

ILIAS-Beispielklausur “Einführung in Datenbanken”

Januar 2020

Basierend auf der Klausur von Wintersemester 2010/2011

Prof. Dr. Wolfgang May

- Diese Klausur dient als Beispiel- und Experimentalklausur mit dem ILIAS-System. Sie ist daher umfangreicher als üblich (Dauer: auf Papier ca. 120-150 Minuten, in Ilias ca. 150-180 Minuten)
- Bearbeiten Sie zuerst Aufgabe 1 (ER-Diagramm), dann *entweder* Aufgabe 2 (Tabellen in Textmode) oder Aufgabe 3 (Tabellen grafisch z.B. mit draw.io), und dann Aufgaben 4 bis 14 (diese bauen auf Aufgaben 1-3 auf); Aufgaben 15-21 sind davon unabhängig.
- Für Ausdrücke der relationalen Algebra sind in dieser Probeklausur die einfachen Aufgaben im Textmodus angelegt (schreiben Sie einfach Klammersausdrücke mit “pi”, “sigma”, “rho”, “join” für die Operatoren), die komplexeren als pdf-upload (Bäume mit draw.io, oder sonstwas).

	Max. Punkte	Erreichte Punkte
Aufgabe 1 (ER-Modell)	20	
Aufgabe 2 (Transformation in das Rel. Modell (Textmode))	9	
Aufgabe 3 (Transformation in das Rel. Modell (grafisch))	9	
Aufgabe 4 (Relationales Modell CREATE TABLE)	7	
Aufgabe 5 (SQL und Relationale Algebra (a1))	2	
Aufgabe 6 (SQL und Relationale Algebra (a2))	2	
Aufgabe 7 (SQL und Relationale Algebra (b1))	3	
Aufgabe 8 (SQL und Relationale Algebra (b2))	3	
Aufgabe 9 (SQL und Relationale Algebra (c))	2	
Aufgabe 10 (SQL und Relationale Algebra (d1))	3	
Aufgabe 11 (SQL und Relationale Algebra (d2))	3	
Aufgabe 12 (SQL und Relationale Algebra (e))	7	
Aufgabe 13 (SQL und Relationale Algebra (f))	6	
Aufgabe 14 (SQL und Relationale Algebra (f2-upload))	6	
Aufgabe 15 (Abstrakt: SQL und Algebra)	5	
Aufgabe 16 (Theorie-Aufgabe)	5	
Aufgabe 17 (Algebra-Anfragebäume/Multiple-Choice)	8	
Aufgabe 18 (Schemaentwurf/Multiple-Choice)	4	
Aufgabe 19 (Ausdruckskraft - (Umordnungs-Beispiel))	4	
Aufgabe 20 (Transaktionen)	6	
Aufgabe 21 (Abb. ER-Modell auf Relationales Modell (a))	2	
Aufgabe 22 (Abb. ER-Modell auf Relationales Modell (b))	6	
Summe	122	

Note:

Themenstellung: Wohnungsgenossenschaft

Alle Klausuraufgaben basieren auf einem gemeinsamen "Auftrag": In der Klausur soll eine Datenbank einer Wohnungsgenossenschaft, die Mietwohnungen in Mehrfamilienhäusern in verschiedenen Städten in ganz Deutschland besitzt, entworfen werden:

1. Es wird angenommen, dass Städte durch ihren Namen eindeutig identifiziert sind.
2. Zu den einzelnen im Besitz der Genossenschaft befindlichen Häusern sind Adresse (Strasse mit Hausnummer) und Stadt abgelegt. Ausserdem ist gespeichert, wieviele Wohnungen in dem Haus sind.
 - (a) Das Gebäude *Hauptstrasse 100* in *Göttingen* besteht aus 50 Wohnungen.
 - (b) Das Gebäude *Rheinstrasse 53* in *Köln* besteht aus 83 Wohnungen.
3. Jede Wohnung hat eine Nummer innerhalb des Hauses. Zu jeder Wohnung sind die Anzahl der Zimmer, die Quadratmeter und die monatliche (Kalt)Miete gespeichert.
 - (a) Die Wohnung mit der Nummer 42 des Hauses *Hauptstrasse 100* in *Göttingen* ist eine 3-Zimmer-Wohnung mit 80qm für 500 Euro im Monat.
 - (b) Die Wohnung mit der Nummer 43 desselben Hauses ist eine 2-Zimmer-Wohnung mit 50qm für 320 Euro im Monat.
 - (c) Die Wohnung mit der Nummer 17 des Hauses *Rheinstrasse 53* in *Köln* ist ein 1-Zimmer-Apartment mit 36qm und kostet 400 Euro im Monat.
4. Für jeden Mietvertrag ist gespeichert, wer welche Wohnung (und seit wann) gemietet hat (es wird angenommen, dass immer auf eine einzige Person als Mieter auftritt, d.h. den Mietvertrag unterschrieben hat). Ausserdem ist abgelegt, ob und zu welchem Zeitpunkt das Mietverhältnis gekündigt ist.
 - (a) *Karl Napf* wohnt seit 1.4.2008 in der Wohnung Nummer 42 des Hauses *Hauptstrasse 100* in *Göttingen*. Das Mietverhältnis ist nicht gekündigt.
 - (b) *Hans Dampf* wohnt seit 1.8.2001 in der Wohnung Nummer 43 des Hauses *Hauptstrasse 100* in *Göttingen*. Er hat die Wohnung zum 30.4.2011 gekündigt.
 - (c) *Lieschen Müller* wohnt seit 1.3.1980 in der Wohnung Wohnung Nr. 17 des Hauses *Rheinstrasse 53* in *Köln*. Sie hat die Wohnung zum 28.2.2011 gekündigt.
5. Wenn ein neuer Mietvertrag abgeschlossen wird, wird dabei auch die (zu dem Zeitpunkt noch aktuelle) Adresse des Mieters sowie der Beginn des Mietverhältnisses gespeichert.
 - (a) *Nils Pferd*, der zur Zeit in *Im Stall 4, Hintertupfing* wohnt (die Wohnung gehört nicht der Wohnungsgenossenschaft), hat einen Mietvertrag für die Wohnung Nummer 43 im Haus *Hauptstrasse 100* in *Göttingen* ab 1.5.2011 abgeschlossen.
 - (b) Der oben genannte *Hans Dampf* hat einen am 1.4.2011 beginnenden Mietvertrag für die Wohnung Nr. 17 im Haus *Rheinstrasse 53* in *Köln* abgeschlossen.
6. Wenn ein Mieter auszieht wird die neue Adresse gespeichert (z.B. für die Nebenkostenabrechnung des Auszugsjahres).
 - (a) Die neue Adresse von *Lieschen Müller* ist *Mühlengasse 24, Mühlhausen* in einem Haus, das nicht der Wohnungsgenossenschaft gehört.
7. Daten über alte Mietverträge werden nicht gelöscht.

Aufgabe 1 (ER-Modell [20 Punkte])

Entwickeln Sie ein ER-Modell für das Szenario. Geben Sie darin die Schlüsselattribute sowie die Beziehungskardinalitäten an.

[File Upload: pdf oder sowas, empfohlen: draw.io]

Aufgabe 2 (Transformation in das Rel. Modell (Textmode) [9 Punkte])

Bearbeiten eine der Aufgaben: Textmode (diese) oder grafisch (die nächste)

Geben Sie an, welche Tabellen (mit Attributen, Schlüsseln etc.) Ihre Datenbank enthält (keine SQL CREATE TABLE-Statements, sondern entsprechend "ASCII-Art" Pseudocode: (Empfehlung: editieren Sie es in einem lokalen File und mausen es am Ende ins Ilias, dann können Sie Ihr lokales File zur Bearbeitung der SQL-Aufgaben auch sehen).

Geben Sie die Tabellen mit jeweils mindestens zwei Beispieldupeln (z.B. denen, die sich aus dem Aufgabentext ergeben, und weiteren erfundenen) an.

- Tabellenschemata:

```
tabname(_attr1_,_attr2_,attr3,attr4)
-----
      bsp11  bsp12   bsp13  bsp14
      bsp21  bsp22   bsp23  bsp24
```

oder analog

```
relname(attr1,attr2,attr3,attr4)  PK: (attr1,attr2)
```

- Angabe von Fremdschlüsselreferenzen:

```
rel1(A,B) -> rel2(X,Y)
```

bzw. bei einelementigen Fremdschlüsseln einfach

```
rel1.A -> rel2.B
```

Aufgabe 3 (Transformation in das Rel. Modell (grafisch) [9 Punkte])

Bearbeiten eine der Aufgaben: grafisch (diese) oder Textmode (die vorhergehende)

Geben Sie an, welche Tabellen (mit Attributen, Schlüsseln etc.) Ihre Datenbank enthält (keine SQL CREATE TABLE-Statements, sondern einfach grafisch).

Markieren Sie dabei auch Schlüssel (durch unterstreichen – das ist in draw.io ganz einfach) und Fremdschlüsselreferenzen (durch überstreichen (frei malen)) oder durch Angabe von Fremdschlüsselreferenzen:

```
rel1(A,B) -> rel2(X,Y)
```

bzw. bei einelementigen Fremdschlüsseln einfach

```
rel1.A -> rel2.B
```

Aufgabe 4 (Relationales Modell CREATE TABLE [7 Punkte])

Geben Sie das CREATE TABLE-Statement für diejenige Tabelle (bzw. die Tabellen), in der bei Ihnen die Daten über die Wohnungen abgespeichert sind, so vollständig wie möglich an.

Aufgabe 5 (SQL und Relationale Algebra (a1) [2 Punkte])

Verwenden Sie für diese und die folgenden Aufgaben die von Ihnen entworfene relationale Datenbasis. Keine der Antworten soll Duplikate enthalten.

Geben Sie **eine SQL-Anfrage** an, die die Namen aller Städte ausgibt, in denen die Genossenschaft mindestens eine 4-Zimmer-Wohnung besitzt.

Aufgabe 6 (SQL und Relationale Algebra (a2) [2 Punkte])

Geben Sie **einen Algebra-Ausdruck** an, der die Namen aller Städte ausgibt, in denen die Genossenschaft mindestens eine 4-Zimmer-Wohnung besitzt.

Aufgabe 7 (SQL und Relationale Algebra (b1) [3 Punkte])

Geben Sie **eine SQL-Anfrage** an, die die Namen aller Personen ausgibt, die gegenwärtig eine 4-Zimmer-Wohnung mit mindestens 100qm für weniger als 600E gemietet haben.

Aufgabe 8 (SQL und Relationale Algebra (b2) [3 Punkte])

Geben Sie **einen Algebra-Ausdruck** an, der die Namen aller Personen ausgibt, die gegenwärtig eine 4-Zimmer-Wohnung mit mindestens 100qm für weniger als 600E gemietet haben.

Aufgabe 9 (SQL und Relationale Algebra (c) [2 Punkte])

Geben Sie **eine SQL-Anfrage** an, die für jede Stadt ausgibt, wieviel Miete die Genossenschaft insgesamt monatlich in dieser Stadt bekommen kann.

Aufgabe 10 (SQL und Relationale Algebra (d1) [3 Punkte])

Geben Sie **eine SQL-Anfrage** an, die die Namen aller Personen ausgibt, die irgendwann in einer der Wohnungen der Wohnungsgenossenschaft gewohnt haben, und die aber nie in Köln in einer Wohnung der Wohnungsgenossenschaft gewohnt haben.

Aufgabe 11 (SQL und Relationale Algebra (d2) [3 Punkte])

Geben Sie **einen Algebra-Ausdruck oder -Baum** an, der die Namen aller Personen ausgibt, die irgendwann in einer der Wohnungen der Wohnungsgenossenschaft gewohnt haben, und die aber nie in Köln in einer Wohnung der Wohnungsgenossenschaft gewohnt haben.

Aufgabe 12 (SQL und Relationale Algebra (e) [7 Punkte])

Geben Sie eine Aufgabenstellung (Textaufgabe) an, deren Lösung eine relationale Division erfordert. (Wenn ihnen nichts sinnvolles einfällt, können Sie dazu auch ein weiteres Attribut zu einer der Tabellen, bzw. eine neue Tabelle hinzufügen).

Geben Sie dazu eine SQL-Anfrage **oder** einen Algebra-Ausdruck an.

Aufgabe 13 (SQL und Relationale Algebra (f) [6 Punkte])

Gibt es eine Möglichkeit, mit SQL aus der Datenbasis alle Personen herauszufinden, die seit ihrem erstmaligen Einzug ausschliesslich, ggf. aber beliebig oft, von einer der Genossenschaft gehörenden Wohnung direkt in eine andere, auch der Genossenschaft gehörende Wohnung gezogen sind?

Falls ja, geben Sie eine solche Anfrage (SQL oder Algebra) an. Falls nein, begründen Sie Ihre Antwort.

Aufgabe 14 (SQL und Relationale Algebra (f2-upload) [6 Punkte])

Falls Sie zu der vorhergehenden Aufgabe (f) einen Algebra-Ausdruck oder -Baum als Grafik-pdf hochladen wollen, können Sie dies hier tun:

Aufgabe 15 (Abstrakt: SQL und Algebra [5 Punkte])

Gegeben sind zwei Relationen $R(A, B, C, D)$ und $S(E, F, G, H)$. Die Attribute $R.C$ und $S.G$ sind numerisch, alle anderen sind Strings.

Geben Sie einen **Algebra-Ausdruck oder -Baum** an, der äquivalent zu der folgenden SQL-Anfrage ist:

```
SELECT A, C
FROM R
WHERE (A, B) IN (SELECT E, F
                FROM S
                WHERE G < C)
```

Aufgabe 16 (Theorie-Aufgabe [5 Punkte])

Seien R und S zwei Relationen.

Geben Sie an, ob, und ggf. unter welchen Bedingungen bzgl. R und S die beiden untenstehenden Ausdrücke der relationalen Algebra äquivalent sind.

Begründen Sie Ihre Aussage.

Beschreiben Sie ggf. auch, bzw. geben Sie ein Gegenbeispiel an, wo die beiden Ausdrücke nicht äquivalent sind.

1. $R \cup S$
2. $R \bowtie S$

Aufgabe 17 (Algebra-Anfragebäume/Multiple-Choice [8 Punkte])

Diese Aufgabe benutzt das Mondial-Schema:

`country(name, code, capital, province, area, population)`

(code, capital, province) geben den Landescode+Stadtname+Provinz der Hauptstadt an

`organization(abbreviation, name, city, country, province, established)`

(city, country, province) geben die Stadt an, in der die Organisation ihren Sitz hat

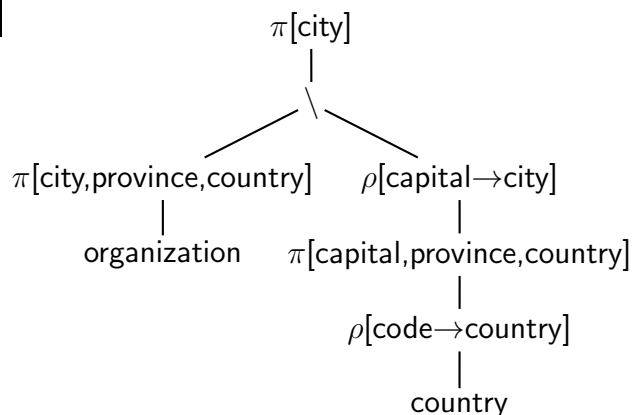
`city(name, country, province, population, latitude, longitude, elevation)`

`ismember(country, organization, type)`

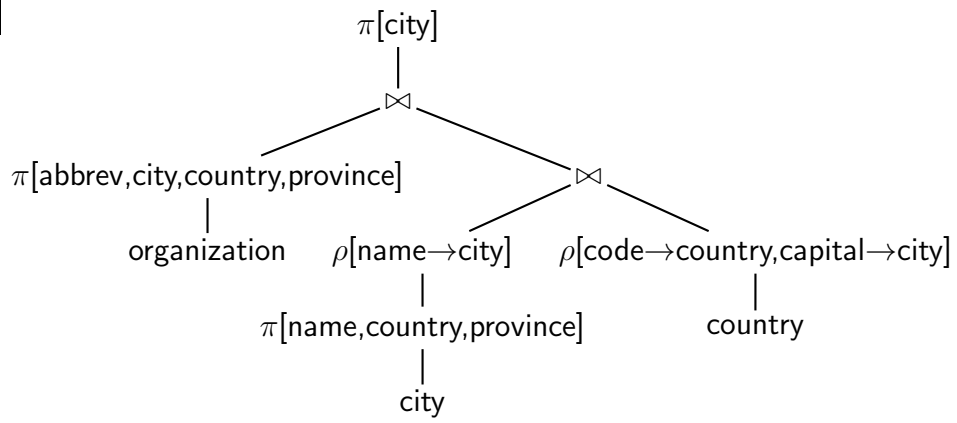
Welche der folgenden Algebra-Bäume beantworten die Anfrage “die Namen aller Städte, die die Hauptstadt eines Landes sind, und in denen eine Organisation ihren Sitz hat” korrekt?

(es können mehrere Bäume richtig sein)

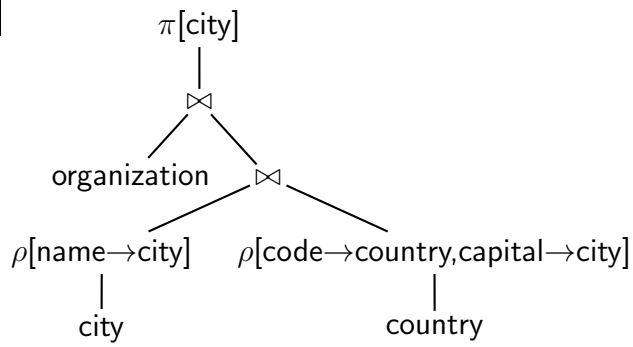
1.



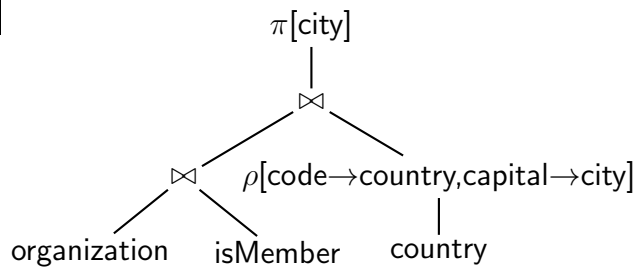
2.



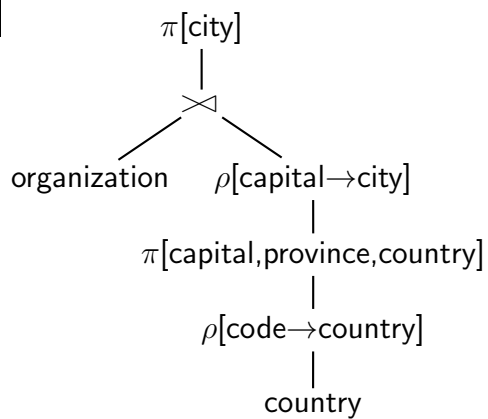
3.



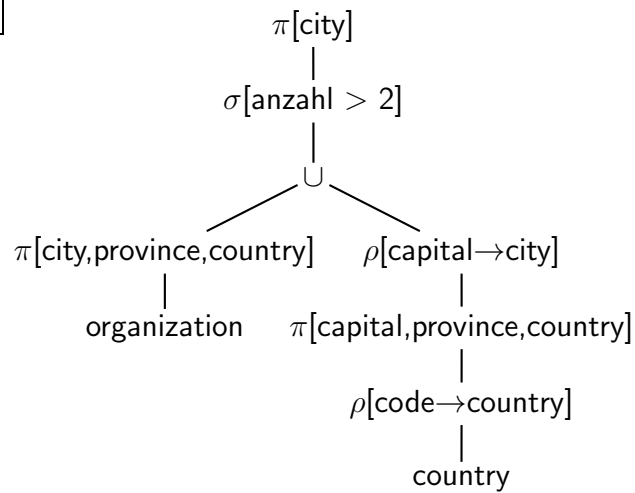
4.



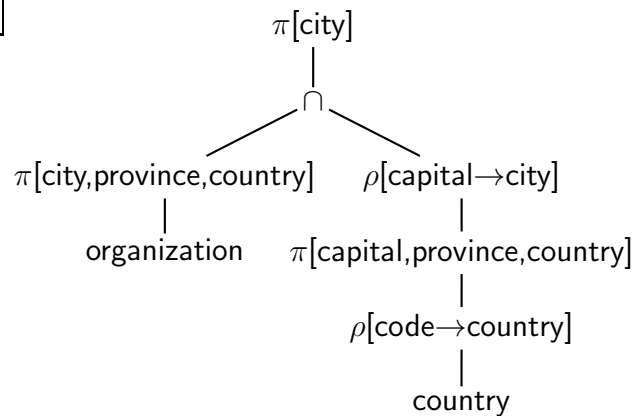
5.



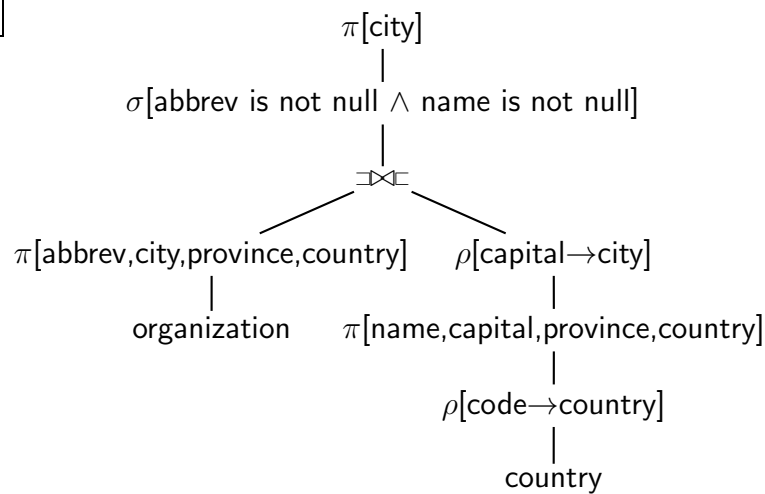
6.



7.



8.



Aufgabe 18 (Schemaentwurf/Multiple-Choice [4 Punkte])

Gegeben seien Relationen

- R(_a1_, a2),
- S(_b1_, b2),
- T(_c1_, c2), und
- X(a, b, c)

X ist offensichtlich eine dreistellige Beziehung. Für X seien sowohl A, als auch B, als auch C (einzeln!) Schlüsselkandidaten.

(Überlegen Sie sich, was man daraus schließen kann)

Es gelten die Fremdschlüsselbeziehungen $X.a \rightarrow R.a1$, $X.b \rightarrow S.b1$, $X.c \rightarrow T.c1$.

Welche der Aussagen ist/sind zutreffend (offensichtliche sich aus den Namen ergebende Fremdschlüsselbedingungen jeweils nicht explizit angegeben):

- Das angegebene Schema ist redundanzfrei (d.h. formal in 3. Normalform).
- Das Schema $R(a1, a2, b, c)$, $S(b1, b2)$, $T(c1, c2)$ ist äquivalent, redundanzfrei und verlustfrei.
- Das Schema $R(a1, a2, b, c)$, $S(b1, b2, a, c)$, $T(c1, c2, a, b)$ ist äquivalent, redundanzfrei und verlustfrei.
- Das Schema $R(a1, a2)$, $S(b1, b2)$, $T(c1, c2)$, $X(a, b)$, $Y(b, c)$, $Z(c, a)$ ist äquivalent, redundanzfrei und verlustfrei.
- Das Schema $R(a1, a2)$, $S(b1, b2)$, $T(c1, c2)$, $X(a, b)$, $Z(c, a)$ ist äquivalent, redundanzfrei und verlustfrei.
- Das Schema $R(a1, a2, b)$, $S(b1, b2, c)$, $T(c1, c2, a, b)$ ist äquivalent, redundanzfrei und verlustfrei.
- Es gibt kein anderes Schema, das äquivalent zu dem gegebenen Schema, redundanzfrei und verlustfrei ist.
- Es gibt noch weitere Schemata, die äquivalent zu dem gegebenen Schema, redundanzfrei und verlustfrei sind.

Aufgabe 19 (Ausdruckskraft - (Umordnungs-Beispiel) [4 Punkte])

Ordnen Sie die folgenden Klassen von Sprachen nach ihrer Ausdruckskraft (stärkste zuerst):

- Konjunktive Anfragen (Selektion/Projektion/Join/Renaming) "SPJR-Algebra"
- Turing-vollständige Programmiersprachen
- Relationale Algebra
- SQL-Anfragen
- nachweisbar terminierende Java-Programme

Aufgabe 20 (Transaktionen [6 Punkte])

- 1) Was bedeutet die Eigenschaft *Durability* (*Dauerhaftigkeit*) von Transaktionen?
- 2) Welche Maßnahmen werden in kommerziell eingesetzten Datenbanksystemen eingesetzt, um dies zu gewährleisten?

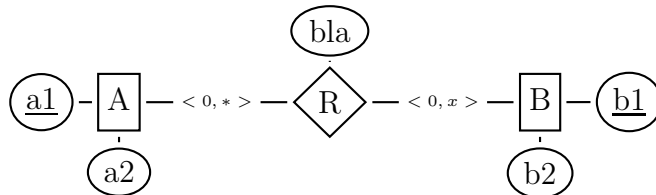
Aufgabe 21 (Abb. ER-Modell auf Relationales Modell (a) [2 Punkte])

Nehmen Sie an, dass Sie ein gutes ER-Diagramm entwickelt haben, und dabei sind, dies in ein relationales Modell umzusetzen.

In welchen Situationen benötigt man für einen Entitätstyp mehrere Relationen, um seine Informationen abzulegen?

Aufgabe 22 (Abb. ER-Modell auf Relationales Modell (b) [6 Punkte])

Gegeben sei folgendes ER-Diagramm:



Geben Sie das Tabellenschema der Tabelle zur Speicherung von R im relationalen Modell an. (2 P)

Welche Attribute sind (u.a. in Abhängigkeit von x in der Kardinalität von B bzgl. R) Schlüsselattribute dieser Relation? (4 P)