

Kapitola 7: Návrh relačních databází

- Nástrahy návrhu relačních databází
- Dekompozice (rozklad)
- Normalizace použitím funkčních závislostí

Nástrahy relačního návrhu

- Návrh relačních databází vyžaduje nalézt nějakou dobrou množinu relačních schémat. Špatný návrh může vést k:
 - Opakování stejné informace
 - Nemožnosti vyjádřit nějakou informaci
- Cíle návrhu:
 - Zabránit redundanci (opakování) dat
 - Zajistit vyjádření všech vztahů mezi atributy
 - Usnadnit kontrolu porušení integritních omezení databáze při změnách dat

Příklad

- Uvažujme relační schéma:
schéma-půjček = (jméno-pobočky, město-pobočky, aktiva, jméno-zákazníka, číslo-půjčky, zůstatek)
- Redundance (opakování, nadbytečnost):
 - Data o *jméno-pobočky, město-pobočky, aktiva* jsou opakována pro každou půjčku, kterou pobočka vytvoří
 - Plýtvá místem a komplikuje provádění změn
- Null hodnoty
 - Nelze uložit informace o pobočce, pokud neexistuje žádná půjčka
 - Lze použít prázdné (null) hodnoty, ale je obtížné s nimi pracovat

Rozklad (dekompozice)

- Rozložit relační schéma *schéma-půjček* do:
pobočka-zákazník-schéma = (jméno-pobočky, město-pobočky, aktiva, jméno-zákazníka)
zákazník-půjčka-schéma = (jméno-zákazníka, číslo-půjčky, zůstatek)
- Všechny atributy původního schématu R se musí objevit v rozkladu (R_1, R_2) :
$$R = R_1 \cup R_2$$
- Rozklad bezetrátového spojení
Pro všechny možné relace r na schématu R platí
$$r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$$

Příklad bezetrátového rozkladu

- Dekompozice schématu $R=(A,B)$

$R_1=(A)$	
A	B
α	1
α	2
β	1

$R_2=(B)$
A
α
β

$\Pi_B(r)$
B
1
2

- $\Pi_A(r) \bowtie \Pi_B(r)$

A	B
α	1
α	2
β	1
β	2

Cíl – navrhnout teorii

- Rozhodnout, zda-li určité relační schéma je v „dobrém“ tvaru.
- V případě, že schéma R není v „dobrém“ tvaru, rozlož jej na množinu $\{R_1, R_2, \dots, R_n\}$ takovou, že
 - Každá relace R_i je v „dobrém“ tvaru
 - Rozklad je bezetrátový
- Tato teorie je založená na:
 - Funkčních závislostech
 - Vícehodnotových závislostech

Normalizace pomocí funkčních závislostí

Pokud dekomponujeme relační schéma R s množinou funkčních závislostí F do schémat R_1 a R_2 , požadujeme aby:

- Bezeztrátovost spojení: nejméně jedna z následujících závislostí je v F^+ :
 - $R_1 \cap R_2 \rightarrow R_1$
 - $R_1 \cap R_2 \rightarrow R_2$
- Žádná redundance: relační schémata R_1 a R_2 by nejlépe měly být v Boyce-Coddově normální formě nebo ve třetí normální formě.
- Uchování závislostí: necht' F_i je množina závislostí v F^+ , která obsahuje pouze atributy ve schématu R_i , ověříme, že platí:
 - $(F_1 \cup F_2)^+ = F^+$
 Jinak je test na porušení funkčních závislostí příliš drahý.

Příklad

- $R = (A, B, C)$
 $F = \{A \rightarrow B, B \rightarrow C\}$
- $R_1 = (A, B), R_2 = (B, C)$
 - Rozklad pomocí bezeztrátového spojení:
 $R_1 \cap R_2 = \{B\}$ a $B \rightarrow BC$
 - Zachovává závislosti
- $R_1 = (A, B), R_2 = (A, C)$
 - Rozklad pomocí bezeztrátového spojení:
 $R_1 \cap R_2 = \{A\}$ a $A \rightarrow AB$
 - Nezachovává závislosti
(nelze otestovat $B \rightarrow C$ bez výpočtu spojení $R_1 \bowtie R_2$)

Boyce-Coddova normální forma

Relační schéma R je v BCNF vzhledem k množině funkčních závislostí F , pokud pro všechny funkční závislosti v F^+ tvaru $\alpha \rightarrow \beta$, kde $\alpha \subseteq R$ a $\beta \subseteq R$, je splněna alespoň jedna z podmínek:

- $\alpha \rightarrow \beta$ je triviální (tj. $\beta \subseteq \alpha$)
- α je superklíč schématu R

Příklad

- $R = (A, B, C)$
 $F = \{A \rightarrow B, B \rightarrow C\}$, klíč = $\{A\}$
- R není v BCNF
- Rozklad na $R_1 = (A, B), R_2 = (B, C)$
 - R_1 a R_2 je v BCNF
 - Splněna podmínka bezeztrátovosti spojení
 - Zachovává závislosti

BCNF – algoritmus rozkladu

```

result := {R};
done := false;
vypočítej  $F^+$ ;
while (not done) do
  if (existuje schéma  $R_i$  v result, které není v BCNF) then
    nechť  $\alpha \rightarrow \beta$  je netriviální funkční závislost, která je splněna na  $R_i$ 
    taková, že  $\alpha \rightarrow R_i$  není v  $F^+$  a  $\alpha \cap \beta = \emptyset$ ;
    result := (result -  $R_i$ )  $\cup$  ( $R_i - \beta$ )  $\cup$  ( $\alpha, \beta$ );
  else
    done := true;
fi

```

Poznámka: každé schéma R_i je v BCNF a rozklad je bezeztrátový podle spojení.

Příklad rozkladu do BCNF

- $R = (\text{jméno-pobočky}, \text{město-pobočky}, \text{aktiva}, \text{jméno-zákazníka}, \text{číslo-půjčky}, \text{zůstatek})$
 $F = \{\text{jméno-pobočky} \rightarrow \text{aktiva} \text{ město-pobočky}$
 $\text{číslo-půjčky} \rightarrow \text{zůstatek} \text{ jméno-pobočky}\}$
 $\text{Key} = \{\text{číslo-půjčky}, \text{jméno-zákazníka}\}$
- První rozklad
 - $R_1 = (\text{jméno-pobočky}, \text{město-pobočky}, \text{aktiva})$
 - $R_2 = (\text{jméno-pobočky}, \text{jméno-zákazníka}, \text{číslo-půjčky}, \text{zůstatek})$
- Druhý rozklad
 - $R_1 = (\text{jméno-pobočky}, \text{město-pobočky}, \text{aktiva})$
 - $R_3 = (\text{jméno-pobočky}, \text{číslo-půjčky}, \text{zůstatek})$
 - $R_4 = (\text{jméno-zákazníka}, \text{číslo-půjčky})$
- Výsledný rozklad:

$$R_1, R_3, R_4$$

BCNF a zachování závislostí

Vždy není možné vytvořit rozklad do BCNF normální formy, který zachovává funkční závislosti.

- $R = (J, K, L)$
 $F = \{JK \rightarrow L$
 $L \rightarrow K\}$
 Dva kandidátní klíče JK a JL
- R není v BCNF
- Jakýkoli rozklad schématu R nebude splňovat závislost:

$$JK \rightarrow L$$

Třetí normální forma

- Relační schéma R je ve třetí normální formě (3NF), jestliže pro všechny závislosti
 $\alpha \rightarrow \beta$ z F^+
 alespoň jedna podmínka z následujících platí:
 - $\alpha \rightarrow \beta$ je triviální (tj. $\beta \subseteq \alpha$)
 - α je superklíč schématu R
 - každý atribut A v $\beta - \alpha$ je obsažený v kandidátním klíči schématu R
- Pokud je relace v BCNF, pak je i v 3NF (protože v BCNF musí platit alespoň jedna z prvních dvou podmínek).
- Příklad:
 - $R = (J, K, L)$
 $F = \{JK \rightarrow L, L \rightarrow K\}$
 - Dva kandidátní klíče: JK a JL
 - R je ve 3NF

$JK \rightarrow L$	JK je superklíč
$L \rightarrow K$	K je obsažený v kandidátním klíči

- Algoritmus pro rozklad relačního schématu R do množiny relačních schémat $\{R_1, R_2, \dots, R_n\}$ zaručuje následující:
 - Každé relační schéma R_j je ve 3NF
 - Rozklad splňuje bezeztrátovost spojení
 - Zachovává závislosti

3NF algoritmus pro rozklad

Nechť F_C je kanonický uzávěr množiny F ;

$i := 0$;

for each funkční závislost $\alpha \rightarrow \beta$ z F_C **do**

if žádné ze schémat $R_j, 1 \leq j \leq i$ neobsahuje $\alpha\beta$ **then**

$i := i + 1$;

$R_i := \alpha\beta$;

if žádné ze schémat $R_j, 1 \leq j \leq i$ neobsahuje kandidátní klíč pro R **then**

$i := i + 1$;

$R_i :=$ libovolný kandidátní klíč pro R ;

return (R_1, R_2, \dots, R_n)

Příklad

- Relační schéma:
 - $\text{schéma-bankéř-info} = (\text{jméno-pobočky}, \text{jméno-zákazníka}, \text{jméno-bankéře}, \text{číslo-kanceláře})$
- Funkční závislosti pro toto schéma jsou:
 - $\text{jméno-bankéře} \rightarrow \text{jméno-pobočky}$ číslo-kanceláře
 - jméno-zákazníka $\text{jméno-pobočky} \rightarrow \text{jméno-bankéře}$
- Klíč schématu je:
 - $\{\text{jméno-zákazníka}, \text{jméno-pobočky}\}$
- **For** cyklus v algoritmu způsobí vytvoření schémat:
 - $\text{schéma-bankéř-kancelář} = (\text{jméno-bankéře}, \text{jméno-pobočky}, \text{číslo-kanceláře})$
 - $\text{schéma-bankéř} = (\text{jméno-zákazníka}, \text{jméno-pobočky}, \text{jméno-bankéře})$
- Protože schéma-bankéř obsahuje kandidátní klíč pro $\text{schéma-bankéř-info}$, jsme hotovi s rozkladem.

Porovnání BCNF a 3NF

- Vždy je možné provést rozklad schématu do více schématů, které jsou v 3NF, a
 - rozklad je bezeztrátový
 - závislosti jsou zachovány
- Vždy je možné provést rozklad schématu do více schématů, které jsou v BCNF, a
 - rozklad je bezeztrátový
 - všechny závislosti nemusí být zachovány

- $R = (J, K, L)$
 $F = \{JK \rightarrow L, L \rightarrow K\}$

- Uvažujme následující relaci:

J	L	K
j_1	l_1	k_1
j_2	l_1	k_1
j_3	l_1	k_1
<i>null</i>	l_2	k_2

- Schéma, které je v 3NF ale ne v BCNF, má následující problémy:
 - opakování informací (např. vztah l_1, k_1)
 - potřebuje používat prázdné hodnoty (null) (např. pro vyjádření vztahu l_2, k_2 nemáme odpovídající hodnotu pro atribut J).

Cíle návrhu

- Cíle návrhu relačních databází jsou:
 - BCNF
 - bezztrátové spojení
 - zachování závislostí
- Pokud předchozího nemůžeme dosáhnout, stačí nám:
 - 3NF
 - bezztrátové spojení
 - zachování závislostí

Další normální formy

- Existují i další normální formy, zejména:
 - 1. normální forma
 - * Relační schéma je v 1NF tehdy a jen tehdy, pokud každý atribut je atomický (tj. není vícehodnotový)
 - 2. normální forma
 - * Relační schéma je v 2NF právě, když je v 1NF a každý atribut závisí na klíči (primárním klíči), pozn. závislost může být i tranzitivní, závislost musí být na celém klíči nikoli jen na některé jeho části.