Kapitel 11: Relationale Entwurfstheorie

Problem

Bestellungen (<u>Datum, KNr, PNr</u>, Bez, Preis, Gewicht, Menge)

"Typische" Ausprägung:

Datum	**KNr**	**PNr**	**Bez**	**Preis**	**Gewicht**	**Menge**
16.7.	1	1	Papier	20.00	2.000	100
21.7.	1	1	Papier	20.00	2.000	200
26.10.	2	1	Papier	20.00	2.000	100
26.10.	2	5	Disketten	20.00	0.500	50

11.1 Funktionalabhängigkeiten (FAs, FDs)

Definition:

Für $X = \{X1, ..., Xm\} \subseteq sch(R)$ und $Y = \{Y1, ..., Yn\} \subseteq sch(R)$ gilt die *Funktionalabhängigkeit* $X \rightarrow Y$, wenn jederzeit für je zwei Tupel $r, s \in val(R)$ gelten muß:

$(r.X1=s.X1 \wedge ... \wedge r.Xm=s.Xm) \Rightarrow (r.Y1=s.Y1 \wedge ... \wedge r.Yn=s.Yn)$

Beispiel:

Bestellungen (<u>Datum, KNr, PNr</u>, Bez, Preis, Gewicht, Menge)

Es gelten:

Datum, KNr, PNR $\rightarrow$ Menge

PNr $\rightarrow$ Bez, Preis, Gewicht

Es gelten *nicht*:

KNr, PNR $\rightarrow$ Menge

PNr $\rightarrow$ Datum

Entwurfsziel: Alle FAs in Primärschlüsselbedingungen ausdrücken

Ableitungsregeln für Funktionalabhängigkeiten

Definition:

Sei F eine Menge von FAs über sch(R). Die **transitive Hülle F^+** ist die Menge der aus F logisch ableitbaren FAs.

Ableitungskalkül (Armstrong-Regeln):

Seien $X \subseteq sch(R)$, $Y \subseteq sch(R)$, $Z \subseteq sch(R)$.

Regel 1 (Reflexivität): Wenn $Y \subseteq X$, dann gilt: $X \rightarrow Y$.

Regel 2 (Erweiterung): Wenn $X \rightarrow Y$, dann gilt: $XZ \rightarrow YZ$

Regel 3 (Transitivität): Wenn $X \rightarrow Y$ und $Y \rightarrow Z$, dann gilt: $X \rightarrow Z$.

Weitere Ableitungsregeln:

Regel 4 (Vereinigung): Wenn $X \rightarrow Y$ und $X \rightarrow Z$, dann gilt: $X \rightarrow YZ$.

Regel 5 (Pseudotrans.): Wenn $X \rightarrow Y$ und $WY \rightarrow Z$ mit $W \subseteq sch(R)$, dann gilt: $XW \rightarrow Z$.

Regel 6 (Zerlegung): Wenn $X \rightarrow Y$ und $Z \subseteq Y$, dann gilt: $X \rightarrow Z$.

Satz: Die Armstrong-Regeln bilden einen korrekten und vollständigen Ableitungskalkül für F^+.

Beispiel: Ableitung von Funktionalabhängigkeiten

Bestellungen (Datum, KNr, PNr, Bez, Preis, Gewicht, Menge)

F = { Datum, KNr, PNR $\rightarrow$ Menge,
PNr $\rightarrow$ Bez, Preis, Gewicht }.

Es folgen:

Datum, KNr, PNR $\rightarrow$ PNr	nach der Reflexivitätsregel,
Bez, Preis, Gewicht $\rightarrow$ Bez	nach der Reflexivitätsregel,
PNr $\rightarrow$ Bez	nach der Transitivitätsregel,
Datum, KNr, PNr $\rightarrow$ Bez	nach der Transitivitätsregel

Ableitbarkeitstest für FAs: Xplus-Algorithmus

Definition:

Sei R eine Relation mit sch(R) und FA-Menge F, und sei $X \subseteq$ sch(R).
Die *Hülle* X^+ von X ist die Menge aller $A \in$ sch(R) mit $X \rightarrow A \in F^+$.

Xplus-Algorithmus:

```
procedure xplus (X: set of attribute): set of attribute;
var H: set of attribute;
      newattr: Boolean;
begin
      H := X; newattr := true;
      while newattr do (* Invariante: X → H *)
          newattr := false;
          for each Y → Z ∈ F do
              if Y ⊆ H then
                  if not (Z ⊆ H) then H := H ∪ Z; newattr := true; fi; fi;
          od
      od
      return H;
end xplus;
```

Beispiel: Xplus-Algorithmus

$sch(R) = \{A, B, C, D, E, G\}$

$F = \{AB \rightarrow C,\ ACD \rightarrow B,$

$\quad BC \rightarrow D,\ BE \rightarrow C,$

$\quad C \rightarrow A,\ CG \rightarrow BD,\ CE \rightarrow AG,$

$\quad D \rightarrow EG\}$

$X = \{B, D\}$

Bestimmung aller Schlüsselkandidaten

Definition:

Sei R eine Relation mit sch(R) und FA-Menge F, und sei $X \subseteq$ sch(R).

$X \subseteq$ sch(R) ist ein *Schlüsselkandidat*, wenn $X \rightarrow$ sch(R) $\in$ F+ gilt und es keine echte Teilmenge von X gibt, die diese Eigenschaft hat.

$A \in$ sch(R) ist Schl.attribut, wenn es in einem Schl.kandidaten vorkommt.

```
procedure findkeys: set of set of attribute;
var  K: set of set of attribute; iskey: Boolean;
     K := ∅;
     for each S ∈ 2^sch(R) do
          iskey := true;
          if xplus (S) ≠ sch(R) then iskey := false fi;
          for each A ∈ S while iskey do
               if xplus (S - {A}) = sch(R) then iskey := false fi
          od;
          if iskey then K := K ∪ {S} fi
     od;
     return K;
```

11.2 Relationale Normalformen

Definition:
Sei R eine Relation mit sch(R) und FA-Menge F.
R ist in ***Boyce-Codd-Normalform (BCNF),*** wenn für jede FA
$X \rightarrow A \in F+$ mit $X \subseteq sch(R)$, $A \in sch(R)$ und $A \notin X$ gilt,
daß X einen Schlüsselkandidaten enthält.

Definition:
Sei R eine Relation mit sch(R) und FA-Menge F.
R ist in ***3. Normalform (3NF),*** wenn für jede FA
$X \rightarrow A \in F+$ mit $X \subseteq sch(R)$, $A \in sch(R)$ und $A \notin X$ gilt,
daß X einen Schlüsselkandidaten enthält
oder A ein Schlüsselattribut ist.

Satz:
Jede Relation in BCNF ist auch in 3NF.

Algorithmus für BCNF-Test

```
procedure BCNFtest: Boolean;
var isbcnf: Boolean;
begin
    isbcnf := true;
    for each X → A ∈ F while isbcnf do
        if A ∉ X then if XPlus(X) ≠ sch(R) then isbcnf := false fi fi;
    od
    return isbcnf;
end BCNFtest;
```

Beispiele: Normalformen

1) Best (Datum, KNr, PNr, Menge) mit
   F = {Datum, KNr, PNr $\rightarrow$ Menge}
   und
   Prod (PNr, Bez, Preis, Gewicht) mit
   F = {PNr $\rightarrow$ Bez, Preis, Gewicht}

2) Vorlesungen (Titel, Zeit, Raum) mit
   F = {Titel $\rightarrow$ Raum,
   Zeit, Raum $\rightarrow$ Titel}

   Beispielausprägung:

Titel	**Zeit**	**Raum**
Datenbanken	Di 9-11	HS 2
Datenbanken	Do 9-11	HS 2
Compiler	Di 9-11	HS 3
Compiler	Mi 13-15	HS 3
Betriebssysteme	Mi 13-15	HS 2

11.3 Relationendekomposition

Definition:
Sei R eine Relation mit sch(R) und FA-Menge F.
Eine Zerlegung von R in R1 (X1), ..., Rk (Xk) mit Xi $\subseteq$ sch(R) und X1 $\cup$ .. $\cup$ Xk = sch(R) heißt ***abhängigkeitsbewahrend***, wenn gilt:

$$\left((F^+|_{X1})^+ \cup \ldots \cup (F^+|_{Xk})^+ \right)^+ = F^+$$

Definition:
Sei R eine Relation mit sch(R) und FA-Menge F.
Eine Zerlegung von R in R1 (X1), ..., Rk (Xk) mit Xi $\subseteq$ sch(R) und X1 $\cup$ ... $\cup$ Xk = sch(R) heißt ***(projektion-join-) verlustfrei***,
wenn für alle möglichen Ausprägungen, die F erfüllen, gilt:

$$\pi[X1](R) \;|\times|\; \ldots \;|\times|\; \pi[Xk](R) = R.$$

Beispiele: Abhängigkeitsbewahrung

1) Bestellungen (Datum, KNr, PNr, Bez, Preis, Gewicht, Menge}mit
   F = {Datum, KNr, PNR $\rightarrow$ Menge,
   PNr $\rightarrow$ Bez, Preis, Gewicht}
   Zerlegung in
   Best (Datum, KNr, PNr, Menge) mit
   F1 = {Datum, KNr, PNR $\rightarrow$ Menge}
   und
   Prod (PNr, Bez, Preis, Gewicht) mit
   F2 = {PNr $\rightarrow$ Bez, Preis, Gewicht}

2) Vorlesungen (Titel, Zeit, Raum) mit
   F = {Titel $\rightarrow$ Raum,
   Zeit, Raum $\rightarrow$ Titel}
   Zerlegung in
   Vorlesungsräume (Titel, Raum) mit
   F1 = {Titel $\rightarrow$ Raum}
   und
   Vorlesungszeiten (Titel, Zeit) mit F2 = $\emptyset$

Beispiele: Verlustfreiheit

Bestellungen (Datum, KNr, PNr, Bez, Preis, Gewicht, Menge) mit

F = {Datum, PNr, KNr $\rightarrow$ Menge, PNr $\rightarrow$ Bez, Preis, Gewicht}

Datum	**KNr**	**PNr**	**Bez**	**Preis**	**Ge-wicht**	**Menge**
16.7.	1	1	Papier	20.00	2.000	100
16.7.	1	5	Disketten	20.00	0.500	50

Zerlegung in:

Datum	**KNr**	**PNr**	**Menge**
16.7.	1	1	100
16.7.	1	5	50

und

Datum	**Bez**	**Preis**	**Gewicht**
16.7.	Papier	20.00	2.000
16.7.	Disket-ten	20.00	0.500

Verlustfreie BCNF-Dekomposition

Satz: Zu jeder Relation R gibt es eine abhängigkeitsbewahrende und verlustfreie Zerlegung von R in 3NF-Relationen.

Satz: Zu jeder Relation R gibt es eine verlustfreie Zerlegung von R in BCNF-Relationen..

Satz: Sei R eine Relation mit sch(R) und einer FA-Menge F.
Sei $X \rightarrow Y \in F+$ mit $X \cap Y = \emptyset$. Die Zerlegung von R in R' (X, Y) und R'' (X, sch(R) - Y) ist verlustfrei.

Algorithmus:

Teste, ob R in BCNF ist
Falls R nicht in BCNF ist
(* es gibt eine Funktionalabhängigkeit $X \rightarrow Y \in F+$ mit
$X \cap Y = \emptyset$ und $X \rightarrow sch(R) \notin F+$ *)
dann
zerlege R in R' (X, Y) und R'' (X, sch(R) - Y)
wiederhole den Zerlegungsalgorithmus für R' und R''

Beispiel: Relationendekomposition

R (FlugNr, Datum, Abflugzeit, FSNr, SitzNr, TicketNr, Name, Adresse, GNr, Gewicht)

F = {FlugNr, Datum $\rightarrow$ Abflugzeit, FSNr,
FlugNr, Datum, TicketNr $\rightarrow$ SitzNr,
TicketNr $\rightarrow$ Name, Adresse,
GNr $\rightarrow$ Gewicht }

Schlüsselkandidat:
<u>FlugNr, Datum, TicketNr, GNr</u>

Zerlegungsschritte:

1. Zerlegung von R entlang von FlugNr, Datum $\rightarrow$ Abflugzeit, FSNr:
   R1 (<u>FlugNr, Datum</u>, Abflugzeit, FSNr) und
   R2 (FlugNr, Datum, SitzNr, TicketNr, Name, Adresse, GNr, Gewicht)
2. Zerlegung von R2 entlang von FlugNr, Datum, TicketNr $\rightarrow$ SitzNr:
   R21 (<u>FlugNr, Datum, TicketNr</u>, SitzNr) und
   R22 (FlugNr, Datum, TicketNr, Name, Adresse, GNr, Gewicht)
3. Zerlegung von R22 entlang von TicketNr $\rightarrow$ Name, Adresse:
   R221 (<u>TicketNr</u>, Name, Adresse) und
   R222 (TicketNr, FlugNr, Datum, GNr, Gewicht)
4. Zerlegung von R222 entlang von GNr $\rightarrow$ Gewicht:
   R2221 (<u>GNr</u>, Gewicht) und
   R2222 (<u>GNr, FlugNr, Datum, TicketNr</u>)

11.4 Relationensynthese

Definition:

Sei eine Relation mit sch(R) und FA-Menge F. Eine FA-Menge F' heißt ***Überdeckung*** (engl. cover) von F, wenn gilt: $(F')^+ = F^+$.

Definition:

Eine Überdeckung F' von F heißt ***minimal***, wenn gilt:

1) In jeder FA von F' hat die rechte Seite ein Attribut.
2) Keine echte Teilmenge von F' ist eine Überdeckung von F.
3) Für jede FA $X \rightarrow A \in F'$ und jede echte Teilmenge X' von X
   $F'' := F' - \{ X \rightarrow A\} \cup \{X' \rightarrow A\}$ keine Überdeckung mehr von F.

Beispiel: minimale Überdeckung

sch(R) = {Ang(estellten)Nr, Name, Gehalt, Leistung(sbeurteilung), Tarif(klasse), Abt(eilungs)Nr, ProjektNr, Projektleiter, Abt(eilungs)leiter}

F = { AngNr $\rightarrow$ Name,
AngNr $\rightarrow$ Gehalt,
AngNr $\rightarrow$ Tarif,
AngNr $\rightarrow$ AbtNr,
AngNr $\rightarrow$ Abtleiter,
ProjektNr $\rightarrow$ Projektleiter,
AbtNr $\rightarrow$ Abtleiter,
AngNr, Abtleiter $\rightarrow$ Leistung,
Tarif, Leistung $\rightarrow$ Gehalt}

Schritt 1: $\rightarrow$ { AngNr $\rightarrow$ Name,
AngNr $\rightarrow$ Gehalt,
AngNr $\rightarrow$ Tarif,
AngNr $\rightarrow$ AbtNr,
AngNr $\rightarrow$ Abtleiter,
ProjektNr $\rightarrow$ Projektleiter,
AbtNr $\rightarrow$ Abtleiter,
AngNr $\rightarrow$ Leistung,
Tarif, Leistung $\rightarrow$ Gehalt}

Schritt 2: $\rightarrow$ { AngNr $\rightarrow$ Name,
AngNr $\rightarrow$ Tarif,
AngNr $\rightarrow$ AbtNr,
ProjektNr $\rightarrow$ Projektleiter,
AbtNr $\rightarrow$ Abtleiter,
AngNr $\rightarrow$ Leistung,
Tarif, Leistung $\rightarrow$ Gehalt} = F'

Algorithmus zur Berechnung der Min-Cover

```
procedure mincover: set of FDs;
  var G: set of FDs;
  procedure xplus (arg1: set of FDs, arg2: set of attribute): set of attribute;
  procedure reduce (arg: set of FDs): set of FDs;
    var H: set of FDs;
    H := arg;
    for each X→A∈H do
      for each C∈X do
        Hred := H - { X→A} ∪ {(X-{C})→A};
        if A ∈ xplus (H, X-{C})) then H := Hred  fi;
      od;
    od;
    return H;

G := reduce(F);
for each X→A∈G do
  G := G - { X→A};
  if not (A ∈ xplus (G, X)) then G := G ∪ { X→A} fi;
od;
return G;
```

Algorithmus zur Relationensynthese

Schritt 1: Bereche eine minimale Überdeckung F' von F.

Schritt 2: Partitioniere die FAs in F' nach gleichen linken Seiten, d.h. für jedes Xi fasse alle $Xi \rightarrow A1$, ..., $Xi \rightarrow Ak_i$ zusammen

Schritt 3: Bilde für jede Partition von Schritt 2 eine Relation Ri mit $sch(Ri) = Xi \cup \{Ai, ..., Ak_i\}$.

Schritt 4: Falls $sch(R) - (\cup_i sch(Ri))$ nichtleer ist, bilde eine weitere Relation R' mit $sch(R') = sch(R) - (\cup_i sch(Ri))$.

Schritt 5: Falls keine der Relationen Ri, R' einen Schlüssel von R enthält, erzeuge eine Relation R'', deren Schema ein Schlüssel von R ist.

Satz:

Der Algorithmus zur Relationensynthese erzeugt eine abhängigkeitsbewahrende und verlustfreie 3NF-Zerlegung von R.

Beispiel: Relationensynthese

sch(R) = {Ang(estellten)Nr, Name, Gehalt,
Leistung(sbeurteilung), Tarif(klasse),
Abt(eilungs)Nr, ProjektNr, Projektleiter, Abt(eilungs)leiter}

F' = { AngNr → Name,
AngNr → Tarif,
AngNr → AbtNr,
ProjektNr → Projektleiter,
AbtNr → Abtleiter,
AngNr → Leistung,
Tarif, Leistung → Gehalt}

Ergebnis der Relationensynthese:

Angestellte (<u>AngNr</u>, Name, Tarif, Leistung, AbtNr)
Projekte (<u>ProjektNr</u>, Projektleiter)
Abteilungen (<u>AbtNr</u>, Abtleiter)
Lohn (<u>Tarif, Leistung</u>, Gehalt)
Projektmitarbeit (<u>AngNr, ProjektNr</u>)

11.5 Ergänzungen (1)

1) weitere Zerlegung von BCNF-Relationen:

Angestellte (<u>ANr, Informatikkenntnisse, Kinder</u>) mit F = ∅

ANr	**Informatikkenntnisse**	**Kinder**
1	Leda	Sabrina
1	Oracle	Sabrina
2	Leda	Pierre
2	Leda	Sabrina

sollte weiter zerlegt werden in

AngInfKenntnisse (<u>ANr, Informatikkenntnisse</u>) und
AngKinder (<u>ANR, Kinder</u>)

2) unnötig feine Zerlegungen:

Produkte (PNr, Bez, Preis) mit F = {$PNr \rightarrow Bez,\quad PNr \rightarrow Preis$}

sollte nicht zerlegt werden in

Produktbez (<u>PNr</u>, Bez) und
Produktpreise (<u>PNr</u>, Preis)

11.5 Ergänzungen (2)

3) Zerlegung aufgrund irrelevanter Funktionalabhängigkeiten:

Bestellungen (Datum, KNr, PNr, Bez, Preis, Gewicht, Menge, Summe)

mit F = {*Datum, KNr, PNr* $\rightarrow$ *Menge,*
PNr $\rightarrow$ *Bez, Preis, Gewicht,*
Preis, Menge $\rightarrow$ *Summe*}

sollte nicht zur Relatione führen

Preise (<u>Preis, Menge</u>, Summe)

4) Nicht-BCNF-Relationen, die unproblematisch sind:

Kunden (KNr, Name, Postleitzahl, Stadt, Strasse)

mit F = {*KNr* $\rightarrow$ *Name, Postleitzahl, Stadt, Strasse,*
Postleitzahl $\rightarrow$ *Stadt* }

muß nicht notwendigerweise zerlegt werden in

Kundenadressen (KNr, Name, Postleitzahl, Strasse) und
Postleitzahlenverzeichnis (Postleitzahl, Stadt)

5) M:N-Relationships ohne Attribute:

Sitze (<u>FlugNr, Datum, SitzNr</u>)

muß bei der Relationendekomposition ggf. hinzugefügt werden