

Einführung in Datenbanken

Übung 14: ER-Diagramme und Logischer Entwurf

Prof. Dr. Stefan Brass

PD Dr. Alexander Hinneburg

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2020/21

<http://www.informatik.uni-halle.de/~brass/db20/>

Inhalt

- 1 SQL Island
- 2 Hausaufgabe 13
- 3 Präsenzaufgabe 13
- 4 ER-Modell
- 5 Normalformen
- 6 Präsenzaufgabe 14

Hausaufgabe 12b (1)

- SQL Island: [<https://sql-island.informatik.uni-kl.de/>]
- Haben Sie SQL Island durchgespielt?
 - A. Nur so viel, wie verlangt war.
 - B. Etwas weiter, aber nicht bis zu Ende.
 - C. Ja, ich habe es ganz durchgespielt.
- Gab es dabei besondere Schwierigkeiten?
- Hat es Ihnen gefallen?
 - A. Sehr.
 - B. Etwas.
 - C. Gar nicht.

Hausaufgabe 12b (2)

- „Hui, was ist passiert? Es scheint, als habe ich als einziger den Flugzeugabsturz überlebt. Gut, dass ich auf dieser Insel gelandet bin. Hier gibt es ja sogar ein paar Dörfer.“
- Beispiel-Anfrage:

```
SELECT * FROM dorf
```

- „Und jede Menge Bewohner gibt es hier auch. Zeige mir die Liste der Bewohner.“

Am unteren Bildschirmrand steht das DB-Schema aus drei Tabellen:

DORF (dorfnr, name, haeuptling)

BEWOHNER (bewohnernr, name, dorfnr, geschlecht, beruf, gold, status)

GEGENSTAND (gegenstand, besitzer)

- Die erwartete Anfrage ist natürlich:

```
SELECT * FROM bewohner
```

Hausaufgabe 12b (3)

- Um die Aufgabe interessanter zu machen, war aber eine andere Anfrage verlangt, die das Spiel als korrekt akzeptiert.
- Wie man dem Artikel auf der BTW 2015 [http://www.btw-2015.de/res/proceedings/Hauptband/Demo/Schildgen-SQL-Grundlagen_spielend.pdf] entnehmen kann,
 - führt das System die eingegebene Anfrage zunächst aus, und gibt eventuelle Syntaxfehler an den Benutzer weiter,
 - und berechnet dann die Differenzen der Ergebnis-Mengen der Nutzer-Anfrage Q und der Musterlösung L :
 $Q \text{ EXCEPT } L$ und $L \text{ EXCEPT } Q$Ist das Ergebnis jeweils leer, so hat die Anfrage im Beispiel-Zustand das korrekte Ergebnis geliefert.

Stimmen Spaltenzahl oder Datentyp nicht, gibt es hier einen Syntaxfehler.

Hausaufgabe 12b (4)

- Jeder Benutzer hat seine eigene SQLite-Datenbank.

Die Benutzer sollen sich ja nicht gegenseitig stören, oder gar das Spiel kaputt machen können. Im späteren Verlauf gibt es auch Updates.

Die Größe der Datenbank wird überwacht und ist auf wenige Kilobytes beschränkt (wenn man Updates erlaubt, könnte jemand sonst das System lahmlegen, indem er die Platte voll macht).

- Dass es eine SQLite-DB ist, kann man z.B. mit folgender Anfrage sehen (es steht aber auch im obigen Artikel):

```
select sqlite_version();
```

- Diese Anfrage zeigt die Tabellen-Deklarationen:

```
select * from sqlite_master;
```

- Mich interessieren diese Fragen, weil wir selbst ein “Retro“-Textadventure als SQL-Lernspiel entwickeln.

Hausaufgabe 12b (5)

- Man kann statt der erwarteten Lösung

```
SELECT * FROM BEWOHNER
```

eine Anfrage eingeben, die äquivalent ist:

```
SELECT *
FROM   BEWOHNER
WHERE  name LIKE 'A%'
OR     name NOT LIKE 'A%'
OR     name IS NULL
```

- Die folgende Anfrage ist nur unter der Voraussetzung äquivalent, dass `dorfnr` wirklich ein Fremdschlüssel ist:

```
SELECT *
FROM   BEWOHNER
WHERE  dorfnr IN (SELECT dorfnr FROM dorf)
```

Hausaufgabe 12b (6)

- Es werden aber auch Anfragen akzeptiert, die nur zufällig das richtige Ergebnis liefern (im Beispiel-Zustand):

```
SELECT *
FROM   BEWOHNER
WHERE  gold < 2500
OR     status = 'boese'
```

- Diese Anfrage ist natürlich nicht äquivalent zur erwarteten Lösung (Musterlösung).

```
select max(gold) from bewohner where status <> 'boese'
```

liefert im Beispiel-Zustand 2280. Würde es einen Bewohner geben, der nicht böse ist (im Beispiel friedlich oder gefangen), und mehr als 2500 Goldstücke besitzt, würde die Anfrage nicht das korrekte Ergebnis liefern.

- Die Äquivalenz von SQL-Anfragen ist nicht entscheidbar, eine immer korrekte automatische Korrektur ist nicht möglich.

Hausaufgabe 12c (1)

- „Mensch, was bin ich hungrig. Ich suche mir erst einmal einen Metzger, bei dem ich eine Scheibe Wurst schnorren kann.“
- Beispiel-Anfrage:

```
SELECT * FROM bewohner WHERE beruf = 'Metzger'
```
- Erich: „Hier, lass es dir schmecken! Und pass bei deiner Reise gut auf, dass du dich von bösen Bewohnern fern hältst, solange du unbewaffnet bist. Denn nicht jeder hier ist friedlich!“
- „Danke Erich! Nagut, dann muss ich mal schauen, welche Bewohner friedlich sind.“
- Das ist die zweite Aufgabe.

Hausaufgabe 12c (2)

- Die erwartete Lösung ist natürlich:

```
SELECT * FROM BEWOHNER
WHERE status = 'friedlich'
```

- Folgende Anfrage liefert im Beispiel-Zustand das gleiche Ergebnis (bis auf Duplikate), setzt aber voraus, dass
 - „friedlich“, „boese“ und „gefangen“ die einzigen Status-Werte sind und
 - DORF nicht leer ist (bzw. nur zusammen mit BEWOHNER, das würde aus dem Fremdschlüssel folgen):

```
SELECT B.*
FROM BEWOHNER B, DORF
WHERE status NOT IN ('boese', 'gefangen')
```

Dies ist wegen des kartesischen Produkts etwas fies, übertreiben Sie es nicht.

Es kommt jede Zeile drei Mal heraus, aber die Anfrage besteht den Test.

Inhalt

- 1 SQL Island
- 2 Hausaufgabe 13**
- 3 Präsenzaufgabe 13
- 4 ER-Modell
- 5 Normalformen
- 6 Präsenzaufgabe 14

Hausaufgabe 13a (2)

- Rebsorten (z.B. Riesling, Spätburgunder) sind durch ihren Namen identifiziert.
- Auch hier ist eine optionale Beschreibung zu speichern.

Rebsorte

Name

o Beschreibung

Hausaufgabe 13a (3)

- Weine sind durch Ihre Bezeichnung identifiziert (dies ist sozusagen der Markenname, der über mehrere Jahrgänge gleich bleibt).
- Optional können der Name des Weinguts sowie eine Beschreibung gespeichert werden.

Wein

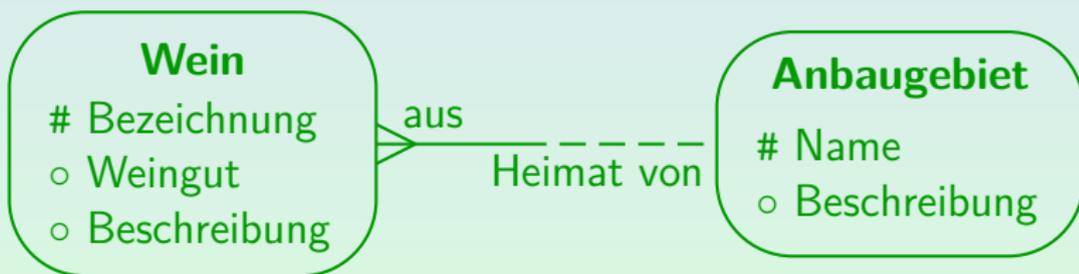
Bezeichnung

○ Weingut

○ Beschreibung

Hausaufgabe 13a (4)

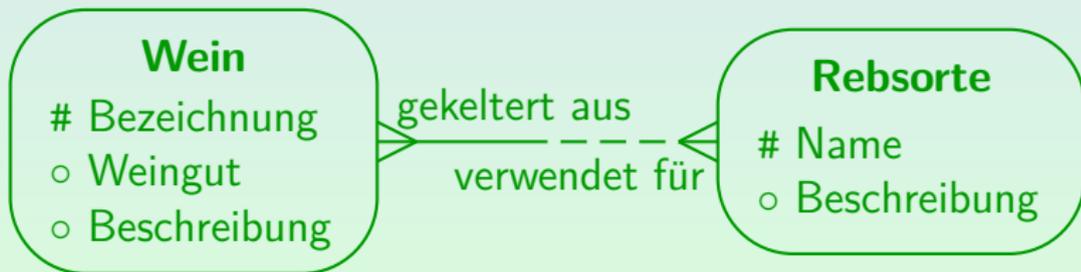
- Weine sind genau einem Weinanbaugebiet zugeordnet (minimal eins und maximal eins).
- Ein Weinanbaugebiet kann beliebig viele Weine enthalten (auch 0).



- Eins-zu-viele Beziehung: Ein Anbaugebiet, viele Weine.

Hausaufgabe 13a (5)

- Für einen Wein wurden ein oder mehr Rebsorten verwendet.
- Eine Rebsorte kann natürlich in mehreren Weinen enthalten sein.
- Es soll aber auch möglich sein, Rebsorten schon zu erfassen, bevor man einen Wein aus ihr hat. D.h. Rebsorten können auch zu 0 Weinen in Beziehung stehen.

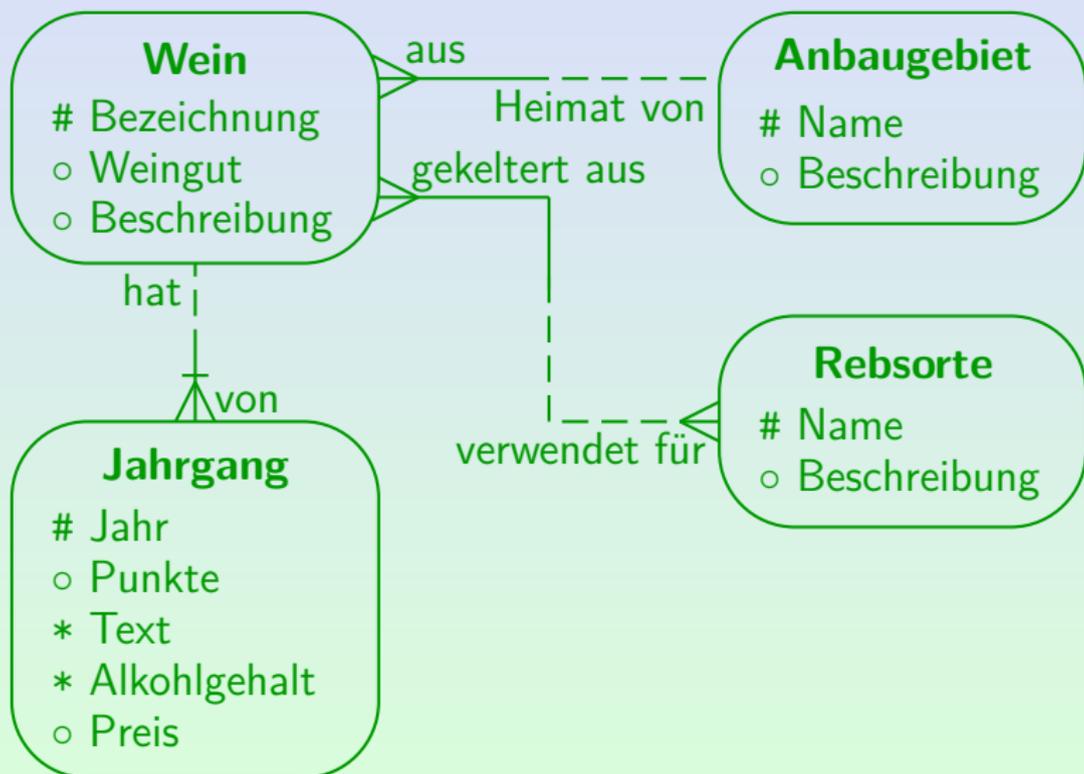


- Viele-zu-viele Beziehung.

Hausaufgabe 13a (6)

- Ein Jahrgang eines Weins ist durch den Wein und die Jahreszahl identifiziert (der Jahrgang ist, was konkret in Flaschen abgefüllt wird).
- Zum Jahrgang möchte der Weinliebhaber einen persönlichen Punktwert speichern (wie sehr ihm der Wein geschmeckt hat), einen kurzen Text, den Alkoholgehalt, sowie den Preis einer Flasche.
- Der Punktwert ist optional, da die Daten der Weinflasche für den Jahrgang schon erfasst werden sollen, bevor der Wein ausgedruckt wurde.
- Der Preis ist auch optional: Er kann unbekannt sein, wenn es z.B. ein Geschenk war.
- Der Text und der Alkoholgehalt sind nicht optional.

Hausaufgabe 13a (8)



Hausaufgabe 13b (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.

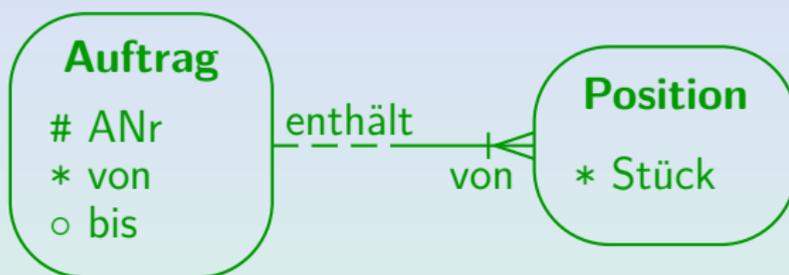
Hausaufgabe 13b (3)

- Zunächst legt man für jeden Entity-Typ eine Tabelle an, und übernimmt die Attribute, Schlüssel, Optionalität:
 - Auftrag(ANr, von, bis^o)
 - 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 13b (4)

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

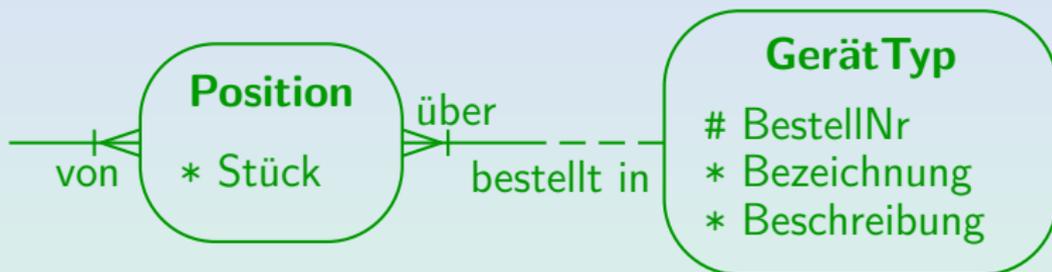


- Der Schlüssel der Eins-Seite (Auftrag) wird als Fremdschlüssel in die viele Seite (Position) eingefügt, um die Eins-zu-viele-Beziehung zu implementieren.
- Da die Beziehung identifizierend ist (Position ist schwaches Entity), wird der Fremdschlüssel Teil des Primärschlüssels.
- **Position(ANr → Auftrag, Stück)**

Man könnte den Fremdschlüssel auch „von“ nennen.

Hausaufgabe 13b (5)

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

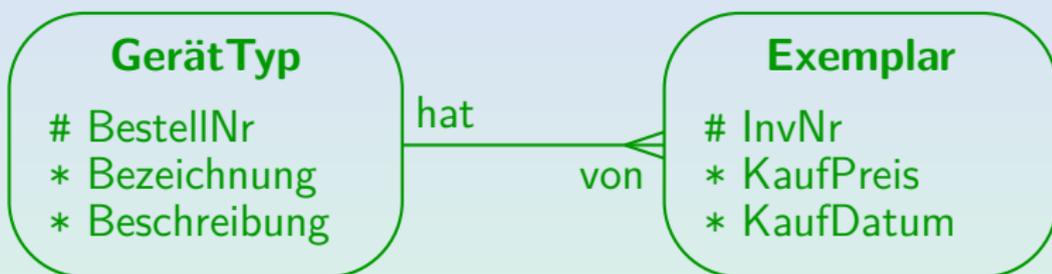


- Also kommt auch der Schlüssel von GerätTyp als Fremdschlüssel in die Tabelle Position und wird dort auch Teil des Schlüssels.
- $\text{Position}(\underline{\text{ANr}} \rightarrow \text{Auftrag}, \underline{\text{BestellNr}} \rightarrow \text{GerätTyp}, \text{Stück})$

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

Hausaufgabe 13b (6)

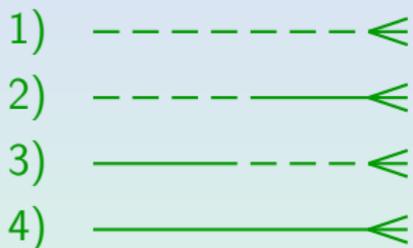
- Zwischen GerätTyp und Exemplar gibt es eine normale 1:n-Beziehung:



- 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(InvNr, KaufPreis, KaufDatum, BestellNr → GerätTyp)`

Hausaufgabe 13b (7)

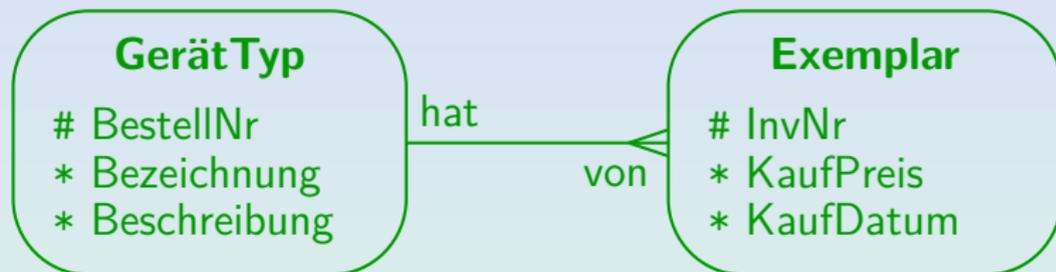
- Welche der vier möglichen 1:n-Beziehungen brauchen eine zusätzliche Integritätsbedingung?



- A. 1 und 2
- B. 3 und 4
- C. 1 und 3
- D. 2 und 4
- E. keine

Hausaufgabe 13b (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 13b (9)

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

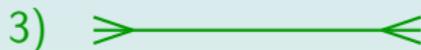


- Viele-zu-viele Beziehungen werden in eine eigene Tabelle übersetzt, deren Schlüssel sich aus zwei Fremdschlüsseln zusammensetzt, die die beiden Entity-Typen referenzieren.
- `verliehen_an(InvNr → Exemplar, ANr → Auftrag)`

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 13b (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 13b (11)

- Endgültige Tabellen:
 - Auftrag(ANr, von, bis^o)
 - 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 13b (12)

- Sie müssen auch die ASCII-Notation für die Tabellen lesen können:
 - `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 gab es eine Zuordnungsaufgabe, bei der man Attribute (teils mit # und -> (Tabelle)) Tabellen zuordnen sollte. Dort war für die Optionen kein Unterstreichen möglich.

Präsenzaufgabe 13: Logischer Entwurf (1)

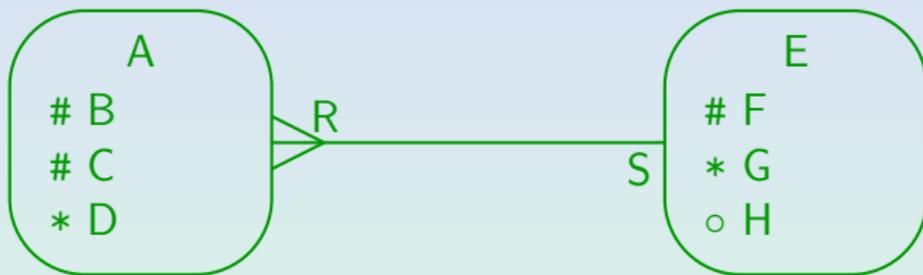
- Übersetzen Sie folgendes ER-Diagramm in das relationale Modell:



- 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.

Präsenzaufgabe 13: Logischer Entwurf (2)

- Gegebenes ER-Diagramm:



- Der logische Entwurf ist ein formal ablaufender Algorithmus, man braucht die Anwendung (Beziehung des ER-Diagramms zur realen Welt) nicht zu verstehen:
 - $E(\underline{F}, G, H^o)$ $E(\#F, G, H?)$
 - $A(\underline{B}, \underline{C}, D, F \rightarrow E)$ $A(\#B, \#C, D, F \rightarrow E)$

Präsenzaufgabe 13: Logischer Entwurf (3)

- Da jedes **E-Entity** eine Beziehung zu mindestens einem **A-Entity** haben muss, braucht man noch folgende Bedingung:

- Zu jeder Zeile in der Tabelle **E** gibt es mindestens eine Zeile in der Tabelle **A** mit dem gleichen **F-Wert**.

- $\forall E x \exists A y: x.F = y.F$

- SQL-Anfrage zum Finden von Integritäts-Verletzungen:

```
SELECT 'E ohne Beziehung zu einem A: ' || E.F
FROM   E
WHERE  E.F NOT IN (SELECT A.F FROM A)
```

- Alternativ:

```
SELECT 'E ohne Beziehung zu einem A: ' || x.F
FROM   E x
WHERE  NOT EXISTS (SELECT * FROM A y
                  WHERE  x.F = y.F)
```


Alte Aufgaben zu Kardinalitäten (1)

- Legen Sie die Kardinalitäten für eine Beziehung zwischen „Bestellung“ und „Kunde“ fest.

Hinweis: Neben normalen Kunden enthält die DB auch solche, die noch nichts bestellt haben, sondern nur einen Produktkatalog erhalten haben. Fragen Sie, wenn noch etwas unklar ist. Es ist normal, dass der Datenbank-Entwerfer den Anwendungsexperten fragt.

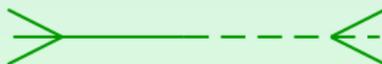
A.



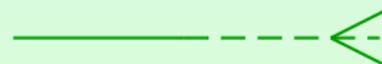
B.



C.



D.



Alte Aufgaben zu Kardinalitäten (2)

- Legen Sie nun die Kardinalitäten zwischen „Bestellung“ und „Produkt“ fest.

Eine Bestellung kann mehrere Produkte umfassen. Das gleiche Produkt kann von mehreren Kunden bestellt werden (z.B. bei einem Online-Buchhandel).

A.



B.



C.



D.



Funktionale Abhängigkeiten

- Funktionale Abhängigkeiten schreibt man in der Form

$$A_1, \dots, A_n \longrightarrow B_1, \dots, B_m.$$

Dabei sind die A_i und B_j Attribute einer Relation R .

In der Normalformtheorie konzentriert man sich immer auf eine einzige Relation. Alle Attribute stammen aus dieser Relation. Deswegen wird die Relation in der Notation von funktionalen Abhängigkeiten (FAen) nicht explizit angegeben.

- „ A_1, \dots, A_n bestimmen [funktional | eindeutig] B_1, \dots, B_m “.
- Die FA $A_1, \dots, A_n \longrightarrow B_1, \dots, B_m$ auf der Relation R ist nur eine Abkürzung für die Formel

$$\forall R X, R Y: \quad X.A_1 = Y.A_1 \wedge \dots \wedge X.A_n = Y.A_n \rightarrow \\ X.B_1 = Y.B_1 \wedge \dots \wedge X.B_m = Y.B_m$$

- D.h.: Wenn zwei Zeilen die gleichen Werte in A_1, \dots, A_n haben, dann müssen sie auch in B_1, \dots, B_m übereinstimmen.

Funktionale Abhängigkeiten: Übung (1)

- Die folgende Relation dient zur Speicherung von Bestellungen:

`BESTELLT(AUF_NR, DATUM, KD_NR,
PRD_NR, PRD_BEZ, MENGE)`

Aufträge einer Blumen-Handlung. AUF_NR: Auftrags-Nummer,
KD_NR: Kunden-Nummer PRD_: Produkt-Nummer.

BESTELLT					
AUF_NR	DATUM	KD_NR	PRD_NR	PRD_BEZ	MENGE
1	14.02.2021	10	100	Rosen	3
1	14.02.2021	10	101	Fresien	5
2	15.02.2021	10	100	Rosen	1
3	15.02.2021	11	100	Rosen	1

- Ursprüngliche Aufgabe: „Geben Sie alle funktionalen Abhängigkeiten an, die für diese Relation gelten.“

In einem Auftrag können mehrere Produkte bestellt werden.

Funktionale Abhängigkeiten: Übung (2)

- Gegeben die Relation mit Informationen zu Bestellungen:

BESTELLT					
AUF_NR	DATUM	KD_NR	PRD_NR	PRD_BEZ	MENGE
1	14.02.2021	10	100	Rosen	3
1	14.02.2021	10	101	Fresien	5
2	15.02.2021	10	100	Rosen	1
3	15.02.2021	11	100	Rosen	1

- Welche der folgenden funktionalen Abhängigkeiten gilt?
 - $AUF_NR \rightarrow DATUM$
 - $DATUM \rightarrow AUF_NR$
 - Beide.
 - Keine von beiden.

Funktionale Abhängigkeiten: Übung (3)

- Gegeben die Relation mit Informationen zu Bestellungen:

BESTELLT					
AUF_NR	DATUM	KD_NR	PRD_NR	PRD_BEZ	MENGE
1	14.02.2021	10	100	Rosen	3
1	14.02.2021	10	101	Fresien	5
2	15.02.2021	10	100	Rosen	1
3	15.02.2021	11	100	Rosen	1

- Was würde die funktionale Abhängigkeit

$AUF_NR \longrightarrow PRD_NR$ bedeuten?

- Jeder Auftrag kann nur ein Produkt enthalten.
- Jedes Produkt kann nur ein Mal bestellt werden.
- Jeder Auftrag muss mindestens ein Produkt enthalten.
- Keine dieser Aussagen entspricht der FA.

Funktionale Abhängigkeiten: Übung (4)

- Im Beispiel-Zustand ist die funktionale Abhängigkeit $AUF_NR \longrightarrow PRD_NR$ verletzt:

BESTELLT					
AUF_NR	DATUM	KD_NR	PRD_NR	PRD_BEZ	MENGE
1	14.02.2021	10	100	Rosen	3
1	14.02.2021	10	101	Fresien	5
2	15.02.2021	10	100	Rosen	1
3	15.02.2021	11	100	Rosen	1

- Es gibt zwei Zeilen mit gleicher Auftrags-Nummer AUF_NR, aber verschiedener Produkt-Nummer PRD_NR.
- Wenn der Beispiel-Zustand zulässig sein soll, darf man diese funktionale Abhängigkeit nicht fordern.

FAen sind Integritätsbedingungen und müssen in jedem Zustand gelten.

Funktionale Abhängigkeiten: Übung (5)

- Gegeben die Relation mit Informationen zu Bestellungen:

BESTELLT					
AUF_NR	DATUM	KD_NR	PRD_NR	PRD_BEZ	MENGE
1	14.02.2021	10	100	Rosen	3
1	14.02.2021	10	101	Fresien	5
2	15.02.2021	10	100	Rosen	1
3	15.02.2021	11	100	Rosen	1

- Was halten Sie von der FA $AUF_NR, PRD_NR \longrightarrow MENGE$?
 - Gilt im Beispielzustand und wäre allgemein sinnvoll.
 - Gilt im Beispielzustand, sollte aber nicht gefordert werden.
 - Gilt nicht im Beispiel-Zustand, macht aber Sinn.
 - Gilt nicht im Beispiel-Zustand, sollte nicht gefordert werden.

Funktionale Abhängigkeiten: Übung (6)

- Man will nie „alle“ funktionalen Abhängigkeiten, die für eine Relation gefordert werden könnten.
- Das würde z.B. auch folgende „triviale“ FA beinhalten:

$$KD_NR \longrightarrow KD_NR$$

Eine FA heißt trivial, wenn die rechte Seite eine Teilmenge der linken Seite ist. Solche Abhängigkeiten gelten in jedem beliebigen Zustand (Tautologie).

- Man will nur eine repräsentative Teilmenge, die alle anderen logisch impliziert.
- Da funktionale Abhängigkeiten nur eine Abkürzung (spezielle Schreibweise) für eine bestimmte Art logischer Formeln sind, ist Implikation für funktionale Abhängigkeiten schon in der Logik definiert.

Funktionale Abhängigkeiten: Übung (7)

- Im Beispiel sind folgende funktionalen Abhängigkeiten sinnvoll:

- $AUF_NR \longrightarrow DATUM, KD_NR$
- $PRD_NR \longrightarrow PRD_BEZ$
- Eventuell $PRD_BEZ \longrightarrow PRD_NR$
- $AUF_NR, PRD_NR \longrightarrow MENGE$

- Implizieren diese FAen die folgende FA (Ja/Nein)?

$AUF_NR, PRD_NR \longrightarrow DATUM$

Im Skript ist erklärt, wie man die Frage mit der Attribut-Hülle entscheidet:

Man berechnet die Attribut-Hülle $\{AUF_NR, PRD_NR\}^+$ (bezüglich der gegebenen FAen) und schaut, ob MENGE darin ist. Wenn ja, ist die FA impliziert.

Funktionale Abhängigkeiten: Übung (8)

- Man kann aus gegebenen funktionalen Abhängigkeiten einen Schlüssel der Relation berechnen.
- Dazu muss man eine Menge von Attributen finden, die alle anderen Attribute funktional bestimmt.
- Im Beispiel wäre das $\{\text{AUF_NR}, \text{PRD_NR}\}$.
 - $\text{AUF_NR} \longrightarrow \text{DATUM}, \text{KD_NR}$
 - $\text{PRD_NR} \longrightarrow \text{PRD_BEZ}$
 - $\text{AUF_NR}, \text{PRD_NR} \longrightarrow \text{MENGE}$
- Falls man zusätzlich
 - $\text{PRD_BEZ} \longrightarrow \text{PRD_NR}$gefordert hat, wäre $\{\text{AUF_NR}, \text{PRD_BEZ}\}$ ein weiterer Schlüssel.

Funktionale Abhängigkeiten: Übung (9)

- Die Tabelle ist natürlich schlecht:
 - Kundennummer und Auftrags-Datum werden zu jeder Position des Auftrags gespeichert.
 - Wenn in einem Auftrag n Produkte bestellt sind, werden diese Daten für den gleichen Auftrag n Mal gespeichert.
 - Entsprechend wird die Produkt-Bezeichnung zu einer Produkt-Nummer für jede Bestellung des Produkts gespeichert.

Und damit gar nicht, solange das Produkt nicht bestellt wurde.

- Die Tabelle verletzt die Boyce-Codd-Normalform (BCNF).

Tatsächlich verletzt sie sogar die zweite Normalform.

Funktionale Abhängigkeiten: Übung (10)

- Man muss prüfen, ob funktionale Abhängigkeiten (die nicht trivial sind) links einen Schlüssel haben (nicht notwendigerweise minimal):
 - $AUF_NR \longrightarrow DATUM, KD_NR$
 - $PRD_NR \longrightarrow PRD_BEZ$
 - $AUF_NR, PRD_NR \longrightarrow MENGE$
- Nur die dritte funktionale Abhängigkeit ist in Ordnung, die ersten beiden verletzen BCNF.

Sie enthalten zwar ein Schlüsselattribut, aber nicht einen vollständigen Schlüssel. Mehr Attribute als im (minimalen) Schlüssel sind in Ordnung (dann ist es ja noch immer ein Schlüssel), weniger nicht.
- Man kann BCNF auch ohne explizite Schlüsselberechnung mit der Attribut-Hülle prüfen (siehe Skript).

Präsenzaufgabe 14: Funktionale Abhängigkeiten

- (2+2 Punkte:) Es sei folgende Tabelle betrachtet:

R		
A	B	C
1	2	3
4	x	y

Geben Sie Werte für x und y aus $\{1, 2, 3, 4, 5, 6\}$ an, so dass

- sowohl die funktionale Abhängigkeit $A \rightarrow B$ erfüllt ist,
- als auch (im gleichen Zustand) die FA $B \rightarrow C$ verletzt ist.
- Bonusaufgabe: Geben Sie eine dritte Zeile (a, b, c) an, so dass $A \rightarrow B$ weiter gilt, aber $A \rightarrow C$ verletzt ist.
- Wenn Sie fertig sind, könnten Sie sich in Ihrer Gruppe noch unterhalten, wie Sie sich auf die Klausur vorbereiten.