

# RDBMS

Relationale  
Datenbankmanagement-Systeme

# AGENDA

## 1. Das relationale Datenbankmodell

- 1.1 Tabellen, Attribute und Tupel
- 1.2 Beziehungen zwischen Tabellen
- 1.3 Primärschlüssel
- 1.4 Fremdschlüssel

## 2. Normalformen

## 3. Beispiele bekannte relationale Datenbank-Systeme

# 1. Das relationale Modell

Tabellen, Attribute und Schlüssel

# Das relationale Modell

**Aufbau einer relationalen Datenbank beruht auf relationalem Modell**

Eine relationale Datenbank ist eine Sammlung von:

- **Tabellen**, in denen sich die eigentlichen Daten befinden
- **Beziehungen** zwischen den Tabellen, die hergestellt werden können über:
  - Primärschlüssel und Fremdschlüssel

# Das relationale Modell - Tabellen

Alle Daten werden **in Tabellen** gespeichert

- Eine Spalte der Tabelle wird als **Attribut** bezeichnet
- Eine Zeile der Tabelle wird als Tupel oder **Datensatz** bezeichnet

Personaldaten können z.B. in einer Tabelle Person gespeichert werden:

**Tabelle Person**

Vorname	Nachname	Geburtsdatum	Geschlecht
Peter	Kiel	01.01.1980	maennlich
Lisa	Lebe	02.02.1970	weiblich
...	..	..	..

**Tupel** → (points to the row containing Lisa)

→ (points to the columns Vorname, Nachname, Geburtsdatum, and Geschlecht)

**Attribute**

# Das relationale Modell - Datentypen

Jedem Attribut wird bei Anlegen einer Tabelle **ein Datentyp** zugeordnet:

- In der Regel werden folgende Datentypen vom DBS unterstützt
  - Numerische Datentypen – Ganzzahl, Gleitkommazahl, Datum
  - Alphanumerische Datentypen - Character, Zeichenkette
  - Binäre Datentypen - Bitfolge fester/variabler Länge
- Unterstützte Datentypen abhängig vom Datenbanksystem

Zeichenkette		Datum	Zeichenkette
Vorname	Nachname	Geburtsdatum	Geschlecht
Peter	Kiel	01.01.1980	maennlich
Lisa	Lebe	02.02.1970	weiblich

**Der Primärschlüssel (Primary Key)** besteht aus einem/mehreren Attributen, über deren Wert jeder Datensatz eindeutig identifiziert werden kann

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht
100	Peter	Kiel	01.01.1980	maennlich
101	Lisa	Lebe	02.02.1970	weiblich
102	...	..	..	..

- Zusammengesetzter Primärschlüssel aus Vorname und Nachname.  
Jeder Datensatz muss über Vornamen + Nachnamen eindeutig identifiziert werden können.  
*Problem: Es dürfen dann keine zwei Personen mit gleichem Namen eingefügt werden*
- Besser: Primärschlüssel auf zusätzliches Attribut Personalnummer

**Primärschlüssel auf PNR → Jede PNR darf nur einmal vergeben werden**

**Fremdschlüssel (Foreign Key)** ermöglicht es, **Beziehungen zwischen Datensätzen verschiedener Tabellen** zu definieren

**Beispiel:** Für jede Person sollen Adressdaten (Straße, Ort) gespeichert werden

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht	Ort	Strasse
100	Peter	Kiel	01.01.1980	maennlich	Trier	Nebenstr.
100	Peter	Kiel	01.01.1980	maennlich	Koblenz	Hauptstr.
101	Lisa	Lebe	02.02.1970	Weiblich	Wittlich	Kirchstr.

**1. Möglichkeit ohne Fremdschlüssel:** Erweiterung der Tabelle Person um 2 zusätzliche Attribute

→ **Problem:** hat eine Person mehrere Adressen, werden Adressdaten mehrfach hinterlegt (unnötige Datenredundanz)



**2. Möglichkeit:** Adressdaten werden in eigener Tabelle Adresse gespeichert.

**Tabelle Person**

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht
100	Peter	Kiel	01.01.1980	maennlich
101	Lisa	Lebe	02.02.1970	Weiblich

**Tabelle Adresse**

PNR	ANR	Ort	Strasse
100	1	Trier	Nebenstr.
100	2	Koblenz	Hauptstr.
101	3	Wittlich	Kirchstr.

## Problem:

Es wird nicht sichergestellt, dass eine PNR in der Adress-Tabelle tatsächlich in der Person-Tabelle existiert

*(z.B. könnte ein Datensatz in der Person-Tabelle nachträglich gelöscht werden)*

## Lösung:

Es wird ein Fremdschlüssel angelegt, der diese Beziehung der Tabellen abbildet.

# Das relationale Modell – Der Fremdschlüssel

Für PNR-Spalte der Adress-Tabelle wird Fremdschlüssel angelegt, der Primärschlüssel-Spalte PNR der Person-Tabelle referenziert.

**Primärschlüssel**

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht
100	Peter	Kiel	01.01.1980	maennlich
101	Lisa	Lebe	02.02.1970	Weiblich

**Fremdschlüssel**

PNR	ANR	Ort	Strasse
100	1	Trier	Nebenstr.
100	2	Koblenz	Hauptstr.
101	3	Wittlich	Kirchstr.

**REFERENZIERT**

**Was passiert, wenn in Adress-Tabelle Datensatz eingefügt wird mit:** PNR 20, ANR 1, Bonn, Hauptstr.

→ Datensatz kann nicht eingefügt werden, da die PNR in der Tabelle Person nicht existiert

**Was passiert, wenn in aus der Person-Tabelle der Satz mit PNR 100 gelöscht wird?**

→ abhängig vom Datenbanksystem und Angabe bei Foreign-Key Definition:

Adressdatensätze mit PNR 100 werden gelöscht

Löschvorgang wird nicht durchgeführt wegen Referenz

In Adress-Tabelle wird bei den ersten zwei Datensätzen die PNR auf NULL gesetzt

# Übungsaufgaben

In einer Datenbank liegen folgende Tabellen vor:

**Fahrer**

fahrerID	Vorname	Nachname	Einstellungsdatum
1	Max	Mustermann	01.08.2012
2	Sarah	Mueller	15.02.2008

**Fahrzeug**

FahrzeugID	Baujahr	Gewicht	Sitzplaetze
100	2005	3000	40
101	2003	4000	45

**Fahrt**

fahrtID	fahrerID	fahrzeugID	Abfahrt	Zielort
1001	1	100	01.01.16	Berlin
1002	2	101	01.11.15	Köln

1. Für welche Spalten der Tabellen Fahrzeug, Fahrt und Fahrer sollten Primärschlüssel angelegt werden?
2. Für welche Spalte welcher Tabelle sollte ein Fremdschlüssel angelegt werden?  
Welche Spalte welcher Tabelle referenziert der Fremdschlüssel?

# Normalisierung

Erste, zweite und dritte Normalform

- Ziel beim Entwurf einer Datenbank ist es, Daten so auf Tabellen zu verteilen, dass Datenredundanz vermieden wird.

## Redundanzfreie Datenspeicherung:

kein Teil des Datenbestandes kann weggelassen werden, ohne dass dabei Informationen verloren gehen.

## Beispiel für redundante Speicherung:

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht	Ort	Strasse
100	Peter	Kiel	01.01.1980	maennlich	Trier	Nebenstr.
100	Peter	Kiel	01.01.1980	maennlich	Koblenz	Hauptstr.
101	Lisa	Lebe	02.02.1970	Weiblich	Wittlich	Kirchstr.

## Normalisierung:

- Attribute werden so auf Tabellen aufgeteilt, dass **keine vermeidbaren Redundanzen** mehr vorliegen.

## Normalformen:

- Im Rahmen der Normalisierung wird Datensammlung schrittweise in die Normalformen 1. bis 5. gebracht: (4. bzw 5. Normalform in Praxis eher unüblich)
- diese bauen aufeinander auf: Datenschema in der 3. Normalform ist gleichzeitig immer in der 1. und 2. Normalform
- 1. Normalform ist dabei am schwächsten normalisiert, 5. Normalform ist am stärksten normalisiert

# Normalisierung - Ausgangslage

**Beispiel:** Ein Onlineshop verkauft Drucker und Zubehör.

KNR	Name	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2

- Alle Daten befinden sich in einer Tabelle
- Bestellt ein Kunde mehrere Artikel, wird für jeden bestellten Artikel ein Datensatz angelegt mit:
  - Datum der Bestellung
  - Kundennummer, Name und Adresse des Bestellers  
(ein Kunde hat hier immer genau 1 Adresse)
  - Artikel, die bestellte Anzahl und die Artikelkategorie/-kategorienummer

# Normalisierung – 1. Normalform

Eine Relation ist in der **ersten Normalform**, wenn alle Attributwerte atomar sind

- Wert eines Attributs darf nicht aus mehreren Werten zusammengesetzt werden
- Es kann ein eindeutiger Primärschlüssel bestimmt werden

Name beinhaltet zusammengesetzte Werte – Vorname und Nachname

unnormalisiert



KNR	Name	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2

1. Normalform



KNR	Vorname	Nachname	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina	Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2



# Normalisierung – 1. Normalform

Es kann ein eindeutiger Primärschlüssel bestimmt werden

- Eindeutiger Primärschlüssel muss hier aus mehreren Attributen zusammengesetzt werden.

KNR	Vorname	Nachname	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina	Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2

Primärschlüsselattribute

Warum kann bei dieser Zusammenstellung ein Kunde einen Artikel an einem Tag nicht mehrmals bestellen?

Eine Relation ist **in der zweiten Normalform**, wenn jedes Nichtschlüsselattribut vom Schlüsselattribut (Primärschlüssel) **voll funktional** abhängig ist.

## Begriffserklärungen

### 1. Schlüsselattribute

Alle Attribute, die Teil des Primärschlüssels sind

### 2. Nichtschlüsselattribute

Alle Attribute, die nicht Teil des Primärschlüssels sind

### 3. voll funktionale abhängig

Ein Nichtschlüssel-Attribut ist abhängig von allen Primärschlüssel-Attributen und nicht nur von einer Teilmenge des Primärschlüssels

Wert eines Nichtschlüssel-Attributs kann nur eindeutig bestimmt werden, wenn alle Attributwerte des zusammengesetzten Primärschlüssels bekannt sind

Sind alle Nichtschlüsselattribute voll funktional abhängig vom Primärschlüssel?

KNR	Vorname	Nachname	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina	Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2

## Vorname, Nachname, Straße und Ort:

- Um Namen und Adressdaten des Kunden herauszufinden, muss nur KNR bekannt sein
- Datum der Bestellung und Artikelnummer sind für diese Attribute nicht wichtig
- **Attribute sind nur abhängig von KNR → nicht voll funktional abhängig**

## Artikel und Kategorie

- Um Artikelnamen und Kategorie herauszufinden, muss nur die ANR bekannt sein
- **Attribute sind nur abhängig von ANR → nicht voll funktional abhängig**

Sind alle Nichtschlüsselattribute voll funktional abhängig vom Primärschlüssel?

KNR	Vorname	Nachname	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina	Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2

## Anzahl

- Um die Anzahl der Artikel herauszufinden, die ein Kunde an einem Tag bestellt hat, müssen sowohl KNR, ANR als auch Datum bekannt sein
- **Menge ist abhängig von allen Schlüsselattributen (KNR, ANR, Datum)**  
→ damit voll funktional abhängig

# Normalisierung – 2. Normalform

Nicht voll funktional abhängige Attribute werden in neue Tabelle übernommen  
- mit dem Schlüsselteil, von dem sie abhängig sind

## 1. Vorname, Nachname, Straße und Ort:

➤ Abhängig von Kundennummer → **Auslagerung in neue Tabelle Kunde**

KNR	Vorname	Nachname	Straße	Ort	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	1	Papier	10	Zubehör	2
122	Tom	Kiel	Nebenstr.	Trier	12.12.14	2	Toner	20	Toner	1
244	Ina	Lebe	Hauptstr.	Köln	03.05.15	1	Papier	10	Zubehör	2



**Bestellung**

KNR	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	12.12.14	1	Papier	10	Zubehör	2
122	12.12.14	2	Toner	20	Toner	1
244	03.05.15	1	Papier	10	Zubehör	2

**Kunde**

KNR	Vorname	Nachname	Straße	Ort
122	Tom	Kiel	Nebenstr.	Trier
244	Ina	Lebe	Hauptstr.	Köln

Nicht voll funktional abhängige Attribute werden in neue Tabelle übernommen  
- mit dem Schlüsselteil, von dem sie abhängig sind

## 2. Artikel, KNR und Kategorie

➡ Abhängig von Artikelnummer → **Auslagerung in neue Tabelle Artikel**

KNR	Datum	ANR	Artikel	KatNR	Kategorie	Anzahl
122	12.12.14	1	Papier	10	Zubehör	2
122	12.12.14	2	Toner	20	Toner	1
244	03.05.15	1	Papier	10	Zubehör	2



**Bestellung**

KNR	Datum	ANR	Anzahl
122	12.12.14	1	2
122	12.12.14	2	1
244	03.05.15	1	2

**Artikel**

ANR	Artikel	KatNR	Kategorie
1	Papier	10	Zubehör
2	Toner	20	Toner

# Normalisierung – 2. Normalform

Vollständiges Tabellenschema in der 2. Normalform

**Bestellung**

KNR	Datum	ANR	Anzahl
122	12.12.14	1	2
122	12.12.14	2	1
244	03.05.15	1	2

**Kunde**

KNR	Vorname	Nachname	Straße	Ort
122	Tom	Kiel	Nebenstr.	Trier
244	Ina	Lebe	Hauptstr.	Köln

**Artikel**

ANR	Artikel	KatNR	Kategorie
1	Papier	10	Zubehör
2	Toner	20	Toner

Eine Relation ist **in der dritten Normalform**, wenn kein Nichtschlüsselattribut vom Primärschlüssel **transitiv** abhängig ist.

## Begriffserklärungen

### 1. Transitive Abhängigkeit

Ein Nichtschlüsselattribut (NSA) C ist von einem anderen NSA B abhängig.  
Das NSA B ist wiederum vom Schlüsselattribut A abhängig.

Ein NSA C ist nur indirekt (über ein anderes NSA B) vom Schlüsselattribut A abhängig

→ Wenn der Attributwert B bekannt ist, kann man den Wert von C bestimmen



# Normalisierung – 3. Normalform

## Überprüfung der Nichtschlüsselattribute auf transitive Abhängigkeit

### Bestellung

KNR	Datum	ANR	Anzahl
122	12.12.14	1	2
122	12.12.14	2	1
244	03.05.15	1	2

Nur ein NSA  
→ **keine transitive Abhängigkeit**

### Kunde

KNR	Vorname	Nachname	Straße	Ort
122	Tom	Kiel	Nebenstr.	Trier
244	Ina	Lebe	Hauptstr.	Köln

Wenn Vorname bekannt ist, kann man daraus nicht auf Nachnamen, Straße oder Ort schließen.

Das gilt auch für alle anderen NSA  
→ **Keine transitive Abhängigkeit**

### Artikel

ANR	Artikel	KatNR	Kategorie
1	Papier	10	Zubehör
2	Toner	20	Toner

Wenn KatNR bekannt ist, kann man Kategorie eindeutig bestimmen

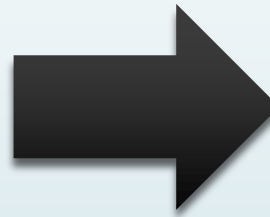
→ **Kategorie ist von ANR nur transitiv abhängig**

# Normalisierung – 3. Normalform

Kategorie wird aus Artikel-Tabelle entfernt und mit KNR in neue Tabelle übernommen

**Artikel**

ANR	Artikel	KatNR	Kategorie
1	Papier	10	Zubehör
2	Toner	20	Toner



**Artikel**

ANR	Artikel	KNR
1	Papier	10
2	Toner	20

**Kategorie**

KatNR	Kategorie
10	Zubehör
20	Toner

# Normalisierung – 3. Normalform

In der dritten Normalform ergibt sich dann also folgendes Schema

## Bestellung

KNR	Datum	ANR	Anzahl
122	12.12.14	1	2
122	12.12.14	2	1
244	03.05.15	1	2

## Artikel

ANR	Artikel	KatNR
1	Papier	10
2	Toner	20

## Kunde

KNR	Vorname	Nachname	Straße	Ort
122	Tom	Kiel	Nebenstr.	Trier
244	Ina	Lebe	Hauptstr.	Köln

## Kategorie

KatNR	Kategorie
10	Zubehör
20	Toner

# Übungsaufgabe

In einer Bibliothek werden Daten aller Bücher wie folgt in einer Tabelle gesammelt:

- Buchid, Autor, Name und Verlag des Buches
- Alle Kapitelnummern des Buches mit Namen und Seitenzahl

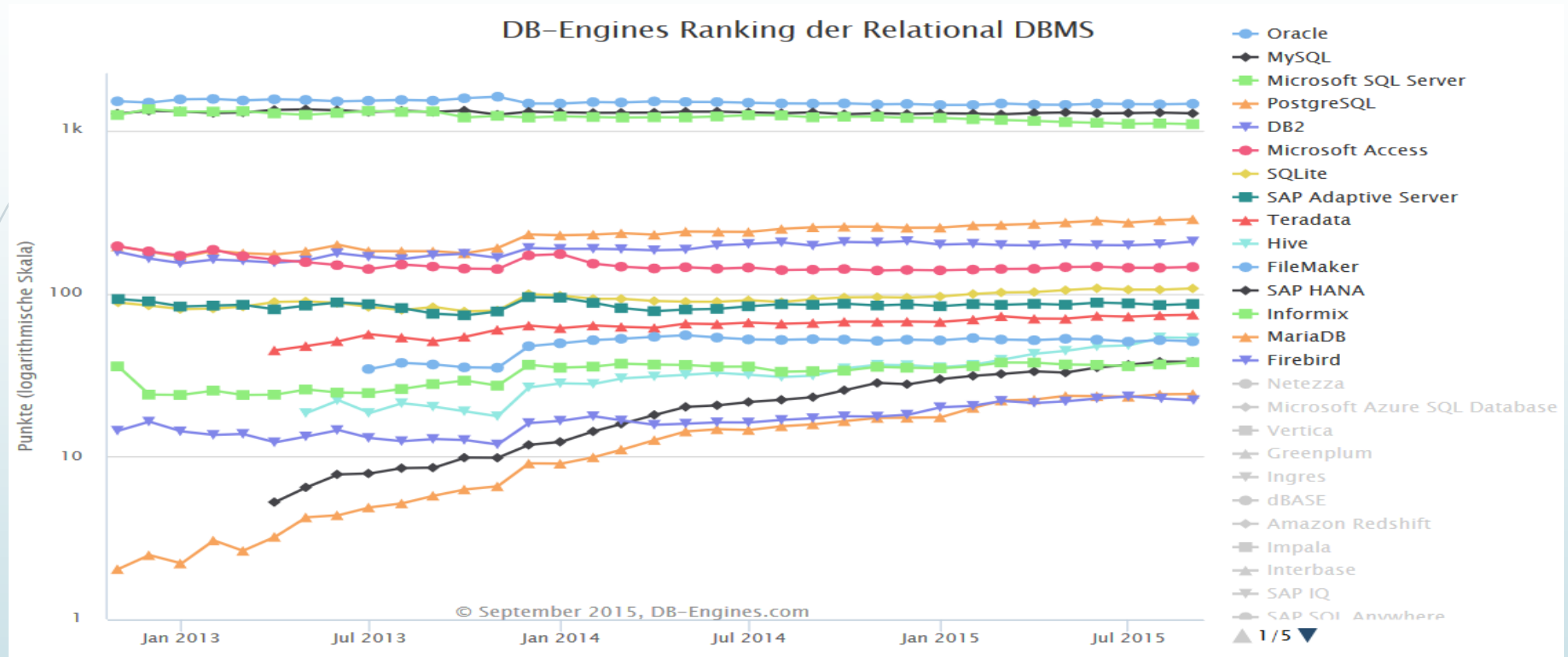
BuchID	Name, Autor	Verlag	KapitelNr	Kapitelname	Kapitelseiten
1	WI Band 1, Mustermann	Springer	1	Einführung	25
1	WI Band 1, Mustermann	Springer	2	Aufgaben der WI	40
2	Grundlagen DB, Mueller	Oreilly	1	Einführung	12

**Bringe die Tabelle schrittweise in die erste und dann in die zweite Normalform.**

# Verschiedene DBMS im Überblick

MySQL, PostgreSQL, MS SQL Server,  
Oracle 12c

# Popularität der RDBMS



Punkt-Berechnung auf Basis versch. Faktoren (Nennungen auf Webseiten, Anzahl technischer Diskussionen, Jobangebote etc)

Quelle: [http://db-engines.com/de/ranking\\_trend/relational+dbms](http://db-engines.com/de/ranking_trend/relational+dbms)

## D A T A B A S E

### Lizenz

- Kommerziell (Eingeschränkt kostenlose Version)

### Entwicklung

- Seit 1979, entwickelt von ORACLE

### Verbreitung und Einsatz

- Zählt zu den am weitesten verbreiteten RDBMS
- Wegen vergleichsweise hoher Lizenzkosten und sehr mächtiger Technologie Einsatz vor allem in größeren Unternehmen

## Lizenz

- Open Source
- mittlerweile auch kommerzielle Lizenz mit erweiterter Funktionalität

## Entwicklung

- Seit 1994 , entwickelt von MySQL AB
- gehört mittlerweile ORACLE

## Verbreitung und Einsatz

- Sehr weit verbreitet
- Einsatz vor allem im Webbereich  
– vergleichsweise einfache Administration





# Microsoft SQL Server

## Lizenz

- Kommerziell (Eingeschränkt kostenlose Version)

## Entwicklung

- Seit 1989
- Entstanden aus Kooperation zwischen Microsoft und Sybase

## Verbreitung und Einsatz

- Weite Verbreitung in kleinen/mittelständigen Unternehmen
- Einsatz auf Windows Betriebssystemen



# PostgreSQL

## Lizenz

- Open Source

## Entwicklung

- Seit 1989 Postgre, seit 1996 Postgre SQL
- Entwickelt von der PostgreSQL Global Development Group

## Verbreitung und Einsatz

- Weite Verbreitung
- in den meisten Linux-Distributionen enthalten

## Sonstiges

- Zählt zu den objektrelationalen Systemen
- Im Vergleich zu MySQL sehr weite Entwicklung

