

29.11.2022

Übungsblatt 7 – Lösungsvorschlag

Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

a) Rekursion und Common Table Expressions (CTE)

a) Beantworten Sie folgende Fragen zu Common Table Expressions:

i. Was macht eine Common Table Expression?

Lösung



Eine CTE produziert ein temporäres Ergebnis, auf das man in der folgenden Query zugreifen kann. CTEs werden mit der **WITH** Klausel erstellt.

ii. Wofür werden Common Table Expressions eingesetzt?

Lösung



CTEs erlauben es, komplexe Queries zu vereinfachen und hierarchische Daten abzufragen.

iii. Wie hängen CTEs mit rekursiven Abfragen zusammen?

Lösung



Rekursive Abfragen sind CTEs, die auf sich selbst verweisen.

iv. Geben Sie zwei Anwendungsfälle für rekursive Abfragen an.

Lösung



Wegsuche, Vorfahrenermittlung

b) Öffnen Sie das im Rahmen einer DBIS-Bachelorarbeit entwickelte Rekursions-Tool unter der Adresse: <https://dbis-uibk.github.io/recursion/>.

Wählen Sie das Datenset *Flights Simple* aus. Nun können Sie anhand von zwei Examples die Ausführung der Abfrage schrittweise verfolgen. Versuchen Sie zu verstehen, wie die Rekursion funktioniert.

b) **Funktionale Abhängigkeiten:**

Gegeben sei Relation $R(A_1, \dots, A_n)$ und $\alpha, \beta \subset \{A_1, \dots, A_n\}$. 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 in der Tabelle gegebenen Daten 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 $A \rightarrow E$?

Lösung



Gilt.

b) Gilt $B \rightarrow D$?

Lösung



Gilt nicht, da für z unterschiedliche Werte vorhanden sind.

c) Gilt $G \rightarrow E$?

Lösung



Gilt.

d) Gilt $AB \rightarrow D$?

Lösung



Gilt (allein dadurch überprüfbar, dass A eindeutig ist).

e) Gilt $EG \rightarrow C$?

Lösung



Gilt nicht, da für ja/klein verschiedene Werte (S, L) vorhanden sind.

c) **Theorie:** Diskutieren Sie folgende Fragen zu Schlüsseln:

- Bei der Modellierung von Personen wird oft z.B. die Sozialversicherungsnummer als Schlüssel verwendet. 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).

- d) **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 (Rekursion)

[3 Punkte]

Gegeben sei eine Datenbank zum Verwalten eines sozialen Netzwerks mit folgenden Tabellen:

```
person(id, firstname, lastname, year_of_birth, country_id)
friendship(person1_id, person2_id, friends_since)
follow(person_id, followed_person_id, follows_since)
```

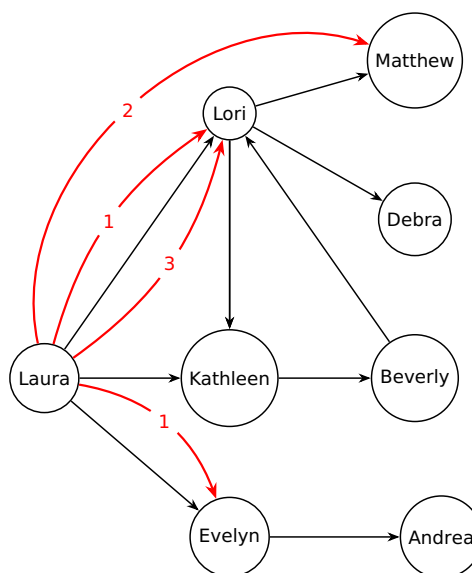


Abbildung 1: follow-Graph (Beispiel)

Ein Eintrag in *friendship* besagt, dass *person1_id* mit *person2_id* befreundet ist. Da Freundschaften bidirektional sind, ist immer nur ein Paar pro Freundschaft verzeichnet: (x, y) bedeutet, dass sowohl Person x mit Person y befreundet ist, als auch Person y mit Person x. Bitte beachten Sie, dass es keinen expliziten Eintrag (y,x) in der Tabelle gibt (obwohl dies semantisch korrekt wäre). Ein Eintrag in *follow* besagt, dass *person_id* (der Follower) der Person *followed_person_id* (der Followee) folgt. In diesem Fall ist es nicht zwingend notwendig, dass auch die Gegenrichtung gilt. D.h. wenn dies der Fall wäre, dann gäbe es dafür auch einen eigenen Eintrag.

Friendship- und follow-Beziehungen kann man übersichtlich in einem Graph darstellen. Der Graph in Abbildung 1 zeigt ein Beispiel der follow-Beziehungen. In diesem Graphen können Sie zum Beispiel sehen, dass Laura drei Personen direkt folgt (und damit ihr *Followee* ist): Lori, Kathleen und Evelyn (die sogenannten *Followees*). Indirekt (also über mehrere Zwischenschritte) folgt Laura aber

auch z.B. Matthew (über Lori). Die Erreichbarkeit bzw. Nähe von anderen Knoten (in diesem Fall Personen) werden meist mit der Anzahl der sogenannten Hops (Zwischenschritte) im Graph gemessen. Wie im Graphen für einige Beispiele in rot gekennzeichnet, folgt Laura Evelyn über einen Hop (also direkt) und Matthew über zwei Hops. Laura folgt Lori einerseits direkt, aber auch über drei Hops (Laura->Kathleen->Beverly->Lori).

Erstellen Sie eine neue Datenbank und laden Sie den Inhalt der Datei `fakebook-backup.sqlOLAT` in diese. Geben Sie für die Aufgaben jeweils die SQL Datei und die zugehörigen Resultate ab.

- a) 1 Punkt Geben Sie die Followees der Person Victor (Vorname) Hall (Nachname) aus, denen diese maximal über 3 Hops folgt. Berechnen Sie zudem den Follow-Pfad (string), aus dem hervorgeht, wie die Beziehung zustande kommt. Der Follow-Pfad soll folgendem Format entsprechen: ID->ID->...->ID (fügen Sie keine Leerzeichen ein!).

Das Ergebnis mit folgenden Spalten soll aufsteigend nach steps ausgegeben werden:

- person_id (Victor Halls ID)
- following (ID des Followees)
- path (der Pfad als string)
- steps (die Anzahl der Hops als integer)

Das Ergebnis sollte wie folgt aussehen (Beispiel):

person_id	following	path	steps
33	45	33->45	1
...
33	313	33->215->313	2
...
33	682	33->422->716->682	3

Tabelle 1: Ausgabe zur Aufgabe 1a.

Hinweis



Sie können zum Verketteten von Zeichenketten die Funktion CONCAT ^a verwenden. Bereiten Sie sich mithilfe einer Common Table Expression (CTE) die Ausgangsrelation so vor, dass sie für eine anschließende Rekursion geeignet ist. Beachten Sie, dass das Schlüsselwort RECURSIVE am Anfang stehen muss, ohne RECURSIVE darf eine CTE nur auf etwas verweisen, das lexikalisch vorher definiert wurde.

^a<https://www.postgresql.org/docs/13/functions-string.html>

Abgabe



1a.sql

1a_result.txt

Lösung



Query

```
1  WITH RECURSIVE wh_follow_path (person_id, following, path, steps)
2  AS
3  (
4      SELECT      f.person_id,
5                  f.followed_person_id,
6                  CONCAT(f.person_id, '->', f.followed_person_id),
7                  1
8      FROM        follow AS f
9      INNER JOIN  person AS p ON f.person_id = p.id
10     WHERE       p.firstname = 'Victor'
11     AND         p.lastname = 'Hall'
12     UNION ALL
13     SELECT      t.person_id,
14                 follow.followed_person_id,
15                 CONCAT(path, '->', follow.followed_person_id),
16                 steps+1
17     FROM        wh_follow_path AS t
18     INNER JOIN  follow ON t.following = follow.person_id
19     WHERE       t.steps < 3
20 )
21 SELECT *
22 FROM          wh_follow_path
23 ORDER BY      steps ASC
```

Result

id	following	path	steps
57	15	57->15	1
57	33	57->33	1
57	786	57->786	1
57	81	57->15->81	2
57	328	57->15->328	2
57	641	57->15->641	2
57	157	57->33->157	2
...			
57	952	57->786->880->952	3

(94 rows)

b) 1 Punkt Berechnen Sie für alle 1990 geborenen Personen, wieviele **unique** Follower sie je Zwischenschritt im Zeitraum von 01.06.2015 bis 01.06.2016 gewonnen haben. Dabei gilt:

- Indirekte Follower über Zwischenschritte sollen nur gezählt werden, wenn sie den entsprechenden Personen in diesem Zeitraum gefolgt sind.

- Betrachten Sie maximal 5 Zwischenschritte und geben Sie für jede Person und jeden Zwischenschritt die Anzahl an Followern aus.
- Direkte Follower zählen nicht als Zwischenschritt, somit beginnt steps bei 0, wie in Tabelle 2 zu sehen ist.

Geben Sie am Ende folgendes aus:

- root_person_id (ID des Followees)
- firstname (Vorname des Followees)
- lastname (Nachname des Followees)
- steps (Zwischenschritte)
- follower_count (Followeranzahl)

Sortieren Sie die Ausgabe nach root_person_id, und anschließend nach steps.

Das Ergebnis sollte wie folgt aussehen (Beispiel):

root_person_id	firstname	lastname	steps	follower_count
11	Sharon	Moore	0	20
11	Sharon	Moore	1	31
11	Sharon	Moore	2	83
11	Sharon	Moore	3	172
11	Sharon	Moore	4	635
11	Sharon	Moore	5	1422
65	Mike	Patton	0	13
...

Tabelle 2: Ausgabe zur Aufgabe 1b.

Hinweis



Beachten Sie, dass in der follow-Relation die Follower auf der **linken** Seite stehen.

Abgabe



1b.sql

1b_result.txt

Lösung



Query

```

1  WITH RECURSIVE RecFollower(root_person_id, follower_id, steps)
2  AS
3  (
4      SELECT      followed_person_id, person_id, 0
5      FROM        follow f
6      INNER JOIN  person p ON (f.followed_person_id = p.id)
7      WHERE       p.year_of_birth = 1990

```

```

8      AND      f.follows_since BETWEEN '2015-06-01' AND '2016-06-01'
9
10     UNION ALL
11
12     SELECT    r.root_person_id,
13               f.person_id,
14               r.steps + 1
15     FROM      RecFollower r
16     INNER JOIN follow f ON (r.follower_id = f.followed_person_id)
17     WHERE     f.follows_since BETWEEN '2015-06-01' AND '2016-06-01'
18     AND      r.steps < 5
19 )
20 SELECT    r.root_person_id,
21           p.firstname,
22           p.lastname,
23           r.steps,
24           COUNT(DISTINCT r.follower_id) AS follower_count
25 FROM      RecFollower r
26 INNER JOIN person p ON (r.root_person_id = p.id)
27 GROUP BY  r.root_person_id, p.firstname, p.lastname, r.steps
28 ORDER BY  r.root_person_id, steps

```

Result

root_person_id	firstname	lastname	steps	follower_count
2	Carol	Garcia	0	6
2	Carol	Garcia	1	17
2	Carol	Garcia	2	54
2	Carol	Garcia	3	148
2	Carol	Garcia	4	353
2	Carol	Garcia	5	642
...				
995	Jose	Mitchell	0	2
995	Jose	Mitchell	1	9
995	Jose	Mitchell	2	20
995	Jose	Mitchell	3	51
995	Jose	Mitchell	4	124
995	Jose	Mitchell	5	324

(198 rows)

- c) 1 Punkt Ändern bzw. erweitern Sie die vorige Abfrage aus 1)b) dahingehend, dass nur die Ergebnisse von Personen angezeigt werden, die mit dem 5. Schritt mehr als 550 Follower haben. Die Ausgabe soll wieder wie in Tabelle 2 aussehen, nur diesmal für jene Personen, bei denen die Bedingung auch zutrifft.

Hinweis



Es sollen von diesen Personen weiterhin alle Zwischenschritte ausgegeben werden, d.h. ein einfaches **HAVING** wird hier nicht funktionieren (überlegen Sie sich noch einmal warum, falls dies nicht ganz klar ist). Erweitern Sie deshalb die über die Rekursion bereits bestehende CTE um das Ergebnis der letzten Aufgabe und arbeiten Sie anschließend mit diesem weiter.

Abgabe



1c.sql

1c_result.txt

Lösung



Query

```
1  WITH RECURSIVE RecFollower(root_person_id, person_id, follower_id, steps)
2  AS
3  (
4      SELECT      followed_person_id, followed_person_id, person_id, 0
5      FROM        follow f
6      INNER JOIN  person p ON (f.followed_person_id = p.id)
7      WHERE       p.year_of_birth = 1990
8      AND         f.follows_since BETWEEN '2015-06-01' AND '2016-06-01'
9
10     UNION ALL
11
12     SELECT      r.root_person_id,
13                f.followed_person_id,
14                f.person_id,
15                r.steps + 1
16     FROM        RecFollower r
17     INNER JOIN  follow f ON (r.follower_id = f.followed_person_id)
18     WHERE       f.follows_since BETWEEN '2015-06-01' AND '2016-06-01'
19     AND         r.steps < 5
20 ),
21 RecFollowerResult AS
22 (
23     SELECT      r.root_person_id, p.firstname, p.lastname, r.steps,
24     COUNT      (DISTINCT r.follower_id) AS follower_count
25     FROM        RecFollower r
26     JOIN        person p ON (r.root_person_id = p.id)
27     GROUP BY    r.root_person_id, p.firstname, p.lastname, r.steps
```

```

28     ORDER BY  r.root_person_id, steps
29 )
30
31 SELECT *
32 FROM   RecFollowerResult
33 WHERE  root_person_id IN
34 (
35     SELECT root_person_id
36     FROM   RecFollowerResult
37     WHERE  steps = 5 AND follower_count > 550
38 )

```

Result

root_person_id	firstname	lastname	steps	follower_count
2	Carol	Garcia	0	6
2	Carol	Garcia	1	17
2	Carol	Garcia	2	52
2	Carol	Garcia	3	148
2	Carol	Garcia	4	353
2	Carol	Garcia	5	642
...				
815	Deborah	Brown	0	10
815	Deborah	Brown	1	37
815	Deborah	Brown	2	94
815	Deborah	Brown	3	240
815	Deborah	Brown	4	496
815	Deborah	Brown	5	762

(48 rows)

Ein einfaches `HAVING steps=5 AND follower_count > 550` funktioniert deshalb nicht, weil dann ausschließlich diese Zeilen angezeigt werden würden. Da aber alle Zwischenschritte angezeigt werden sollen, wird das Ergebnis als weitere CTE definiert, worauf am Ende einfach selektiert werden kann.

Aufgabe 2 (Normalformenbestimmung)

[3 Punkte]

Gegeben seien die abstrakten Relationenschemata \mathcal{R} über die Attribute 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

2a.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

2b.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

2c.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

2d.txt

Aufgabe 3 (Algorithmen)**[4 Punkte]**

- a) 2 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. Geben Sie im Abgabefile die durchgeführten Schritte nachvollziehbar an.

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



3a.pdf

- b) 2 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. Geben Sie im Abgabefile die durchgeführten Schritte nachvollziehbar an.

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



3b.pdf

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 23:59 (Mitternacht) ab.