Vorlesung aus dem WS21/22

Datenbanksysteme Übungsblatt

Prof. Dr. C. B.

geTEXt von Ningh

Contents

1 Übungsblatt 1

2 Übungsblatt 2

Thema: **Schlüssel, SQL-DDL**

2.1 Aufgabe 2-1

2.1.1 a)

Einfügen von Duplikaten wird verweigert.

2.1.2 b)

Ermğlicht effizientere Übungsprüfung von Schüssel/Fremdschüsselbeziehung.

2.1.3 c)

Zu jedem Attributwert in abhängiger Relation muss auch entsprechenden Schlüsselwert in referenzierter Relation existieren. (No dangling references)

2.1.4 d)

- Nein, Referenzielle Integrität verletzt.
- Nein, Schlüsseleigenschaft verletzt.
- Gut.

2.2 Aufgabe 2-2

```
a)
create table L
(
    lnr integer primary key,
    lname varchar(100) not null,
    sitz varchar(100)
);

create table T
(
    tnr integer primary key,
    tname varchar(100) not null,
    farbe varchar(100),
    gewicht float,
    preis decimal(8,4) check (preis > 0)
);

create table P
(
    pnr integer primary key,
```

```
    pname varchar(100) not null,
    ort varchar(100)
);


create table LTP
(
    lnr integer,
    tnr integer,
    pnr integer,
    menge integer check (menge > 0),
    primary key (lnr, tnr, pnr),
    foreign key (lnr) references L(lnr),
    foreign key (tnr) references T(tnr),
    foreign key (pnr) references P(pnr)
)
```

b)

```
alter table P add
(
    status integer default 5
)
```

c)

```
alter table T modify gewicht float check(gewicht >0)
```

d)

```
alter table P drop ort
```

e)

```
drop table LTP;
drop table L;
drop table T;
drop table P
```

3 Übungsblatt 3

3.1 Aufgabe 3-1

[Natural Join]

$$R \bowtie S \overset{\text{def}}{=} \left\{ rs_{[C_1,\ldots,C_o]} \mid r \in R \wedge s \in S \wedge r_{[B_1,\ldots,B_m]} = s_{[B_1,\ldots,B_m]} \right\}$$

. Da $r_{[B_1,\ldots,B_m]} = s_{[B_1,\ldots,B_m]} \Leftrightarrow r = s$:

$$R \bowtie S = \{r \mid r \in R \wedge s \in S \wedge r = s\}$$

Und r = s:

$$R \bowtie S = \{r \mid r \in R \wedge r \in S\} = R \cap S$$

3.2 Aufgabe 3-2

[Ableitung des Quotient-Operators]

$$R \div S \overset{!}{=} \Pi_{R-S}(R) - \Pi_{R-S}\left((\Pi_{R-S}(R) \times S) - R\right)$$

Beweis: 1) $R \div S \subseteq \Pi_{R-S}(R)$: trivial 2) z.z.: $\forall t \in \Pi_{R-S}(R)$ gilt: $t \in \Pi_{R-S}\left((\Pi_{R-S}(R) \times S) - R\right) \Leftrightarrow t \notin R \div S$

Sei $t = (t_1, \ldots, t_i) \in \Pi_{R-S}(R),$ dann gilt:

$$\begin{aligned}
& t \in \Pi_{R-S}\left((\Pi_{R-S}(R) \times S) - R\right) \\
\Leftrightarrow \quad & \exists x = (x_{i+1}, \ldots, x_n) \in S : tx = (t_1, \ldots, t_i, x_{i+1}, \ldots, x_n) \in (\Pi_{R-S}(R) \times S) - R \\
\Leftrightarrow \quad & \exists x \in S : tx \in \Pi_{R-S}(R) \times S \wedge tx \notin R \\
\Leftrightarrow \quad & \exists x \in S : t \in \Pi_{R-S}(R) \wedge tx \notin R \\
\Leftrightarrow \quad & \neg\forall x \in S : t \notin \Pi_{R-S}(R) \vee tx \in R \\
\Leftrightarrow \quad & t \notin \Pi_{R-S}(R) \vee \neg\forall x \in S : tx \in R \\
\overset{\text{Def.}}{\Leftrightarrow} \quad & t \notin R \div S
\end{aligned}$$

3.3 Aufgabe 3-3

[Anfragen in relationaler Algebra]

3.3.1 a)

$$\pi_{\text{pname}}\left(\sigma_{\text{ort}='\text{BERLIN}'}(P)\right)$$

3.3.2 b)

$$\pi_{\text{tname}}\left(\sigma_{\text{P.ort}} =' \text{BERLIN}' \wedge P.pnr = LTP.pnr \wedge T.tnr = LTP.tnr(P \times (T \times LTP))\right)$$
$$\pi_{\text{tname}}\left(\sigma_{\text{P.ort}='\text{BERLIN}'}(P \bowtie (LTP \bowtie T))\right)$$

3.3.3 c)

$$\pi_{\text{tname , tnr}}\ (\sigma_{\text{T.tnr}=LTP.tnr \wedge L.lnr=LTP.lnr \wedge \text{L. lname}='SCHULZ'}(\mathsf{T} \times (\mathsf{L} \times \mathsf{LTP})))$$

$$\pi_{\text{tname,tnr}}\ (\sigma_{\text{L.lname}='SCHULZ'}(\mathsf{T} \bowtie (\mathsf{L} \bowtie \mathsf{LTP})))$$

3.3.4 d)

$$\pi_{\text{lname}}\ \left(\sigma_{\text{L.sitz}='\ \text{MESCHEDE}\ ' \wedge \text{L. lnr} = LTP.\ \text{lnr}\ \wedge P.pnr=LTP.pnr \wedge\ \text{P.ort}='\ \text{WETTER}\ '}(\mathsf{L} \times \mathsf{P} \times \mathsf{LTP})\right)$$

$$\pi_{\text{lname}}\ (\sigma_{L.sitz='MESCHEDE' \wedge P.ort='WETTER'}(\mathsf{L} \bowtie \mathsf{P} \bowtie \mathsf{LTP}))$$

3.3.5 e)

$$\pi_{pnr,ort}\ \left(\sigma_{\text{T.farbe}='ROT' \wedge \text{T.gewicht}\ >5 \wedge T.tnr=LTP.tnr \wedge LTP.pnr=P.pnr}(\mathsf{T} \times (\mathsf{LTP} \times \mathsf{P}))\right)$$

$$\pi_{pnr,ort}\ \left(\sigma_{\text{T.farbe}='ROT' \wedge \text{T.gewich t}>5}(\mathsf{T} \bowtie (\mathsf{LTP} \bowtie \mathsf{P}))\right)$$

3.4 Aufgabe 3-4

[Anfragen mit dem Quotient-Operator]

(i) Nummern der Lieferanten, die an mind. ein Projekt jedes rote Teil in gleicher Menge liefern
(ii) Nummern der Lieferanten, die an mind. ein Projekt jedes rote Teil liefern (L1, L5)
(iii) Nummern der Lieferanten, die jedes rote Teil liefern

4 Übungsblatt 4

TODO:THIS IS OK. MAYBE I DO THIS A LITTLE LATER>

5 Übungsblatt 5

5.1 Aufgabe 5-1

5.1.1 a)

```
Tupelkalkül:
Schema(t) = (Nanme : String)
{ t | \exists l \in Lieferant :
t.Name = l.Name \wedge l.Land = 'Sachsen'}
Or,
Schema(l) = Schema(Lieferant)
{[l.Name] | l \in Lieferant \wedge l.Land = 'Sachsne'}
Bereichskalkül:
{ n | \exists nr, st, la :
Lieferant(nr, n, st, la) \wedge la = 'Sachsen'}
Or,
{ n | Lieferant(_, n, _, 'Sachsen')}
```

5.1.2 b)

```
Tupelkalkül:
Schema(t) = (Nummer : Integer, Name : String, Bestand : Integer, Preis: Dec
{ t | (\exists ar \in Artikel, ab \in Abteilung)
    (t.Nummer = ar.Nummer \wedge t.Name = ar.Name \wedge
    t.bestand = ar.Bestand \wedge t.Preis = ar.Preis
    ar.Abteilung = ab.Nummer \wedge ab.Name = 'Eletronik; ) }
Or,
Schema(ar) = Schema(Artikel)
{ [ar.Nummer, ar.Name, ar.Bestand, ar.Preis] |
  Artikel(ar) \wedge (\exists ab \in Abteilung:ar.Abteilung  = ab.Nummer
  \wedge ab.Nummer \wedge ab.Name = 'Eletronik')}
Bereichskalkül:
{ nr, na, be, pr | \exists abNr:
  Artikel(nr, na, abNr, pr, be, _) \wedge
  Abteilung(abNr, 'Eletronik', _, _, _)}
```

5.1.3 c)

```
Tupelkalkül:
Schema(t) = (Name : String, Preis : Decimal)
{ t | (\exists art \in Artikel, abt \in Abteilung, f \in FIliale)
 (t.Name = art.Name \wedge t.Preis = art.Preis \wedge
 art.Abteilung = abt.Nummer \wedge abt.FIliale = f.Nummer \wedge
 f.Stadt = 'Hamburg')}
Or,
Schema(art) = Schema(Artikel)
{ [art.Name, art.Preis] | Artikel(art) \wedge
```

```
  (\exists abt \in Abteilung, f \in Filiale)
  (art.Abteilung = abt.Nummer \wedge abt.Filiale = f.Nummer \wedge
  f.Stadt = 'Hamburg' ) }
Bereichskalkül:
{ na, pr | \exists abtNr, filNt :
  Artikel(_, na, abtNr, pr, _, _) \wedge
  Abteilung(abtNr, _, filNr, _, _) \wedge
  Filiale(filNr, 'Hamburg', _) }
```

5.1.4 d)

```
Tupelkalkül:
Schema(t) = (AName : String, LName : String, Geburtsjahr : Integer)
{ t | (\exists abt \in Abteilung, ang \in Angestellter :
 t.AName = abt.Name\wedge t.Lname = ang.Name \wedge
 t.Geburtsjahr = ang.Geburtsjahr \wedge
 abt.Leiter = ang.Nummer) }
Or,
Schema(abt) = Schema(Abteilung)
Schema(ang) = Schema(Angestellter)
{ [abt.Name, ang.Name, ang. Geburtsjahr] |
  Abteilung(abt) \wedge Angestellter(ang) \ wedge
  abt.Leiter = ang.Nummer }
Bereichskalkül:
{ aname, lname, geb | \exists lnr :
  Abteilung(_, aname, _, _, lnr) \wedge
  Angestellter(lnr, lname, _, _, geb, _)}
```

5.1.5 e)

```
Tupelkalkül:
Schema(t) = (Artikel)
{ art | Artikel(art) \wedge
  (\exists an \in Angestellter, ab \in Abteilung, v \in Verkauf
  art.Nummer = v.Artikel \wedge v.Abteilung = ab.Nummer \wedge
  ab.Leiter = an.Nummer \wedge
  an.Nummer = 'Edgar F. Codd' \wedge an.Geburtsjahr = 1923 ) }

Bereichskalkül:
{ nr, na, abt, pr, be, l | \exists abtNr, lnr :
  Artikel(nr, na, abt, pr, be, l) \wedge
  Verkauf(_, _, abtNr, nr, _, _, _) \wedge
  Abteilung(abtNr, _, _, _, lnr) \wedge
  Angestellter(lnr, 'Edgar F. Codd', _, _, 1923, _) }
```

5.2 Aufgabe 5-2

5.2.1 a)

$\sigma_{A=x}(R(A,B,C))$

```
Tupelkalkül:
Schema(t) = Schema(R)
{ t | R(t) \wedge t.A = x }

Bereichskalkül:
{ a, b, c | R(a,b,c) \wedge a = x}
```

5.2.2 b)

$\Pi_{A,B}(R(A,B,C))$

```
Tupelkalkül:
Schema(t) = (A : dom(R.A), B : dom(R.b))
{ t | \exists r \in R : (t.A = r.A \wedge t.B = r.B)}

Bereichskalkül:
{ a, b | \exists c : R(a,b,c)}
```

5.2.3 c)

$R(A,B,C) \bowtie S(C,D,E)$

```
Tupelkalkül:
Schema(t) = (A : dom(R.A), B : dom(R.B), C : dom(R.C), D : dom(S.D), E : dom
{ t | \exists r \in R, s \in S: (t.A = r.A \wedge
t.B = r.B \wedge t.C = r.C \wedge t.D = s.D \wedge
t.E = S.E \wedge r.C = s,C)}

Bereichskalkül:
{ a, b, c, d, e | R(a,b,c) \wedge S(c,d,e)}
```

5.2.4 d)

$R(A,B,C) \cup S(A,B,C)$

```
Tupelkalkül:
Schema(t) = Schema(R)
{ t | R(t) \vee S(t)}
Bereichskalkül:
{ a, b, c | R(a,b,c) \vee S(a,b,c)}
```

5.2.5 e)

$R(A, B, C) \cap S(A, B, C)$

```
Tupelkalkül:
Schema(t) = Schema(R)
{ t | R(t) \wedge S(t)}
Bereichskalkül:
{ a, b, c | R(a,b,c) \wedge S(a,b,c)}
```

5.2.6 f)

$R(A, B, C) - S(A, B < C)$

```
Tupelkalkül:
Schema(t) = Schema(R)
{ t | R(t) \wedge \neg S(t)}
Bereichskalkül:
{ a, b, c | R(a,b,c) \wedge \neg S(a,b,c)}
```

5.2.7 g)

$R(A, B, C) \times S(D, E, F)$

```
Tupelkalkül:
Schema(t) = (A : dom(R.A), B : dom(R.B), C : dom(R.C),
 D : dom(S.D), E : dom(S.E), F : dom(S.F))
{ t | \exists r \in R, s \in S: (t.A = r.A \wedge
t.B = r.B \wedge t.C = r.C \wedge t.D = s.D \wedge
t.E = s.E \wedge t.F = s.F)}
Bereichskalkül:
{ a, b, c, d, e, f | R(a,b,c) \wedge S(d,e,f)}
```

5.2.8 h)

$R(A, B) \div S(A)$

```
Tupelkalkül:
{ t | \forall s \in S \exists r \in R: s.A = r.A \wedge
 t.B = r.B}
Bereichskalkül:
{ b | \forall a: S(a) \Rightarrow R(a,b)}
```