

Normalizace a normální formy

Michal Valenta

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze
©Michal Valenta, 2022

BI-DBS, LS 2021/2022

<https://courses.fit.cvut.cz/BI-DBS/>



Návrh relačního schématu

Existují dva přístupy:

- 1 normalizační teorie
Metoda návrhu pomocí funkčních závislostí.
- 2 z konceptuálního schématu
Metoda použití transformačních pravidel.

Poznámka: použijeme-li transformaci z konceptuálního schématu, **nemáme zaručeno**, že výsledné schéma bude normalizované! Samotné konceptuální schéma totiž nemusí být normalizované.

Praxe:

- 1 konceptuální schéma (i zde je možné normalizovat)
- 2 transformace na relační schéma
- 3 kontrola a případná normalizace (dekompozicí) nenormalizovaných relací

Kvalita schématu a normalizace

Uvažujme relaci:

PROGRAM(NAZEV_K, JMENO_F, ADRESA, DATUM)

Aktualizační anomálie (Codd)

- Změní-li se adresa kina, je nutné ji měnit víckrát,
- nehraje-li kino zrovna nic, ztrácíme jeho adresu,
- chceme-li přidat nové kino s adresou, lze to jen když se tam hraje nějaký film.

Jak to vyřešíme?

Normalizace dekompozicí

KINO(NAZEV_K, ADRESA)

MA_NA_PROGRAMU(NAZEV_K, JMENO_F, DATUM)

$MA_NA_PROGRAMU[NAZEV_K] \subseteq KINO[NAZEV_K]$

Dekompozicí jsme se zbavili všech aktualizačních anomálií.

Funkční závislost – neformálně

Hodnoty některých atributů funkčně závisí na hodnotách jiných atributů. **Například:**

- 1 Ke každému kinu existuje nejvýše jedna adresa.
- 2 Pro každou dvojici {kino, film} existuje nejvýše jedno datum, kdy dané kino má daný film na programu.

Což budeme zapisovat:

- 1 NAZEV_K \rightarrow ADRESA
- 2 {NAZEV_K, JMENO_F} \rightarrow DATUM

A číst:

- 1 “Atribut NAZEV_K (funkčně) určuje atribut ADRESA.”
nebo:
“Atribut ADRESA (funkčně) závisí na atributu NAZEV_K.”
- 2 “Dvojice (atributů) NAZEV_K, JMENO_F (funkčně) určuje atribut DATUM.”
nebo:
“Atribut DATUM (funkčně) závisí na (dvojici atributů) NAZEV_K, JMENO_F.”

Funkční závislosti – integritní omezení

- funkční závislosti (FZ) vyjadřují integritní omezení
- **připomínka**: integritní omezení (obecně) jsou tvrzení, které určují jaká data v databázi být mohou a jaká ne
- **připomínka**: schéma relační databáze je $\{R(A), I\}$
- FZ uvádí do souvislosti prvky z domén příslušných atributů, je to funkce $f : A_1 \rightarrow A_2$

Funkční závislosti jsou jeden ze způsobů vyjádření integritních omezení.

Všechna integritní omezení se nedají vyjádřit pomocí FZ.

Kvalita schématu - příklad

Mějme databázi s rozvrhem předmětů:

Rozvrh (Přednáška, Učitel, Místnost, Hodina, Student, Znamka)

Nechť platí toto (vnitropodnikové) pravidlo (IO1):

„Každá přednáška je přednášena nejvýše jedním učitelem.“

Z pohledu relační teorie:

„K jedné hodnotě z **dom(Přednáška)** se přiřadí nejvýše jedna hodnota z **dom(Učitel)**.“

Přednáška \rightarrow Učitel

což budeme v dalším výkladu zkracovat:

$P \rightarrow U$

Kvalita schématu – příklad

Pokud budou všechna data v **jedné relaci ROZVRH**, bude v ní zřejmě **mnoho aktualizáčních anomálií**. Nahraďme relaci rozvrh několika jinými relacemi (schématem relační DB) tak, aby výsledek měl „**rozumné vlastnosti**“.

výchozí relace: $R(P, U, M, H, S, Z)$ (stručněji: $PUMHSZ$)
možné náhrady:

$$S_I = \{PU, HMP, HUM, PSZ, HSM\}$$

$$S_{II} = \{PU, HSP, PSZ, HSM\}$$

$$S_{III} = \{PU, HSM, PSZ, HMP\}$$

$$S_{IV} = \{PU, HMP, PSZ, PSZ\}$$

$$S_V = \{HMPU, PSZ, HSM\}$$

$$S_{VI} = \{PU, HMP, PSZ\}$$

$$S_{VII} = \{PSUHM, PSZ\}$$

Které ze schémat $S_I..S_{VII}$ je nejlepší?

Odhalení funkčních závislostí mezi atributy

Odhalení FZ mezi atributy schématu:

P	U	M	H	S	Z
Programování	Kryl	S7	Po9	Novák	2
Programování	Kryl	S3	Út3	Novák	2
Programování	Kryl	S7	Po9	Volák	3
Programování	Kryl	S3	Út3	Volák	3
Systémy	Král	S4	Po7	Zíka	1
Systémy	Král	S4	Po7	Tupý	2
Systémy	Král	S4	Po7	Novák	2
Systémy	Král	S4	Po7	Bílý	1

? platí ? $U \rightarrow HM$! jistě neplatí !, tedy: $U \nrightarrow HM$

zřejmě platí: $P \rightarrow U$, $HM \rightarrow P$, $HU \rightarrow M$, $HS \rightarrow M$

? a co toto ? : $PS \rightarrow Z$

X-hodnota (k formální definici funkční závislosti)

Mějme schéma $R(A)$, uvažujme $X \subseteq A$

X-hodnota

Jsou-li atributy v $\mathbf{X} \{X_1 : dom(X_1), \dots, X_n : dom(X_n)\}$,
pak **X-hodnotou** je libovolný prvek z kartézského součinu
 $dom(X_1) \times dom(X_2) \times \dots \times dom(X_n)$.

x1	x2	x3	x4	x5	x6	x7	x8	x9

Funkční závislost (formální definice)

Mějme schéma $R(A)$.

Funkční závislost

Mějme množiny atributů $B \subseteq A$, $C \subseteq A$. Říkáme, že C závisí funkčně na B (nebo B funkčně určuje C), jestliže ke každé B -hodnotě existuje nejvýše jedna C -hodnota.

značíme:

$$B \rightarrow C$$

resp.:

$$B \nrightarrow C$$

Pozorování

$PS \rightarrow S$ platí vždy

$$\left. \begin{array}{l} PS \rightarrow S \\ PS \rightarrow Z \end{array} \right\} \Leftrightarrow PS \rightarrow SZ$$

Z výchozí množiny funkčních závislostí lze pomocí „určitých“ **pravidel** odvozovat další FZ.

Sada “korektních” odvozovacích pravidel

Mějme $R(A)$, nechť $X \subseteq A$, $Y \subseteq A$, $Z \subseteq A$.

Armstrongova pravidla

triviální funkční závislosti

jestliže $Y \subseteq X$, pak $X \rightarrow Y$ (FZ1)

př.: $UM \rightarrow U$

tranzitivita

jestliže $X \rightarrow Y$ a $Y \rightarrow Z$, pak $X \rightarrow Z$ (FZ2)

př.: $HS \rightarrow HM$ a $HM \rightarrow P$, pak také platí $HS \rightarrow P$

kompozice pravé strany

jestliže $X \rightarrow Y$ a $X \rightarrow Z$, pak $X \rightarrow YZ$ (FZ3)

dekompozice pravé strany

jestliže $X \rightarrow YZ$, pak $X \rightarrow Y$ a $X \rightarrow Z$ (FZ4)

Použití odvozovacích pravidel

Mějme vstupní relaci $R(M, H, U, P, S, Z)$

a sadu funkčních závislostí:

$$F = \{P \rightarrow U, HM \rightarrow P, HU \rightarrow M, PS \rightarrow Z, HS \rightarrow M\}$$

Odvodíme:

- Podle (FZ1) platí $HM \rightarrow H$ a $HU \rightarrow H$.
- Podle (FZ3) z $HU \rightarrow H$ a $HU \rightarrow M$ odvodíme $HU \rightarrow HM$.
- Podle (FZ2) z $HM \rightarrow P$ a $P \rightarrow U$ odvodíme $HM \rightarrow U$.
- Podle (FZ3) z $HM \rightarrow H$ a $HM \rightarrow U$ odvodíme $HM \rightarrow HU$.

Vidíme, že HM a HU jsou **funkčně ekvivalentní**: $HM \leftrightarrow HU$

Tranzitivní uzávěr, klíč relace

Tranzitivní uzávěr množiny atributů X^+ vzhledem k F

je množina všech atributů funkčně závislých na X . Označujeme jej X^+ .

Výpočet X^+ (slovní popis):

- 1 Do X^+ přidáme X .
- 2 Procházíme F , je-li levá strana FZ již v X^+ , přidáme stranu pravou.
- 3 Opakujeme předchozí bod dokud do X^+ něco přibývá.

Klíč relace

Mějme $R(A)$, necht' $K \subseteq A$.

K je klíčem schématu $R(A)$, jestliže splňuje dvě vlastnosti:

- 1 $K \rightarrow A$
- 2 neexistuje $K' \subset K$ taková, že $K' \rightarrow A$.

Příklad – nalezení klíče relace

Mějme vstupní relaci $R(M, H, U, P, S, Z)$ a sadu funkčních závislostí:
 $F = \{P \rightarrow U, HM \rightarrow P, HU \rightarrow M, PS \rightarrow Z, HS \rightarrow M\}$

Úkol: najděte alespoň jeden klíč relace R vzhledem k F .

Výpočet:

$$P^+ = \{P, U\}$$

$$HM^+ = \{H, M, P, U\}$$

$$HU^+ = \{H, U, M, P\}$$

$$PS^+ = \{P, S, Z, U\}$$

$$HS^+ = \{H, S, M, P, Z, U\} \text{ (tranzitivní uzávěr HS obsahuje celou relaci R)}$$

(test minimality) Protože $H^+ = \{H\}$ a $S^+ = \{S\}$, je **HS klíčem relace R**.

Postup slovně:

- Do klíče určitě patří atributy, které nejsou v žádné FZ.
- Idea: Atribut z levé strany nějaké FZ by mohl do klíče patřit.
 - ▶ Vezmeme nějakou FZ a spočítáme **tranzitivní uzávěr** levé strany.
 - ▶ Obsahuje celou relaci? ANO: Je minimální? ANO: **máme klíč**
 - ▶ NE: Zkusíme jinou FZ nebo něco přidáme.
 - ▶ Opakujeme dokud klíč nenajdeme.
 - ▶ **Klíčů může být více**, tento postup nezaručí, že najdeme všechny.

Normální formy – motivační příklad 1/2 (CK->N)

PROGRAM	NÁZEV_K	JMÉNO_F	ADRESA	DATUM
	Blaník	Top gun	Václavské nám. 4	29.03.94
	Blaník	Kmotr	Václavské nám. 4	08.03.94
	Mír	Nováček	Starostrašnická 3	10.03.94
	Mír	Top gun	Starostrašnická 3	09.03.94
	Mír	Kmotr	Starostrašnická 3	08.03.94

Integritní omezení:

- IO1: Klíčem schématu je NÁZEV_K, JMÉNO_F.
- IO2: Každé kino má právě jednu adresu.

Relace obsahuje redundance a mohou nastat aktualizací anomálie.

Normální formy – motivační příklad 1/2 (CK->N)

Intuitivním řešením je dekompozice
ADRESÁŘ(NÁZEV_K, ADRESA),
PROGRAMY(NÁZEV_K, JMÉNO_F, DATUM)

PROGRAMY

NÁZEV_K	JMÉNO_F	DATUM
Blaník	Top gun	29.03.94
Blaník	Kmotr	08.03.94
Mír	Nováček	10.03.94
Mír	Top gun	09.03.94
Mír	Kmotr	08.03.94

ADRESÁŘ

NÁZEV_K	ADRESA
Blaník	Václavské nám. 4
Mír	Starotrašnická 3

- adresa kina je pouze jednou (odstraněna redundance)
- lze evidovat i kino, kde se (právě) nic nehraje (nehrozí ztráta informace o kinu, když bude 'stát')

podstata řešení: odstraněna závislost neklíče (adresa) na pouhém podklíči(Název_k)

Normální formy – motivační příklad 2/2 (N1->N2)

FILM1	JMÉNO_F	HEREC	OBČANSTVÍ	ROK
	Černí baroni	Landovský	CZ	94
	Top gun	Cruise	USA	86
	Kmotr	Brando	USA	72
	Nováček	Brando	USA	90
	Vzorec	Brando	USA	80

Integritní omezení:

- IO1: Klíčem schématu je JMÉNO_F.
- IO2: Každý herec má právě jedno občanství

Relace obsahuje redundance a mohou nastat aktualizací anomálie.

Normální formy – motivační příklad 2/2 (N1->N2)

Intuitivním řešením je dekompozice
OSOBNÍ_ÚDAJE(HEREC, OBČANSTVÍ)
FILM2(JMÉNO_F, HEREC, ROK)

OSOBNÍ_ÚDAJE		FILM2		
HEREC	OBČANSTVÍ	JMÉNO_F	HEREC	ROK
Landovský	CZ	Černí baroni	Landovský	94
Cruise	USA	Top gun	Cruise	86
Brando	USA	Kmotr	Brando	72
		Nováček	Brando	90
		Vzorec	Brando	80

- občanství herce je pouze jednou (odstraněna redundance)
- lze evidovat i občanství herce, jehož filmy vypadly z db (nehrozí ztráta informace o občanství herce, který “stojí”)

podstata řešení: odstraněna závislost neklíče (občanství) na jiném neklíči (herce)

Normální formy – motivační příklady – rozbor, 3NF

V obou předchozích příkladech byly neklíčové atributy závislé na klíči. Některé z nich však nepřímo - tranzitivně.

V prvním případě šlo o tranzitivitu:

klíč \rightarrow podklíč \rightarrow neklíč

V druhém případě šlo o tranzitivitu:

klíč \rightarrow neklíč \rightarrow neklíč

Jsou-li všechny **neklíčové atributy závislé na klíči přímo** a nikoliv tranzitivně, pak je **relace ve 3. normální formě (3NF)**.

Poznámka1: Má-li schéma více klíčů ($\text{klíč1} \leftrightarrow \text{klíč2}$), nebude nám vadit $\text{klíč1} \rightarrow \text{klíč2} \rightarrow \text{neklíč}$.

Poznámka2: Jsou-li všechny atributy schématu součástí nějakého klíče, je schéma ve 3NF.

Tranzitivní závislost, 3NF – definice

Tranzitivní závislost

Mějme $R(A)$ Nechť $X \subset A$, $Y \subset A$ a $C \in A$, $C \notin X$ a $C \notin Y$.

Nechť dále $X \rightarrow Y \rightarrow C$ a neplatí, že $Y \rightarrow X$.

Pak říkáme, že C je tranzitivně závislý na X .

Třetí normální forma (3NF)

Říkáme, že schéma relace R je ve 3. normální formě (3NF), jestliže žádný neklíčový atribut schématu R není tranzitivně závislý na žádném klíči schématu.

3NF (jinými slovy)

Každý neklíčový atribut je na (každém) klíči schématu závislý přímo, nikoliv tranzitivně.

Poznámka: závislost dvou podklíčů 3NF neřeší, tomu se věnuje BCNF (viz dále).

BCNF – motivační příklad

Mějme *ROZVRH*(*MHUP*) a platí

$HU \rightarrow M$, $HM \rightarrow P$, $P \rightarrow U$

Lze odvodit klíče: *HU*, *HM*, *HP*.

$P \rightarrow U$ je závislost mezi dvěma podklíči.

ROZVRH vyhovuje kritériu pro 3NF. Proč?

... a přeci je v datech redundance!

ROZVRH	PREDNASKA	UCITEL	MISTNOST	HODINA
	Systémy	Král	S4	Po7
	Programování	Kryl	S7	Po9
	Programování	Kryl	S3	Ut11

BCNF – motivační příklad

Existuje zde závislost $\text{část_klíče1} \rightarrow \text{část_klíče2}$.

V našem případě: $P \rightarrow U$.

Dekompozice:

OBS(P,U)

ROZVRH1(HMP)

Opět platí, že:

- zmizela redundance v atributu U
- neztratí se informace, že Kryl přednáší Programování, když toto vypadne z rozvrhu

Řešení spočívá v odstranění závislosti části jednoho klíče na části druhého klíče.

BCNF

Říkáme, že schéma relace R je v Boyce - Coddově normální formě (BCNF), jestliže pro každou netriviální závislost $X \rightarrow Y$ platí, že X obsahuje klíč schématu R .

Poznámky:

Každé schéma, které je v BCNF, je také ve 3NF. Obrácené tvrzení obecně neplatí.

Má-li ale schéma jediný klíč, nebo jednoduché klíče, potom je-li ve 3NF je také v BCNF.

BCNF – (další) příklad

Uvažujme schéma relace:

ADRESÁŘ(MĚSTO, ULICE, DUM, PSČ)

F: {MĚSTO, ULICE} \rightarrow PSČ, PSČ \rightarrow MĚSTO

{MĚSTO, ULICE, DUM} je klíčem (\rightarrow {PSČ, MĚSTO, ULICE, DUM})

{PSČ, ULICE, DUM} je klíčem (\rightarrow {PSČ, MĚSTO, ULICE, DUM})

Schéma nemá žádný neklíčový atribut a je tedy ve 3NF. Nikoliv však v BCNF.

ADRESÁŘ lze nahradit dekompozicí.

dekompozice1:

A1(PSČ, MĚSTO)

B1(PSČ, ULICE, DUM)

dekompozice2:

A2(MĚSTO, ULICE, PSČ)

B2(MĚSTO, ULICE, DUM)

Ani jedna dekompozice v tomto případě není optimální, viz dále.

Normální formy – shrnutí

- Přehled normálních forem:
 - ▶ 1NF – atributy jsou atomické (nemáme strukturované a vícehodnotové atributy)
 - ▶ 2NF – žádný neklíčový atribut není závislý na části klíče (vždy závisí na celém klíči)
 - ▶ 3NF – neklíčové atributy závisí na klíči přímo (nikoliv tranzitivně)
 - ▶ BCNF – levá strana každé (netriviální) FZ obsahuje klíč schématu (nepřipouští se závislost typu “část klíče 1” \rightarrow “část klíče 2”)
 - ▶ teoreticky existují i další NF, ale pro základní seznámení stačí tyto
- Jak zjistíme, jestli je relace ve 3NF? Pro každou FZ platí (jedno z následujících tvrzení):
 - ▶ závislost je triviální ($A \rightarrow A$, $AB \rightarrow A$, ...)
 - ▶ na levé straně FZ je (některý) z klíčů relace
 - ▶ na pravé straně je část (některého) klíče relace
- Jak zjistím, že relace je v BCNF? Jako výše, jen bez poslední podmínky.
- Pokud je relace ve 3NF, je i ve 2NF.
- K čemu je dobrá normalizace? Snižuje redundanci a riziko aktualizací anomálií.

Normalizace dekompozicí

Normalizace je eliminace aktualizačních anomálií převedením relačního schématu do 3NF, resp. BCNF.

Dekompozice

Původní schéma:

$$R(U, F)$$

U je množina atributů relace R , F je množina FZ, které na R platí.

Dekomponované schéma:

$$\{R_i(U_i, F_i)\}_{i=1}^n, \text{ kde } \cup_{i=1}^n U_i = U$$

Dekompozicí vznikne množina relací. Každá má své atributy a FZ.

Požadavky na kvalitu dekompozice:

P1: Neztratíme žádnou FZ, která platila na původní relaci.

Formálně: $F^+ = (\cup_{i=1}^n F_i)^+$

P2: Spojením nově vzniklých relací nevzniknou nové n-tice.

Formálně: $R = *_{i=1}^n R_i[A_i]$, kde $*$ znamená přirozené spojení definované v RA.

P1: pokrytí původní množiny funkčních závislostí

Původní schéma a schémata získaná dekompozicí musí mít stejné funkční závislosti.

$$F^+ = (\cup_{i=1}^n F_i)^+$$

Příklad 1

ADRESÁŘ(MĚSTO, ULICE, DUM, PSČ)

$F = \{ \{ \text{MĚSTO, ULICE} \} \rightarrow \text{PSČ}, \text{PSČ} \rightarrow \text{MĚSTO} \}$

Dekompozice:

SEZNAM_POŠT(PSČ, MĚSTO), $F_1 = \{ \text{PSČ} \rightarrow \text{MĚSTO} \}$

POŠTOVNÍ_RAJON(PSČ, ULICE, DUM), $F_2 = \{ \}$

FZ $\{ \text{MĚSTO, ULICE} \} \rightarrow \text{PSČ}$ z F zde pokryta není.

Dekompozice nesplňuje podmínku P1 (pokrytí množiny FZ)

P1: pokrytí původní množiny funkčních závislostí

Příklad 2:

FILM1(JMÉNO_F, HEREC, PŘÍSLUŠNOST)

$F = \{ \text{HEREC} \rightarrow \text{PŘÍSLUŠNOST}, \text{JMÉNO_F} \rightarrow \text{HEREC}, \text{JMÉNO_F} \rightarrow \text{PŘÍSLUŠNOST} \}$

Dekompozice podle $\text{HEREC} \rightarrow \text{PŘÍSLUŠNOST}$:

OSOBNÍ_ÚDAJE(HEREC, PŘÍSLUŠNOST),

$F1 = \{ \text{HEREC} \rightarrow \text{PŘÍSLUŠNOST} \}$

FILM2(JMÉNO_F, ROK, HEREC), $F2 = \{ \text{JMÉNO_F} \rightarrow \text{HEREC} \}$

FZ $\text{JMÉNO_F} \rightarrow \text{PŘÍSLUŠNOST}$ z F **zdánlivě pokryta není**.

Je ji ale možné **odvodit z F1 a F2** pomocí Armstrongových pravidel.

Dekompozice splňuje podmínku P1 (pokrytí množiny FZ)

Poznámka: z množiny FZ F vyplývá, že klíčem schématu FILM1 je atribut JMÉNO_F.

P2: bezztrátová dekompozice

- Spojením nově vzniklých relací nesmí vzniknout žádné nové n-tice, které nebyly v relaci původní. Tedy, nové relace musí obsahovat stejná data jako měla relace původní.
- Používá se také pojem „bezztrátové spojení“ (loseless join).

O dekompozici schématu můžeme (z pohledu dat) uvažovat jako o několika projekcích původní relace. Každá projekce definuje novou relaci. Spojením všech dílčích relací pak musíme dostat stejná data jako měla relace původní.

$$R = *_{i=1}^n R_i[A_i]$$

Poznámka: na následujícím slide je příklad špatně provedené dekompozice.

Špatná dekompozice – příklad

Dekomponujeme relaci ZKOUSENI:

ZAPIS (PREDN, STUDENT) a
HODNOCENI (PREDN, ZNAMKA).

ZKOUSENI

PREDN	STUDENT	ZNAMKA
Programování	Novák	2
Programování	Volák	3
Systémy	Zika	1
Systémy	Tupý	2
Systémy	Novák	2
Systémy	Bílý	1

ZAPIS := ZKOUSENI[PREDN,STUDENT]

PREDN	STUDENT
Programování	Novák
Programování	Volák
Systémy	Zika
Systémy	Tupý
Systémy	Novák
Systémy	Bílý

HODNOCENI := ZKOUSENI[PREDN,ZNAMKA]

PREDN	ZNAMKA
Programování	2
Programování	3
Systémy	1
Systémy	2
Systémy	2
Systémy	1

Spojení $HODNOCENI \star ZAPIS$ **nebude stejné** jako ZKOUSENI.

Bude navíc obsahovat n-tice (Programování, Novák, 3) a (Programování, Volák, 2), které v relaci ZKOUSENI nejsou.

Dostaneme více entic, ale informace je méně, protože nevíme co platí.

Tato dekompozice **není BEZZTRÁTOVÁ (loseless)**.

Normalizace bezztrátovou dekompozicí

Normalizace dekompozicí – postup

- 1 určíme klíče schématu
- 2 najdeme libovolnou FZ, která porušuje 3NF/BCNF
- 3 dle této FZ bezztrátová dekompozice
- 4 výsledné relace otestujeme na 3NF/BCNF
- 5 je-li třeba pokračujeme v dekompozici

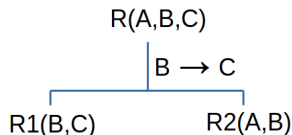
Normalizace dekompozicí – výsledek

- výsledné relace jsou ve 3NF/BCNF
- **P2** bezztrátovost zaručena
- **P1** pokrytí FZ zaručena není

Bezztrátová decomp.

Mějme schéma **R(A,B,C)**, kde A, B, C jsou disjunkttní množiny atributů a funkční závislost **B → C**.

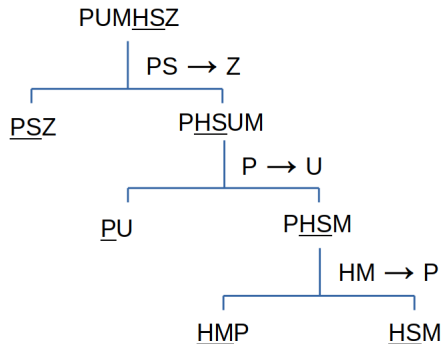
Rozložíme-li R na schémata **R1(B,C)** a **R2(A,B)**, je takto provedená dekompozice bezztrátová (loseless).



Poznámka: možných výsledků je více. Záleží na výběru FZ pro dekompozici.

Normalizace bezztrátovou dekompozicí – příklad 1

Rozvrh (Předmět, Učitel, Místnost, Hodina, Student, Známk) $R(\underline{PUMHSZ})$, $F = \{ P \rightarrow U, HM \rightarrow P, HU \rightarrow M, PS \rightarrow Z, HS \rightarrow M \}$



Výsledné relace jsou ve **3NF** (zde dokonce v **BCNF**).

Schéma je **normalizované**, tj: v relacích, které nahradily původní jednu, jsme snížili redundanci a vyloučili aktualizací anomálie.

Jedná se o schéma

$S_{III} = \{PU, HSM, PSZ, HMP\}$

ze slide **Kvalita schématu – příklad**.

Cenou je, že občas budeme v dotazech spojovat.

Nesplněna podmínka pokrytí FZ (**P1**). $HU \rightarrow M$ nelze z

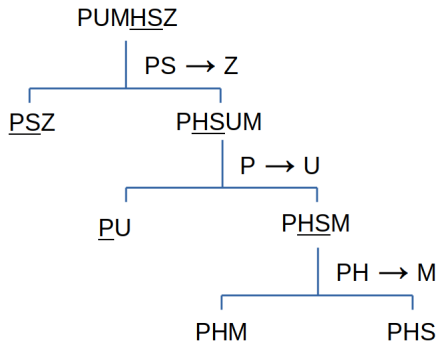
$G = \{ P \rightarrow U, HM \rightarrow P, PS \rightarrow Z, HS \rightarrow M \}$ odvodit ($F^+ \neq (\cup_{i=1}^n F_i)^+$).

Splněna podmínka bezztrátového spojení (**P2**) ($R = *_{i=1}^n R_i[A_i]$).

Normalizace bezztrátovou dekompozicí – příklad 2

Rozvrh (Předmět, Učitel, Místnost, Hodina, Student, Známk)

$R(\underline{PUMHSZ})$, $F = \{ P \rightarrow U, \textcolor{red}{HM} \rightarrow \textcolor{red}{P}, \textcolor{red}{HU} \rightarrow \textcolor{red}{M}, PS \rightarrow Z, \textcolor{red}{HS} \rightarrow \textcolor{red}{M} \}$



FZ $\textcolor{blue}{PH} \rightarrow \textcolor{blue}{M}$ lze pro dekompozici použít, patří totiž do F^+ (z F je **odvoditelná**).

Výsledek je ve **3NF** i **BCNF**.

Schema je tedy **normalizované**.

Jedná se o schéma

$S_{IV} = \{ \underline{PU}, \underline{HMP}, \underline{PSZ}, \underline{PHS} \}$

ze slide **Kvalita schématu – příklad**.

Cenou je, že občas budeme v dotazech spojovat.

Nesplněna (P1). $\{ \textcolor{red}{HM} \rightarrow \textcolor{red}{P}, \textcolor{red}{HU} \rightarrow \textcolor{red}{M}, \textcolor{red}{HS} \rightarrow \textcolor{red}{M} \}$ nelze z

$G = \{ P \rightarrow U, PH \rightarrow M, PS \rightarrow Z \}$ odvodit ($F^+ \neq (\bigcup_{i=1}^n F_i)^+$).

Splněna podmínka bezztrátového spojení (**P2**) ($R = *_{i=1}^n R_i[A_i]$).

Normalizace dekompozicí – shrnutí

- Normalizace bezztrátovou dekompozicí:
 - ▶ zaručuje bezztrátové spojení (podmínka **P2**),
 - ▶ **nezaručuje** pokrytí množiny funkčních závislostí (podmínka **P1**).
- Existuje metoda (algoritmus), která zaručuje platnost obou podmínek – **Bernsteinův algoritmus syntézy**. Je však náročnější na výpočet, vyžaduje spočítat tzv. „minimální pokrytí“.
- Nebudu ho přednášet ani vyžadovat.
- Protože normalizace bezztrátovou dekompozicí v praktických případech postačuje dost dobře.

Shrnutí

- Cílem normalizace je minimalizovat redundanci dat a aktualizací anomálie. To je důležité zejména u „write intensive“ (Insert/Update/Delete) databází. U „read only“ databází se někdy vyplatí denormalizovat – zavést redundanci kvůli snížení odezvy na (komplexní) dotazy. Normalizovaná databáze má více tabulek – při komplexních dotazech musíme více spojovat, což může být dražší.
- Kritériem jsou normální formy – v praxi se snažíme o 3NF nebo BCNF.
- Pomůckou jsou funkční závislosti a Armstrongova pravidla a definice a postupy:
 - ▶ klíč schématu
 - ▶ nalezení tranzitivního úzávěru množiny atributů
 - ▶ bezztrátová dekompozice
- **Praktický přístup k normalizaci: článek od Joe Celka (anglicky)**