

Fragestunde EDB

Tobias Bittner

2025-02-05

Übersicht I

- 1 Anwartschaftssperren
- 2 Normalisierung
- 3 Constraints

Übersicht I

- 1 Anwartschaftssperren
- 2 Normalisierung
- 3 Constraints

Probleme mit Sperren



Sperrgranulat Tupel

- Nicht immer effizient
 - Aufwändig bei Transaktionen, die viele (alle) Tupel einer Relation benötigen
 - Große Sperrtabellen, hohe Verwaltungskosten
- Nicht ausreichend, um alle Fehlerklassen auszuschließen

Phantom-Problem

- Sperren nur auf *existierende* Tupel
- Nicht existierende Tupel können jederzeit eingefügt werden – sind dann aber nicht gesperrt (= Phantome).

Lesetransaktion (Gehaltssumme bestimmen)

```
SELECT SUM(Gehalt)
INTO :summe1
FROM Pers
WHERE ANr = 17;

...

SELECT SUM(Gehalt)
INTO :summe2
FROM Pers
WHERE ANr = 17;

if summe1 ≠ summe2
then <Fehler>;
```

Änderungstransaktion (Einfügen eines neuen Angestellten)

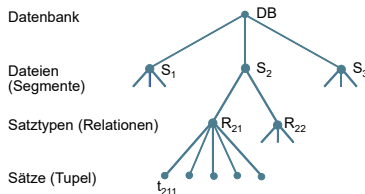
```
INSERT INTO Pers
(PNr, ANr, Gehalt)
VALUES (4567, 17, 55.000);
COMMIT;
```

Zeit



Hierarchische Schachtelung der Datenobjekte

- erlaubt Flexibilität bei der Wahl der Sperrgranulate
- Synchronisation langer TAen auf Relationenebene
- Synchronisation kurzer TAen auf Tupelebene



Anwartschaftssperren



Nachteil von reinen S- und X-Sperren

- Alle Nachfolgeknoten implizit mitgesperrt
- Alle Vorgängerknoten ebenfalls sperren, um Unverträglichkeiten zu erkennen
 - X-Sperre auf DB: erzwingt Einbenutzerbetrieb
 - S-Sperre auf DB: nur Lese-Transaktionen können parallel laufen

Verwendung von **Anwartschaftssperren** (intention locks)

- I-Sperre oder Sperranzeige
 - **IS-Sperre** (intention share), falls auf untergeordnete Objekte nur lesend zugegriffen wird
 - **IX-Sperre** (intention exclusive), falls auf untergeordnete Objekte auch schreibend zugegriffen wird
- Ersetzt die Sperren für Datenobjekte auf den höheren Hierarchie-Ebenen
- Nutzung einer Untermenge wird angezeigt, aber in der Untermenge werden noch explizit Sperren gesetzt

Verwendung von Anwartschaftssperren



Top Down beim Erwerb von Sperren

- Bevor ein Knoten mit S oder IS gesperrt werden darf, müssen alle Vorgänger in der Hierarchie im IX- oder im IS-Modus gesperrt worden sein.
- Bevor ein Knoten mit X oder IX gesperrt werden darf, müssen alle Vorgänger in der Hierarchie im IX-Modus gesperrt worden sein.

Bottom Up bei der Freigabe von Sperren

- Freigabe der Sperren von unten nach oben
- Bei keinem Knoten darf die Sperre aufgehoben werden, wenn die betreffende Transaktion noch Nachfolger dieses Knotens gesperrt hat.

Beispiel

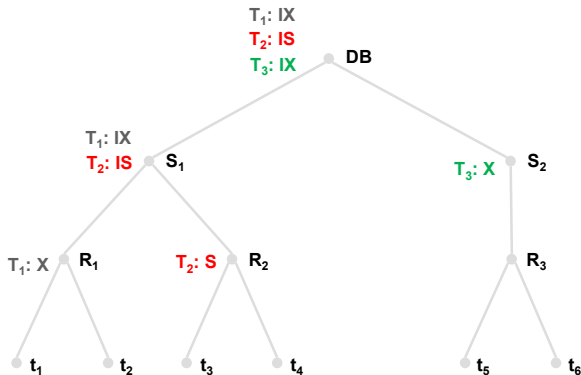
- T_1 : benötigt exklusive Sperre von R_1 in S_1
- T_2 : benötigt Lesesperre von R_2 in S_1
- T_3 : benötigt exklusive Sperre von Segment S_2

Anwartschaftssperren - Wiederholung

Beispiel zu Anwartschaftssperren



FAU



Ein Datenbanksystem bietet folgende Sperrgranularitäten an:
Tupel - Relation - Datenbank

Entscheiden Sie für jede Aussage, ob diese richtig oder falsch ist:

- 1 Um ein Tupel zu ändern (**schreiben**), benötigt man nur eine X-Sperre auf dem Tupel und eine IX-Sperre auf der Relation.
- 2 Hält eine Transaktion eine IX-Sperre auf der Datenbank, so kann sie ohne zusätzliche Sperren eine X-Sperre auf einem Tupel anfordern.
- 3 Solange die Transaktion T1 Leserechte auf einem Tupel hat, kann keine andere Transaktion eine IX-Sperre auf der Datenbank erhalten.

Aktuell hält die Transaktion T1 eine X-Sperre auf der Datenbank, sonst sind keine weitere Sperren vorhanden. Geben sie alle Sperren an, welche eine Transaktion T2 erhalten kann:

- IS
- SIX
- IX
- X
- S
- Es kann keine gewährt werden

Geben Sie alle Sperren an, welche einer Transaktion T1 gewährt werden können, unter der Voraussetzung, dass die einzige aktive Sperre auf der Datenbank eine S-Sperre von Transaktion T2 ist.

- IS
- IX
- X
- SIX
- S
- Es sind keine weiteren Sperren möglich

Übersicht I

- 1 Anwartschaftssperren
- 2 Normalisierung
- 3 Constraints

Funktionale Abhängigkeiten



Definition:

- Sei R eine Relation: $R(A_1, A_2, \dots, A_n)$
- $X \subset \{A_1, A_2, \dots, A_n\}, \quad Y \subset \{A_1, A_2, \dots, A_n\}$,
- **Y ist funktional abhängig von X** (notiert als: **$X \rightarrow Y$**),
wenn es keine zwei Tupel geben darf,
in denen für *gleiche* X -Werte *verschiedene* Y -Werte auftreten.

Beispiele:

- Funktion im mathematischen Sinne: $Y = X^2$
 - d.h. $X \rightarrow Y$ (auch: "X bestimmt Y")
- Matrikelnummer $\rightarrow$ Name
- Rechnungsnummer $\rightarrow$ Betrag

Funktionale Abhängigkeiten (3)



Regeln für Funktionale Abhängigkeiten:

Armstrong-Axiome

Reflexivität: Aus $\beta \subseteq \alpha$ folgt: $\alpha \rightarrow \beta$

Verstärkung: Aus $\alpha \rightarrow \beta$ folgt: $\alpha\gamma \rightarrow \beta\gamma$ für $\gamma \subseteq U$

Transitivität: Aus $\alpha \rightarrow \beta$
und $\beta \rightarrow \gamma$ folgt: $\alpha \rightarrow \gamma$

Funktionale Abhängigkeiten (4)



Weitere Regeln für Funktionale Abhängigkeiten:
durch **Herleitung aus den Armstrong-Axiomen**

- Vereinigung: Aus $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$ folgt: $\alpha \rightarrow \beta\gamma$
- Dekomposition: Aus $\alpha \rightarrow \beta\gamma$ folgt: $\alpha \rightarrow \gamma$ und $\alpha \rightarrow \beta$
- Pseudotransitivität: Aus $\alpha \rightarrow \beta$ und $\gamma\beta \rightarrow \delta$ folgt: $\alpha\gamma \rightarrow \delta$

Funktionale Abhängigkeiten (5)



"Volle Funktionale Abhängigkeit"

- Sei $X \rightarrow Y$
- **Y ist voll funktional abhängig von X**,
wenn es **keine echte Teilmenge** Z von X gibt,
für die gilt $Z \rightarrow Y$.
- Alle Attribute in X werden gebraucht, um Y zu bestimmen.
- Beispiele:
 - Matrikelnummer, Fach $\rightarrow$ Note
 - Matrikelnummer $\rightarrow$ Name

Begriffsdefinitionen



- **Superschlüssel:**
 - Attribut oder Attributkombination, von der alle Attribute einer Relation funktional abhängen
- **Schlüsselkandidat:**
 - Minimaler Superschlüssel
 - Keine Teilmenge des Superschlüssels ist ebenfalls Superschlüssel.
- **Schlüsselattribut:**
 - Attribut, das Teil eines Schlüsselkandidaten ist
 - Nicht-Schlüsselattribut:
 - Attribut, das an keinem Schlüsselkandidaten beteiligt ist

Erste Normalform (1NF)



MERKE !

Definition:

Eine Relation ist in erster Normalform (1NF), wenn sie **nur atomare Attributwerte besitzt.**

- Das ist nach unserer Definition einer Relation bereits der Fall.
- Allgemein gilt für eine Relation (Tabelle) in 1NF:
 - Jeder Attributwert ist atomar.
 - Es gibt keine zwei gleichen Tupel (Zeilen).
 - Die Reihenfolge der Tupel ist unerheblich.
 - Jeder Attributwert (in einer Spalte) gehört zum Wertebereich des Attributs.
 - Jedes Attribut (jede Spalte) hat einen eindeutigen Namen.
 - Die Reihenfolge der Attribute ist unerheblich.

Erste Normalform (2NF)

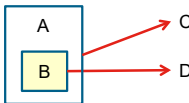


MERKE !

Definition:

Eine Relation ist in zweiter Normalform (2NF), wenn sie in erster Normalform ist und alle Nicht-Schlüsselattribute voll funktional von jedem Schlüsselkandidaten abhängen.

- Schlüsselattribut (siehe oben):
 - Attribut, das Teil eines Schlüsselkandidaten ist
- Ziel der zweiten Normalform:
 - Eliminierung "partieller Abhängigkeiten" der Form:



Dritte Normalform (3NF)



MERKE !

Definition:

Eine Relation ist in dritter Normalform (3NF),
wenn **kein Nicht-Schlüsselattribut transitiv abhängig von einem Schlüsselkandidaten ist.**

Transitivität: Aus $\alpha \rightarrow \beta$
und $\beta \rightarrow \gamma$ folgt: $\alpha \rightarrow \gamma$

– Transitive Abhängigkeit:

C ist **transitiv** abhängig von **A**, wenn es ein **B** gibt, so dass gilt:



Achtung:
B ist hier kein
Superschlüssel !

Boyce-Codd Normalform (BCNF)



FAU

MERKE !

Definition:

Eine Relation ist in Boyce-Codd-Normalform (BCNF), wenn jede Determinante einer FA ein Superschlüssel ist.

– Impliziert 3NF

Algorithmus:

- Liste alle Funktionalen Abhängigkeiten auf.
- Erzeuge für jede FA, deren Determinante kein Superschlüssel ist, eine neue Relation.

Anmerkungen:

- Die BCNF ist eine Verschärfung der 3NF
→ Eine Tabelle R, die in BCNF ist, ist immer auch in 3NF
- Eine Tabelle R, die in 3NF ist, ist immer auch in BCNF wenn R keine überlappenden Schlüsselkandidaten hat

Vierte Normalform (4NF)



■ Mehrwertige Abhängigkeit (Multi-valued Dependency, MVD)

- Sei R eine Relation, und A und B seien jeweils Attributmengen aus R .
 C sei $R - AB$ (alle Attribute aus R außer die in A und B).
 B ist mehrwertig abhängig von A ,
wenn durch den Wert von A eine Menge von Werten in B bestimmt wird
und diese Werte unabhängig sind von den in C vorkommenden Attributwerten. D.h.
die Werte, die in B vorkommen, sind nur von A abhängig und werden nicht
fallabhängig durch C weiter eingeschränkt.
- Eine Mehrwertige Abhängigkeit $A \twoheadrightarrow B$ in R ist **trivial**,
wenn B in A enthalten ist oder
außer A und B keine weiteren Attribute in R enthalten sind. ($C = \{\}$)



Beispiele: $(\text{Matrikelnummer}, \text{Fach}) \twoheadrightarrow \text{Fach}$ ist trivial
In der Relation Fächer(Matrikelnummer, Fach) gilt:
 $\text{Matrikelnummer} \twoheadrightarrow \text{Fach}$ ist trivial

Vierte Normalform (4NF) (2)



MERKE !

- Definition:

Eine Relation R ist in vierter Normalform (4NF), wenn
für jede nicht-triviale Mehrwertige Abhängigkeit
 $X \twoheadrightarrow A$ in R gilt: X ist Superschlüssel von R.

- Superschlüssel (siehe oben):
 - Jede Obermenge eines Schlüsselkandidaten ist ein Superschlüssel.

Vierte Normalform (4NF) – Beispiel



Interessen

Student	Fächer	Hobbies
Hannes	Informatik	Sport
Hannes	Informatik	Klavier
Josef	Botanik	Basteln
Josef	Nautik	Basteln
Josef	Musik	Basteln

Nicht in 4NF

Mehrwertige Abhängigkeiten:

Student $\longrightarrow\longrightarrow$ Fächer

Student $\longrightarrow\longrightarrow$ Hobbies

Vierte Normalform (4NF) – Beispiel (2)



Fächer

Student	Fächer
Hannes	Informatik
Josef	Botanik
Josef	Nautik
Josef	Musik

Hobbies

Student	Hobbies
Hannes	Sport
Hannes	Klavier
Josef	Basteln

4NF

Gegeben ist die Relation $R(A, B, C, D)$. Alle Attributwerte sind atomar.

- Es gelten folgende funktionale Abhängigkeiten: $D \rightarrow B$
Welche ist die **höchste** Normalform, die R erfüllt?
NFNF, 1NF, 2NF, 3NF, BCNF
- Es gelten folgende funktionale Abhängigkeiten: $A \rightarrow BCD, BC \rightarrow D$
Welche ist die **höchste** Normalform, die R erfüllt?
NFNF, 1NF, 2NF, 3NF, BCNF

Welche Aussagen zur Normalisierung sind richtig?

- Eine Tabelle ist in 3NF, wenn sie in 1NF ist und alle Determinanten funktionaler Abhängigkeiten Superschlüssel sind.
- Jede Tabelle kann verlustfrei und abhängigkeiterhaltend in BCNF normalisiert werden.
- Eine Tabelle in 1NF mit nur 2 Attributen ist immer in 4NF.
- Ein Superschlüssel ist immer eine Teilmenge eines Schlüsselkandidaten.

Übersicht I

- 1 Anwartschaftssperren
- 2 Normalisierung
- 3 Constraints**

Integritätsbedingungen (Constraints)



```
CREATE TABLE Angestellte
(
    PersNr          INT,
    Vorname         VARCHAR(20),
    Nachname        VARCHAR(30),
    AbtNr           INT NOT NULL DEFAULT 1,
    ...
    VorgesNr        INT,
    CONSTRAINT MitarbeiterPK
        PRIMARY KEY (PersNr),
    CONSTRAINT VorgesetzterFK
        FOREIGN KEY (VorgesNr) REFERENCES Angestellte(PersNr)
            ON DELETE SET NULL
            ON UPDATE CASCADE,
    CONSTRAINT AbteilungFK
        FOREIGN KEY (AbtNr) REFERENCES Abteilungen(AbtNr)
            ON DELETE SET DEFAULT
            ON UPDATE CASCADE,
    ... );
```

Check Constraints



Eingrenzung von Wertebereichen

```
CREATE TABLE Angestellte
(
  PersNr      INT,
  Nachname    VARCHAR(30),
  AbtNr       INT NOT NULL DEFAULT 1
              CHECK (AbtNr > 0 AND AbtNr < 100),
  ...

```

oder:

```
CREATE DOMAIN AbtNr_D AS INTEGER
CHECK (AbtNr_D > 0 AND AbtNr_D < 100);
```

```
CREATE TABLE Angestellte
(
  PersNr      INT,
  Nachname    VARCHAR(30),
  AbtNr       AbtNr_D NOT NULL DEFAULT 1,
  ... );
```

funktioniert in Oracle nicht

Nennen Sie eine Möglichkeit, um falsche Dateneingaben des Anwenders mithilfe des DBMS, durch eine geeignete Schemadefinition, zu verhindern.