

Datenbankentwurf

- Bisher haben wir mit “fertigen” Datenbanken gearbeitet
 - Tabellen und zugehörige Datenfelder waren vorgegeben
 - Wir haben Daten daraus abgefragt und ggf. ergänzt
- Warum waren die Tabellen so aufgebaut, wie sie es waren?

- Die Beispieltabellen auf den folgenden Folien wurden aus Wikipedia, Artikel “Normalisierung (Datenbank)” entnommen

- Was ist von dieser Tabelle zu halten?

CD_Lied				
CD_ID	Album	Gründungsjahr	Erscheinungsjahr	Titelliste
4711	Anastacia – Not That Kind	1999	2000	{1. Not That Kind, 2. I'm Outta Love, 3. Cowboys & Kisses}
4712	Pink Floyd – Wish You Were Here	1965	1975	{1. Shine On You Crazy Diamond}
4713	Anastacia – Freak of Nature	1999	2001	{1. Paid my Dues}

(Die blau hinterlegten Spalten bilden den Primärschlüssel der Tabelle)

- Die Inhalte der rot hinterlegten Attribute sind nicht **atomar**
 - Ein Attribut ist dann atomar, wenn es sich nicht sinnvoll in weitere Teilattribute zerlegen lässt
- Probleme durch nicht-atomare Attribute:
 - Wie sortiert man die Tabelle nach dem Albumtitel?
 - Wie kann man nur einzelne Titel suchen/ausgeben?
- Daher:
 - Attributwerte müssen atomar sein!

- Was ist von der so angepassten Tabelle zu halten?

CD_Lied					
CD_ID	Albumtitel	Interpret	Gründungsjahr	Track	Titel
4711	Not That Kind	Anastacia	1999	1	Not That Kind
4711	Not That Kind	Anastacia	1999	2	I'm Outta Love
4711	Not That Kind	Anastacia	1999	3	Cowboys & Kisses
4712	Wish You Were Here	Pink Floyd	1965	1	Shine On You Crazy Diamond
4713	Freak of Nature	Anastacia	1999	1	Paid my Dues

- In der Tabelle auf der letzten Folie kamen etliche Attributwerte mehrfach vor, insbesondere:
 - “Anastacia” im Datenfeld “Interpret”
 - “Not that Kind” im Datenfeld “Albumtitel”
 - “4711” im Datenfeld “CD_ID”
- Man bezeichnet diese “Mehrfachvorkommen” als **Redundanz**
 - Redundanz in Datenbanken will man in der Regel vermeiden

Warum keine Redundanz?

- In vielen Bereichen ist Redundanz positiv und gewollt (Ersatzfallschirm, Reaktorkühlung, ...)
 - Warum nicht hier?
- Redundanz innerhalb einer Datenbank verkompliziert Einfügen, Ändern, Löschen von Daten
 - Es besteht die Gefahr von **Inkonsistenz**

Einfügeanomalie Ein Datensatz kann nicht eingefügt werden, weil (noch) nicht für alle Datenfelder Werte vorliegen

...

Beispiel: Wie nimmt man eine Künstlerin in die Beispieltabelle auf, die noch kein Album veröffentlicht hat?

Änderungsanomalie Nicht alle Vorkommen eines Attributwerts werden gleichzeitig geändert

...

Beispiel: Was passiert, wenn eine Band ihren Namen ändert und man ein Vorkommen übersieht/sich vertippt?

Löschanomalie Beim Löschen eines Datensatzes gehen Informationen verloren, die eigentlich erhalten bleiben sollen

...

Beispiel: Werden alle Lieder eines Künstlers gelöscht, so

Fallbeispiel - Erste Optimierung

- Zur Vermeidung von Redundanz werden folgende Änderungen durchgeführt:
 - Streiche Albumtitel, Interpret und Gründungsjahr aus der ursprünglichen Tabelle
 - Erstelle eine neue Tabelle "CD" mit diesen drei Attributen und dem Teil des ursprünglichen Primärschlüssels, von dem sie abhängen (CD_ID)
- Ergebnis:

CD				Lied		
CD_ID	Albumtitel	Interpret	Gründungsjahr	CD_ID	Track	Titel
4711	Not That Kind	Anastacia	1999	4711	1	Not That Kind
4712	Wish You Were Here	Pink Floyd	1965	4711	2	I'm Outta Love
4713	Freak of Nature	Anastacia	1999	4711	3	Cowboys & Kisses
				4712	1	Shine On You Crazy Diamond
				4713	1	Paid my Dues

Fallbeispiel - Alles gut?

- Was ist mit dieser Tabelle?

CD			
CD_ID	Albumtitel	Interpret	Gründungsjahr
4711	Not That Kind	Anastacia	1999
4712	Wish You Were Here	Pink Floyd	1965
4713	Freak of Nature	Anastacia	1999

- Das “Gründungsjahr” hängt vom Interpreten ab und ist für einen bestimmten Interpreten immer gleich
 - Wieder: Redundante Speicherung von Daten

Fallbeispiel - Zweite Optimierung

- Zur Vermeidung dieser weiteren Redundanz werden folgende Änderungen durchgeführt:
 - Streiche Interpret und Gründungsjahr aus der ursprünglichen Tabelle
 - Erstelle eine neue Tabelle “Künstler” mit diesen Attributen
 - Ergänze in der Tabelle “CD” einen Fremdschlüssel zum Verweis auf den jeweiligen Künstler
- Ergebnis:

CD			Künstler		
<i>CD_ID</i>	Albumtitel	Interpret_ID	<i>Interpret_ID</i>	Interpret	Gründungsjahr
4711	Not That Kind	311	311	Anastacia	1999
4712	Wish You Were Here	312	312	Pink Floyd	1965
4713	Freak of Nature	311			

- Die so erhaltene Aufteilung bezeichnet man als **Dritte Normalform**, kurz **3 NF**

- Bei sorgfältiger Modellierung erhält man oft schon eine (weitestgehend) normalisierte Datenbank
- Alternativer Ansatz:
 - Starte mit einer großen Tabelle
 - Identifiziere Redundanzen und Abhängigkeiten
 - Zerlege die Tabelle wie im Fallbeispiel schrittweise in weitere Tabellen

- Neben der Darstellung der Datenbankstruktur als Relationenschema hat sich die grafische Darstellung mittels **ER-Diagrammen** (Entity-Relationship-Diagramme) etabliert:
 - Tabellen als Rechtecke
 - Attribute in Ellipsen
 - Beziehungen durch Linien und Rauten
 - Primärschlüssel durch Unterstreichen
 - Kardinalitäten als Zahlen an den Verbindungslinien

Beispiel für ein ER-Diagramm

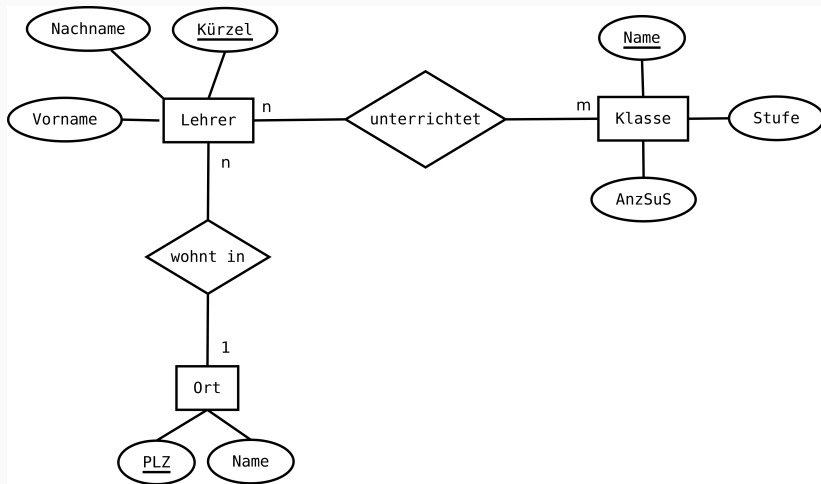


Abbildung 1: Beispiel für ein ER-Diagramm

- Auch eine Darstellung mittels UML ist möglich
 - Jede Tabelle wird in Form eines Klassendiagramms dargestellt
 - Tabellenname statt Klassenname
 - Datenfelder statt Attributen
 - Zugriffsrechte spielen keine Rolle
 - Beziehungen werden als Assoziation dargestellt
 - Benennung durch Text an der Verbindungslinie
 - Kardinalitäten wie bei ER-Diagrammen

Beispiel für ein UML-Diagramm

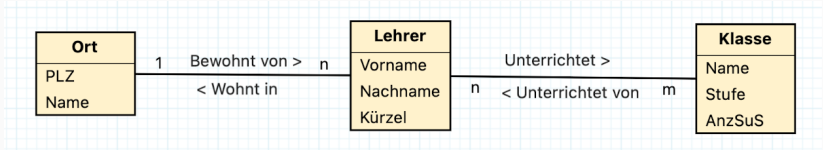


Abbildung 2: Beispiel für ein UML-Diagramm

- Bearbeite die Aufgabe B1.1 aus dem Abitur 2019