

## Normalizace relací

### NORMALIZACE RELACÍ = LOGICKÝ NÁVRH DATABÁZE

Kontrola normalizace relací se provádí, když mám hotový konceptuální návrh databáze. Je to soubor určitých pravidel, která jsou určena za účelem, aby se z logického pohledu zkontroloval navržený konceptuální návrh, že je opravdu správně navržen.

Logický návrh databáze má svá pozitiva i negativa. Kontrola konceptuálního modelu z pohledu logického schématu, že je databáze **logicky relačně správně navržená**, je dána v podstatě tím, do jaké míry si dám práci s konceptuálním návrhem. Když ho odfláknu, budu mít více práce s převodem na logické schéma.

Kontrola logického návrhu databáze v podstatě určuje do jaké míry bude výsledná databáze efektivně rychlá v tvorbě dotazů.

Je několik stupňů normalizace relací, asi aktuálně 12. V tomto dokumentu ukáži některé, protože kdybych chtěl doslova dodržet veškerou normalizaci, tak se mi databáze rozpadne. Je nutné s citem normalizovat. Vědět, kdy ještě normalizovat, kdy už ne, aby to nezpomalovalo databázi. **Normální formou (NF) mohu odstranit redundanci (nadbytečnost).**

### Postup normalizace:

Kouknu se na třídu entit (tabulku) a na tu relaci a budu zjišťovat, jestli to pravidlo splňuje, nesplňuje nebo jestli už chci ukončit normalizaci. Čím více jsou tabulky normalizovány, tím lépe se s nimi pracuje z hlediska aplikační logiky.

Na začátku mám nenormalizovanou databázi nebo relaci. Vyberu si jednu třídu entit. Kouknu se na pravidla pro první normální formu, pokud jsou splněna, označím, že daná třída entit nebo relace je v 1NF. Podívám se na pravidla 2NF, pokud splňuje tato pravidla, tak budu tvrdit, že daná třída entit je v 2NF, a tak budu postupovat dál a dál, až když narazím na NF která není splněna, tak zůstanu u předchozí NF a tím končím.

Nepsané pravidlo: měl bych se dopracovat k 3NF.

### Proč normalizovat

Nenormalizované relace způsobují řadu problémů ať už při přidávání nějakých informací, odebrání nějakých informací, případně při aktualizaci. Tyto problémy se označují jako anomálie. Existují tři typy anomálií: při vkládání, při odstraňování, při aktualizaci. Příklad:

číslo zákaz.	jméno zákaz.	adresa zákaz.	město zákaz.	PSC zákaz.	telefon zákazníka	datum objednávky	výrobek	množství	jednotková cena	Cena za výrobek
2345	Jan K	Důl 5	Kadaň	43201	776555405	20.6.2016	stůl	2	5000	10000
							židle	8	2000	16000
							prkno	2	500	1000
							stojan	2	3000	6000

V příkladu je jeden jediný záznam, ve kterém si 1 zákazník objednal v objednávce určité množství výrobků, což je evidence faktury. Jedná se o nenormalizovanou relaci. V tomto příkladu bereme v potaz, že: když vyplňuji záznam, tak už vyplňuji záznam o objednávce, a až pak zaeviduji zákazníka.

Anomálie při vkládání: již eviduji číslo zákazníka ještě bez vytvořené objednávky. Problém by bylo přidat slevu pro nové zákazníky.

Anomálie při odstraňování: nějaký zákazník udělá objednávku, zapíše si veškeré informace, konkrétní objednávku, objednávku vyřídím, vyfakturuji, objednávku automaticky vymažu, problém: smažou se informace i o zákazníkovi, pak v případě problémů je problém dohledat zákazníkovo předchozí objednávky. Nebo ztráta informací o zákazníkovi vytvoří problém jak dohledat druhou objednávku, kterou v ten den ještě provedl.

Anomálie při aktualizaci: jestli zákazník udělá během týdne například 5 objednávek a najednou zjistím že má změnu telefonního čísla, musím u těch objednávek přepisovat telefonní čísla. V případě že mu to potřebuji odeslat, potřebuji vědět aktuální informace o zákazníkovi. K tomu slouží normalizace.

## NULTÁ NORMÁLNÍ FORMA

Mám entitu (tabulku) v Excelu kde jsem všechno evidoval. Abych tabulku mohl označit nultou normální formou, tak musí obsahovat **alespoň jednu víceznačnou hodnotu**. ONF splňuje jakákoliv tabulka, ale tabulku označuji tou nejvyšší normální formou. Například atribut Kurs obsahuje hodnoty, které by šli oddělit. V databázi se entita označená ONF hodí pro dodatečné nepotřebné informace u které nepotřebuji třídit, řadit, vyhledávat. Pokud bych měl atribut anotace (stručný popis obsahu dokumentu o několika větách), tak mám víceznačné hodnoty, na každém řádku dokumentu může být popis, podle kterých nebudu vyhledávat. Na vyhledávání mám něco jiného, většinou klíčová slova.

Tabulky (třídy entit) které obsahují víceznačné hodnoty lze označit pouze nultou normální formou. Atribut Jméno obsahuje také víceznačné hodnoty.

ONF:

Jméno	Ulice	Město	PSČ	Kurs
Milan Mroczkowski	Husova 1324	Kadaň	43201	AJ, NJ

## PRVNÍ NORMÁLNÍ FORMA

Převod: ONF → 1NF se provede tak, že se zvýší počet záznamů, víceznačné hodnoty se roztrhnou, aby vznikli atomické hodnoty jednotlivých atributů, to znamená dále **nedělitelné (jednoznačné) hodnoty**. Zvyšují nadbytečnost dat, v dalších NF se to již snižuje. **Nemá primární klíč.**

Hodnota Milan má vlastní atribut Jméno, hodnota Mroczkowski má vlastní atribut Příjmení, tedy, relace je v 1NF. Poznámka: v 1NF často bývají spojovací tabulky.

1NF:

Jméno	Příjmení	Ulice	Město	PSČ	kurs
Milan	Mroczkowski	Husova 1324	Kadaň	43201	AJ
Milan	Mroczkowski	Husova 1324	Kadaň	43201	NJ

## DRUHÁ NORMÁLNÍ FORMA

Je takzvaná funkční závislost.

Třída entit (tabulka) nebo relace musí být první v 1NF, pak musí (určit) **existovat primární klíč** tabulky pro jednoznačnou identifikaci a **každý neklíčový atribut** té tabulky (třídy entit) nebo relace je **závislý na celém primárním klíči**. Každý atribut, který není primárním klíčem nebo cizím klíčem je závislý na všech sloupečcích toho primárního klíče.

Přirozenému primárnímu klíči dávám vždy přednost před umělým primárním klíčem, ale narazím na něj málokdy. Když nenajdu přirozený, použiji umělý primární klíč. Pokud mám malou knihovničku, tak stačí ISBN pro odlišení knih – přirozený primární klíč. Pokud mám vědeckou knihovnu, kde je každá publikace 5x, je jasné, že tam kromě ISBN budu muset mít evidované i číslo konkrétní knížky, řeší se to přes vícesloupcový primární klíč.

Pokud je v 1NF a mám k dispozici pouze jeden cizí klíč FK, tak je vždy v 2NF automaticky. FK je vždy jedno-sloupcový.

2NF se odstraní částečně závislé atributy, tedy závislé jenom na části klíče.

V 2NF se rozpadá původní tabulka na více větších tabulek, ale nemusí se rozpadnout také vůbec, pokud byla dobře konceptuálně navržena.

Poznámka: Ve škole se eviduje jako primární klíč dveře, ne čísla třídy, číslo třídy se může přesouvat. Pak vybírám nejkratší primární klíč.

### 2NF:

ID_STU	Jméno	Příjmení	Ulice	Město	PSČ
01	Milan	Mroczkowski	Husova 1324	Kadaň	43201
02	Milan	Mroczkowski	Husova 1324	Kadaň	43201

<u>kurs</u>	Název
AJ	Anglický jazyk
NJ	Německý jazyk

### NENÍ 2NF:

Jméno	Příjmení	Ulice	Město	PSČ
Milan	Mroczkowski	Husova 1324	Kadaň	43201
Miluše	Mroczkowska	Husova 1324	Kadaň	43201

Jeden primární klíč zde chápat jako tři sloupce: Jméno, Příjmení, Ulice. Město určitě bude záviset na tom v jaké ulici bydlím, určitě ne na Jménu a Příjmení.

Podle jména a příjmení nezjistím v jakém městě bydlím. Jde vidět, že město závisí jen na části primárního klíče, není závislý na celém vícesloupcovém primárním klíči.

## TŘETÍ NORMÁLNÍ FORMA

Je takzvaná tranzitivní (nepřímá) závislost.

Tabulka (třída entit) nebo relace musí být ve 2NF a navíc **nesmí existovat závislost neklíčových atributů navzájem na sobě**.

Když budu kontrolovat 3NF, tak vypustím existenci primárního klíče, vypustím cizí klíče a to co mi zůstane v tabulce nesmí být na sobě navzájem závislé. Odstraní se přímo závislé atributy.

Většinou se doporučuje dostat se do třídy entit té 3NF, aby se zajistila tranzitivní závislost. Udává se, že je to dostatečná normalizace pro databázi, není to vždy nutné dodržet, protože záleží na návrhu databáze administrátora.

Přechod z 2NF → 3NF se řeší rozpadem tabulky na více dílčích tabulek.

**NENÍ 3NF:**

ID_STU	Jméno	Příjmení	Ulice	Město	PSČ
01	Milan	Mroczkowski	Husova 1324	Kadaň	43201
02	Milan	Mroczkowski	Husova 1324	Kadaň	43201

Jméno není závislé na příjmení.

To že tam mám jméno Milan, neodvodíte že mám příjmení Mroczkowski.

To že mám příjmení Mroczkowski neodvodíte že bydlím v ulici Husova 1324, z toho slova Mroczkowski to neodvodíte.

To že bydlím na Husově ulici, neodvodíte že je to v Kadani.

Ale, dá se odvodit: když znám ulici a město, mohu odvodit PSČ. Když znám PSČ a ulici, odvodím město.

3NF:

ID_STU	Jméno	Příjmení	PSČ
01	Milan	Mroczkowski	43201
02	Milan	Mroczkowski	43201

2NF:

PSČ	Ulice	Město
43201	Husova 1324	Kadaň

Co trochu narušuje ty NF a vede k de-normalizaci je Adresa. Pokud chci mít pouze adresu proto, abych mohl na tu adresu někam posílat, tak většinou mi stačí Ulice, Město, PSČ. A nebudu požadovat, aby ta tabulka byla ve 3NF, bude mi stačit ve 2NF.

## Boyce-Coddova NF (3NF+)

Stěží se se s tím setkám. Je trochu silnější než 3NF.

3NF+

ID_STU	Jméno	Příjmení
01	Milan	Mroczkowski
02	Milan	Mroczkowski

**Řeší anomálie, kdy je některý neklíčový atribut určující pro atribut PK („zamaskovaná“ anomálie). Přechod od 3NF → 3NF+ se řeší rozpadem tabulky na více dílčích tabulek.**

PSČ
43201

2NF

PSČ	Ulice	Město
43201	Husova 1324	Kadaň

“Zamaskovaná“ anomálie je toto PSČ.

**Neklíčový atribut nesmí určovat hodnotu jakéhokoliv jiného atributu a to ani atributu PK.**

Ulice	Město
Husova 1324	Kadaň

## ČTVRTÁ NORMÁLNÍ FORMA

- tabulka je ve 3NF
- **tabulka popisuje jeden fakt nebo souvislost**

4NF

ID_STU	Jméno	Příjmení
01	Milan	Mroczkowski
02	Milan	Mroczkowski

Když mám u studenta evidované jméno a příjmení, je jasné že to popisuje jeden fakt nebo souvislost: název toho studenta.

**Není 4NF**

ID_STU	Jméno	Příjmení	Datum_narození
01	Milan	Mroczkowski	1983
02	Milan	Mroczkowski	1983

Tabulka nepopisuje jeden fakt, ale popisuje název studenta a datum\_narození

## PÁTÁ NORMÁLOVÁ FORMA

vše co se dá označit za **číselníky**, většinou to obsahuje dva sloupce, v jednom sloupci je primární klíč, v druhém sloupci mám označení toho primárního klíče.

Když přidáme atribut do tabulky, tak se tabulka rozpadne na několik dílčích – číselníky.

Musí být ve 4NF, má jen jeden jednosloupcový neklíčový atribut.

Není možné přidat do tabulky nový atribut bez toho, aby se rozpadla na několik dílčích tabulek.

5NF


kurz	Název
AJ	Anglický jazyk
NJ	Německý jazyk

Kurz s Názvem popisují jeden fakt nebo souvislost.

Představa: kliknu na internetu na státy a vyjede mi seznam zemí.




## Milan Mroczkowski (\* 2. května 1983)

 Validovaný uživatel


 [sata150@gmail.com](mailto:sata150@gmail.com)

 +420.776555405

 Husova 1324, 43201 Kadaň, Česká republika



ICQ: 162118480  
Skype: milanmr  
Webová adresa: <http://yesit.cz/>  
Osobní stránka: [http://yesit.cz/o\\_mne/o\\_mne.php](http://yesit.cz/o_mne/o_mne.php)

 Stáhnout vizitku



Powered by 

**Materiál vznikl z mých zápisků na školách (především z UJEPu), které jsem doplnil mými komentáři.**

Nejvýraznějším zdrojem byla prezentace:

[https://ki.ujep.cz/enastenka/S%c3%bdkorov%c3%a1%20Kv%c4%9btu%c5%a1e%20Mgr/DSY%20\(P502\)/dbs\\_2\\_logic.pdf](https://ki.ujep.cz/enastenka/S%c3%bdkorov%c3%a1%20Kv%c4%9btu%c5%a1e%20Mgr/DSY%20(P502)/dbs_2_logic.pdf) Květuše Sýkorová