MOBI-DBS-B: Datenbanksysteme Einführung

Vorlesung Sommersemester 2019

Tanya Braun, Universität zu Lübeck

Lehrauftrag SoSe 19, Universität Bamberg

Einführung

Inhalte

- Begriff der Datenbank (DB)
- Datenbankmanagementsystem (DBMS)
- Datenbanksystem (DBS)
- Charakteristika von DBs
- Datenabstraktion
- Datenunabhängigkeit
- DB-Sprachen
- DBS-Umgebung
- Phasen des DB-Entwurfs

Kompetenzen

- Die Rolle von Datenbanken (in IT-Architekturen) verstehen
- DB-Architekturen und DB-Modelle beschreiben

Daten, Daten, Daten

-bank

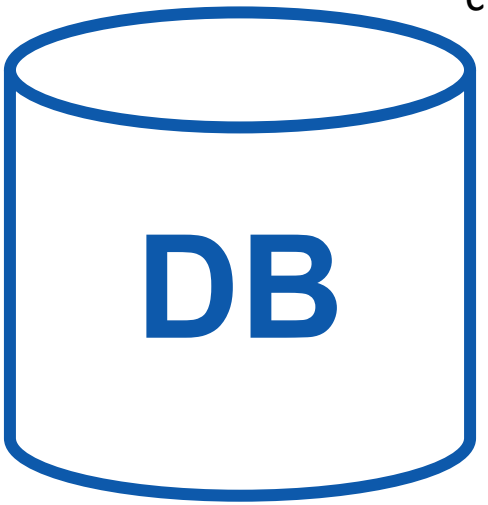
-bankmanagementsystem

-banksystem

Datenbank (DB)

Sammlung von Daten
- idR logisch zusammenhängend

"closed world assumption"



Miniwelt oder
Universe of Discourse

Daten → Information
(Interpretation)

Kennzeichen von Daten in DBs
- lange Lebensdauer (Jahre, Jahrzehnte)
- reguläre Strukturen
- große Datenobjekte, große Datenmengen
- stetig anwachsende integrierte Bestände (Giga-, Tera-, Petabyte)

Erstes Beispiel einer (relationalen) DB

- Datenbank für Inventar eines Weinkellers

Tabelle Weinkeller

Gestell	Sorte	Jahrgang	Anzahl_Flaschen
2	Franken	2009	5
1	Baden	2006	3
4	Rheinhessen	2007	10
1	Mosel	2013	2
2	Franken	2010	10

E. F. Codd, A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387, **1970**

Erste Anfrage

- Alle Infos zu Weinen, von denen es mindestens 4 Flaschen gibt

Tabelle Weinkeller

Gestell	Sorte	Jahrgang	Anzahl_Flaschen
2	Franken	2009	**5**
~~1~~	~~Baden~~	~~2006~~	~~3~~
4	Rheinhessen	2007	**10**
~~1~~	~~Mosel~~	~~2013~~	~~2~~
2	Franken	2010	**10**

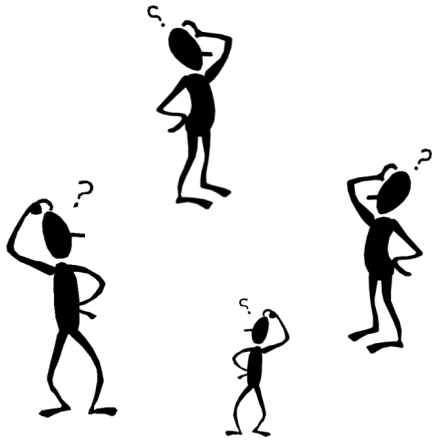
SELECT Gestell, Sorte, Jahrgang
FROM Weinkeller
WHERE Anzahl_Flaschen **>= 4**

Gestell	Sorte	Jahrgang
2	Franken	2009
4	Rheinhessen	2007
2	Franken	2010

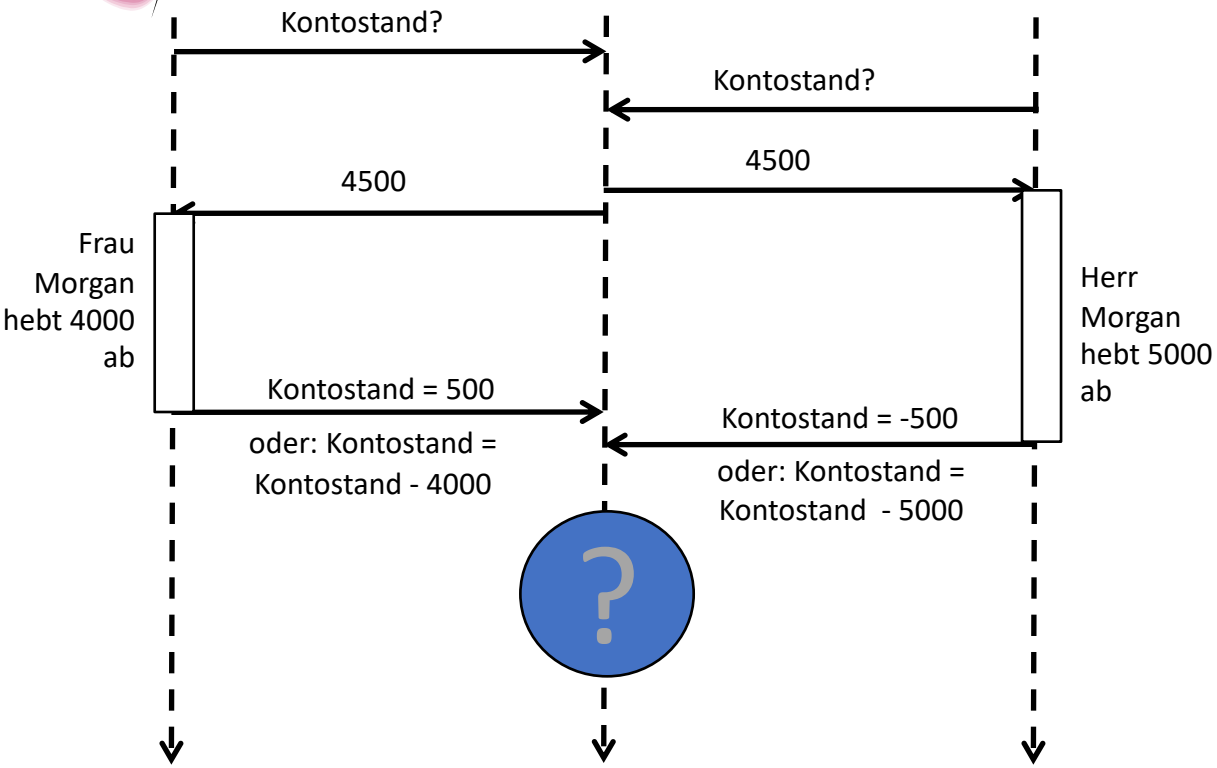
Datenbankmanagementsystem (DBMS)

- Software-System, um DBs zu verwalten
  - DB definieren (was kann alles drinstehen)
    - Definition der Miniwelt
  - DBs füllen
  - Daten abfragen
  - Daten verändern

- Viele gleichzeitige Benutzer!



Viele gleichzeitige Benutzer...



DBMS: Eigenschaften von Transaktionen

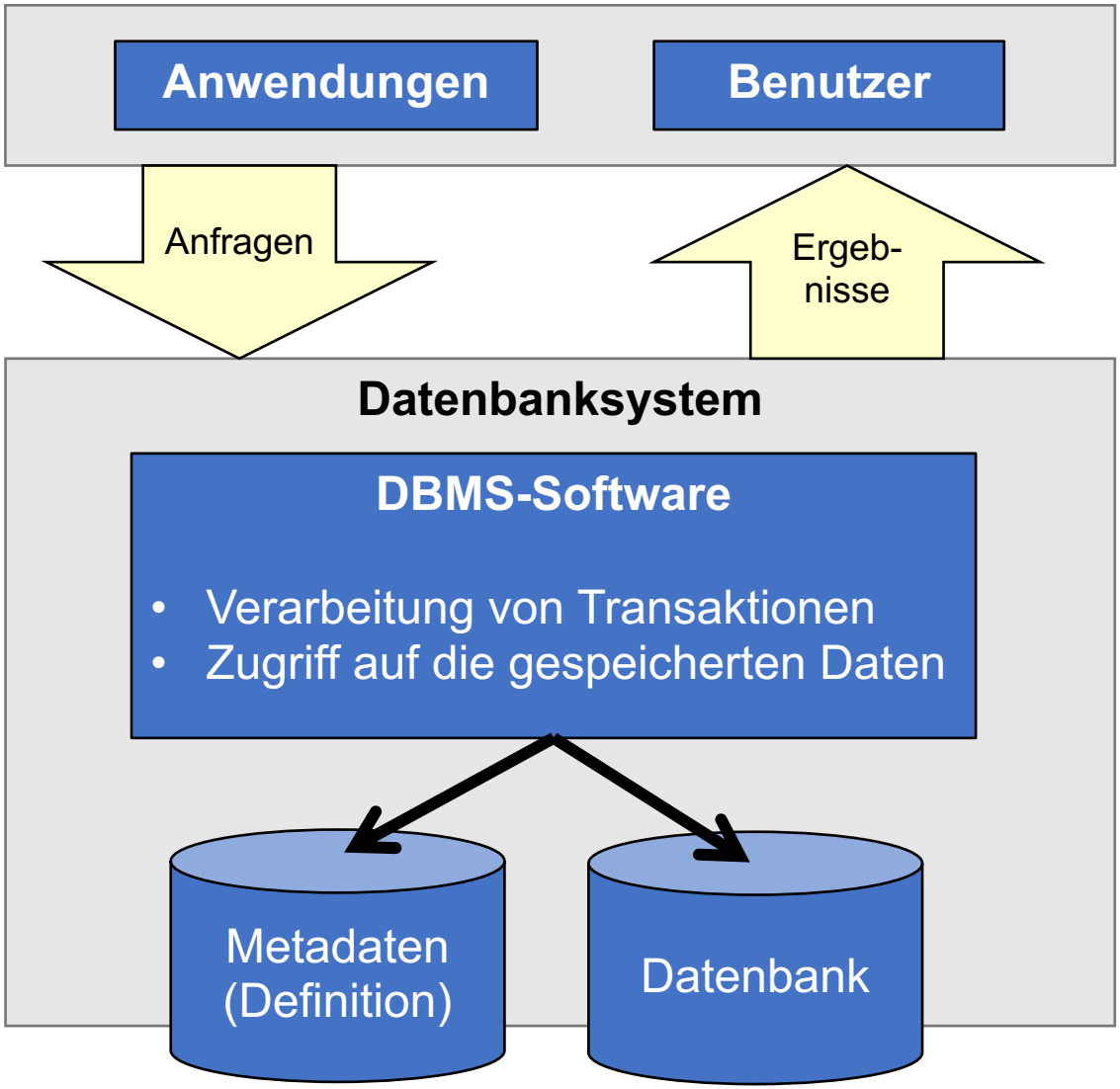
- Transaktion
  - zusammenhängende Abfolge von Datenbank-Befehlen
  - z.B. Kontostand abfragen, Überweisung von Konto 1 auf Konto 2

- **ACID**-Eigenschaften (kurz, später im Kapitel „Transaktionen“ mehr)
  - **A**tomicity (Atomarität):
    Alles oder nichts
  - **C**onsistency (Konsistenz):
    Vorher OK, hinterher OK
  - **I**solation (Isolation):
    Jeder denkt, er sei alleine auf der DB
  - **D**urability (Dauerhaftigkeit):
    Transaktionen bestätigt? Dann sind die Daten jetzt sicher

DBMS: Weitere Funktionen

- Sicherheitsmechanismen
- Fehlerbehandlung
- Integritätskonzepte
- Verschiedene Sichten auf die DB
- Optimierung von Anfragen

Datenbanksystem (DBS) = DBMS + DB



Daten, Daten, Daten

-abstraktion

-modell

-unabhängigkeit

Datenabstraktion

- abs-trahere = "abziehen, entfernen"
  → Weglassen von Einzelheiten, Überführen in etwas Einfacheres
- Abstraktion: ein Grundprinzip der Informatik!
  - Abstraktionsschicht verbirgt Einzelheiten/Aufgaben
    → Unabhängigkeit für darüber liegende Schichten
- DBMS bieten:
  - Abstraktion von der Speicherung
  - Abstraktion von Anwendungen (= mehrere Anwendungen möglich)
  - eine konzeptuelle Sicht auf die Datenbanken
- Basis für Abstraktion: Datenmodell

Datenmodell

- DB: Sammlung von Daten
- DB-Struktur/Schema:
  Elemente zur Definition, welche Daten möglich sind
  - Datentypen (z.B. String, Integer, ...)
  - Beziehungen (z.B. „Jeder Mitarbeiter hat einen Vorgesetzten.“)
  - Einschränkungen (z.B. Das Geburtsdatum muss in der Vergangenheit liegen)

- Datenmodell
  - Elemente zur Definition einer Datenbankstruktur
  - Basis-Operationen zur Abfrage und Änderung von Daten
  - Erweiterungen durch benutzerdefinierte Operationen

- Populäres Beispiel: das **relationale Datenmodell**

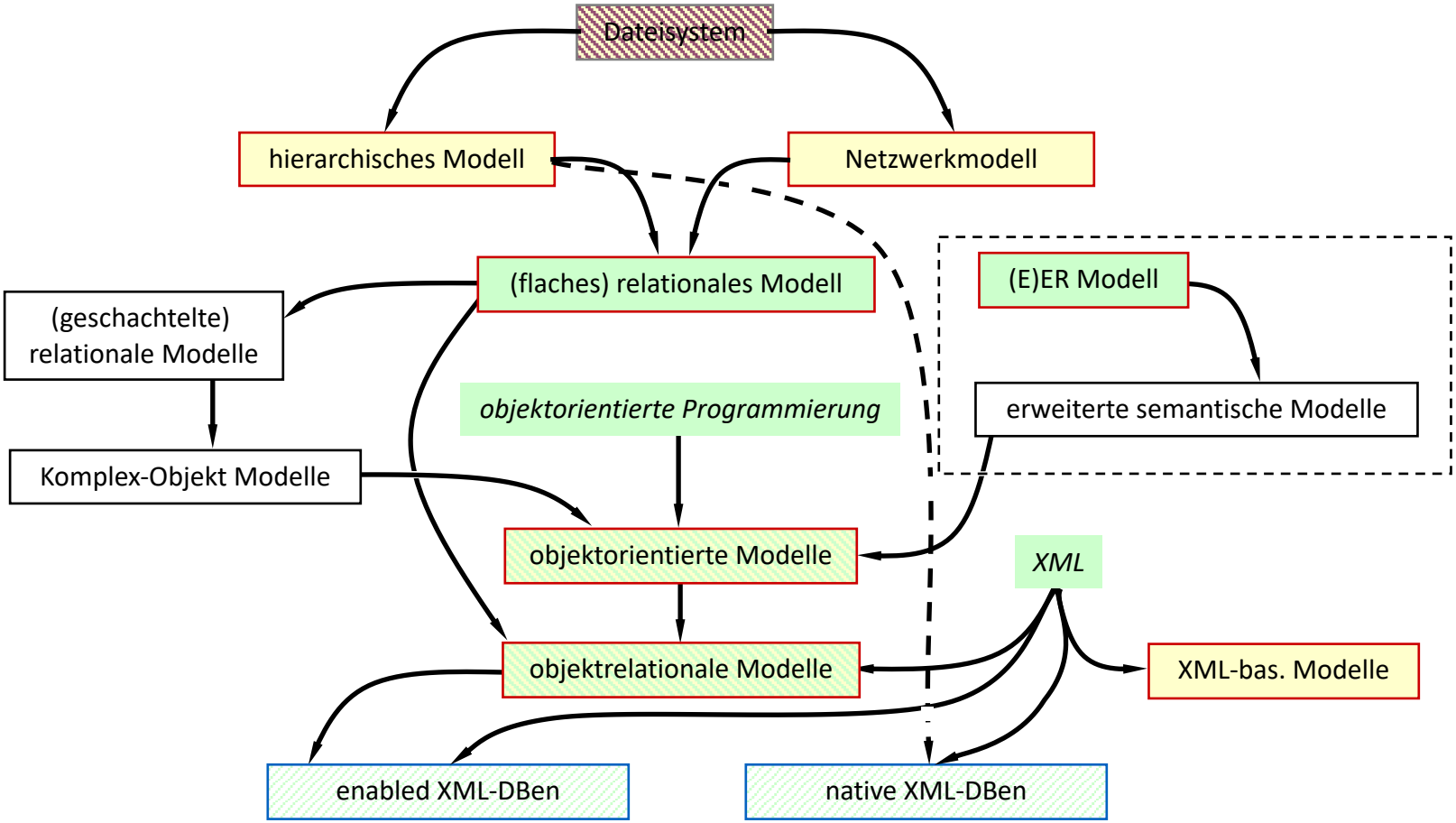
Modellierungsebenen von Datenmodellen

- Konzeptuelle Datenmodelle
  - Zur Definition der Miniwelt (→ Kundenanforderungen)
  - Beispiel: Entity-Relationship-Modell (ER-Modell)

- Logische Datenmodelle
  - Spezifikation, die leicht implementiert werden kann (→ für Entwickler, unabhängig vom DBMS)
  - Beispiel: Relationales Datenmodell

- Physische Datenmodelle
  - Spezifikation der konkreten Datenspeicherung (für ein konkretes DBMS)

Eine kurze Geschichte der Datenmodelle



Hierarchisches Modell

- Ältestes klassisches Datenmodell
- Ein Datensatz und alle von ihm abhängigen Datensätze → hierarchische Einheit

- Natürliche Hierarchie:
  - Unistrukturdatei (Universität -> Fakultät -> Institut -> Department)

- Künstliche Hierarchie:
  - Artikeldatei (Artikel -> Lieferant)

  könnte auch sein:
  - Artikeldatei (Lieferant -> Artikel)

Eigenschaften des Hierarchischen Modells

- Modellierung von Beziehungen:
  - 1 : 1 - Elternelement wird Kindelement zuordnet
  - 1 : n - Elternelement wird mehreren Kindelementen zugeordnet
  - m : n - Nicht direkt darstellbar
  - Keine Zyklen

- Zugriff
  - Entlang der Hierarchie sehr effizient
  - „Quer dazu“ sehr ineffizient
    - Beispiel: Teilnehmerdatei (Vorlesung -> Student)
      - Alle Studierenden von MOBI-DBS-B → prima
      - Alle Vorlesungen von Martha Mustermann → eeek …
- Fazit:
  - Gut bei klar definiertem Einsatz (z.B. directories, index-Verwaltung, …)
  - Ineffizient bei allgemeinem Einsatz (wg. mangelnder Flexibilität)

Netzwerkmodell

- Schreibt das hierarchische Modell fort:
  - Ein Element kann mehreren Gruppen zugeordnet werden
    - Student in Vorlesung, in Studiengang, in Semester, …
  - Mehrere Wurzelelemente möglich

  → Jetzt auch n:m Beziehungen; allerdings nicht direkt:
  Kursbelegung (Student → Belegung ← Kurs)

- Nachteile:
  - Wird leicht unübersichtlich
  - Für Abfragen ist genaue Kenntnis der Struktur nötig
  - Sequentielles Lesen („alle Studierenden einer Vorlesung“) wird ineffizienter

Das (flache) relationale Modell

- Für die Praxis das wichtigste Datenmodell!
- Tabellen (mit Attributen = Spaltenüberschriften) sind Container für komplexe Datenobjekte

Teilnehmer

MatrikelNr	Vorname	Nachname	Semester
4711	Martha	Mustermann	3
42	Arthur	Dent	21

- Beziehungen:
  - durch Gruppierung in Tabellen
  - durch wertebasierte Zusammenhänge zwischen Tabellen:

Vorlesung

VorlesungNr	Titel
42	MOBI-DBS-B
48151623	Altrömisches Abwasserecht

Belegung

VorlesungNr	MatrikelNr
42	4711
48151623	42

Physische Datenmodelle

- Physische Datenmodelle beschreiben konkret, wie die Daten anhand von Angaben zu
  - Datensatzformaten,
  - Datensatzanordnungen und
  - Zugriffspfaden

  physisch gespeichert werden (sollen)

- Ein Zugriffspfad ist eine Datenstruktur, welche die Suche nach Datensätzen in einer DB unterstützt/beschleunigt

- Beispiele: B-Bäume, B*-Bäume, R-Bäume, …

- Davon abstrahiert ein DBMS!

Schema / Instanz / DB-Zustand

- Schema (Intension)
  - Beschreibung der kompletten Struktur einer DB
  - Ändert sich (hoffentlich) selten
- Instanz (elementare Extension)
  - *Einzelne*, der vorgegebenen DB-Struktur entsprechende, aus konkreten Datenelementen bestehende Datensätze
- DB-Zustand (Gesamt-Extension oder Snapshot)
  - Die Gesamtheit der aktuell in einer DB gespeicherten Daten

Teilnehmer

MatrikelNr	Vorname	Nachname	Semester
4711	Martha	Mustermann	3

Vorlesung

VorlesungNr	Titel

Belegung

VorlesungNr	MatrikelNr

Datenunabhängigkeit

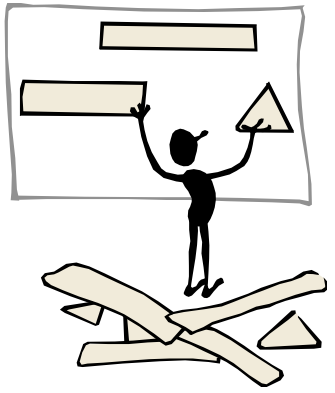
- Ein Schema kann geändert werden, ohne zwangsläufig auch auf der nächsthöheren Ebene Änderungen vornehmen zu müssen.

- Logische Datenunabhängigkeit
  - Ein konzeptuelles/logisches Schema ändern, ohne immer auch externe Schemata oder Applikationen ändern zu müssen.

Teilnehmer

MatrikelNr	Vorname	Nachname	Semester

- Physische Datenunabhängigkeit
  - Ein internes Schema ändern, ohne konzeptuelle/ logische und externe Schemata sowie Applikationen ändern zu müssen.



Beispiele für Änderungen

- Logische Datenunabhängigkeit
  - Ein konzeptuelles/logisches Schema ändern
  - Hinzufügen von Attributen und Tabellen zum konzeptionellen Schema
  - Verändern der Tabellenstruktur
  - DB-Erweiterung durch neue Datensatztypen/Datenfelder
  - DB-Reduktion/Streichung bestehender Datensatztypen oder Datenfelder
  - Erweiterung („Verschärfung“) oder Reduktion („Entschärfung“) von Einschränkungen der Schemata

  → Wirkt sich nur auf externe Schemata aus, die sie nutzen
- Physische Datenunabhängigkeit
  - Ein internes Schema ändern
  - Veränderung des Speicherortes
  - Änderung des Speicherformates
  - Anlegen/Löschen von Indizes (für Anfrageoptimierung)

  → Verbleiben die gleichen Daten in der DB, so muss das konzeptuelle/logische DB-Schema i.d.R. nicht angepasst werden.

Sprachen und Umgebung

In und um Datenbanken

DB-Sprachen

- Definition von DBs:
  - View Definition Languages (VDLs): extern
  - Data Definition Languages (DDLs): logisch
  - Storage Definition Languages (SDLs): intern

- Zugriff auf DBs
  (Einfügen, Ändern, Löschen und Anfragen von Datensätzen):
  - Data Manipulation Languages (DMLs)
    - Einfüge-, Änderungs- und Löschoperationen: Updates
    - Reine Anfragen: „Queries“
    - Alle Zugriffsarten: „Manipulation“

SQL – die Standard-DB-Sprache

- SQL : Structured Query Language
- Universal-Sprache für Datenbanken
  - VDL, DDL, SDL und DML in einem

Haupteigenschaften:

- Mengenorientiertes Arbeiten
  - Adressierung einer Menge von Datensätzen (zurückgegeben, geändert, gelöscht)
  - Menge kann auch nur ein Element enthalten

- Deklarativ
  - Angabe darüber, welche Daten man möchte
  - Keine Angabe darüber wie der Zugriff erfolgen soll (Abstraktion; → Optimierungsmöglichkeit für das DBMS!)

SQL-Beispiel

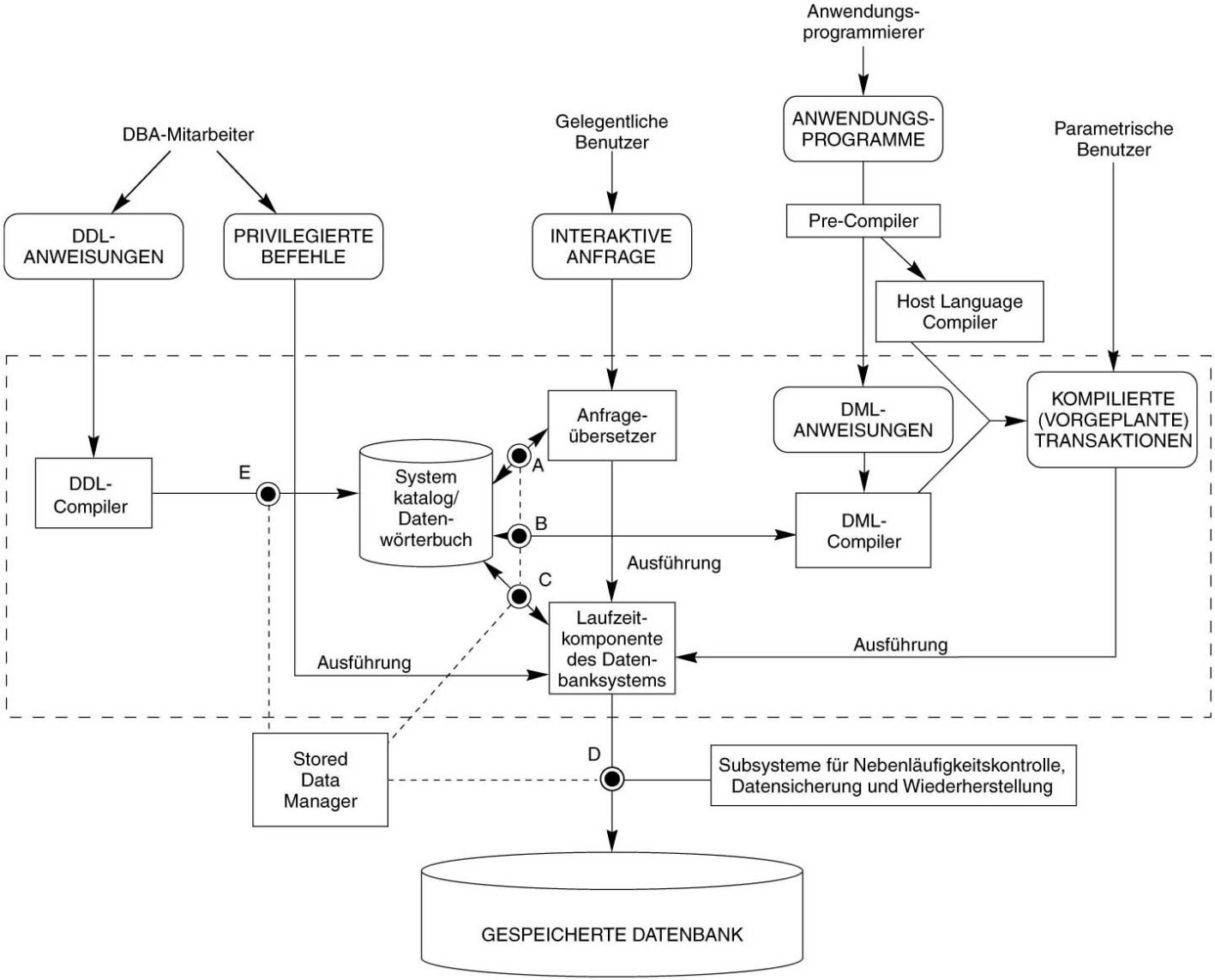
```
CREATE TABLE Student (
  Name          VARCHAR2(100) NOT NULL,
  StudentNumber NUMBER(10)    PRIMARY KEY,
  Class         NUMBER(2)     DEFAULT 1 NOT NULL,
  Major         VARCHAR2(10)
);

INSERT INTO Student (Name, StudentNumber, Class, Major)
VALUES ('Smith', 17, 1, 'CS');

INSERT INTO Student (Name, StudentNumber, Class, Major)
VALUES ('Brown', 8, 2, 'CS');

SELECT Name
FROM Student
WHERE Major = 'CS';
```

DBMS-Systemumgebung



Rückblick

- Was ist eine Datenbank?
  - DB stellt Aspekte eines (realen) Weltausschnitts dar (Anwendung).
  - DB ist logisch zusammenhängende Sammlung von Daten mit inhärenter Bedeutung.
  - DB wird i.d.R. für bestimmten Zweck entworfen, entwickelt und mit Daten gefüllt.
  - DB wird von autorisierten Benutzergruppen in zweckbezogenen Anwendungen verwendet.
- Charakteristika von Datenbanken
  - Persistente Speicherung großer Datenmengen
  - Metadaten in DB-Katalog; Integritäts- und Konsistenzbedingungen formulierbar; Abstraktion (Programm- und Datenunabhängigkeit)
  - Mehrbenutzerfähigkeit; Transaktionen (ACID-Eigenschaften); Datenschutz (Rechteverwaltung); Views (individualisierte Sichten auf Daten)
  - Recovery

Rückblick

- Datenabstraktion
  - Datenmodelle: konzeptuell, logisch und physisch.
  - Schema (Intension), Instanz und DB-Zustand (Extension).
- Datenunabhängigkeit
  - Möglichkeit, ein Schema auf einer Ebene zu ändern, ohne zwangsläufig auch auf der nächsthöheren Ebene Änderungen vornehmen zu müssen.
  - Genauere Betrachtung: logische und physische Datenunabhängigkeit.
- DB-Sprachen
  - VDL, DDL, SDL und DML → SQL
- DB-Systemumgebung