

12.12.2023

Übungsblatt 9 – Lösungsvorschlag

Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

a) Funktionale Abhängigkeiten:

Gegeben sei Relation $R(A_1, \dots, A_n)$ und $\alpha, \beta \in \{A_1, \dots, A_n\}$. Eine Attributkombination β heißt funktional abhängig von α , wenn in jedem möglichen Tupel von R die Werte von β durch die von α eindeutig bestimmt sind. In Zeichen: $\alpha \rightarrow \beta$. Das bedeutet im Umkehrschluss: falls es mehrere Tupel in R gibt, die gleiche α -Werte haben, dann müssen auch die β -Werte übereinstimmen.

Prüfen Sie, ob folgende funktionale Abhängigkeiten auf den Daten in der Tabelle gelten:

A	B	C	D	E	F	G
1	x	S	1.4	ja	101	klein
2	y	M	2.4	nein	102	groß
3	x	XL	1.5	ja	101	mittel
4	z	L	1.4	ja	105	klein
5	z	M	0.0	ja	110	mittel

- a) Gilt $A \rightarrow E$? d) Gilt $AB \rightarrow D$?
b) Gilt $B \rightarrow D$? e) Gilt $EG \rightarrow C$?
c) Gilt $G \rightarrow E$?

Lösung



- a) Gilt.
b) Gilt nicht, da für z unterschiedliche Werte vorhanden sind.
c) Gilt.
d) Gilt (allein dadurch überprüfbar, dass A eindeutig ist).
e) Gilt nicht, da für, z.B. ja/klein, verschiedene Werte bei C (S, L) vorhanden sind.

b) Theorie: Diskutieren Sie folgende Fragen zu Schlüsseln:

- Bei der Modellierung von Personen könnte z.B. die Sozialversicherungsnummer als Schlüssel verwendet werden. Welche funktionalen Abhängigkeiten ergeben sich daraus? Zählen Sie einige Beispiele auf.

Lösung

Von der Sozialversicherungsnummer kann eindeutig auf alle Attribute der Person z.B. auf den Namen oder das Geburtsdatum der Person geschlossen werden.

- Wie unterscheiden sich funktionale Abhängigkeit und voll funktionale Abhängigkeit?

Lösung

Eine funktionale Abhängigkeit $\alpha \rightarrow \beta$ bedeutet, dass für alle vorhandenen Tupel einer Relation R die Werte von β eindeutig durch α bestimmt sind (d.h. von α abgeleitet werden können). Voll funktionale Abhängigkeit ist hingegen strenger, da eine Attributkombination β nur dann als voll funktional abhängig von α gilt, wenn α nicht verkleinerbar ist (d.h. es keine echte Teilmenge von α gibt, von der β funktional abhängig ist).

- Warum erfüllt eine Relation in 1NF, die nur einelementige Schlüsselkandidaten besitzt, automatisch die 2NF?

Lösung

Die 2NF wird erfüllt, da kein Nichtschlüsselattribut von einem Teil eines Schlüssels abhängen kann (die Schlüssel sind einelementig und können nicht mehr verkleinert werden).

- c) **NF-Bestimmung:** Gegeben seien die abstrakten Relationenschemata R über die Attribute A, B, C, D, E mit den zugehörigen funktionalen Abhängigkeiten FA . Sie können dabei davon ausgehen, dass alle Attributwerte atomar sind.

Bestimmen Sie für jedes Schema alle möglichen Schlüsselkandidaten und prüfen Sie, ob die 1NF, 2NF, 3NF oder BCNF vorliegt.

- $R(A, B, C, D, E)$ mit $FA = \{DE \rightarrow AC, B \rightarrow ADE\}$

Lösung

$SK = \{B\}$

\mathcal{R} befindet sich in der 2. NF, da die folgende NF durch die FA $DE \rightarrow AC$ verletzt wird.

- $R(A, B, C, D, E)$ mit $FA = \{BC \rightarrow AE, E \rightarrow A, AC \rightarrow BD\}$

Lösung

$SK = \{CE, BC, AC\}$

\mathcal{R} befindet sich in der 3. NF, da die folgende NF durch die FA $E \rightarrow A$ verletzt wird.

- $R(A, B, C, D, E)$ mit $FA = \{AE \rightarrow BC, CE \rightarrow ABD\}$

Lösung

$$SK = \{CE, AE\}$$

\mathcal{R} befindet sich in der BCNF (auf der linken Seite stehen nur Superschlüssel).

Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)**Aufgabe 1 (Normalformenbestimmung)****[3 Punkte]**

Gegeben sei das Schema $R(A, B, C, D, E)$ mit den zugehörigen funktionalen Abhängigkeiten FA . Bestimmen Sie für jedes Schema alle möglichen Schlüsselkandidaten und prüfen Sie, ob die 1NF, 2NF, 3NF oder BCNF vorliegt und begründen Sie Ihre Entscheidung. Sie können dabei davon ausgehen, dass alle Attributwerte atomar sind.

- a) 0.75 Punkte $FA = \{ABD \rightarrow C, A \rightarrow E, E \rightarrow A\}$

Lösung

$$SK = \{ABD, BDE\}$$

3NF, da z.B. $A \rightarrow E$ die BCNF verletzt

Abgabe

1a.txt

- b) 0.75 Punkte $FA = \{BD \rightarrow E, CE \rightarrow BD, BC \rightarrow AD, A \rightarrow BCE\}$

Lösung

$$SK = \{CE, BC, A\}$$

3NF, da z.B. $BD \rightarrow E$ die BCNF verletzt

Abgabe

1b.txt

- c) 0.75 Punkte $FA = \{AE \rightarrow C, C \rightarrow AB, D \rightarrow CE\}$

Lösung

$$SK = \{D\}$$

2NF, da z.B. $C \rightarrow AB$ die 3NF verletzt

Abgabe

1c.txt

d) 0.75 Punkte $FA = \{A \rightarrow B, BC \rightarrow A\}$


Lösung



$SK = \{ACDE, BCDE\}$ (E und D müssen dabei sein, da sie bei keiner FA vorkommen)
3NF, da z.B. $A \rightarrow B$ die BCNF verletzt

Abgabe



 1d.txt

Aufgabe 2 (Algorithmen)

[3 Punkte]

a) 1.5 Punkte **Kanonische Überdeckung:** Gegeben sei das Schema $R(A, B, C, D, E)$ mit den funktionalen Abhängigkeiten $FA = \{A \rightarrow BCDE, BC \rightarrow DE, C \rightarrow AD, AB \rightarrow BE\}$.

Berechnen Sie die kanonische Überdeckung.

Lösung



- 1) Linksreduktion:
 $BC \rightarrow DE$ reduzieren zu $C \rightarrow DE$
 $AB \rightarrow BE$ reduzieren zu $A \rightarrow BE$
gesamt: $A \rightarrow BCDE, C \rightarrow DE, C \rightarrow AD, A \rightarrow BE$
- 2) Rechtsreduktion:
 $A \rightarrow BCDE$ reduzieren zu $A \rightarrow C$
 $C \rightarrow DE$ reduzieren zu $C \rightarrow \{\}$ (alles rechts ist überflüssig)
gesamt: $A \rightarrow C, C \rightarrow \{\}, C \rightarrow AD, A \rightarrow BE$
- 3) Eliminierung von $C \rightarrow \{\}$: $A \rightarrow C, C \rightarrow AD, A \rightarrow BE$
- 4) Zusammenfassen von der A-Regeln zu $A \rightarrow BCE$
- 5) kanonische Überdeckung ist somit: $A \rightarrow BCE, C \rightarrow AD$

Abgabe



 2a.pdf (samt Zwischenschritte)

b) 1.5 Punkte **Relationensynthese:** Gegeben sei das Schema $R(A, B, C, D, E, F, G, H)$ mit der kanonischen Überdeckung $F_C = \{A \rightarrow DF, G \rightarrow B, C \rightarrow AH, E \rightarrow G, B \rightarrow C\}$.

Berechnen Sie eine verlustfreie und abhängigkeitsbewahrende Relationenzerlegung von R in 3NF mithilfe der Relationensynthese.

Lösung



- 1) (Schritt 1) kanonische Überdeckung bereits gegeben
- 2) (Schritt 2) erzeuge Relation $R_1(A, D, F)$ mit $F_1 = \{A \rightarrow DF\}$

- 3) (Schritt 3) erzeuge Relation $R_2(B, G)$ mit $F_2 = \{G \rightarrow B\}$
- 4) (Schritt 4) erzeuge Relation $R_3(A, C, H)$ mit $F_3 = \{C \rightarrow AH\}$
- 5) (Schritt 5) erzeuge Relation $R_4(E, G)$ mit $F_4 = \{E \rightarrow G\}$
- 6) (Schritt 6) erzeuge Relation $R_5(B, C)$ mit $F_5 = \{B \rightarrow C\}$
- 7) (Schritt 7) Schlüsselkandidat E , daher keine weitere Relation nötig.

Abgabe



2b.pdf (samt Zwischenschritte)

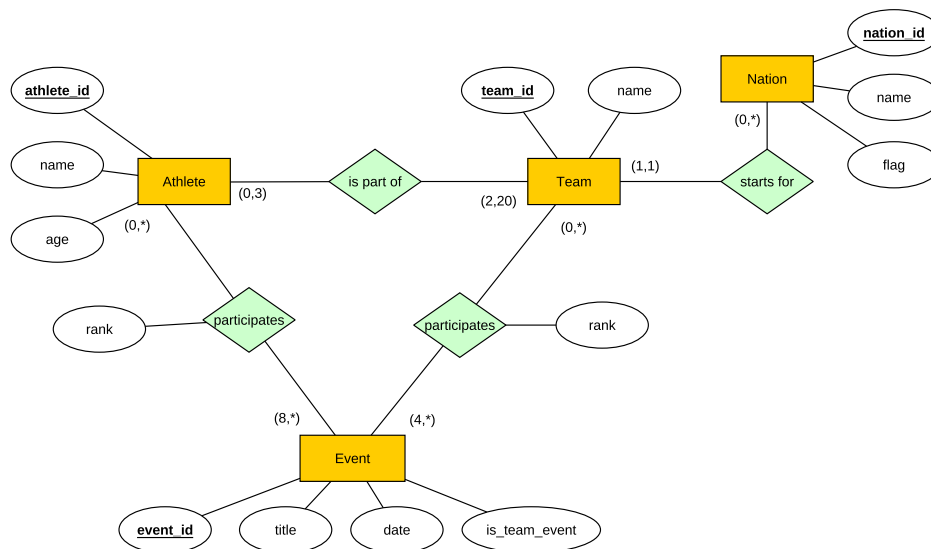
Aufgabe 3 (Trigger)

[4 Punkte]

Trigger erlauben es uns, in einer relationalen Datenbank ereignisabhängig Logik auszuführen, wodurch komplexere Einschränkungen überprüft werden können.

Ziehen Sie für die Umsetzung die PostgreSQL-Dokumentation zu Trigger¹ und Funktionen² zu rate.

- a) 1 Punkt Legen Sie für folgendes ER-Diagramm eine Datenbank in PostgreSQL an. Achten Sie darauf, auch Fremdschlüsselbeziehungen etc. mit anzulegen und notwendige Mapping-Tabellen für n:m-Beziehungen zu erstellen.



Abgabe



3a.sql (create-Statements)

¹<https://www.postgresql.org/docs/15/sql-createtrigger.html>

²<https://www.postgresql.org/docs/15/sql-createfunction.html>

```
1  CREATE TABLE nation (  
2      nation_id int PRIMARY KEY,  
3      name varchar(64) NOT NULL,  
4      flag bytea NULL  
5  );  
6  
7  CREATE TABLE team (  
8      team_id int PRIMARY KEY,  
9      name varchar(64) NOT NULL,  
10     starts_for int NOT NULL,  
11     CONSTRAINT fk_team_nation FOREIGN KEY (starts_for)  
12         REFERENCES nation (nation_id)  
13 );  
14  
15 CREATE TABLE athlete (  
16     athlete_id int PRIMARY KEY,  
17     name varchar(64) NOT NULL,  
18     age int NOT NULL  
19 );  
20  
21 CREATE TABLE "event" (  
22     event_id int PRIMARY KEY,  
23     title varchar(64) NOT NULL,  
24     "date" date NOT NULL,  
25     is_team_event boolean NOT NULL  
26 );  
27  
28 CREATE TABLE athlete_in_team (  
29     athlete_id int NOT NULL,  
30     team_id int NOT NULL,  
31     CONSTRAINT fk_athlete_in_team_athlete  
32         FOREIGN KEY (athlete_id)  
33         REFERENCES athlete (athlete_id),  
34     CONSTRAINT fk_athlete_in_team_team  
35         FOREIGN KEY (team_id)  
36         REFERENCES team (team_id)  
37 );  
38  
39 CREATE TABLE athlete_participates_in_event (  
40     athlete_id int NOT NULL,  
41     event_id int NOT NULL,  
42     CONSTRAINT fk_athlete_participates_in_event_athlete  
43         FOREIGN KEY (athlete_id)  
44         REFERENCES athlete (athlete_id),  
45     CONSTRAINT fk_athlete_participates_in_event_event  
46         FOREIGN KEY (event_id)  
47         REFERENCES "event" (event_id)  
48 );
```

```

40     athlete_id int NOT NULL,
41     event_id int NOT NULL,
42     rank int NOT NULL,
43     CONSTRAINT fk_athlete_participates_in_event_athlete
44         FOREIGN KEY (athlete_id)
45         REFERENCES athlete (athlete_id),
46     CONSTRAINT fk_athlete_participates_in_event_event
47         FOREIGN KEY (event_id)
48         REFERENCES "event" (event_id)
49 );
50
51 CREATE TABLE team_participates_in_event (
52     team_id int NOT NULL,
53     event_id int NOT NULL,
54     rank int NOT NULL,
55     CONSTRAINT fk_team_participates_in_event_team
56         FOREIGN KEY (team_id)
57         REFERENCES team (team_id),
58     CONSTRAINT fk_athlete_participates_in_event_event
59         FOREIGN KEY (event_id)
60         REFERENCES "event" (event_id)
61 );


```

- b) 2 Punkte Legen Sie einen TRIGGER an der verhindert, dass für ein Team mehr als 20 Einträge in die von Ihnen in Aufgabe a) angelegte Mapping-Tabelle zwischen Athlet*innen und Teams eingefügt werden können.

Wenn die Bedingung verletzt wird, soll eine sinnvolle Fehlermeldung ausgegeben werden.

Abgabe



 3b.sql (Trigger Code)

Lösung



```

1  CREATE FUNCTION check_team_size()
2  RETURNS TRIGGER
3  AS $$
4      DECLARE num_athletes int;
5      BEGIN
6          -- Get the number of athletes for the given team.
7          SELECT INTO num_athletes COUNT(*)

```

```


8      FROM      athlete_in_team
9      WHERE      team_id = NEW.team_id;
10
11      -- Check condition.
12      IF num_athletes > 20 THEN
13          RAISE EXCEPTION 'Teams can not consist of more than 20 athletes!';
14      END IF;
15
16      -- Done, all good.
17      RETURN NEW;
18  END
19  $$ LANGUAGE plpgsql;
20
21
22
23  CREATE TRIGGER      trigger_check_team_size
24  AFTER INSERT ON      athlete_in_team
25  FOR EACH ROW
26  EXECUTE PROCEDURE check_team_size();

```

- c) 1 Punkt Testen Sie nun Ihren Trigger. Legen Sie mindestens ein Team und mindestens 21 Athlet*innen in Ihrer Datenbank an und versuchen Sie dann, alle 21 Athlet*innen dem selben Team zuzuweisen. Was passiert?

Abgabe



 3c.sql (insert-Statements)

Lösung



```

1  INSERT INTO nation (nation_id, name)
2  VALUES      (0, 'Austria');
3
4  INSERT INTO team (team_id, name, starts_for)
5  VALUES      (0, 'Test team', 0);
6
7  INSERT INTO athlete (athlete_id, name, age)
8  VALUES      (0, 'Test Athlete', 25),
9              (1, 'Test Athlete', 25),
10             (2, 'Test Athlete', 25),
11             (3, 'Test Athlete', 25),

```



```

12         (4, 'Test Athlete', 25),
13         (5, 'Test Athlete', 25),
14         (6, 'Test Athlete', 25),
15         (7, 'Test Athlete', 25),
16         (8, 'Test Athlete', 25),
17         (9, 'Test Athlete', 25),
18         (10, 'Test Athlete', 25),
19         (11, 'Test Athlete', 25),
20         (12, 'Test Athlete', 25),
21         (13, 'Test Athlete', 25),
22         (14, 'Test Athlete', 25),
23         (15, 'Test Athlete', 25),
24         (16, 'Test Athlete', 25),
25         (17, 'Test Athlete', 25),
26         (18, 'Test Athlete', 25),
27         (19, 'Test Athlete', 25),
28         (20, 'Test Athlete', 25);
29
30 INSERT INTO athlete_in_team (athlete_id, team_id)
31 VALUES
32     (0, 0),
33     (1, 0),
34     (2, 0),
35     (3, 0),
36     (4, 0),
37     (5, 0),
38     (6, 0),
39     (7, 0),
40     (8, 0),
41     (9, 0),
42     (10, 0),
43     (11, 0),
44     (12, 0),
45     (13, 0),
46     (14, 0),
47     (15, 0),
48     (16, 0),
49     (17, 0),
50     (18, 0),
51     (19, 0),
52     (20, 0);

```

Wichtig: Laden Sie bitte Ihre Lösung in OLAT hoch und geben Sie mittels der Ankreuzliste auch unbedingt an, welche Aufgaben Sie gelöst haben. Die Deadline dafür läuft am Vortag des Proseminars um 16:00 ab.