

Informatik-GK

Q2 Datenbanken

Thomas Klein

April 2022



Inhaltsverzeichnis

1	Entitäten und Relationen	3
1.1	Daten und Informationen	3
1.2	Das ER-Modell	3
1.3	ER-Diagramme	4
1.4	Kardinalitäten und Optionalitäten	5
2	Das Relationenmodell	7
2.1	Relationen und Schlüssel	7
2.2	Überführen des ERD in das Relationenmodell	8
3	Abfragen mit SQL	10
3.1	Der SELECT-Befehl	10
3.2	Formeln und Aggregatfunktionen	11
3.3	Sortierung und Gruppierung	12
3.4	Joins	13
4	Relationale Algebra	15
4.1	Projektionen und Selektionen	15
4.2	Kreuzprodukt und Join	16
5	Datenschutz und Datensicherheit	18
5.1	Datenschutz	18
5.2	Datensicherheit	20



Vorwort und Übersicht

Eine Datenbank besteht aus einer Anzahl von Tabellen (sog. **Relationen**, die **Daten** zu bestimmten Themen enthalten. Mit Hilfe dieser Daten ist es möglich, **Informationen** zu generieren.

In diesem Halbjahr geht es zunächst darum, wie man eine Datenbank designt. Dazu erstellt man zunächst ein **Entity-Relationship-Diagramm**, das alle wesentlichen Informationen und Beziehungen enthält. Aus dem ER-Diagramm leitet man dann das sog. **Relationenmodell** ab, das angibt, welche Tabellen benötigt werden und welche Spaltenüberschriften verwendet werden müssen. Jede Tabelle/Relation benötigt einen **Primärschlüssel**, anhand dessen man die einzelnen **Datensätze** eindeutig identifizieren kann. Beziehungen zwischen Entitäten werden dabei über sog. **Fremdschlüssel** modelliert.

Damit ist das Datenbank-Design abgeschlossen und die anfangs leere Datenbank kann mit Daten gefüllt werden. Sind nun Daten vorhanden, so können mit Hilfe der Sprache **SQL** sog. **Abfragen** gegen die Datenbank getätigt werden, um Informationen zu erlangen. Mit **Aggregat-Funktionen**, Formeln und **Gruppierungen** können statistische Daten gewonnen werden, während es **Joins** erlauben, Informationen zu erhalten, die über mehrere Tabellen verteilt sind.

Weitere Themen der Q2 sind **Relationale Algebra**, **Webdatenbank-Projekt** und **Datenschutz und Datensicherheit**. Welches dieser Themen für das Landesabitur relevant ist, wird vom Kultusministerium per Erlass festgelegt.



1 Entitäten und Relationen

Eine Datenbank hat immer den Zweck, einen bestimmten Ausschnitt der realen Welt im Computer abzubilden. Dabei ist die Haupt-Aufgabe zu entscheiden, welche Daten für die Problemstellung relevant sind und welche unwichtig: Sowohl in der Kundendatenbank eines Autoverkäufers als auch in der Patientendatenbank eines Arztes sind Personen gespeichert. Aber in beiden Szenarien sind höchst unterschiedliche Daten relevant: Die Krankenversicherung spielt für den Autoverkäufer keine Rolle, während der Arzt sich nicht um die Konto-Verbindung des Patienten schert. Und für beide ist die Schuhgröße der Person im höchsten Maße irrelevant.

1.1 Daten und Informationen

Zunächst einmal müssen wir klären, was »Daten« (im Sinne der Informatik) eigentlich sind und welcher Zusammenhang zum Begriff »Information« besteht:

Information 1.1

- (a) Ein **Datum** (Mehrzahl: **Daten**) ist eine Zeichenkette. Da jede Zeichenkette als Binärzahl codiert werden kann, bestehen Daten also im Wesentlichen aus Nullen und Einsen.
- (b) Durch Hinzunehmen eines Kontextes kann ein Datum als **Information** interpretiert werden.

Beispiel 1.2

- (a) 183 ist ein Datum (als Bytes im ANSI-Code: 00110001 00111000 00110011). Wenn man weiß, dass es sich um eine Körpergröße in Zentimetern handelt, erhält man eine andere Information als wenn es sich um einen Kontostand handelt.
- (b) Auch R7B31 ist ein Datum. Im Kontext einer Bibliothek könnte sich dahinter die Information »Reihe 7, Buch Nummer 31« handeln.

#

Bemerkung 1.3 Es ist äußerst wichtig zu verstehen, dass pure Daten nutzlos werden, wenn man den Kontext nicht mehr weiß. Hinter der Bytefolge 00110001 00111000 00110011 aus dem Beispiel kann sich eine beliebige Information verbergen (z.B. der Anfang eines Musikstücks, die Hintergrundfarbe einer Website, ...).

#

1.2 Das ER-Modell

Datenbanken dienen dazu, Daten nicht nur zu speichern, sondern auch miteinander zu verknüpfen:



Beispiel 1.4 Zum Kunden »Max Mustermann« gehört die Kundennummer K34L1294, das Geburtsdatum 05.03.1992 und die Adresse »Hauptstr. 1439, 10115 Berlin«. Er ist mit der Kundin »Marie Mustermann« (Kundennummer K2744t103 geboren am 25.11.1994, wohnhaft an derselben Adresse) seit dem 11.11.2011 verheiratet. #

Offenbar haben wir es hier mit zwei »Dingen« (Max und Marie) zu tun, die einerseits bestimmte Eigenschaften haben (Kundennummer, Geburtsdatum, Adresse) und die miteinander in Beziehung stehen (verheiratet).

Die entsprechenden Fachbegriffe dazu lauten:

Information 1.5

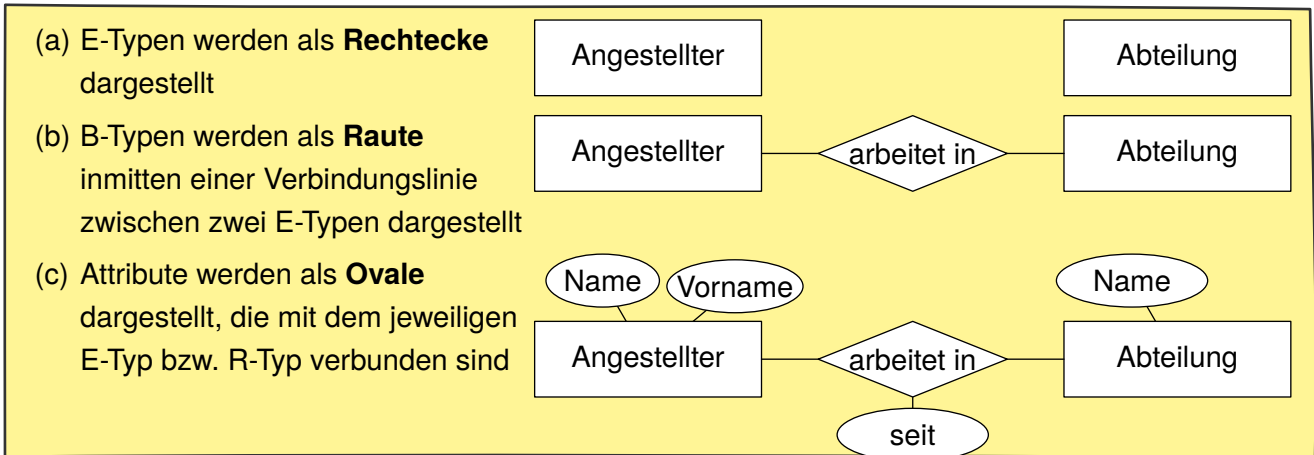
- (a) Eine **Entität** ist ein Objekt der realen Welt, z.B. eine Person, ein Gegenstand, eine Stadt aber auch ein nicht-materielles Ding wie eine Reservierung, ein Prozess, ein Flug von Rio nach Rom u.ä.
- (b) Gleichartige Entitäten bilden einen **Entitätstypen**, z.B. alle Angestellten, alle Flüge, alle Klassenräume usw. Ein Entitätstyp wird stets *im Singular* benannt, z.B. »Angestellter«, »Flug«, »Klassenraum«,
- (c) Eine **Beziehung** besteht zwischen zwei Entitäten, z.B. »Cora Müller arbeitet in der PR-Abteilung«.
- (d) Gleichartige Beziehungen bilden einen **Beziehungstyp** zwischen zwei Entitätstypen, z.B. »arbeitet in Abteilung« zwischen den Entitätstypen »Mitarbeiter« und »Abteilung«. Ein Beziehungstyp heißt **rekursiv**, wenn ein E-Typ mit sich selbst in Beziehung steht.
- (e) Ein **Attribut** ist eine Eigenschaft eines Entitäts- oder eines Beziehungstyps, die alle Entitäten bzw. Beziehungen dieses Typs aufweisen, z.B. »Vorname« des E-Typs »Mitarbeiter« oder »Datum« des B-Typs »hat bestellt«. Dabei können die konkreten Werte der jeweiligen Attribute für jede Entität bzw. jede Beziehung verschieden sein.

Die Werte der Attribute stellen die eigentlichen Daten dar, die E-Typen und B-Typen bilden die Verknüpfungen zwischen diesen Daten!

1.3 ER-Diagramme

Wenn man die Entitätstypen und Beziehungstypen für die Datenbank identifiziert hat, stellt man diese in einem sog. »Entity-Relationship-Diagramm« dar:

Information 1.6 Ein **Entity-Relationship-Diagramm** (kurz: **ER-Diagramm** oder **ERD**) ist eine grafische Darstellung der Entitäts- und Beziehungstypen sowie der zugehörigen Attribute.



1.4 Kardinalitäten und Optionalitäten

Eine Grund-Schwierigkeit beim Umgang mit Beziehungstypen liegt darin, dass man Beziehungen stets in *zwei Richtungen* lesen kann: »Der Wagen gehört der Person.« vs. »Die Person besitzt den Wagen«. Genau genommen hat man es jeweils mit zwei Beziehungen zu tun.

Für die Umsetzung eines Realitäts-Ausschnittes in eine Datenbank sind die sog. »Kardinalitäten« und »Optionalitäten« von entscheidender Bedeutung:

Information 1.7 Für jeden Beziehungstyp

gibt es zwei Lesarten:

(a) »Jeder A (muss/kann) in Beziehung R stehen mit (einem/beliebig vielen) B.«
 Im ERD:

(b) »Jeder B (muss/kann) in Beziehung R stehen mit (einem/beliebig vielen) A.«
 Im ERD:

In einem ERD werden beide Beschriftungen parallel verwendet:

Die Mengenangabe 1 bzw. n heißt **Kardinalität** und das Hilfsverb kann bzw. muss heißt **Optionalität**.

Je nach Kardinalitäten spricht man von einer **1:1-Beziehung**, einer **1:n-Beziehung** oder einer **n:m-Beziehung**.



Beispiel 1.8 In der Datenbank einer Bücherei werden Bücher verwaltet. Jedes Buch hat eine ISBN, einen Titel und wurde von genau einem Autor verfasst. Jeder Autor muss mindestens ein Buch geschrieben haben (sonst wäre er kein Autor). Über die Autoren werden außerdem noch Name, Vorname und Herkunftsland gespeichert. Jedes Buch wird außerdem einem gewisse Genre (Fantasy, Historisch, Krimi, ...) zugeordnet. Jedes Genre hat eine Bezeichnung und einen Beschreibungstext.

#



2 Das Relationenmodell

Das Entity-Relationship-Diagramm, das man nach Analyse der Problemstellung entwickelt, beinhaltet alle für die Problemstellung wesentlichen Informationen über die Beziehungen zwischen den Daten. Um dieses Diagramm im Computer abzubilden, überführt man es in ein sogenanntes »Relationenmodell«. Diese Relationen können dann als Tabellen im Computer abgespeichert werden.

2.1 Relationen und Schlüssel

Definition 2.1 Eine **Relation** ist eine Tabelle mit mindestens einer Spalte. Die Überschriften der Spalten heißen **Attribute** der Relation. Die Zeilen der Relation heißen **Datensätze**.

Relationen werden in der Form »Name(Attribut1, Attribut2, ...)« angegeben.

Beispiel 2.2 Die Relation `Kunde(Vorname, Nachname, KNr, Strasse, Plz)` könnte z.B. folgende Tabelle sein:

Kunde				
Vorname	Nachname	KNr	Strasse	Plz
Thomas	Klein	27435	Hauptstraße 943	11010
Max	Mustermann	50374	Goethestr. 8	65597
Marie	Mustermann	50374	Goethestr. 8	65597

#

Um diese Tabellen sinnvoll einsetzen zu können, muss man in der Lage sein, einzelne Datensätze auf eindeutige Weise auszuwählen:

Definition 2.3 Jede Relation benötigt einen **Primärschlüssel**. Ein Primärschlüssel besteht aus einem oder mehreren Attributen, die einen Datensatz eindeutig identifizieren. Die Attribute, die den Primärschlüssel bilden, werden unterstrichen (sowohl im ERD als auch im Relationenmodell).

Beispiel 2.4 Bei der Relation `Kunde` wäre das Paar (Vorname,Nachname) ein möglicher Schlüsselkandidat, allerdings nur, wenn man ausschließen kann, dass dieselbe Kombination niemals zweimal vorkommt. Daher ist es sinnvoller, die Kundennummer als Primärschlüssel zu verwenden.

Die Relation lautet daher `Kunde(Vorname, Nachname, KNr, Strasse, Plz)`

#

Bemerkung 2.5 In der Praxis verwendet man meistens eine eindeutige ID als Primärschlüssel. Z.B. erhält jeder Kunde eine Kundennummer, jeder Auftrag eine Auftragsnummer, jede Rechnung eine Rechnungsnummer.

#

Um die Beziehungen zwischen Entitäten abzubilden, kommen in Relationen auch häufig die Primärschlüssel von anderen Relationen vor (siehe nächster Abschnitt):



Definition 2.6 Ein Attribut einer Relation, das Primärschlüssel einer anderen Relation ist, heißt **Fremdschlüssel**. Fremdschlüssel werden im ERD nicht aufgeführt. Im Relationenmodell wird ihnen ein nach oben zeigender Pfeil (↑) vorangestellt.

Beispiel 2.7 Z.B. könnte die Relation Kunde um den Fremdschlüssel `verheiratetMit` ergänzt werden. In dieser Spalte würde dann die Kundennummer des Kunden stehen, der mit dem Kunden verheiratet ist:

Kunde					
Vorname	Nachname	<u>KNr</u>	Strasse	Plz	↑verheiratetMit
Thomas	Klein	27435	Hauptstraße 943	11010	
Max	Mustermann	50374	Goethestr. 8	65597	20745
Marie	Mustermann	20745	Goethestr. 8	65597	50374

#

2.2 Überführen des ERD in das Relationenmodell

Aus einem Entity-Relationship-Diagramm kann man relativ einfach das Relationenmodell ableiten:

Verfahren 2.8

Gegeben: ER-Diagramm

Gesucht: Relationenmodell

Lösung:

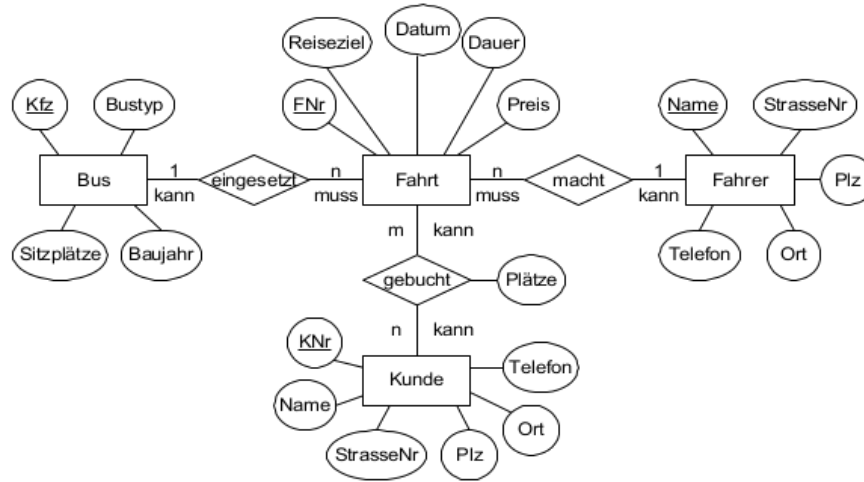
- (1) Entitätstypen: Jeder Entitätstyp bildet eine Relation. Die Attribute des E-Typen sind die Attribute der Relation.
- (2) 1:n-Beziehungstypen: Die Relation der n-Seite erhält den Primärschlüssel der anderen Relation als Fremdschlüssel. Außerdem erhält sie alle Attribute des Beziehungstyps.
- (3) 1:1-Beziehungstypen: Wie bei 1:n und es ist egal, welche Seite den Fremdschlüssel und die Beziehungs-Attribute erhält.
- (4) n:m-Beziehungstypen: Jeder n:m-Beziehungstyp bildet eine weitere Relation. Diese erhält die beiden Primärschlüssel der in Beziehung stehenden Entitätstypen als Fremdschlüssel. Die beiden Fremdschlüssel bilden gemeinsam den Primärschlüssel der neuen Relation. Die neue Relation erhält außerdem alle Attribute des Beziehungstyps.

Bemerkung 2.9 Bei der Überführung des ERD in das Relationenmodell spielen die Optionalitäten keine Rolle. Nur die Kardinalitäten sind entscheidend.

#



Beispiel 2.10 Wir überführen das folgende ER-Diagramm (Quelle: Glossar Landesabitur Informatik, 2017) in das Relationenmodell:



(1) Zunächst erhalten wir zu jedem Entitätstyp eine Relation:

- Bus(Kfz, Bustyp, Sitzplätze, Baujahr)
- Fahrt(FNr, Reiseziel, Datum, Dauer, Preis)
- Fahrer(Name, StrasseNr, Plz, Telefon, Ort)
- Kunde(KNr, Name, StrasseNr, Plz, Ort, Telefon)

(2) Jetzt bilden wir die 1:n-Relationen mittels Fremdschlüsseln ab:

- Fahrt(FNr, Reiseziel, Datum, Dauer, Preis, ↑BusKfz, ↑FahrerName)

(3) Zuletzt erhalten wir für den »gebucht«-Beziehungstyp eine neue Relation:

- Buchung(↑FNr, ↑KNr, Plätze)

Insgesamt erhalten wir also folgendes Relationenmodell:

- Bus(Kfz, Bustyp, Sitzplätze, Baujahr)
- Fahrt(FNr, Reiseziel, Datum, Dauer, Preis, ↑BusKfz, ↑FahrerName)
- Fahrer(Name, StrasseNr, Plz, Telefon, Ort)
- Kunde(KNr, Name, StrasseNr, Plz, Ort, Telefon)
- Buchung(↑FNr, ↑KNr, Plätze)

#



3 Abfragen mit SQL

Wir haben nun gesehen, wie man einen für die Informatik relevanten Teil der realen Welt als Entity-Relationship-Diagramm modelliert und dieses in ein Relationenmodell überführt, das die Tabellen liefert, aus denen die Datenbank letztendlich besteht.

Ist die Datenbank nun implementiert und mit Daten gefüllt, so besteht der nächste Schritt darin, Informationen aus der Datenbank zu extrahieren. Für relationale Datenbanken wurde zu diesem Zweck die Sprache SQL (»Structured Query Language«) entwickelt.

3.1 Der SELECT-Befehl

SQL bietet verschiedene Befehle an. Für uns ist nur der SELECT-Befehl relevant:

Information 3.1 Der **SELECT**-Befehl dient dazu, Daten aus einer Datenbank abzufragen. Der grundlegende Aufbau ist

```
1  SELECT spalte1, spalte2, ..., spalteN
2  FROM tabelle
3  WHERE bedingung1
4  AND/OR bedingung2
5  ...
6  AND/OR bedingungN
```

Dazu folgende Bemerkungen:

- Die Groß- und Kleinschreibung spielt dabei keine Rolle.
- Schreibt man `SELECT * FROM ...`, so werden alle Spalten der Tabelle ausgegeben.
- Strings werden in einfache Hochkommata geschrieben, z.B. `'Max Mustermann'`.

Information 3.2 Für die Bedingungen gibt es folgende Operatoren:

=	Gleich.
<>	Ungleich.
<	Kleiner als.
>	Größer als.



<code><=</code>	Kleiner oder gleich.
<code>>=</code>	Größer oder gleich.
<code>BETWEEN</code>	<code>WHERE Alter BETWEEN 13 and 19</code> wählt alle Datensätze aus, bei denen das Alter zwischen 13 und 19 liegt (jeweils einschließlich).
<code>LIKE</code>	<code>WHERE Name LIKE 'Max Muster%'</code> wählt alle Datensätze aus, bei denen der Name mit »Max Muster« anfängt und dann beliebig weitergeht, z.B. »Max Mustermann«, »Max Musterfrau«, etc.
<code>IN</code>	<code>WHERE Alter in (17, 19, 23, 29)</code> wählt alle Datensätze aus, bei denen das Alter einem der Werte in der Klammer entspricht.

3.2 Formeln und Aggregatfunktionen

Es ist möglich, mit ausgelesenen Werten zu rechnen. Dazu kann man die normalen Rechenoperatoren `+`, `-`, `*` und `/` verwenden.

Beispiel 3.3 Man kann eine Temperatur x in Grad Fahrenheit in eine Temperatur y in Grad Celsius umrechnen mit Hilfe der Formel

$$y = \frac{5}{9} \cdot (x - 5) - 15$$

Z.B. erhält man für $x = 14^\circ\text{F} \rightarrow \frac{5}{9} \cdot (14 - 5) - 15 = \frac{5}{9} \cdot 9 - 15 = -10^\circ\text{C}$.

Angenommen, wir haben eine Tabelle `Wetter`, in der Temperaturen in Fahrenheit stehen (in der Spalte `Temperatur`), dann erhalten wir folgendermaßen die Temperatur in Celsius:

```
1 SELECT 5*(Temperatur-5)/9-15
2 FROM Wetter
```

#

Information 3.4 Ausgelesene Werte können verarbeitet und miteinander kombiniert werden, um neue Informationen zu generieren.



- (a) Zugelassen sind Formeln mit den Grundrechenarten und Klammern.
- (b) Zugelassen sind außerdem die folgenden **Aggregatfunktionen**: `AVG` (Durchschnitt aller Werte), `COUNT` (Anzahl), `MAX` (Maximaler Wert), `MIN` (minimaler Wert), `SUM` (Summe aller Werte). Diese können auch beliebig kombiniert werden.
- (c) Ergebnisspalten können mit dem Schlüsselwort `AS` umbenannt werden.

Beispiel 3.5 Der folgende Code liefert das BIP pro Kopf und gibt es in einer Tabelle der Form (Land, Pro-Kopf-BIP) an:

```
1 SELECT Name, bip/einwohner AS 'Pro-Kopf-BIP'
2 FROM cia
```

#

Bemerkung 3.6 Viele Aggregat-Funktionen machen erst im Zusammenhang mit Gruppierungen Sinn (siehe nächster Abschnitt). #

3.3 Sortierung und Gruppierung

Um die Ausgabe übersichtlicher zu gestalten, kann man sich die Werte sortiert ausgeben lassen:

Information 3.7 Mit dem Schlüsselwort `ORDER BY` kann man Daten aufsteigend (`ASC`, Standard) oder absteigend (`DESC`) sortieren. Es können mehrere Spalten angegeben werden, die dann nacheinander als Sortierkriterium verwendet werden, d.h., wenn der Wert der ersten Spalte gleich ist, wird der Wert der zweiten Spalte verglichen usw.

Beispiel 3.8

#

Ein sehr mächtiges Werkzeug stellt das Gruppieren dar:

Information 3.9 Mit dem Schlüsselwort `GROUP BY` kann man Daten nach bestimmten Attributen gruppieren. Das bedeutet, dass alle Datensätze, bei denen diese Attribute den gleichen Wert haben, als eine Gruppe behandelt werden. Aggregatfunktionen werden auf jede Gruppe getrennt angewendet.

Beispiel 3.10 Gesucht sind die jeweiligen Durchschnittsgehälter der Spieler der einzelnen Bundesliga-Mannschaften:

```

1  SELECT Mannschaft, AVG (Gehalt)
2  FROM Spieler
3  GROUP BY Mannschaft
    
```

#

Beispiel 3.11 Es ist nicht möglich, auf Attribute zuzugreifen, die innerhalb einer Gruppe verschiedene Werte haben. Zum Beispiel ist die Anweisung

```

1  SELECT Name, Mannschaft, AVG (Gehalt)
2  FROM Spieler
3  GROUP BY Mannschaft
    
```

nicht möglich, weil innerhalb der Mannschafts-Gruppen die Spieler verschiedene Namen haben. #

3.4 Joins

Joins (»Verbindungen«) kommen immer dann ins Spiel, wenn man die Daten aus mehreren Tabellen miteinander kombinieren möchte.

Information 3.12 Man kann in einem SELECT-Befehl mehrere Tabellen aufführen (durch Komma getrennt). Diese Tabellen werden kombiniert in dem Sinne, das alle Datensätze der einen Tabelle mit allen Datensätzen der anderen Tabelle kombiniert werden.

Die neue Tabelle hat die Spaltennamen der Ursprungstabellen. Bei gleichen Attributnamen kann man Tabellename.Attribut schreiben.

Beispiel 3.13 Für die beiden Tabellen (die Gehälter sind erfunden)

Team		
Name	Stadt	TrainerName
Bayern München	München	Carlo Ancelotti
Eintracht Frankfurt	Frankfurt	Niko Kovac

Trainer	
Name	Gehalt
Niko Kovac	200000
Manuel Baum	100000
Carlo Ancelotti	1500000

ergibt die Abfrage

```

1  SELECT *
2  FROM Team, Trainer
    
```

die folgende Ausgabe:

Team, Trainer				
Team.Name	Stadt	TrainerName	Trainer.Name	Gehalt
Bayern München	München	Carlo Ancelotti	Niko Kovac	200000
Bayern München	München	Carlo Ancelotti	Manuel Baum	100000
Bayern München	München	Carlo Ancelotti	Carlo Ancelotti	1500000
Eintracht Frankfurt	Frankfurt	Niko Kovac	Niko Kovac	200000
Eintracht Frankfurt	Frankfurt	Niko Kovac	Manuel Baum	100000
Eintracht Frankfurt	Frankfurt	Niko Kovac	Carlo Ancelotti	1500000

#

Dieses Kombinieren aller Datensätze miteinander ergibt natürlich in den meisten Fällen keinen Sinn. Stattdessen führt man einen »Join« durch:

Information 3.14 Ein **Join** kombiniert zwei oder mehr Tabellen und legt dabei fest, welche Attribute gleich sein müssen, damit die Datensätze kombiniert werden. Ein **Natural Join** ist einer, bei dem die Namen der Attribute gleich sind.

Beispiel 3.15 Wir wollen bei obigem Beispiel eine Tabelle aller Mannschaften mit dem jeweiligen Trainer und dem Gehalt des Trainers erhalten. Wir benötigen also einen Join über die Attribute `TrainerName` aus der Tabelle `Team` und `Name` aus der Tabelle `Trainer`:

```

1  SELECT Team.Name, TrainerName, Gehalt
2  FROM Team, Trainer
3  WHERE Team.TrainerName=Trainer.Name
    
```

ergibt

Team, Trainer		
Team.Name	TrainerName	Gehalt
Bayern München	Carlo Ancelotti	1500000
Eintracht Frankfurt	Niko Kovac	200000

#

Bemerkung 3.16 Dieses Kapitel gibt nur einen knappen Überblick über die für das Landesabitur relevanten SQL-Sprachkonstrukte. Genau wie bei der Programmierung ist es wichtig, diese Konzepte anzuwenden, um Aufgaben damit zu lösen. Es gibt im Internet eine ganze Reihe hervorragender Übungsseiten für SQL. Suchen Sie bspw. nach »sqlzoo« oder »sql imoodle«.

#



4 Relationale Algebra

Datenbanken können mit Hilfe von SQL-Befehlen abgefragt werden. Wir haben gesehen, dass ein solcher `SELECT`-Befehl relativ lange und kompliziert werden kann.

Um Beziehungen zwischen Daten zu untersuchen, ist es sinnvoll, eine alternative Beschreibungssprache für Abfragen zu finden. Diese soll aus möglichst einfachen Grundkomponenten bestehen und trotzdem mächtig genug sein, alle Arten von Abfragen darzustellen. Diese Aufgabe erfüllt die sog. *relationale Algebra* par excellence.

Alle Beispiele in diesem Kapitel beziehen sich auf die Relationen in **Abb. 1**.

Klassen			Lehrer			Schueler			
ID	Klassenlehrer	Raum	ID	Nachname	Vorname	ID	Nachname	Vorname	Klasse
5e	MU	13	MU	Müller	Andreas	107	Mann	Julia	7b
7a	HA	21	HA	Hansen	Doris	211	Braun	Zoe	5e
7b	GR	25	GR	Gros	Johannes	143	Kremer	Jan	5e
			TI	Tillisch	Petra				

Abbildung 1 Alle verwendeten Relationen dieses Kapitels

4.1 Projektionen und Selektionen

Zunächst machen wir eine relativ abstrakte Definition:

Definition 4.1 Eine **Abbildung** f bildet jedes Element a einer Menge A eindeutig auf ein Element $f(a)$ einer Menge B ab.

Beispiel 4.2 Wenn $A = \mathbb{R}$ und $B = \mathbb{R}$, dann sind die Abbildungen von A nach B klassische Funktionen aus dem Mathematik-Unterricht. #

Wir interessieren uns nun für Abbildungen, die Relationen auf Relationen abbilden:

Definition 4.3

(a) Die **Projektion** $\pi_{a_1, a_2, \dots, a_n}$ bildet eine Relation r auf eine neue Relation r' ab, die nur noch die Attribute a_1, a_2, \dots, a_n enthält. Alle anderen Attribute werden entfernt.

(b) Die **Selektion** $\sigma_{\text{Bedingung}}$ bildet eine Relation r auf eine neue Relation r' ab, die nur noch die Zeilen enthält, die die *Bedingung* erfüllen. Alle anderen Zeilen werden entfernt.

Beispiel 4.4

(a) $\pi_{ID, Vorname}(Lehrer)$ ergibt die Relation

$\pi_{ID, Vorname}(Lehrer)$	
ID	Vorname
MU	Andreas
HA	Doris
GR	Johannes
TI	Petra

(b) $\sigma_{Klasse='5e'}(Schueler)$ ergibt die Relation

$\sigma_{Klasse='5e'}(Schueler)$			
ID	Nachname	Vorname	Klasse
211	Braun	Zoe	5e
143	Kremer	Jan	5e

(c) $\pi_{Nachname}(\sigma_{Klasse='5e'}(Schueler))$ ergibt die Relation

$\pi_{Nachname}(\sigma_{Klasse='5e'}(Schueler))$
Nachname
Braun
Kremer

#

Bemerkung 4.5 Mit Hilfe von Projektionen und Selektionen lassen sich also einfache SELECT-Anweisungen der Form

```

1  SELECT attribut1, attribut2, ..., attributN
2  FROM tabelle
3  WHERE bedingung
    
```

nachbauen.

#

4.2 Kreuzprodukt und Join

Definition 4.6 Für zwei Relationen R und S definieren wir:



(a) Das **kartesische Produkt** $R \times S$ ist die Relation, bei der alle Zeilen von R mit allen Zeilen von S kombiniert werden.

(b) Der **Join** $R \bowtie_{a=b} S$ ist die Relation, bei der alle Zeilen von R mit allen Zeilen von S kombiniert werden, bei denen die Bedingung erfüllt ist, dass der Wert des Attributs a von R identisch ist mit dem Wert des Attributs b von S . Haben die beiden Attribute denselben Namen, so handelt es sich um einen **natural join**, geschrieben $R \bowtie S$.

Beispiel 4.7

(a) $Klassen \times Schueler$ ergibt die (ziemlich unsinnige) Relation

Klassen \times Schueler						
ID	Klassenlehrer	Raum	Schueler.ID	Nachname	Vorname	Klasse
5e	MU	13	107	Mann	Julia	7b
5e	MU	13	211	Braun	Zoe	5e
5e	MU	13	143	Kremer	Jan	5e
7a	HA	21	107	Mann	Julia	7b
7a	HA	21	211	Braun	Zoe	5e
7a	HA	21	143	Kremer	Jan	5e
7b	GR	25	107	Mann	Julia	7b
7b	GR	25	211	Braun	Zoe	5e
7b	GR	25	143	Kremer	Jan	5e

(b) Deutlich sinnvoller ist $Klassen \bowtie_{ID=Klasse} Schueler$:

Klassen $\bowtie_{ID=Klasse}$ Schueler						
ID	Klassenlehrer	Raum	Schueler.ID	Nachname	Vorname	Klasse
5e	MU	13	211	Braun	Zoe	5e
5e	MU	13	143	Kremer	Jan	5e
7b	GR	25	107	Mann	Julia	7b

#



5 Datenschutz und Datensicherheit

»Big Data«, »maschinelles Lernen«, »Google Analytics«, »Alexa« und »künstliche Intelligenz«: Die Verfügbarkeit vieler Daten erlaubt es, Informationen aller Art zu gewinnen, zu gebrauchen aber auch zu missbrauchen.

In diesem Kapitel geht es einerseits darum, welche Gesetze verhindern, dass persönliche Daten beliebig gespeichert und verarbeitet werden (Datenschutz) und andererseits, mit welchen technischen Hilfsmitteln Datendiebstahl verhindert werden kann (Datensicherheit).

5.1 Datenschutz

Definition 5.1 *Der Begriff **Datenschutz** ist nicht einheitlich definiert. Darum hier einige Möglichkeiten:*

(a) *»Zweck dieses Gesetzes ist es, den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.«*

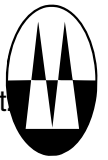
(§1 (1) Bundesdatenschutzgesetz (BDSG))

(b) *»Diese Verordnung enthält Vorschriften zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Verkehr solcher Daten.«*

(§1 (1) Europäische Datenschutz-Grundverordnung (EUDSGVO))

Insgesamt kann man festhalten:

- *Datenschutz bezieht sich immer auf natürliche, einzelne Personen, also weder auf Firmen, noch Vereine o.ä. Geschützt wird also jeder Einzelne von uns!*
- *Datenschutz schränkt die Verarbeitung sog. »personenbezogener Daten« ein:
»Personenbezogene Daten sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbarer natürlichen Person (Betroffener).« (§3 (1) BDSG)*
- *Das BDSG unterscheidet zusätzlich noch **besondere Arten personenbezogener Daten**:
»Besondere Arten personenbezogener Daten sind Angaben über die rassische und ethnische Herkunft, politische Meinungen, religiöse oder philosophische Überzeugungen, Gewerkschaftszugehörigkeit, Gesundheit oder Sexualleben.« (§3 (9) BDSG)
Diese besonderen Arten personenbezogener Daten werden besonders geschützt (§28 (6)-(9) BDSG).*



- Unter **Verarbeitung** versteht man »Speichern, Verändern, Übermitteln, Sperren und Löschen personenbezogener Daten« (§3 (4) BDSG).
- Datenschutz betrifft nur automatisiertes Erheben, Verarbeiten und Nutzen von Daten. Handschriftliche Aufzeichnungen, Ausdrücke etc. fallen also nicht unter das Datenschutzgesetz.

Im folgenden beziehen wir uns auf die Paragraphen des BDGS. Es geht darum, einen Überblick über die wichtigsten Paragraphen zu geben — ohne Anspruch auf Vollständigkeit. Um einen konkreten Fall zu prüfen, muss man immer das BDSG konsultieren.

Information 5.2 §3a: Datenvermeidung und Datensparsamkeit

»Die Erhebung, Verarbeitung und Nutzung personenbezogener Daten und die Auswahl und Gestaltung von Datenverarbeitungssystemen sind an dem Ziel auszurichten, so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten oder zu nutzen. Insbesondere sind personenbezogene Daten zu anonymisieren oder zu pseudonymisieren, soweit dies nach dem Verwendungszweck möglich ist und keinen im Verhältnis zu dem angestrebten Schutzzweck unverhältnismäßigen Aufwand erfordert.«

(§3a BDSG)

Verkürzt: Unternehmen dürfen nur Daten erheben/verarbeiten, die für den unmittelbaren Zweck notwendig sind — wenn möglich anonymisiert.

Information 5.3 §6/6a: Rechte des Betroffenen

»Die Rechte des Betroffenen auf Auskunft (§§ 19, 34) und auf Berichtigung, Löschung oder Sperrung (§§ 20, 35) können nicht durch Rechtsgeschäft ausgeschlossen oder beschränkt werden.«

(§6 (1) BDSG)

»Entscheidungen, die für den Betroffenen eine rechtliche Folge nach sich ziehen oder ihn erheblich beeinträchtigen, dürfen nicht ausschließlich auf eine automatisierte Verarbeitung personenbezogener Daten gestützt werden, die der Bewertung einzelner Persönlichkeitsmerkmale dienen. Eine ausschließlich auf eine automatisierte Verarbeitung gestützte Entscheidung liegt insbesondere dann vor, wenn keine inhaltliche Bewertung und darauf gestützte Entscheidung durch eine natürliche Person stattgefunden hat.«

(§6a (1) BDSG)

Verkürzt: Vollautomatische Entscheidungen sind rechtlich als sehr kritisch zu bewerten. Im Einzelfall muss (gerichtlich) geklärt werden, was »erheblich« bedeutet.

Information 5.4 §28: Datenerhebung und -speicherung für eigene Geschäftszwecke Dieser Paragraph ist viel zu umfangreich, um ihn vollständig zu besprechen. Für viele Anwendungsfälle ist er der relevante Paragraph!

»Das Erheben, Speichern, Verändern oder Übermitteln personenbezogener Daten oder ihre Nutzung als Mittel für die Erfüllung eigener Geschäftszwecke ist zulässig



- (1) wenn es für die Begründung, Durchführung oder Beendigung eines rechtsgeschäftlichen oder rechtsgeschäftsähnlichen Schuldverhältnisses mit dem Betroffenen erforderlich ist,
- (2) soweit es zur Wahrung berechtigter Interessen der verantwortlichen Stelle erforderlich ist und kein Grund zu der Annahme besteht, dass das schutzwürdige Interesse des Betroffenen an dem Ausschluss der Verarbeitung oder Nutzung überwiegt, oder
- (3) wenn die Daten allgemein zugänglich sind oder die verantwortliche Stelle sie veröffentlichen dürfte, es sei denn, dass das schutzwürdige Interesse des Betroffenen an dem Ausschluss der Verarbeitung oder Nutzung gegenüber dem berechtigten Interesse der verantwortlichen Stelle offensichtlich überwiegt.

Bei der Erhebung personenbezogener Daten sind die Zwecke, für die die Daten verarbeitet oder genutzt werden sollen, konkret festzulegen.«

(§28 (1) BDSG)

Verkürzt: Datenverarbeitung ist grundsätzlich erlaubt, wenn die Daten für die Erfüllung der eigenen Geschäftszwecke notwendig sind.

»Die Verarbeitung oder Nutzung personenbezogener Daten für Zwecke des Adresshandels oder der Werbung ist zulässig, soweit der Betroffene eingewilligt hat und im Falle einer nicht schriftlich erteilten Einwilligung die verantwortliche Stelle nach Absatz 3a verfährt.«

(§28 (3) BDSG)

Verkürzt: Der Betroffene muss zustimmen, wenn die Daten für andere Zwecke verwendet werden sollen.

Information 5.5 §33/34: Bußgeld- und Strafvorschriften In diesen Paragraphen wird geregelt, unter welchen Umständen Datenschutzverstöße eine Ordnungswidrigkeit (Folge: Bußgeld) darstellen und unter welchen Umständen eine Straftat (Folge: Geld- oder Freiheitsstrafe).

5.2 Datensicherheit