

# Datenbanken / Datenbankmanagementsystem

## 1. Einführung

Daten, Informationen; Datenbank, Datenbanksystem;  
Relationale Datenbanksysteme; Beispiel Access

## 2. Aufbau von Datenbanken

Datenanalyse; Entitäten-Beziehungsmodell;  
Datendefinition; Datendefinition in Access;  
Normalformen; Strukturelle Integritätsbedingungen

## 3. Nutzung von Datenbanken

Datenmanipulation mit Hilfe von Datenblattansichten,  
Berichten, Formularen, Abfragen (QBE, SQL)

## 4. Weiterführende Themen

Mehrbenutzerbetrieb; objektorientierte Datenbanken;  
Erweiterungen des Entitäten-Beziehungsmodell

---

## 1. Einführung

### DATEN – INFORMATIONEN

**Daten** sind eine Repräsentation von Fakten, Konzepten oder auch Instruktionen in einer formalisierten Art und Weise, die sie für die Kommunikation, Interpretation und Verarbeitung durch **Maschinen** geeignet machen.

**Informationen** beziehen sich auf die Bedeutung, die **Menschen** durch vereinbarte Konventionen diesen Daten zuordnen.

Sie sind interpretierte Daten, bilden Grundlage für menschliche **Entscheidungen**, bringen **Nutzen**, besitzen infolgedessen einen monetären Wert

### TRANSIENTE DATEN – FLÜCHTIGE DATEN

Transiente Daten befinden sich auf einem flüchtigen Speichermedium eines Computers.

Ein Prozess (ein ablaufendes Programm) belegt Speicherplätze im Arbeitsspeicher, legt dort transiente Daten an und weist diesen Werte zu.

Die Speicherplätze werden spätestens dann freigegeben, wenn der Prozess endet. Die transienten Daten hören dann auf, zu existieren. Deshalb kann beim Neustart des Systems nicht auf die Werte der Daten früherer Programmabläufe zurückgegriffen werden.

### PERSISTENTE DATEN (DAUERHAFTEN DATEN)

Persistente Daten befinden sich auf einem nichtflüchtigen Speichermedium eines Computers. Wichtigstes Beispiel für persistente Daten: Dateien.

Persistente Daten können durch Prozesse erzeugt, verändert und gelöscht werden. Sie existieren weiter, wenn der Prozess sich beendet, der sie erzeugt hat.

Sie hören erst auf, zu existieren, wenn ein Prozess sie ausdrücklich löscht. Andere Prozesse können (prinzipiell) auf persistente Daten zugreifen. Auch beim Neustart eines Programms.

## WAS IST EINE DATENBANK?

### Erste Definition:

Datenbank = Verwaltungskomponente + Speicherkomponente für persistente Daten, die einem bestimmten Zweck dienen

**Frage:** Warum genügt hierfür nicht das Dateisystem eines Computers?

## WAS IST EIN DATENBANKSYSTEM?

Ein Datenbanksystem (auch Datenbankmanagementsystem) ist ein Softwaresystem, das den Aufbau und den Betrieb einer Datenbank unterstützt.

Beispiele für verbreitete Datenbanksysteme: Access, SQL Server, MySQL, DB2, Oracle usw.

Eine **Datenbank** mit dem Namen „Adressen“ dient zur Verwaltung von Postadressen und Telefonnummern.

Die Datenbank „Adressen“ wurde/wird mit Hilfe des **Datenbanksystems** „Access“ aufgebaut und betrieben.

**Abkürzungen:** DB = Datenbank, DBMS = Datenbank(management)system

## FUNKTIONEN EINES

## DATENBANKSYSTEMS

- Dauerhafte Speicherung von Daten
- Einrichten einer Datenbank mit Hilfe einer Datendefinitionssprache z.B. Einrichten einer Adressdatenbank
- Abfragen und Modifikation von Daten mit Hilfe einer Datenmanipulationssprache: Selektieren, Anzeigen, Erzeugen, Verändern, Verknüpfen, Löschen von Daten z.B.: Nachschlagen, Eintragen, Ändern, Löschen von Adressen in einer Adressdatenbank
- Interaktive Benutzungsoberfläche für Datendefinition und Datenmanipulation (ohne Programmierkenntnisse)
- Programmschnittstelle, ermöglicht Datendefinition und Datenmanipulation aus einem Programm heraus

## BESONDERE SYSTEMMERKMALE

## EINES DATENBANKSYSTEMS

- **Datenunabhängigkeit:** Das Datenbanksystem macht die Nutzer unabhängig von den computerinternen Speicherstrukturen zur Datenablage
- Daten können über genormte Programmschnittstelle von verschiedenen Programmen genutzt werden
- **Mehrbenutzerfähigkeit:** Störungsfreier paralleler Zugriff auf eine Datenbank durch mehrere Prozesse (auch über Netze) mit Hilfe eines so genannten Transaktionskonzepts
- **Konsistenzerhaltung** auch bei Systemabsturz  
Beispiel: Überweisung eines Betrags von einem Konto auf ein anderes Konto. Es darf nicht passieren, dass nur ein Konto verändert wird und das andere nicht.
- Fortlaufende **Datensicherung** und **Wiederanlauf** nach Systemabsturz

## WAS IST EIN INFORMATIONSSYSTEM?

ein Anwendungsprogramm, das die Bereitstellung von Informationen aus Daten und die Ablage von Informationen in Form von Daten ermöglicht

D.h. es unterstützt die Vorgänge der Interpretation von Daten und der Repräsentation von Informationen  
Kurz: eine Datenbank mit komfortabler Benutzungsoberfläche

## RELATIONALE DATENBANKSYSTEME (RDBMS)

### TABELLEN (RELATIONEN)

Relationale DBMS speichern Daten in **Tabellen** (auch **Relationen** genannt)  
Tabellen: intuitiv verständliche, einfach interpretierbare Art der Repräsentation von Informationen

Die **Zeilen** der Tabelle (auch **Datensätze** genannt) repräsentieren untereinander gleichartige Informationseinheiten

Datensätze sind gegliedert in **Felder** (auch **Merkmale** oder **Attribute** genannt).

Die **Spalten** der Tabelle enthalten gleichartige Felder der Datensätze. Sie sind mit den Namen der Felder überschrieben

In den Spalten stehen **Datenwerte** von gleichartigem **Datentyp** mit vordefiniertem **Wertebereich** (engl. domain)

### FELDER UND DATENWERTE

Felder besitzen einen **Felddatentyp**, die wichtigsten sind:

- Zahl
- Text mit/ohne festgelegte Maximallänge
- Datum/Uhrzeit

besitzen einen **Wertebereich** (engl.: domain), z.B. gültige Matrikelnr. oder Artikeln.

### PRIMÄRSCHLÜSSEL

**Primärschlüssel** = Feld oder Kombination von Feldern, deren Werte einen Datensatz eindeutig identifizieren

Beispiele:

Kundennummer, Matrikelnummer, Ortskennzahl, Flughafencode

Oft werden „**künstliche**“ Primärschlüssel, z.B. laufende Nummern, eingeführt, um Datensätze eindeutig identifizieren zu können.

### TABELLEN: ZUSAMMENFASSUNG

Tabelle = Menge von Datensätzen, die in Felder gegliedert sind, mit folgenden Eigenschaften:

- Die Tabelle hat einen eindeutigen Tabellennamen.
- Innerhalb der Tabelle ist jeder Feldnamen eindeutig und bezeichnet eine Spalte mit der betreffenden Eigenschaft.
- Die Reihenfolge der Spalten ist bedeutungslos.
- Die Reihenfolge der Datensätze ist bedeutungslos.
- Es gibt einen Primärschlüssel, d.h. eines der Felder oder eine Feldkombination identifiziert die Datensätze eindeutig.
- Es gibt keine Begrenzung für die Anzahl der Spalten und der Zeilen einer Tabelle (zumindest in der Theorie).

Beispiel DBMS: ACCESS

**ACCESS-DATENBANKEN** ENTHALTEN u.a. folgende OBJEKTE:

1. Tabellen
2. Abfragen
3. Formulare
4. Berichte

## 2. Aufbau von Datenbanken

Der Aufbau von Datenbanken erfolgt in vier Schritten:

1. **Datenanalyse**: Erfassung der zum Aufbau der Datenbank erforderlichen Sachverhalte in der Sprache der künftigen Anwender
2. **Entitäten-Beziehungsmodell**: Entwurf einer graphischen Repräsentation der darzustellenden Objekte („Entitäten“) und ihrer Beziehungen
3. Umsetzung des Entitäten-Beziehungsmodells durch **Definition** von geeigneten **Tabellen**
4. **Füllen** der Datenbank mit aktuellen **Inhalten**

### 1. DATENANALYSE

#### Datenanalyse:

Erfassung der zum Aufbau der Datenbank erforderlichen Sachverhalte in der Sprache der künftigen Anwender. Typischer Anwendungsfall: Der Datenbankentwickler führt ein Gespräch mit den künftigen Anwendern der Datenbank. Er erfragt, welche Informationen und Abfrageinstrumente in der Datenbank repräsentiert werden sollen (z.B. Personalverwaltung):

- Jede **Abteilung** besitzt eine Bezeichnung.
- Jeder **Mitarbeiter** ist charakterisiert durch einen Namen sowie durch Straße und Ort, in welchen er wohnt.
- Jeder **Arbeitsvertrag** legt eine Funktion (z.B. „Buchhalter“) eines Mitarbeiters fest sowie das Gehalt, das er verdient.
- Jedes **Projekt** besitzt einen Namen und eine eindeutige Projektnummer.
- Für jeden Mitarbeiter ist ein Arbeitsvertrag **gültig**, jeder Arbeitsvertrag ist genau für einen Mitarbeiter gültig.
- Jeder Mitarbeiter ist einer Abteilung **unterstellt**.
- Mitarbeiter können gleichzeitig an mehreren Projekten **teilnehmen**, wobei die jeweiligen Prozentanteile erfasst werden.

### 2. ENTITÄTEN-BEZIEHUNGSMODELL (*entity relationship model - ERM*)

- Graphischer Entwurf der Datenbank
- Alle wesentlichen Elemente werden graphisch dargestellt
- Umsetzung der Datenanalyse in ein erstes Design dient auch zur Kommunikation mit dem Kunden und zur Verifizierung der Datenanalyse (Analogie: Architektenplan)

### 3. DATENDEFINITION

Als nächster Schritt muss das Entitäten-Beziehungsmodell in Form von Tabellen umgesetzt werden. Dies ist der Vorgang der **Datendefinition**. Darunter verstehen wir die Festlegung des so genannten **Datenbankschemas**, das die **Struktur** einer Datenbank beschreibt:

- Namen der Tabellen und ihrer Felder
- allgemeine Eigenschaften zu den Tabellen und Feldern

Die **Inhalte** der Datenbank gehören nicht zur Datendefinition (s. 4. **Datenmanipulation**).

#### PRIMÄR- UND FREMDSCHLÜSSEL:

Ein Feld oder eine Feldkombination in einer Tabelle heißt **eindeutig**, wenn dadurch jeder Datensatz in der Tabelle eindeutig bestimmt ist.

- Eine eindeutige Feldkombination heißt **minimal**, wenn keines der Felder weggelassen werden kann, ohne die Eindeutigkeit aufzugeben.
- Einer der Schlüsselkandidaten einer Tabelle hat die Rolle des **Primärschlüssels**. Ein solcher dient dazu, die Datensätze in der Tabelle eindeutig zu bezeichnen.

#### DEFINITION VON BEZIEHUNGSEIGENSCHAFTEN

Access versteht unter „Beziehung“ jede Verknüpfung zwischen Tabellen über Schlüssel.

- Praktisch immer müssen sog. **1:n-Verknüpfungen** verwendet werden. Seltene Ausnahme: **1:1-Verknüpfung** zwischen Primärschlüsseln
- In Access gibt es keine „**n:m-Verknüpfungen**“. Solche Beziehungsmengen werden durch eine Zwischentabelle, ein „drittes Objekt“ (mit 1:n Beziehungen) dargestellt.

### Aufgabe

Entwickle für das nachfolgend beschriebene Beispiel einer CD-Sammlung ein Datenbankmodell. Erstelle zunächst anhand der Entwurfsansicht geeignete Tabellenschemata mit passenden Datentypen. Definiere sodann die Beziehungen zwischen den erstellten Tabellen und entwickle das entsprechende ERM:

- Jede CD besitzt einen eindeutigen Identifikator, einen Titel und ein Erscheinungsjahr
- Es gibt Urheber, die durch Namen, Vornamen, Heimatland genauer definiert sind
- CDs haben einen oder mehrere Urheber und umgekehrt.
- Die Urheberschaft kann durch eine Rolle (z.B. Komponist, Interpret) definiert sein.
- Jeder CD ist genau ein Verlag zugeordnet (mit Namen und Adressangaben)

## 3. Nutzung von Datenbanken

### DATENMANIPULATION

- das erstmalige **Füllen einer Datenbank mit Inhalten**.
- Das **Abrufen** von Daten aus der Datenbank
- Das **Ändern** (modification) von Daten in der Datenbank
- Das **Eintragen** (insertion) von Daten in die Datenbank
- Das **Löschen** (deletion) von Daten in der Datenbank

### BENUTZUNGSOBERFLÄCHEN ZUR DATENMANIPULATION

- **Datenblattansichten von Tabellen**: Editierbare Tabellen zum Anzeigen und Ändern von Tabelleninhalten
- **Formulare**: Anzeigen und Ändern von Tabelleninhalten
  - Die Inhalte zusammengehöriger Tabellen können in einem Formular mit Unterformular dargestellt werden
- **Berichte**: Ähnlich wie Formulare, aber nur zum Anzeigen
- **Abfragen**:
  - **Auswahlabfragen** zur (vorübergehenden) Erzeugung neuer Tabellen aus den vorhandenen durch Auswahl und Kombination nach bestimmten Kriterien
  - **Aktionsabfragen** zum Ändern, Eintragen und Löschen von Datensätzen nach bestimmten Kriterien

### ARTEN VON ABFRAGEN AM BEISPIEL ACCESS

- **Auswahlabfragen**: Daten aus ausgewählten Feldern einer oder mehrerer Tabellen werden nach bestimmten Kriterien selektiert. Das Ergebnis ist wieder eine Tabelle, die aber nur vorübergehende Existenz besitzt.
- **Aktionsabfragen**: Datenbankinhalte werden nach bestimmten Kriterien verändert. Wir unterscheiden:
  - **Aktualisierungsabfragen**: Ändern von Datenwerten vorhandener Datensätze
  - **Löschabfragen**: Löschen von Datensätzen
  - **Anfügeabfragen**: Anfügen von neuen Datensätzen an vorhandene Tabellen
  - **Tabellenerstellungsabfragen**: Wie Auswahlabfragen, aber Ergebnis wird als dauerhafte Tabelle gespeichert

## 4. Weiterführende Themen

### OBJEKTORIENTIERTE DATENBANKSYSTEME (OODBMS)

OODBMS besitzen ein objektorientiertes Datenmodell (kein relationales).

Das zugrunde liegende Konzept ist das **Objekt** (anstelle der *Tabelle* in relationalen DBMS):

- Objekte gehören einer **Klasse** an (z.B. Buch).
- Objekte haben eine eindeutige **Identität** (z.B. Inventar-Nr.).
- Objekte besitzen **Merkmale** und Merkmalswerte, diese können auch komplex sein (z.B. Menge anderer Objekte).
- **Aggregation**: Objekte können sich aus anderen Objekten zusammensetzen (z.B. ein Buch besteht aus Kapiteln).
- **Generalisierung**: Klassen können eine Hierarchie aus Ober- und Unterklassen bilden (z.B. Publikation ist eine Oberklasse von Buch und Zeitschrift).
- Objekte zeigen „Verhalten“, d.h. sie besitzen **Methoden** (z.B. „Entleihen“, „Zurückgeben“).

### BEISPIEL FÜR EIN OBJEKT

Buch „Kant und das Problem der Einheit von theoretischen und praktischer Vernunft“:

- **Klasse**: Buch
- **Identität**: eine Nummer
- **Merkmale** (die Merkmalswerte können Zahlen, Texte, Objekte oder komplexe Merkmalswerte, d.h. Mengen sein):
  - Jahr
  - Titel
  - Verlag
  - Objekt
  - Text
  - Zahl
  - Autoren
- **Methoden** (Programme, die auf Anforderung ausgeführt werden, und das Objekt zum Gegenstand haben):
  - Entleihen
  - Zurückgeben

### VORTEILE OBJEKTORIENTIERTER DATENBANKSYSTEME

- Das Konzept eines Objekts ist anschaulicher als das einer Tabelle. Wir sind eher gewohnt, in Objekten zu denken. In einem Objekt bilden Zustand (Merkmale) und Verhalten (Methoden) eine Einheit. In relationalen Datenbanken müssen Objekte hingegen auf unterschiedliche Datensätze verteilt werden.
- Objektmerkmale können auch komplex sein. Dies wird im Bibliotheksbereich oft gewünscht (z.B. Autorenliste).
- Objekte können auf der Benutzungsoberfläche als Bildschirmobjekte (Fenster, Symbole) dargestellt werden. Die Merkmale der Objekte lassen sich so visualisieren, die Methoden können durch Zeigeoperationen (Anklicken, Menüauswahl, Ziehen) mit der Maus ausgelöst werden.
- OODBMS lassen sich gut mit objektorientierten Programmiersprachen (z.B. Java, C++) koppeln.