Datenmodellierung physikalischer Design

Mehrbenutzerbetrieb / gemeinsamer Datenzugriff

- Wenn mehrere Benutzer gleichzeitig auf dieselben Datenbeständen zugegreiffen, müssen Massnahmen gegen Inkonsistenz und Verklemmung vorgenommen werden
- Transaktion
  - Folge von Operationen, die atomar, konsistent, isoliert und dauerhaft sein muss. Überführen eines gesicherten, konsistenten Datensatzes in einen anderen, gesicherten konsistenten Datensatz (mutieren)
  - ACID-Prinzip einer Transaktion (Atomary, Consistency, Isolation, Durability)
- Transaktionskonzept (Verfahren zur Sicherung der Transaktionen)
  - Pessimistisches Verfahren
    - Transaktionen absichern, durch Sperren der zu lesenden oder zu verändernden Daten vor anderen Zugriffen
  - Optimistisches Verfahren
    - Geht davon aus, dass Konflikte konkurrierender Transaktionen selten vorkommen.
    - Verzicht auf Sperren um die Parallelität und Performance zu steigern
    - Transaktionen durchläuft die drei Phasen: Lesen, Validieren, Schreiben
  - Deadlocks (Verklemmungen)
    - Entstehen wenn Transkationen wechselseitig aufeinander warten oder wenn zyklische Abhängigkeiten vorliegen

Physikalischer Datenbankdesign

- Das konzeptionelle Datenmodell (z.B. ERM) muss unter Berücksichtigung der Eigenschaften der DBMS und der Hardware in eine physische Datenorganisation überführt werden
- Wichtige Faktoren
  - Relationen
    - Entitäten welche auf konzeptioneller Ebene getrennt wurden, müssen u.U. auf der physikalischen. Ebene wieder zusammengeführt werden (Denormalisierung)
  - Schlüssel
    - Die im log. Datenbankschema enthaltenen Schlüssel können noch verändert werden.
  - Indexe
    - Leistungsoptimierung mittels Zugriffspfadmatrix
  - DDL/DML
    - Mächtigkeit und Konsistenzbedingungen

Verteilte Datenbanken

- Eine Datenbank ist dezentral wenn sie durch ein einziges konzeptionelles Datenbankschema beschrieben, aber durch mehrere physikalische Tabellenfragmente auf örtlich verteilten Rechnern gehalten wird
- Anforderungen
  - Lokale Autonomie: jeder Knoten verwaltet lokale Daten selber
  - Dauerbetrieb, kein Unterbruch bei Zu- und Abschalten von Standorten
  - Lokale Transparenz (Benutzer muss Datenstandort nicht kennen)
  - Transparenz von HW, Betriebssystem, Netzwerk- und DBMS
  - Aktualität und Verteilung der Daten muss sichergestellt werden, damit überall die aktuellen Datenversionen vorhanden sind
- Vorteile
  - Erhöhte Verfügbarkeit des Gesamtsystems
  - Hohe Flexibilität
  - Übereinstimmung mit dezentral organisierten Unternehmungen
- Nachteile
  - Hoher Administrationsaufwand
  - Oft geringes Know-how in dezentralen Standorten
  - Wenige ausgereifte DBMS
  - Schwierigkeit der Verteilung bestehender Datenbestände
- Werden aufgrund der heutigen Technologie kaum eingesetzt
- Formen der Datenverteilung
  - Horizontale Fragmentierung
    - Verteilen von bestimmten Tupeln (Selektion)
  - Vertikale Fragmentierung
    - Verteilen von bestimmten Attributen (Projektion)
  - Replikate
    - Kopien der Datenbestände auf allen Systemen

CASE-Tools

- Unterstützen die Datenmodellierung indem sie Beziehungen selber knüpfen aufgrund von vergebenen Attributen
- Testen Beziehungen aufgrund internerRegeln
- Drucken von Data Dictionaries