



Datenbanken

Wiederholung: Grundlagen der semantischen Datenmodelle
Sprachgebrauch in
Programmierung und Datenbankentwicklung

Dipl.-Inf. Robert Hartmann

Fachbereich Informatik (Sankt Augustin)

Mai 2024, KW 19

Datenbanken dienen der effizienten und flexiblen Verwaltung und Nutzung strukturierter Daten.

- **OLTP (Online Transaction Processing):**

- Optimiert für den täglichen Betrieb von Geschäftsprozessen.
- Optimiert auf Einhaltung der ACID-Eigenschaften (Atomarität, Konsistenz, Isolation, Dauerhaftigkeit)
 - Verarbeitet schnelle, einzelne Transaktionen.
(Eine Transaktion ist eine ununterbrechbare Folge von Anweisungen.)
 - Gewährleistet Integrität und Konsistenz der Daten.

- **OLAP (Online Analytical Processing):**

- Optimiert für die Analyse und Berichterstellung.
- Unterstützt komplexe Abfragen und Analysen.
- Bietet schnelle Antwortzeiten auf komplexe Abfragen.

Unterstützung von Geschäftsprozessen

- **Verkaufs- und Inventarmanagement:** Relationale Datenbanken werden verwendet, um Verkaufs-, Bestands- und Lieferinformationen zu verwalten. Sie ermöglichen schnelle, einzelne Transaktionen, die für die Echtzeit-Entscheidungsfindung und Bestellungen notwendig sind.
- **Kundenbeziehungsmanagement (CRM):** CRM-Systeme, die auf relationalen Datenbanken basieren, verarbeiten Transaktionen wie Kundendatenaktualisierungen, Interaktionen und Verkaufschancen. Diese Systeme sind darauf ausgelegt, schnelle, einfache Transaktionen zu verarbeiten, die für die Verbesserung der Kundenbeziehungen und die Steigerung der Verkaufszahlen entscheidend sind.
- **Personalverwaltung:** Relationale Datenbanken unterstützen die Verwaltung von Mitarbeiterdaten, Gehaltsinformationen und Urlaubsanfragen. Sie ermöglichen schnelle, einfache Transaktionen, die für die effiziente Verwaltung und Analyse von Personalressourcen notwendig sind.

Unterstützung der Forschung

- **Forschungsdatenmanagement:** Relationale Datenbanken werden für die Speicherung und Verwaltung strukturierter Daten wie Experimentdaten, Patienteninformationen oder Literaturdaten verwendet. Sie ermöglichen eine effiziente Verwaltung und Analyse dieser Daten, indem sie die Datenintegrität und -konsistenz gewährleisten.
- **Bioinformatik:** In der Bioinformatik werden relationale Datenbanken eingesetzt, um komplexe Analysen auf Genomdaten, Proteinsequenzen und anderen biologischen Daten durchzuführen. Sie unterstützen Forschende bei der Identifizierung von Genen und Proteinen sowie bei der Untersuchung von Krankheitsmechanismen durch die Analyse großer Datenmengen.
- **Klimawissenschaften:** Relationale Datenbanken werden in der Klimawissenschaft verwendet, um Klimadaten, Wetterdaten und Umweltinformationen zu analysieren. Sie ermöglichen die Durchführung komplexer Abfragen und Analysen, um Trends und Zusammenhänge in Klimadaten zu identifizieren.

- **Objekte bzw. Entitäten:**

- Individuell identifizierbare - materielle oder immaterielle - Dinge aus Realität oder Fiktion: wie Gegenstände, Personen, Tiere, Patente, Geldkonten.
- In der objektorientierten Programmierung (OOP): Instanzen einer Klasse mit datenbefüllten Attributen und Methoden.
- In Datenbanken (DB): Eine Zeile einer Tabelle (oder Zeile mehrere normalisierter zusammengehöriger Tabellen) mit Daten.

- **Objektmengen bzw. Entitätsmenge:**

- Sammlungen von Objekten/Entitäten.
- Beispiel (OOP): Array mit Instanzen der Klasse Person.
- Beispiel (DB): Tabelle mit Daten über Personen

- **Beziehungen zwischen Objekten:**

- **Assoziation** ist eine

Kommunikations-/Verwendungs-Beziehung:

Es kommuniziert ein Objekt mit anderen Objekten, um die Funktionalität und Dienstleistungen dieser Objekte zu nutzen.

OOP	Methode eines Objektes nimmt als Parameter
Erinnerung	Referenzen zu anderen Objekten entgegen.

- **Aggregation** ist eine unidirektionale *Eigentums-Beziehung*:

Eigentumsverhältnisse sind änderbar.

Besitzbare Objekte können ohne Eigentümer-Objekt existieren.

OOP	Objekt besitzt instanzgebundene <i>Referenzvariablen</i>
Erinnerung	zum Zugriff auf ggf. existente besitzbare Objekte.

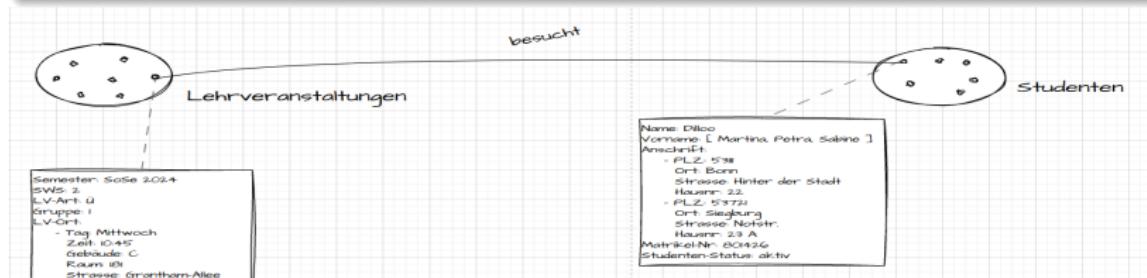
- **Komposition** ist eine *Teil-von-Beziehung*:

Komposition ist eine stärkere Form der Aggregation, bei der beteiligte Objekte existenziell eng miteinander verbunden sind und nicht ohne das Kompositions-Objekt existieren können.

OOP	Der Konstruktor des Kompositions-Objekts erzeugt
Erinnerung	intern zu kombinierende Objekte und bindet diese an <i>Referenzvariablen</i> innerhalb des Kompositions-Objekts.

Graphik-Elemente des Objekt-Datenmengen-Beziehungsdiagramms

- Ein Punkt
 - stellt Objekt bzw. Entität dar.
 - kann gestrichelt verbunden werden mit Rechteck. Rechteck enthält textuelle strukturierte Beschreibung der Entität.
- Eine Ellipse repräsentiert eine Entitätsmenge, die exemplarische Entitäten enthält.
- Eine Kante, die als geschlossene Linie zwei Punkte verbindet, repräsentiert eine Beziehung zwischen Entitäten.
 - Eine Beziehung zwischen mehreren Entitäten wird dargestellt durch eine **Hyper-Kante**, die mehrere Punkte verbindet.
 - Eine reflexive Beziehung wird durch **Schlinge** dargestellt, die einen Punkt mit sich selbst verbindet.
 - Beziehungen können auch mit Beschreibungsrechtecken verbunden sein.



- **Objekt- und Beziehungstypen:**

- Abstrakte Beschreibungen der Mengen von Objekten/Entitäten und deren Beziehungen.
 - Vererbung (Generalisierung, Spezialisierung)
ist eine besondere Beziehung zwischen Objekt-/Entitäts-Typen
- Beispiel (OOP): Klassen in Java, Klassendiagramm in UML2.
Aktivitäts- und Sequenzdiagramm in UML2.
- Beispiel (DB) : Tabellendefinition mit SQL/DDL,
(Enhanced) Entity-Relationship-Diagramm

Erfassung zur dynamischen Verwendung
von Objekten und ihrem Verhalten

ist Betrachtungsgegenstand der Algorithmitik, Programmierung und
des Software Engineerings.

Erfassung zur Speicherung und Suche
von Objekten mit ihren Beziehungen

ist Betrachtungsgegenstand der Datenbank-Veranstaltungen.