

Normalizace relační databáze - význam normalizace, normální formy, ukázky použití na příkladech, normalizace vs denormalizace, důvody

Normalizace relační databáze - význam normalizace, normální formy, ukázky použití na příkladech, normalizace vs denormalizace, důvody.....	1
Normalizace.....	2
1. NF - konzistence.....	3
2. NF - redundance.....	4
3. NF.....	5
Denormalizace.....	6
Normalizace vs. Denormalizace.....	6

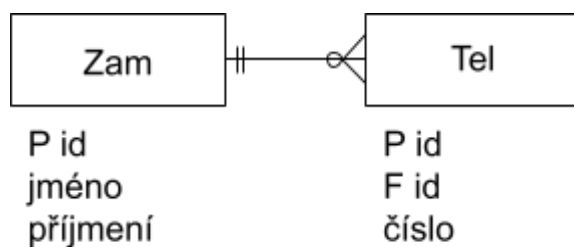
Normalizace

- Úprava struktury databáze (tabulek) tak, aby nedocházelo k redundanci dat (2. a 3. NF) a k zajištění konzistence dat při vyhledávání (1. NF)
- Zajištěna pomocí normálních forem
- Normalizace dat je proces transformace entit a relací z konceptuálního modelu do struktury tabulek v reálné databázi tak, aby vznikla konzistentní databáze, která neukládá redundantní data, a zároveň je snadno rozšiřitelná, upravitelná a udržitelná.
- Cíl normalizace
 - odstranění redundantních (opakujících se) dat
 - omezení složitosti - rozložení složitých relací
 - snaha o rozšiřitelnou, upravitelnou a výkonnou databázi
 - umožňuje vytvořit tabulky, které lze snadno udržovat a efektivně sestavovat příkazy
- Normální formy
 - 1. NF - atomárnost
 - 2. NF - každý neklíčový atribut závisí na celém primárním klíči
 - 3. NF - neklíčové atributy závisí pouze na klíči
 - BCNF - Boyce Coddova normální forma
 - 4. NF
 - 5. NF

1. NF - konzistence

- každý atribut musí mít atomární hodnoty
1. více hodnot, ale zde rozdělit na daný, předem známý počet (max. 5)
 - řešení: každá hodnota bude ve vlastním atributu
 - např.: adresa => město, ulice, číslo popisné
 2. více hodnot, více než 5 nebo není známý počet
 - řešení: rozdělit na více tabulek
 - např.:

jméno	příjmení	telefon
Jan	Kos	777 777 7777, 333 333 333
David	Kohout	111 111 111
Karel	Novák	444 444 444, 987 654 321, 123 456 789

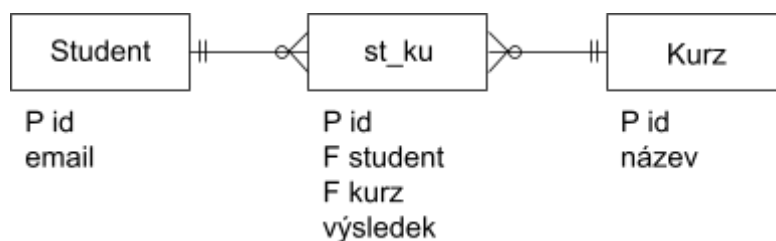


2. NF - redundance

- řešíme je při složeném PK a výskytu redundancí (pro jednoduchý je automaticky splněna)
 - složený PK = z více atributů
- je splněna, pokud existuje (je definován) PK, je splněna 1.NF a zároveň všechny neklíčové atributy závisí na celém PK
- např.:

student_id	kurz_id	název_kurzu	email_studenta	výsledek
1	1	Java	kos@gmail.com	80
2	1	Java	novak@gmail.com	60
3	2	Python	plch@gmail.com	100
1	2	Python	kos@gmail.com	95
4	2	Python	horak@gmail.com	70

- student_id; kurz_id PK
- název_kurzu; email_studenta; výsledek neklíčové atributy
- název_kurzu kurz_id závislé
- název_kurzu student_id nezávislé
- výsledek závislé na celém klíči - pokaždé se mění
- řešení:

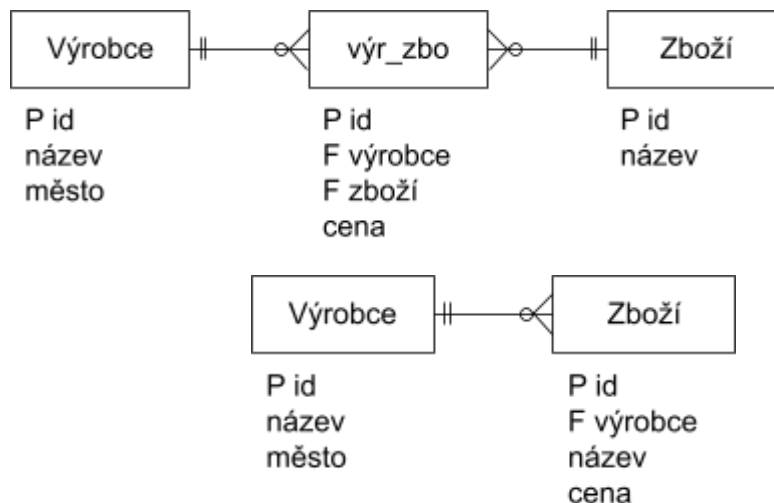


3. NF

- při jednoduchém PK
- je splněna, pokud existuje PK, je splněna 1.NF a 2.NF a zároveň všechny neklíčové atributy jsou mezi sebou nezávislé (tranzitivní nezávislost)
- např.:

id	název zboží	název výrobce	město výrobce	cena zboží
1	mléko	Olma	Olomouc	15
2	tvaroh	Olma	Olomouc	30
3	mléko	Tatra	Kolín	25
4	jogurt	Tatra	Kolín	15
5	jogurt	Daume	Praha	12

- název výrobce město výrobce závislé
- řešení:



denormalizace =>
- bude redundance

Denormalizace

- Denormalizace je opačný proces normalizace. Zatímco normalizace se zaměřuje na odstranění redundantních dat a optimalizaci struktury databáze, denormalizace záměrně využívá redundanci a slučuje tabulky s cílem zlepšit výkon databáze, především při čtení dat.
- může obsahovat
 - redundantní data
 - např.: kontakt na zákazníka je přímo v objednávce
 - důvodem je, že se vyhneme složitým selectům s nutnými joins nebo pod selecty
 - uložené informace
 - např.: předem spočítané hodnoty agregačními funkcemi
 - trigger na dopočítání ceny celkem
 - cena_položky, cena_kus, počet_kusů
 - důvodem je zrychlení statistik přehledů
 - **Ukládání odvoditelných dat** - Data, která mohou být snadno spočítána na základě jiných hodnot, jsou uložena přímo v tabulce. Tím se šetří čas při vykonávání dotazů
 - spojená data z více tabulek do jedné (datový sklad)
 - např.: produkt, dodavatel v dodávce
 - důvodem je zjednodušení dotazů - dotaz pouze nad 1 tabulkou, zrychlení reportů, přehledů, statistik
 - **Ukládání konstantních hodnot** - Často používané hodnoty (např. názvy států nebo měn) mohou být uloženy přímo v tabulce, místo aby byly načteny z jiné tabulky, což zrychluje dotazy.

Normalizace vs. Denormalizace

- Normalizace
 - lepší integrita, bez redundancí
 - jednodušší správa
 - rychlejší vkládání, mazání a aktualizace dat
 - pomalejší dotazy (čtení), pomalejší přehledy a statistiky
- Denormalizace
 - rychlejší hledání, optimalizuje výkon čtení
 - rychlejší přehledy, statistiky
 - redundance, nejednoznačnost
 - nekonzistentní data
 - složitější údržba