

## Navrhovanie databáz

- **Formálne metódy:**

identifikácia, či test objektov a optimalizácia návrhu databázy pre zvolený dátový model (relačný model)

- **Polo-formálne metódy:**

analýza reálnej skutočnosti a komunikácia s koncovým užívateľom (ER-model, NIAM, ...)

navrhovanie databáz

1

## Vlastnosti funkčných závislostí

(Armstrongové axiómy)

- (A1)  $x \subseteq y \Rightarrow y \rightarrow x$  reflexívnosť
- (A2)  $\forall z x \rightarrow y \Rightarrow xz \rightarrow yz$  augmetation
- (A3)  $(x \rightarrow y) \wedge (y \rightarrow z) \Rightarrow x \rightarrow z$  tranzitívnosť

Dôkaz dosadením do definície funkčnej závislosti:

$$(A1) (\forall xyz_1z_2)(x \subseteq y \wedge R(yz_1) \wedge R(yz_2) \Rightarrow x = z)$$

$$(A2) (\forall xzy_1y_2t_1t_2)(R(xzy_1t_1) \wedge R(xzy_2t_2) \Rightarrow y_1 = y_2) \\ (\forall xzy_1y_2t_1t_2)(R(xy_1zt_1) \wedge R(xy_2zt_2) \Rightarrow y_1z = y_2z)$$

$$(A3) (\forall xy_1z_1y_2z_2t_1t_2)(R(xy_1z_1t_1) \wedge R(xy_2z_2t_2) \Rightarrow y_1 = y_2 = y) \\ (\forall xyz_1z_2t_1t_2)(R(xyz_1t_1) \wedge R(xyz_2t_2) \Rightarrow z_1 = z_2)$$

navrhovanie databáz

3

## Základné pojmy pre navrhovanie v relačnom modeli

Závislosti -  $\forall \exists \wedge \Rightarrow$  (induktívne Hornové formuly)

- Funkčné závislosti  $x \rightarrow y$   
 $(\forall xy_1z_1y_2z_2)(R(xy_1z_1) \wedge R(xy_2z_2) \Rightarrow y_1 = y_2)$
- Multi-závislosti  $x \twoheadrightarrow y$  (multivalued dependencies)  
 $(\forall xy_1z_1y_2z_2)(R(xy_1z_1) \wedge R(xy_2z_2) \Rightarrow R(xy_1z_2))$
- Uzáver množiny závislostí  $F^*$  je množina všetkých závislostí, ktoré vyplývajú z  $F$ .
- Úplné pokrytie je  $F^+$  je priemet  $F^*$  na závislosti daného typu
- Pokrytie množiny závislostí  $F$  je ľubovoľná množina závislostí  $G$  taká, že  $F^+ = G^+$ .

navrhovanie databáz

2

## Ďalšie vlastnosti funkčných závislostí

- $(x \rightarrow y) \wedge (x \rightarrow z) \Rightarrow x \rightarrow yz$  (union rule)
- $(x \rightarrow y) \wedge (wy \rightarrow z) \Rightarrow wx \rightarrow wz$  (pseudotransitivity)
- $(x \rightarrow y) \wedge (z \subseteq y) \Rightarrow x \rightarrow z$  (decomposition)

Dôkazová technika:

$$\left. \begin{array}{l} x \rightarrow y \Rightarrow x \rightarrow xy \text{ podľa (A2)} \\ x \rightarrow z \Rightarrow xy \rightarrow yz \text{ podľa (A2)} \end{array} \right\} \Rightarrow x \rightarrow yz \text{ podľa (A3)}$$

Zvyšné dôkazy sa robia podobne podľa Armstrongových axióm. (Urobte ich ako cvičenie.)

navrhovanie databáz

4

## Uzáver množiny atribútov

Nech  $\mathbf{x}$  je množina atribútov a  $F$  je množina funkčných závislostí. Potom uzáverom  $\mathbf{x}^+$  množiny  $\mathbf{x}$  w.r.t.  $F$  rozumieme množinu  $\mathbf{x}^+$  všetkých atribútov  $\mathbf{x}$  takých, že  $\mathbf{x} \rightarrow \mathbf{y}$  pomocou závislostí v  $F$ .

Lema:  $\mathbf{x} \rightarrow \mathbf{y}$  sa dá odvodiť z  $F$  pomocou Armstrongových axióm práve vtedy keď  $\mathbf{y} \subseteq \mathbf{x}^+$  w.r.t.  $F$ .

Pre každý atribút  $a \in \mathbf{y} \subseteq \mathbf{x}^+$ . Platí  $\mathbf{x} \rightarrow a$  podľa definície uzáveru  $\mathbf{x}^+$ . Podľa union rule platí aj  $\mathbf{x} \rightarrow \mathbf{y}$ .

Naopak nech  $\mathbf{x} \rightarrow \mathbf{y}$  sa dá odvodiť. Potom pre každé  $a \in \mathbf{y}$  platí  $\mathbf{x} \rightarrow a$  podľa decomposition rule a  $a \in \mathbf{x}^+$ .

navrhovanie databáz

5

## Charaktrizácia úplného pokrytia množiny funkčných závislostí

Množina  $F^+$  je príliš obsiahla, stačí však uvádzať maximálne závislosti. Závislosť je maximálna ak nemôžeme vynechať žiaden atribút na ľavej strane alebo pridať nejaký atribút na pravú stranu bez porušenia jej platnosti.

K charakterizácii stačia nasýtené množiny = pravé strany maximálnych závislostí. (Spätná rekonštrukcia maximálnych závislostí.)

Veta: Každá úplná množina funkčných závislostí má model v nejakej relácii nad doménou  $D = \{0, 1\}$ .

navrhovanie databáz

7

## Úplnosť „Armstrongových axióm“

Veta: Funkčná závislosť  $\mathbf{x} \rightarrow \mathbf{y}$  sa dá odvodiť z  $F$  pomocou Armstrongových axióm práve vtedy, keď  $\mathbf{x} \rightarrow \mathbf{y}$  je dôsledkom  $F$ .

Dôkaz: Pretože Armstrongové axiómy sú dôsledkom definície funkčnej závislosti, dajú sa odvodiť len platné závislosti.

Opačne predpokladajme, že závislosť  $\mathbf{x} \rightarrow \mathbf{y}$  platí ale nedá sa odvodiť pomocou Armstrongových axióm. Uvažujme reláciu  $\mathbf{R}$

Atribúty $\mathbf{x}^+$	ostatné atribúty	
u: 1 1 ... 1	1 1 ... 1	Všetky závislosti z $F$ sú splnené v $\mathbf{R}$
v: 1 1 ... 1	0 0 ... 0	

Nech  $\mathbf{v} \rightarrow \mathbf{w}$  je dôsledkom  $F$ , ale nie je splnené v  $\mathbf{R}$ . Potom  $\mathbf{v} \subseteq \mathbf{x}^+$ . Predpoklad o  $\mathbf{w}$  vedie vždy k sporu s predošlou lemov.

navrhovanie databáz

6

## Vyplývanie medzi funkčnými závislosťami

Výpočet  $F^+$  a testovanie ekvivalencie  $F^+ = G^+$  je vo všeobecnosti náročná (exponenciálna) záležitosť. Našťastie stačí počítať uzavery množiny atribútov vzhľadom k  $F^+$ .

navrhovanie databáz

8

## Minimálne pokrytie množiny funkčných závislostí

Kánonické závislosti na pravej strane len jeden atribút.

Minimálne pokrytie je pokrytie kánonickými závislosťami z ktorých sa žiadna nedá vynechať bez toho, aby sa porušila vlastnosť byť pokrytím.

Minimálne pokrytia

AB → C	D → E	CG → B	AB → C	AB → C
C → A	D → G	CG → D	C → A	C → A
BC → D	BE → C	CE → A	BC → D	BC → D
ACD → B	CE → G	D → E	D → E	D → E
		D → G	D → G	D → G
		BE → C	BE → C	BE → C
		CE → G	CE → G	CE → G
		CD → B	CD → B	CD → B
		CG → D	CG → D	CG → D

navrhovanie databáz

9

## Bezstrátové spojenia

$\rho = \{R_1, \dots, R_k\}$ ,  $R = R_1 \cup \dots \cup R_k$

$m_\rho(r) = \Pi_{R_1}(r) \bowtie \dots \bowtie \Pi_{R_k}(r)$  Join project mapping

Vlastnosti:  $r \subseteq m_\rho(r)$   
 $m_\rho(r) = m_\rho(m_\rho(r))$

Hovoríme, že dekompozícia má bezstrátové spojenie ak  $r = m_\rho(r)$ .

Tabuľková metóda testovania.

R= SAIP	<u>S</u>	<u>A</u>	<u>I</u>	<u>P</u>
S → A	a <sub>1</sub>	a <sub>2</sub>	b <sub>13</sub>	b <sub>14</sub>
SI → P	a <sub>1</sub>	b <sub>23</sub>	a <sub>3</sub>	a <sub>4</sub>

navrhovanie databáz

11

## Nadklíúče a klíúče

Nech je daná relácia  $R(U)$ . Potom množinu atribútov  $K$  takú, že  $K \rightarrow U$  nazývame nadklíúč. Minimálny nadklíúč v zmysle množinovej inklúzie nazývame klíúč.

**Koľko klíúčov môže mať relácia o n atribútoch ?**

Príklad:

$R(A_1, \dots, A_k, B_1, \dots, B_k, C)$

$F = \{A_i \leftrightarrow B_i \text{ pre } 1 \leq i \leq k\} \cup \{A_1 \dots A_k \rightarrow C\}$

navrhovanie databáz

10

## Normálne formy (BCNF, 3NF)

BCNF: Relačná schéma R je v BCNF, keď pre každú v nej platnú funkčnú závislosť  $x \rightarrow y$  platí  $x$  je nadklíúč.

3NF: Relačná schéma R je v 3NF, keď pre každú v nej platnú funkčnú závislosť  $x \rightarrow y$  platí  $x$  je nadklíúč alebo  $y$  je prvok nejakého klíúča (primárny atribút) R.

Lema: a.) Každá binárna relácia je v BCNF.

b.) Ak R nie je v BCNF. Potom v nej existujú atribúty A a B také, že  $(R - AB) \rightarrow A$ .  
 (Môže a nemusí platiť  $(R - AB) \rightarrow B$ .)

navrhovanie databáz

12

## 3NF zachovávajúca závislosti

Príklad:  $R = \text{MAP}$  (Mesto, Adresa, PSČ)  
Závislosti:  $MA \rightarrow P, P \rightarrow M$

Hovoríme že dekompozícia  $R = R_1 \cup \dots \cup R_k$  zachováva závislosti  $F$ , ak každá závislosť z  $F$  je v uzávere tých závislostí  $x \rightarrow y$  z  $F$ , že  $xy \subseteq R_i$ .

Algoritmus testovania zachovania závislosti  $x \rightarrow y$   
 $z := x$ ; **while** sa z zmenilo **do**  
    **for**  $i := 1$  **to**  $k$  **do**  
         $z := z \cup ((z \cap R_i)^+ \cap R_i)$  {uzáver w.r.t.  $F$ };

navrhovanie databáz

13

## Algoritmus normalizácie podrobnejšie

```
Z := R;  
repeat bcnf:= decompose( Z, Y, A);  
    Z := Z - A;  
until bcnf;  
function decompose( Z, Y, A) boolean ;  
if Z neobsahuje atribúty A, B také, že A je v  $(Z - AB)^+$  then  
    begin Y:= Z; bcnf= true end  
else begin najdi A a B;  
    Y:= Z - B;  
    while Y obsahuje A a B také, že  $(Y-AB)^+ \rightarrow A$  do  
        Y:= Y - B;  
    bcnf := false;  
end;
```

navrhovanie databáz

15

## Algoritmus normalizácie do BCNF

Vstup: Relačná schéma  $R$  a množina funkčných závislostí  $F$ .

Výstup: Množina relačných schém  $R_1 \dots R_k$  v BCNF.

Metóda: Dekompozícia na dve schémy podľa predošlej lemy jednu  $XA$  zodpovedajúcu závislosti  $X \rightarrow A$ , ktorá je v BCNF a druhú  $R - A$ , na ktorú použijeme algoritmus rekurzívne.

navrhovanie databáz

14

## Normalizácia do 3NF zachovávajúcej závislosti

Vstup: Relačná schéma  $R$  a minimálne pokrytie  $F$ .

Výstup: Relačné schémy bestrátovej dekompozície do 3NF.

Metóda: Ak  $F$  obsahuje závislosť, ktorá obsahuje všetky atribúty  $R$ , potom  $R$  je už v 3NF.  
Inak každej funkčnej závislosti v  $F$  zodpovedá jedna relačná schéma. Treba pridať ešte relačnú schému pre atribúty  $R$ , ktoré sa nevyskytujú v žiadnej funkčnej závislosti  $F$ . Tieto atribúty musia byť súčasťou každého kľúča, aby došlo k spojeniu treba ich doplniť na kľúč  $R$ .

navrhovanie databáz

16

## Pravidlá pre multizávislosti

- (A1)  $x \subseteq y \subseteq U \Rightarrow y \rightarrow x$  reflexívnosť
- (A2)  $\forall z x \rightarrow y \Rightarrow xz \rightarrow yz$  augmentation
- (A3)  $(x \rightarrow y) \wedge (y \rightarrow z) \Rightarrow x \rightarrow z$  tranzitívnosť
- (A4)  $x \twoheadrightarrow y \Rightarrow x \twoheadrightarrow U - x - y$  complementation
- (A5)  $(x \twoheadrightarrow y) \wedge (v \subseteq w) \Rightarrow wx \twoheadrightarrow vy$  augmentation
- (A6)  $(x \twoheadrightarrow y) \wedge (y \twoheadrightarrow z) \Rightarrow x \twoheadrightarrow (z - y)$
- (A7)  $x \rightarrow y \Rightarrow x \twoheadrightarrow y$
- (A8)  $(x \twoheadrightarrow y) \wedge (z \subseteq y) \wedge (w \cap y = \emptyset) \wedge (w \rightarrow z) \Rightarrow x \rightarrow z$

navrhovanie databáz

17

## Poloformálne metódy - mapovanie reality

- Entitno-relačný model
- Binárny model
- NIAM
- Sémantický model
- O - O model
- Sieťový model - Bachmanové diagramy
- Automatické navrhovadlá (designer2000, access, ...)
- HIT

Grafická reprezentácia vizualizácia

navrhovanie databáz

19

## 4NF

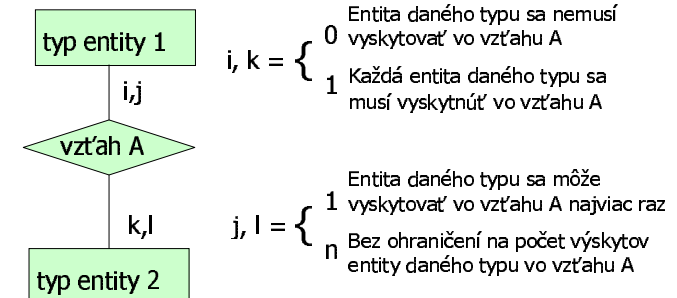
Nech  $D^+$  je množina všetkých platných závislostí a multizávislostí v relačnej schéme  $R$ . Hovoríme, že relačná schéma  $R$  je v 4NF, ak pre každú multizávislosť  $x \twoheadrightarrow y$  takú, že  $R = x \cup y$  a  $x \not\subseteq y$ , platí  $x$  je nadkľúč.

**Veta:** 4NF  $\Rightarrow$  BCNF  $\Rightarrow$  3NF

navrhovanie databáz

18

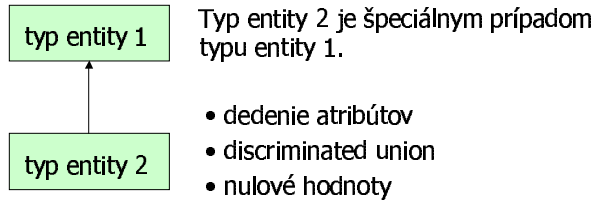
## Entitno relačný (ERA) model



navrhovanie databáz

20

## Vzťah generalizácie - is a



Atribúty - vpisujú sa do typov entít  
označenie kľúčov (a cudzích kľúčov)

navrhovanie databáz

21

## Binárny model - NIAM

- slovný popis
- grafická reprezentácia

Pojmy: • Typ  
• Populácia  
• Výskyt (occurrence)

Lexikálne (LOT) a nelexikálne typy objektov (NOLOT)

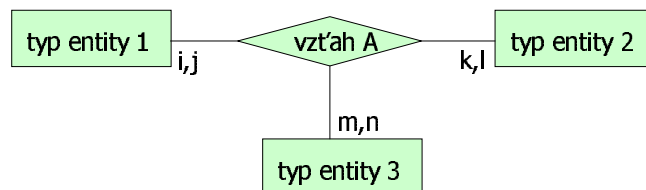
Typy vzťahov:

- *Idea* - medzi nelexikálnymi typmi objektov
- *Bridge* - medzi nelexikálnym a lexikálnym objektom
- *Phrase* - medzi lexikálnymi objektami

navrhovanie databáz

23

## Ternárne a n-árne vzťahy



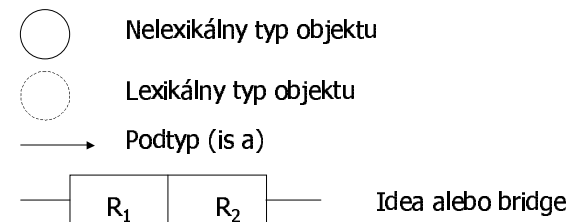
Problém ohraničení počtu výskytov

- objektifikácia binárneho vzťahu
- určenie funkčnej závislosti

navrhovanie databáz

22

## Grafická notácia




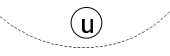

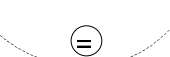
Podmienky - constraints

- ↔ Nad menom role, znamená že táto rola jednoznačne určuje druhú rolu vo vťahu
- ↔ surjekcia (totalita)

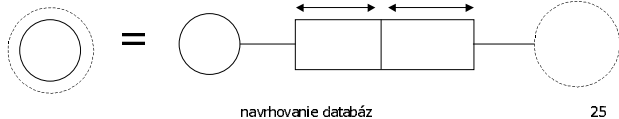
navrhovanie databáz

24

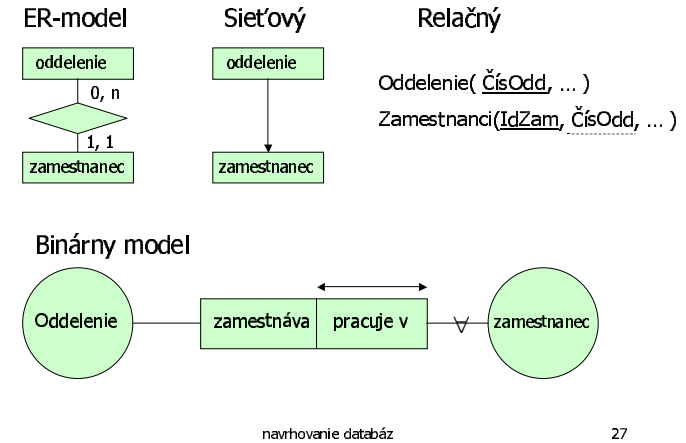
## Podmienky - constraints

-  Disjunktnosť (vylúčenie) medzi podtypmi
-  Jednoznačné určenie výskytu (kombinácie)
-  Inklúzia medzi populáciami rolí
-  Rovnosť populácii rolí

## Makro



## Základné konštrukcie I



## O – O model

interface: Študenti (  
 attribute string meno;  
 attribute integer rodné\_číslo;  
 attribute Struct(deň, mesiac, rok) Dátum\_narodenia;  
 relationship Set(Prednášky) zapísal\_si  
 inverse Prednášky :: majú\_zapísané )

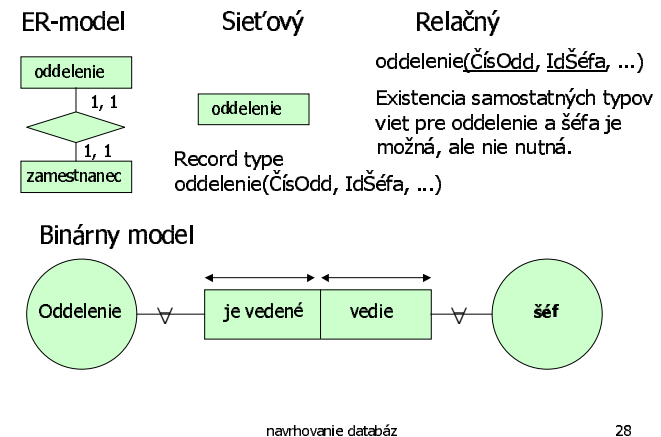
V oblasti návrhu objektový model zodpovedá binárnemu modelu.

Navyše je detailnejší umožňuje podrobne popísať typy atribútov. Používa konštruktory typov (Set - množina, bag - multimnožina, struct - record, list - zoznam, array - pole, ...).

navrhovanie databáz

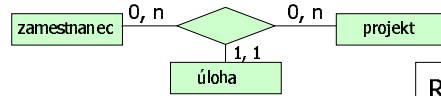
26

## Základné konštrukcie II

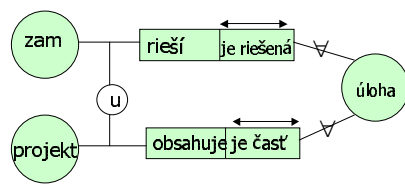


## Ternárne vzťahy I

ER-model



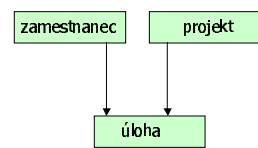
Binárny model



Relačný model

zamestnanci(IdZam, ... )  
 projekty(ČísProj, ... )  
 úlohy(IdZam, ČísProj, ... )

Sieťový model

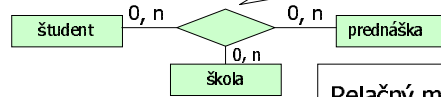


navrhovanie databáz

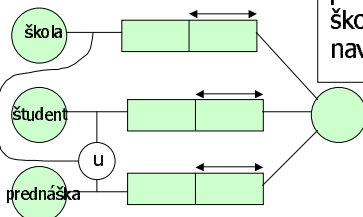
29

## Ternárne vzťahy II

ER-model



Binárny model



Relačný model

študent(RodČís, ... )  
 prednáška(názov, ... )  
 škola(IČO, ... )  
 navštevuje(názov, IČO, RodČís)

navrhovanie databáz

30