbetrieb
5. ad hoc-Abfragen
6. Datenschutz

ANSI/SPARC Architektur



Datenbankentwurfsprozeß



PERSISTENTE DATENHALTUNG

- Daten überleben die Programmausführung
- Jede Art von Daten kann persistent sein
- Die Persistenz von Daten ist *implizit*; es ist nicht notwendig Daten explizit vom Hintergrundspeicher zu lesen bzw. auf den Hintergrundspeicher zu schreiben

HINTERGRUNDSPEICHERVERWALTUNG

Große Datenmengen: Daten finden nicht zur Gänze im Hauptspeicher Platz, auch ein virtueller Speicher ist zu klein oder zu ineffizient

Spezielle Techniken sind notwendig

- Puffern von Daten
- Clustern von Daten
- Indizieren von Daten
- Abfragenoptimierung

Physische Datenunabhängigkeit

TRANSAKTIONEN

Transaktionen führen einen konsistenten Datenbankzustand in einen anderen konsistenten Datenbankzustand über

Eigenschaften:

- Atomarität
- Konsistenz (Serialisierbarkeit)
- Isolation
- Dauerhaftigkeit

MEHRBENUTZERBETRIEB

Daten werden von mehreren Benutzern gemeinsam verwendet.

Jeder Benutzer kann das System benutzen, als ob er der einzige wäre:

- Spezielle Techniken sind notwendig, um zu verhindern, daß sich mehrere Benutzer gegenseitig stören (*Concurrency Control)*.
- Traditionelles Korrektheitskriterium: *Serialisierbarkeit* von Transaktionen

WIEDERANLAUF

Ziel: Herstellung eines konsistenten Datenbankzustands nach einem Fehler

- vom Anwendungsprogramm erkannte logische Fehler (z.B.: "Konto nicht gedeckt")
- vom System erkannte Fehler (z.B.: zyklisches Warten mehrerer Programme auf die Freigabe einer Resource)
- Hautspeicherverlust (z.B.: Stromausfall)
- Hintergrundspeicherverlust (z.B.: disk crash)

Konzept: Transaktionen führen einen konsistenten Datenbankzustand in einen konsistenten Datenbankzustand über

DATENSCHUTZ

- Nicht alle Benutzer sind gleich
- Klassifikation in Benutzergruppen
- Zugriffsrechte (ändern, lesen, ausführen)
- Granularität der Objekte, für die Zugriffsrechte definiert werden können: Datenbank, Relation, Attribut
- Weitergabe und Rücknahme von Rechten

AD-HOC ABFRAGEN

Nicht alle Abfragen sind a-priori bekannt.

Es ist nicht notwendig für allgemeine Abfragen spezielle Anwendungsprogramme zu schreiben.

Direkte Abfragen an die Datenbank:

- einfach formulierbar
- deklarativ gestellt: es wird die Bedingung angegeben, die die Antwortmenge erfüllen soll, nicht aber das Verfahren, wie sie ermittelt wird
- werden vom System optimiert