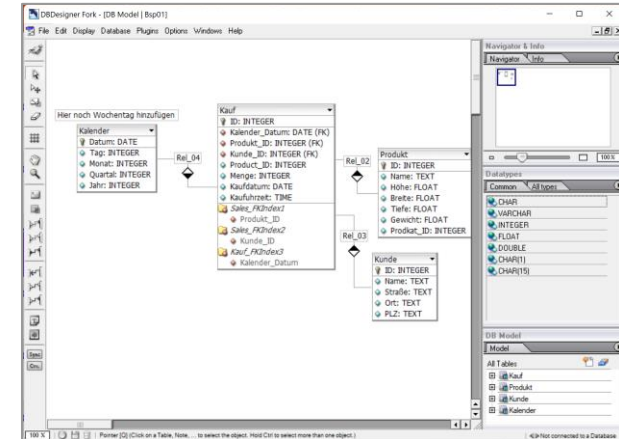
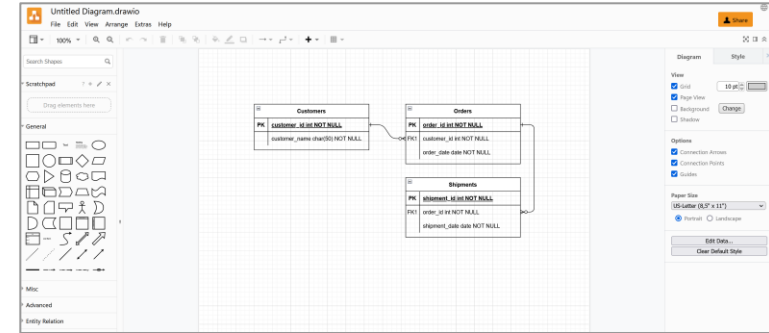


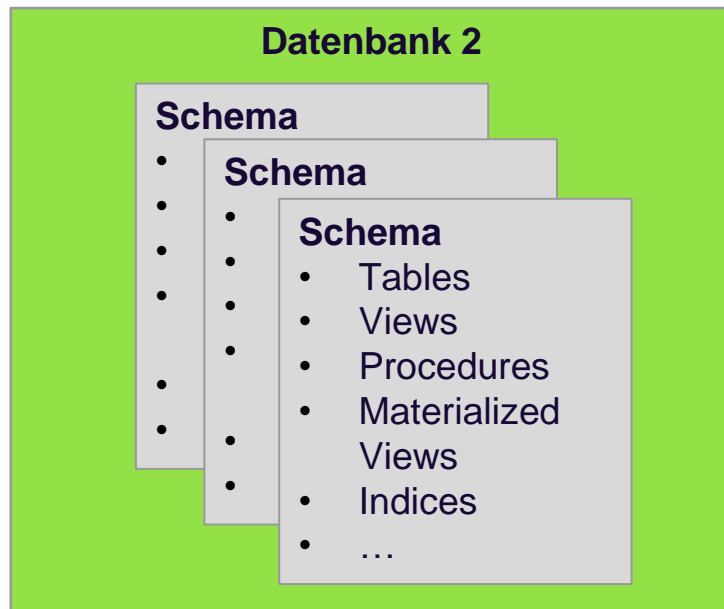
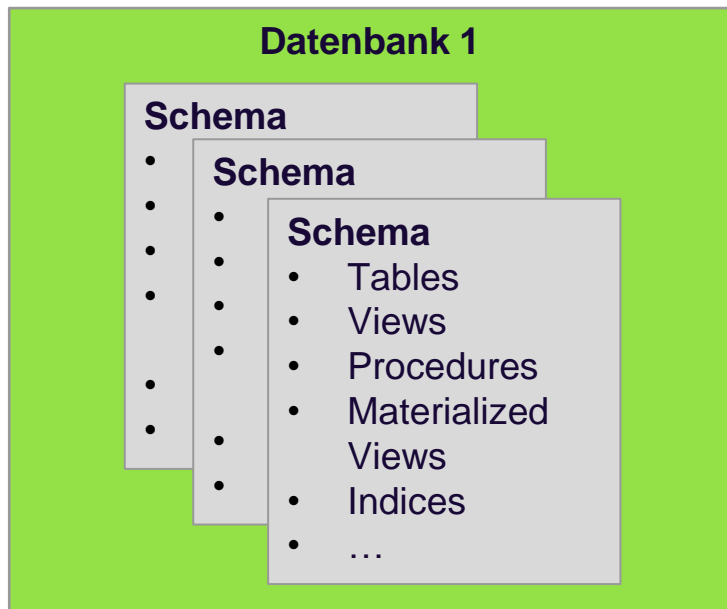
# 9

# Datenbank- Modellierung

- Es gibt viele Programme, um ERDs zu erstellen. Die meisten sind nicht kostenlos bzw. in der freien Version stark eingeschränkt
- Draw.io ist ein mächtiges Visualisierungstools, das viele Diagrammarten beherrscht. Es gibt eine Web- und eine Desktopversion.
- DBDesignerFork ist etwas veraltet, dafür aber einfach zu bedienen und es kann SQL erzeugen



Der PostgreSQL-Server verwaltet mehrere Datenbanken. Diese enthalten Schemata, welche Tabellen und weitere Objekte enthalten.



- **hohe Datenkonsistenz:** Die Daten sind korrekt
  - Der Primärschlüssel ist eindeutig
  - Fremdschlüssel sind leer oder existieren
  - Wertebereiche werden eingehalten (z.B. Alter)
  - Logische Konsistenz (z.B. Kinder müssen nach den Eltern geboren sein)
- **Redundanzfreiheit:** Die Daten sollten in der Regel nicht mehrfach gespeichert werden. Ausnahmen kann es aus Performance-Gründen geben, um z.B. JOINS zu vermeiden.

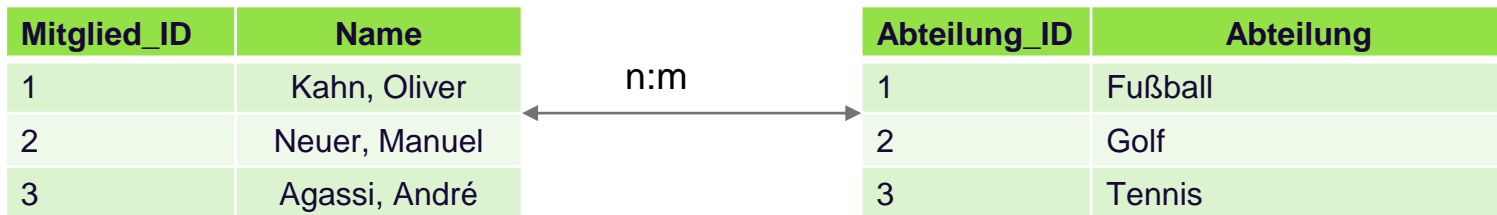
## Kardinalitäten

- **1:1** – Die Tabellen stehen in einer 1:1-Beziehung. Für jede Zeile in der einen Tabelle gibt es höchstens eine Zeile in der anderen Tabelle. Z.B. Person und Personendetails
- **1:n** – Zu eine Zeile der einen Tabelle gibt es mehrere Zeilen in der anderen Tabelle, aber umgekehrt nur eine. Z.B. Person und Geburtsort
- **n:m** – Es werden beliebig viele Zeilen der einen Tabelle der anderen Tabelle zugeordnet und umgekehrt. Zum Beispiel kann jeder Aktionär Aktion von beliebig vielen Unternehmen haben. Umgekehrt hat ein Unternehmen auch beliebig viele Aktionäre

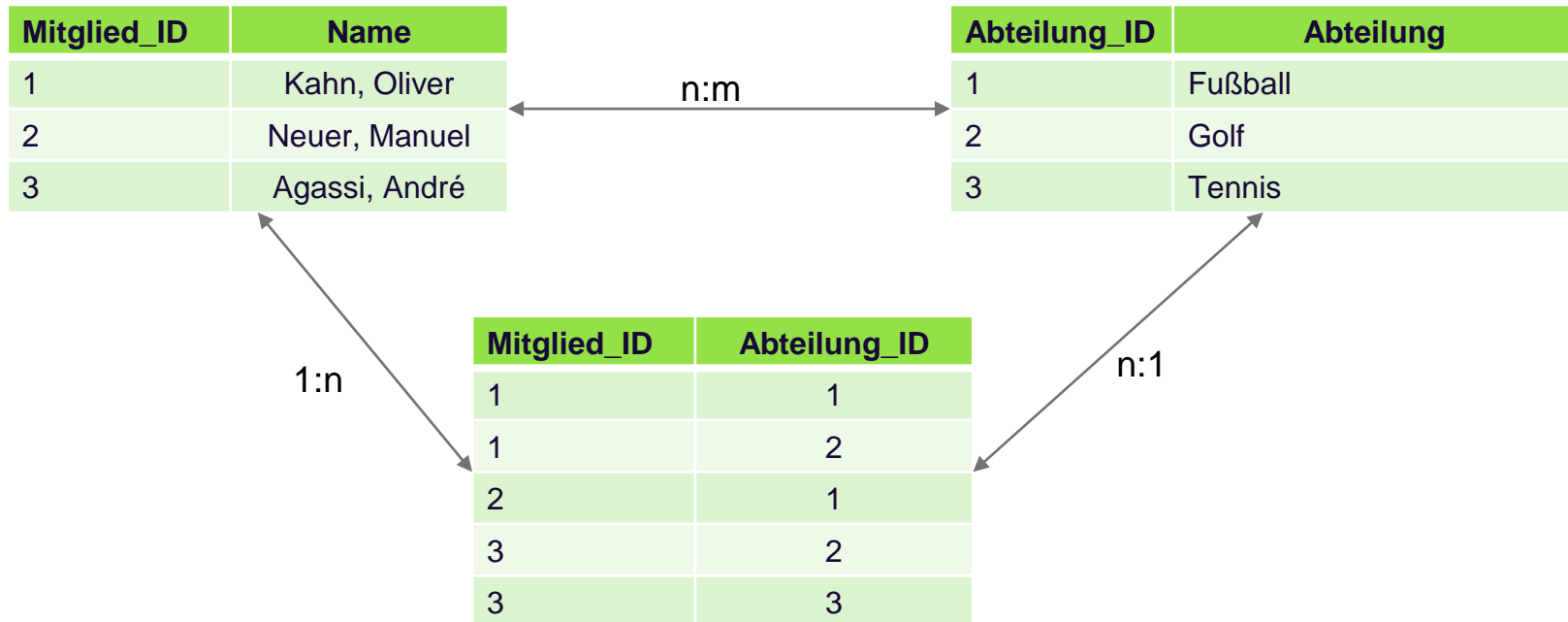
n:m-Beziehungen sind problematisch. Fasst man mehrere Elemente in einer Zelle zusammen, dann ist Sortieren nicht und Suche nur aufwändig möglich oder man muss die Daten redundant speichern.

Mitglied_ID	Name	Abteilung
1	Kahn, Oliver	Fußball, Golf
2	Neuer, Manuel	Fußball
3	Agassi, André	Golf, Tennis

hier kann nicht (einfach) nach Abteilung sortiert oder filtern



Eine n:m – Beziehung kann man auflösen, indem man eine Zwischentabelle (Linktabelle) einführt

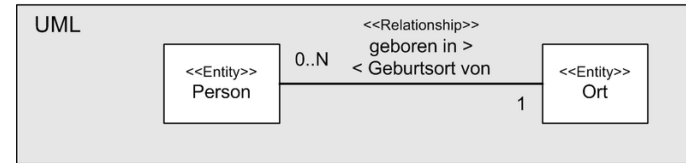
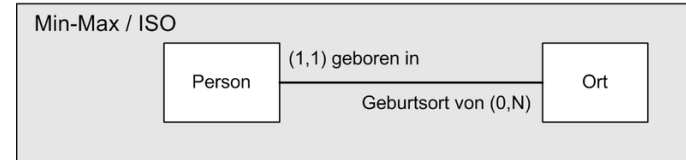
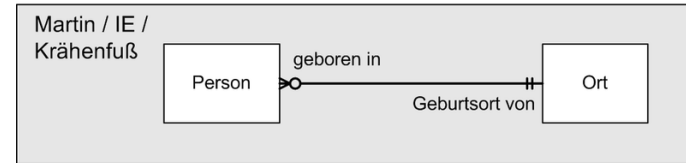
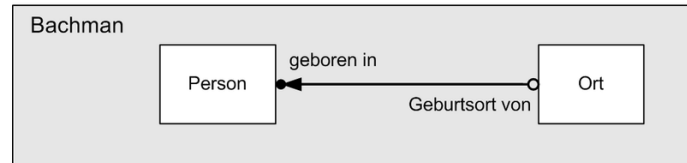
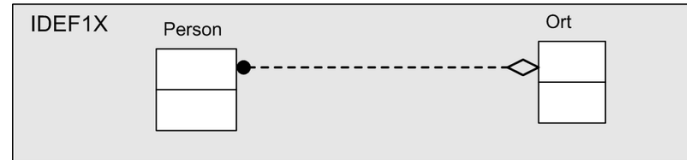
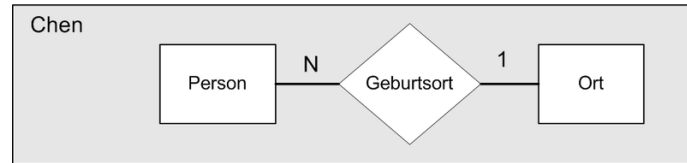


Ein **Schlüssel** dient dazu, eine Zeile einer Tabelle eindeutig zu identifizieren. Ein Schlüssel kann mehrere Spalten umfassen.

- **Primärschlüssel:** Der Hauptschlüssel, der in anderen Tabellen als Fremdschlüssel verwendet wird. Er sollte möglichst nur eine Spalte umfassen und einen simplen Datentyp haben.
- **Fremdschlüssel:** ist ein Attribut (Spalte), welches auf eine andere Relation (Tabelle) verweist



Je nach Programm bzw. Notation werden die Beziehungen anders dargestellt.  
Inhaltlich ist es aber kein Unterschied



Unter **Normalisierung eines relationalen Datenschemas** (Tabellenstruktur) versteht man die Aufteilung von Attributen (Tabellenspalten) in mehrere Relationen (Tabellen) gemäß gewisser Regeln, so dass keine Redundanzen mehr enthalten sind.

Es gibt mehrere Normalformen, wobei nur wenige in der Praxis relevant sind

## 1. Normalform

Jede Tabellenspalte muss einen atomaren Wertebereich haben (z.B. Adresse in Straße, Hausnummer, ... aufgeteilt) und die Tabelle muss frei von Wiederholungsgruppen sein. Letzteres heißt, dass Wiederholungen in andere Tabellen ausgelagert werden (z.B. Ort)

## Beispiel: 1. Normalform

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}



CD_ID	Albumtitel	Interpret	Gründungsjahr	Erscheinungsjahr	Track	Titel
4711	Not That Kind	Anastacia	1999	2000	1	Not That Kind
4711	Not That Kind	Anastacia	1999	2000	2	I'm Outta Love
4711	Not That Kind	Anastacia	1999	2000	3	Cowboys & Kisses
4712	Wish You Were Here	Pink Floyd	1965	1975	1	Shine On You Crazy Diamond
4713	Freak of Nature	Anastacia	1999	2001	1	Paid my Dues

## 2. Normalform

Erste Normalform und kein Nichtprimärattribut (d.h. nicht Teil des Schlüssels) hängt nicht nur von einem Teil des Schlüssels ab

CD_ID	Albumtitel	Interpret	Gründungsjahr	Erscheinungsjahr	Track	Titel
4711	Not That Kind	Anastacia	1999	2000	1	Not That Kind
4711	Not That Kind	Anastacia	1999	2000	2	I'm Outta Love
4711	Not That Kind	Anastacia	1999	2000	3	Cowboys & Kisses
4712	Wish You Were Here	Pink Floyd	1965	1975	1	Shine On You Crazy Diamond
4713	Freak of Nature	Anastacia	1999	2001	1	Paid my Dues



CD_ID	Albumtitel	Interpret	Gründungsjahr	Erscheinungsjahr
4711	Not That Kind	Anastacia	1999	2000
4712	Wish You Were Here	Pink Floyd	1965	1975
4713	Freak of Nature	Anastacia	1999	2001

CD_ID	Track	Titel
4711	1	Not That Kind
4711	2	I'm Outta Love
4711	3	Cowboys & Kisses
4712	1	Shine On You Crazy Diamond
4713	1	Paid my Dues

## 3. Normalform

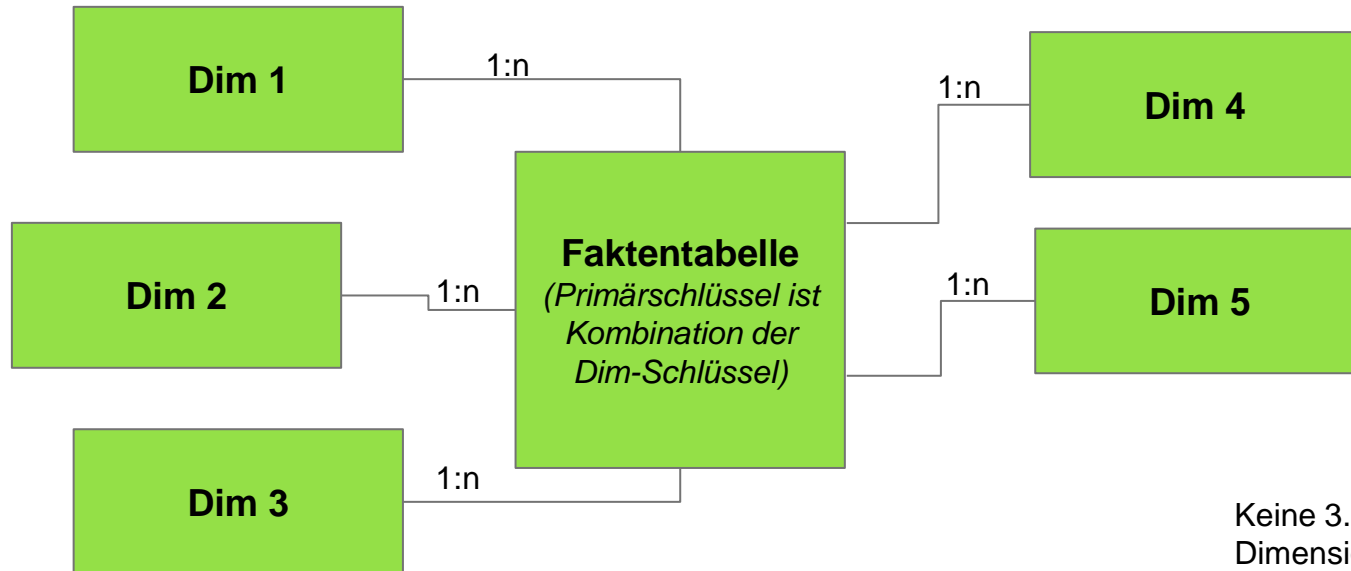
Zweite Normalform und kein Nichtschlüssel hängt von anderem Nichtschlüssel ab

CD_ID	Albumtitel	Interpret	Gründungsjahr	Erscheinungsjahr	CD_ID	Track	Titel
4711	Not That Kind	Anastacia	1999	2000	4711	1	Not That Kind
4712	Wish You Were Here	Pink Floyd	1965	1975	4711	2	I'm Outta Love
4713	Freak of Nature	Anastacia	1999	2001	4711	3	Cowboys & Kisses
					4712	1	Shine On You Crazy Diamond
					4713	1	Paid my Dues



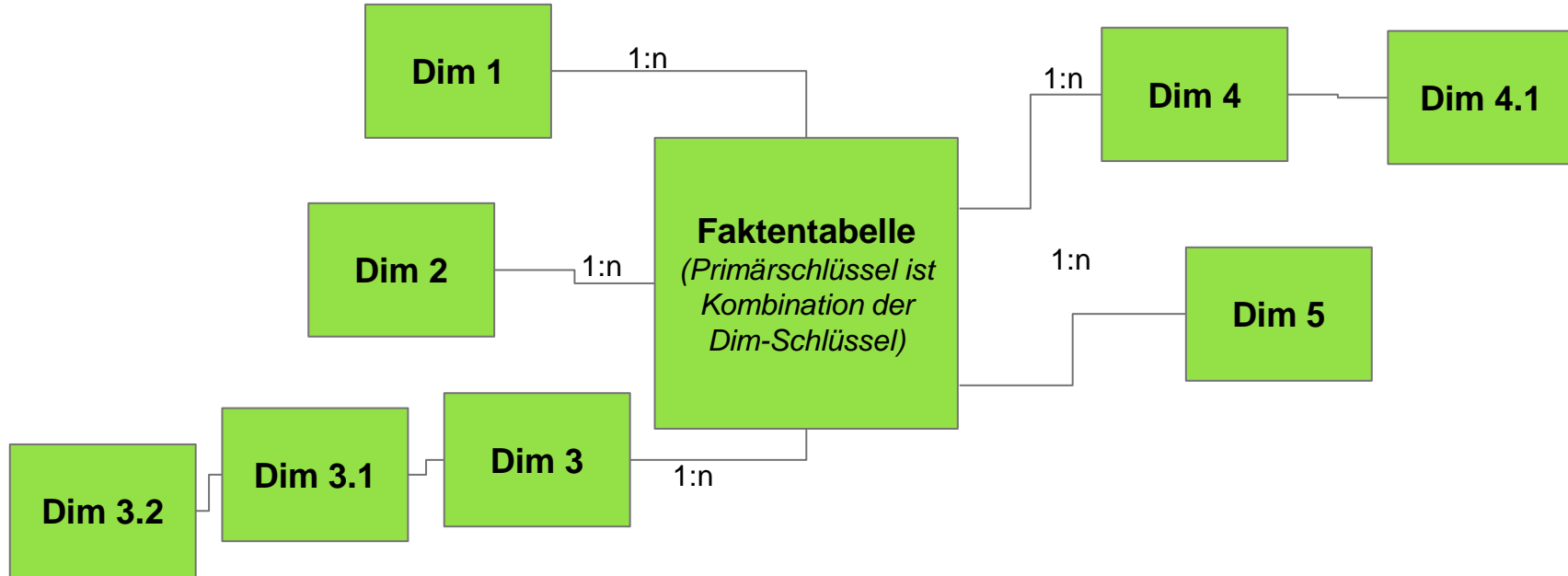
CD_ID	Albumtitel	Interpret	Erscheinungsjahr	Interpret	Gründungsjahr	CD_ID	Track	Titel
4711	Not That Kind	Anastacia	2000	Anastacia	1999	4711	1	Not That Kind
4712	Wish You Were Here	Pink Floyd	1975	Pink Floyd	1965	4711	2	I'm Outta Love
4713	Freak of Nature	Anastacia	2001			4711	3	Cowboys & Kisses
						4712	1	Shine On You Crazy Diamond
						4713	1	Paid my Dues

Das **Sternschema** ist ein Datenmodell, dessen Ziel nicht die Normalisierung, sondern die Optimierung auf effiziente Leseoperationen ist. Effiziente Leseoperationen sind für Datenanalyse (Data Warehouse, OLAP)



Keine 3. Normalform, da Dimensionstabellen nicht aufgetrennt

Das **Schneeflockenschema** erweitert das Sternschema, indem die Dimensionstabellen aufgeteilt werden, so dass die 3. Normalform vorliegt



Ein **Datenbankindex** ist eine von der Datenstruktur getrennte Indexstruktur, die die Suche und das Sortieren nach bestimmten Feldern beschleunigt. Z.B. wird das Jahr eines Datums als Index gespeichert, da häufig nur ein Jahr selektiert wird.

- **Clustered Index:** Das DBMS versucht, neue Datensätze physisch im Speicher anhand des Index abzulegen

Indizes verlangsamen das Einfügen von Daten, können aber massiv die Abfrage beschleunigen.