

## Normalizácia relačných databáz - 1

### Normálizácia

**1NF (1. normálna forma)**

**2NF závislosť medzi kľúčmi a atribútmi**

**3NF závislosť medzi kľúčmi a atribútmi**

**HBCNF (Heat-Boyce-Codd) závislosť medzi kľúčmi**

### Normalizácia

**Normalizácia** je proces eliminovania duplicit, nekonzistentností v dátach a závislostiach v relačných databázach pomocou

- pravidiel medzi atribútmi a funkčnými závislosťami (funkčným závislostiam sa budeme podrobnejšie venovať v letnom semestri)
- **dekompozície** tabuľky (náhrada tabuľky s veľkým počtom stĺpcov s viacerými tabuľkami s menším počtom stĺpcov)
- s cieľom zabezpečiť integritu dát a úsporne šetriť miesto na ukladanie údajov.

Pritom atribúty sa rozdeľujú na kľúčové a neklúčové.

Kľúče

KK - kandidátny kľúč: a)  $KK \rightarrow R$  b)  $\neg \exists K \subset KK: K \rightarrow R$

PK - primárny kľúč (jeden z KK)

Atribúty KA - kľúčový/é atribút/y: je/sú súčasťou nejakého KK

NA - neklúčový atribút: nie je súčasťou žiadneho KK

Tab1

idStudent	Meno	Aresa	idPredmet	NazovPred	Ucitel	Miestnost
1	On	Ke	11	PAZ	M	P3
1	On	Ke	12	DBS	T	P1
2	Ona	Rv	12	DBS	T	P1

- Duplicita - redundantnosť

- Problémy s - obnovením

- vložení

- vymazaním

	Výhody	Nevýhody
Pôvodná tabuľka	Prehľadnejšia	Duplic. + Problémy
Dekompozícia	Menej Duplic., Problémov	Viac tab. => View

### Dekompozícia:

Tab2

idStudent	Meno	Aresa
1	On	Ke
2	Ona	Rv

Tab3

idStudent	idPredmet
1	11
1	12
2	12

Tab4

idPredmet	NazovPred	Ucitel	Miestnost
11	PAZ	M	P3
12	DBS	T	P1

Tab1 obsahuje  $3 \cdot 7 = 21$  údajov a Tab2-Tab4 obsahujú  $2 \cdot 3 + 3 \cdot 2 + 2 \cdot 4 = 20$  údajov. To nevyzerá na prvý pohľad veľkou úsporou. Ale skúsme vnoriť do nich údaj, že ona si zapísala aj predmet PAZ. V Tab1 potrebujeme na to v novom riadku až 7 údajov kým v prípade Tab2-Tab4 iba 2 údaje ako nový riadok v Tab3.

Normálne formy predstavujú skupinu kritérií, ktoré sa používajú na normalizáciu tabuliek. Máme 6 normálnych foriem NF. Edgar Frank Codd definoval prvé 3+1 (1971+1974, Heat-Boyce-Codd NF), ale iba druhá nie je problematická.

Ďalšie normálne formy boli definované Faginom a Datem.

- 4, 5 DKNF Fagin 1977-1981,

- 6 NF Chris Date 2002,

- najnovšia NF je **Essential tuple normal form** ETNF, Darwen-Date-Fagin, 2012, ktorej sa budeme venovať v letnom semestri.

Hovorí sa, že tabuľka je v KNF (K-tej NF), ak spĺňa požiadavky KNF.

Každá NF má silnejšie požiadavky ako predchádzajúca.

KNF vyžaduje, aby (K-1)NF bola splnená, kde  $K \geq 2,6$ .

Ale tabuľka, ktorá je v KNF nemusí byť v (K+1)NF.

1NF (jedna formulácia) v podstate zodpovedá definícii relácie.

Druhá a tretia NF sa zaoberá vzťahom neklúčových atribútov ku kompozitným kľúčom: požadujú, aby všetky neklúčové atribúty boli funkčne závislé na „(celom) kľúči a ničom inom ako na kľúči“.

### **Denormalizácia**

Keď tabuľky sú už príliš štíhle, je ich príliš veľa a obsahujú malý počet atribútov, často pristupujú k opačnému procesu – k denormalizácii tabuliek.

## 1NF

Tu v podstate ide o požiadavky, pravidlá, ktoré súvisia s definíciou *relácie*, tabuľky (ďalšie súvisia s *relačnou schémou*).

Codd pôvodne požiadaval splnenie nasledujúcich podmienok (prvé dva sú problematické)

1. - jedinečnosť riadkov, tzn. aby relácia mala primárny kľúč (dnes DB systémy netrvajú na tom, aby tabuľky mali PK)
2. - atomické (ďalej už nedeliteľné) hodnoty pre atribúty (dátum, desatinné číslo, reťazec – nie sú skutočne atomické, iba na základe dohody).
3. - tabuľka nesmie obsahovať podobné, opakujúce sa atribúty (pozri 3. riešenie)!

### Príklad:

Tab-Student

idStud	Krstne	Priezvisko	CisloTel
1	Jan	X	1
2	Oto	Y	2

Problémy s touto tabuľkou môžu nastať, keď dovoľíme, aby študenti mohli mať viac telefónov. Napríklad: 2, Oto, Y, 6. Ako to vyriešiť?

Ukážeme 4 riešenia, z ktorých iba posledné je akceptovateľné.

1. riešenie: zoznam hodnôt

2	Oto	Y	2, 6
---	-----	---	------

Tu máme nevhodný dátový typ **2, 6** a dané riešenie sťažuje dopytovanie.

2. riešenie: nový riadok

2	Oto	Y	2
2	Oto	Y	6

Tu môžu nastať problémy s obnovením a budeme mať viac duplicitných hodnôt.

3. riešenie: opakujúce sa atribúty

idStud	Krstne	Priezvisko	CisloTel1	CisloTel2	CisloTel3
1	Jan	X	1		
2	Oto	Y	2	6	

Pri tomto riešení na jednej strane môžeme mať stĺpec samých *null* hodnôt a na druhej, čo keď niekto má až štyri čísla.

4. riešenie: dekompozícia na dve tabuľky:

idStud	Krstne	Priezvisko
1	Jan	X
2	Oto	Y

idStud	CisloTel
1	1
2	2
2	6

## 2NF

**Definícia:** Druhá normálna forma požaduje,

- aby relácia (presnejšie RSch) bola v prvej NF a zároveň
- aby každý **neklúčový** atribút bol **úplne funkčne závislý** na kľúči (na **celom kľúči**, nie len na jeho podmnožine).

⇔ neklúčový atribút upresňuje, charakterizuje celý kľúč

⇔ neexistuje neklúčový atribút, ktorý by upresňoval, charakterizoval časť kľúča

Ak tabuľka je v 1. NF a nemá kompozitný kľúč, potom automaticky je v 2. NF.

### **Príklad 1.**

Tab: Predmety

Kompozitný kľúč

<u>Ucitel</u>	<u>Predmet</u>	Adresa
A	PAZ	Ke
B	DBS1	Rv
B	DBS2	Rv

Adresa charakterizuje učiteľa, ale nie Predmet.

Dôsledok: duplicita, problémy s obnovou

Riešenie:

Tab: Učitelia

<u>Ucitel</u>	Adresa
A	Ke
B	Rv

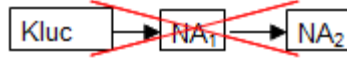
Tab: Predmety

<u>Ucitel</u>	<u>Predmet</u>
A	PAZ
B	DBS1
B	DBS2

### 3NF

**Definícia:** Tretia normálna forma požaduje,

- aby R Sch bola v 2. NF a zároveň
- (Codd 1972) žiadny **NA nezávisel tranzitívne** od kľúča.



⇔ Každý **neklúčový** atribút má priamo, teda **netranzitívne** závisieť od kľúča.

⇔ neklúčový atribút upresňuje, charakterizuje iba kľúč, ale nie iný neklúčový atribút.

Ak tabuľka je v 2NF a nemá neklúčový atribút, automaticky je v 3NF – aj v **HBCNF**.

#### Príklad 1.

Kľúč

Turnaj	Rok	Vitaz	DatumNarodenia

Vek charakterizuje Víťaza => dekompozícia:

Turnaj	Rok	Vitaz

Pretekar	DatumNarodenia

## HBC-NF (Heat-Boyce-Codd)

3NF  $\Leftrightarrow$

**Nekľúčový** atribút závisí [od celého kľúča a] iba od kľúča.

### **3.5NF**

**Každý** atribút závisí [od celého kľúča a] **iba od** kľúča.

Poznámka

- reprezentuje fakt o celom kľúči
- závisí od všetkých kľúčov

**Atribúty závisia od kľúča [1NF], od celého kľúča [2NF] a iba od kľúča [3.5NF].**

Poznámky :

- Vo väčšine prípadov, ak tabuľka je v 3NF, je aj v **HBC-NF**.
- 3NF tabuľka nespĺňa požiadavky HBCNF ak jej kompozitné kľúče sa prelínajú (majú spoločný atribút).

V LS ukážeme

- **R(A, B) s kľúčom je vždy v HBC-NF**
- $\{AB \rightarrow C, C \rightarrow B\}$  nemôže byť reprezentovaná v HBC-NF / r. 1979

## Príklad normalizačného procesu

Uvažujme tabuľku

Student0

<u>idStud</u>	<u>Ucitel</u>	<u>UciKabinet</u>	<u>Predmet1</u>	<u>Predmet2</u>	<u>Predmet3</u>
1	A	101	DBS1	DBS2	MPJ
2	B	102	PAZ1	PAZ2	Siete

Pri normalizácii hľadáme, čo je zlé.

Na ceste k **dosiahnutiu až 3NF** zrušme najskôr opakujúce sa atribúty kvôli 1NF bez dekompozície:

Student1

idStud	Ucitel	UciKabinet	Predmet
1	A	101	DBS1
1	A	101	DBS2
1	A	101	MPJ
2	B	102	PAZ1
2	B	102	PAZ2
2	B	102	Siete

Existuje kľúč K: K → (určuje) Predmet ?



Nepotvrďujeme to, čo sa zdá byť správne, napr.

idStud → Ucitel alebo idStud → UciKabinet,

ale hľadáme to, čo je určite zlé:

idStud **ne** → Predmet a ani (idStud, Ucitel, UciKabinet) **ne** → Predmet.

Teda predmet **funkčne NEzávisí od kľúča** idStud.

⇒ Dekompozícia (zrušenie duplicit) - Student2 už bude v 2NF

Student2

<u>idStud</u>	<u>Ucitel</u>	<u>UciKabinet</u>
1	A	101
2	B	102

VyberPredmetov

idStud	Predmet
1	DBS1
1	DBS2
1	MPJ
2	PAZ1
2	PAZ2
2	Siete

Zatiaľ obe tabuľky sú problematické z hľadiska 3NF.

Riešme najprv ľavú tabuľku. Aby ona bola v 3NF, musíme zrušiť atribúty, ktoré závisia nie od kľúča: UciKabinet závisí od/charakterizuje Ucitel-a.

⇒ Dekompozícia - kompozitný kľúč pre Studentš

Student3

Ucitel

<u>idStud</u>	<u>Ucitel</u>
1	A
2	B
1	C

<u>Ucitel</u>	<u>UciKabinet</u>
A	101
B	102

Pozrime sa teraz na druhú tabuľku. Môže byť Predmet kľúčom? Čo ak DBS1 zapíše aj ďalší študent? Teda čo bude kľúčom, ktorá najmenšia množina atribútov určuje jednoznačne riadok?

Každá z daných troch tabuliek už bude v 3. NF:

Student3

<u>idStud</u>	<u>Ucitel</u>
1	A
2	B
1	C

Ucitel

<u>Ucitel</u>	UciKabinet
A	101
B	102

VyberPredmetov

<u>idStud</u>	<u>Predmet</u>
1	DBS1
1	DBS2
1	MPJ
2	PAZ1
2	PAZ2
2	Siete
2	DBS1

## Normalizácia relačných databáz – 2

LetnýSemester - Normalizácia na základe funkčných závislostí + ETNF