PRÉCIS

STRUKTUROVANÁ DATABÁZE JAKO ODPOVĚĎ NA NESTRUKTUROVANÝ DOTAZ

Dominik Fišer, Jiří Schejbal http://www.doser.cz

Obsah – část 1

- přednáší Dominik Fišer

- Co je to Précis?
- Datový model – relační schéma
- Cesty a jejich váhy v grafu
- Dotazovací model
- Výsledky experimentu

Obsah – část 2

- přednáší Jiří Schejbal

- Architektura systému
- Tvorba výsledného schématu
- Tvorba databáze odpovídající dotazu
- Překlad výsledku dotazu do přirozeného jazyka
- Algoritmy
- Použité zdroje

Co to je Précis?

nový DBMS?

rozšíření do Oraclu?

…

ani jedno!

Slovo précis

- **précis /'preisi:/**

  a shortened form of a piece of writing or of what someone has said, giving only the main points (Longman Dictionary)
- *česky souhrn, abstrakt*

Motivační příklad

- zeptáme se na heslo „Woody Allen“
- jedna z možných odpovědí

  „*Woody Allen was born on December 1, 1935 in Brooklyn, New York, USA. As a director, Woody Allen’s work includes Match Point (2005), Melinda and Melinda (2004), Anything Else (2003). As an actor, Woody Allen’s work includes Hollywood Ending (2002), The Curse of the Jade Scorpion (2001).*“

Základní charakteristika Précis

- nestrukturované dotazování nad strukturovanými daty
- nestrukturovaný dotazovací jazyk
  - uživatel nemusí znát žádný dotazovací jazyk
  - dotazování pomocí klíčových slov
- odpověď obsahuje i všechny související informace s klíčovým slovem v uživatelsky přívětivé podobě
  - uživatel začíná hledání od nějakého klíčového slova a podle nalezených informací pokračuje ve vyhledávání dál

Précis – vlastnosti (1)

- dotazování probíhá nad běžnou relační databází
- dotaz se může skládat z více klíčových slov
- podpora logických spojek AND, OR a NOT a frází
- odpovědí na dotaz je multi-relační databáze
  - nikoli pouze řádek z databáze
  - dynamicky generované schéma, omezení, klíče a data, logická podmnožina původní databáze
  - omezení schématu

Précis – vlastnosti (2)

- pouze výběrové klauzule
- odpovědi jsou šité na míru uživateli na základě jeho preferencí
  - uživatelské profily
  - při pokládání dotazu
  - statistika na základě logu dotazů
- odpovědi zahrnují i související informace k hledanému klíčovému slovu
- odpovědi jsou v přirozeném jazyce

K čemu to může být dobré

- dotazy – uživatelsky přívětivé vyhledávání
- on-line přístupné databáze např. knihovny, muzea
  - různá vyhledávací rozhraní a možnosti
  - často složitá
  - než získáme to co chceme, stojí to čas
  - musíme vědět co hledáme
- odpovědi – výřezy z velkých databází, např. ukázka nové funkcionality na části databáze

Datový model

databázové schéma a jeho graf

Datový model – značení

- databázové schéma $\mathbf{D} = \{\mathbf{R_i}:1 \leq i \leq m\}$
  - množina relačních schémat $\mathbf{R_i}$
- relační schéma $\mathbf{R_i}(A_{1i}, A_{2i}, \ldots, A_{ki})$
  - jméno relace + množina atributů $A_i = \{A_{ji}:1 \leq j \leq k_i\}$
- relační schéma $\mathbf{R_i}$ ~ relace R_i
- databázové schéma $\mathbf{D}$ ~ databáze D

Graf databázového schématu

- značení **G**(**V**,**E**)
- ohodnocený orientovaný graf
- odpovídá databázovému schématu **D**
- vrcholy ve **V**
  - relace – pro každou relaci R ve schématu
  - atribut – pro každý atribut A každé relace R ve schématu
- hrany v **E**
  - projekce – hrana relace-atribut
  - spojení – hrana relace-relace

Ohodnocení hran

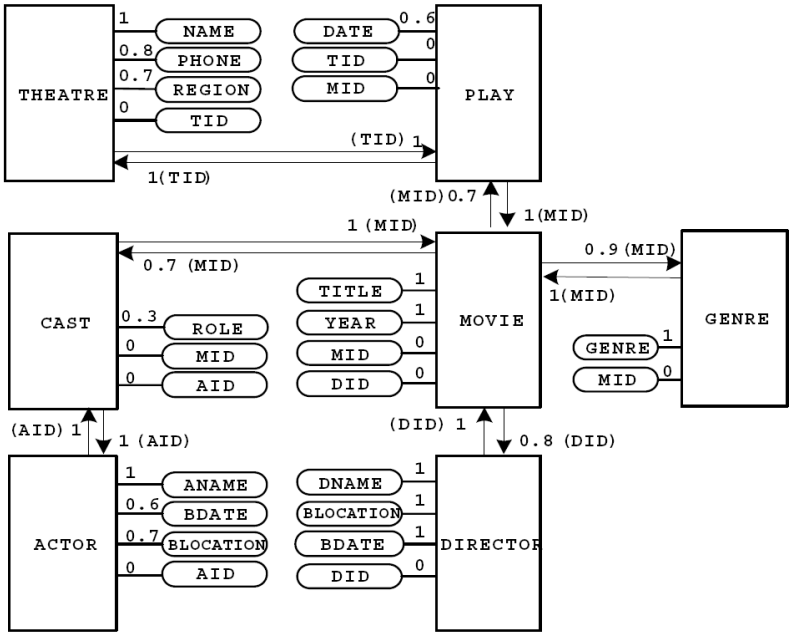
- váha hrany $w \in [0,1]$
- významnost vazby
  - 1 … silná vazba
    - pokud se jeden z vrcholů objeví v odpovědi, měla by se v ní promítnou i hrana
  - 0 … žádná vazba
    - pokud se jeden z vrcholů objeví v odpovědi, pro další vrcholy to nic neznamená
- mezi dvěma vrcholy může vést každým směrem hrana s různou váhou

Orientace hran a její interpretace

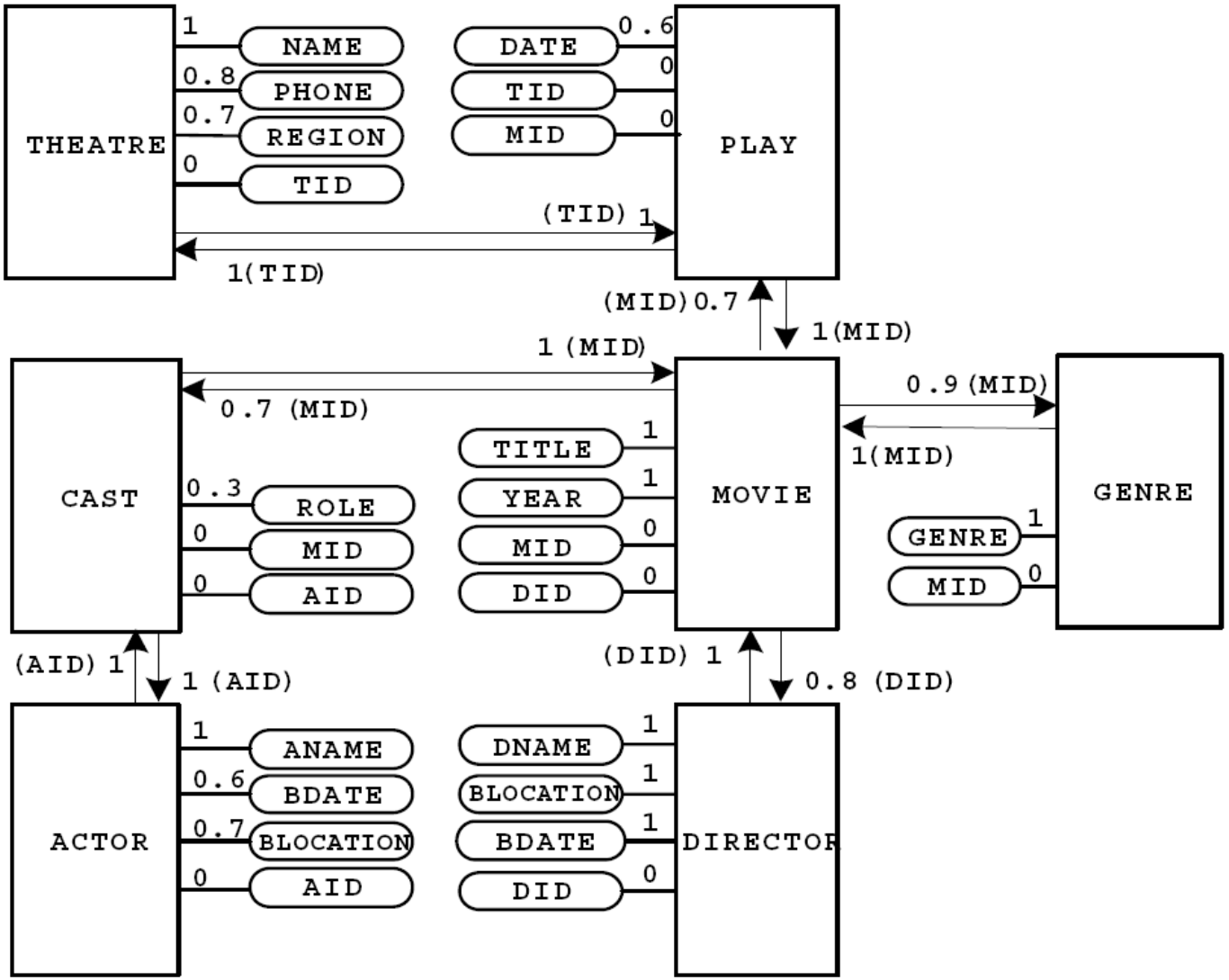
- orientovaná hrana určuje závislost levé části spojení na pravé
- levá část označuje relaci již zahrnutou v odpovědi, pravá část relaci, která by měla být v odpovědi zohledněna
- orientované hrany nejsou násobné

Příklad – databázové schéma

- *THEATRE(<u>tid</u>,name,phone,region)*
- *PLAY(<u>tid</u>,<u>mid</u>,<u>date</u>), GENRE(<u>mid</u>,<u>genre</u>)*
- *ACTOR(<u>aid</u>,aname,blocation,bdate)*
- *DIRECTOR(<u>did</u>,dname,blocation,bdate)*
- *MOVIE(<u>mid</u>,title,year,did)*
- *CAST(<u>mid</u>,aid,role)*

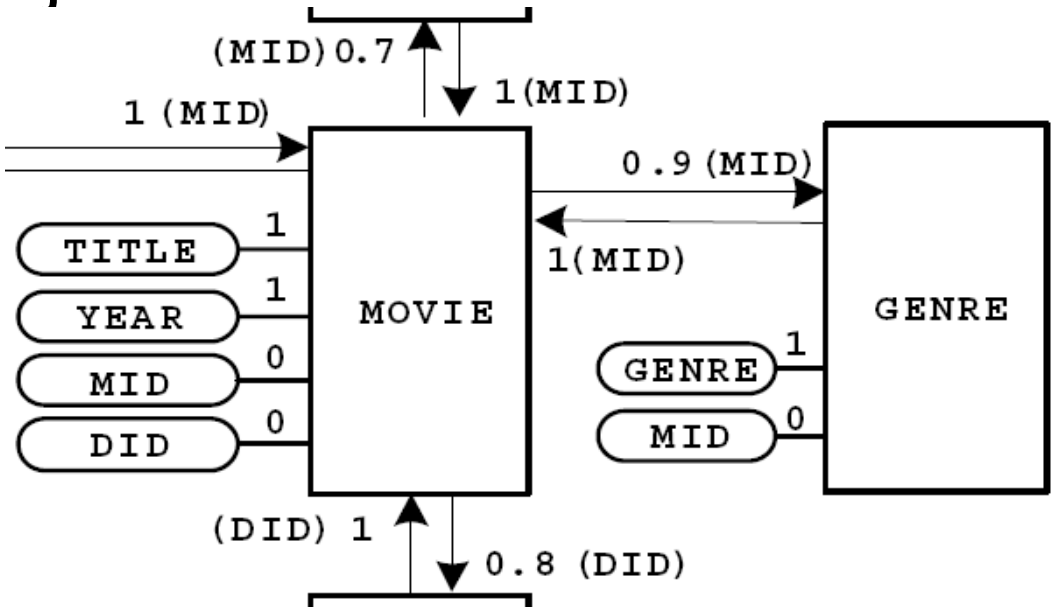


Příklad – databázový graf



Příklad – relace *MOVIE-GENRE*

- žánr závisí na filmu víc
- odpověď na žánr bude vždy obsahovat informace o souvisejících filmech
- odpověď na film nemusí nutně obsahovat informace o souvisejícím žánru
- $w_{MOVIE->GENRE} = 0{,}9$
- $w_{GENRE<-MOVIE} = 1$



Ohodnocení hran – důsledky

- váhy může určovat
  - uživatel – při pokládání dotazu, ve svém profilu
  - designer – předdefinovaná nastavení pro různé skupiny uživatelů
    - návštěvník kina, filmový kritik, dramaturg
- použitím různých vah na hranách získáme odlišné odpovědi na stejný dotaz
- interaktivní prohlížení obsahu databáze
  - pokládáme stejný dotaz s různými váhami podle toho co nás zaujme zrovna nejvíce
- uživatelské preference – odpovědi na míru

Cesty a jejich váhy v grafu

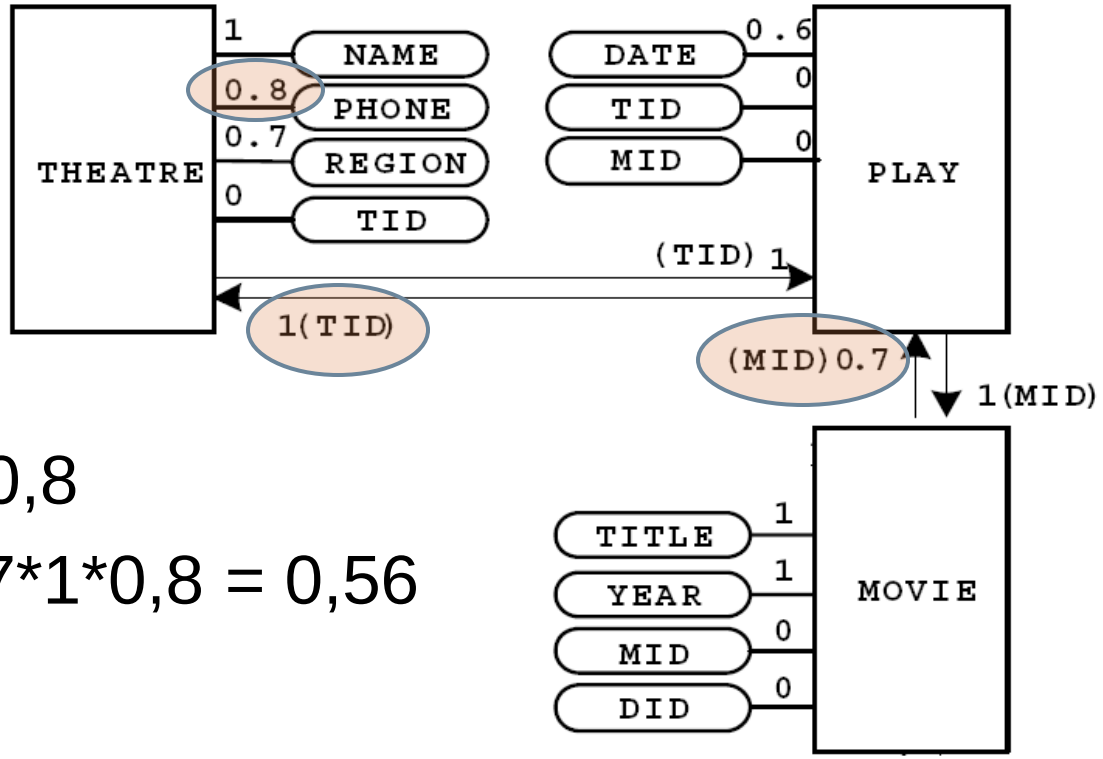
cesta jako vazba

Orientované cesty

- orientovaná cesta *p*
- dva typy cest
  - tranzitivní spojení
    - mezi dvěma vrcholy typu relace
    - hrany typu spojení
    - reprezentuje implicitní spojení mezi relacemi
  - tranzitivní projekce
    - mezi vrcholem typu relace a typu atribut
    - hrany typu spojení i projekce
    - reprezentuje projekci atributu na relaci

Váha cesty w_p

- funkce vah hran, $w_p \leq \min w_e$
  - měla by klesat s rostoucí délkou cesty
- v této implementaci použita funkce násobení



- příklad
  - $w_{PHONE,THEATRE} = 0,8$
  - $w_{MOVIE,PHONE} = 0,7*1*0,8 = 0,56$

Dotazovací model

vznik databázového schématu odpovědi

Dotazovací model – značení

- databáze D
- précis query $Q = \{k_1, k_2, \ldots, k_m\}$
- précis - nová databáze D'
  - odpověď na dotaz Q nad D

Pravidla pro novou databázi *D´*

- množina jmen relací v **D´** je podmnožina jmen v **D**
- pro každou relaci $\mathbf{R_i}$ v **D´** je její množina atributů $\mathbf{B_i} = \{B_{ji}:1 \leq j \leq l_i\}$ v D´ podmnožinou $\mathbf{A_i} = \{A_{ji}:1 \leq j \leq k_i\}$ v **D**
- pro každou relaci $\mathbf{R_i}$ v **D´** je množina záznamů odpovídajících relaci R_i´ podmnožinou záznamů v původní relaci R_i
- databáze *D´* je generovaná spojovacími dotazy (cizí klíče) od relace, ve které se hledané klíčové slovo vyskytlo, a tranzitivním rozšířením nad celé databázové schéma **D**
- výsledná množina relací, atributů a záznamů v *D´* je zúžená omezeními

Omezení na databázi

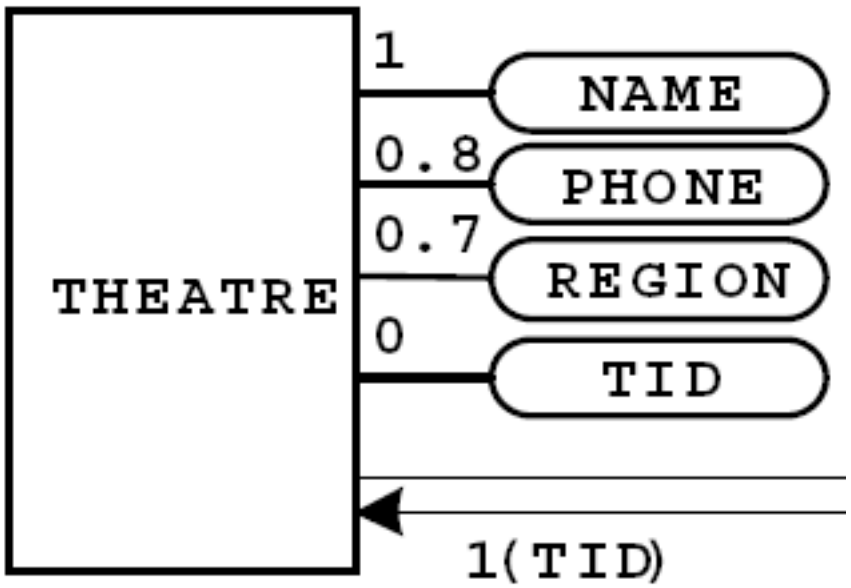
- omezení stupně *d*
  - atributy, relace v *D´*
    - maximální počet atributů v *D´*
    - minimální váha cesty typu projekce v grafu **G**
- omezení kardinality *c*
  - počet záznamů v *D´*
    - maximální počet záznamů v *D´*
    - maximální počet záznamů v relaci v *D´*

Omezení stupně – příklad

- film může mít více režisérů
  - přidáme novou relaci *DIRECTED_BY(mid,did)*
- cesta mezi *MOVIE* a *DIRECTOR* se prodlouží stejně jako počet relací v *D´* potřebných při dotazu na režiséra

Omezení kardinality – příklad

- *THEATRE*
  - různé váhy jednotlivých atributů
  - použitím vhodných kritérií můžeme dosáhnout
    - odpověď obsahuje pouze *NAME*
    - odpověď obsahuje i *PHONE* a *REGION*



Omezení – důsledky

- použitím různých omezení získáme odlišné odpovědi na stejný dotaz se stejnými váhami
- podobně jako váhy může omezení určovat
  - uživatel – při pokládání dotazu, ve svém profilu
  - designer – předdefinovaná nastavení pro různé skupiny uživatelů

Výsledky experimentu

není to jen teorie, opravdu to funguje

Výsledky experimentu

- prototyp v C++ na Oraclu 9i R2
- data z IMDB, informace o 340 tis. filmů
- generování schématu databáze odpovědi
  - zanedbatelná doba
- generování databáze odpovědi
  - časově nejnáročnější

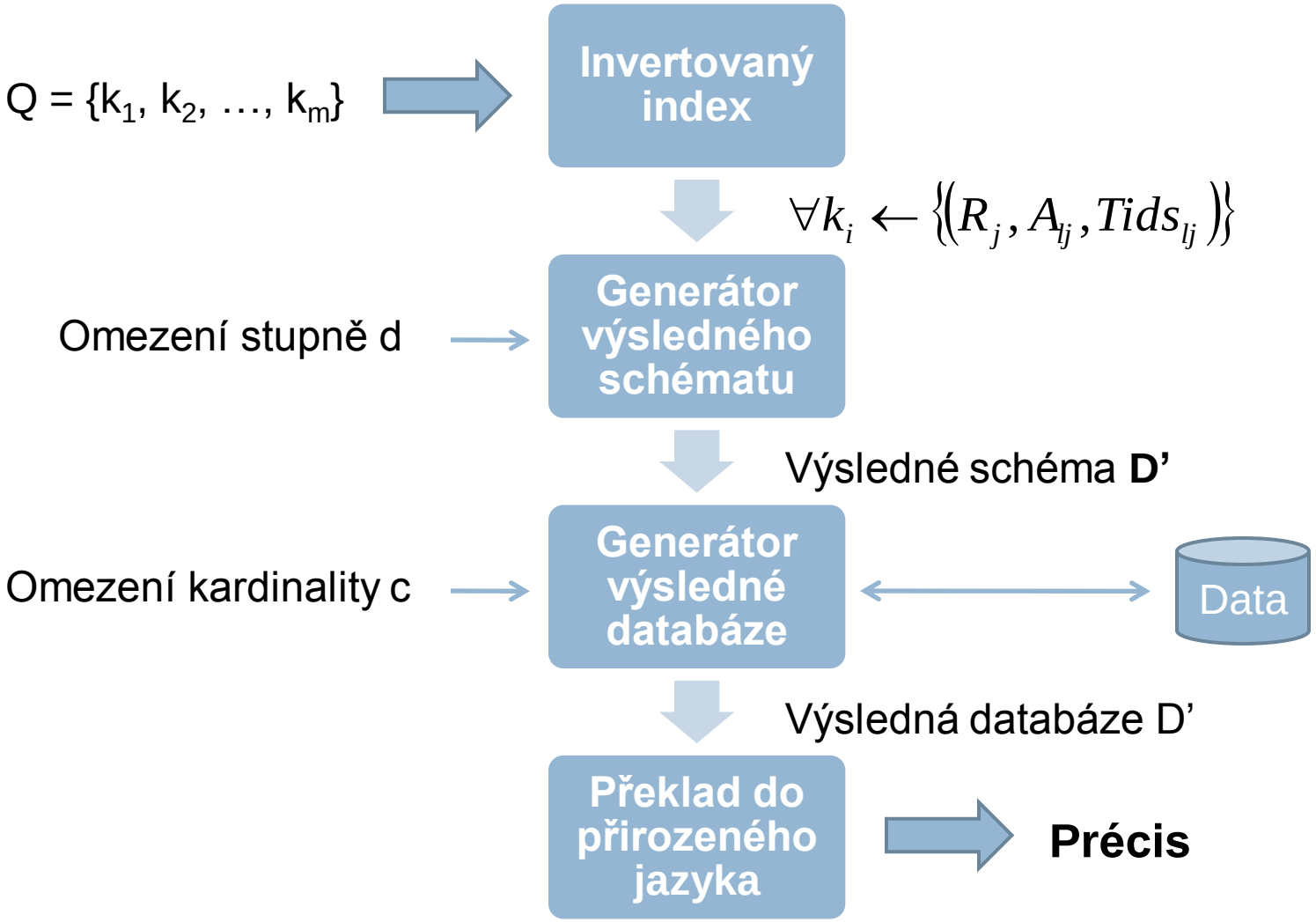
Část 2

přednáší Jiří Schejbal

Architektura systému

od množiny zadaných tokenů ke srozumitelnému výsledku

Architektura systému



Invertovaný index

- Token ~ seznam výskytů v DB
- Výskyt – dvojice atribut,relace – (R_j, A_{lj})
- ke každé takové dvojici seznam záznamů z R_j, ve které atribut A_{lj} obsahuje token ($Tids_{lj}$)
- Token se může vyskytovat ve více relacích, v různých atributech

$$k_i \rightarrow \left\{\left(R_j, A_{lj}, Tids_{lj}\right)\right\}, \forall k_i \in Q$$

Generátor výsledného schématu

- Najde části původního databázového schématu, které jsou relevantní pro daný dotaz
- Výstup: **D’**
  - Podmnožina původního schématu
  - Relace spojené tranzitivně
  - Splňující omezení stupně d

Generátor výsledné databáze

- Databáze D’ odpovídající schématu **D’**
- Začíná v relacích s tokeny, postupně se tranzitivně rozšiřuje přes spojení (přes cizí klíče)
- Splňuje podmínku kardinality c

Překlad do přirozeného jazyka

- Volitelný krok
- Reprezentace výsledné databáze v přirozeném jazyce
- Využívá šablony, definované:
  - Návrhářem DB
  - Administrátorem
  - Uživatelem

Příklad

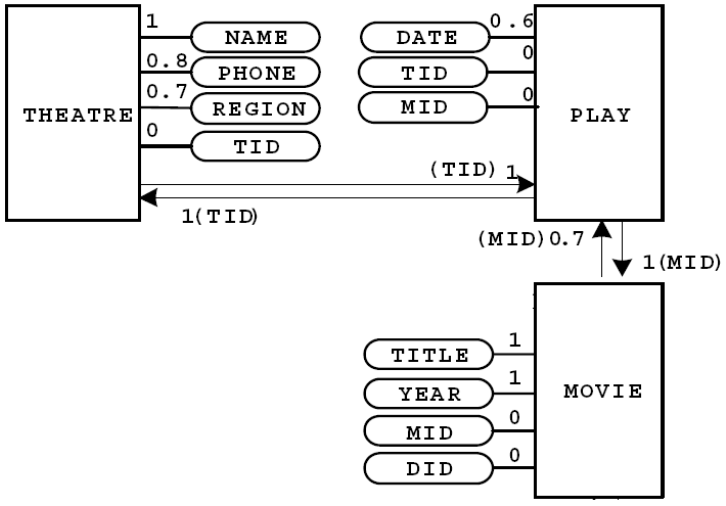
- zeptáme se na heslo „Woody Allen“
- Q = {‘Woody Allen’}
- Omezení stupně d
  - Projekce s vahou ≥ 0,9
- Omezení kardinality c
  - Max. 3 záznamy v relaci

Generátor výsledného schématu

omezení původního schématu na relevantní části

Generátor výsledného schématu

- Relevantní části schématu pro daný dotaz
- Vstupy:
  - Graf **G(V, E)** pro původní DB schéma **D**
  - Výchozí relace (relace s hledanými tokeny)
  - Množina všech (tranzitivních) acyklických projekčních cest $\mathbf{P_n}$ z výchozích relací
    - $\mathbf{P_n} = \{p_i | i \in [1,n]; w_{i-1} \geq w_i\}$
    - (seřazeny sestupně podle vah)
  - Omezení stupně d



Generované schéma

- Podgraf **G**
- Cesty P_d jsou podmnožinou P_n
  - d = max({t|t [1,n]: p_t splňuje d})

- p_t splňuje d (druhy podmínek):
  - $t \leq r$ max. r projekcí s nejvyšší vahou
  - $w_t \geq w_o$ projekce s vahou větší než w_o
  - Délka cesty $p_t \leq l_o$

Algoritmus – generátor schématu (1)

Vstup: graf G (původního schématu), omezení stupně d, množina $\{R_j | R_j$ obsahuje token$\}$

1. $QP = \{\}$, $P_d = \{\}$, $G' = \{\}$
2. **Foreach** e hrana spojená s R_j: $e(R_j, x) \in E$, $x \in V$,
   $QP \leftarrow e$, příslušná úprava uzlů, hran a vstupních stupňů v G'

   End For
3. **While** (QP neprázdný)
   1. p ← první z QP
   2. **If** $((Pd \cup \{p\})$ nesplňuje podmínku d) **Then** exit **End If**
   3. **If** (p je projekční cesta) **Then** $P_d \leftarrow$ p **End If**
   4. **…**

Algoritmus – generátor schématu (2)

3. **While** (QP neprázdný)

   …

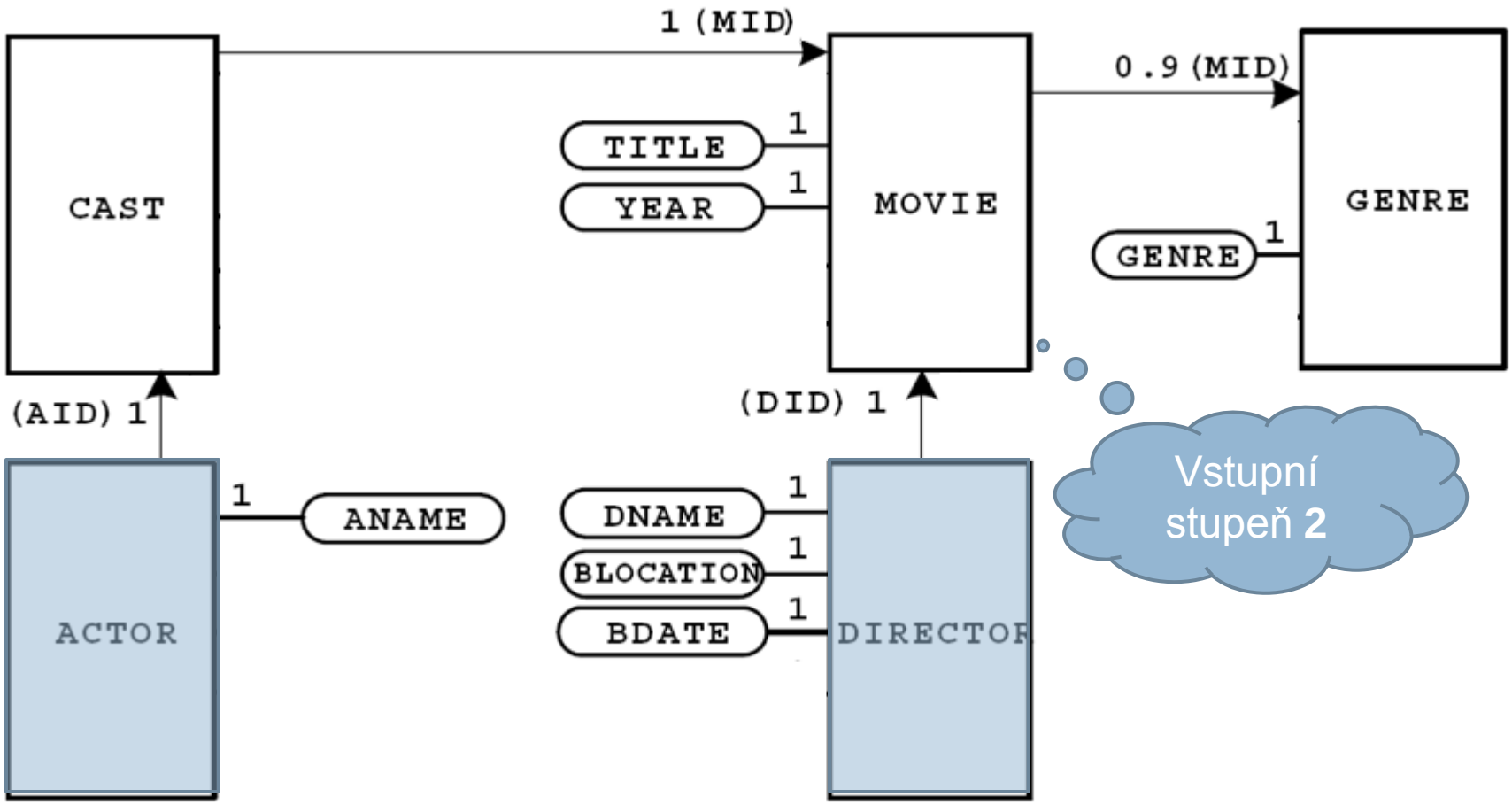
   4. **If** (p je spojovací cesta) **Then**
      Foreach e hrana z G spojená s p
         p’ = spojení p a e, p’ je acyklická
         If ((Pd ∪ p’) nesplňuje d) **Then** Exit For
      End If
         QP ← p’
      End For
   End If

Výstup: **G’**

Popis algoritmu

- QP – fronta probíraných cest
- Preference cest
  - Vyšší váha
  - Kratší cesta
- Pokud projekční cesta splňuje podmínku d, je přidána do G’
- Vstupní stupně u vrcholů (počet relací, ze kterých vede do daného vrcholu projekční cesta)

Příklad (G')


1 (MID)
0 .9 (MID)
CAST
TITLE
YEAR
MOVIE
GENRE
GENRE
(AID) 1
(DID) 1
ACTOR
ANAME
DNAME
BLOCATION
BDATE
DIRECTOR
Vstupní stupeň 2

Generátor výsledné databáze

cesta k relevantním datům

Generátor výsledné databáze

- Databáze D’ odpovídající schématu **D’** (jeho odpovídající graf **G’**)
- Výběr záznamů „zbylých“ relací po generování schématu
- Další zmenšení množiny záznamů podle omezení
  kardinality c

Rozšiřování databáze

- D_0 = vstupní relace se záznamy obsahující tokeny
- Možné výsledné databáze
  - $D_1 \leftarrow D_0 * R_1$
  - $D_2 \leftarrow D_1 * R_2$
  - …
  - $D_n \leftarrow D_{n-1} * R_n$
- R_i je připojeno pokud existuje spojovací hrana z R_i do D_{i-1}
- Preference hran – vyšší váha

Omezení kardinality

- Značení
  - $card(D_i)/card(R_i)$ – počty záznamů v DB/relacích
- D_c = výsledná DB (D')
- $c = max(\{t|t \in [0,n]: D_t$ splňuje **c**$\})$
- D_t splňuje **c** (druhy podmínek):
  - $card(D_t) \leq c_o$ max. počet záznamů v DB
  - $card(R_t) \leq c_o$ max. počet záznamů v relacích

Algoritmus – generátor databáze (1)

Vstup: {R_j|R_j obsahuje token}, {$Tids_j$|$Tids_j$ je množina ident. v R_j}, **G'**, omezení kardinality c

Výstup: D'

1. D' ← {NaiveQ(
   $\sigma_{Tidsj}(R_j)[\pi(R_j)]$,
   $c(\sigma_{Tidsj}(R_j)[\pi(R_j)])$
   ), pro každé R_j}

Algoritmus – generátor databáze (2)

2. **Foreach** e hrana z G’ se vstupním stupněm = 1
   1. **If** (vazba přes e je ? : N) **Then**
      D’ ← RoundRobin(D’ * R_i, c(D’ * R_i))
      Else
      D’ ← NaiveQ(D’ * R_i, c(D’ * R_i))
      End If
   2. Zmenši vstupní stupeň R_i

   End For

NaiveQ

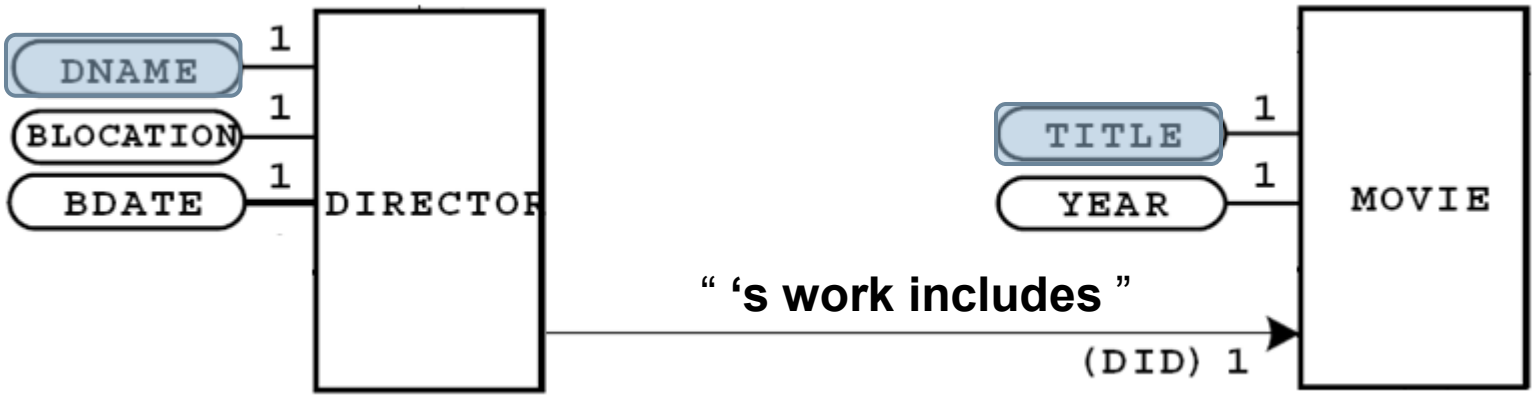
- NaiveQ(dotaz, n) – dosažení omezení kardinality
- Vybere n záznamů z výsledků dotazu
  - Př. Oracle – pseudo-sloupec RowNum
- Spojení R_i * R_j
  - Vazba ? : 1 – náhodných n záznamů z R_j
  - Vazba ? : N – při rovnoměrném rozložení lze
    - Lepší **RoundRobin**

RoundRobin

- RoundRobin(dotaz, n)
- Výběr n záznamů k dosažení omezení kardinality
- Při spojení R_i * R_j
- Pro každý záznam z R_i' se spustí hledání připojených záznamů z R_j, do D' se od každého výsledku přidávají záznamy po jednom

Příklad

Match Point	2005
Melinda and Melinda	2004
Anything Else	2003



Woody Allen	1 December, 1935	Brooklyn, New York, USA

Převod do přirozeného jazyka

od databáze ke srozumitelnému výsledku

Překlad do přirozeného jazyka

- Zachycení sémantiky relací, atributů a vztahů v přirozeném jazyce
- Předpoklady:
  - Název relace zachycuje její konceptuální význam
  - Fyzický význam reprezentován názvem alespoň jednoho atributu z dané relace charakterizuje záznam
    - => **hlavní atribut** – projekční hrana má hodnotu 1
    - Určuje doménový expert
- Syntéza – přes primární a cizí klíče

Šablony (1)

- Alfanumerické výrazy k obohacení výsledku dotazu
- Na hranách grafu databázového schématu
- **label(u, v)** – vztahová šablona ($e(u,z) \in E$, $G=(V, E)$)
  - Projekční hrana – vztah atributu k hlavnímu atributu
    - Př. YEAR of MOVIE (.TITLE)
    - TITLE of MOVIE (u hlavního atributu)
  - Spojovací hrana – vztah hlavních atributů spojených relací
    - Př. GENRE (.GENRE) of MOVIE (.TITLE)

Šablony (2)

- **l(n)** – popisek uzlu n
  - Př. l(TITLE) = „title“
- label(a,b) = expr1+ l(u) + expr2 + l(z) + expr3
- Podpora pro tvorbu šablon
  - Proměnné
  - Cykly
  - Funkce
  - Makra

Príklady šablon relací

- DIRECTOR
  - *@DNAME* + “ *was born on* ” + *@BDATE* + “ *in* “ + *@BLOCATION*
- MOVIE
  - @TITLE + “ (” + @YEAR + “)”

Příklad šablon vztahů

- Mezi relacemi DIRECTOR a MOVIE
  - label(DIRECTOR, MOVIE) = expr1 + @DNAME + expr2 + MOVIE_LIST
- Makro MOVIE_LIST
  - DEFINE MOVIE_LIST as
    [i < arityOf(@TITLE)]
    {@TITLE[$i] + " (" + @YEAR[$i]+ "), "}
    [i = arityOf(@TITLE)]
    {@TITLE[$i] + " (" + @YEAR[$i]+ "). "}

- expr1 ← "As a director, "
- expr2 ← " 's work includes"

Výsledek

- „*Woody Allen was born on December 1, 1935 in Brooklyn, New York, USA. As a director, Woody Allen's work includes Match Point (2005), Melinda and Melinda (2004), Anything Else (2003). MatchPoint is a Drama, Thriller. Melinda and Melinda is Comedy, Drama. Anathing Else is Comedy, Romace.*“

  - Obdobně pro Woodyho Allena jako herce
    - (“As an actor, …”)

Použité zdroje

kde najít podrobnosti

Použité zdroje

- G. Koutrika, A. Simitsis, Y. Ioannidis: Précis: The Essence of a Query Answer.
  - http://doi.ieeecomputersociety.org/10.1109/ICDE.2006.114
  - výtah z druhého článku
- A. Simitsis, G. Koutrika, Y. Ioannidis: Présic: from unstructured keywords as queries to structured databases as answer.
  - http://dx.doi.org/10.1007/s00778-007-0075-9