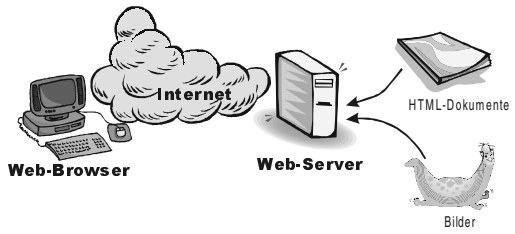
Grundbegriffe

- World Wide Web
  - Architektur, Protokolle
  - Inhalte, Sprachen
- Datenbanken
  - Aufgaben
  - Architekturen
- Multimedia
  - Medien
  - Multimedia-DBMS

World Wide Web – Architektur



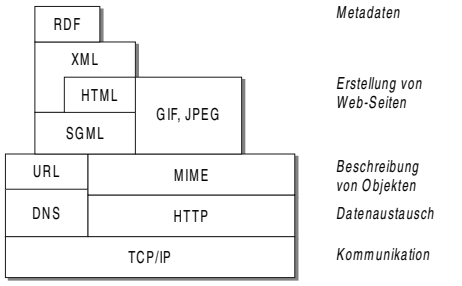
Dokumente und Links

- **Hypertext-Dokumente** (HTML) mit Verweisen (Links) auf andere Objekte (Dokumente, Multimedia-Objekte)
- Von Web-Servern bereitgestellt
- **Links**: durch URL spezifiziert
- **Web-Datei:** einzelne Datei (mit einem HTTP-Request)
- **Web-Seite:** Menge von Web-Dateien (eine Nutzeraktion)
- **Web-Site:** Menge von Web-Seiten (ein Server)

Uniform Resource Locator (URL)

- Adresse einer Ressource im Web
- Format:
  `<schema>:<schema spezifikation>`
- `<schema>` Protokoll (http, news, mailto, telnet)
- Für HTTP
  - Vollständig: http://<host>.<domain>:<port>/<path>/<filename>
  - Relativ: <path>/<filename>
  - Verweis in Dokument: http://<host>.<domain>:<port>/<path>/<filename>#<anchor>
  - Funktionsaufruf: http://<host>.<domain>:<port>/<path>/<procname>?<params>

Web-Protokolle und -Sprachen



Hypertext Transfer Protocol (HTTP)

- Einfaches Protokoll zum Datenaustausch auf Basis von TCP/IP
- **Zustandsloses Protokoll:** jede Anforderung ist unabhängig von allen anderen
- Methoden/Kommandos:
  - GET: Holen einer Web-Seite
  - HEAD: Holen der Header-Informationen
  - POST: Senden von Daten (Bytestrom) zum Server

HTTP - Beispiel

telnet www.fh-brandenburg.de 80
>GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 24 Sep 1999 06:46:29 GMT
Server: Apache/1.3.9 (Unix)
Content-Type: text/html
...

<HTML>
...

MIME

- Multipurpose Internet Mail Extensions
- Spezifikation von Inhaltstypen von Emails, Web-Dokumenten
- Form `<typ>/<subtyp>`
- Wird als Antwort auf ein HTTP-GET vor dem Dokument geliefert (Content-Type)
- Typen: text, image, audio, video, application
- Bsp.:
  - `text/html`, `image/gif`, `application/postscript`

Einbettung von Programmen

- Bereitstellung von Diensten, Ausführen von Programmen
  - Aktive Inhalte
    - Audio-/Video-Wiedergabe
    - Navigation in 3D-Modellen
  - Suche in Datenbanken
    - Suchmaschinen
    - Datenbankanfragen
  - Datenübergabe
    - Benutzereingaben

Client-seitige Einbettung

- Erweiterung des Browsers
  - Plugins für bestimmte Web-Inhalte
    - PDF, Video, Vektorgraphik, Animationen
  - Ausführung von Programmen und Skripten
    - Javascript
    - Java-Applets
- Bereitstellung der Programme → Web als Verteilungsplattform
- Sicherheitsproblematik

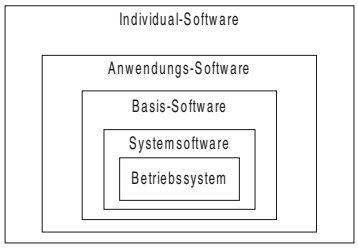
Server-seitige Einbettung

- Erweiterung des Web-Servers
  - Anbindung externer Programme
  - Zugriff auf externe Dienste (DBMS)
- Techniken
  - Server-seitige Skripte
  - CGI
  - ...

Datenbanken

- Motivation
- Komponenten und Funktionen
- Systemarchitekturen
- Anwendungsarchitekturen
- Quelle: Heuer/Saake: „Datenbanken – Konzepte und Sprachen“, Internat. Thomson Publ. 1995

Softwareschichten



Ohne Datenbanken: Datenredundanz

- Softwaresystem verwaltet selbständig eigene Daten in eigenen Formaten
  - Textverarbeitung: Texte
  - Lagerverwaltung: Artikel, Aufträge
  - CAD-System: Technische Bauteile
- Mehrfache Speicherung (Redundanz):
  - Erhöhter Speicherplatzbedarf
  - „Vergessen“ von Änderungen
  - Keine einheitliche Darstellung

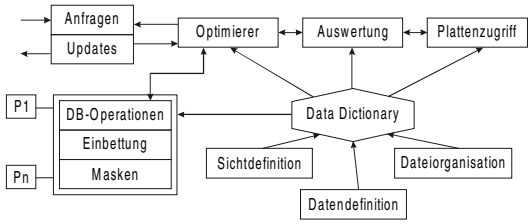
Datenredundanz - Probleme

- Keine effiziente Verarbeitung durch andere Systeme
- Kein paralleler Zugriff ohne Beeinflussung
- Abhängigkeit von interner Datendarstellung
- Eingeschränkter Datenschutz und Datensicherheit

Mit Datenbanken: Datenintegration

- Alle Systeme arbeiten auf denselben Daten
  - Datenbankentwurf, Datendefinition
- Bsp.: Adressen/Artikel nur einmal gespeichert
- Vorteile:
  - Effiziente Verwaltung (Anfragesprachen, Optimierung)
  - Paralleler Zugriff (Transaktionskonzept)
  - Datenunabhängigkeit (3-Ebenen-Konzept)
  - Datenschutz (kein unbefugter Zugriff)
  - Datensicherheit (kein Datenverlust)

Architektur eines Datenbanksystems



Funktionen

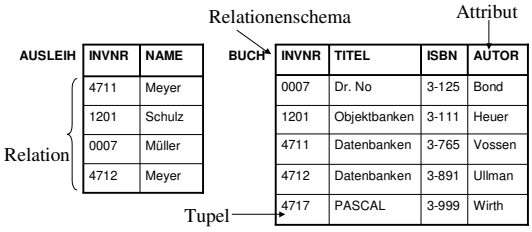
- Dateiorganisation: Definition der internen Dateiorganisation und Zugriffspfade
- Datendefinition: Konzeptuelle Datendefinition
- Sichtdefinition: Definition von Benutzersichten
- Masken: Entwurf von Masken und Menüs
- DB-Operationen: Anfrage- und Update-Operationen

Funktionen

- Einbettung: Einbettung dieser Operationen in Anwendungsprogramme
- P1, ..., Pn: verschiedene Anwendungsprogramme
- Optimierer und Auswertung: effiziente Umsetzung dieser Operationen
- Plattenzugriff: Plattenzugriffssteuerung

Relationenmodell

- Datenbank = Menge von Tabellen



Integritätsbedingungen

- Relationenschema + lokale Integritätsbedingungen
  - INVNR ist Schlüssel für BUCH
    - Darf nicht doppelt vergeben werden
- Datenbankschema = Menge von Relationenschemata + globale Integritätsbedingungen
  - INVNR in AUSLEIH ist Fremdschlüssel bzgl. BUCH
    - INVNR muß Schlüssel in BUCH sein

Anfrageoperationen

- Selektion: Tupel auswählen
  - SEL[NAME=´Meyer´] (AUSLEIH)

INVNR	NAME
4711	Meyer
4712	Meyer

- Projektion: Attribute auswählen
  - PROJ[INVNR, TITEL] (BUCH)

INVNR	TITEL
0007	Dr. No
1201	Objektbanken
...	...

Anfrageoperationen

- Verbund (Join): Tabellen über gleichbenannte Attribute und gleiche Werte verknüpfen
  - PROJ[INVNR, TITEL](BUCH) JOIN SEL[NAME=´Meyer´](AUSLEIH)

INVNR	TITEL	NAME
4711	Datenbanken	Meyer
4712	Datenbanken	Meyer

- Weitere Operationen: Vereingung, Differenz, ...
- Alle Operationen beliebig kombinierbar

Sprachen

- Anfragesprache
  - Interaktive Formulierung und Ausführung von Anfragen
  - Relationenalgebra + Funktionen (SUM, MIN, MAX, COUNT, ...) + arithmetische Operationen
- Update-Komponente
  - Interaktives Eingeben, Löschen und Ändern von Tupeln
  - Prüfung der Integritätsbedingungen

Benutzersichten

- Speichern von häufig vorkommenden Anfragen als „virtuelle“ Tabelle mit „Sichtnamen“
- MEYERS :=
  PROJ[INVNR, TITEL](BUCH) JOIN
  SEL[NAME=´Meyer´](AUSLEIH)
- Tabelle MEYERS

INVNR	TITEL	NAME
4711	Datenbanken	Meyer
4712	Datenbanken	Meyer

Prinzipien

- DBMS: Datenbank-Management-System
- DBS: Datenbanksystem (DBMS + Datenbank)
- Prinzip:
  - 3-Ebenen-Architektur (physische und logische Datenunabhängigkeit)
  - Trennung zwischen Schema (Tabellenstruktur) und Instanz (Tabelleninhalt)

Codd´sche Regeln

1. Integration
   - Einheitliche, nichtredundante Datenverwaltung
2. Operationen
   - Speichern, Suchen, Ändern
3. Katalog
   - Zugriff auf Datenbankbeschreibung im Data Dictionary
4. Benutzersichten
5. Konsistenzüberwachung
   - Korrektheit des Datenbankinhalts

Codd´sche Regeln

5. Datenschutz
   - Ausschluß unauthorisierter Zugriffe
6. Transaktionen
   - Mehrere DB-Operationen als Funktionseinheit
7. Synchronisation
   - Parallele Transaktionen koordinieren
8. Datensicherung
   - Wiederherstellen von Daten nach Systemfehlern

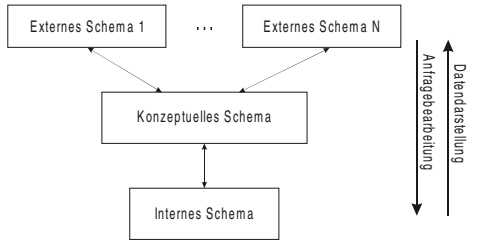
Entwicklungslinien

- 60er Jahre
  - DBS basierend auf hierarchischem Modell, Netzwekmodell
    - Zeigerstrukturen zwischen Daten
    - Schwache Trennung interne/konzeptuelle Ebene
    - Navigierende DML
- 70er und 80er Jahre
  - Relationale DBS
    - Daten in Tabellenstrukturen
    - 3-Ebenen-Konzept, Deklarative DML

Entwicklungslinien

- (80er und) 90er Jahre
  - Wissensbanksysteme
    - Daten in Tabellenstrukturen
    - Stark deklarative DML
    - Integrierte Datenbankprogrammiersprache
  - Objektorientierte DBS
    - Daten in komplexeren Objektstrukturen
    - Deklarative oder navigierende DML
    - Oft integrierte Datenbankprogrammiersprache
    - Oft keine vollständige Ebenentrennung

Schema-Architektur



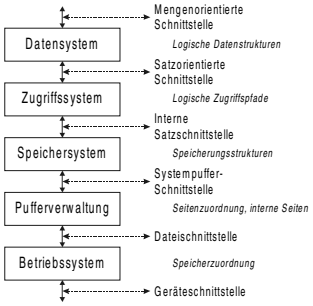
Datenunabhängigkeit

- Stabilität der Benutzerschnittstelle gegen Änderungen
- **Physisch**: Änderungen der Dateiorganisationen und Zugriffspfade haben keinen Einfluß auf das konzeptuelle Schema
- **Logisch**: Änderungen am konzeptuellen (und gewissen externen) Schemata haben keine Auswirkungen auf andere externe Schemata und Anwendungsprogramme

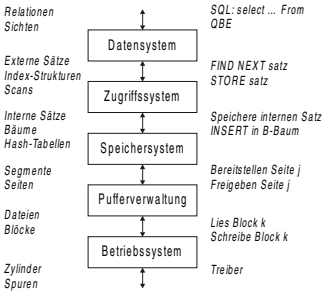
Fünf-Schichten-Architektur

- System-Architektur: Beschreibung der Transformationskomponenten [Senko, Härder]
  - Mengenorientierte Schnittstelle (dekl. DML)
  - Satzorientierte Schnittstelle (Sätze, nav. Zugriff)
  - Interne Satzschnittstelle (Manipulation von Sätzen)
  - Pufferschnittstelle (Seiten, Seitenadresses)
  - Datei- oder Seitenschnittstelle (Holen und Schreiben)
  - Geräteschnittstelle (Spuren, Zylinder)

Funktionen



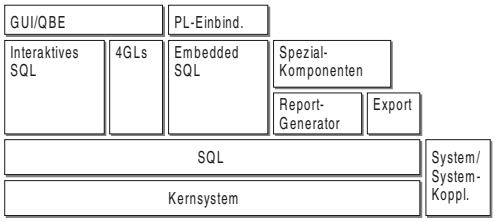
Objekte und Operationen



Relationale Systeme

- 3-Ebenen-Architektur nach ANSI-SPARC
- Einheitliche Datenbanksprache (SQL)
- Einbettung von SQL in Programmiersprachen
- Werkzeuge für
  - Definition, Anfrage und Darstellung von Daten
  - Entwurf von DB-Anwendungsprogrammen
  - Benutzerinteraktion
- Kontrollierter Mehrbenutzerbetrieb
- Datenschutz- und Sicherungsmechanismen

Anwendungsarchitekturen



Multimedia

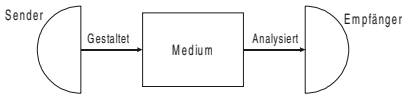
- Begriffe
- Klassifikation
- Multimedia-DBMS

Signale und Medium

- **Signal** := „der eine Nachricht übertragende (und damit Information wiedergebende) Verlauf einer physikalischen Größe“ [Bauer, Goos]
- **Medium** := Nachrichtenträger
  - Mittelglied, Mittler [Duden]
  - Vermittelt zwischen Kommunikationspartnern
  - Sender: erzeugt Signale auf Medium
  - Empfänger: nimmt Signale auf zur
    - Interpretation, Wandlung, Weiterleitung

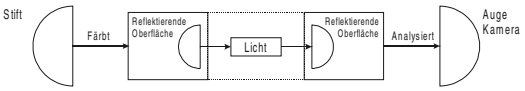
Physikalische Medien

- Konkrete Medien
  - z.B. Schall in festen, flüssigen und gasförmigen Stoffen
  - z.B. elektromagnetische Wellen
  - Bilden Basis für abstrakte Medien



Abstrakte Medien

- Bauen auf anderen Medien auf
  → Hierarchien
- Nutzen andere Sender/Medium/Empfänger-Einheiten als Medium
  - z.B. Folien als Medium



Weitere Eigenschaften von Medien

- Darstellungsmächtigkeit
  - z.B. Geruch vs. Bild
- Medienübersetzung oft möglich
- Komposition möglich
  → **Multimedia**

Klassifikationen

- Nach Sinnesorganen
  - Visuell (optisch)
    - Sprachlich (symbolgebunden), Nicht-sprachlich
  - Akustisch
  - Haptisch (Tasten)
  - Gustorisch (Schmecken)
  - Olfaktorisch (Riechen)
- Nach Zeit
  - Relativ zeitinvariant: Text, Bild, Fotos
  - Relativ zeitgebunden: Video, Sprache

Medien auf Computern

- Text
- Grafik
- Rasterbild
- Tonaufnahmen (Audio)
- Bewegtbild (Video)

VITA = **V**ideo + **I**mage + **T**ext + **A**udio

Weitere Begriffe

- **Medien-Objekt:** computerverwaltete logische Informationseinheit in einem Medium
- **Multimedia-Objekt:** ~ in einem oder mehreren Medien
- **Medien-Daten bzw. Multimedia-Daten:** heterogene Menge von Medien- bzw. Multimedia-Objekten
- **Multimedia-System:** Computersystem, das an der Benutzeroberfläche verschiedene Medien zum Informationsaustausch unterstützt

Multimedia-DBMS

- MMDBMS = DBMS +
  - Hohe Kapazität
  - Hohe Performance
  - Unterstützung von Multimedia-Datentypen (Datentyp = Struktur + Operationen)
  - Kombination von
    - Konventionellen DBMS-Fähigkeiten
    - Unterstützung von hierarchischen Speichersystemen (RAM, Festplatte, CD, ...)
    - Information Retrieval