Datenbanken

Informatik, ICS und als Wahlfach

3. Datenbankmodelle

Prof. Dr. Markus Goldstein

SoSe 2022

Inhalt

Fallstudie - Kiosk

- Sie haben nun ein **konzeptionelles Datenmodell** für Ihren Kiosk erstellt. Jetzt möchten Sie dieses Modell als Datenbank nutzen. Hierfür gibt es unterschiedliche Datenbanksoftware-Anbieter.

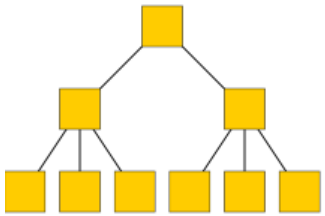
- Jeder Datenbanksoftware liegt ein sogenanntes **Datenbankmodell** zu Grunde, dass verwendet wird, um die Datensätze und deren Beziehungen abzulegen.

Datenbankmodelle

Datenbankmodell (engl. database model)

- logisches Beziehungsgebilde
- Beschreibung der Art und Weise der *Verbindung* von Datensätzen

Traditionelle Datenbankmodelle



- Hierarchisch: nur einen Elternknoten (→ 1:1 und 1:N)
- Netzwerk (Erweiterung; auch N:M)
- **Relational (RDBM)**
- Objektorientiert
- Objekt-relational (Mischform aus Relational und OO)

Datenbankmodelle

Datenbankmodell (engl. database model)

- logisches Beziehungsgebilde
- Beschreibung der Art und Weise der *Verbindung* von Datensätzen

„Neue“ Datenbankmodelle (seit 2005): NoSQL

- Spaltenbasiert (engl. wide column)
- Dokumentenorientiert
- Graph
- Key-Value

Inhalt

Relationales Datenmodell

Theorie

- Streng mathematisch (Relationale Algebra)
- Entitätstypen werden als **Relationen** (Tabellen) gespeichert
- Beziehungen zwischen Entitäten werden über **Referenzattribute** (Fremdschlüssel) in Tabellenspalten gespeichert
- Definition *Relation*:
  Teilmenge der Produktmenge der Wertebereiche der Attribute

Relation

Beispiel - Kartenspiel

- F = {Karo, Herz, Pik, Kreuz}
- C = {7, 8, 9, 10, Bube, Dame, König, Ass}

→ **Produktmenge F x C** entspricht vollständigem Poker-Blatt **(kartesisches Produkt)**

Relation als Teilmenge der Produktmenge

- z.B. Full House:
  {(Pik, Ass), (Kreuz, Ass), (Herz, Ass), (Herz, König), (Karo, König)}

→ Darstellung als Tabelle

Farbe	Karte
Pik	Ass
Kreuz	Ass
Herz	Ass
Herz	König
Karo	König

RDBM - Speicherung

Darstellung des Entitätstyps „Lehrer“

Typbeschreibung + Relation

Lehrer (Personalnummer, Vorname, Familienname, Geburtsdatum, Familienstand)

{ (54321, Theo, Retisch, 31.12.1980, ledig),
(43210, Ellen, Bogen, 01.01.1970, verheiratet),
(32109, Claire, Grube, 30.06.1975, geschieden)
}

Darstellung als Tabelle

Lehrer

Personalnummer	Vorname	Familienname	Geburtsdatum	Familienstand
54321	Theo	Retisch	31.12.1980	ledig
43210	Ellen	Bogen	01.01.1970	verheiratet
32109	Claire	Grube	30.06.1975	geschieden

Relation (Mathematische Darstellung)

Als Tabelle

RDBM - Speicherung

Schreibweise als Relation im Allgemeinen

Tabellenname (Attribut1, Attribut2, …)

- Primärschlüssel werden unterstrichen

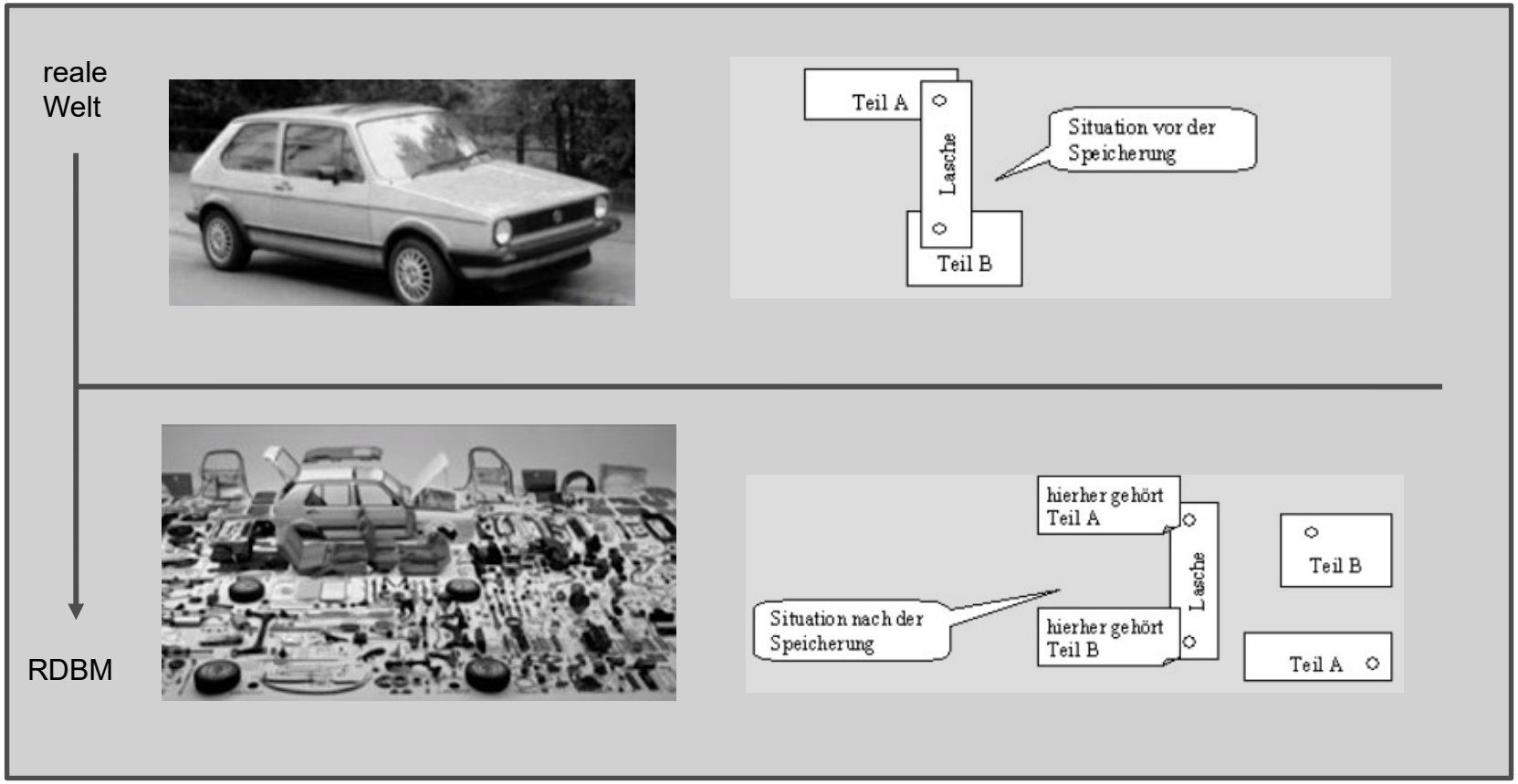
Darstellung mit Daten (= Relation) als Tupel ( ) einer Menge { }:

Tabellenname (Attribut1, Attribut2, …) {
(WertA1, WertA2, …), (WertB1, WertB2, …),
…
}

RDBM - Speicherung

Komplexe Entitätstypen werden in kleinere zerlegt

- Vor- und Nachteil zugleich (vergl. Kapitel 2)



RDBM - Grundbegriffe

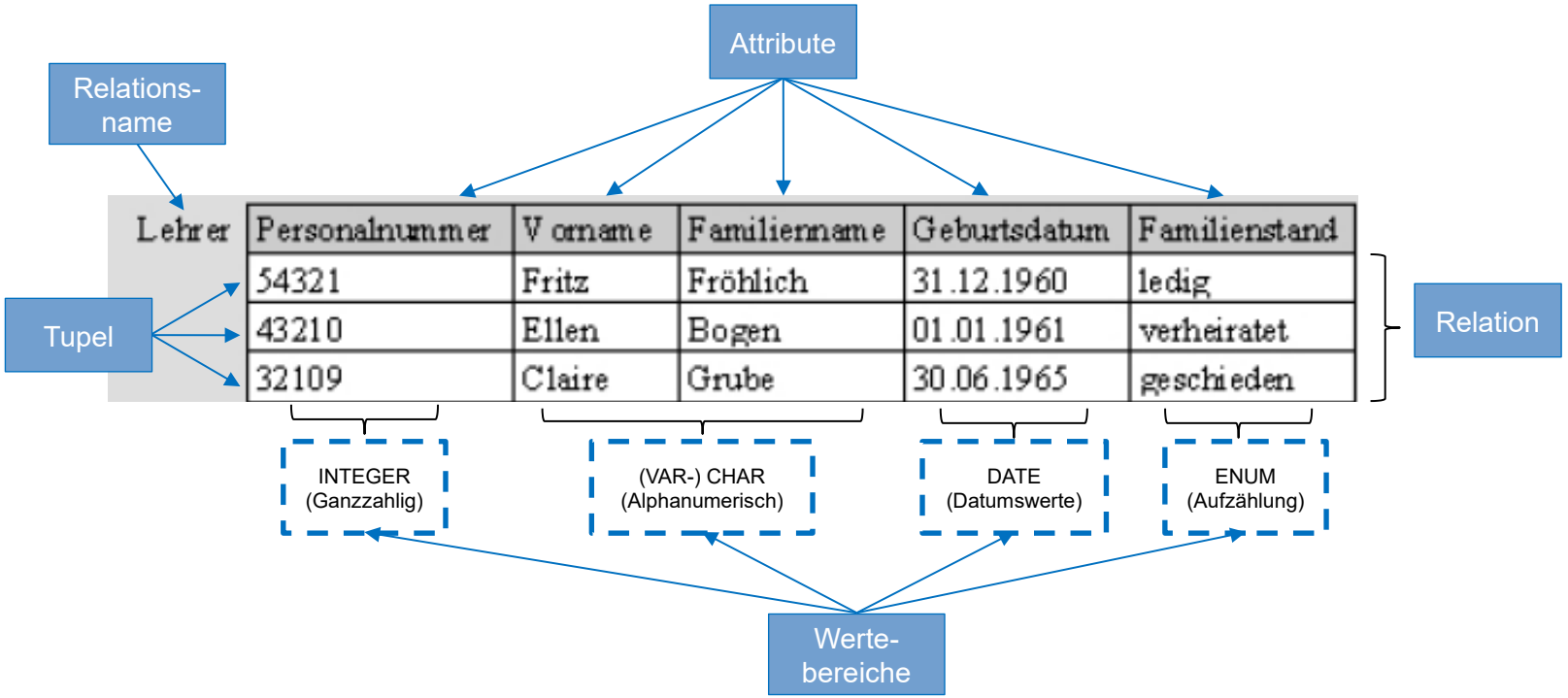
Grundbegriffe

- **Relation** repräsentiert einen Entitätstypen und ist umgesetzt als **Tabelle**
- Ein **Attribut** repräsentiert eine Eigenschaft eines Entitätstyps und wird zur **Tabellenspalte**
  - Wertebereich/Domäne fasst mögliche Werte eines Attributs zusammen (s. Datentypen)
- Ein **Tupel** ist Element einer Relation, wird zur **Tabellenzeile** und repräsentiert eine konkrete **Entität**.

RDBM - Übersicht

Zusammenfassung der Begrifflichkeiten

- Relation → Darstellung als **Tabelle**
- Reihenfolge der Zeilen(/Spalten) spielt keine Rolle



Schlüssel

Ein **Schlüssel** (engl. key) ist jede identifizierende Attributmenge, die minimal ist.

- Einfach: ein Attribut
- oder zusammengesetzt: Attributkombination (composite key)

  (Im relationalen Datenmodell gibt es *keinen* identifizierenden Beziehungstyp mehr)

Eigenschaften

- Eindeutige Identifizierung von Tabellenzeilen
- Minimal heißt, Untermenge nicht (mehr) ausreichend zur eindeutigen Identifizierung
- Evtl. mehrere **Schlüsselkandidaten** (engl. candidate key)

Spezielle Schlüssel

Primärschlüssel (engl. primary key) ist ein ausgewählter Schlüsselkandidat der für die Identifizierung einzelner Tupel verwendet wird.

- Empfehlungen:
  Auswahl des Schlüssels mit minimaler Anzahl Zeichen
  Auswahl passend zur Semantik
  Einführung eines Surrogats, falls der PK als FK verwendet wird

- Nicht ausgewählte Schlüsselkandidaten
  - Alternativ- (engl. alternate key) bzw.
  - Sekundärschlüssel (engl. secondary key)

Definition Fremdschlüssel

Fremdschlüssel (engl. foreign key) ist eine Attributmenge, die einen Primärschlüssel einer (anderen) Relation referenziert

- Primärschlüssel aus Tabelle A wird als (Verweis-) Attribut in Tabelle B aufgenommen
- bewirkt logische Verbindung von Tabellen
  (also z.B.: „hierhin gehört Teil A“)
- Darstellung: Nicht unterstrichen (aber ggf. kursiv oder mit Symbol ⇑)

Fremdschlüssel – einfaches Beispiel

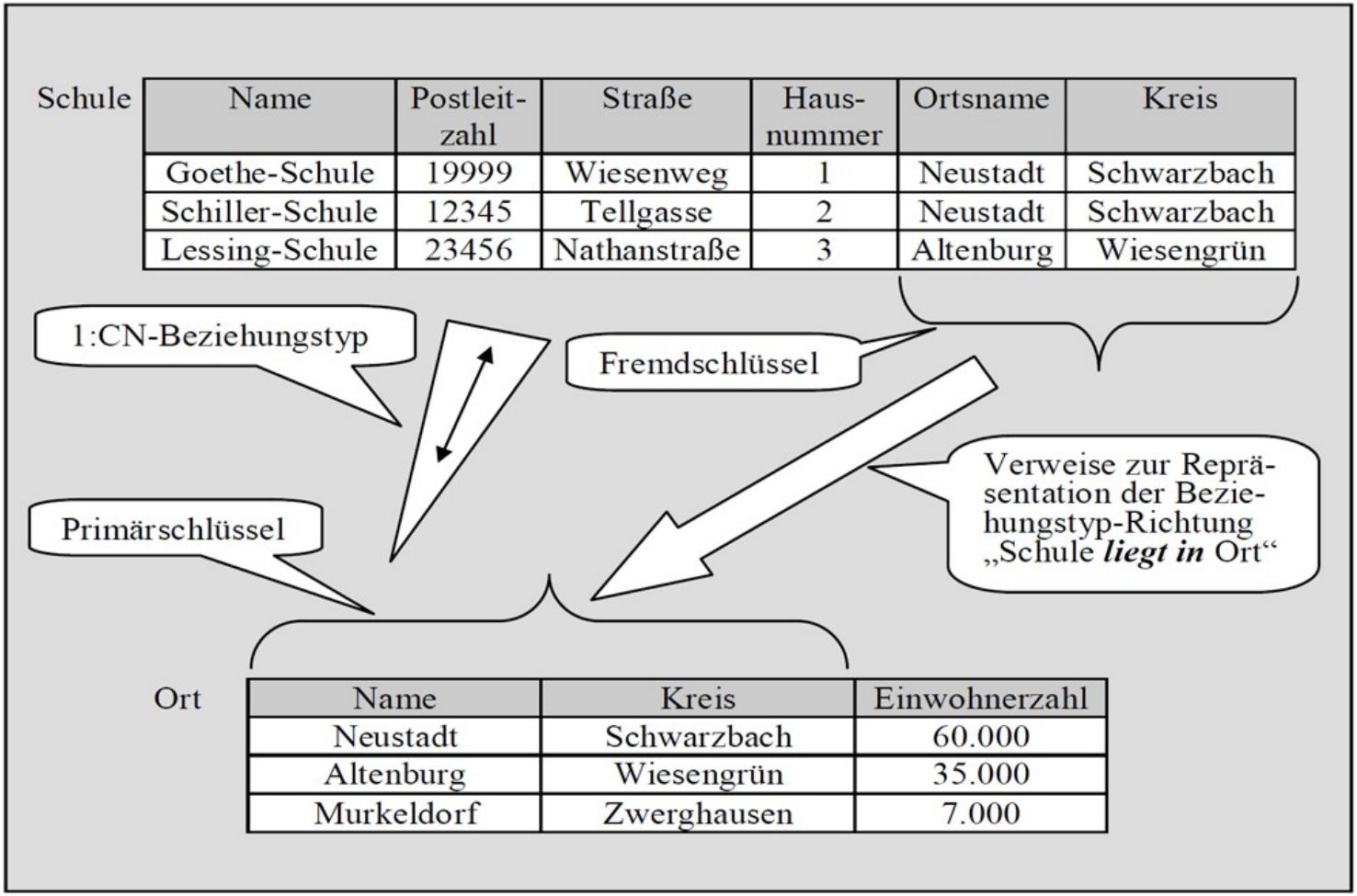
Fremdschlüssel zur Repräsentation von Beziehungstypen

KundeID	Name	⇑AutoID⇑
33	Müller	1
34	Fröhlich	3
35	Dampf	1
…	…	

AutoID	Name
1	TukTuk
2	Merzädes
3	CMW
…	…

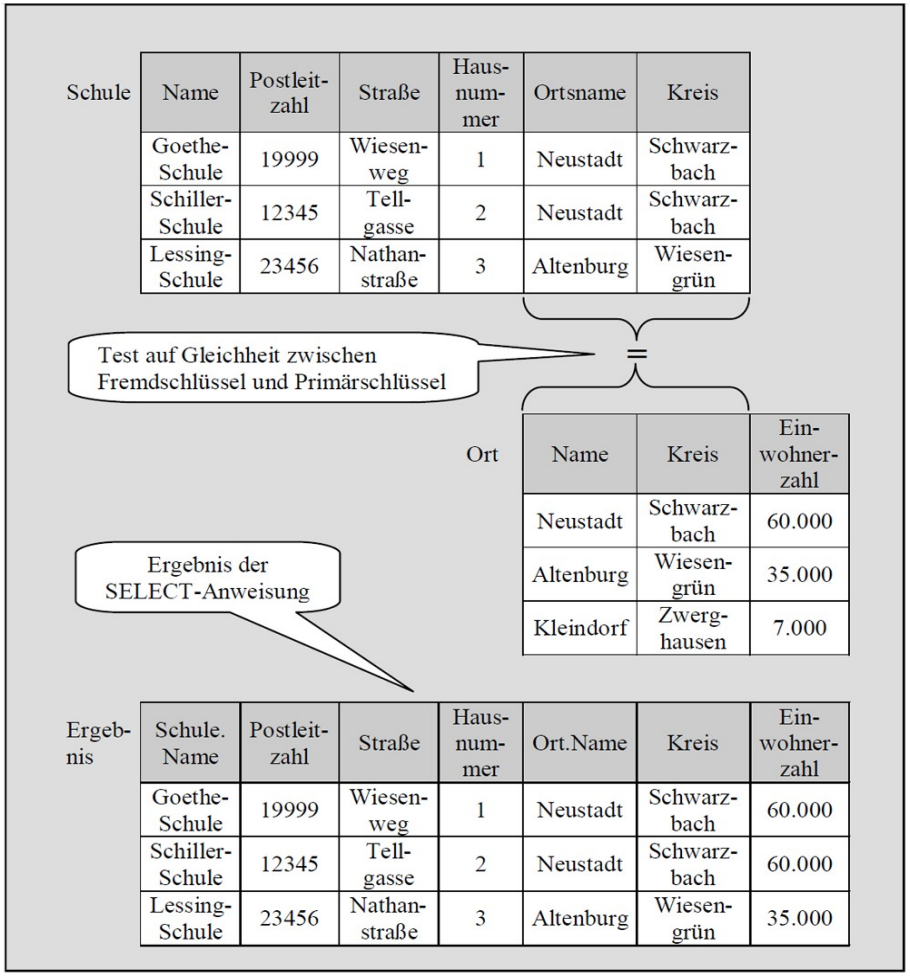
Fremdschlüssel - Zweck

Fremdschlüssel zur Repräsentation von Beziehungstypen



Fremdschlüssel – Zweck II

Fremdschlüssel zur Herstellung von Verbundinformationen (JOINs)



Integritätsbedingung RDBM

Datenintegrität muss trotz Aufteilung gewährleistet sein:

- **Entitätsintegrität** (engl. entity integrity)
  - Jedes (Primär-)Schlüsselattribut *muss einen Wert haben* (NOT NULL)
  - Kombination der Attributwerte (=Primärschlüssel) muss innerhalb einer Tabelle *eindeutig* sein.

- **Referenzielle Integrität** (engl. referential integrity)
  - Fremdschlüsselwerte *müssen* in referenzierter Tabelle als Primärschlüsselwerte existieren
  - oder Fremdschlüsselattribut hat keinen Wert (NULL)
    (→ nur bei Optionalität)
  - Umgangssprachlich: „broken links“ sind verboten!

Integritätsbedingung RDBM

Datenintegrität wird durch RDBMS unterstützt

- **Entitätsintegrität** (unterstützt jedes RDBMS)
  Einfügen eines nicht-existierenden Wertes (NULL) oder ändern auf einen solchen wird abgelehnt. Doppelte Einträge werden auch abgelehnt.

- **Referenzielle Integrität** (unterstützt nicht jedes RDBMS)
  - Was passiert bei Änderung/Löschung eines Primärschlüsselwertes?
    ON DELETE ...
  - ... RESTRICT → Ablehnung (oft Standard)
  - … CASCADE → Fremdschlüsselwert bei Änderung des Primärschlüsselwerts anpassen bzw. löschen
  - … SET NULL → Fremdschlüsselwert auf NULL setzen (nur bei Optionalität möglich!)

Integritätsbedingung RDBM

Vorsicht bei ON DELETE CASCADE

- Hier im Beispiel: Alle Lieferungen werden mit gelöscht!

(angelehnt an Thomas Kudraß, Taschenbuch Datenbanken, 2. Auflage, Hanser Verlag 2015, S. 74)

Inhalt

NoSQL Datenbankmodelle

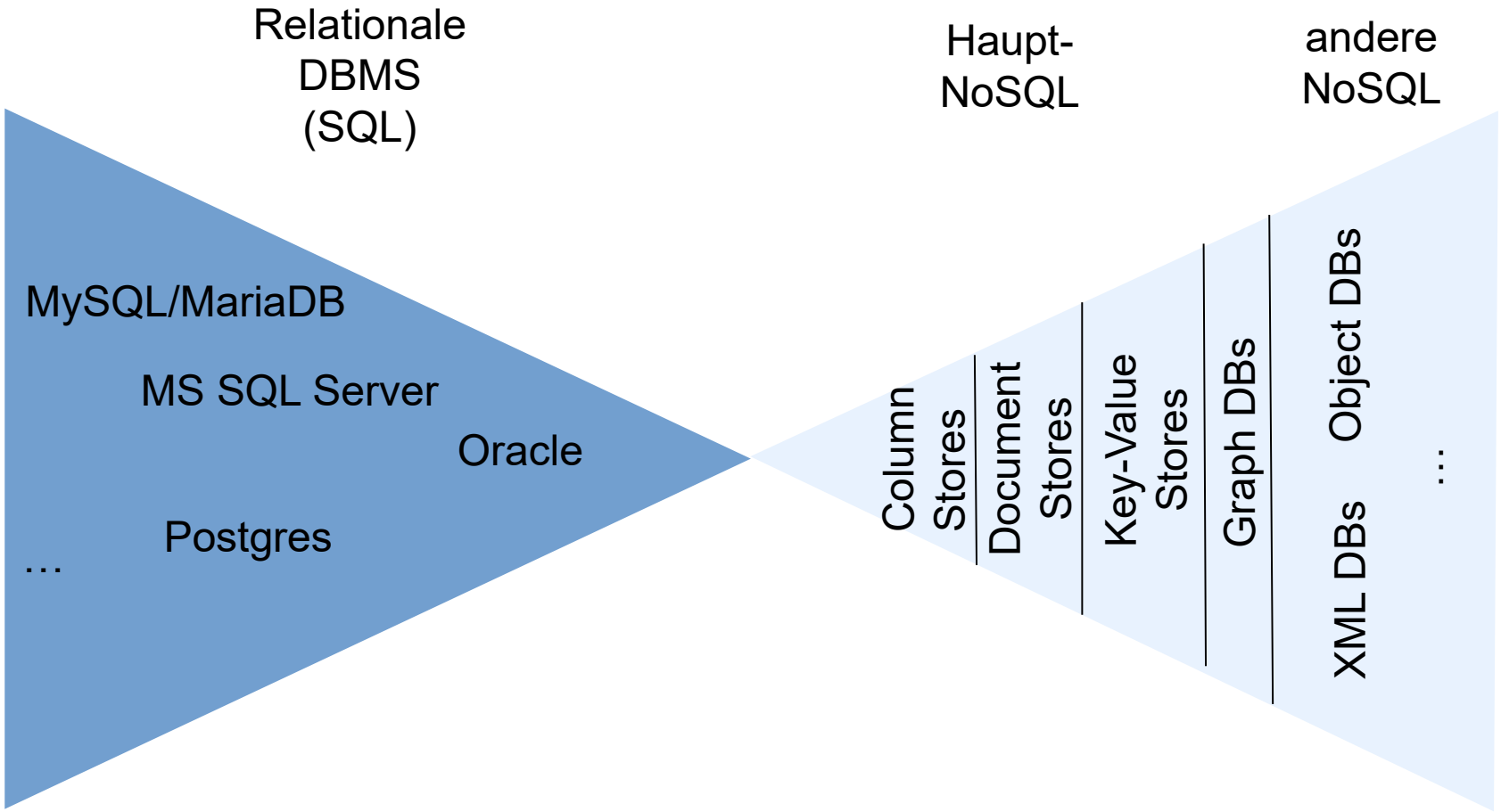
NoSQL Eigenschaften

- das Datenmodell ist nicht-relational („not-only-SQL“)
- keine Relationen/Tabellen
- **Sammelbegriff** für nicht-relationale Datenmodelle

- NoSQL Modell ist oft frei von einem mathematischen Schema
  - Schemaverantwortung in Anwendungsprogramm
  - Höhere Flexibilität
  - Oft verwendet für sehr große Datenmengen („Big Data“)
- NoSQL DBMS nutzen oft einfache API
- i.d.R. keine SQL-Schnittstelle sondern REST oder andere

RDBM vs. NoSQL

- Übersicht



Quelle: in Anlehnung an Edlich et al. , NoSQL – Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Hanser 2. Auflage 2011, p. 6

Key-Value Datenbankmodell

Einfachstes Datenbankmodell. Eintrag besteht aus

- Key – Schlüssel (eindeutig)
- Value – Wert (eigentliche Daten), beliebig, z.B.
  - Zeichenketten
  - Arrays, Listen, Mengen
  - Binärdaten

- Konzept in Java: „Hash Map“
- Werte haben **kein festes Schema**
- Auf Festplatte und in-memory (nur im Arbeitsspeicher)
- Einfache API: put/ get/ delete

Key-Value Datenbankmodell

Beispiele

Kunden

Key	Value
customer:1:name	Muscle Inc.
customer:1:phone	(234) 567-8901
customer:1:city	New York
…	…

Telefonbuch

Key	Value
Mike	(123) 456-7890
Tom	+49 176 678901
Tina	(345) 678-9012
Adele	(456) 789-0123
…	…

IP-Forwarding

Key	Value
202.45.12.34	01:23:36:0f:a2:33
202.45.123.4	00:25:33:da:4c:01
245.12.33.45	02:03:33:10:e2:b1
…	…

Spaltenorientiertes Datenbankmodell

- Daten werden spaltenweise gespeichert, nicht als Tupel (zeilenweise)
- Umgangssprachlich: „Jedes Attribut hat eine eigene Tabelle“
- Vergleich zu RDBMS

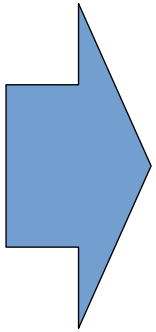
Spaltenbasiert	Relational
`1, 2, 3, Tom, Mary, Paul, 42, 18, 36`	`1, Tom, 42, 2, Mary, 18, 3, Paul, 36`

Vorteile	Nachteile
Schnelles Lesen einzelner Spalten (→ „Analytics“)	Langsames Aktualisieren und Einfügen
Kompression/Aggregation	Langsames Lesen von ganzen Tupeln
Für sehr große Datenmengen	

Spaltenorientiertes Datenbankmodell

- Beispiel

Id	Product	Name	Date	Price
1	Beer	Thomas	2017-02-27	2
2	Beer	Thomas	2017-02-28	2
3	Vodka	Thomas	2017-02-28	10
4	Whiskey	Chris	2017-02-28	5
5	Whiskey	Chris	2017-02-28	5
6	Vodka	Michael	2017-02-29	10



Id	Product
1	Beer
2	Beer
3	Vodka
4	Whiskey
5	Whiskey
6	Vodka

Id	Name
1	Thomas
2	Thomas
3	Thomas
4	Chris
5	Chris
6	Michael

Id	Price
1	2
2	2
3	10
4	5
5	5
6	10

Id	Date
1	2017-02-27
2	2017-02-28
3	2017-02-28
4	2017-02-28
5	2017-02-28
6	2017-02-29

Spaltenorientiertes Datenbankmodell

- Beispiel: Kompression

Id	Product
1-2	Beer
3	Vodka
4-5	Whiskey
6	Vodka

Id	Name
1-3	Thomas
5	Chris
6	Michael

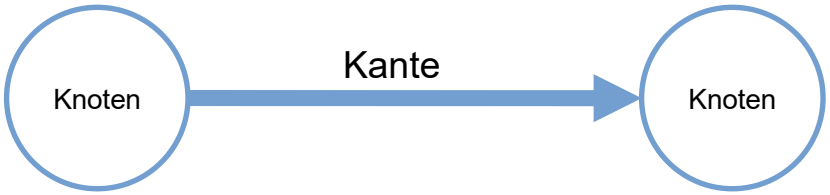
Id	Price
1-2	2
3	10
4-5	5
6	10

Id	Date
1	2017-02-27
2-5	2017-02-28
6	2017-02-29

- Technische Realisierung: Map von Maps → Spaltenwerte dürfen fehlen („NULL“ → Eintrag in einer Spalte fehlt)
- „Verteilen“ von Tabellen auf mehrere Server möglich
- Schnelles Bearbeiten von Spalten

Graph als Datenbankmodell

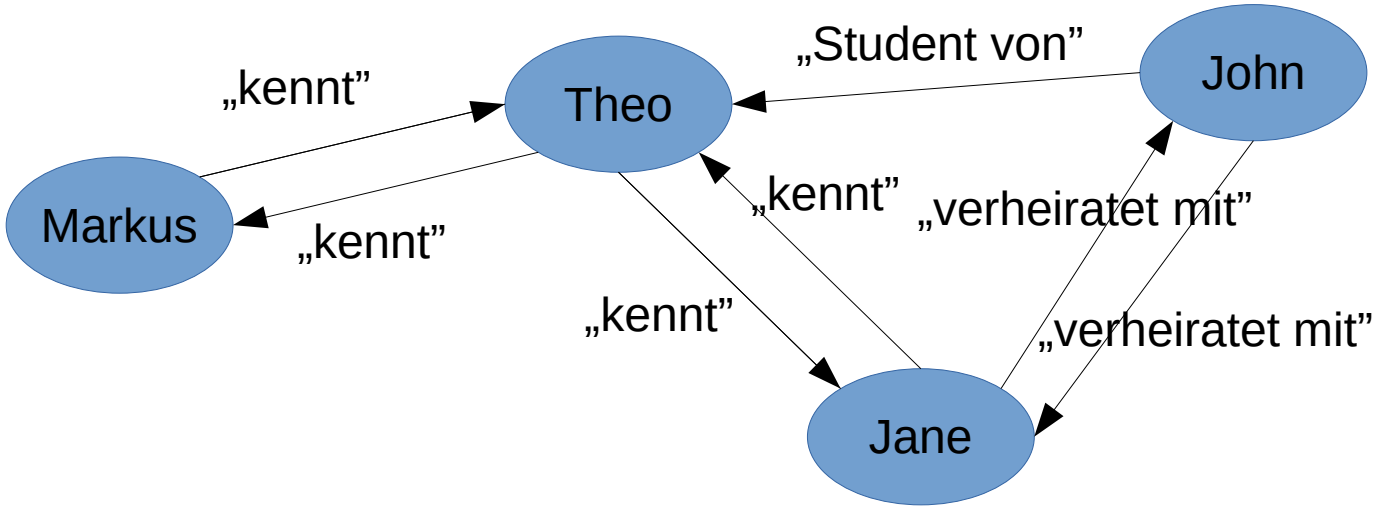
Ein Graph ist ein mathematisches Modell



- Knoten sind Entitäten (nicht Entitätstyp!)
  - Attribute am Knoten speicherbar
- Kanten sind Beziehungen zwischen einzelnen Entitäten
  - Kanten sind gerichtet: Beziehung nur in eine Richtung
  - Bei gegenseitiger Beziehung: weitere Kante
  - Kanten können Rollennamen bekommen
- **Viel flexibler als RDBMS, da Beziehungen zwischen Entitäten, nicht nur zwischen Entitätstypen**

Graph als Datenbankmodell

Beispiel



- Graphdatenbanken Eigenschaften
  - Effiziente Speicherung
  - Implementieren von Algorithmen („Kennt Markus John über Umwege?“)
  - Eigene Abfragesprache für Graphen (kein Standard)

Dokumentbasiertes Datenbankmodell

Datenbanken speichern Dokumente

- Struktur wird durch die Anwendung festgelegt
- Unterschiedliche Dokumente in einer Datenbank müssen **nicht** die gleiche Struktur aufweisen (großer Unterschied zu RDBM)
- Dokument über Primärschlüssel identifiziert (Surrogat „id“)
- Dokumente sind oft in JSON definiert

```
{
    "id": 1,
    "name": "A green door",
    "price": 12.50,
    "tags": ["home", "green"]
}
```

Exkurs: JSON

JSON besteht aus (nur) zwei Strukturelementen

- Variablen: Sammlung von Name/Wert-Paaren
  `string: value z.B. "id": 1`
- Komplexe Strukturen: Liste von Werten, Unterobjekte
  `liste: [value1, value2,…] z.B. ["home", "green"]`
  `unterobjekt: {...}`

Werte sind vom Typ

- Zeichenkette (String)
- (Komma)zahl
- Boolean (true or false)
- Liste [...]
- NULL
- Object {}
  Objekte können also Unterobjekte von anderen Objekten sein

Exkurs: JSON

Beispiele

Style

```
{"height": 250, "color" :"blue"}
```

Mitarbeiter

```
{"employees": [
    {"firstName": "John","lastName":"Doe","age":30},
    {"firstName": "Anna","lastName":"Smith","age":21},
    {"firstName": "Peter","lastName":"Jones"},
]}
```

- Peter hat kein Alter (Schemafreiheit!)

- **Referenzen möglich über**
  - Fremdschlüssel
  - Einfache Unterdokumente/Unterobjekte (Vorsicht: Redundanz!)

Aufgaben

Bitte bearbeiten Sie jetzt die Aufgaben in Moodle zum Kapitel 3.

- **Teil B**

Inhalt

Datentypen

Attributwerte werden unter Verwendung von **Datentypen** gespeichert

- Angabe der Datentypen bei der Modellierung
- Definition der Datentypen beim Anlegen der Tabelle

Datentypen haben unterschiedliche Größe (in Bit)

- Auswahl der Datentypen hat Auswirkung auf Speicherbedarf der Datenbank
- Ziel ist die Verwendung der Datentypen mit minimalem Speicherbedarf
- ABER: Datentypen unter Berücksichtigung zukünftiger Werte

Datentypen

Jedem Attribut (Spalte) wird ein Datentyp zugewiesen

- Vergleiche Kiosk-Beispiel

SchülerIn	Klasse	Geschlecht	Sparte	Artikel	Preis	Zahlung	MWST	Laden	Datum	Zeit	Kommentar
Joel Brandeis	2f00	m	Lebensmitte	Eier	SFr. 2,80	bar	231102	Migros Höngg	6.4.00	15:30	Kasse 1
Joel Brandeis	2f00	m	Lebensmitte	Sprite	SFr. 1,20	bar	380507	Epa 8001 Zürich	2.2.01	12:24	
Joel Brandeis	2f00	m	Lebensmitte	Käse	SFr. 4,85	bar	231074	Coop Zürich-Linth	26.6.95	12:39	
Alana Gerdes	2b01	w	Lebensmitte	Farmer-Stengel	SFr. 4,00	Bar	388507	Epa Aarau	11.4.00	10:03	Kasse 3
Alana Gerdes	2b01	w	Transport	Tageskarte Zone 10	SFr. 5,20	bar		Looren Apotheke Witikonerstrasse 397 8053 Z	1.9.99	3:36	0230 0001 00004006 00142
Alana Gerdes	2b01	w	Kleider	Nachtwollhemd	SFr. 30,00	Bar	231102	Migros Ebmatingen	22.3.01	9:22	
Caroline Diethelm	2b01	w	Lebensmitte	Toni Vollrahm	SFr. 2,10	Bar	231074	Coop Zuerich	3.3.00	15:55	
Caroline Diethelm	2b01	w	Lebensmitte	Tragtasche	SFr. 0,30	bar	235289	Studentenladen	25.7.00	12:54	
Caroline Diethelm	2b01	w	Lebensmitte	Rohschinken	SFr. 15,42	bar	231074	coop Dietikon 8953	1.9.00	16:19	Fr. B. Holzer Kasse 01
Caroline Diethelm	2b01	w	Transport	Tageskarte	SFr. 5,20	bar	231102	Migros Rigiplatz 8006 Zürich	2.9.00	9:35	Frau B. Holzer Kasse 1
Caroline Diethelm	2b01	w	Lebensmitte	Barclay	SFr. 4,70	bar	231074	Coop Zürichbergstrasse 8032 Zürich	17.3.01	13:39	R.Budel, Kasse 1
Caroline Diethelm	2b01	w	Lebensmitte	Paprikachips	SFr. 3,95	bar	231074	COOP Zuerich-Linth ,Cl. Römerhof	7.4.00	7:40	
Caroline Diethelm	2b01	w	Lebensmitte	Zwiebeln gehackt	SFr. 1,90	Bar	251030	Body Shop 8001 Zürich 2606922	10.12.99	15:02	
Caroline Diethelm	2b01	w	Lebensmitte	DIM SUM Poulet	SFr. 4,50	Karte		Selecta Automat	10.10.00	14:50	
Caroline Diethelm	2b01	w	Lebensmitte	Edelnuss Misch.	SFr. 3,90	bar	231074	Coop Zürich-Linth St. Annahof Volkiland	10.3.00	17:05	
Caroline Diethelm	2b01	w	Papeterie	Papeterie	SFr. 1,40	Karte	231074	Coop Zürichbergstr.	5.8.00	11:36	M.Mueller
Caroline Diethelm	2b01	w	Lebensmitte	Toni Vm Past	SFr. 1,60	Bar	231074	Coop 8706 Feldmeilen	7.9.00	12:31	
Caroline Diethelm	2b01	w	Lebensmitte	Nektarinen Gelb	SFr. 0,50	bar	231102	Migros Höschgasse Zürich	31.8.00	14:40	
Caroline Diethelm	2b01	w	Lebensmitte	Truffes-Cake gross	SFr. 27,00	Bar	231102	M Rigiplatz	3.9.99		Kasse 2
Caroline Diethelm	2b01	w	Papeterie	Mäppchen	SFr. 1,90	bar	231124	Beach Mountain	24.7.00	15:16	
Caroline Diethelm	2b01	w	Lebensmitte	Melonen	SFr. 1,80	Bar	380507	EPA Bellevue Zürich	3.4.00	17:57	
Caroline Diethelm	2b01	w	Kleider	Hose Figura	SFr. 6,50	Bar	231074	COOP Zürich-Linth	17.7.99	10:12	
Caroline Diethelm	2b01	w	Unterhaltung	Notting Hill	SFr. 11,00		231074	Coop Zürich-Linth	4.9.99	13:50	Balkon links Grosse Bühne Reihe 10 Platz 775

Nummerische Datentypen

Numerische Datentypen

- Ablage von numerischen Werten (Zahlenwerte)
- Nicht jede Zahl hat einen Wert (z. B. Postfach oder Hausnummer)
  NULL vs Wert „0“
- Vorzeichenbehaftet (**SIGNED**) oder vorzeichenlos (**UNSIGNED**)

Je nach Bereich der Attributwerte

- Ganzzahlige Werte
  -4,-3,-2,-1,0,1,2,3,4,… z.B. Anzahl Teilnehmer
- Kommazahlen
  3.14159265359, 0.5 z.B. PI oder Preis

Ganzzahlige Datentypen

- **BOOLEAN** (1 Byte)
  - Wahrheitswerte wahr oder falsch
  - Ablage als 1 = wahr oder 0 = falsch
- **TINYINT** (1 Byte)
  - Sehr kleine Werte
  - -128 bis 127 (0 bis 255 UNSIGNED)
- **SMALLINT** (2 Byte)
  - -32.768 bis 32.767 (0 bis 65.535 UNSIGNED)

Ganzzahlige Datentypen II

- **MEDIUMINT** (3 Byte)
  - -8.388.608 bis 8.388.607
  - (0 bis 16.777.215 UNSIGNED)
- **INT** bzw. INTEGER (4 Byte)
  - -2.147.483.648 bis 2.147.483.647
  - (0 bis 4.294.967.295 UNSIGNED)
- **BIGINT** (8 Byte)
  - Sehr große Zahlen
  - -9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807
  - (0 bis 18.446.744.073.709.551.615 UNSIGNED)

Kommazahlen

Unterscheidung nach Fließkomma und Festkommazahl

- **Festkommazahl** hat immer den gleichen Exponenten
  → Dezimalpunkt ist immer an gleicher Stelle
  z.B. Preise, Ausmaße, etc.

- **Fließkommazahl** hat wechselnde Exponenten
  → Komma ‚wandert' je nach Exponent
  Beispiel: $0.01 * 10^{1} = 0.1 * 10^{0} = 1.0 * 10^{-1} = 10.0 * 10^{-2}$

  - Berechnete Werte

  - Exponentialdarstellung bzw. wissenschaftliche Notation
    $1 * 10^{1}$ entspricht 1E+1
    $1 * 10^{-1}$ entspricht 1E-1

Fließkommazahlen

- **FLOAT** (4 Byte)
  - Kleine Fließkommazahlen mit einfacher Genauigkeit
  - Wertebereich
    -3.402823466E+38 bis -1.175494351E-38
    0
    1.175494351E-38 bis 3.402823466E+38

- **DOUBLE** (8 Byte)
  - Fließkommazahlen mit doppelter Genauigkeit
  - Wertebereich
    -1.7976931348623157E+308 bis -2.2250738585072014E-308
    0
    2.2250738585072014E-308 bis 1.7976931348623157E+308

Festpunktzahlen

DECIMAL(M,D)

- Angabe von exakten Zahlen
- Festlegung der Länge sowie der Nachkommastellen
  - M = Anzahl Gesamtstellen der Zahl (1 - 64)
    Standard: M = 10
  - D = Anzahl Stellen nach dem Komma (0 – 30)
    Standard: D = 0
- Beispiel: Angabe des Datentyps für das Attribut Preis
  - **DECIMAL(7,2) UNSIGNED**
    → Preise von 0.00 – 99 999.99

Datums-Datentyp

Speicherung von Datums- und Zeitwerten

- Dienen zur Berechnung von Datums- und Zeitwerten
- **DATETIME, DATE** und **TIMESTAMP** gehören zusammen
  - DATETIME speichert Datum und Uhrzeit
    '1000-01-01 00:00:00' bis '9999-12-31 23:59:59
  - DATE speichert nur Datum
    '1000-01-01' bis '9999-12-31'
  - TIMESTAMP speichert Zeitstempel im UNIX-Format
    '1970-01-01' bis '2037-12-31' mit Sekundenbruchteilen (microseconds)
- **YEAR**
  - Werte von ‘1901‘ bis ‘2155‘
- **TIME**
  - Erfassung von Zeiten (auch >24h): '-838:59:59' bis '838:59:59'

String Datentypen

Speicherung von **alphanumerischen Zeichenketten (String)**
z.B. Name, Vorname, etc.

- Unterscheidung in feste und variable Länge
- **CHAR(N)**
  - Ablage von Zeichenketten der festen Länge N
  - 0 <= N <= 255
  - Ungenutzte Zeichen werden mit Leerzeichen aufgefüllt
- **VARCHAR(N)**
  - Ablage von Zeichenketten mit variabler Länge bis zu N
  - N <= 255
  - Ungenutzte Zeichen werden nicht abgelegt
  - Leerzeichen am Ende werden entfernt

String Datentypen

Speicherplatzbedarf CHAR vs. VARCHAR

- CHAR benötigt 1 Byte pro Zeichen
  - Auch Leerzeichen benötigen Platz
  - Verschwendung bei vielen kleineren Strings
- VARCHAR benötigt 1 Byte zusätzlich pro Spalte
  - Spart Platz bei variabler Länger der Strings
  - Überflüssig bei String mit fester Länge
- Speicherplatzbedarf mit UTF-8 ggf. höher

Wert	CHAR(4)	Erforderlicher Speicherplatz	VARCHAR(4)	Erforderlicher Speicherplatz
''	' '	4 Byte	''	1 Byte
'ab'	'ab '	4 Byte	'ab '	3 Byte
'abcd'	'abcd'	4 Byte	'abcd'	5 Byte
'abcdefgh'	'abcd'	4 Byte	'abcd'	5 Byte

Längere String Datentypen

Speichern von längeren Texten (Strings der Länge L) mit **TEXT-Typen**

- **TINYTEXT** (L+1 Byte, wobei $L < 2^8$)
- **TEXT** (L+1 Byte, wobei $L < 2^{16}$)
- **MEDIUMTEXT** (L+1 Byte, wobei $L < 2^{24}$)
- und **LONGTEXT** (L+1 Byte, wobei $L < 2^{32}$)

- Analog hierzu **binäre Datentypen**
  - TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
  - BLOB: **B**inary **L**arge **Ob**ject

String Datentypen ENUM

Spalten mit fest vorgegebenen Werten: **Aufzählung**

- **ENUM(‘s1‘, ‘s2‘, ‘s3‘,..)** - Enumeration
  - Nur Werte aus der Liste und NULL werden akzeptiert
    sonst Fehlerwert → ‘‘ (keine Fehlermeldung!)
  - z.B. Spalte Geschlecht ENUM(‘w‘, ‘m‘),
    → mögliche Werte: NULL, ‘‘, ‘m‘, ‘w‘
- **SET(‘s1‘, ‘s2‘, ‘s3‘,..)** - Menge
  - Nur Werte aus der Liste und NULL werden akzeptiert
    sonst Fehlerwert → ‘‘ (keine Fehlermeldung!)
  - z.B. Spalte Führerscheinklasse SET(‘A‘, ‘B‘, ‘C‘, …),
    → mögliche Werte: NULL, ‘‘, ‘A‘,‘B‘,‘C‘,‘A,B‘,‘B,C‘,‘A,C‘, ‘A,B,C‘,…
  - **VORSICHT:** Verletzt ggf. 1. Normalform