Einführung in Datenbanken

Übung 15: Logischer Entwurf, Relationale Normalformen

Prof. Dr. Stefan Brass
PD Dr. Alexander Hinneburg
Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2023/24

http://www.informatik.uni-halle.de/~brass/db23/

Inhalt

1. Organisatorisches
2. Präsenzaufgabe 10
3. Aufgabe 14.1
4. Aufgabe 14.2

Haben Sie Fragen/Anmerkungen?

- Falls Sie noch Fragen haben, stellen Sie diese über das Forum in StudIP.

  Ich habe dort den Bereich „Fragen nach Ende der Vorlesung“ eingerichtet.

- Fragen, die nur Sie betreffen, können Sie per EMail schicken.

  Wenn Sie keine Antwort bekommen, schicken Sie sie nach einigen Tagen nochmals.

- Falls Sie Fehler in meinen Skript finden, schicken Sie mir unbedingt eine Email.

  Selbst wenn es nur leichte Tippfehler sind. Auch relativ präzise und möglichst konstruktive Verbesserungsvorschläge würden mich interessieren.

- Wenn Sie eindeutig nützliche Quellen im Internet finden, schicken Sie mir bitte auch eine EMail mit dem Link.

  Ich möchte die Linkliste auf meiner Webseite weiter verbessern.

Hinweise zur Klausur (1)

- Der Ersttermin der Klausur ist am Donnerstag, den 07.03.2024, 10:00–12:00, im Prüfungszentrum der Universität.

  Sie sollten 15 Minuten früher da sein. Es ist geplant, dass die Klausur 10:00 beginnt und 120 Minuten dauert.
  Der Zweittermin ist am 08.07.2024, 10:00–12:00.

- Beim Ersttermin dürfen Sie 10 Seiten (DIN A4) mit Notizen mitbringen (beim Zweittermin 6 Seiten).

  10 Seiten entsprechen 5 Blättern mit Vorder- und Rückseite beschriftet.
  Es sind aber auch 10 Blätter einseitig beschrieben/bedruckt erlaubt.
  Die Notizen müssen auf Papier vorliegen. Die Benutzung von Tabletts und anderen elektronischen Geräten ist nicht erlaubt. Alle Blätter, auf die Sie während der Klausur Notizen machen, dürfen den Raum nicht verlassen (Sie müssen Sie am Ende abgeben).

Hinweise zur Klausur (2)

- Es sind keine Smart-Watches oder andere Geräte erlaubt, die zur Kommunikation verwendet werden könnten.

  Handys müssen ausgeschaltet oder mindestens stumm geschaltet in Ihrer Tasche sein, auf die Sie während der Klausur bitte nicht zugreifen (notfalls rufen Sie eine Aufsichtsperson zur Kontrolle). Manche anderen Dozenten wollen, dass das Handy ausgeschaltet auf dem Tisch liegt. Die Regeln sind also dozenten-abhängig. Ich möchte, dass es möglichst überhaupt nicht zugreifbar ist.

- Selbstverständlich sind auch alle Formen von Kameras und Ohrhörern verboten.

  Wenn die Aufsicht es als nötig empfindet, müssen Sie eventuelle Mützen kurz abnehmen, damit man sehen kann, dass Sie keinen Ohrhörer haben.

Hinweise zur Klausur (3)

Datenschutz-Hinweis:

- Alle Zugriffe auf Adminer und ILIAS werden mitgeloggt und können für die Analyse auf Täuschungsversuche verwendet werden.

  Und für Forschungszwecke, werden aber selbstverständlich nur in anonymisierter Form veröffentlicht (wenn überhaupt).

Inhalt

1. Organisatorisches
2. Präsenzaufgabe 10
3. Aufgabe 14.1
4. Aufgabe 14.2

Präsenzaufgabe 10: Logischer Entwurf

- Übersetzen Sie folgendes ER-Diagramm in das relationale Modell:

- Geben Sie das Schema in der Kurznotation (ASCII-Version) ab, z.B.: `STUDENTEN(#SID, VORNAME, NACHNAME, EMAIL?)`
- Falls zusätzliche Bedingungen überwacht werden müssen, um die Äquivalenz zum ER-Diagramm sicherzustellen, schreiben Sie eine SQL-Anfrage, die Fehlermeldungen berechnet.

Lösung der Präsenzaufgabe (1)

- Gegebenes ER-Diagramm:

- Man legt zuerst für jeden Entity-Typ eine Tabelle an und überträgt dabei die Attribute, die Schlüssel und die Festlegung, ob Nullwerte erlaubt sind:

A(#B, #C, D)	A(B, C, D)
E(#F, G, H?)	E(F, G, H°)

Lösung der Präsenzaufgabe (2)

- Gegebenes ER-Diagramm:

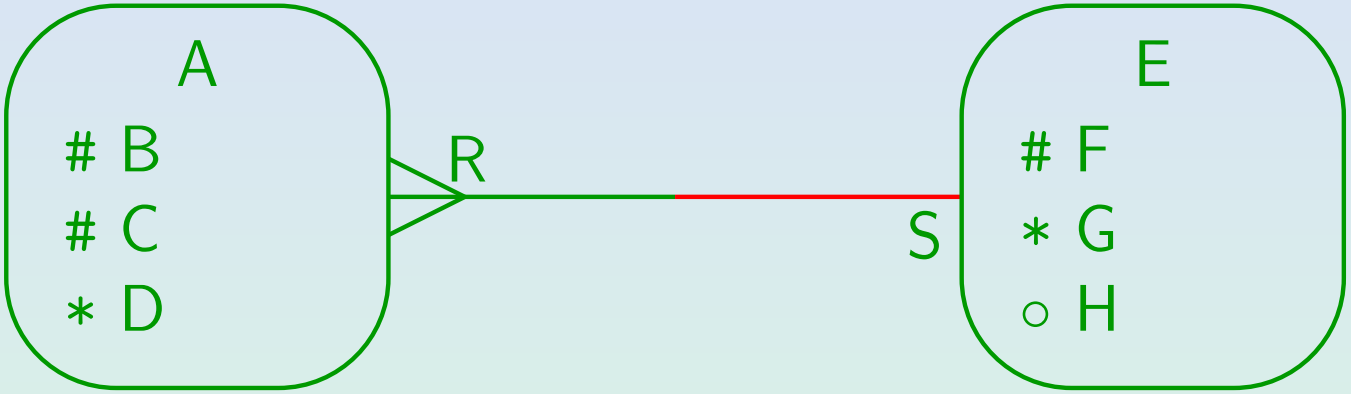
- Für die eins-zu-viele Beziehung fügt man den Schlüssel der „eins“-Seite als Fremdschlüssel zur „viele“-Seite hinzu:

```
E(#F, G, H?)                E(F, G, H°)
A(#B, #C, D, F->E)          A(B, C, D, F→E)
```

Falls möglich, sollte man die Relationen so anordnen, dass Fremdschlüssel auf vorher definierte Relationen verweisen. Der Fremdschlüssel könnte auch anders heißen, z.B. `R` oder `E_F`. Man sollte aber auf Konsistenz achten.

Lösung der Präsenzaufgabe (3)

- Gegebenes ER-Diagramm:



- Die Übersetzung stellt sicher, dass jedes A-Entity an der Beziehung teilnimmt (der Fremdschlüssel ist nicht Null).
- Die Übersetzung stellt nicht sicher, dass jedes E-Entity auch an der Beziehung teilnimmt.

  Also von einem Fremdschlüssel-Wert referenziert wird.

Lösung der Präsenzaufgabe (4)

- Zusätzlich nötige Integritätsbedingung: „Jeder Wert für den Schlüssel F in der Tabelle E taucht auch in der Tabelle A als Fremdschlüssel-Wert auf.“
- SQL-Anfrage zur Berechnung von Fehlermeldungen bei Verletzung der Integritätsbedingung:

```
SELECT 'E-Entity ohne A-Entity: ' || X.F
FROM   E X
WHERE  NOT EXISTS(SELECT *
                  FROM   A Y
                  WHERE  Y.F = X.F)
```

Wenn das Beispiel nicht so abstrakt wäre, könnte man die Fehlermeldung etwas aussagekräftiger gestalten.

Lösung der Präsenzaufgabe (5)

- Alternative Lösung mit `NOT IN`:

```
SELECT 'E-Entity ohne A-Entity: ' || F
FROM   E
WHERE  F NOT IN (SELECT F
                 FROM   A)
```

Wie bekannt, muss man sich bei `NOT IN`-Anfragen versichern, dass die Unteranfrage keinen Nullwert liefert. Das ist hier gegeben (der Fremdschlüssel ist `NOT NULL`). Wäre die Linie auf der `A`-Seite dagegen gestrichelt, wären Nullwerte für den Fremdschlüssel erlaubt, und man müsste sie in der Unteranfrage explizit ausschließen. Bei der Variante mit `NOT EXISTS` gibt es dieses Problem nicht.

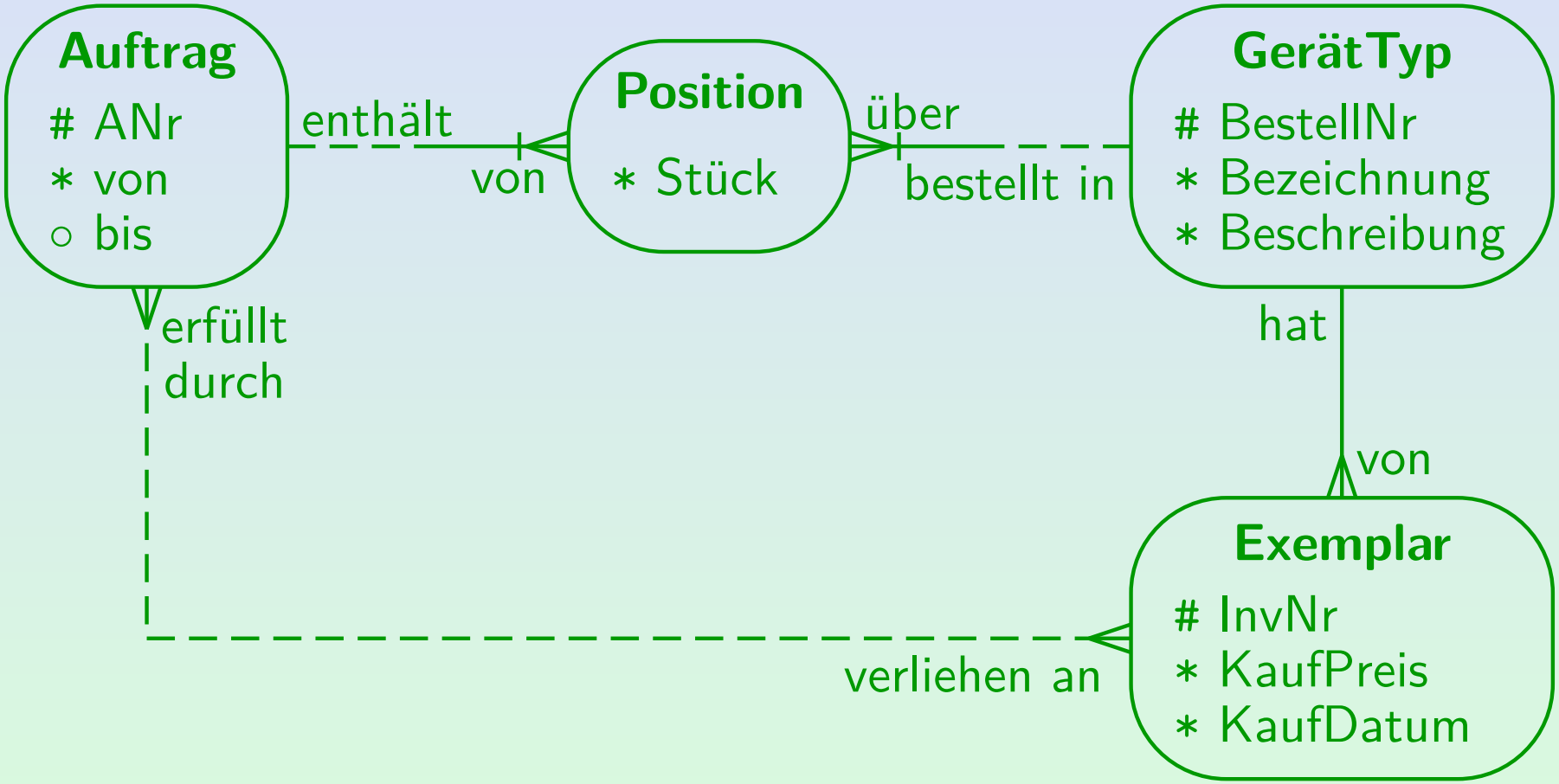
Inhalt

1. Organisatorisches
2. Präsenzaufgabe 10
3. Aufgabe 14.1
4. Aufgabe 14.2

Hausaufgabe 14.1: Logischer Entwurf (1)

- Übersetzen Sie das unten stehende ER-Diagramm in ein relationales Schema.
- Nutzen Sie die Kurznotation aus der Vorlesung für das relationale Schema,
  - entweder in der Variante mit dem Unterstreichen von Primärschlüssel-Attributen oder in der reinen ASCII-Variante (mit vorangestelltem # für Primärschlüssel-Attribute).
  - Markieren Sie auch Fremdschlüssel (inklusive der referenzierten Tabelle)
  - und Attribute, die Nullwerte erlauben (Attribut gefolgt von kleinem hochgestellten „o“ oder „?“).
- Bei Bedarf geben Sie auch weitere Integritätsbedingungen an, die notwendig sind, um die Äquivalenz des relationalen Schemas zum ER-Schema sicherzustellen.

Hausaufgabe 14.1: Logischer Entwurf (2)



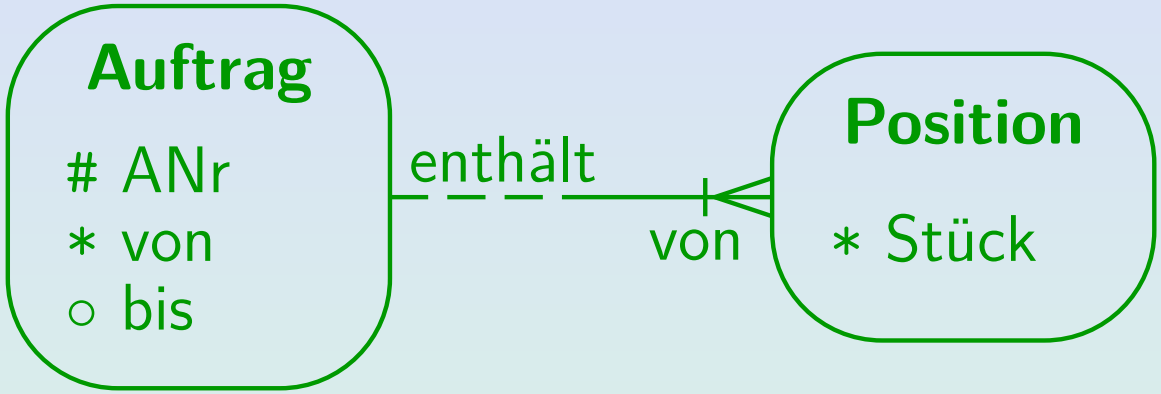
Hausaufgabe 14.1: Logischer Entwurf (3)

- Zunächst legt man für jeden Entity-Typ eine Tabelle an, und übernimmt die Attribute, Schlüssel, Optionalität:
  - Auftrag(ANr, von, bis°)
  - Position(Stück)
  - GerätTyp(BestellNr, Bezeichnung, Beschreibung)
  - Exemplar(InvNr, KaufPreis, KaufDatum)
- Man kann auch die Plural-Form der Entity-Typen als Tabellennamen nehmen.

  Aber natürlich konsistent: Entweder für alle Tabellen, oder für keine.
- Zweiter Schritt sind Fremdschlüssel für 1:n-Beziehungen, wobei die Fremdschlüssel für identifizierende Beziehungen zu schwachen Entities gleichzeitig Teil des Schlüssels werden.

Hausaufgabe 14.1: Logischer Entwurf (4)

- Beispiel für 1:n-Beziehung, die auch identifizierend ist:

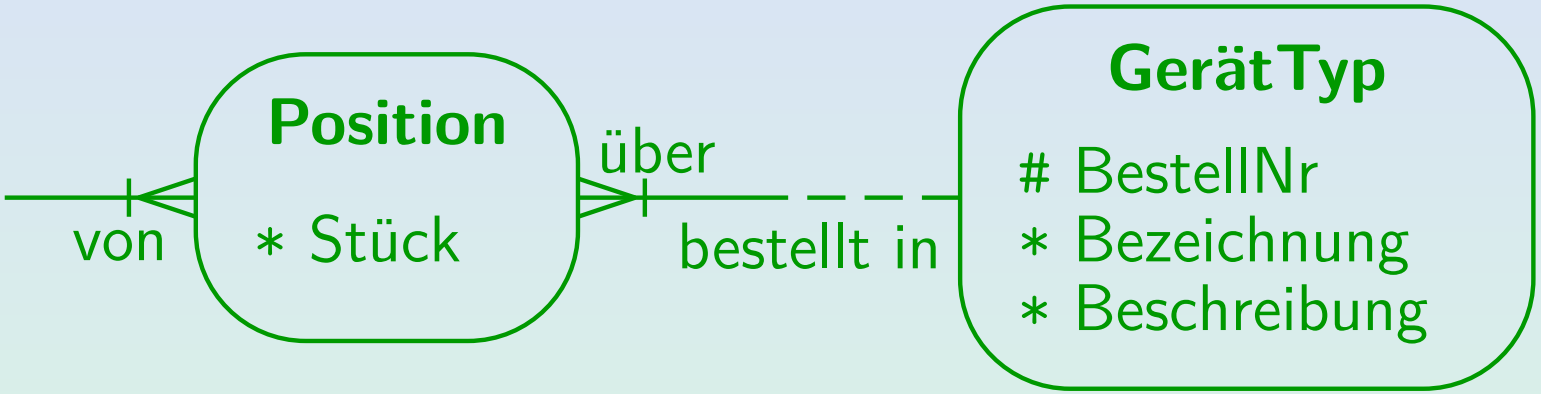


  - Der Schlüssel der Eins-Seite (Auftrag) wird als Fremdschlüssel in die viele Seite (Position) eingefügt, um die Eins-zu-viele-Beziehung zu implementieren.
  - Da die Beziehung identifizierend ist (`Position` ist schwaches Entity), wird der Fremdschlüssel Teil des Primärschlüssels.
  - `Position(`<u>`ANr`</u> $\rightarrow$ `Auftrag, Stück)`

    Man könnte den Fremdschlüssel auch „`von`“ nennen.

Hausaufgabe 14.1: Logischer Entwurf (5)

- `Position` hat zwei identifizierende Beziehungen (es ist ein „doppelt schwaches Entity“, d.h. „Association Entity“):

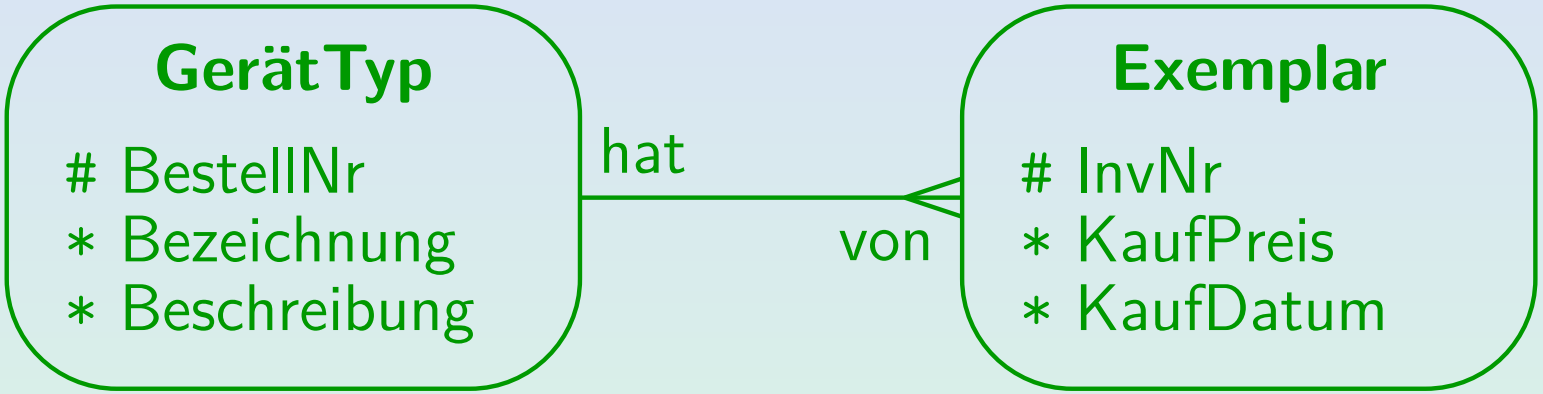


  - Also kommt auch der Schlüssel von `GerätTyp` als Fremdschlüssel in die Tabelle `Position` und wird dort auch Teil des Schlüssels.
  - `Position(`<u>`ANr`</u> $\rightarrow$ `Auftrag,` <u>`BestellNr`</u> $\rightarrow$ `GerätTyp, Stück)`

    Üblicherweise schreibt man Primärschlüsselattribute ganz vorne in der Liste der Attribute.

Hausaufgabe 14.1: Logischer Entwurf (6)

- Zwischen `GerätTyp` und `Exemplar` gibt es eine normale 1:n-Beziehung:



  - Also kommt auch der Schlüssel von `GerätTyp` als Fremdschlüssel in die Tabelle `Exemplar`.
  - Weil die Beziehung nicht identifizierend ist, wird er nicht Teil des Schlüssels.
  - `Exemplar(`<u>`InvNr`</u>`, KaufPreis, KaufDatum, BestellNr` $\rightarrow$ `GerätTyp)`

Hausaufgabe 14.1: Logischer Entwurf (7)

- Welche der vier möglichen 1:n-Beziehungen brauchen eine zusätzliche Integritätsbedingung?

A. 1 und 2

B. 3 und 4

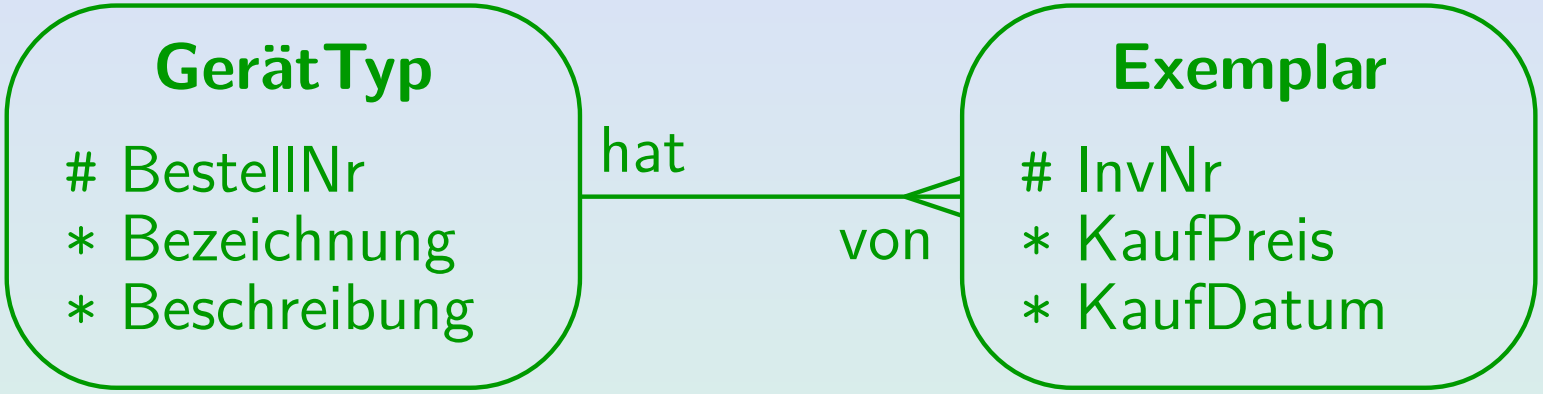
C. 1 und 3

D. 2 und 4

E. keine

Hausaufgabe 14.1: Logischer Entwurf (8)

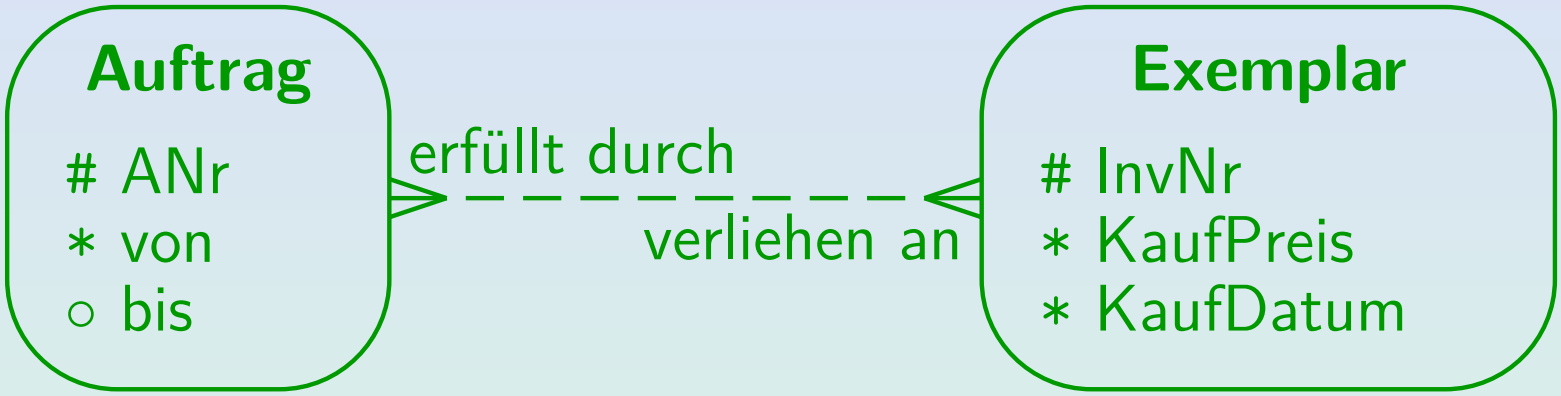
- 1:n-Beziehung mit verpflichtender Teilnahme auf beiden Seiten:



  - `Exemplar(InvNr, KaufPreis, KaufDatum, BestellNr → GerätTyp)`
  - Der Fremdschlüssel stellt sicher, dass es zu jedem `Exemplar` einen `GerätTyp` gibt. Aber es wäre möglich, dass es einen `GerätTyp` ohne `Exemplar` gibt.
  - Also ist eine zusätzliche Integritätsbedingung notwendig: „Zu jeder Zeile in `GerätTyp` muss es mindestens eine Zeile in `Exemplar` geben mit der gleichen `BestellNr`.“

Hausaufgabe 14.1: Logischer Entwurf (9)

- Es fehlt noch die viele-zu-viele Beziehung:

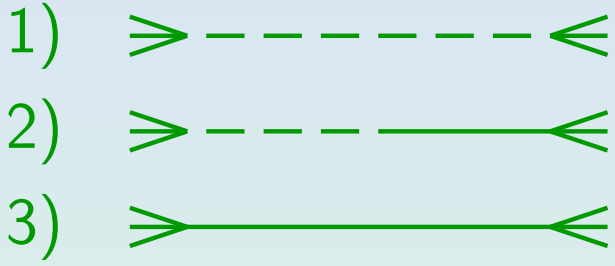


  - Viele-zu-viele Beziehungen werden in eine eigene Tabelle übersetzt, deren Schlüssel sich aus zwei Fremdschlüsseln zusammensetzt, die die beiden Entity-Typen referenzieren.
  - `verliehen_an(`<u>`InvNr`</u> $\rightarrow$ `Exemplar,` <u>`ANr`</u> $\rightarrow$ `Auftrag)`

    Man kann auch die umgekehrte Leserichtung der Beziehung wählen, wenn man die natürlicher findet. Man kann die Tabelle auch manuell umbenennen, z.B. „`Verleih`“.

Hausaufgabe 14.1: Logischer Entwurf (10)

- Welche der drei möglichen n:m-Beziehungen werden mit dieser Methode korrekt übersetzt?



A. 1

B. 2

C. 3

D. 1 und 3

E. alle

Hausaufgabe 14.1: Logischer Entwurf (11)

- Endgültige Tabellen:
  - Auftrag(ANr, von, bis°)
  - GerätTyp(BestellNr, Bezeichnung, Beschreibung)
  - Position(ANr → Auftrag, BestellNr → GerätTyp, Stück)
  - Exemplar(InvNr, KaufPreis, KaufDatum, BestellNr → GerätTyp)
  - verliehen_an(InvNr → Exemplar, ANr → Auftrag)
- Zusätzlich notwendige Integritätsbedingung:
  - „Zu jeder Zeile in GerätTyp muss es mindestens eine Zeile in Exemplar geben mit der gleichen BestellNr.“

Hausaufgabe 14.1: Logischer Entwurf (12)

- In der Aufgabe war eigentlich die ASCII-Notation verlangt. Dies lässt sich auch bei der Klausur leichter eingeben:
  - `Auftrag(#ANr, von, bis?)`
  - `GerätTyp(#BestellNr, Bezeichnung, Beschreibung)`
  - `Position(#ANr -> Auftrag,`
    `#BestellNr -> GerätTyp, Stück)`
  - `Exemplar(#InvNr, KaufPreis, KaufDatum,`
    `BestellNr -> GerätTyp)`
  - `verliehen_an(#InvNr -> Exemplar,`
    `#ANr -> Auftrag)`

  In der Klausur wird vorausgesetzt, dass Sie beide Varianten dieser Notation für relationale Schemata kennen.

Inhalt

1. Organisatorisches
2. Präsenzaufgabe 10
3. Aufgabe 14.1
4. Aufgabe 14.2

Aufgabe 14.2: Normalformen (1)

- Gegeben sei folgende Tabelle mit einigen Daten einer Schule:

SCHULE				
SCHUELER	KLASSE	FACH	LEHRER	RAUM
Karin Koch	7a	Deutsch	Herr Goetz	305
Karin Koch	7a	Englisch	Frau Rowling	305
Karin Koch	7a	Mathematik	Frau Noether	305
Frank Fischer	7a	Deutsch	Herr Goetz	305
Frank Fischer	7a	Englisch	Frau Rowling	305
Frank Fischer	7a	Mathematik	Frau Noether	305
Ronald Richter	10b	Mathematik	Frau Noether	210
Ronald Richter	10b	Englisch	Herr Pratchett	210
Ronald Richter	10b	Pyrotechnik	Herr Shimizu	210
Ronald Richter	10b	Religion	Herr Lewis	210

Die letzte Spalte heißt eigentlich „Klassenraum“. Das ist wichtig, weil die funktionale Abhängigkeit „Klasse bestimmt Klassenraum“ vorausgesetzt wird.

Aufgabe 14.2: Normalformen (2)

- Es seien folgende funktionale Abhängigkeiten gegeben:
  - SCHUELER $\longrightarrow$ KLASSE
  - KLASSE, FACH $\longrightarrow$ LEHRER
  - KLASSE $\longrightarrow$ KLASSENRAUM
  - LEHRER, KLASSENRAUM $\rightarrow$ LEHRER
  - SCHUELER, FACH $\rightarrow$ KLASSENRAUM
- Z.B. bedeutet die erste funktionale Abhängigkeit (FA)

  SCHUELER $\longrightarrow$ KLASSE:

  „Wenn zwei Zeilen den gleichen Wert in der Spalte SCHUELER haben, dann müssen sie auch den gleichen Wert in der Spalte KLASSE haben.“

  Jeder Schüler gehört zu einer eindeutig bestimmten Klasse, diese ist nicht vom Unterrichtsfach abhängig.

Aufgabe 14.2 a): Weitere FA (1)

- Geben Sie eine weitere funktionale Abhängigkeit an,
  - die nicht von den explizit gegebenen Abhängigkeiten impliziert wird,
  - aber im Beispielzustand der Tabelle gilt.

  Es ist nicht verlangt, dass die funktionale Abhängigkeit im realen Leben realistisch ist.

- Mögliche Lösung:

  $$\texttt{KLASSENRAUM} \longrightarrow \texttt{KLASSE}$$

  Dies wurde bei den funktionalen Abhängigkeiten vergessen. Es macht sogar im realen Leben Sinn: Ein Raum kann nur für eine Klasse Klassenraum sein (Forts. siehe nächste Folie).

Aufgabe 14.2 a): Weitere FA (2)

- Um zu prüfen, dass

$$\texttt{KLASSENRAUM} \longrightarrow \texttt{KLASSE}$$

  im Beispiel-Zustand gilt, muss man nachschauen, dass in Gruppen von Zeilen mit gleichem KLASSENRAUM auch überall die gleiche KLASSE steht.

  Man könnte eine SQL-Anfrage dafür schreiben.

- Zum Beweis, dass diese FA nicht von den gegebenen FAen impliziert wird, kann man die Attributhülle

$$\{\texttt{KLASSENRAUM}\}^+ = \{\texttt{KLASSENRAUM}\}$$

  berechnen (bezüglich der gegebenen FAen).

- Da sie nicht KLASSE enthält, wird $\texttt{KLASSENRAUM} \longrightarrow \texttt{KLASSE}$ nicht impliziert.

Aufgabe 14.2 a): Weitere FA (3)

- Es gibt noch weitere Lösungen, z.B.

$$\texttt{LEHRER} \longrightarrow \texttt{FACH}$$

- Im Beispiel-Zustand unterrichtet jeder Lehrer nur ein Fach.
- Dies wäre in der Realität aber eher nicht so.

  Das kann natürlich von der Schule abhängen. Man muss im Rahmen des Datenbank-Entwurfs welche funktionalen Abhängigkeiten vorausgesetzt werden können.

- Man könnte diese beiden funktionalen Abhängigkeiten auch weiter abschwächen, z.B. wäre auch eine Lösung:

$$\texttt{LEHRER, KLASSENRAUM} \rightarrow \texttt{FACH}$$

Aufgabe 14.2 b): Attributhülle (1)

- Berechnen Sie die Attributhülle $\{\texttt{SCHUELER}\}^+$.
- Lösung: Man startet mit dem gegebenen Attribut:

$$\{\texttt{SCHUELER}\}^+ = \{\texttt{SCHUELER}, \ldots\}.$$

- Aufgrund der funktionalen Abhängigkeit

$$\texttt{SCHUELER} \longrightarrow \texttt{KLASSE}$$

kann man KLASSE hinzufügen:

$$\{\texttt{SCHUELER}\}^+ = \{\texttt{SCHUELER}, \texttt{KLASSE}, \ldots\}.$$

Wenn die linke Seite einer funktionalen Abhängigkeit bereits in der Attributhülle ist, kann man auch die rechte Seite hinzufügen.

- Fortsetzung siehe nächste Folie.

Aufgabe 14.2 b): Attributhülle (2)

- Da man nun KLASSE hat, kann man die funktionale Abhängigkeit

$$\texttt{KLASSE} \longrightarrow \texttt{KLASSENRAUM}$$

anwenden:

$$\{\texttt{SCHUELER}\}^+ = \{\texttt{SCHUELER}, \texttt{KLASSE}, \texttt{KLASSENRAUM}\}.$$

- Weitere funktionale Abhängigkeiten können nicht angewendet werden:
  - $\texttt{KLASSE, FACH} \longrightarrow \texttt{LEHRER}$
  - $\texttt{LEHRER, KLASSENRAUM} \rightarrow \texttt{LEHRER}$
  - $\texttt{SCHUELER, FACH} \rightarrow \texttt{KLASSENRAUM}$

Aufgabe 14.2 c): Implizierte FA

- Implizieren die gegebenen funktionalen Abhängigkeiten die folgende Abhängigkeit?

$$\texttt{SCHUELER} \longrightarrow \texttt{KLASSENRAUM}$$

  Begründen Sie Ihre Aussage.

- Lösung: Ja, die FA wird logisch impliziert. Begründung:
  - Wir haben gerade die Attributhülle $\{\texttt{SCHUELER}\}^+$ berechnet.

    Im allgemeinen berechnet man für Aufgaben dieses Typs die Attributhülle der linken Seite der fraglichen funktionalen Abhängigkeit (bezüglich der gegebenen funktionalen Abhängigkeiten).

  - Die Attributhülle

    $$\{\texttt{SCHUELER}\}^+ = \{\texttt{SCHUELER}, \texttt{KLASSE}, \texttt{KLASSENRAUM}\}.$$

    enthält die rechte Seite der zu prüfenden FA (KLASSENRAUM).

Aufgabe 14.2 d): Schlüssel (1)

- Geben Sie alle minimalen Schlüssel der Tabelle an. Begründen Sie Ihre Aussage, d.h.
  - dass es jeweils ein Schlüssel ist,
  - dass dieser bzw. diese Schlüssel minimal sind, und
  - dass es keine weiteren Schlüssel gibt.
- Lösung: Wir müssen $\subseteq$-minimale Mengen $\mathcal{A}$ von Attributen finden, so dass $\mathcal{A}^+$ die Menge aller Attribute der Tabelle ist.

  Ein Schlüssel bestimmt alle Attribute der Tabelle funktional.

  Die Attributhülle $\mathcal{A}^+$ ist die Menge alle von $\mathcal{A}$ funktional bestimmten Attribute.

Aufgabe 14.2 d): Schlüssel (2)

- Die gegebenen funktionale Abhängigkeiten sind:
  - SCHUELER $\longrightarrow$ KLASSE
  - KLASSE, FACH $\longrightarrow$ LEHRER
  - KLASSE $\longrightarrow$ KLASSENRAUM
  - LEHRER, KLASSENRAUM $\rightarrow$ LEHRER
  - SCHUELER, FACH $\rightarrow$ KLASSENRAUM
- Die Attribute SCHUELER und FACH kommen auf keiner rechten Seite vor.
- Wenn sie nicht von Anfang an in $\mathcal{A}$ enthalten sind, können sie nicht in $\mathcal{A}^+$ enthalten sein.

Aufgabe 14.2 d): Schlüssel (3)

- Die beiden Attribute SCHUELER und FACH müssen also in jedem Schlüssel enthalten sein.
- Wenn man nun

$$\{\texttt{SCHUELER, FACH}\}^+$$

berechnet, stellt man fest, dass es alle Attribute sind:

  - SCHUELER $\longrightarrow$ KLASSE
  - KLASSE, FACH $\longrightarrow$ LEHRER
  - KLASSE $\longrightarrow$ KLASSENRAUM
- Damit haben wir Glück: Es gibt nur einen minimalen Schlüssel, nämlich diese beiden Attribute.

Aufgabe 14.2 e): BCNF prüfen (1)

- Ist die Tabelle bezüglich der fünf gegebenen funktionalen Abhängigkeiten in BCNF? Begründen Sie Ihre Aussage.
- Lösung: Nein, die drei ersten funktionalen Abhängigkeiten verletzen BCNF, da die linke Seite kein Schlüssel (nicht notwendig minimal) ist, und sie nicht trivial sind:
  - SCHUELER $\longrightarrow$ KLASSE
  - KLASSE, FACH $\longrightarrow$ LEHRER
  - KLASSE $\longrightarrow$ KLASSENRAUM
  - LEHRER, KLASSENRAUM $\rightarrow$ LEHRER
  - SCHUELER, FACH $\rightarrow$ KLASSENRAUM

Aufgabe 14.2 e): BCNF prüfen (2)

- Die vierte funktionale Abhängigkeit verletzt BCNF nicht, da sie trivial ist:

  $$\texttt{LEHRER, KLASSENRAUM} \rightarrow \texttt{LEHRER}$$

  Die rechte Seite (LEHRER) kommt auch links vor.

- Die fünfte funktionale Abhängigkeit verletzt BCNF nicht, da links ein vollständiger Schlüssel steht:

  $$\texttt{SCHUELER, FACH} \rightarrow \texttt{KLASSENRAUM}$$

  Es könnten links auch noch weitere Attribute stehen, das wäre kein Problem. Man kann die BCNF-Bedingung auch ohne Vorab-Berechnung der Schlüssel testen, indem man die Attributhülle der linken Seiten berechnet. Dies muss jeweils die Menge aller Attribute sein (es sei denn, die FA wäre trivial, dann kann man sie sowieso streichen).

Aufgabe 14.2 e): BCNF herstellen (1)

- Zweiter Teil der Aufgabe: Falls die Tabelle nicht in BCNF sein sollte, überführen Sie die Tabelle durch verlustlose Aufspaltung in mehrere Tabellen, die in BCNF sind.

  Alternativ dürfen Sie auch den 3NF Synthesealgorithmus verwenden.

- Verletzt $\alpha \longrightarrow \beta \in \mathcal{F}$ die BCNF, erzeugt man
  - R_1 mit den Attributen α^+.

    Dies enthält natürlich die Attribute $\alpha \cup \beta$ (wie in der oben gezeigten Dekomposition), könnte aber noch weitere Attribute enthalten (aber nur, wenn noch mehr FAen BCNF verletzen).

  - R_2 mit den Attributen $(\mathcal{A} \setminus \alpha^+) \cup (\alpha \setminus \beta)$.

Aufgabe 14.2 e): BCNF herstellen (2)

- Wenn nur eine funktionale Abhängigkeit $\alpha \longrightarrow \beta$ BCNF verletzt, erzeugt man einfach eine Relation mit den Attributen $\alpha \cup \beta$ (wobei α Schlüssel ist), und entfernt die rechte Seite β aus der ursprünglichen Relation.

  Voraussetzung dabei ist, dass $\alpha \cap \beta = \emptyset$. Ansonsten entfernt man den „trivialen Anteil“ $\alpha \cap \beta$ einfach von der rechten Seite.

- Im Beispiel gibt es leider mehrere Verletzungen.

  Z.B. wäre das Ergebnis schlecht, wenn man zuerst die Verletzung durch `SCHUELER` $\longrightarrow$ `KLASSE` angeht: Dann würde man das Attribut `KLASSE` aus der ursprünglichen Tabelle entfernen (nach obigem Algorithmus immerhin zusammen mit `KLASSENRAUM`). Anschließend könnte man die funktionale Abhängigkeit `KLASSE, FACH` $\longrightarrow$ `LEHRER` nicht mehr ausdrücken. Man bekommt durch sukzessive Aufspaltung immer Tabellen in BCNF, und die Aufspaltung ist verlustlos, nicht immer das gefühlt beste Ergebnis.

Aufgabe 14.2 e): BCNF herstellen (3)

- Man sollte bei mehreren Verletzungen die FAen möglichst so anordnen, dass Attribute auf der rechten Seite (die entfernt werden) nicht in einer späteren FA links vorkommen.
- Starten wir also mit:

  KLASSE, FACH $\longrightarrow$ LEHRER

  Man könnte genauso gut mit KLASSE $\longrightarrow$ KLASSENRAUM starten, dagegen wäre SCHUELER $\longrightarrow$ KLASSE ungünstig.

- Man legt eine neue Tabelle mit den Attributen der FA an:

  LEHRER_EINTEILUNG(<u>KLASSE</u>, <u>FACH</u>, LEHRER)

- Die rechte Seite wird aus der ursprünglichen Tabelle entfernt:

  SCHULE1(SCHUELER, KLASSE, FACH, KLASSENRAUM)

  Dabei bilden KLASSE und FACH einen Fremdschlüssel zu LEHRER_EINTEILUNG.

Aufgabe 14.2 e): BCNF herstellen (4)

- Nun verletzt SCHULE1 noch immer BCNF, z.B. wegen

  KLASSE $\longrightarrow$ KLASSENRAUM.

  Auch wegen SCHUELER $\longrightarrow$ KLASSE, aber das heben wir uns noch auf.

- Also legt man eine neue Tabelle mit den Attributen der FA an:

  KLASSEN(<u>KLASSE</u>, KLASSENRAUM).

  Die linke Seite der FA wird immer Schlüssel. Sie bestimmt nach der Konstruktion ja die übrigen Attribute.

- Wieder entfernen wir die rechte Seite der FA aus der ursprünglichen Tabelle:

  SCHULE2(SCHUELER, KLASSE$\rightarrow$KLASSEN, FACH)

Aufgabe 14.2 e): BCNF herstellen (5)

- Auch SCHULE2 verletzt noch BCNF wegen

$$\text{SCHUELER} \longrightarrow \text{KLASSE},$$

- Also machen wir auch dazu eine Tabelle:

  KLASSEN_EINTEILUNG(<u>SCHUELER</u>, KLASSE→KLASSEN).

- Die rechte Seite wird aus der ursprünglichen Tabelle entfernt, übrig bleibt:

  LERNT(SCHUELER→KLASSEN_EINTEILUNG, FACH)

  Leider wird dabei der ursprüngliche Fremdschlüssel zerstört: Bei SCHULE1 wurde festgelegt, dass KLASSE und FACH zusammen einen Fremdschlüssel bilden, der auf LEHRER_EINTEILUNG verweist. Jetzt werden die beiden Attribute aber auseinander gerissen.

Aufgabe 14.2 e): BCNF herstellen (6)

- Das Endergebnis ist also:
  - LEHRER_EINTEILUNG(KLASSE, FACH, LEHRER)
  - KLASSEN(KLASSE, KLASSENRAUM).
  - KLASSEN_EINTEILUNG(SCHUELER, KLASSE→KLASSEN).
  - LERNT(SCHUELER→KLASSEN_EINTEILUNG, FACH)

    Da es für diese Tabelle keine FAen mehr gibt, sind beide Attribute zusammen der Schlüssel.
- Im Prinzip hat man aus jeder der interessanten FAen eine Tabelle gemacht, und LERNT ist der Rest.

  Der 3NF-Synthesealgorithmus macht im Prinzip genau das. Im Skript ist bei diesem Algorithmus erklärt, wie man die gegebenen FAen aufbereiten muss, damit das funktioniert. Im Beispiel würden bei der Berechnung der „kanonischen Überdeckung“ die vierte und fünfte FA eliminiert.

Aufgabe 14.2 e): Exkurs zu 4NF (1)

Nicht prüfungsrelevant:

- Das Beispiel ist leider komplizierter als beabsichtigt.
- Intuitiv ist die Tabelle

  LERNT(SCHUELER, FACH)

  für diese Anwendung falsch.
- Die Fächer hängen normalerweise nur von der Klasse ab, und nicht vom einzelnen Schüler. Dann würde man eher diese Tabellen wollen:
  - KLASSEN(KLASSE, KLASSENRAUM)
  - FAECHER(KLASSE→KLASSEN, FACH, LEHRER)
  - KLASSEN_EINTEILUNG(SCHUELER, KLASSE→KLASSEN)

Aufgabe 14.2 e): Exkurs zu 4NF (2)

- Die gegebenen funktionalen Abhängigkeiten sagen aber nicht aus, dass die Fächer nur von der Klasse abhängen.

  Das kann man beweisen, indem man einen DB-Zustand macht, der alle FAen erfüllt, aber zwei Schüler in der gleichen Klasse enthält, die verschiedene Fächer haben.

- Es kommt ja auch vor, dass Klassen bei bestimmten Fächern aufgespalten werden:
  - Ein Teil der Schüler hat Latein, der andere Teil Französisch.
  - Oder evangelische Religion, katholische Religion, Ethik.
- Wenn die Fächer tatsächlich vom Schüler abhängen und nicht von der Klasse, war der obige Entwurf doch richtig.

Aufgabe 14.2 e): Exkurs zu 4NF (3)

- Will man ausdrücken, dass zwei Schüler in der gleichen Klasse immer die gleiche Menge von Fächern haben, braucht man dazu „mehrwertige Abhängigkeiten“:
  - `KLASSE` $\twoheadrightarrow$ `SCHUELER`
  - `KLASSE` $\twoheadrightarrow$ `FACH, LEHRER`
- Diese Art von Integritätsbedingungen bedeutet: „Wenn zwei Zeilen in der linken Seite übereinstimmen, kann man die Werte der rechten Seite vertauschen, und bekommt zwei Zeilen, die auch in der Tabelle enthalten sein müssen.“

  Z.B. kann man für zwei Zeilen mit der gleichen `KLASSE` die `SCHUELER` vertauschen: Die so entstandenen „neuen“ Zeilen müssen bereits in der Tabelle enthalten sein.

Aufgabe 14.2 e): Exkurs zu 4NF (4)

- Normalisierung mit mehrwertigen Abhängigkeiten führt zu 4NF.
- Die Reihenfolge der Normalformen ist: 2NF, 3NF, BCNF, 4NF, 5NF.
  - Normalformen später in der Liste implizieren Normalformen früher in der Liste.
  - Z.B. ist eine Tabelle in BCNF automatisch in 3NF.
  - Von diesen Normalformen basieren 2NF, 3NF und BCNF auf funktionalen Abhängigkeiten.

    Wenn man mehr will, braucht man allgemeinere Integritätsbedingungen.
- Dies alles wird ausführlich in der Vorlesung „Datenbank-Entwurf“ im Master behandelt.