

Klausur Datenbanken
Wintersemester 2014/2015
Prof. Dr. Wolfgang May
27. Januar 2015, 12-14 Uhr
Bearbeitungszeit: 90 Minuten

Vorname:

Nachname:

Matrikelnummer:

Studiengang:

Bei der Klausur sind **keine Hilfsmittel** (Skripten, Taschenrechner, etc.) erlaubt. Handies müssen ausgeschaltet sein. Papier wird gestellt. Benutzen Sie nur die **ausgeteilten**, zusammengehefteten **Blätter** für Ihre Antworten. Schreiben Sie mit blauem/schwarzem Kugelschreiber, Füller, etc.; Bleistift ist nicht erlaubt.

Zum **Bestehen** der Klausur sind **45** Punkte hinreichend.

- meine Note soll mit Matrikelnummer so bald wie möglich auf der Vorlesungs-Webseite veröffentlicht werden.
- meine Note soll nicht veröffentlicht werden; ich erfahre sie dann aus FlexNever oder beim zuständigen Prüfungsamt.

	Max. Punkte	Schätzung für "4"
Aufgabe 1 (ER-Modell)	16	12
Aufgabe 2 (Transformation in das Relationale Modell)	20	15
Aufgabe 3 (SQL und Relationale Algebra)	39	18
Aufgabe 4 (Verschiedenes)	15	7
Summe	90	52

Note:

Themenstellung: Datenbank für Sport(Fußball)ergebnisse

Alle Klausuraufgaben basieren auf einem gemeinsamen "Auftrag": In der Klausur soll eine Datenbank zur Speicherung und Verwaltung von aktuellen und historischen Fußballergebnissen (bzw. ähnlichen sportlichen Wettbewerben) erstellt werden.

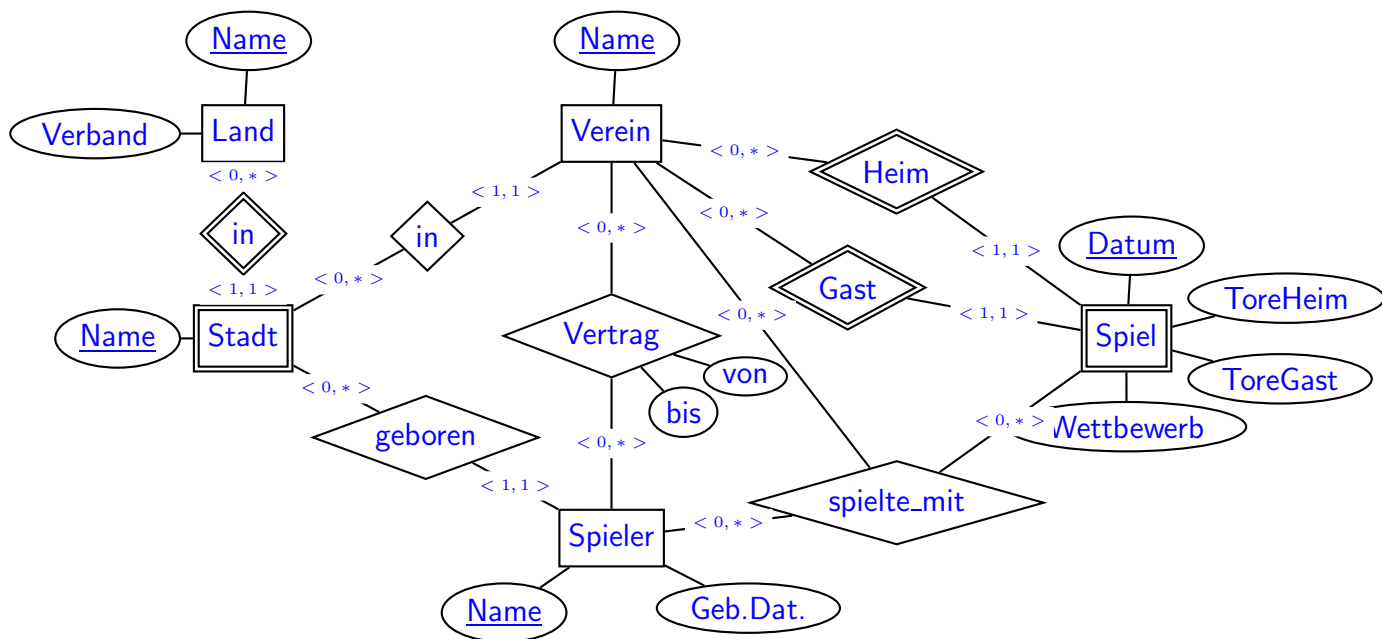
1. Es gibt viele Vereine, z.B. *Bayern München*, *1860 München*, *Bayer Leverkusen*, und *Real Madrid*. Jeder Verein hat einen Namen und ist in einer Stadt zuhause; in einer Stadt kann es mehrere Vereine geben.
Bayern München und *1860 München* sind in *München* zuhause. *Bayer Leverkusen* ist in *Leverkusen* zuhause; *Real Madrid* ist in *Madrid* zuhause.
2. Jede Stadt liegt in einem Land. In jedem Land liegen viele Städte (es wird angenommen, dass es keine zwei namensgleichen Städte in einem Land gibt, es kann aber Städte gleichen Namens in verschiedenen Ländern geben).
Die Städte *München* und *Leverkusen* liegen in *Deutschland*. *Madrid* liegt in *Spanien*.¹
3. Jedes Land gehört zu einem Kontinentalverband. Z.B. gehören *Deutschland* und *Spanien* zur *UEFA* (*Union of European Football Associations*), *Brasilien* gehört zur *CONMEBOL* (*Confederacao Sul-Americana de Futebol*).
4. Es werden Daten über nationale Ligaspiele und über internationale Wettbewerbe (UEFA Champions League, Europa League) gespeichert. Zu jedem Spiel wird gespeichert, wer gegen wen (Heimverein zuerst), wann (Datum), in welchem Wettbewerb, und mit welchem Ergebnis, gespielt hat.
Bayer Leverkusen hat gegen *Bayern München* am 5.10.2013 in der *Deutschen Bundesliga* 1:1 (d.h., unentschieden) gespielt. *Bayern München* hat in der *UEFA Champions League* am 29.4.2014 gegen *Real Madrid* 0:4 gespielt (d.h., *Bayern München* hat kein Tor geschossen, *Real Madrid* als Gastverein hat 4 Tore geschossen), d.h., verloren.
5. Für jeden Spieler ist der Name gespeichert, z.B. *Toni Kroos*, *Manuel Neuer*, oder einfach nur *Pepe* (es wird angenommen, dass es keine zwei Spieler mit demselben Namen gibt), ausserdem sein Geburtsdatum und sein Geburtsort und -land.
Toni Kroos wurde am 4.1.1990 in *Greifswald* in *Deutschland* geboren. *Pepe* wurde am 26. Februar 1983 in *Maceió* in *Brasilien* geboren. *Joachim Löw* wurde am 3.2.1960 in *Schönan* in *Deutschland* geboren (da auch historische Ergebnisse gespeichert werden, sind natürlich auch Daten über Spieler gespeichert, die schon lange nicht mehr aktiv sind).
6. Es wird gespeichert, welche Spieler in welchen Zeiträumen bei welchen Vereinen unter Vertrag standen. Ein Spieler kann mehrmals zu unterschiedlichen Zeiträumen bei demselben Verein gewesen sein.
Toni Kroos war vom 1.7.2007 bis 31.12.2008 bei *Bayern München*, vom 1.1.2009 bis 30.6.2010 bei *Bayer Leverkusen*, vom 1.7.2010 bis 30.6.2014 wieder bei *Bayern München*, und ist seit 1.7.2014 bei *Real Madrid*, wo er einen Vertrag bis 30.6.2020 hat. *Joachim Löw* hatte seinen letzten Profivertrag als Spieler beim *SC Freiburg* vom 1.7.1985 bis 30.6.1989.
7. Zu jedem Spiel ist gespeichert, welche Spieler für welchen Verein tatsächlich gespielt haben.
Bei dem o.g. Champions-League-Spiel haben unter anderem *Toni Kroos* und *Manuel Neuer* für *Bayern München* gespielt, und *Pepe* für *Real Madrid*.

¹"Mailand oder Madrid, Hauptsache Italien" (Andreas Möller, 1992)

Aufgabe 1 (ER-Modell [16 Punkte])

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

Lösung



Alternative Modellierungen und Kommentare:

- Bei dem Entitätstyp "Spiel" würde Datum als Schlüsselattribut sowie "Heim" als einziger identifizierende Beziehung ausreichen, da ein Team an einem Tag nur ein (Pflicht)Spiel macht.
Die obige Modellierung mit "Heim" und "Gast" als identifizierende Beziehungen ist aber intuitiver ("das Spiel zwischen Bayern München und Real Madrid am 29.4.2014"), und erleichtert oft auch das Arbeiten mit der Datenbank.
- Für die Beziehung "spielt_mit" wurde ebenfalls eine intuitive Modellierung als dreistellige Beziehung gewählt ("Toni Kroos spielte für die Bayern bei dem Spiel zwischen Bayern München und Real Madrid am 29.4.2014"), die direkt der Formulierung in der Aufgabenstellung entspricht.
Korrekt und ausreichend wäre auch eine binäre Beziehung zwischen Spieler und Spiel ("Toni Kroos spielte bei dem Spiel zwischen Bayern München und Real Madrid am 29.4.2014 mit"), da man über die Vertragsdaten herausfinden kann, dass er zu dieser Zeit bei Bayern München unter Vertrag stand. (Hinweis: dies ist ähnlich der in der Vorlesung/Aufgaben besprochenen Aufspaltung einer dreistelligen Beziehung in zwei zweistellige, wobei allerdings kein einfaches Join zugrundeliegt, sondern das Spieldatum und Datumsarithmetik verwendet werden muss).
- Man kann die Kardinalität von "Spiel" in "spielt_mit" auch durch <22,28> spezifizieren, um auszudrücken, dass es 2x11 Spieler sind, und dreimal ausgewechselt werden darf. Dies ist aber eher eine zu strenge und unflexible Modellierung (und für alle anderen Sportarten unpassend).

U.a. würde eine Mindestkardinalität es verbieten, Spieldaten bereits vorher anzulegen und Ergebnis und Mitspielende erst nach dem Spiel einzutragen.

- Statt "ToreHeim" und "ToreGast" kann man "Ergebnis" (ggf wiederum als strukturiertes Attribut) machen. Dies erschwert aber nachher ggf. Anfragen nach Gewinnern und Verlierern (weshalb keine der Anfragen in Aufgabe 3 dieses benutzt).
- Hinweis: Das Attribut Vertrag.bis ist nicht als Schlüsselattribut unterstrichen, da im ER-Diagramm Beziehungen keine Schlüssel haben! Diese werden erst im ER-Diagramm bestimmt (vgl. Übungsaufgaben).

Aufgabe 2 (Transformation in das Relationale Modell [20 Punkte])

- a) Lösen Sie diesen Aufgabenteil auf dem *letzten* Blatt und trennen dieses ab (und geben es am Ende mit ab!). Dann haben Sie dieses Blatt separat zugreifbar um später damit die Aufgaben 2b, 3 und 4 (SQL, Relationale Algebra+SQL, Diverses) zu lösen.

Geben Sie an, welche Tabellen (mit Attributen, Schlüsseln etc.) Ihre Datenbank enthält (keine SQL CREATE TABLE-Statements, sondern einfach grafisch). (12 P)

Markieren Sie dabei auch Schlüssel (durch unterstreichen) und Fremdschlüssel (durch überstreichen).

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

Lösung

Land	
<u>Name</u>	Verband
Deutschland	UEFA
Spanien	UEFA
Brasilien	CONMEBOL
:	:

Verein		
<u>Name</u>	Stadt	Land
Bayern München	München	Deutschland
1860 München	München	Deutschland
Real Madrid	Madrid	Spanien
:	:	:

Spieler			
<u>Name</u>	GebDat	GebOrt	GebLand
Toni Kroos	4.1.1990	Greifswald	Deutschland
Pepe	26.2.1983	Maceió	Brasilien
Joachim Löw	3.2.1960	Schönau	Deutschland
:	:	:	:

Vertrag			
<u>Spieler</u>	Verein	<u>von</u>	bis
Toni Kroos	Bayern München	1.7.2007	31.12.2008
Toni Kroos	Bayer Leverkusen	1.1.2009	30.6.2010
Toni Kroos	Bayern München	1.7.2010	30.6.2014
Toni Kroos	Real Madrid	1.7.2014	30.6.2020
Joachim Löw	SC Freiburg	1.7.1985	30.6.1989
:	:	:	:

Spiel					
<u>Heim</u>	<u>Gast</u>	<u>Datum</u>	Wettbew	ToreHeim	ToreGast
Bayer Leverkusen	Bayern München	5.10.2012	Dtsch.Bundesliga	1	1
Bayern München	Real Madrid	29.4.2014	UEFA Champions L.	0	4
:	:	:	:	:	:

spielte				
<u>Heim</u>	<u>Gast</u>	<u>Datum</u>	<u>Spieler</u>	fürVerein
Bayern München	Real Madrid	29.4.2014	Toni Kroos	Bayern München
Bayern München	Real Madrid	29.4.2014	Manuel Neuer	Bayern München
Bayern München	Real Madrid	29.4.2014	Pepe	Real Madrid
:	:	:	:	:

Alternative Modellierungen und Kommentare:

- Es gibt keine Tabelle "Stadt(Name,Land)" und entsprechende Foreign Keys darauf: Bei strenger Übersetzung aus dem ER-Modell wäre sie notwendig. Allerdings wird sie nie wirklich gebraucht, weil Städte in dieser Anwendung keine Rolle spielen. Ein Nachteil, wenn man die Tabelle und die Foreign Keys hätte, wäre, dass man nicht nur alle "grossen" Städte², in denen es Vereine gibt, sondern auch alle Geburtsstädte der Spieler eintragen muss.
 - In "Vertrag" gehört das Attribut "bis" nicht zum Schlüssel.
 - In "Vertrag" gehört das Attribut "Verein" nicht zum Schlüssel, da (Spieler, von) ausreichen.
 - In der Tabelle "Spiel" würde prinzipiell auch "Heim" und "Datum" als Schlüssel ausreichen (vgl. Anmerkung zum ER-Modell); entsprechend muss dann auch der Schlüssel von "spielte" gewählt werden.
 - In der Tabelle "spielte" ist die Spalte "Verein" nicht unbedingt notwendig, da man dies im Prinzip aus den Verträgen schließen könnte (vgl. Anmerkung zum ER-Modell).
 - Die "Minimalmodellierung", wo bei "Spiel" nur (Datum, Heim) Schlüssel sind, und bei "spielte" im ER-Modell die Beziehung zu "Verein" (fürVerein) weggelassen ist, würde zu der Tabelle spielte(Spieler, Datum, Heim) führen, in der z.B. das Tupel (Pepe, 29.4.2014, Bayern München) enthalten ist (was für den Datenbestand vollkommen ausreicht, aber nicht direkt ahnen läßt, dass Pepe für Real Madrid auf dem Platz stand).
- b) Geben Sie das CREATE TABLE-Statement für diejenige Tabelle, in der die Informationen über die Spielergebnisse abgespeichert sind, so vollständig wie möglich an (verwenden Sie u.a. den Datentyp DATE; 8 P).

Lösung

```
CREATE TABLE Spiel                                     Basis 4P      + Constraints:
( Heim          VARCHAR2(30) REFERENCES VEREIN(name),   1/2P
  Gast          VARCHAR2(30) REFERENCES VEREIN(name),   1/2P
  Datum         DATE NOT NULL,                          1/2P
  Wettbew       VARCHAR2(30) NOT NULL,                  1/2P
  ToreHeim      NUMBER CHECK (ToreHeim >=0),           1/2P
  ToreGast      NUMBER CHECK (ToreGast >=0),           1/2P
  CONSTRAINT SpielKey PRIMARY KEY (Heim, Gast, Datum)) 1P
```

ToreHeim, ToreGast NOT NULL würde dazu führen, dass man keine Spiele im Voraus anlegen kann (nicht als falsch bewertet).

²hmmja, und sowas wie Hoffenheim

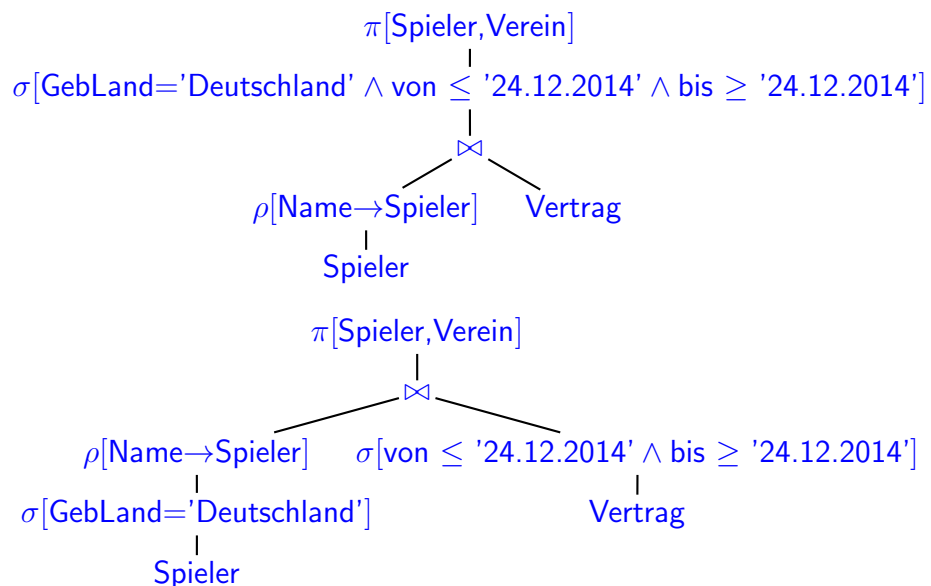
Aufgabe 3 (SQL und Relationale Algebra [39 Punkte])

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

- a) Geben Sie eine **SQL-Anfrage** und einen **Algebra-Ausdruck** oder **-Baum** an, die für alle in Deutschland geborenen Spieler, die am 24.12.2014 bei einem Verein unter Vertrag standen, ihren Namen sowie den Namen dieses Vereins angibt. (2.5+2.5P)

Lösung

```
SELECT Spieler.name, Vertrag.Verein
  -- kein DISTINCT, jeder Spieler hat nur einen aktuellen Vertrag
FROM Spieler, Vertrag
WHERE Spieler.name = Vertrag.Spieler
  AND Spieler.GebLand = 'Deutschland'
  AND von <= '24.12.2014' and bis >= '24.12.2014'
  -- alternativ AND '24.12.2014' BETWEEN von AND bis
```



- b) Geben Sie eine **SQL-Anfrage** an, die für jeden Spieler, der mindestens einmal bei einem *UEFA Champions League*-Spiel mitgespielt hat, angibt, wieviele *UEFA Champions League*-Spiele er schon gemacht hat. (4 P)

Lösung

```
SELECT spielte.Spieler, COUNT(*)
FROM spielte x, Spiel y
WHERE spielte.heim = Spiel.heim AND spielte.gast = Spiel.gast
  AND spielte.datum = Spiel.datum
  AND Spiel.Wettbew = 'UEFA Champions League'
GROUP BY spielte.Spieler
```

- c) Geben Sie **eine SQL-Anfrage und einen Algebra-Ausdruck oder -Baum** an, die die Namen aller Spieler ausgeben, die mindestens einmal für einen Verein bei einem *UEFA Champions League*-Spiel gegen einen Verein angetreten sind, bei dem sie vorher selber einmal unter Vertrag gestanden haben.
(dies wäre z.B. der Fall, wenn *Toni Kroos* demnächst mit *Real Madrid* gegen seinen ehemaligen Verein *Bayern München* spielen würde.) (4+4P)

Lösung

```

SELECT DISTINCT spielte.Spieler
FROM spielte, Spiel, Vertrag
  -- spielte: das betrachtete Spiel;
  -- Vertrag: der ehemals existierende Vertrag beim aktuellen Gegner
WHERE spielte.Spieler = Vertrag.Spieler
  AND spielte.Heim = Spiel.Heim
  AND spielte.Gast = Spiel.Gast
  AND spielte.Datum = Spiel.Datum
  AND Spiel.Wettbewerb = 'UEFA Champions League'
  AND spielte.Datum > Vertrag.bis
  AND ((spielte.fuerVerein = Spiel.Heim
        AND Spiel.Gast = Vertrag.Verein)
       OR
        (spielte.fuerVerein = Spiel.Gast
         AND Spiel.Heim = Vertrag.Verein))

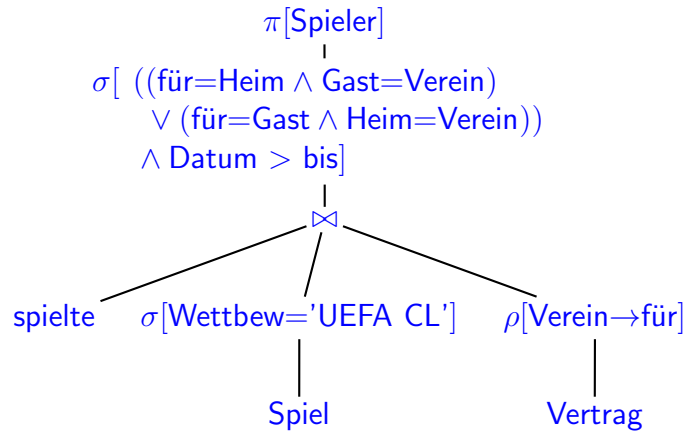
```

Dasselbe mit der Minimalmodellierung `spielte(Spieler, Datum, Heim)`:
 * Gegnerverein des Spiels aus `Spiel` nehmen (einfach)
 * fuer wen er spielte aus `Vertraegen` raussuchen:

```

SELECT DISTINCT spielte.Spieler
FROM spielte, Spiel, Vertrag fruehV, Vertrag aktV
WHERE spielte.Spieler = fruehV.Spieler
  AND spielte.Heim = Spiel.Heim
  AND spielte.Datum = Spiel.Datum
  AND Spiel.Wettbewerb = 'UEFA Champions League'
  AND spielte.Datum > .bis
  -- aktuellen Vertrag raussuchen, kann nur einer sein
  AND spielte.Spieler = aktV.Spieler
  AND Spiel.Datum BETWEEN aktV.von and aktV.bis
  AND ((aktV.Verein = Spiel.Heim
        AND Spiel.Gast = Vertrag.Verein)
       OR
        (aktV.fuerVerein = Spiel.Gast
         AND Spiel.Heim = Vertrag.Verein))

```

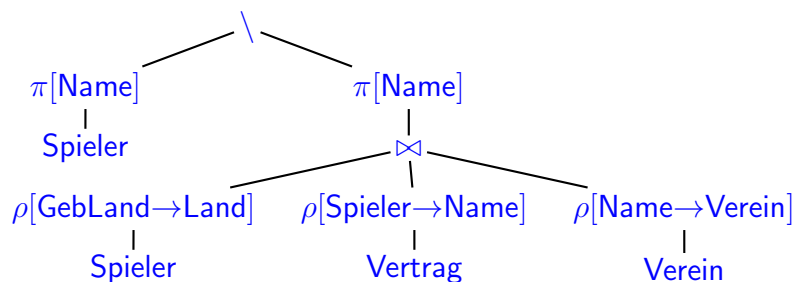



- d) Geben Sie **eine SQL-Anfrage und einen Algebra-Ausdruck oder -Baum** an, die die Namen aller Spieler ausgeben, die nie einen Vertrag bei einem Verein in ihrem Geburtsland hatten. (4+4P)

Lösung

```
SELECT name
FROM Spieler
WHERE NOT EXISTS
  (SELECT *
   FROM Vertrag, Verein
   WHERE Vertrag.Spieler = Spieler.Name
     AND Vertrag.Verein = Verein.Name
     AND Verein.Land = Spieler.GebLand)
```

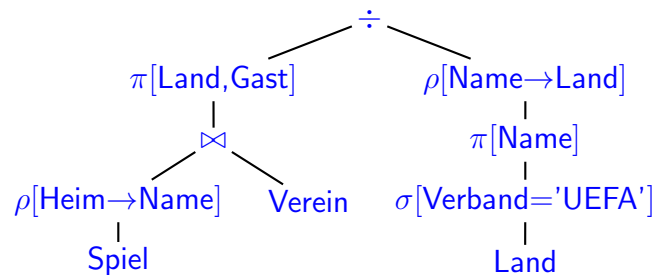
```
(SELECT Name
 FROM Spieler)
MINUS
(SELECT Spieler.Name -- Spieler, die mal einen Vertrag im Geburtsland hatten
 FROM Spieler, Vertrag, Verein
 WHERE Spieler.Name = Vertrag.Spieler
   AND Vertrag.Verein = Verein.Name
   AND Verein.Land = Spieler.GebLand)
```



- e) Geben Sie **eine SQL-Anfrage und einen Algebra-Ausdruck oder -Baum** an, die die Namen aller Vereine ausgibt, die schon in allen der UEFA angehörenden Ländern (d.h., in einem Spiel bei irgendeinem Verein jedes Landes) mindestens einmal als Gastverein angetreten sind. (5+5 P)

Lösung

```
SELECT name
FROM Verein v1
WHERE NOT EXISTS (SELECT Name
                  FROM Land l
                  WHERE Verband = 'UEFA'
                  AND NOT EXISTS (SELECT *
                                FROM Verein v2, Spiel s
                                WHERE v2.land = l.land
                                       AND v2.name = Spiel.Heim
                                       AND v1.name = Spiel.Gast))
```



- f) **Etwas Theorie:** Gegeben sind zwei beliebige Relationen $R(\bar{X}, \bar{Y})$ und $S(\bar{Y})$. Sei $cond(\bar{X})$ eine Bedingung, die nur Attribute aus \bar{X} verwendet.

Beweisen oder widerlegen Sie, ob die Gleichung

$$\sigma[cond(\bar{X})](R \div S) = (\sigma[cond(\bar{X})](R)) \div S$$

allgemeingültig ist. Begründen Sie Ihre Aussage. (4 P)

Lösung Sie gilt.

- Intuitiv: wenn man R durch S dividiert, und einen nur $R(\bar{X})$ interessieren, die $cond(\bar{X})$ erfüllen, kann man auch vor der Division auf diese einschränken.
- Begründung: Die linke Seite beschreibt alle diejenigen Tupel t über $\pi[X](R)$, die (i) mit jedem \bar{Y} -Tupel aus S in der Beziehung R stehen, und ("danach") die Bedingung (ii) $cond(\bar{X})$ erfüllen.

Vertausche die beiden Teilbedingungen des im obigen Satz hervorgehobenen *und*: Dies sind alle diejenigen Tupel t über $\pi[X](R)$, die (ii) die Bedingung $cond(\bar{X})$ erfüllen (d.h., $t \in \sigma[cond(\bar{X})](\pi[X](R))$), und (i) die mit jedem \bar{Y} -Tupel aus S in der Beziehung R stehen.

Zu (ii): $t \in \sigma[cond(\bar{X})](\pi[X](R))$ ist genau dann der Fall, wenn auch das Tupel r in R , aus dem t durch Projektion auf \bar{X} entsteht, $\sigma[cond(\bar{X})]$ erfüllt, d.h. $r \in \sigma[cond(\bar{X})](R)$.

Nimmt man (i) jetzt wieder dazu, sieht man, dass die durch die linke Seite beschriebene Menge gleich $(\sigma[\text{cond}(\bar{X})](R)) \div S$ ist.

– Formaler Beweis:

Linke Seite mathematisch ausformulieren:

$$\begin{aligned}\sigma[\text{cond}(\bar{X})](R \div S) &= \sigma[\text{cond}(\bar{X})](\{t \in \text{Tup}(\bar{X}) \mid \{t\} \times S \subseteq R\}) \\ &= \{t \in \text{Tup}(\bar{X}) \mid \{t\} \times S \subseteq R \text{ und } t \text{ erfüllt } \text{cond}\}\end{aligned}$$

Rechte Seite genauso, R ist über $\bar{X}\bar{Y}$:

$$\begin{aligned}(\sigma[\text{cond}(\bar{X})](R)) \div S &= (\{u \in R(\bar{X}\bar{Y}) \mid u \text{ erfüllt } \text{cond}\}) \div S \\ &\quad \text{da } \text{cond} \text{ nur } \bar{X} \text{ betrifft:} \\ &= (\{u \in R(\bar{X}\bar{Y}) \mid u[\bar{X}] \text{ erfüllt } \text{cond}\}) \div S \\ &\quad \text{Division links auswerten, von } \bar{X}\bar{Y} \text{ nach } \bar{X} \text{ runter} \\ &= \{t \in \text{Tup}(\bar{X}) \mid t \text{ erfüllt } \text{cond} \text{ und } \{t\} \times S \subseteq R\}\end{aligned}$$

– Andere formale Lösung:

\supseteq : trivial: cond betrachtet einzelne Tupel, und nur den \bar{X} -Anteil, während \div "zählt". Zuerst (linke Seite) \div zu machen, ergibt also auf jeden Fall eine Obermenge.

Das "Risiko" ist also nur, dass die rechte Seite eine echte Teilmenge ist.

\subseteq : $\sigma[\text{cond}(\bar{X})](R)$ behält alle Tupel, die cond erfüllen, und entfernt nur solche, die es nicht erfüllen. D.h., auch $(\sigma[\text{cond}(\bar{X})](R)) \div S$ unterscheidet sich von $R \div S$ nur dadurch, dass Ergebnisse fehlen, die cond nicht erfüllen, diese sind aber in der linken Seite auch weg.

Diese Überlegung zeigt, dass es effizienter ist, erst die Bedingung auszuwerten, und dann die (komplexere) Division (nur noch für die verbleibenden Kandidaten $\subseteq R$) auszuwerten.

(Beispiel aus einer realistischen Anwendung: "Alle Produkte, die von allen Kunden in Stadt S bezogen werden, und über 1000 EUR kosten" besser als "erstmal alle Produkte, die über 1000 EUR kosten, auswählen, und dann für diese prüfen, ob sie von allen Kunden in Stadt S bezogen werden" auswerten).

Aufgabe 4 (Verschiedenes [15 Punkte])

- a) Am 27.1.2015 stellt sich heraus, dass *Toni Kroos* mit sofortiger Wirkung seinen laufenden Vertrag auflöst und zu *Borussia Dortmund*³ wechselt, und dort einen Vertrag bis 30.6.2018 bekommt.

Welche Änderungen müssen auf Ihrer Datenbank ausgeführt werden? Geben Sie geeignete SQL-Statements an (5 P).

Lösung Das neue Ende-Datum in den laufenden Vertrag von Toni Kroos eintragen, und den neuen Vertrag bei Borussia Dortmund einfügen:

```
UPDATE Vertrag
SET bis = '27.1.2015'
WHERE Spieler = 'Toni Kroos'
    AND bis >= '27.1.2015'                -- OR bis is NULL          3P
```

```
UPDATE Vertrag
SET bis = '27.1.2015'
WHERE Spieler = 'Toni Kroos'
    AND bis = (SELECT MAX(bis)
               FROM Vertrag
               WHERE Spieler = 'Toni Kroos') -- OR bis IS NULL
```

```
INSERT INTO Vertrag
VALUES('Toni Kroos', 'Bor. Dortmund', '27.1.2015', '30.6.2018'); 2P
```

- b) Jemand hat für einen Teil der Datenbank die folgende Tabelle mit der Bedeutung

“Welcher Spieler ist wann und wo geboren, und hat ab wann bei welchem Verein in welchem Land gespielt”

angelegt:

Spieler					
Spieler	Geb.Dat.	Geb.Ort	Verein	Land	ab
Toni Kroos	4.1.1990	Greifswald	Bayern München	Deutschland	1.7.2007
Toni Kroos	4.1.1990	Greifswald	Bayer Leverkusen	Deutschland	1.1.2009
Toni Kroos	4.1.1990	Greifswald	Bayern München	Deutschland	1.7.2010
Toni Kroos	4.1.1990	Greifswald	Real Madrid	Spanien	1.7.2014
Pepe	26.2.1983	Maceió	Real Madrid	Spanien	1.7.2007
Joachim Löw	3.2.1960	Schönau	SC Freiburg	Deutschland	1.7.1985
:	:	:	:	:	:

- i) Welche grundlegenden funktionalen Abhängigkeiten bestehen in dieser Tabelle? (4 P)

Lösung

³in der Winterpause 2014/15 stand Dortmund auf dem vorletzten Platz in der Deutschen Bundesliga.

Spieler → Geb.Dat, Verein → Land,
Spieler → Geb.Ort,
(Spieler,ab) → Verein.
und natürlich noch weitere sich daraus ergebende, z.B. (Spieler,Verein) → Geb.Dat.

- ii) Welche Attribute bilden einen möglichen Primärschlüssel für diese Tabelle?
(2 P)

Lösung (Spieler, ab)

- iii) Welche Probleme hat dieser Entwurf (kurz, Stichworte)? (4 P)

Lösung (Je nach Sichtweise sind unterschiedliche Formulierungen möglich)

- * Geburtsdatum und -ort für jeden Spieler ist mehrfach abgelegt
(praktische Konsequenz: Update-Anomalien bei Geburtsdatum und -ort)
- * das Geburtsland fehlt (Stadt.Name ist auch kein alleiniger Schlüssel von Stadt,
also schon bzgl. der Modellierung unsinnig), die "Land"-Spalte bezieht sich auf
den Verein.
- * Zu jedem Verein das Land hier noch einmal abzulegen, ist nutzlos (zum einen
muss zum einen noch einmal in einer Tabelle sein, wo zu den Vereinen die Stadt
abgelegt ist, zum anderen ist z.B. die Information, dass Bayern München ein
deutscher Verein ist, mehrfach gespeichert);
(praktische Konsequenz: Update-Anomalien bei Verein→Land),
- * das Vertragsende-Datum fehlt. Man kann es oft aus dem Beginn des nächsten
Vertrages schliessen, aber man könnte nicht mehr feststellen, ob Spieler ihre
Karriere schon beendet haben, z.B. Joachim Löw.
- * es sind mehrere funktionale Abhängigkeiten enthalten, die nicht vom komplet-
ten Schlüssel abhängig sind.

Diese Tabelle ist ein Teil des Joins aus Spieler ⋈ Vertrag ⋈ Verein.