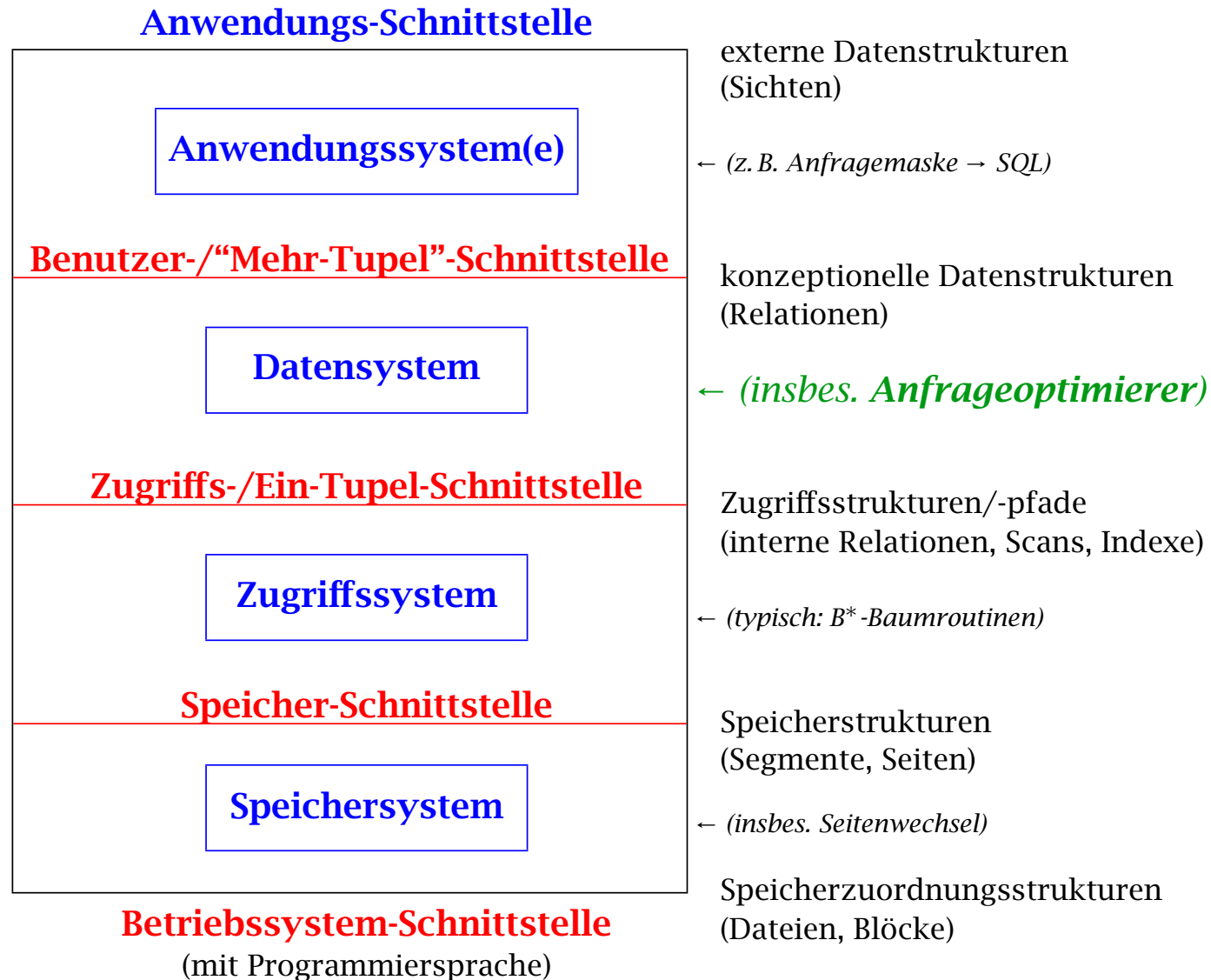


Kapitel 8

Ausblicke

8.1 Typischer Aufbau von relationalen DBMS



An den **Schnittstellen** werden folgende **Objekte**
(und zugehörige Operationen darauf) angeboten:

- **Mehr-Tupel-Schnittstelle:** Relationen (Tabellen), z. B.:

BESTELLUNG	<u>KName</u>	<u>Ware</u>	Menge
			A

KUNDE	<u>KName</u>	KAdr	Kto

Objekte an Schnittstellen (Forts.)

- **Ein-Tupel-/Zugriffs-Schnittstelle:**

interne Relationen mit int. Tupeln
fester Länge, etwa für Bestellungen

“Zugriffspfade”:

Indexe,
z. B. I(BESTELLUNG(Ware)):

•	Kaffee
•	Kakao
•	Milch

interne Relationen mit int. Tupeln
variabler Länge, etwa für Kunden

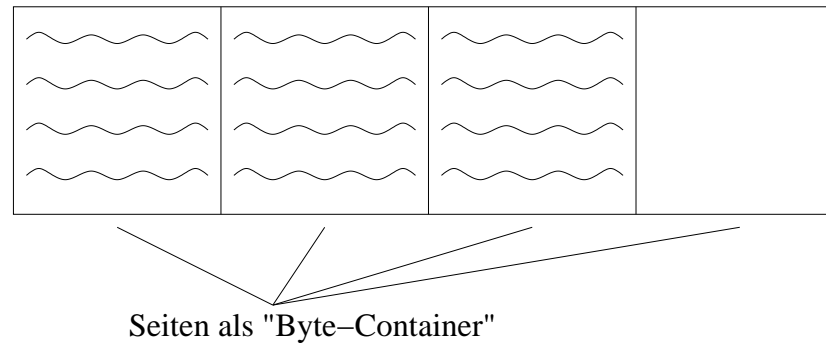
Relationen-

Index-

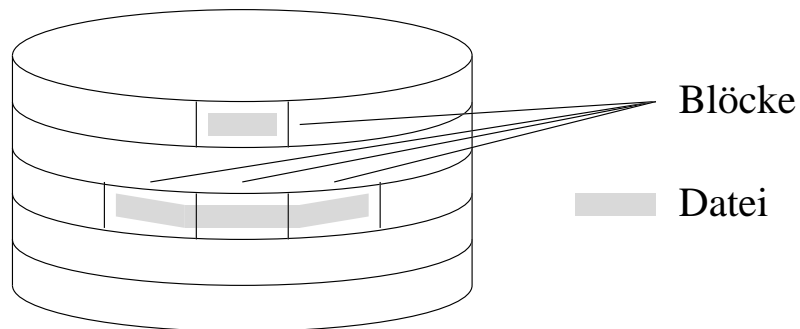
Scans

Objekte an Schnittstellen (Forts.)

- **Speicher-Schnittstelle:** linearer, in Seiten unterteilter Speicher¹



- **Betriebssystem-Schnittstelle** (bzw. Schnittst. zum externen Speicher)²



¹Idealer virtueller Speicher der gesamten Datenbank. Eine Seite kann genau einen Block des externen Speichers aufnehmen; deshalb spricht man auch von "logischen" und "physischen" Blöcken. Das unterliegende Speichersystem muss implementieren, dass jede benötigte Seite bei Bedarf im internen Speicher verfügbar ist. Da der interne Speicher aber nicht alle Blöcke des externen Speichers gleichzeitig aufnehmen kann, muss versteckt mit einem kleineren Puffer und mit Seitenwechselstrategien gearbeitet werden.

²Physisch kann eine Datenbank als eine einzige Datei im Sinne des Betriebssystems abgelegt sein !

Aufgaben des Datensystems

Implementierung der “Mehr-Benutzer-/Mehr-Tupel-Schnittstelle”, d. h.:

Frontends:

- Dialogführung mit Endbenutzern
- Vorübersetzung von Anwendungsprogrammen
- Menü/Masken-Oberfläche
- u.a.

Kern:

- Übersetzung von Anfragen usw. in die Ein-Tupel-Schnittstelle:
 - syntaktische, semantische Analyse gemäß DB-Katalog³ einschließlich Zugriffs- und Integritätsüberwachung bei Updates
 - Übersetzung in eine interne Darstellung, z. B. in Terme der Relationenalgebra
 - Optimierung, insbes. Ausnutzung von Zugriffspfaden
 - Zusammensetzung von Algorithmen für Relationenoperationen

³Im Vergleich zur Übersetzung üblicher Programme agiert der DB-Katalog wie eine dauerhaft gespeicherte (“persistente”) Symboltabelle.

8.2 Objekt-relationale DBS (Erweiterungen relationaler DBMSe)

Wichtigste neue Konzepte: (als SQL-Erweiterungen)

- komplexe Objekte, u.a. geschachtelte Tabellen
- Methoden (als benutzerdefinierte Funktionen)
- Vererbung, u.a. Typ- und Tabellenhierarchien
- Erweiterbarkeit um neue Basisdatentypen

Treibende Kräfte:

- Nicht-Standard-Anwendungen (vgl. Kap.1):
komplexe Objekte, komplexe Algorithmen, lange Transaktionen
- Nutzung komplexerer Daten auch für Standard-DB-Anwendungen
- objektorientierte Programmierung

Typkonstruktoren (komplexe Objekte)

- Tupel bzw. Records
- Mengen
- Referenzen bzw. Zeiger (auf Objekte in Tabellen)
- ... (evtl. noch andere, z.B. Arrays)
- Tabellen = Container für Mengen von Tupelobjekten

Beispiel:

```
create type phone (area ..., number ..., description ...);
```

```
create type car (name ..., year ..., license ...);
```

```
create type dept
```

```
  (dname      varchar(30),
```

```
   floor      int,
```

```
   phone      phone,
```

```
   cars        setof(car),
```

```
   mgr_ref     ref(emp),
```

```
   workers     setof(ref(emp)) );
```

```
create table DEPARTMENT of type dept;
```

```
create table EMPLOYEE of type empl;
```


Beispiel für Nicht-Standard-Daten mit ungewöhnlicher Anfrage

„State of California Department of Water Resources (DWR)“

```
create table DWR-SLIDES
    (id      int,
     date    date,
     caption document,
     picture photo_CD_image);
```

```
create table LANDMARKS
    (name    varchar(30),
     location point);
```

```
select id from DWR-SLIDES P, LANDMARKS L, LANDMARKS S
where sunset (P.picture)
and    contains (P.caption, L.name)
and    L.location  $\approx$  S.location
and    S.name = 'Sacramento';
```

Welche Dias zeigen Sonnenuntergänge an Landmarken in der Nähe von Sacramento?

Beispiele für neue Datentypen

- 2D/3D Spatial **DataBlade Modules**⁴:
z. B. für geographische Informationssysteme
 - Typen: point, line, polygon, path, circle, rectangle, ...
 - Funktionen (polymorph): distance, contained, overlaps, make, ...
- Image DataBlade Module: z. B. für digitale Bibliotheken
 - Typ: image (Speicherformate flexibel)
 - Funktionen: rotate, transpose, flip, crop, ...
enhance, oil_painting, ...
plus, minus, multiply, intersection, union, ...
(color) histogram, edge (detection), similarity, ...
- Text DataBlade Module

⁴Sprachgebrauch von Illustra, dem Vorläufer der objektrelationalen DBMSe

Beispiele für neue Datentypen (Forts.)

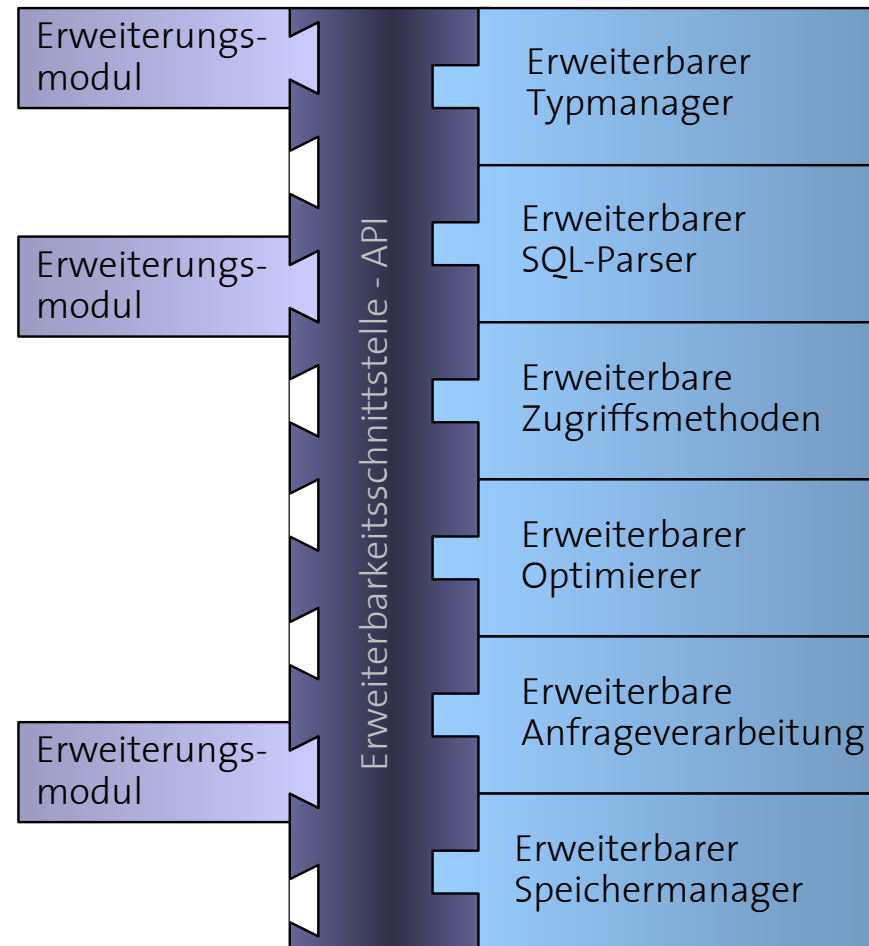
- TimeSeries DataBlade Module: z. B. für Börsenanwendungen
 - type: `time_series_of(...)`
 - functions: `previous`, `next`, `last`, `append`, ..., `moving_avg`, ...
- Web DataBlade Module
- ...

Solche Erweiterungsmodule heißen bei Oracle "**Data Cartridges**" und bei IBM "**Extender**". Diese werden in vorbereiteter Form zum Kauf angeboten oder können durch (fortgeschrittene) Benutzer selber entwickelt werden.

~> **erweiterbare DBMS**

Erweiterungsmodule

- **Serverseitige Pakete, die Funktionalität für bestimmte Anwendungsfelder bieten**
- **Kommerzielle Module (DataBlades, Cartridges, Extenders) beinhalten**
 - Spezielle, benutzerdefinierte Datentypen und Operationen
 - Spezielle Indextypen und Zugriffsmethoden
 - Regeln und Informationen für den Anfrageoptimierer



8.3 Einige weitere Stichworte zu DBS-Entwicklungen

- “deduktive”, “aktive” Datenbanksysteme
- “temporale” Datenbanksysteme
- “Real-Zeit-”, “eingebettete” Datenbanksysteme
- “Data Warehouses”, “Online Analytical Processing” (OLAP)
- “Data Mining”, “Knowledge Discovery”
- “Data Streaming”
- “maßgeschneiderte” (tailored) Datenbanksysteme,
DBS für Web-Anwendungen (z.B. Suchmaschinen, Webshops), “NoSQL”
- ...

8.4 Nachfolgende Lehrveranstaltungen

- **Datenbanksysteme [II]** (WS, Lipeck)
(Bachelor: Grundlagen der Informatik, Wahlpflicht;
Voraussetzung für DBS-Vertiefungen!)
 - Ausdrucksfähigkeit von relationalen Anfragesprachen
 - Anfragebearbeitung und -optimierung
 - Relationaler Datenbankentwurf (Normalisierung)
 - Physischer Datenbankentwurf
 - Verteilte Datenbanksysteme
 - Transaktionsmanagement

Nachfolgende Lehrveranstaltungen (Master)

- **Datenbanksprachen** (jährlich WS, Brüggemann, next: WS 14/15)
 - Logische Datenbanken, Optimierung logischer Anfragen
 - XML und Datenbanken
- **Integration von Datenbanken** (2-jährlich WS, Lipeck, next: WS 14/15)
- **Data Mining** (2-jährlich SS, Lipeck, next: SS 15)
- **Mehrdimensionale Datenbanken = Temporale und Räumliche DBS**
(2-jährlich, Lipeck, next: WS 15/16)
- **NoSQL-Datenbanken** (2-jährlich? SS, Lipeck, next: SS 16?)
- **Seminar oder Projektseminar zu DBS** (ca. jährlich, next: SS 14)
- **Datenbank-Praktikum** (unregelmäßig)

8.5 Spektrum der Klausuraufgaben

- ja/nein-Wissensfragen
- im EER-Modell modellieren
- EER- in relationales DB-Schema transformieren
- • • Anfragen formulieren, verstehen und übersetzen:
verbal \leftrightarrow Relationenalgebra \leftrightarrow SQL \leftrightarrow verbal
- Updates und Tabellendefinitionen in SQL schreiben
- PL/SQL-Programme lesen, ergänzen und schreiben
- verbal \rightarrow Integritätsbedingungen \rightarrow kritische Operationen / Trigger

Klausurbeginn: Do, 18. September 2014, 09:15 Uhr, F102/F303