

29.11.2022

Übungsblatt 7

Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

a) Rekursion und Common Table Expressions (CTE)

a) Beantworten Sie folgende Fragen zu Common Table Expressions:

- Was macht eine Common Table Expression?
- Wofür werden Common Table Expressions eingesetzt?
- Wie hängen CTEs mit rekursiven Abfragen zusammen?
- Geben Sie zwei Anwendungsfälle für rekursive Abfragen an.

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

- Gilt $A \rightarrow E$?
- Gilt $B \rightarrow D$?
- Gilt $G \rightarrow E$?
- Gilt $AB \rightarrow D$?
- Gilt $EG \rightarrow C$?

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.
- Wie unterscheiden sich funktionale Abhängigkeit und voll funktionale Abhängigkeit?
- Warum erfüllt eine Relation in 1NF, die nur einelementige Schlüsselkandidaten besitzt, automatisch die 2NF?

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

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)
```

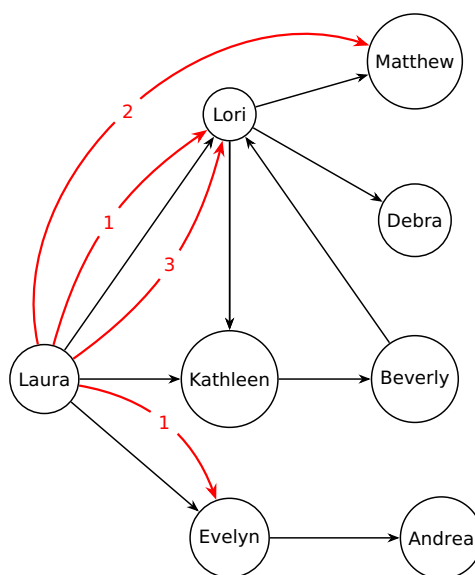


Abbildung 1: follow-Graph (Beispiel)

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-backupOLAT.sql` 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

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):

<code>root_person_id</code>	<code>firstname</code>	<code>lastname</code>	<code>steps</code>	<code>follower_count</code>
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

- 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

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 $FA = \{ABD \rightarrow C, A \rightarrow E, E \rightarrow A\}$

Abgabe

2a.txt

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

Abgabe

2b.txt

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


Abgabe

2c.txt

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

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.

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.

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.