

# Návrh databázového modelu

Informační a znalostní systémy



- 1 Návrh modelu
- 2 Konceptuální model
  - Konflikty
- 3 Logický model
  - Funkční závislost
  - Dekompozice
  - Vazby
  - Datové typy

# Návrh modelu

- návrh musí
  - pokrývat požadavky zadavatele
- návrh by měl
  - reflektovat i možné budoucí poslán
- návrh
  - od shora dolů
  - zdola nahoru

# Vývoj modelu

## 1 konceptuální model

- výběr konceptů
- volba rozlišení (jaké detaily)
- nástin vazeb

## 2 logický model

- indentifikace vlastností - převod koncept entita
- identifikace vazeb - relace s kardinalitami
- dekompozice - podmínka splnění požadované normální formy

## 3 fyzický model

- způsob fyzického uložení dat
- v případě databází není součástí návrhu

# Konceptuální model

- definuje koncepty a vazby mezi nimi

# Konceptuální model

- definuje koncepty a vazby mezi nimi

## Úloha:

Navrhňte model pro uchovávání naměřených dat ze senzorové sítě. Každý senzor je umístěn v jedné místnosti a může měřit i více veličin. U místnosti definujeme budovu a u budovy adresu.

# Konceptuální model

- definuje koncepty a vazby mezi nimi

## Úloha:

Navrhnete model pro uchování naměřených dat ze sensorové sítě. Každý sensor je umístěn v jedné místnosti a může měřit i více veličin. U místnosti definujeme budovu a u budovy adresu.

- Nejjednodušší přístup - jediný všezahrnující koncept



Budova  
Místnost  
Sensor  
Velicina  
Čas

# Konceptuální model

- definuje koncepty a vazby mezi nimi

## Úloha:

Navrhněte model pro uchovávání naměřených dat ze senzorové sítě. Každý senzor je umístěn v jedné místnosti a může měřit i více veličin. U místnosti definujeme budovu a u budovy adresu.

- Nejjednodušší přístup - jediný všezahrnující koncept



Budova  
Místnost  
Senzor  
Velicina  
Čas

čas	budova	adresa	místnost	senzor	veličina	hodnota
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	teplota	20
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	tlak	1020
2010-01-01 15:00	B	Technická 3, Praha 6	100	S2	teplota	28



# Konflikty

čas	budova	adresa	místnost	senzor	veličina	hodnota
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	teplota	20
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	tlak	1020
2010-01-01 15:00	B	Technická 3, Praha 6	100	S2	teplota	28

- Pokud
  - úprava adresy u prvního záznamu, vznikne nekonzistence  
(není patrné, která adresa platí)

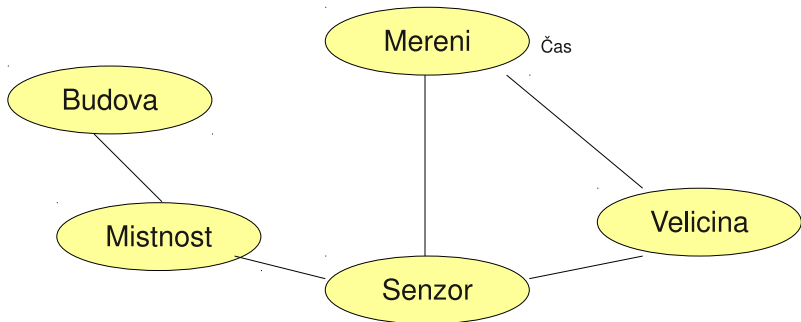
# Konflikty

čas	budova	adresa	místnost	senzor	veličina	hodnota
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	teplota	20
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	tlak	1020
2010-01-01 15:00	B	Technická 3, Praha 6	100	S2	teplota	28

- Pokud
  - úprava adresy u prvního záznamu, vznikne nekonzistence  
(není patrné, která adresa platí)
  - smazáním třetího záznamu smažeme i adresu budovy  
(což zcela jistě nechceme)
- Z toho důvodu
  - snaha rozložit model na dílčí atomické koncepty a identifikovat mezi koncepty vazby

## Dekompozice v konceptuálním modelu

- dekompozice - dílčí atomické koncepty
  - umožňují lepší náhled
  - umožňují lépe vyjádřit důležité detaily
  - zjednodušují práci s logickým modelem



# Logický model

- úkolem logického modelu je najít
  - souvislosti mezi atributy (vlastnostmi) a koncepty
    - ke kterému konceptu přesně vlastnost patří
  - souvislosti mezi koncepty - identifikovat typy vazeb

# Logický model

- úkolem logického modelu je najít
  - souvislosti mezi atributy (vlastnostmi) a koncepty
    - ke kterému konceptu přesně vlastnost patří
  - souvislosti mezi koncepty - identifikovat typy vazeb
- v případě potřeby
  - je možné se vrátit na konceptuální úroveň a model modifikovat

# Logický model

- úkolem logického modelu je najít
  - souvislosti mezi atributy (vlastnostmi) a koncepty
    - ke kterému konceptu přesně vlastnost patří
  - souvislosti mezi koncepty - identifikovat typy vazeb
- v případě potřeby
  - je možné se vrátit na konceptuální úroveň a model modifikovat
- entita: koncept rozšířený o výčet atributů

## Funkční závislost

- na základě hodnot atributů na levé straně jsem schopen jednoznačně odvodit hodnoty atributů na pravé straně
- jsou součástí modelu - tj jsou definovány návrhářem

## Funkční závislost

- na základě hodnot atributů na levé straně jsem schopen jednoznačně odvodit hodnoty atributů na pravé straně
- jsou součástí modelu - tj jsou definovány návrhářem
- příklad
  - definuji, že na základě označení budovy jsem schopen zjistit adresu budovy, tj.

budova → adresa



## Funkční závislost

- na základě hodnot atributů na levé straně jsem schopen jednoznačně odvodit hodnoty atributů na pravé straně
- jsou součástí modelu - tj jsou definovány návrhářem
- příklad
  - definuji, že na základě označení budovy jsem schopen zjistit adresu budovy, tj.

budova  $\rightarrow$  adresa

- definuji, že na základě času, senzoru a veličiny jsem schopen zjistit naměřenou hodnotu, tj.

{cas, senzor, velicina}  $\rightarrow$  hodnota

## Funkční závislost

- na základě hodnot atributů na levé straně jsem schopen jednoznačně odvodit hodnoty atributů na pravé straně
- jsou součástí modelu - tj jsou definovány návrhářem
- příklad
  - definuji, že na základě označení budovy jsem schopen zjistit adresu budovy, tj.

budova → adresa

- definuji, že na základě času, senzoru a veličiny jsem schopen zjistit naměřenou hodnotu, tj.

{ čas, senzor, velicina } → hodnota

- pozor, funkční závislost obecně nelze odvozovat z dat

čas	budova	adresa	místnost	senzor	veličina	hodnota
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	teplota	20
2010-01-01 15:00	A	Technická 2, Praha 6	200	S1	tlak	1020
2010-01-01 15:00	B	Technická 3, Praha 6	100	S2	teplota	28

- např. z toho, že byl měřen tlak, nevyplývá, že měření proběhlo 2010-01-01 15:00

# Dekompozice

- pokud jsme nedekomponovali na úrovni konceptů, můžeme dekomponovat na logické úrovni
  - Pokud v entitě je funkční závislost  $A_1 \rightarrow A_2$  a zároveň  $A_2 \rightarrow A_3$ , provedeme dekompozici
    - vytvoříme novou entitu obsahující atributy  $A_2 \cup A_3$
    - z původní entity odstraníme atributy  $A_3$
  - hledáme maximálně početné  $A_3$  a minimálně početné  $A_2$
  - Příklad:

čas, budova, adresa, místnost, senzor, veličina, hodnota

# Dekompozice

- pokud jsme nedekomponovali na úrovni konceptů, můžeme dekomponovat na logické úrovni
  - Pokud v entitě je funkční závislost  $A_1 \rightarrow A_2$  a zároveň  $A_2 \rightarrow A_3$ , provedeme dekompozici
    - vytvoříme novou entitu obsahující atributy  $A_2 \cup A_3$
    - z původní entity odstraníme atributy  $A_3$
  - hledáme maximálně početné  $A_3$  a minimálně početné  $A_2$
  - Příklad:

čas, budova, adresa, místnost, senzor, veličina, hodnota  
čas, budova, místnost, senzor, veličina, hodnota

↔

budova, adresa

# Dekompozice

- pokud jsme nedekomponovali na úrovni konceptů, můžeme dekomponovat na logické úrovni
  - Pokud v entitě je funkční závislost  $A_1 \rightarrow A_2$  a zároveň  $A_2 \rightarrow A_3$ , provedeme dekompozici
    - vytvoříme novou entitu obsahující atributy  $A_2 \cup A_3$
    - z původní entity odstraníme atributy  $A_3$
  - hledáme maximálně početné  $A_3$  a minimálně početné  $A_2$
  - Příklad:

čas, budova, adresa, místnost, senzor, veličina, hodnota

čas, budova, místnost, senzor, veličina, hodnota

čas, místnost, senzor, veličina, hodnota

↔

budova, adresa

↔

místnost, budova

# Dekompozice

- pokud jsme nedekomponovali na úrovni konceptů, můžeme dekomponovat na logické úrovni
  - Pokud v entitě je funkční závislost  $A_1 \rightarrow A_2$  a zároveň  $A_2 \rightarrow A_3$ , provedeme dekompozici
    - vytvoříme novou entitu obsahující atributy  $A_2 \cup A_3$
    - z původní entity odstraníme atributy  $A_3$
  - hledáme maximálně početné  $A_3$  a minimálně početné  $A_2$
  - Příklad:

čas,budova,adresa,místnost,senzor,veličina,hodnota

čas,budova,místnost,senzor,veličina,hodnota

čas,místnost,senzor,veličina,hodnota

čas,senzor,veličina,hodnota

čas,senzor,veličina,hodnota

↔

budova, adresa

↔

místnost, budova

↔

senzor, místnost

↔

senzor, veličina

# Dekompozice

- pokud jsme nedekomponovali na úrovni konceptů, můžeme dekomponovat na logické úrovni
  - Pokud v entitě je funkční závislost  $A_1 \rightarrow A_2$  a zároveň  $A_2 \rightarrow A_3$ , provedeme dekompozici
    - vytvoříme novou entitu obsahující atributy  $A_2 \cup A_3$
    - z původní entity odstraníme atributy  $A_3$
  - hledáme maximálně početné  $A_3$  a minimálně početné  $A_2$
  - Příklad:

čas, budova, adresa, místnost, senzor, veličina, hodnota

čas, budova, místnost, senzor, veličina, hodnota

čas, místnost, senzor, veličina, hodnota

čas, senzor, veličina, hodnota

čas, senzor, veličina, hodnota

↔

budova, adresa

↔

místnost, budova

↔

senzor, místnost

↔

senzor, veličina

- takto dekomponovaný model splňuje požadavek 3NF

## Vazby mezi entitami

- při návrhu potřeba identifikovat druhy vazeb



## Vazby mezi entitami

- při návrhu potřeba identifikovat druhy vazeb
- maximální kardinalita
  - 1:1  
(jedné entitě na levé straně vazby odpovídá jedna entita na pravé straně)
  - 1:N  
(jedné entitě na levé straně vazby odpovídá několik entit na pravé straně)
    - budova má více místností, místnost patří do jedné budovy
  - N:M  
(několika entitám na levé straně vazby odpovídá několik entit na pravé straně)
    - jeden senzor může měřit více veličin, jednu veličinu může měřit více senzorů

## Vazby mezi entitami

- při návrhu potřeba identifikovat druhy vazeb
- maximální kardinalita
  - 1:1  
(jedné entitě na levé straně vazby odpovídá jedna entita na pravé straně)
  - 1:N  
(jedné entitě na levé straně vazby odpovídá několik entit na pravé straně)
    - budova má více místností, místnost patří do jedné budovy
  - N:M  
(několika entitám na levé straně vazby odpovídá několik entit na pravé straně)
    - jeden senzor může měřit více veličin, jednu veličinu může měřit více senzorů
- minimální kardinalita
  - 1 .. vztahu se účastní jedna entita  
povinně vyplnitelný údaj
  - 0 .. vztahu se nemusí účastnit žádná entita  
nepovinně vyplnitelný údaj

# Vazby mezi entitami

(příklad tabule)

## Dekompozice N:M vazby

- N:M vazbu lze vyjádřit pomocí tzv. vazební tabulky
  - obsahuje dvě 1:N vazby
- ER Model
  - Model obsahující pouze entity a 1:N vazby (relationships)

## Primární a cizí klíč

- Primární klíč: minimální množina atributů z entity,
  - která jednoznačně identifikuje záznam
  - ze které lze jednoznačně odvodit zbylé atributy z entity
- Cizí klíč: množina atributů v nějaké entitě, které odpovídá primární klíč v odkazované entitě
- příklad:
  - u dekompozice ( $A_1 \rightarrow A_2 \rightarrow A_3$ ):  
 $A_2$  v nové entitě je primárním klíčem  
tj.  $A_3$  v nové entitě je z  $A_2$  odvoditelné  
(nebo funkční závislost  $A_2 \rightarrow A_3$ )  
 $A_2$  ponechané v původní entitě - cizí klíč do nové entity
- Primární a cizí klíč jsou svázány referenční integritou
  - cizímu klíči zajišťuje existenci odpovídající hodnoty primárního klíče

# Datové typy

- číselné
  - integer, float, numeric
- textové
  - char ( $n$ ) .. text o pevné délce  $n$  znaků
  - varchar( $n$ ) .. text o proměnné délce do  $n$  znaků,  $n < 256$
  - text
- ostatní
  - timestamp .. čas
  - další, viz dokumentace

## Datový typ klíče

- datové typy, porovnání jejichž hodnot je výpočetně složité, jsou nahrazeny datovými typy s jednoduchým porovnáním
- typický příklad: text  $\rightsquigarrow$  integer
- POZOR:
  - nutno zachovat unikátnost původního klíče  
klíčové slovo UNIQUE
- příklad
  - entita budova obsahuje primární klíč udávající označení budovy
  - tento klíč je nahrazen "virtuálním" klíčem id\_budova typu integer
  - k původnímu primárnímu klíči je vytvořen unikátní index