

Datenbanksysteme für Hörer anderer Fachrichtungen

Vorlesung im Sommersemester 2004
Dr. Birgitta König-Ries
Informatik III
Technische Universität München

Programm heute

- Begrüßung
- Organisatorisches
- Inhaltlicher Einstieg:
 - Warum überhaupt Datenbanksysteme?
- Inhaltliches:
 - die behandelten Themen im Überblick
- Formales:
 - Konventionen zur Notation
- Kapitel 1: Datenbankdienste

Organisatorisches (1)

- Webseite zur Vorlesung:
 - <http://www3.in.tum.de/lehre/SS2004/DBSother-Koenig/>
 - Termine
 - Folien
 - Übungsblätter
 - externes Material
 - alles, was sonst noch relevant sein könnte
- für weitere Fragen, Anregungen, Kritik,...:
 - koenigri@in.tum.de
 - FMI Gebäude in Garching Zimmer 02.11.044
 - von Montag bis Mittwoch im Prinzip jederzeit, lieber nachmittags, am liebsten nach Terminabsprache per mail
 - 089/289-17261

→ nach d. Vorlesung

Organisatorisches (2)

- Folien werden jeweils vor der Vorlesung auf der Webseite verfügbar sein.
- Es ist sinnvoll diese (entweder ausgedruckt oder im Rechner) zur Vorlesung mitzubringen.

→ spätestens mitte d. Woche
ppt pdfs

Organisatorisches (3): Übung

- 1 SWS
- Alternativen:
 - wöchentlich eine Stunde
 - vor der Vorlesung
 - 09:15 – 10:00 Uhr Übung
 - 10:15 – 11:45 Uhr Vorlesung
 - nach der Vorlesung
 - 09:15 – 10:45 Uhr Vorlesung
 - 11:00 – 11:45 Uhr Übung
 - ca. 14-tägig 2 Stunden am Stück nach der Vorlesung
 - 09:15 – 10:45 Uhr Vorlesung
 - 11:00 – 12:30 Uhr Übung
- Übungsblätter sind jeweils vorher auf der Webseite.
- Bitte vor der Übung bearbeiten!

SS 2004

B. König-Ries: Datenbanksysteme

5

Organisatorisches (4)

- Umfang 2 SWS + 1 SWS Übung (4 ECTS)
- Klausur: Dienstag, 20.07.2004, ~~09:15~~ 10:15 Uhr, 60 Minuten
- **Wichtig:** Bitte klären Sie, welche Voraussetzungen nach Ihren Prüfungsordnungen jeweils erfüllt sein müssen, damit die Vorlesung anerkannt wird (z.B. hinsichtlich Anwesenheit, Anmeldeformalitäten für die Klausur, ...) und mailen Sie mir dies (mit Angabe Ihres Studienfaches).

Koenigri@in.tum.de

SS 2004

B. König-Ries: Datenbanksysteme

6

Material zur Vorlesung

- Buch zur Vorlesung:
 - Alfons Kemper, André Eickler:
Datenbanksysteme: eine Einführung
Oldenbourg Verlag
- ergänzend:
 - Abeck, Lockemann, Schiller, Seitz:
Verteilte Informationssysteme, dpunkt-Verlag
(Teil I, III und IV)
- Die Folien stammen zu einem großen Teil ursprünglich aus der Vorlesung Kommunikation und Datenhaltung von Prof. Lockemann (Universität Karlsruhe (TH)). Vielen Dank!

SS 2004

B. König-Ries: Datenbanksysteme

7

Warum überhaupt Datenbanksysteme?

Problemstellung:

Sie werden an einem Uni-Institut als Hiwi eingestellt, um die Verwaltung der Institutsbibliothek zu digitalisieren. Bisher sieht die Bibliothek so aus:

Die Bücher, soweit sie nicht ausgeliehen sind, stehen alphabetisch nach Autor sortiert im Regal. Daneben steht ein Karteikasten. Der Kasten enthält für jedes Buch eine Karte. Auf der Karte sind die folgenden Angaben vermerkt:

Autoren oder Herausgeber,

Titel,

Schlüsselwörter

Erscheinungsjahr

Wenn ein Mitarbeiter ein Buch ausleiht, vermerkt er das mit Datum auf der Karte, bei Rückgabe streicht er seinen Namen wieder durch.

Sie sollen nun eine rechnergestützte Lösung entwickeln. Nehmen Sie an, Sie verwenden hierzu ein Dateisystem.

SS 2004

B. König-Ries: Datenbanksysteme

8

Warum überhaupt Datenbanksysteme?

- Welche Daten müssen Sie in Ihrem System speichern?

Titel, Autor, Schlagwort, Jahr Bücher
Name, Adresse, Tel.Nr. Benutzer

- Wie organisieren Sie diese Daten?

Anleihtdatum, Anleiher, ... , Vorbestellung
Verlängerung
Tabellen Struktur

SS 2004

B. König-Ries: Datenbanksysteme

9

Warum überhaupt Datenbanksysteme?

- Wie muss auf die Daten lesend zugegriffen werden?

→ Schlagwortsuche
Zugriff über Titel, Autor
Wer hat Buch ausgeliehen?

- Welche Änderungen müssen möglich sein?

Anleihen, Rückgaben
neue Bücher, Benutzer

SS 2004

B. König-Ries: Datenbanksysteme

10

Warum überhaupt Datenbanksysteme?

- Wie stellen Sie sicher, dass alles korrekt abläuft, wenn mehrere Benutzer gleichzeitig zugreifen?

{ Suche 'xmi'
mehres xmi-Buch
löschen -h-

- Wie sorgen Sie dafür, dass nichts verloren geht, wenn Ihr Rechner abstürzt?

SS 2004

B. König-Ries: Datenbanksysteme

11

Warum überhaupt Datenbanksysteme?

- Welche Vorteile böte es, ein Datenbanksystem zu verwenden?

-

SS 2004

B. König-Ries: Datenbanksysteme

12

Fazit

- Selbst eine relativ einfache Anwendung erfordert jede Menge Arbeit, wenn alles korrekt ablaufen soll.
- Vieles davon tritt bei vielen Anwendungen gleichzeitig auf.
- Wenig sinnvoll, wenn jeder Anwendungsprogrammierer neue Lösungen entwickelt.
- Datenbankmanagementsysteme bieten generisch (d.h. anwendungsunabhängig) häufig benötigte Funktionalität, z.B.
 - dauerhafte Speicherung von Daten
 - Beschreibung der Daten
 - Anfrageschnittstelle
 - Mehrbenutzersynchronisation
 - Fehlerbehandlung

SS 2004

B. König-Ries: Datenbanksysteme

13

Überblick über die behandelten Themen

- Was machen Datenbanksysteme?
 - Datenbankdienste
- Wie sind Datenbanksysteme aufgebaut?
 - Architektur von Datenbanksystemen
- Wie sind die Daten in Datenbanksystemen organisiert?
 - Relationales Modell (OO, XML)
- Wie werden die Daten in Datenbanksystemen beschrieben und manipuliert?
 - Relationalenalgebra
 - SQL
- Wie entwickelt man eine Datenbank?
 - DB Entwurf
- Wie stellt man sicher, dass Datenbankzugriffe korrekt ablaufen?
 - Transaktionsverwaltung
- Wie sorgt man dafür, dass Datenbanken schnell genug sind?
 - Implementierung
- Müssen immer alle Daten an einem Ort liegen?
 - Verteilung

SS 2004

B. König-Ries: Datenbanksysteme

14

Farbkodierung

Definition 1.1 (Transaktion)

Feiefen dfetieonten stenesoente
Defente

Satz 1.2 (Teref)

Feinen geingeneon

Beweis

Feitententnetnentent

Beispiel 1.3

Schedule $s = r_1(x)w_2(y)w_1(y)c_1$

Bemerkung 1.4

fefegee

SS 2004

B. König-Ries: Datenbanksysteme

15

Kapitel 1: Datenbankdienste

Einführung: Dienste
Dienstfunktionalität
Dienstmerkmale
Datenbanksysteme

viele Folien: © Prof. Lockemann, IPD, Uni Karlsruhe

SS 2004

B. König-Ries: Datenbanksysteme

16

Kapitel 1: Datenbankdienste

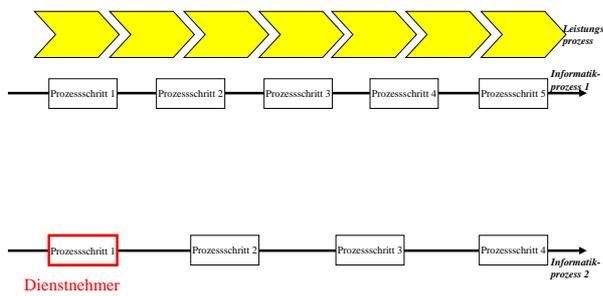
Einführung: Dienste

- Dienstfunktionalität
- Dienstmerkmale
- Datenbanksysteme

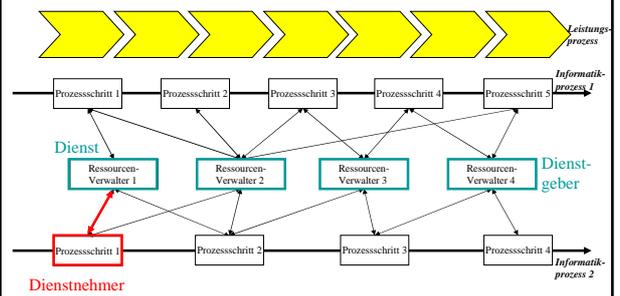
Leistungsprozesse und Ressourcen (1)

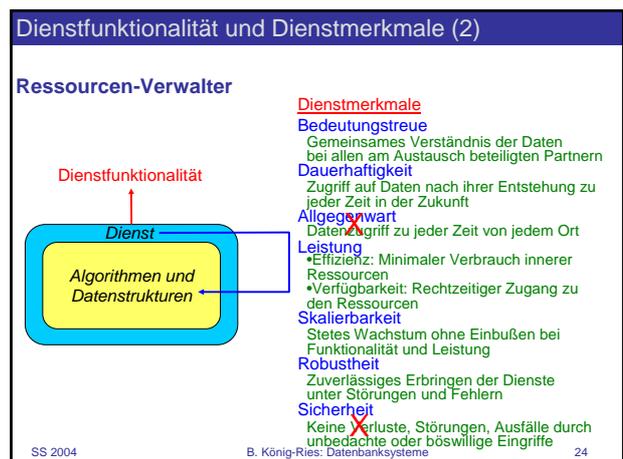
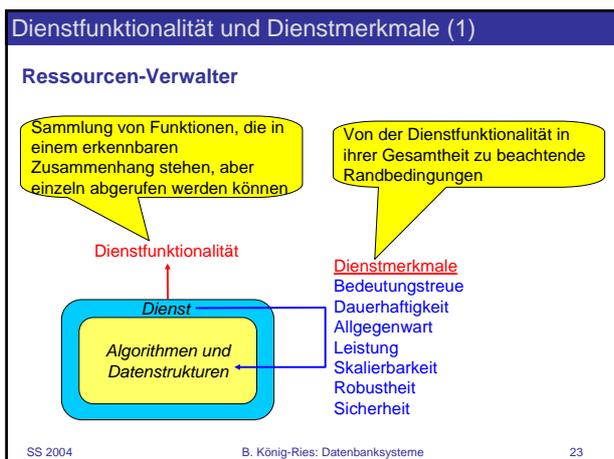
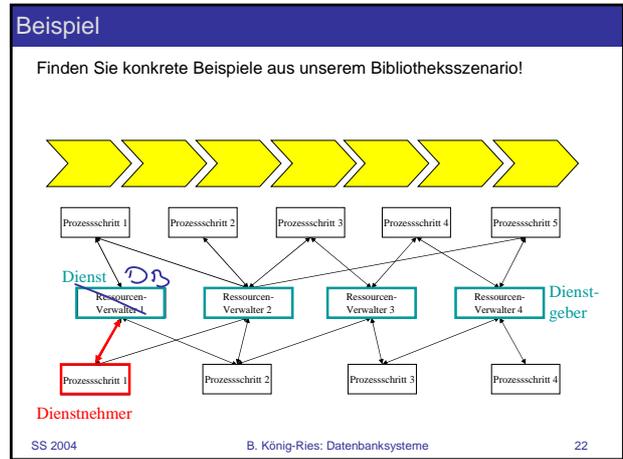
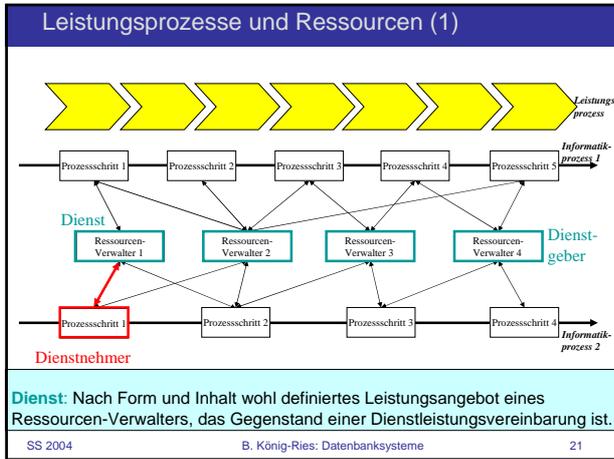


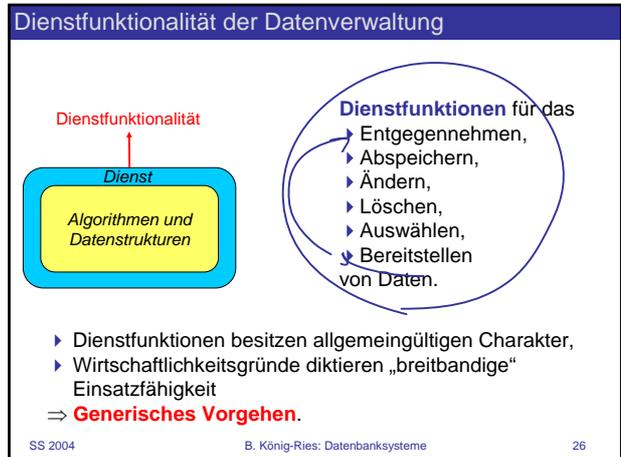
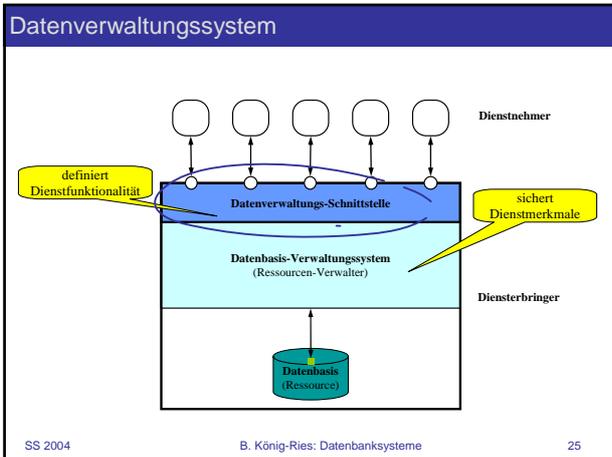
Leistungsprozesse und Ressourcen (1)



Leistungsprozesse und Ressourcen (1)







Kapitel 1: Datenbankdienste

- Einführung: Dienste
- Dienstfunktionalität**
- Dienstmerkmale
- Datenbanksysteme

SS 2004 B. König-Ries: Datenbanksysteme 27

- ### Datenmodell
- #### Aspekte der Dienstfunktionalität
- Beschreibung der zulässigen Datenbasiszustände
=> Generisch: Mittel zur inhaltsneutralen Strukturierung der **Datenbasis**
 - Beschreibung der zulässigen Zustandsübergänge, im wesentlichen in Form der anwendbaren Operatoren
=> Generische Operatoren
- Relationales Modell*
=: **Datenmodell**
- SS 2004 B. König-Ries: Datenbanksysteme 28

Zustände einer Datenbasis (1)

Grundgedanke: Datentyp = Menge zulässiger Zustände

- **Typ:** Universum von Objekten gleicher mathematischer Struktur.
- **Ausprägung (Instanz):** Element eines Typs.
- **Polymorphes Typsystem:** Universum von Typen, i.d.R. beschrieben durch:
 - die Festlegung gewisser **atomarer Typen**, z.B.
 - int = Menge der ganzen Zahlen,
 - bool = {true, false},
 - date = Menge der Daten des Gregorianischen Kalenders;
 - die Angabe von **Typkonstruktoren**, mit denen Typen zu neuen Typen kombiniert werden können, z.B.
 - record(t_1, t_2, \dots, t_n): Menge der Tupel mit Komponenten aus t_1, t_2, \dots, t_n ,
 - set(t) = Menge der Mengen mit Elementen aus t ,
 - list(t) = Menge der Listen mit Elementen aus t .

SS 2004

B. König-Ries: Datenbanksysteme

29

Zustände einer Datenbasis (2)

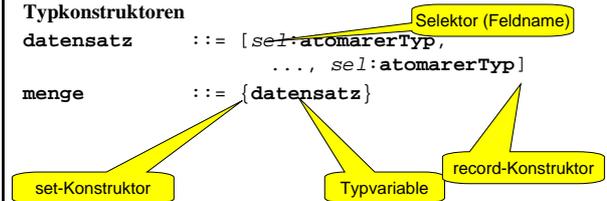
Beispiel für ein einfaches polymorphes Typsystem:

Atomare Typen int, float, string, time

Typkonstruktoren

datensatz ::= [sel:**atomarerTyp**,
..., sel:**atomarerTyp**]

menge ::= {**datensatz**}



SS 2004

B. König-Ries: Datenbanksysteme

30

Zustandsbeschränkungen

- Typkonstruktoren geben an, wie Zustände zu neuen Zuständen zusammengestellt werden können.
- Konsistenzbedingungen: Einschränkungen auf den konstruierbaren Zustandsmengen.
- **Polymorphe Konsistenzbedingung:** Generische Vorschrift für Konsistenzbedingungen als Teil des Datenmodells.

Beispiel: Jeder Datensatz in einer Menge besitzt einen **Schlüssel**, d.h. einen Wert, der ihn eindeutig unter allen Datensätzen der Menge identifiziert. Formal: Es existiert Funktion

key: **menge** × **atomarerTyp** → **datensatz**

SS 2004

B. König-Ries: Datenbanksysteme

31

Zustandsübergänge mittels Operatoren

- **Operatoren:** mathematische Funktionen, die auf Elemente von Typen angewendet werden können, z.B.
 - Gleichheitstest ($x = y$): anwendbar auf beliebige Typen
 - Anordnung ($x < y$): anwendbar auf Zahlen, Daten, Zeichenketten,...
 - arithmetische Operationen (+, -, ×, /): anwendbar auf Zahlen
 - logische Operationen (*and*, *or*, *not*): anwendbar auf boolesche Werte
 - Mengenoperationen (\cup , \cap , $-$): anwendbar auf Typen, die durch Anwendung des set-Typkonstruktors entstanden sind
- **Monomorphe** Operatoren können nur auf Elemente bestimmter Typen angewendet werden (z.B. *not*).
- **Polymorphe** Operatoren können auf Elemente unterschiedlicher (wenn auch nicht aller) Typen angewendet werden (z.B. =, < oder \cup).

SS 2004

B. König-Ries: Datenbanksysteme

32

Zustandsübergänge einer Datenbasis

- Ein polymorphes Typsystem hat zwingend polymorphe Operatoren, erfüllt also die Forderung nach generischen Operatoren.
- Beispiele : Abspeichern, Löschen von Ausprägungen von Typen in Ausprägungen von Mengentypen.
- Auswählen und Bereitstellen von Elementen der Datenbasis lässt sich als der identische Zustandsübergang erfassen.

Erweiterung unseres Beispiels:

Polymorphe Operatoren

```
union: menge × menge → menge
intersect: menge × menge → menge
insert: menge × datensatz → menge
deleteByKey: menge × atomarerTyp → menge
findByKey: menge × atomarerTyp → datensatz
```

SS 2004

B. König-Ries: Datenbanksysteme

33

Zusammenfassung Datenmodell

- Pragmatische Definition Datenmodell:
 - System aus atomaren Typen, Typkonstruktoren, polymorphen Konsistenzbedingungen und polymorphen Operatoren.

SS 2004

B. König-Ries: Datenbanksysteme

34

Kapitel 1: Datenbankdienste

Einführung: Dienste
Dienstfunktionalität
Dienstmerkmale
Datenbanksysteme

SS 2004

B. König-Ries: Datenbanksysteme

35

Bedeutungstreue in der Datenverwaltung



- ▶ Daten müssen interpretierbar sein (**Information**)
- ▶ Daten müssen bei allen am Austausch beteiligten Partnern dieselbe Information besitzen
- ▶ Interpretation muss über die Zeit dieselbe bleiben
- ▶ Konventionen zur Interpretation begleiten die Daten ⇒ Aufgabe des Datenverwaltungssystems

SS 2004

B. König-Ries: Datenbanksysteme

36

Gewinnung der Konventionen

- Interpretation durch Bindung an eine auf bestimmte Aspekte beschränkte Anwendungswelt, die **Miniwelt**.
 - Informationen sind somit gedankliche Abstraktionen (**Modelle**) der Miniwelt.
 - Daten sind daher Repräsentationen von Modellen.

 - **Eine Datenbasis ist bedeutungstreu, wenn ihre Elemente Modelle einer gegebenen Miniwelt repräsentieren. (Datenbasiskonsistenz)**
 - Datenbasis beschreibt Zustand der Miniwelt durch mathematische Strukturen (Mengen, Tupel, Funktionen, ...).
- ⇒ Datenbasis ist damit **formales Modell** der Miniwelt.

SS 2004

B. König-Ries: Datenbanksysteme

37

Beispiel Realwelt: Flugwesen

- Reale, anfassbare Gegenstände:
 - Flugzeuge mit Flugzeugtyp, Besatzung, Sitzplätzen, Laderäumen, Farbanstrichen;
 - Flughäfen mit Bezeichnung, Flugsteigen, Passagieraufkommen, Rollbahnen;
 - Passagiere mit Namen, Anschriften, Telefonnummern, Größe, Gewicht, Haarfarbe, Religionszugehörigkeit;
 - Flugscheine mit Vordrucken und Aufdrucken.
- Gedankliche, in ihren Auswirkungen beobachtbare Phänomene:
 - Flüge mit Abflugs- und Ankunftszeiten, Flugzeugtyp, Startgeschwindigkeit, Fluggeschwindigkeit, -höhe und -route;
 - Reservierungen mit Flugnummer und -datum, Sitzplatz, Passagier.

SS 2004

B. König-Ries: Datenbanksysteme

38

Miniwelt Platzbuchung

- Reale, anfassbare Gegenstände:
 - Flugzeuge mit Flugzeugtyp, Besatzung, Sitzplätzen, Laderäumen, Farbanstrichen;
 - Flughäfen mit Bezeichnung, Flugsteigen, Passagieraufkommen, Rollbahnen;
 - Passagiere mit Namen, Anschriften, Telefonnummern, Größe, Gewicht, Haarfarbe, Religionszugehörigkeit;
 - Flugscheine mit Vordrucken und Aufdrucken.
- Gedankliche, in ihren Auswirkungen beobachtbare Phänomene:
 - Flüge mit Abflugs- und Ankunftszeiten, Flugzeugtyp, Startgeschwindigkeit, Fluggeschwindigkeit, -höhe und -route;
 - Reservierungen mit Flugnummer und -datum, Sitzplatz, Passagier.

SS 2004

B. König-Ries: Datenbanksysteme

39

Datenbasisschemata (1)

- **Vorgehen**: Einsetzen konkreter (aus dem Modell der Miniwelt gewonnener) Typen und Bezeichner für die Typvariablen bzw. Selektoren in den Typkonstruktoren und polymorphen Konsistenzbedingungen.
- Das polymorphe Typsystem wird zu einem monomorphen Typsystem instantiiert.
- Erst jetzt ist die Datenbasis definierbar.
 - Erst jetzt liegt ein formales Modell der Miniwelt vor.
- **Datenbasistyp**: Ergebnis der Konkretisierung.
- **Datenbasisschema** (kurz: **Schema**): Beschreibung eines Datenbasistyps.

SS 2004

B. König-Ries: Datenbanksysteme

40

Beispiel

Zugrundeliegender polymorpher Typ
(Schablone für monomorphe Typen)

```
Datenbasistyp Buchungsdatabasi

Typ Flughafen ::= datensatz [
  FlughCode: string, Stadt: string, Land: string,
  Name: string, Zeitzone: int ]
Typ Flugzeugtyp ::= datensatz [
  FtypId: string, Name: string, First: int,
  Business: int, Economy: int ]
Typ Flug ::= datensatz [
  FlugNr: string, von: string, nach: string,
  FtypId: string, Wochentag: string,
  Abflugszeit: time, Ankunftszeit: time,
  Entfernung: int ]
Typ Flughäfen ::= menge {Flughafen} key FlughCode
Typ Flugzeugtypen ::= menge {Flugzeugtyp} key FtypId
Typ Flüge ::= menge {Flug} key FlugNr
```

monomorphe Konsistenzbedingung

SS 2004

B. König-Ries: Datenbanksysteme

41

Weitere denkbare Konsistenzbedingungen

Datenbasistyp Buchungsdatabasi

```
Typ Flughafen ::= datensatz [
  FlughCode: string, Stadt: string, Land: string,
  Name: string, Zeitzone: int ]
Typ Flugzeugtyp ::= datensatz [
  FtypId: string, Name: string, First: int,
  Business: int, Economy: int ]
Typ Flug ::= datensatz [
  FlugNr: string, von: string, nach: string,
  FtypId: string, Wochentag: string,
  Abflugszeit: time, Ankunftszeit: time,
  Entfernung: int ]
Typ Flughäfen ::= menge {Flughafen} key FlughCode
Typ Flugzeugtypen ::= menge {Flugzeugtyp} key FtypId
Typ Flüge ::= menge {Flug} key FlugNr
```

Land → Zeitzone

Wert unter von
kommt als Wert unter
FlughCode vor

(von,nach) → Entfernung

SS 2004

B. König-Ries: Datenbanksysteme

42

Beispiel

- Wie könnten Schema und Konsistenzbedingungen im Bibliotheksbeispiel aussehen?

Buch (Signatur, Titel, Autor, ...)

Benutzer (BNr, Name, ...)

Verwaltung (
 Ausleihen (Signatur, BNr, Datum)

SS 2004

B. König-Ries: Datenbanksysteme

43

Zustandsübergänge (1)

- Ein Datenverwaltungssystem gewährleistet (Schema-) Konsistenz, wenn seine Dienstfunktionen stets einen (schema-)konsistenten Zustand seiner Datenbasis wieder in einen (schema-)konsistenten Zustand überführen.
- Gehorcht die Datenbasis einer wie immer gearteten Schemakonsistenz vor Ausführen einer Dienstfunktion, so gehorcht sie ihr auch zum Abschluss der Ausführung.

SS 2004

B. König-Ries: Datenbanksysteme

44

Konsistente Zustände (1)

