

5. Normalizace návrhu databáze

Ing. Vladimír Bartík, Ph.D.
RNDr. Marek Rychlý, Ph.D.



Osnova

5.1. Úvod do teorie závislostí

5.1.1. Funkční závislost

5.2. Využití teorie závislostí při návrhu databáze

5.2.1. Normalizace

5.2.2. Normální formy

5.2.3. Obecný postup odstranění částečných a tranzitivních závislostí

5.2.4. Příklad na normalizaci tabulky

5.1. Úvod do teorie závislostí

- Normalizace schématu databáze
 - druhá významná teoretická podpora relačních databází
 - je založena na třech typech závislostí mezi atributy relace:
 - **funkční závislosti**
 - vícehodnotové závislosti (multizávislosti)
 - závislosti na spojení

5.1.1. Funkční závislost

- Hodnota atributu relace určuje jednoznačně hodnotu jiného atributu téže relace
- Zapisujeme $X \rightarrow Y$
- **Vyplývá z významu atributů, představuje integritní omezení**

Př.) Klient(r_číslo, jméno, ulice, město)

Hodnota rodného čísla jednoznačně určuje hodnoty ostatních atributů.

$r_číslo \rightarrow jméno$

$\rightarrow ulice$

$\rightarrow město$

$r_číslo \rightarrow (jméno, ulice, město)$

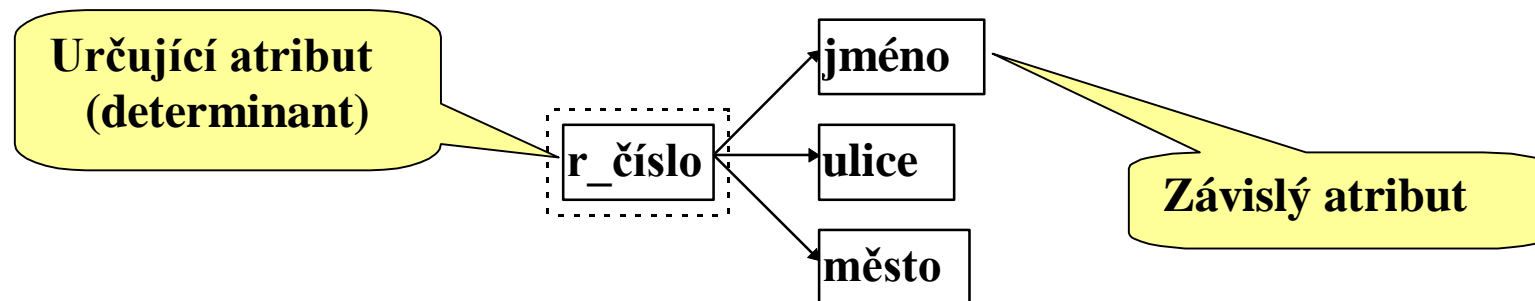
$jméno \nrightarrow (ulice, město)$

5.1.1. Funkční závislost

Nechť X a Y jsou atributy relace R . Řekneme, že **Y funkčně závisí na X** , zapisujeme **$X \rightarrow Y$** ,

právě když pro libovolné dvě n -tice t_1 a t_2 každého přípustného stavu relace R platí, že je-li x_1 , resp. y_1 hodnota atributu X , resp. Y v n -tici t_1 a x_2 , resp. y_2 hodnota atributu X , resp. Y v n -tici t_2 a $x_1 = x_2$, potom i $y_1 = y_2$.

- Diagram funkčních závislostí



Praktický důsledek (integritní omezení): **Opakuje-li se v relaci stejná hodnota determinantu, musí se opakovat i odpovídající stejné hodnoty závislého atributu.**

5.1.1. Funkční závislost

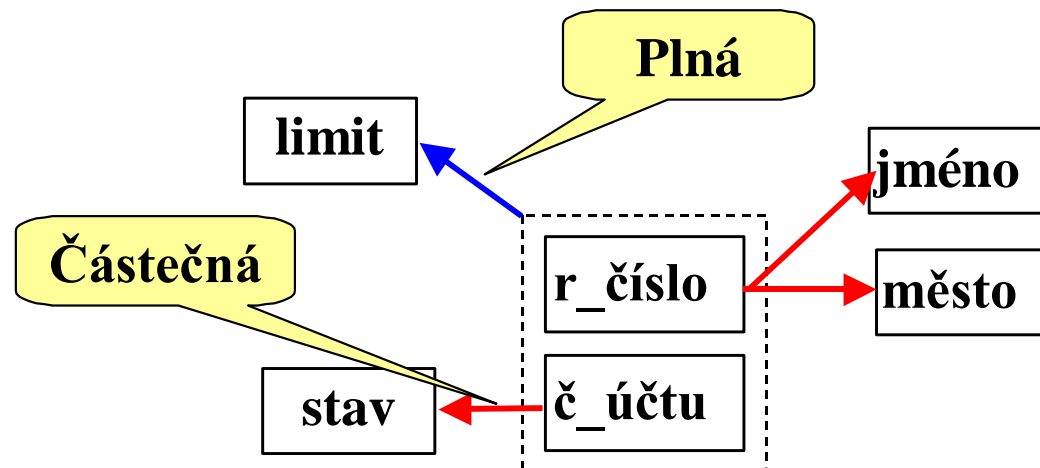
- Triviální funkční závislost

$X \rightarrow Y$ platí pro každý atribut $Y \subseteq X$.

- Plná funkční závislost

- atribut je funkčně závislý na celém složeném atributu a není funkčně závislý jen na některé jeho části

Př) Disponuje(r_číslo, jméno, město, č_úctu, stav, limit)



5.1.1. Funkční závislost

- Plná funkční závislost (pokračování)

Nechť X a Y jsou atributy relace R . Řekneme, že atribut Y je *plně funkčně závislý* na atributu X , právě když je funkčně závislý na X a není funkčně závislý na žádném atributu $Z \subset X$.

- Praktický důsledek: je-li kandidátní klíč relace složený, stejné hodnoty složek se mohou opakovat \Rightarrow musí se opakovat i hodnoty atributů, které jsou částečně (ne plně) závislé.

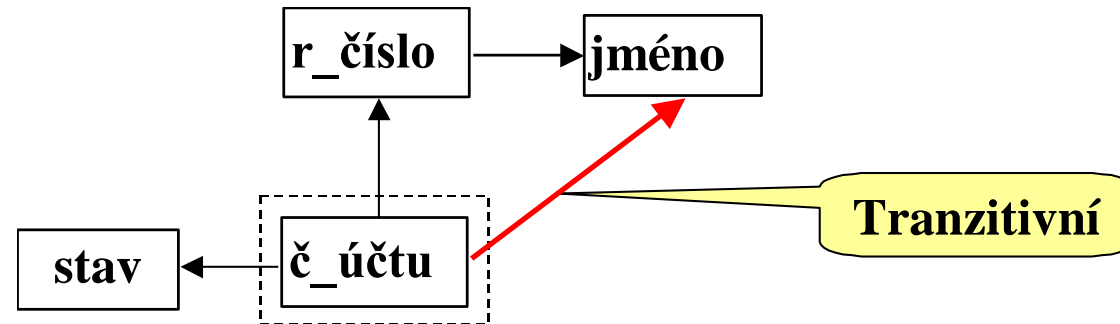
č_úctu	r_číslo	stav	jméno	limit	město
100	600528/0275	100000	Novák	10000	Praha
100	581015/9327	100000	Malá	3000	Brno
130	600528/0275	50000	Novák	5000	Praha
150	450205/3419	150000	Veselý	5000	Ostrava

5.1.1. Funkční závislost

- Tranzitivní závislost

- atribut je funkčně závislý na jiném funkčně závislém atributu

Př) Účet(č_úctu, stav, r_číslo, jméno)



Nechť X a Y jsou atributy relace R a nechť platí $X \rightarrow Y$, avšak neplatí $Y \rightarrow X$, a nechť existuje atribut Z relace R , který není v X , ani Y , a platí $Y \rightarrow Z$. Potom říkáme, že Z je **tranzitivně závislý** na X .

5.1.1. Funkční závislost

- Tranzitivní závislost (pokračování)

Praktický důsledek: **existuje-li funkční závislost na atributu, který není kandidátním klíčem, hodnota se může opakovat \Rightarrow musí se opakovat i hodnoty závislého atributu.**

Př.)

č_účtu	stav	r_číslo	jméno
100	100000	600528/0275	Novák
120	135000	581015/9327	Malá
130	50000	600528/0275	Novák
150	150000	450205/3419	Veselý

5.2. Využití teorie závislostí při návrhu databáze

- Proces návrhu založený na teorii závislostí se nazývá **normalizace**.
- Postup návrhu
 - seznam atributů (univerzální relace) → postupná dekompozice na schéma v dostatečně vysoké **normální formě**
- Praktický postup
 - datový model (ER diagram) → transformace na schéma relační databáze → zjemnění využitím normalizace resp. normalizace ER modelu před transformací.

5.2. Využití teorie závislostí při návrhu databáze

- Hlavní problémy špatného návrhu
 - opakující se informace (redundance)
 - nemožnost reprezentovat určitou informaci
 - ztráta informace
 - složitá kontrola integritních omezení

Př) Účet1(č_úctu, r_číslo, stav, pobočka, jmění)

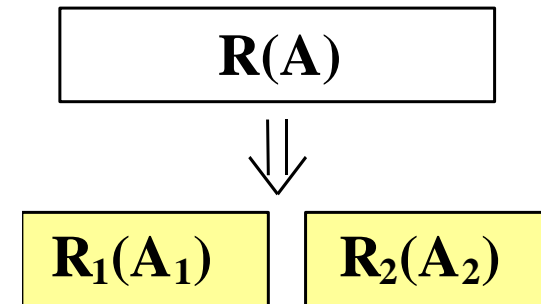
Předpokládejme, že s účtem může disponovat více osob.

č_úctu	r_číslo	stav	pobočka	jmění
100	600528/0275	100000	Jánská	10000000
100	581015/9327	100000	Jánská	10000000
130	600528/0275	50000	Palackého	5000000
150	450205/3419	150000	Palackého	5000000

5.2.1. Normalizace

- Postupná transformace tabulky do vhodnějšího tvaru (postupná dekompozice).

Nechť $R(A)$ je schéma relace R . Množina schémat $\{R_1(A_1), R_2(A_2), \dots, R_n(A_n)\}$ je **dekompozicí schématu** $R(A)$, jestliže $A = A_1 \cup A_2 \cup \dots \cup A_n$.



- Žádoucí vlastnosti dekompozice:
 - bezeztrátovost při zpětném spojení
 - zachování závislostí
 - odstranění opakování informace (redundance)
- **Bezeztrátová dekompozice (Lossless-Join / Nonloss decomp.)**
 - spojení tabulek, které vzniknou dekompozicí musí dát přesně původní tabulku

5.2.1. Normalizace

Př) Účet1(č_úctu, r_číslo,
stav, pobočka, jmění)
– viz předchozí

č_úctu	r_číslo	stav	pobočka	jmění
100	600528/0275	100000	Jánská	10000000
100	581015/9327	100000	Jánská	10000000
130	600528/0275	50000	Palackého	5000000
150	450205/3419	150000	Palackého	5000000

Účet_v_pob

č_úctu	r_číslo	pobočka	jmění
100	600528/0275	Jánská	10000000
100	581015/9327	Jánská	10000000
130	600528/0275	Palackého	5000000
150	450205/3419	Palackého	5000000

Stav

r_číslo	stav
600528/0275	100000
581015/9327	100000
600528/0275	50000
450205/3419	150000

Tabulka Účet1 po
zpětném spojení

č_úctu	r_číslo	stav	pobočka	jmění
100	600528/0275	100000	Jánská	10000000
100	600528/0275	50000	Jánská	10000000
100	581015/9327	100000	Jánská	10000000
130	600528/0275	50000	Palackého	5000000
130	600528/0275	100000	Palackého	5000000
150	450205/3419	150000	Palackého	5000000

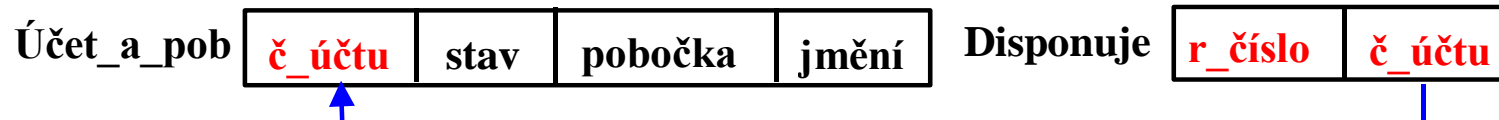
5.2.1. Normalizace

Podmínka bezeztrátové dekompozice:

Pro dekompozici se schémata $R1(A1)$ a $R2(A2)$ musí platit:

$A1 \cap A2 \rightarrow A1$ nebo $A1 \cap A2 \rightarrow A2$, tj. společný atribut musí být kandidátním klíčem alespoň jedné z tabulek.

Př.)



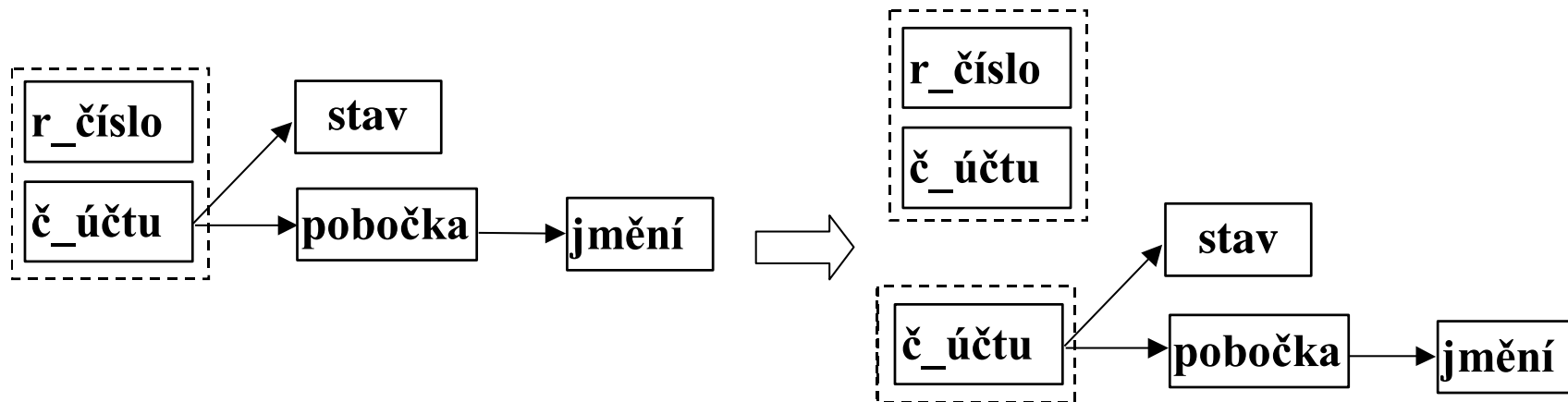
č_úctu	Stav	pobočka	jmění
100	100000	Jánská	10000000
130	50000	Palackého	5000000
150	150000	Palackého	5000000

č_úctu	r_číslo
100	600528/0275
100	581015/9327
130	600528/0275
150	450205/3419

5.2.1. Normalizace

- **Zachování závislostí**

- všechny původní závislosti musí být zachovány a snadno kontrolovatelné (v rámci jedné tabulky)

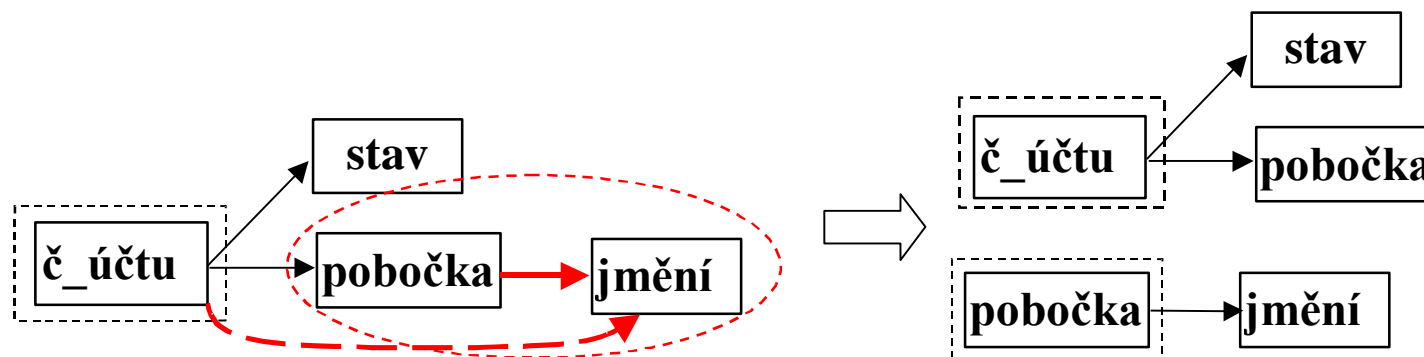


5.2.1. Normalizace

- Odstranění opakování informace

Př.)

č účtu	stav	pobočka	jmění
100	100000	Jánská	10000000
130	50000	Palackého	5000000
150	150000	Palackého	5000000



Účet

č účtu	stav	pobočka
100	100000	Jánská
130	50000	Palackého
150	150000	Palackého

Pobočka

pobočka	jmění
Jánská	10000000
Palackého	5000000

5.2.1. Normalizace

- **Boyce-Coddova normální forma (BCNF)**
 - odstraňuje opakování informace
 - všechny netriviální funkční závislosti jsou dány závislostí na kandidátních klících
 - ne každá dekompozice do BCNF zachovává závislosti \Rightarrow potom stačí **3NF**

5.2.2. Normální formy

- Definují požadavky na vlastnosti schématu relace z pohledu závislostí mezi atributy
- Hierarchie normálních forem (Codd: 1NF až 3NF, BCNF, 4NF, 5NF), tj. n-tá normální forma musí splňovat podmínky (n-1) normální formy a něco navíc.
- Požadavky na návrh založený na normalizaci:
 - bezeztrátovost dekompozice
 - zachování závislostí
 - dosažení minimálně BCNF, resp. 3NF
- Terminologie:
 - **Klíčový atribut** – atribut, který je součástí kandidátního klíče, ostatní **neklíčové**.
 - **Superklíč** – nadmnožina kandidátního klíče ($CK \subseteq SK$).

5.2.2. Normální formy

- První normální forma (1NF)

Relace je v *první normální formě*, právě když **všechny** její **jednoduché domény** obsahují pouze **atomické hodnoty**.

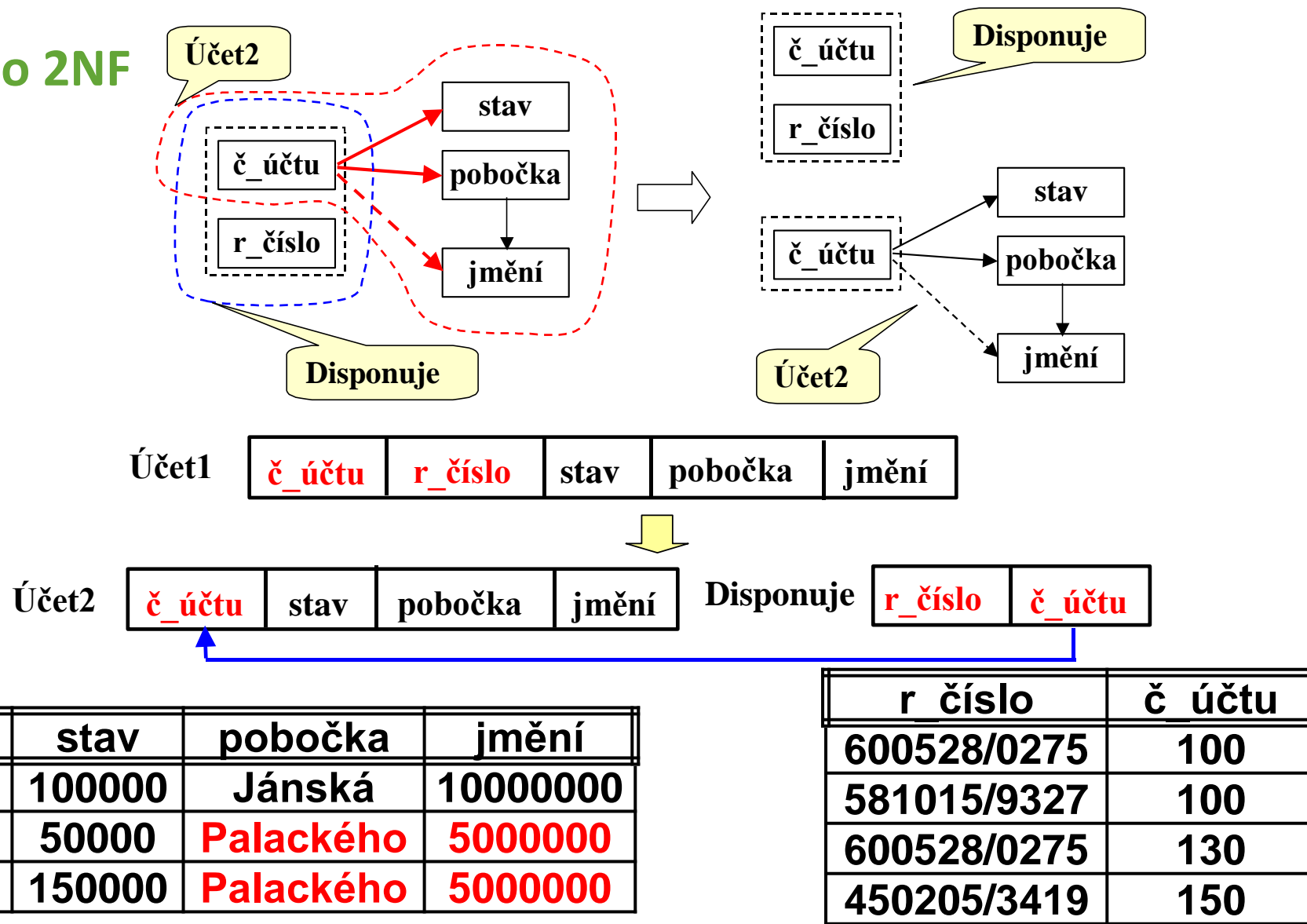
Př.) Účet1(č_úctu, r_číslo, stav, pobočka, jmění)

- Druhá normální forma (2NF)

Schéma relace je ve *druhé normální formě*, právě když je v 1NF a **každý její neklíčový atribut**, je **plně funkčně závislý** na **každém kandidátním klíči**.

5.2.2. Normální formy

Př.) Převod do 2NF

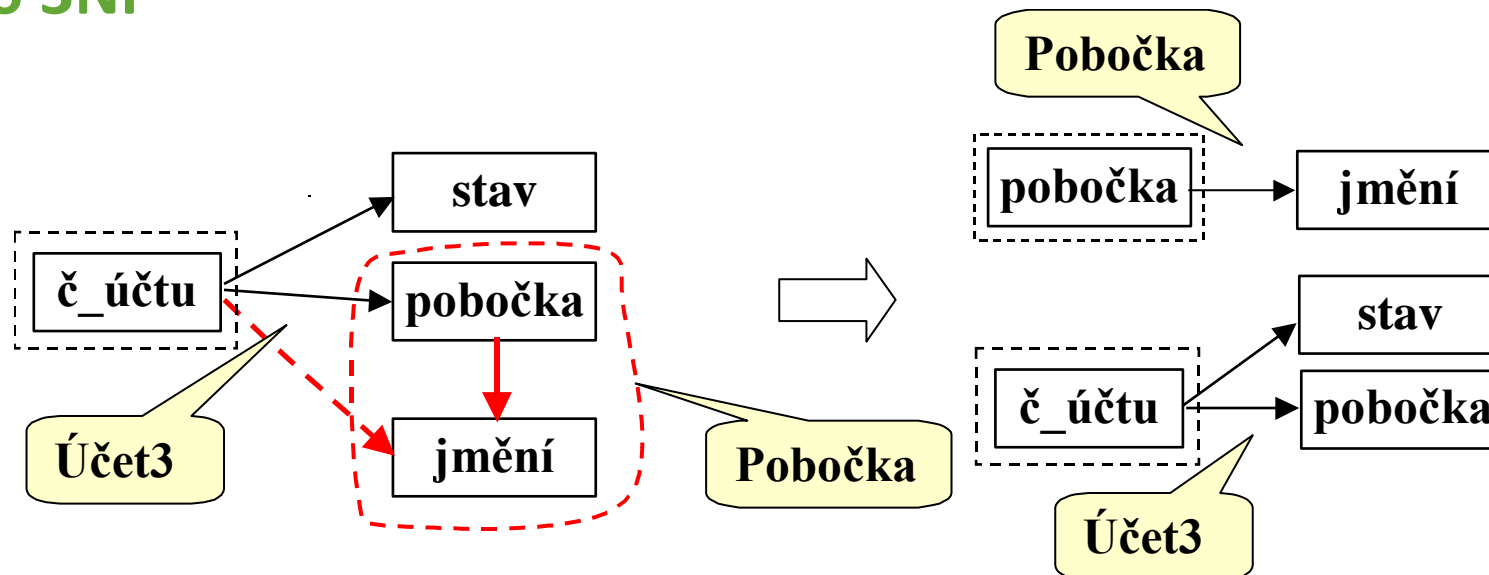


5.2.2. Normální formy

- Třetí normální forma (3NF)

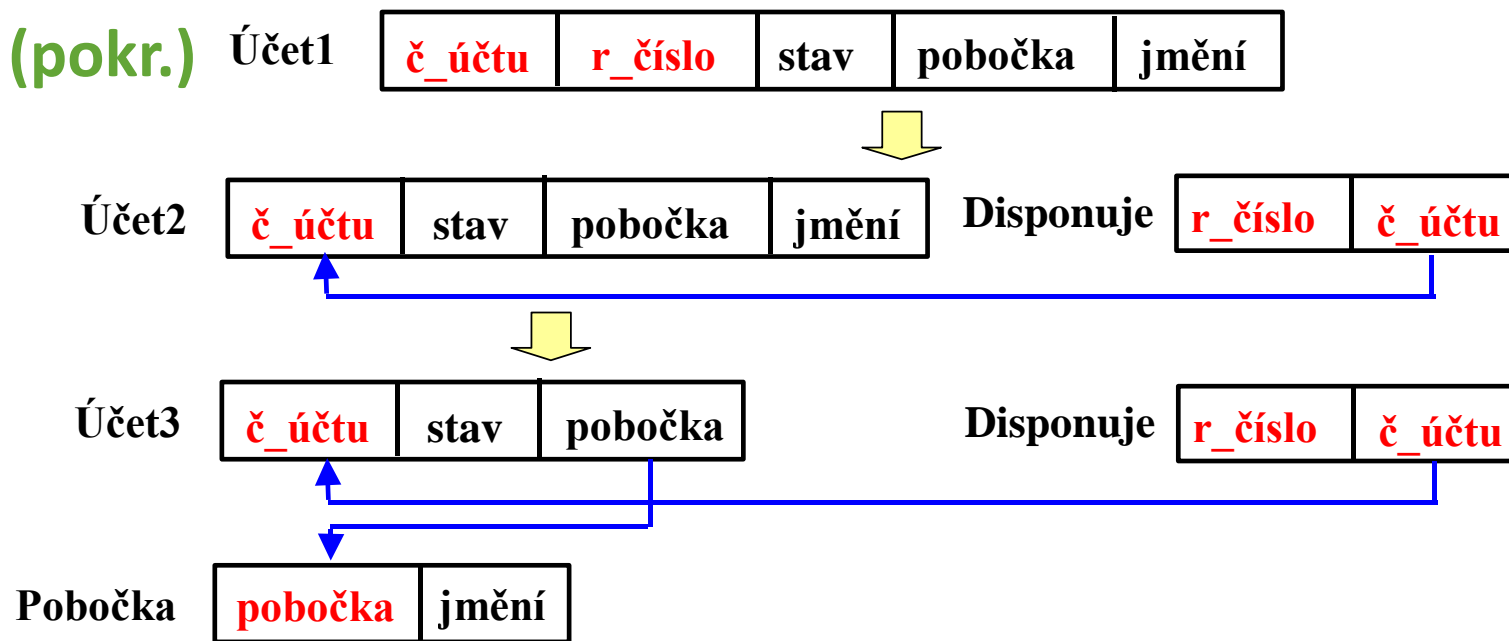
Schéma relace je ve *třetí normální formě*, právě když je ve 2NF a neexistuje žádný neklíčový atribut, který je tranzitivně závislý na některém kandidátním klíči.

Př.) Převod do 3NF



5.2.2. Normální formy

Př.) Převod do 3NF (pokr.)



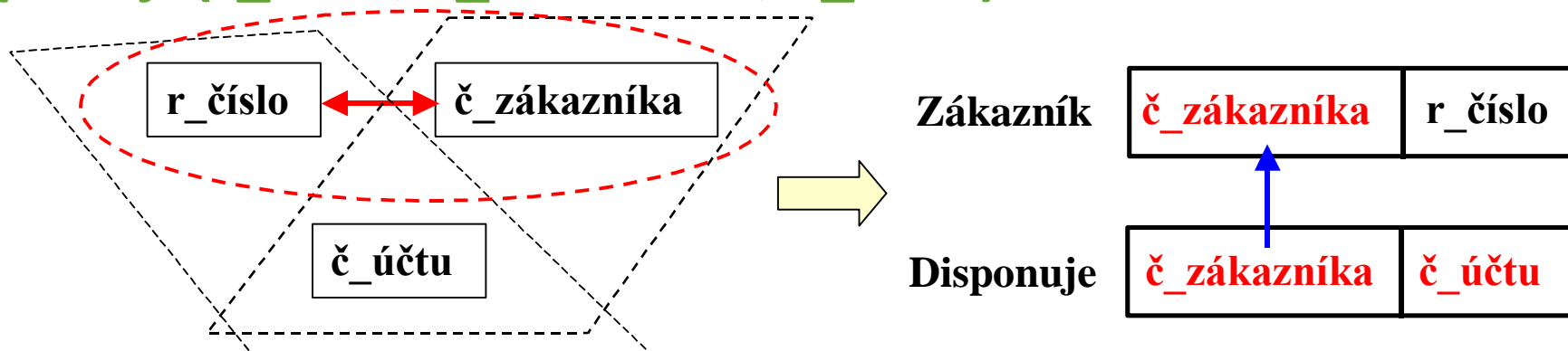
pobočka	jmění
Jánská	10000000
Palackého	5000000

č_úctu	stav	pobočka
1035853	100000	Jánská
1348427	50000	Palackého
1529054	150000	Palackého

5.2.2. Normální formy

- **Boyce-Coddova normální forma (BCNF)**

Př) Disponuje(r_číslo, č_zákazníka, č_úctu)



- může existovat několik kandidátních klíčů,
- kandidátní klíče mohou být složené,
- kandidátní klíče se mohou překrývat
- 3NF neřeší, tj. připouští **závislosti mezi klíčovými atributy**

Schéma relace je v *Boyce-Coddově normální formě*, jestliže pro každou netriviální funkční závislost $X \rightarrow Y$ je X superklíčem.

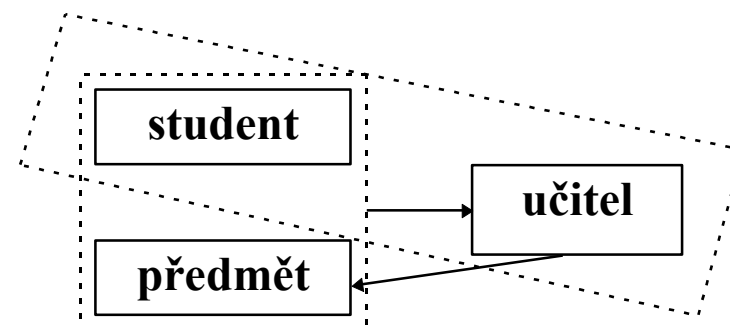
5.2.2. Normální formy

Př) Student_předmět_učitel(student, předmět, učitel)

Sémantika:

- každého studenta učí z daného předmětu jen jeden učitel, každý učitel učí jen jeden předmět
- jeden předmět může učit několik učitelů

student	předmět	učitel
Novák	matematika	Prof.Adam
Novák	fyzika	Doc.Kovář
Veselý	matematika	Prof.Adam
Veselý	fyzika	Doc.Zelený



- dekompozice Učí_předmět(učitel, předmět) a Učí_studenta(student, učitel) nezachovává závislosti v rámci relací (relace nejsou nezávislé)

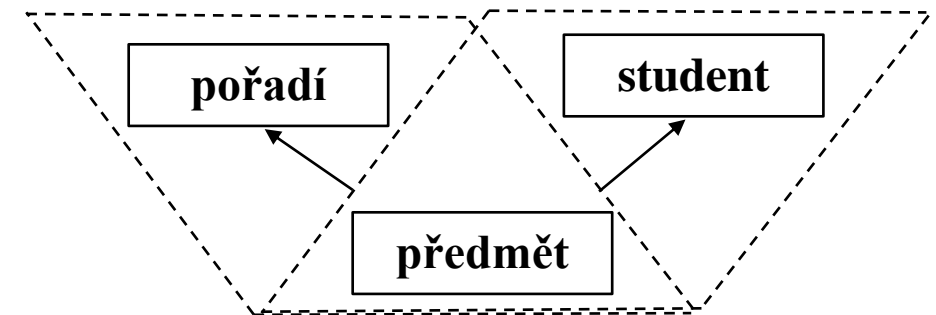
→ zachování závislostí a dosažení BCNF není vždy splnitelné

5.2.2. Normální formy

Př) Zkoušky (student , předmět, pozice)

Sémantika: Dva studenti nemohou být na stejné pozici v jednom předmětu. Předpokládáme jednoznačné jméno.

student	předmět	pozice
Novák	matematika	10
Veselý	matematika	7
Novák	fyzika	5
Veselý	fyzika	2



→ překrývající se kandidátní klíče ještě nemusí způsobovat problémy

5.2.2. Normální formy

- **Čtvrtá normální forma (4NF)**

- vymezuje vlastnosti, které musí splňovat atributy relace s ohledem na vícehodnotové závislosti.

- **Pátá normální forma (5NF)**

- vymezuje vlastnosti, které musí splňovat atributy relace s ohledem na závislosti na spojení.

- Normální forma schématu databáze

Návrh databáze je v n -té normální formě, je-li každá jeho relace (schéma) alespoň v n -té normální formě.

5.2.3. Obecný postup odstranění částečných a tranzitivních závislostí

- Převod do 2NF

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

PRIMARY KEY (A,B)

$A \rightarrow C$

$A \rightarrow D$

$A \rightarrow E$

- Převod do 3NF

R(A, B, C, D)

PRIMARY KEY (A)

$C \rightarrow D$

R1(A, C, D, E)

PRIMARY KEY (A)

R2(A, B)

PRIMARY KEY(A,B)

FOREIGN KEY(A) REFERENCES R1

R1(C, D)

PRIMARY KEY (C)

R2(A, B, C)

PRIMARY KEY(A)

FOREIGN KEY(C) REFERENCES R1

5.2.4. Příklad na normalizaci tabulky

- Zadání

- Uvažujte následující tabulku s údaji o předmětech, studentech zapsaných v jednotlivých letech, získaném počtu bodů, počtu kreditů a garantech v jednotlivých letech. Atribut *c_stud* je unikátní pro studenty, *os_c_gar* pro garanta, *zkratka* unikátní pro předměty, název předmětu nemusí být unikátní. Garant předmětu a kredity za předmět se mohou v jednotlivých letech lišit.

c_stud	jmeno_s	zkratka	nazev	kredity	body	os_c_gar	jmeno_g	ak_rok
--------	---------	---------	-------	---------	------	----------	---------	--------

1. Určete množinu kandidátních klíčů
2. Nakreslete diagram funkčních závislostí
3. Určete, v jaké normální formě relace je
4. Normalizujte tabulku tak, aby výsledné schéma bylo alespoň v BCNF

5.2.4. Příklad na normalizaci tabulky

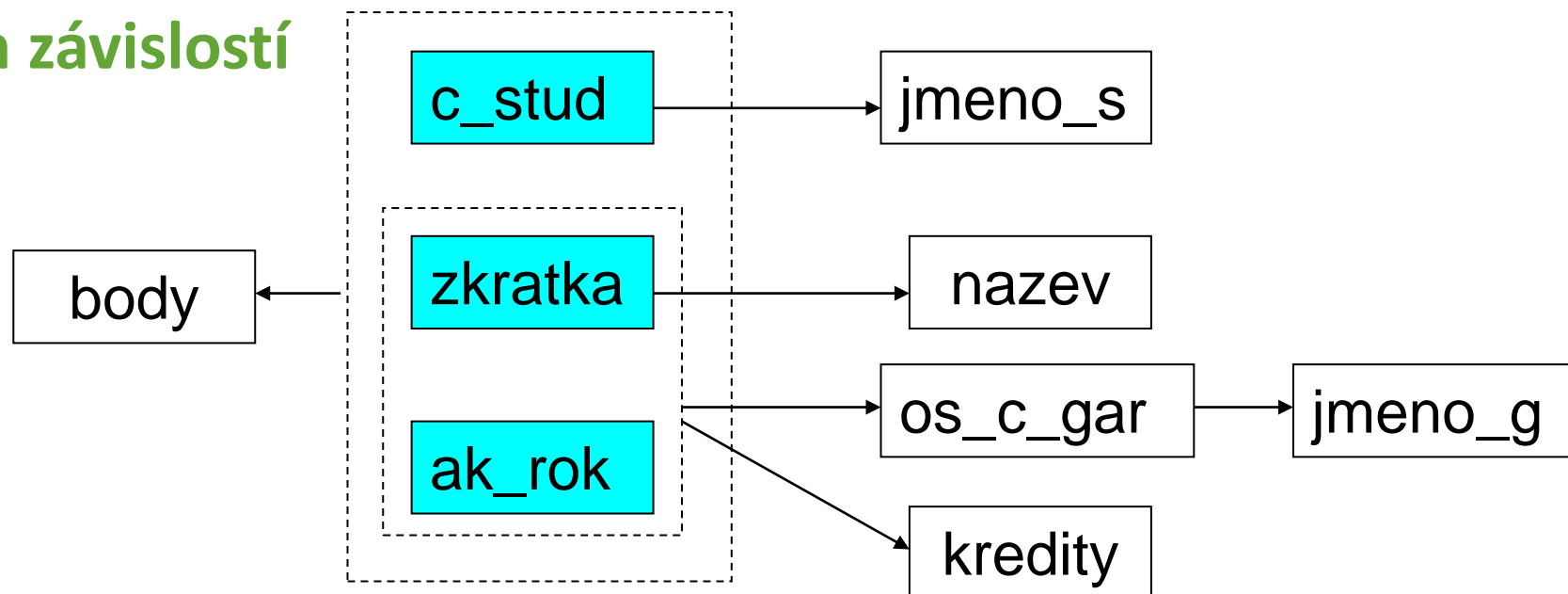
c_stud	jmeno_s	zkratka	nazev	kredity	body	os_c_gar	jmeno_g	ak_rok
--------	---------	---------	-------	---------	------	----------	---------	--------

1. Množina kandidátních klíčů

- Ověření splnění podmínky 1NF: **OK**
- Určení množiny kandidátních klíčů

KK = { (c_stud, zkratka, ak_rok) }, tj. 1 složený kandidátní klíč

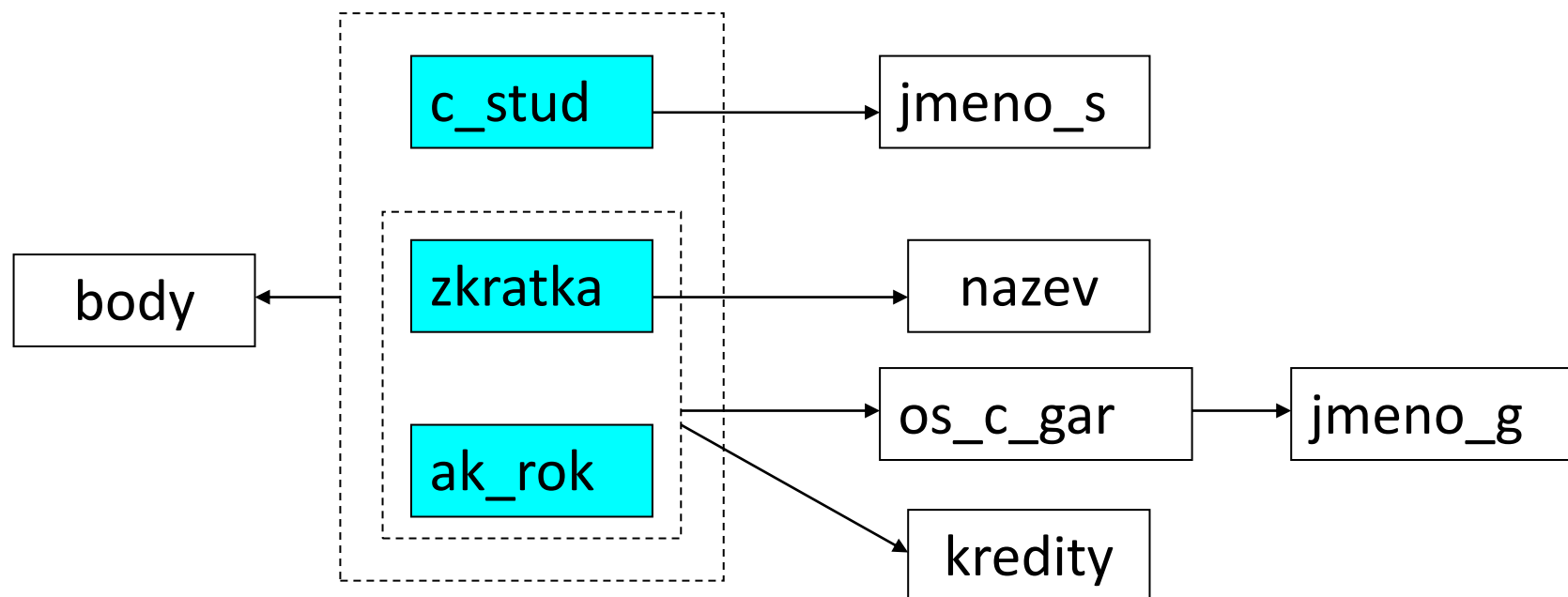
2. Diagram funkčních závislostí



5.2.4. Příklad na normalizaci tabulky

c_stud	jmeno_s	zkratka	nazev	kredity	body	os_c_gar	jmeno_g	ak_rok
--------	---------	---------	-------	---------	------	----------	---------	--------

3. Normální forma tabulky

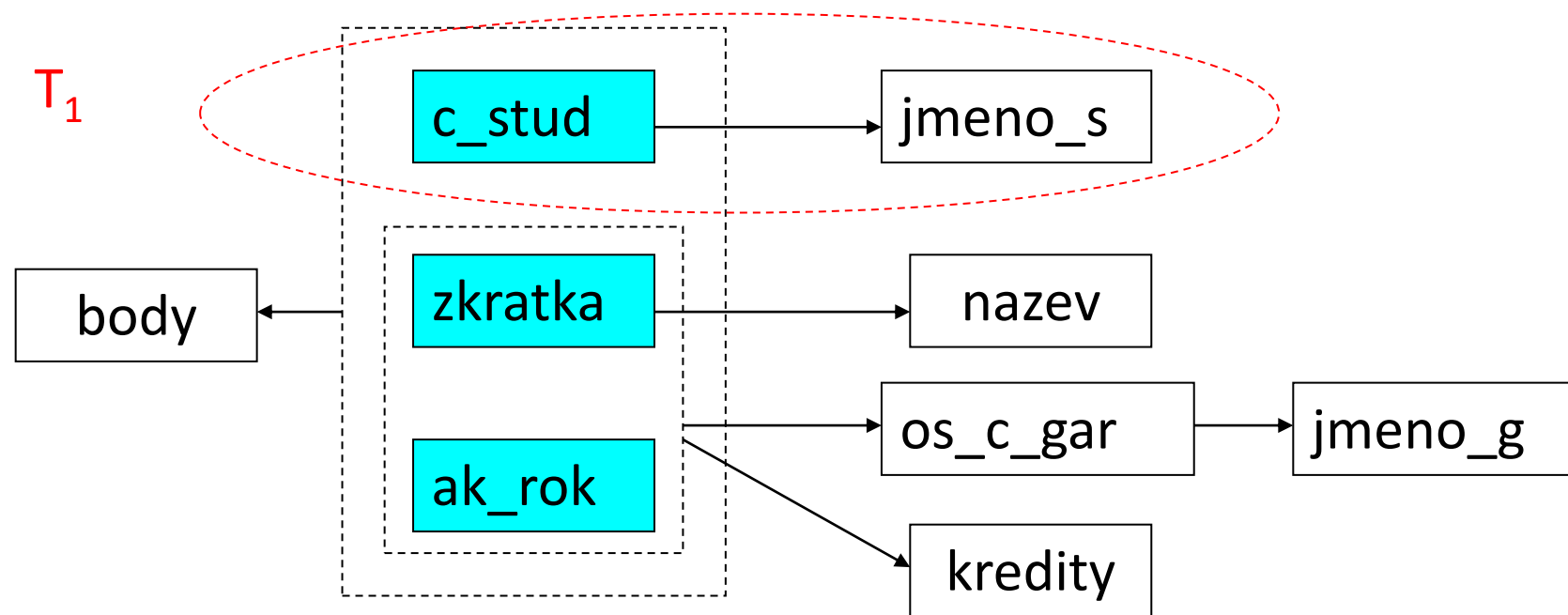


- 1NF: OK
- 2NF: nesplňuje

⇒ pouze 1NF

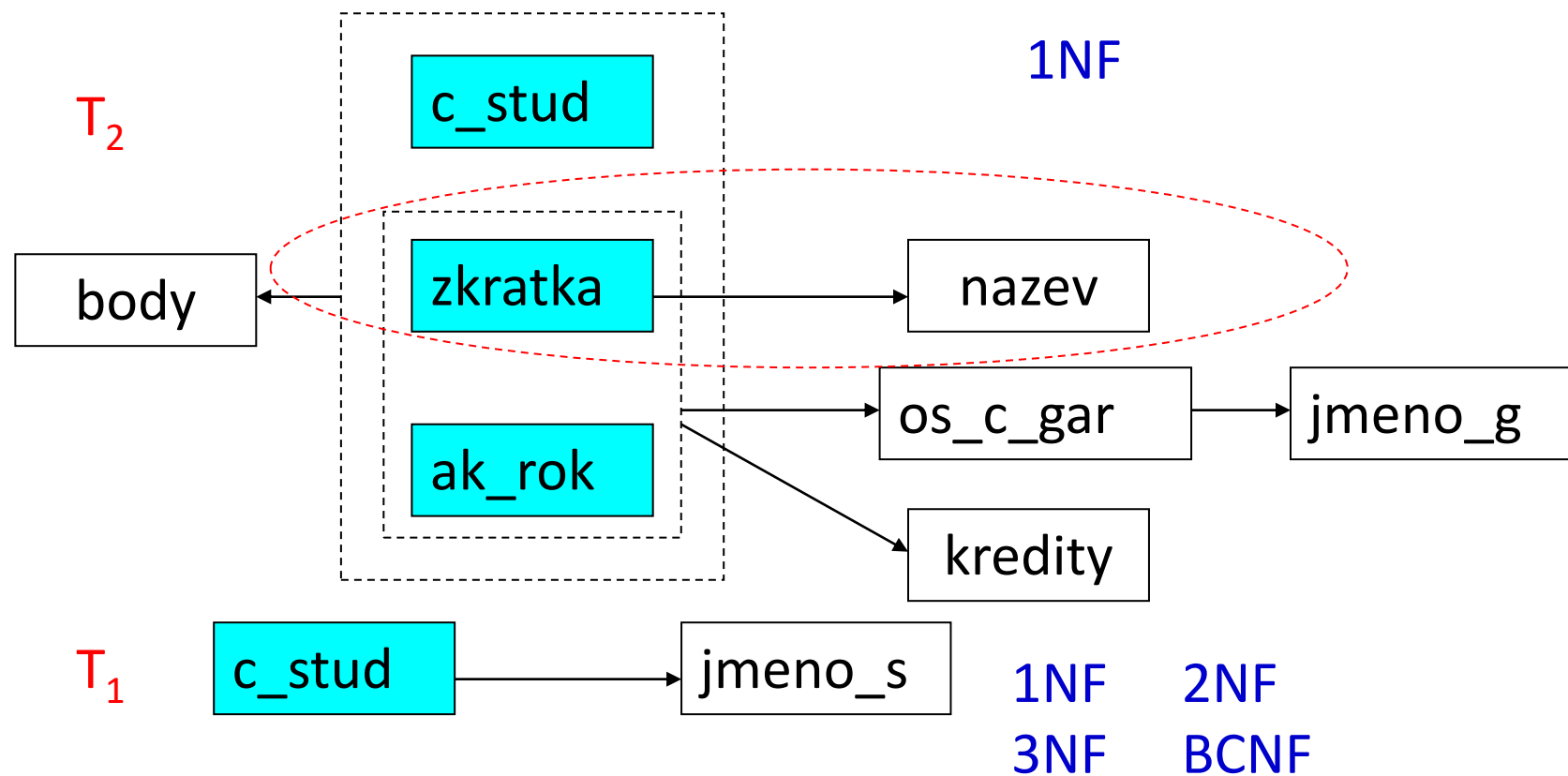
5.2.4. Příklad na normalizaci tabulky

4. Normalizace (1)



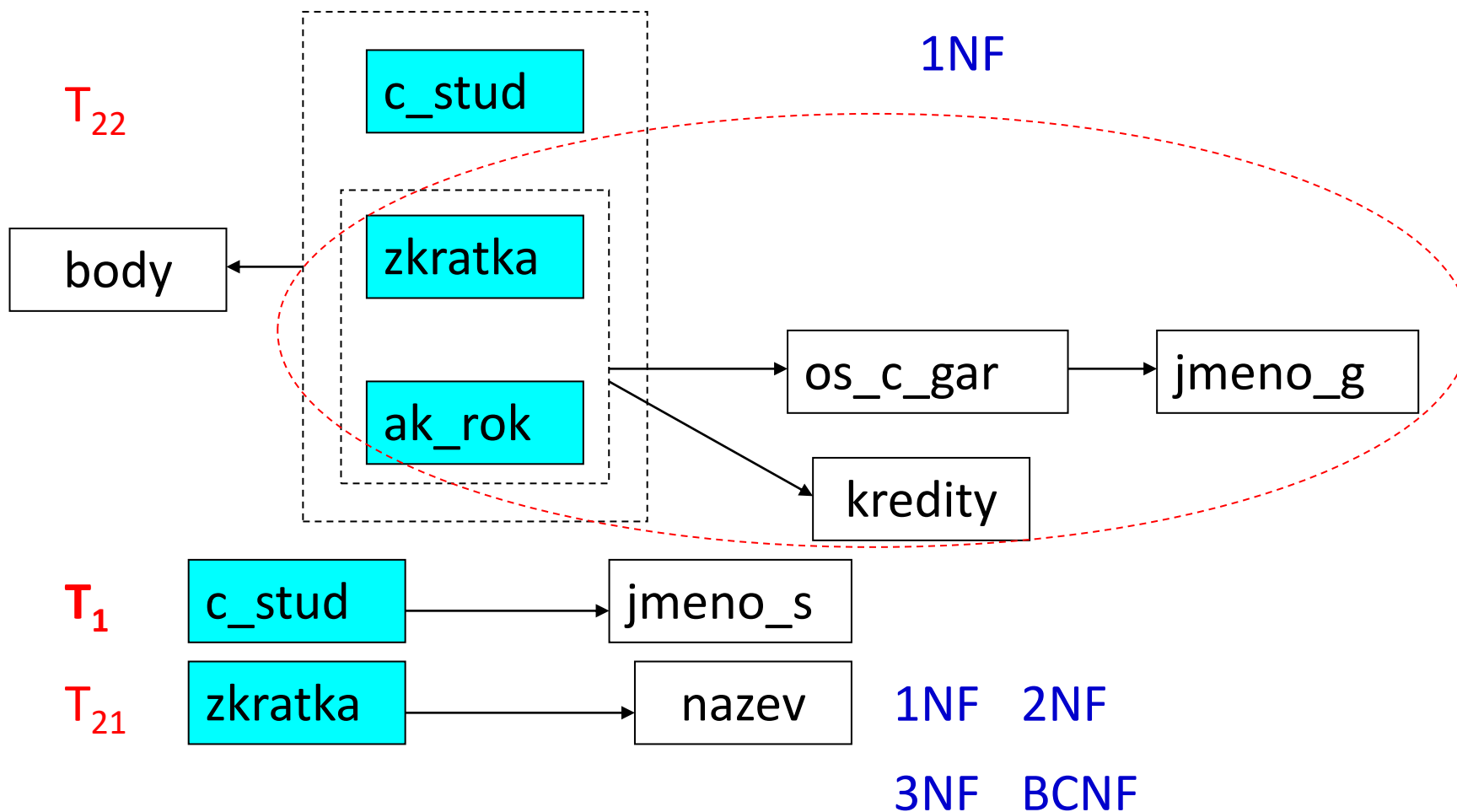
5.2.4. Příklad na normalizaci tabulky

4. Normalizace (2)



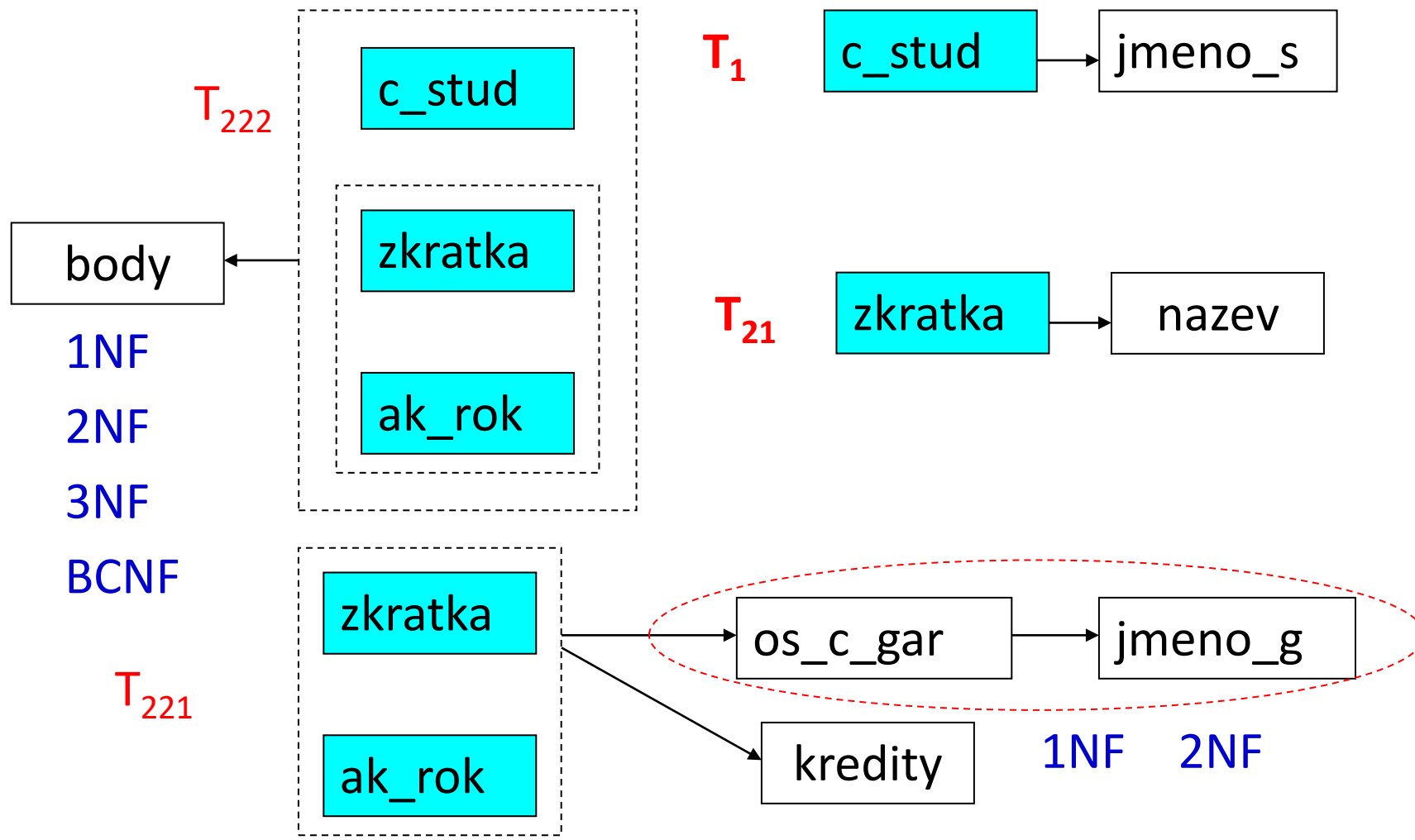
5.2.4. Příklad na normalizaci tabulky

4. Normalizace (3)



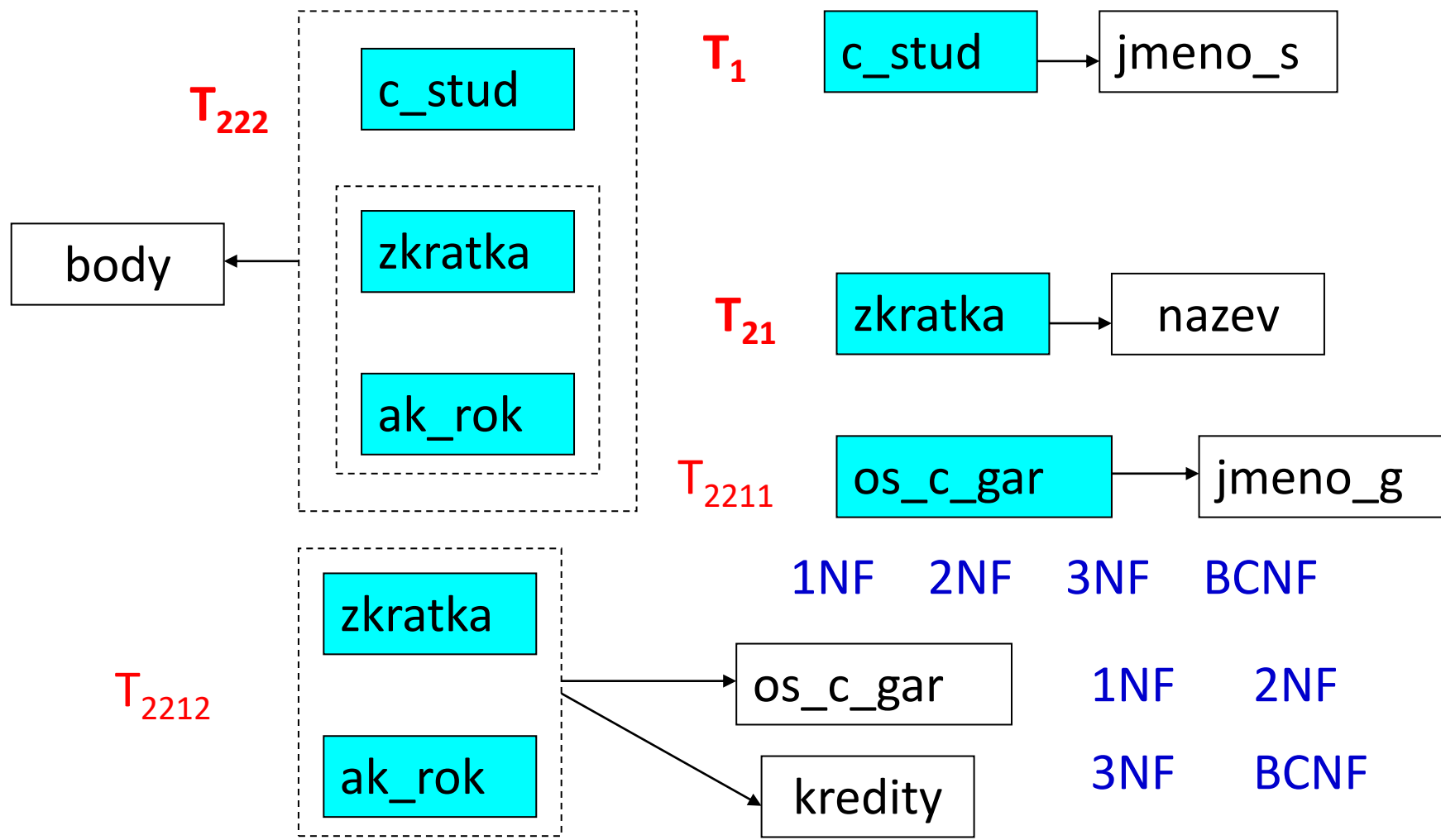
5.2.4. Příklad na normalizaci tabulky

4. Normalizace (4)



5.2.4. Příklad na normalizaci tabulky

4. Normalizace (5)



5.2.4. Příklad na normalizaci tabulky

5. Výsledné schéma

T_1 (Student)

c_studenta	jmeno
-------------------	-------

T_{222} (Zapsal)

c_stud	zkratka	ak_rok	body
---------------	----------------	---------------	------

T_{21} (Predmet)

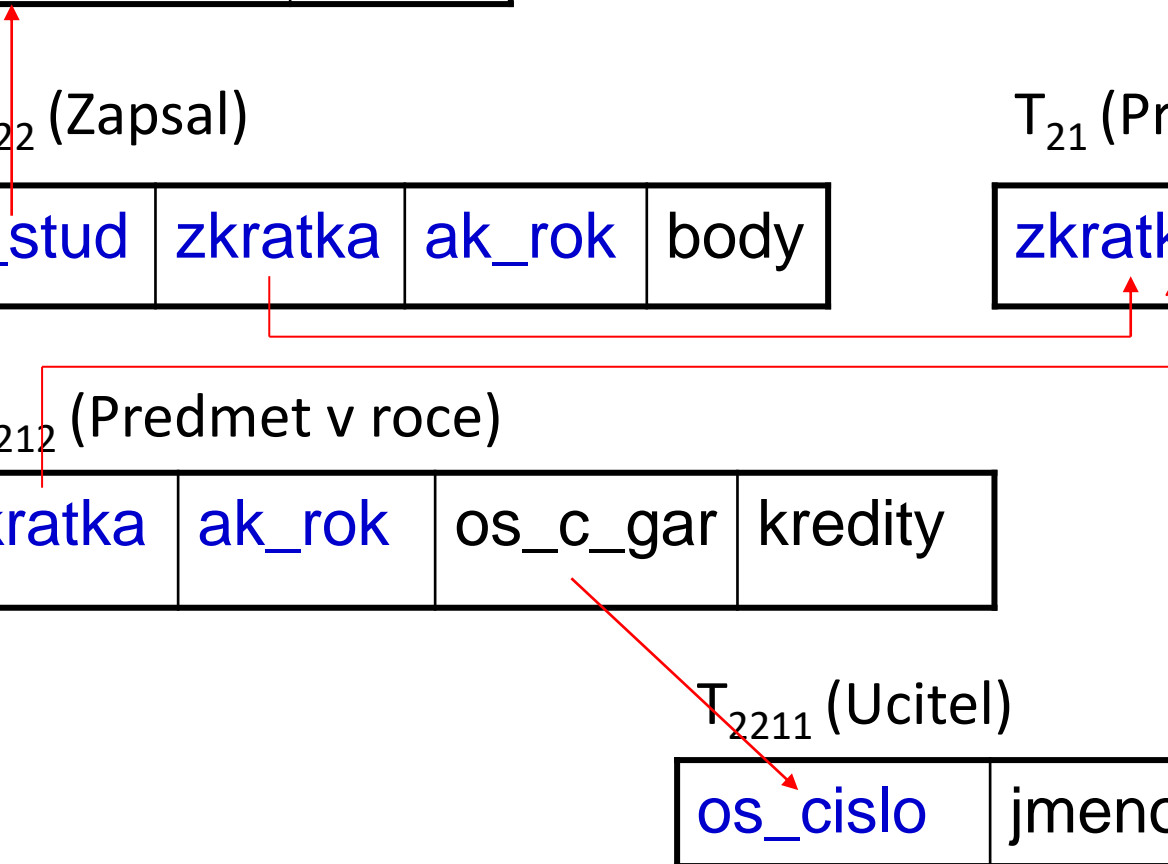
zkratka	nazev
----------------	-------

T_{2212} (Predmet v roce)

zkratka	ak_rok	os_c_gar	kredity
----------------	---------------	-----------------	---------

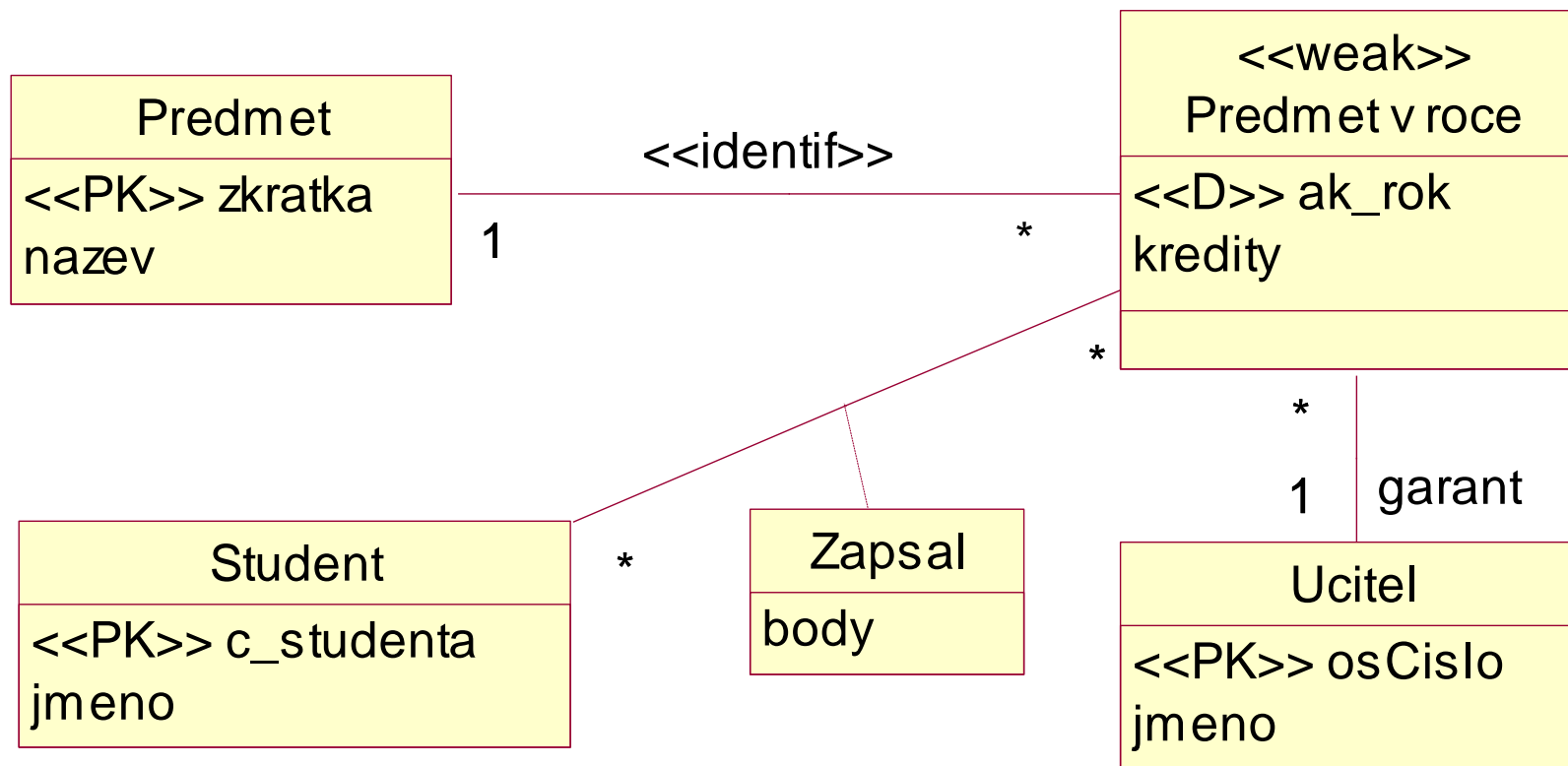
T_{2211} (Ucitel)

os_cislo	jmeno
-----------------	-------



5.2.4. Příklad na normalizaci tabulky

6. ER diagram modelující totéž



5.2.4. Příklad na normalizaci tabulky II

- **Zadání:** Uvažujte následující tabulku IS atletického klubu:

cisloA	RCA	jmenoA	sezona	disciplina	zavod	stadion	mestoS	kategorie	vykon
--------	-----	--------	--------	------------	-------	---------	--------	-----------	-------

kde **cisloA** je jednoznačné číslo atleta, který je členem klubu, **RCA** je jeho (jednoznačné) rodné číslo; **jmenoA** je jeho (nejednoznačné) jméno; **sezona** je jednoznačné označení sezony (např. 2021/2022); **disciplina** je název atletické disciplíny (např. hod diskem), **zavod** je název závodu (např. Brno Open, předpokládejte, že se pořádá pravidelně jednou za sezonu); **stadion** je stadion, na němž se daný závod každoročně koná (pro daný závod je vždy stejný); **mestoS** je město, ve kterém se daný stadion nachází; **kategorie** je věková kategorie, do níž atlet spadá (např. mladší žáci; atlet v čase mění své kategorie, ale během jedné sezony se nemění); **vykon** udává výkon atleta v určité disciplíně v jednom závodu v konkrétní sezoně (např. výkon atleta Jana Nováka na závodě Brno Open v sezoně 2022/23 ve skoku do výšky byl 215 cm). Předpokládejte, že se může atlet v rámci jednoho závodu zúčastnit i více než jedné disciplíny.

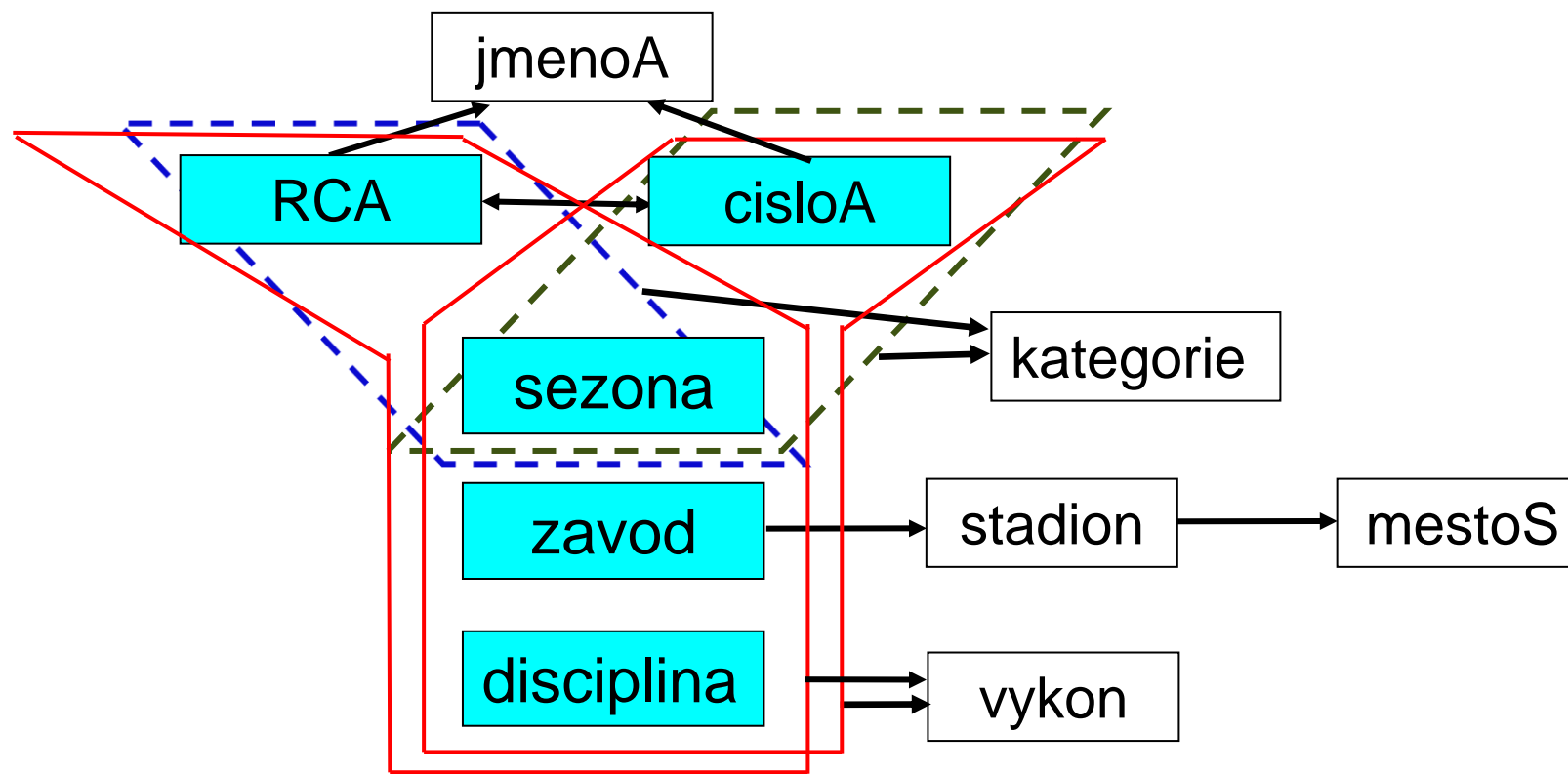
Určete množinu KK, nakreslete diagram funkčních závislostí, určete, v jaké normální formě relace je a normalizujte tabulku tak, aby výsledné schéma bylo alespoň v BCNF

5.2.4. Příklad na normalizaci tabulky II

cisloA	RCA	jmenoA	sezona	disciplina	zavod	stadion	mestoS	kategorie	vykon
--------	-----	--------	--------	------------	-------	---------	--------	-----------	-------

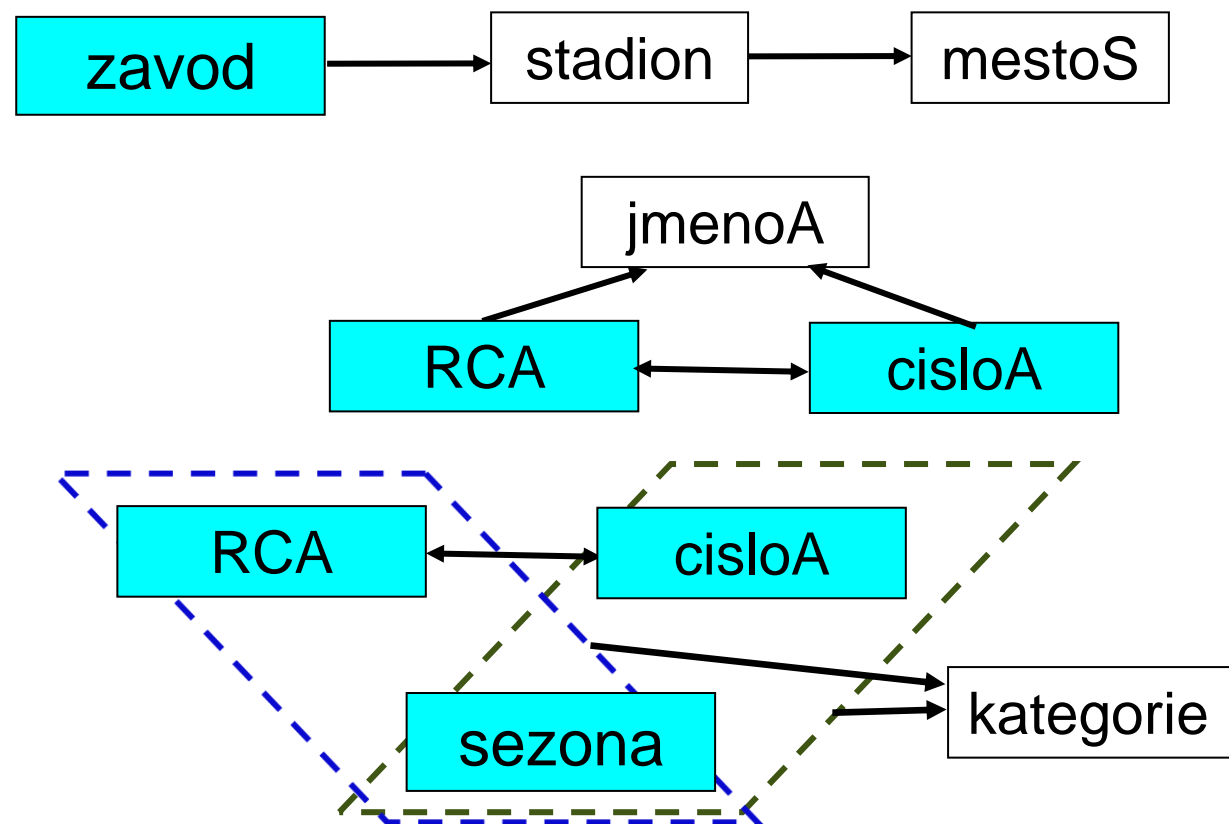
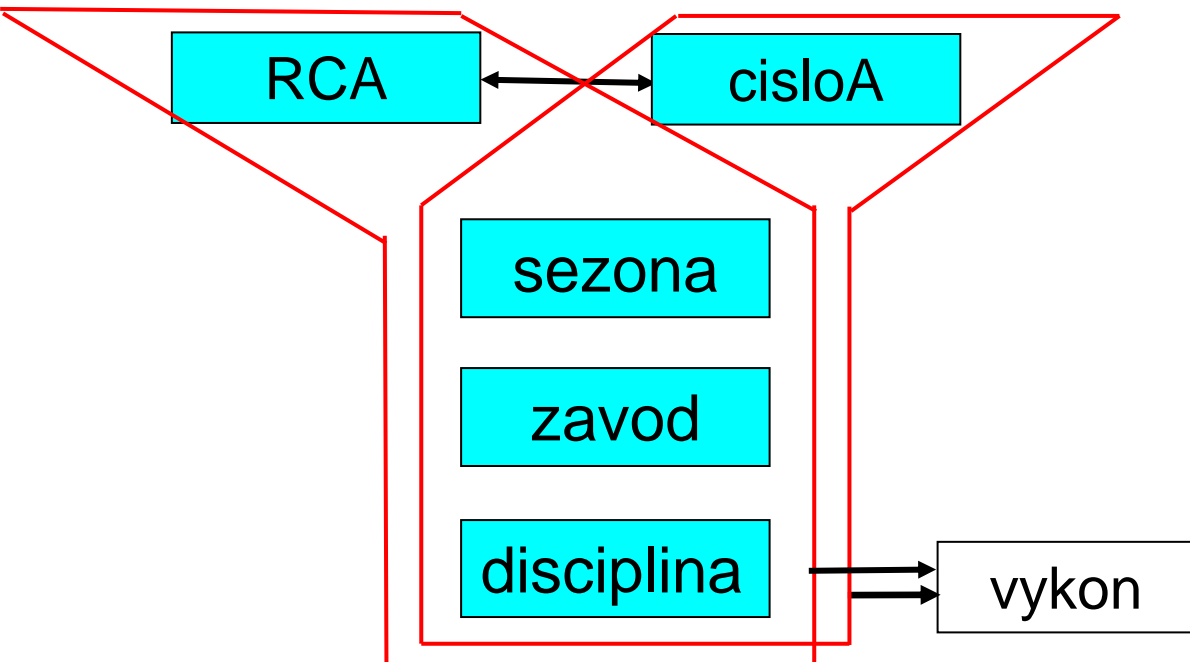
- Ověření splnění podmínky 1NF: **OK**
- Určení množiny kandidátních klíčů:

KK = { (RCA, sezona, zavod, disciplina), (cisloA, sezona, zavod, disciplina)}, tj. 2 složené KK



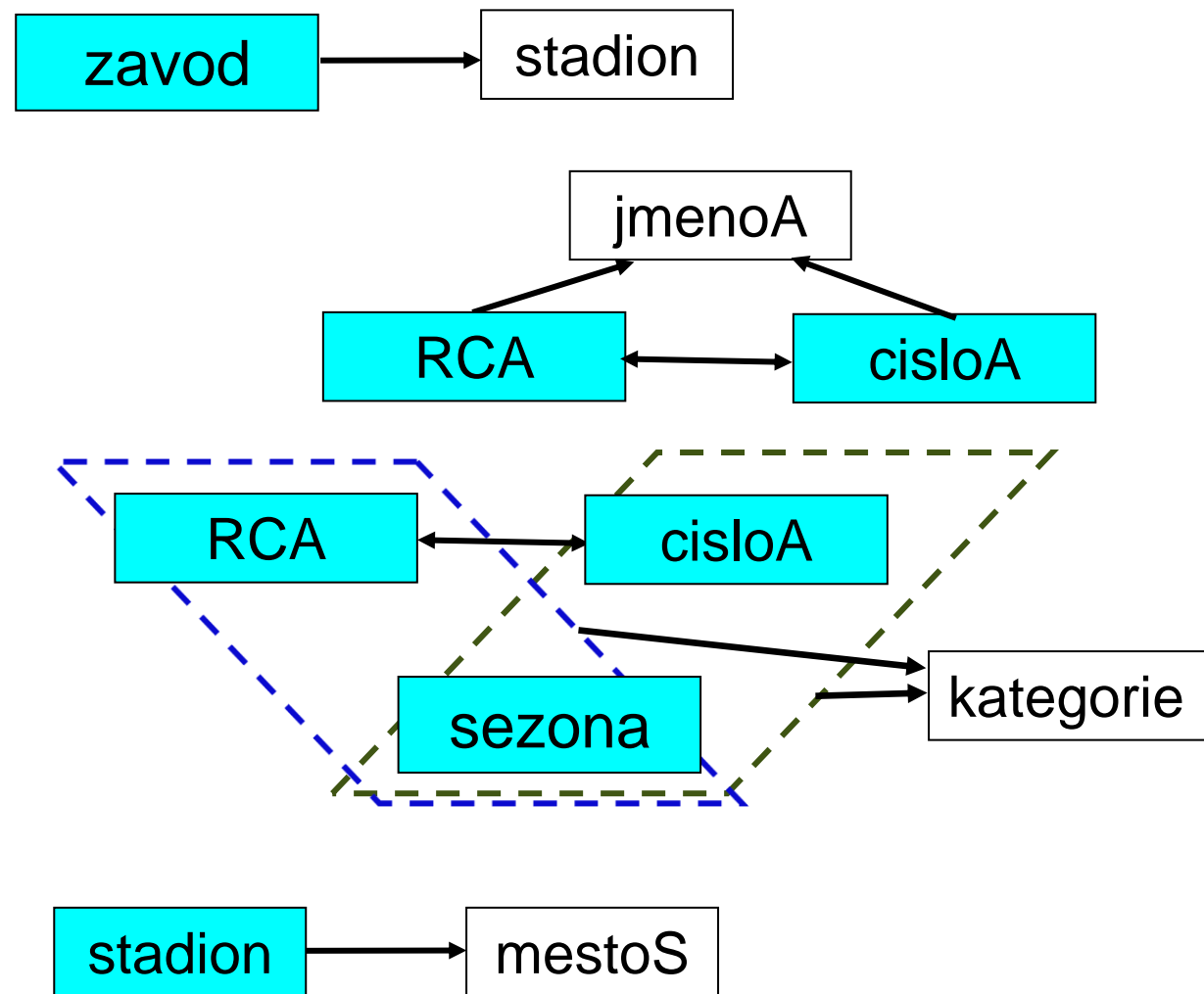
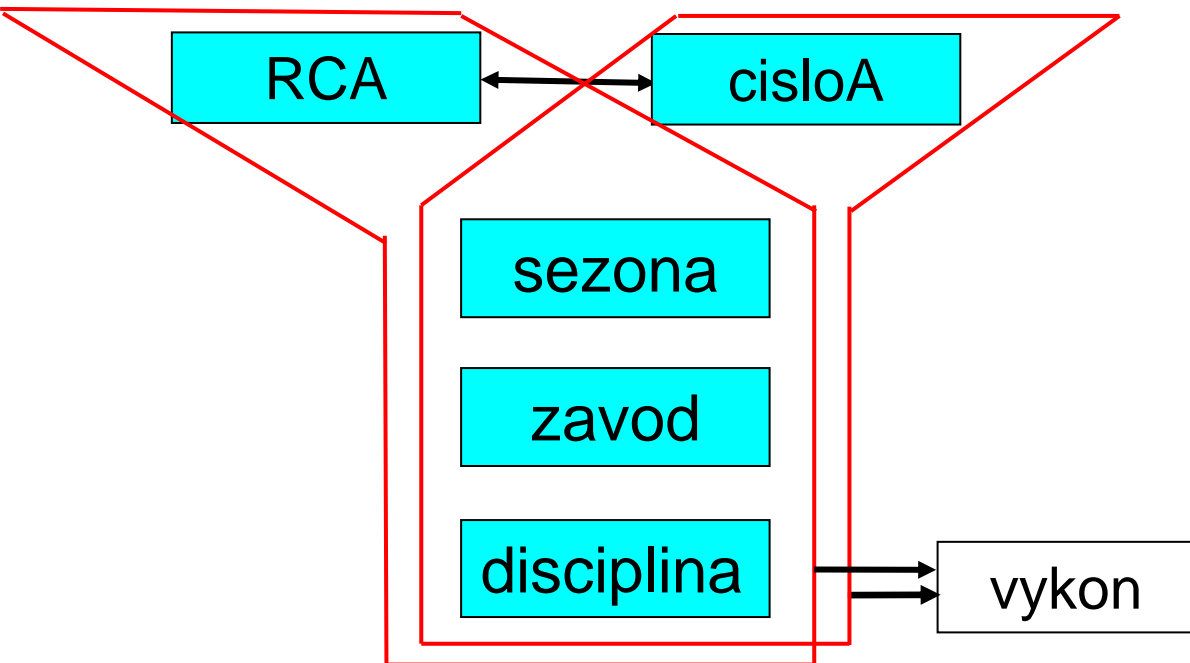
5.2.4. Příklad na normalizaci tabulky II

▫ Normalizace do 2NF



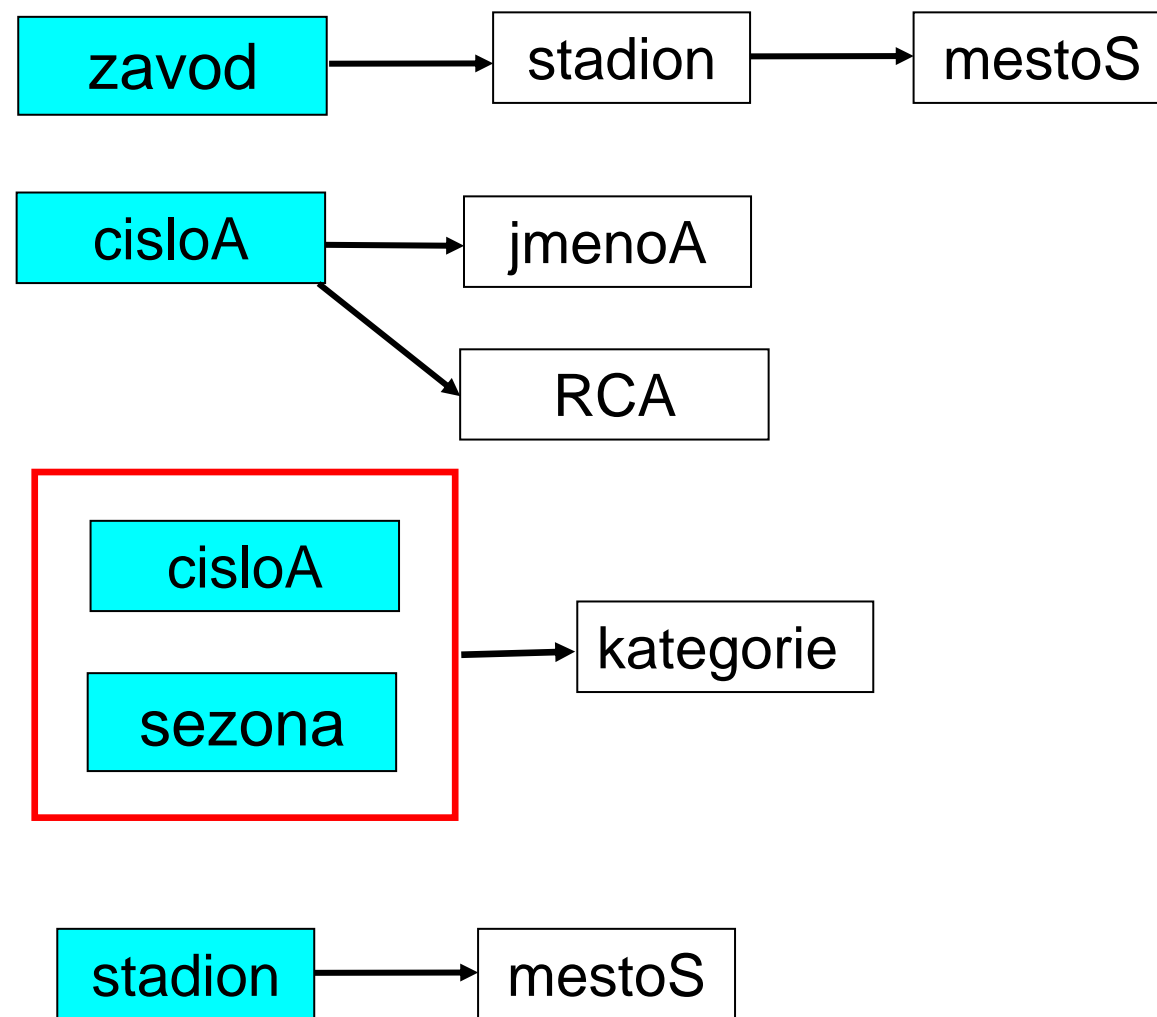
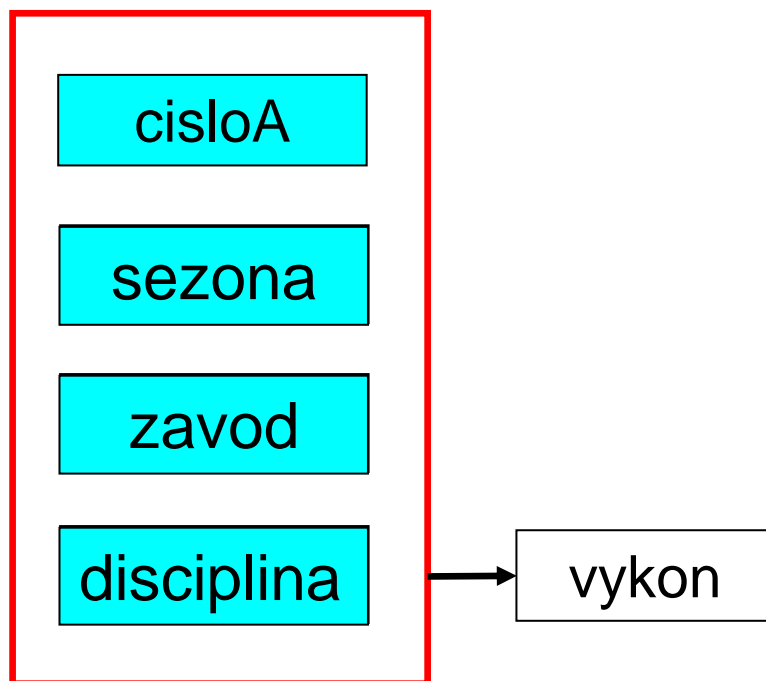
5.2.4. Příklad na normalizaci tabulky II

▫ Normalizace do 3NF



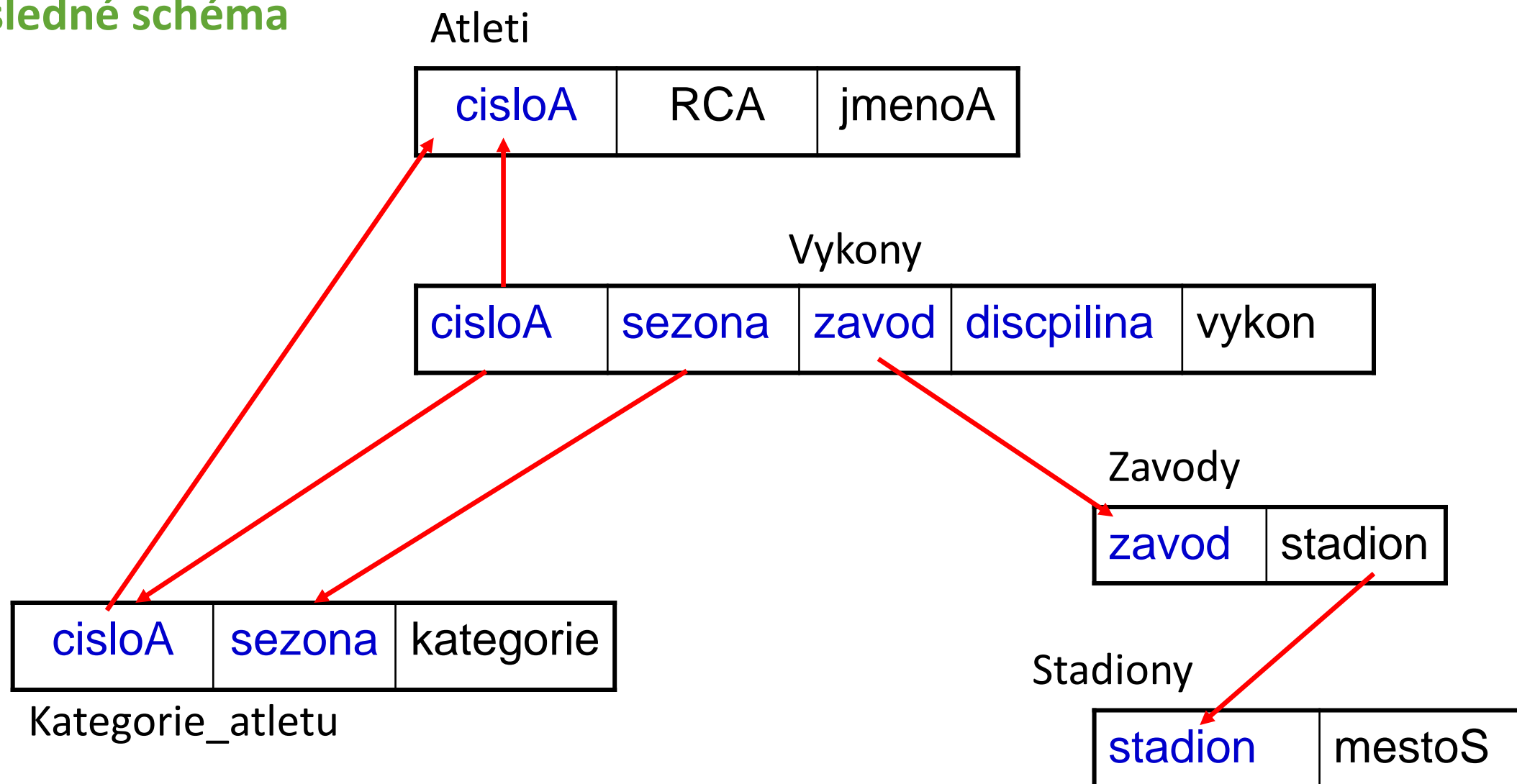
5.2.4. Příklad na normalizaci tabulky II

▫ Normalizace do BCNF



5.2.4. Příklad na normalizaci tabulky

Výsledné schéma



Literatura

1. Silberschatz, A., Korth H.F., Sudarshan, S.: Database System Concepts. Fifth Edition. McGRAW-HILL. 2006, str. 263-310.
2. Lemahieu, W., Broucke, S., Baesens, B.: Principles of Database Management. The Practical Guide to Storing, Managing and Analyzing Big and Small Data. Cambridge University Press 2018, str. 111-120.
3. Zendulka, J., Rudolfová, I.: Databázové systémy. IDS. Studijní opora. FIT VUT v Brně. 2006, str. 86-109.