

Úvod do databázových systémů

10. cvičení

Ing. Petr Lukáš

petr.lukas@nativa.cz

Ostrava, 2012

Opakování

Univerzální relační schéma – „široká“ nepřehledná tabulka, která obsahuje všechny atributy (např. modelovaného systému)

Funkční závislost – ze znalosti hodnot nějaké množiny hodnot atributů (a samozřejmě obsahu databáze) znám i množinu hodnot jiných atributů.

Armstrongovy axiomy – odvozovací pravidla pro funkční závislosti.

Uzávěry množiny atributů – hodnoty kterých všech atributů jsem schopný získat na základě určité dané množiny atributů

Klíč schématu – atributy, kterými jednoznačně identifikuji celý záznam, tzn. pokud znám hodnoty těchto atributů, umím v univerzálním schématu dohledat obsah celého záznamu

Minimální neredundantní pokrytí – rozložím FZ na elementární FZ, z levých stran odstraním nadbytečné atributy a nakonec celé nadbytečné FZ.

Pojmy

1. *Redundance, konzistence, integrita*
2. *Klíč, primární a sekundární atributy*
3. *Normální formy relací (1. NF, 2. NF, 3. NF, BCNF)*
4. *Dekompozice schématu relace*
5. *Zákon zachování informace*
6. *Zákon zachování množiny FZ*
7. *Algoritmus dekompozice*
8. *Algoritmus syntézy*

Redundance, konzistence, integrita

Redundance

Nadbytečnost dat. Stejná data jsou v databázi uložena vícekrát.

Konzistence

Data označující stejnou realitu jsou stejná. Jinými slovy, když už máme redundanci, musíme zajistit konzistenci.

Integrita

Data respektují všechna integritní omezení. Někdy se uvádí, že integrita znamená soulad s realitou.

Redundance, konzistence, integrita

idZam	jméno	příjmení	idKatedry	název	předmět	kredity	učebna	hodina
1	Petr	Lukáš	1	ABC	UDBS	6	A1033	16:00
2	Pavel	Mladý	2	DEF	ML	6	A1034	17:30
3	Jan	Starý	2	DEF	TIS	4	A1033	09:00
1	Petr	Lukáš	1	ABC	DAIS	4	G317b	10:45
2	Pavel	Mladý	2	DEF	TIS	4	G317a	09:00
4	Karel	Velký	3	XYZ	ML	6	E320	09:00
5	Josef	Dlouhý	3	XYZ	UDBS	6	E320	10:45
7	Martin	Zelený	1	ABC	TIS	4	E322	12:30
2	Pavel	Mladý	2	DEF	DAIS	4	A1033	12:30

Redundance?

Konzistence?

Integrita?

Redundance, konzistence, integrita

idZam	jméno	příjmení	idKatedry	název	předmět	kredity	učebna	hodina
1	Petr	Lukáš	1	ABC	UDBS	6	A1033	16:00
2	Pavel	Mladý	2	DEF	ML	6	A1034	17:30
3	Jan	Starý	2	DEF	TIS	4	A1033	09:00
1	Petr	Lukáš	1	ABC	DAIS	4	G317b	10:45
2	Pavel	Mladý	2	DEF	TIS	4	G317a	09:00
4	Karel	Velký	3	XYZ	ML	6	E320	09:00
5	Josef	Dlouhý	3	XYZ	UDBS	6	E320	10:45
7	Martin	Zelený	1	ABC	TIS	4	E322	12:30
2	Pavel	Mladý	2	DEF	DAIS	4	A1033	12:30

Redundance? **ANO**

Konzistence? **ANO**

Integrita? **ANO**

Klíč, primární a sekundární atributy

Klíč

Množina atributů K je klíčem R , pokud jsou všechny atributy z R funkčně závislé na K . Omezujeme se pouze na ty klíče, kde žádná podmnožina už sama klíčem není. Protože těchto klíčů může být více, vybíráme jeden a ten označíme jako **primární**.

Primární atributy

Atributy, které jsou součástí některého z klíčů.

Sekundární atributy

Atributy, které nejsou součástí žádného klíče.

Klíč, primární a sekundární atributy

Osoba (id, rč, jméno, příjmení, ulice, město, psc)

Klíče:

Primární klíč:

Primární atributy:

Sekundární atributy:

Klíč, primární a sekundární atributy

Osoba (id, rč, jméno, příjmení, ulice, město, psc)

Klíče:

{id}, {rč}

(pozor, ne {id, rč} – neplatila by podmínka, že žádná podmnožina sama o sobě není klíčem)

Primární klíč:

např. {id}

Primární atributy:

{id, rc}

Sekundární atributy:

{jméno, příjmení, ulice, mesto, psc}

Normální formy relací

1. Normální forma

Relační schéma je v 1. NF, jestliže obsahuje pouze atomické (dále nedělitelné) atributy.

př. Osoba (id, jméno, adresa)

Normální formy relací

1. Normální forma

Relační schéma je v 1. NF, jestliže obsahuje pouze atomické (dále nedělitelné) atributy.

př. Osoba (id, jméno, adresa)

Není v 1. NF – adresa není atomický atribut

Normální formy relací

2. Normální forma

Relační schéma je ve 2. NF, jestliže je zároveň v 1. NF a zároveň je každý sekundární atribut **úplně** závislý na každém klíči. Tzn. není závislý na žádném podklíči.

př. Rozvrh (hodina, učebna, předmět, kapacita)

Normální formy relací

2. Normální forma

Relační schéma je ve 2. NF, jestliže je zároveň v 1. NF a zároveň je každý sekundární atribut **úplně** závislý na každém klíči. Tzn. není závislý na žádném podklíči.

př. Rozvrh (hodina, učebna, předmět, kapacita)

Není ve 2. NF – kapacita **není úplně závislá** na kombinaci hodina vs. učebna, ale jen na učebně.

Normální formy relací

3. Normální forma

Relační schéma je ve 3. NF, jestliže je zároveň ve 2. NF a zároveň neexistují netriviální závislosti mezi sekundárními atributy.

př. **Osoba** (id, jméno, firma, adresa_firmy)

Normální formy relací

3. Normální forma

Relační schéma je ve 3. NF, jestliže je zároveň ve 2. NF a zároveň neexistují netriviální závislosti mezi sekundárními atributy.

př. **Osoba** (id, jméno, firma, adresa_firmy)

Není ve 3. NF – existuje netriviální závislost mezi firmou a adresou firmy.

Normální formy relací

Boyce-Coddova normální forma (BCNF)

Relační schéma je v BCNF, jestliže je zároveň ve 3. NF a pro každou FZ $X \rightarrow Y$ platí, že X je klíč nebo nadklíč.

př. Koupil (pld, zld, názevP, jménoZ)

Normální formy relací

Boyce-Coddova normální forma (BCNF)

Relační schéma je v BCNF, jestliže je zároveň ve 3. NF a pro každou FZ $X \rightarrow Y$ platí, že X je klíč nebo nadklíč.

př. Koupil (pld, zld, názevP, jménoZ)

Není v BCNF – klíčem je {pld, zld}, ale budou platit závislosti $pld \rightarrow \text{názevP}$ a $zld \rightarrow \text{jménoZ}$. Levé strany obou závislostí nejsou klčem ani nadklíčem.

Dekompozice schématu relace

Dekompozice relačního schématu $R(A)$ je množina relačních schémata $RO = \{R_1(A_1), R_2(A_2), \dots, R_n(A_n)\}$, přičemž platí, že $A_1 \cup A_2 \cup \dots \cup A_n = A$.

Obykle se omezujeme jen na binární dekompozici.

př. Osoba (id, jméno, firma, adresa_firmy)

Jeden z možných rozkladů může vypadat např.:

O_1 (id, jméno, firma)

O_2 (firma, adresa_firmy)

Zákon zachování informace

1. Mějme původní schéma $R(A)$
2. Množinu funkčních závislostí F
3. Rozklad $RO = \{R_1(A_1), R_2(A_2)\}$

Zákon zachování informace

Ke ztrátě informace vzhledem k F nedojde, jestliže pro každou relaci R platí, že $R = R_1 \bowtie R_2$.

R (*idZam, jméno, firma, název*)

idZam	jméno	firma	název
1	Petr	1	VŠB
2	Pavel	2	Tieto
3	Lukáš	1	VŠB
4	Jakub	2	Tieto

=

R_1 (*idZam, jméno, firma*)

idZam	jméno	firma
1	Petr	1
2	Pavel	2
3	Lukáš	1
4	Jakub	2

\bowtie

R_2 (*firma, název*)

firma	název
1	VŠB
2	Tieto

Zákon zachování množiny FZ

1. Mějme původní schéma $R(A)$
2. Množinu funkčních závislostí F
3. Rozklad $RO = \{R_1(A_1), R_2(A_2)\}$

Pokrytí F (značeno $F+$)

Všechny možné funkční závislosti, které jsou buď přímo v F nebo je lze z F odvodit.

Projekce množiny FZ

Označme $F[A_1]$ (resp. $F[A_2]$) jako projekci množiny FZ na množinu atributů A_1 (resp. A_2). Jedná se o ty FZ z $F+$, jejichž levá i pravá strana je podmnožinou A_1 (resp. A_2).

Zákon zachování množiny FZ

Ke ztrátě množiny FZ nedojde, jestliže $(F[A_1] \cup F[A_2])+ = F+$.

Zákon zachování množiny FZ

R (idZam, jméno, firma, název)

idZam	jméno	firma	název
1	Petr	1	VŠB
2	Pavel	2	Tieto
3	Lukáš	1	VŠB
4	Jakub	2	Tieto

idZam \rightarrow jméno, firma
firma \rightarrow název

R₁ (idZam, jméno)

idZam	jméno
1	Petr
2	Pavel
3	Lukáš
4	Jakub

F[R₁]: idZam \rightarrow jméno

R₂ (idZam, firma)

idZam	firma
1	1
2	2
3	1
4	2

F[R₂]: idZam \rightarrow firma

R₃ (firma, název)

firma	název
1	VŠB
2	Tieto

F[R₃]: firma \rightarrow název

Zákon zachování množiny FZ

R (idZam, jméno, firma, název)

idZam	jméno	firma	název
1	Petr	1	VŠB
2	Pavel	2	Tieto
3	Lukáš	1	VŠB
4	Jakub	2	Tieto

idZam → jméno, firma
firma → název

R₁ (idZam, jméno)

idZam	jméno
1	Petr
2	Pavel
3	Lukáš
4	Jakub

F[R₁]: idZam → jméno

R₂ (firma, název)

firma	název
1	VŠB
2	Tieto

F[R₂]: firma → název

**Ztratila se nám závislost
mezi idZam a firmou.**

Jak navrhnout schéma v BCNF?

1. Prvním krokem je vždy vytvoření univerzálního schématu. Tzn. jedné tabulky se všemi atributy.
2. Provedeme dekompozici takového schématu pomocí:
 - a) Algoritmu dekompozice
 - b) Algoritmu syntézy

Algoritmus dekompozice

R (A, B, C, D, E, F, G)

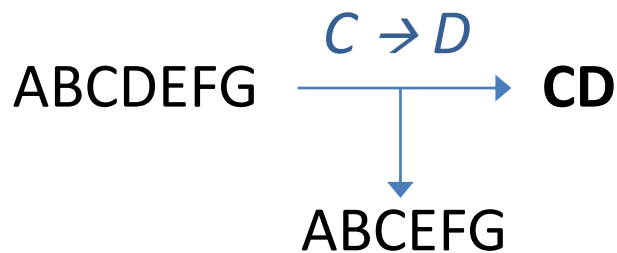
F: {AB \rightarrow C, C \rightarrow D, B \rightarrow E, E \rightarrow F, C \rightarrow G}

ABCDEFGG

Algoritmus dekompozice

R (A, B, C, D, E, F, G)

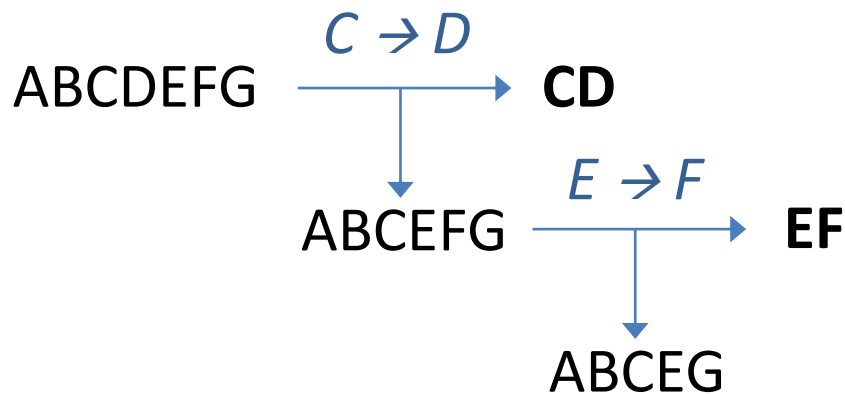
F: {AB → C, C → D, B → E, E → F, C → G}



Algoritmus dekompozice

R (A, B, C, D, E, F, G)

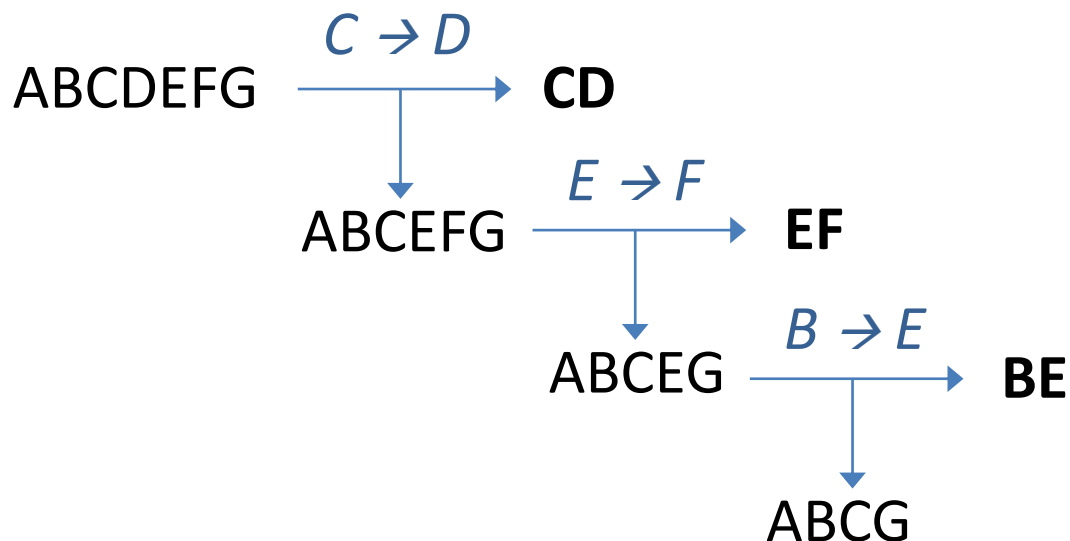
F: {AB → C, C → D, B → E, E → F, C → G}



Algoritmus dekompozice

R (A, B, C, D, E, F, G)

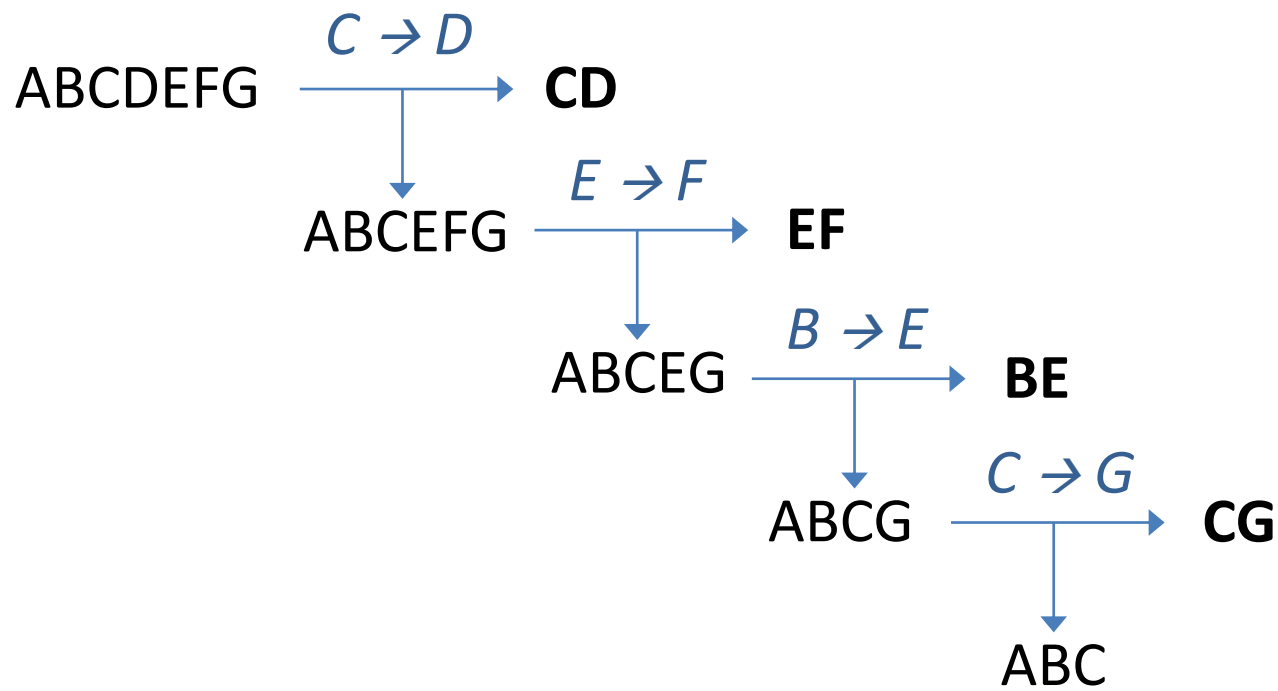
F: {AB → C, C → D, B → E, E → F, C → G}



Algoritmus dekompozice

R (A, B, C, D, E, F, G)

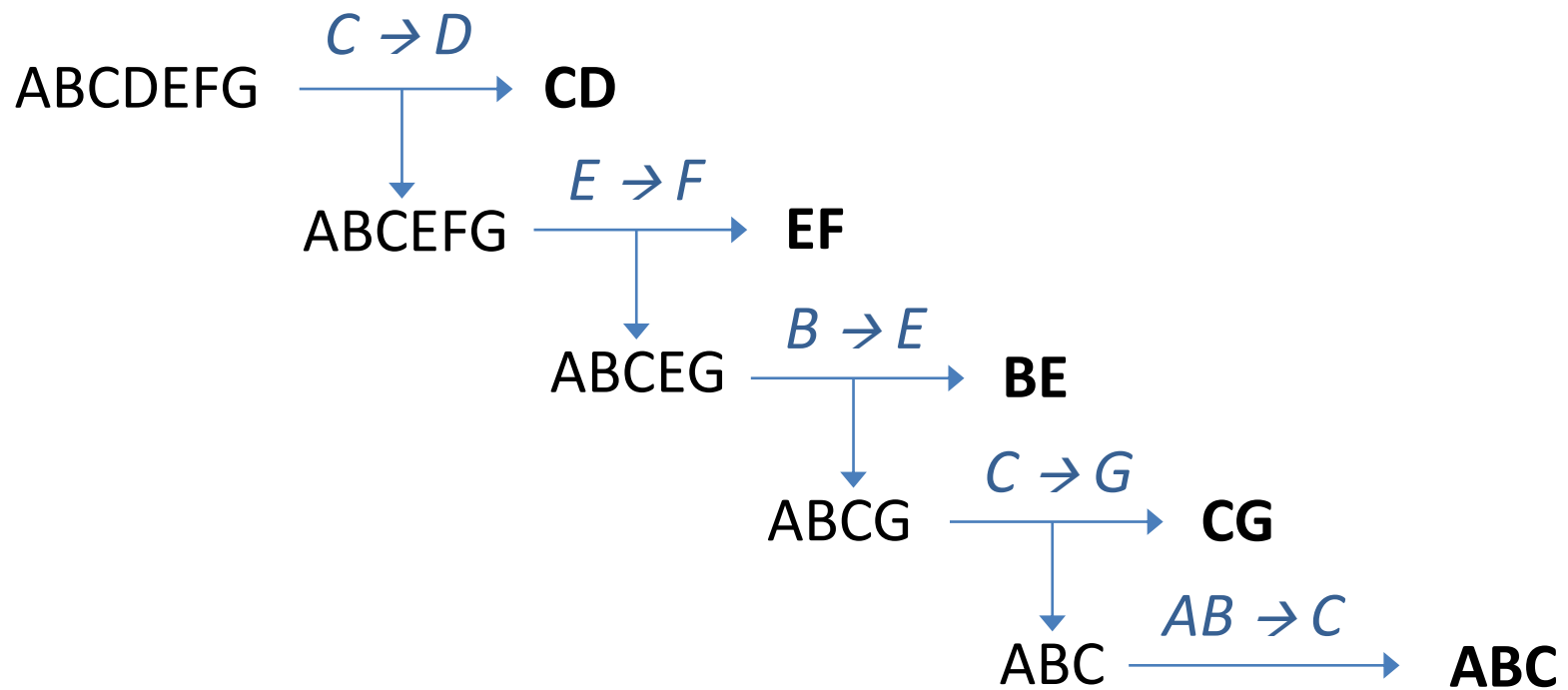
F: {AB → C, C → D, B → E, E → F, C → G}



Algoritmus dekompozice

R (A, B, C, D, E, F, G)

F: {AB → C, C → D, B → E, E → F, C → G}



Algoritmus dekompozice

Získali jsme tedy rozklad

$R_1(\text{CD}), R_2(\text{EF}), R_3(\text{BE}), R_4(\text{CG}), R_5(\text{ABC})$

pozn.:

1. Výsledná schémata po použití algoritmu dekompozice jsou v BCNF
2. Výsledek algoritmu závisí na pořadí „vytahování“ FZ
3. Ne vždy je možno dekompozici použít. V mnoha případech vzniklý rozklad nedodrží zákon o zachování množiny FZ.

Algoritmus syntézy

1. Sestavíme univerzální schéma, minimální neredundantní pokrytí množiny FZ a nalezneme klíč. Pro každou FZ vytvoříme zvlášť relaci, tzn. v prvním kroku bude přesně tolik „tabulek“, kolik máme FZ v min. nered. pokrytí.
2. Spojíme relace které vznikly z FZ se stejnými levými stranami.
3. Spojíme relace s ekvivalentními klíči, tzn. těmi klíči, jejichž uzávěr je stejný. Pozor. v tomto kroku může dojít k porušení podmínek pro BCNF.
4. Pokud existuje atribut univerzálního schématu, který dosud není zařazen v žádné ze vzniklých relací (tzn. nebyl obsažen v žádné FZ), pak jej přidáme do libovolné vzniklé relace.
5. Pokud žádná z relací neobsahuje celý klíč univerzálního schématu, pak vytvoříme novou relaci, která se bude skládat z atributů tohoto klíče.

Algoritmus syntézy

R (A, B, C, D, E, F, G)

F: {AB → C, C → D, B → E, E → F, C → G}

1. Minimální neredundantní pokrytí a vytvoření mnoha „malých“ relací. Klíč schématu je **AB**.
RO₁ = (R₁(ABC), R₂(CD), R₃(BE), R₄(EF) R₅(CG))
2. Spojíme relace, které vznikly z FZ se stejnými levými stranami. Žádné takové nejsou, takže **RO₂ = RO₁**
3. Spojíme relace s ekvivalentními klíči. Opět žádné nejsou, takže **RO₃ = RO₂**
4. Pokud zbyl nějaký nezpracovaný atribut, přidáme jej do kterékoli z relací. Nezbyl, takže nic, **RO₄ = RO₃**
5. Není-li klíč obsažen v žádné z relací, pak vytvoříme novou relaci obsahující atributy klíče. Klíč **AB** je obsažen v **R₁**, takže **RO₅ = RO₄** a máme výsledek **R₁(ABC), R₂(CD), R₃(BE), R₄(EF) R₅(CG)**

Shrnutí

1. *Redundance, konzistence, integrita*
2. *Klíč, primární a sekundární atributy*
3. *Normální formy relací (1. NF, 2. NF, 3. NF, BCNF)*
4. *Dekompozice schématu relace*
5. *Zákon zachování informace*
6. *Zákon zachování množiny FZ*
7. *Algoritmus dekompozice*
8. *Algoritmus syntézy*

Úvod do databázových systémů

www.dbedu.cs.vsb.cz

jméno: **student**

heslo: **tuodbedu**

Vpravo sloupec -> České kurzy -> UDBS -> Cvičení – Relaçní algebra