Freie Universität Berlin

Prof. Dr. Agn‘es Voisard, Nicolas Lehmann

Datenbanksysteme, SoSe 2017

Übungsblatt 6

Tutor: Nicolas Lehmann
Tutorium 10

Boyan Hristov, Julian Habib

15. Juni 2017

Link zum Git Repository: https://github.com/BoyanH/Freie-Universitaet-Berlin/tree/master/Datenbanksysteme/Solutions/homework6

1. Aufgabe

a) Alle Attributen in R_1 haben atomäre Domäne, sind also keine Relationen. Z.B. alle Einträge haben für den Attribut A die Werte a_1, a_2, a_3 und keine Werte wie z.B. (a_1, a_2).

b)

$$FD(R_1) = \{$$
$$A \rightarrow B$$
$$A \rightarrow C$$
$$D \rightarrow C$$
$$BD \rightarrow A$$
$$\}$$

$\Rightarrow$ AD und BD sind Superschlüssel und auch beide Kandidatschlüsseln.
Primäre Attributen sind A, B und D.

Da C kein Primärattribut ist, von A abhängig ist und da A eine Untermenge eines Schlüssels ist (AD), ist R_1 nicht in 2NF.

c)

$$FD(R_2) = \{$$
$$E \rightarrow F$$
$$E \rightarrow G$$
$$FG \rightarrow E$$
$$\}$$

$\Rightarrow$ E und FG sind Superschlüssel, Primärattributen sind E, F und G.

In den ersten zwei funktionalen Abhängigkeiten ist die linke Seite ein Superschlüssel, in der 3. Abhängigkeit ist die rechte Seite ein Primärattribut. $\Rightarrow R_2$ ist in 3NF.

d)

$$FD^+(R_2) = \{$$
$$E \to F$$
$$E \to G$$
$$E \to FG$$
$$FG \to E$$
$$\}$$

Da wir schon aus c) kennen, dass E und FG Superschlüssel sind und da diese alle mögliche linke Seiten von einer funktionalen Abhängigkeit sind, ist R_2 in Boyce-Codd Normalform (BCNF).

2. Aufgabe

a)

$$FD(R_3) = \{$$
$$H \to JK$$
$$I \to HJ$$
$$K \to L$$
$$\}$$

$$I \to H \wedge H \to JK \Rightarrow I \to HJK$$
$\Rightarrow$I $\to$ HJ kann durch I $\to$ H erstetzt werden

Minimal $FD(R_3) = \{$
$$H \to JK$$
$$I \to H$$
$$K \to L$$
$$\}$$

Gute Zerlegung: $R_{31}(H, J, K)$ und $R_{32}(H, I, K, L)$

b)

$R_{31} \cap R_{32} \equiv HK$
$R_{31} - R_{32} \equiv J$ $\qquad$ $R_{32} - R_{31} \equiv IL$

$R_{31} \cap R_{32} \to HJK$ (wegen H $\to$ JK) $\to J \to R_{31} - R_{32}$
$\Rightarrow$ Unsere Zerlegung ist verlustlos

c)

$FD(R_{31}) = \{H \to JK\}$
$FD(R_{32}) = \{H \to K, I \to H, K \to L\}$
$FD(R_{31}) \cup FD(R_{32}) \equiv \{H \to JK, I \to H, K \to L\} \equiv FD(R_3)$
$\Rightarrow$ Unsere Zerlegung ist abhängigkeitserhaltend

3. Aufgabe

a) Aus jedem Item kann man durch eine Hash-Funktion ein Hash erzeugen durch mathematische Umformungen aus allen Suchschlüssel. Mehrere Items werden in dem selben Bucket gespeichert (nacheinandere folgende Blöcke im Speicher) wenn die Hashfunktion das gleiche Hash aus ihren Suchschlüssel erzeugt. Danach, beim Suchen von Lösch / Einfüge / Lese Position müssen alle Einträge in dem entsprechenden Bucket nach einander geprüft werden. Deswegen erzeugen gute Hashfunktione randomisierte Hashes, damit es ungefähr genau so viele Items pro Bucket gibt.

b) In einem Dense-Index gibt es Indizes für alle Suchschlüsselwerte. Z.b. wenn das Attribut A ein Primärindex ist, gibt es für alle mögliche Werte von A ein Index, die Einträge mit dem gleichen Wert von A stehen nach einander im Speicher.

   Der Sparse-Index hat nur für gewählte Werte von den Suchschlüssel Indizes. Dabei müssen aber immer die Einträge im Speicher nach dem Primärindex sortiert werden. Damit sucht man nach dem Index des alphabetisch größten kleineren Wert und muss dann alle weitere Einträge durchsuchen, um den Item zu finden. Dabei haben wir viel speicher für den Index gespart, die Suchzeiten haben sich aber vergrößert. Ein gutes Balance zwischen den beiden ist zu finden.