**Nultá normální forma (0NF)**

**Nultá normální forma** obvykle není zvažována, jelikož její splnění bývá v praxi automaticky zaručeno. Přesto lze nalézt následující definici:

·        Schéma relace je v nulté normální formě právě tehdy, když existuje alespoň jeden atribut, který obsahuje více než jednu hodnotu

**První normální forma (1NF)**

Pro splnění **první normální formy** je zapotřebí zajistit následující:

·        splnění nulté normální formy (0NF)

·        všechny atributy tabulky musí být [atomické](https://cs.wikipedia.org/wiki/Atomicita), tedy dále nedělitelné

**Příklad 1**

Klasickým příkladem tabulky porušující první normální formu bývá nejčastěji problém s telefonními čísly, kdy naším cílem je umožnit evidovat pro každou osobu dvě různá telefonní čísla, jak lze vidět v tabulce níže:

·        Poznámka: podtržené názvy atributů představují [primární klíč](https://cs.wikipedia.org/wiki/Prim%C3%A1rn%C3%AD_kl%C3%AD%C4%8D).

|  |  |  |
| --- | --- | --- |
| **Tabulka Osoba** | | |
| **ID Osoby** | **Jméno** | **Telefonní číslo** |
| 1 | Petr Novák | +420 111 222 333 |
| 2 | Jarmil Hnízdo | +420 123 123 123, +420 123 123 124 |

Tato tabulka ovšem porušuje první normální formu, jelikož sloupec (atribut) telefonní číslo není atomický. Zjevným řešením této situace by mohlo být přidání druhého sloupce pro telefonní číslo, abychom zajistili splnění pravidla atomických atributů:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabulka Osoba** | | | |
| **ID Osoby** | **Jméno** | **Telefonní číslo 1** | **Telefonní číslo 2** |
| 1 | Petr Novák | +420 111 222 333 |  |
| 2 | Jarmil Hnízdo | +420 123 123 123 | +420 123 123 124 |

Přestože je tato tabulka formálně správná a již neporušuje pravidlo první normální formy, její návrh je stále problematický. Problém nastane zejména v případě, že bude zapotřebí evidovat čísel více. Rozšiřování tabulky o další sloupce telefonních čísel není často realizovatelné a také se označuje za špatný návrh. Správným řešením je vytvoření nové tabulky a odstranění sloupce telefonních čísel z tabulky osob, jak lze vidět na příkladu níže:

|  |  |
| --- | --- |
| **Tabulka Osoba** | |
| **ID Osoby** | **Jméno** |
| 1 | Petr Novák |
| 2 | Jarmil Hnízdo |

|  |  |  |
| --- | --- | --- |
| **Tabulka Kontakt** | | |
| **ID Kontaktu** | **ID Osoby** | **Telefonní číslo** |
| 1 | 1 | +420 111 222 333 |
| 2 | 2 | +420 123 123 123 |
| 3 | 2 | +420 123 123 124 |

V tomto případě je tedy první normální forma splněna a rozdělením tabulek jsme umožnili bezproblémové přidávání libovolného počtu telefonních čísel pro každou osobu. Všimněme si, že tabulka s kontakty obsahuje navíc kromě primárního klíče také [cizí klíč](https://cs.wikipedia.org/wiki/Ciz%C3%AD_kl%C3%AD%C4%8D), který nám zajišťuje svázání telefonního čísla se správnou osobou.

**Druhá normální forma (2NF)**

Pravidla definovaná **druhou normální formou** lze shrnout na následující:

·        tabulka musí být v první normální formě (1NF)

·        každý neklíčový atribut musí být plně závislý na každém [kandidátním klíči](https://cs.wikipedia.org/wiki/Kandid%C3%A1tn%C3%AD_kl%C3%AD%C4%8D) (neklíčovým atributem rozumíme atribut, který není součástí žádného kandidátního klíče)

Druhá normální forma klade důraz především na odstranění možných duplicit v záznamech.

**Příklad 2**

Uveďme si příklad, kdy máme tabulku evidující následující informace o kurzech. Všimněme si především toho, že tabulka má složený primární klíč {ID Kurzu, ID Semestru}:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabulka Záznam** | | | |
| **ID Kurzu** | **ID Semestru** | **Počet míst** | **Jméno kurzu** |
| IT101 | ls 2017 | 100 | Programování |
| IT101 | zs 2017 | 100 | Programování |
| IT102 | ls 2017 | 200 | Databáze |
| IT102 | zs 2017 | 150 | Databáze |
| IT103 | zs 2017 | 120 | Web design |

Tato tabulka porušuje druhou normální formu, jelikož sloupec *Jméno kurzu* není plně závislý na celém primárním klíči. *Jméno kurzu* je závislé na sloupci *ID Kurzu*, ovšem není již závislé na sloupci *ID Semestru*. V takovéto tabulce navíc dochází k redundanci dat, jak lze vidět na opakujících se jménech každého kurzu. Při vypisování již existujících kurzů v budoucnu (v nových semestrech) by docházelo k neustálému opakování těchto záznamů.

Způsob, jakým lze tento problém vyřešit je dekompozice tabulky a zajištění, že všechny neklíčové atributy dané tabulky budou závislé na celém klíči. Možným řešením může být například následující rozdělení tabulky:

|  |  |  |
| --- | --- | --- |
| **Tabulka Záznam** | | |
| **ID Kurzu** | **ID Semestru** | **Počet míst** |
| IT101 | ls 2017 | 100 |
| IT101 | zs 2017 | 100 |
| IT102 | ls 2017 | 200 |
| IT102 | zs 2017 | 150 |
| IT103 | zs 2017 | 120 |

|  |  |
| --- | --- |
| **Tabulka Kurz** | |
| **ID Kurzu** | **Jméno kurzu** |
| IT101 | Programování |
| IT102 | Databáze |
| IT103 | Web design |

Výše uvedená dekompozice tabulky zajišťuje splnění druhé normální formy. Jediný neklíčový atribut v původní tabulce – *Počet míst* je již závislý na celém složeném primárním klíči, obdobně jako jediný neklíčový atribut *Jméno kurzu* v tabulce Kurz, který je již závislý na jediném primárním klíči nově vytvořené tabulky.

**Třetí normální forma (3NF)**

**Třetí normální forma** klade následující podmínky:

·        tabulka je ve druhé normální formě (2NF)

·        neobsahuje tranzitivní závislosti

Funkční závislost v databázích chápeme jako vztah mezi atributy, kdy atribut Y je funkčně závislý na atributu X.

Tato závislost zajišťuje, že dva řádky mající stejnou hodnotu atributu X budou mít vždy stejnou hodnotu atributu Y. Tranzitivní závislost pak chápeme jako vztah mezi třemi atributy (např. X, Y, Z), kdy atribut Y je funkčně závislý na atributu X, atribut Z je funkčně závislý na atributu Y, a proto lze implikovat, že atribut Z je také funkčně závislý na atributu X.

**Příklad 3**

Nejlépe však tuto normální formu lze pochopit na vypovídajícím příkladu. Mějme následující tabulku evidující jednotlivé filmy, jejich žánry a délku jejich stopáže:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabulka Film** | | | |
| **ID Filmu** | **ID Žánru** | **Název žánru** | **Délka filmu** |
| 1 | 3 | Dokumentární | 01:20:00 |
| 2 | 2 | Akční | 01:05:00 |
| 3 | 1 | Komedie | 01:50:00 |
| 4 | 2 | Akční | 01:10:00 |
| 5 | 3 | Dokumentární | 01:11:00 |

Na první pohled není s tabulkou nic v nepořádku. Tabulka splňuje druhou normální formu, všechny neklíčové atributy jsou závislé na celém primárním klíči (tedy v tomto případě na atributu *ID Filmu*). Přesto tato tabulka nesplňuje pravidla určená třetí normální formou.  
V tabulce výše vidíme, že primární klíč identifikující každý film rozhoduje o tom, jaký žánr bude zvolen, tedy jinými slovy atribut *ID Žánru* je funkčně závislý na atributu *ID Filmu* (ID Filmu → ID Žánru). Další závislost, kterou můžeme identifikovat je vztah identifikačního čísla žánru a názvu daného žánru, jinými slovy atribut *Název žánru* je funkčně závislý na atributu *ID žánru* (ID Žánru → Název žánru). Jak již z definice tranzitivní závislosti víme, z těchto dvou vztahů plyne i vztah třetí, určující funkční závislost atributu *Název žánru* na primárním klíči *ID Filmu*.

Způsob, jakým lze třetí normální formu uspokojit je dekompozice tabulky, díky čemuž dosáhneme odstranění nalezené tranzitivní závislosti. Pro příklad výše by dekompozice vypadala následovně:

|  |  |  |
| --- | --- | --- |
| **Tabulka Film** | | |
| **ID Filmu** | **ID Žánru** | **Délka filmu** |
| 1 | 3 | 01:20:00 |
| 2 | 2 | 01:05:00 |
| 3 | 1 | 01:50:00 |
| 4 | 2 | 01:10:00 |
| 5 | 3 | 01:11:00 |

|  |  |
| --- | --- |
| **Tabulka Žánr** | |
| **ID Žánru** | **Název žánru** |
| 1 | Komedie |
| 2 | Akční |
| 3 | Dokumentární |

Rozkladem původní tabulky nyní dostáváme dvě tabulky, které již neobsahují žádné tranzitivní závislosti. Navíc jsme díky naplnění třetí normální formy napomohli ke snížení redundance dat.

-----Dála jen pro zvídavé studenty, není požadováno k maturitě:-----

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

**Čtvrtá normální forma (4NF)**

**Čtvrtá normální forma** je dalším krokem po uplatnění Boyceho-Coddovy normální formy. Jak již bylo nastíněno v počátku, druhá, třetí a Boyceho-Coddova normální forma řeší funkční závislosti jednotlivých atributů, kdežto čtvrtá a pátá normální forma zkoumá vztahy složených primárních klíčů. Dle jedné z definic navíc od splnění všech předešlých normálních forem umožňuje rozlišení a oddělení nezávislých vícehodnotových atributů vytvářejících složený primární klíč.

Požadavky pro splnění 4NF jsou tedy následující:

·        relace je v BCNF

·        pro každou netriviální vícehodnotovou závislost X ↠ Y je X superklíčem v dané relaci (superklíč je jakékoliv uskupení atributů, které jednoznačně identifikuje záznam v tabulce)

**Příklad 4**

Mějme následující tabulku evidující záznamy o prodejcích, prodávaných produktech a pobočkách, kde nabízejí své produkty:

|  |  |  |
| --- | --- | --- |
| **Tabulka Prodejce-Produkt-Pobočka** | | |
| **Prodejce** | **Produkt** | **Pobočka** |
| Prodejce 01 | Lyže | Praha |
| Prodejce 01 | Lyže | Liberec |
| Prodejce 01 | Lyže | Pardubice |
| Prodejce 01 | Lyžařské brýle | Praha |
| Prodejce 01 | Lyžařské brýle | Liberec |
| Prodejce 01 | Lyžařské brýle | Pardubice |
| Prodejce 02 | Lyže | Praha |
| Prodejce 02 | Lyže | Liberec |
| Prodejce 02 | Lyže | Pardubice |
| Prodejce 02 | Lyžařské brýle | Praha |
| Prodejce 02 | Lyžařské brýle | Liberec |
| Prodejce 02 | Lyžařské brýle | Pardubice |
| Prodejce 03 | Lyže | Pardubice |

Stejně jako předchozí normální formy, i čtvrtá normální forma se snaží zamezit zbytečné redundanci dat. Množství záznamů v tabulce výše bylo zvoleno především proto, aby bylo možné ilustrovat, že i tato normální forma napomáhá ke snížení redundance dat.

Jelikož tabulka neobsahuje žádné neklíčové atributy (je tvořena pouze složeným primárním klíčem {Prodejce, Produkt, Pobočka}), máme zajištěné splnění všech předchozích normálních forem. Tato tabulka však nesplňuje pravidla určená čtvrtou normální formou. Vezmeme-li v úvahu, že všichni prodejci poskytují všechny své produkty na všech svých pobočkách, zjistíme, že složený klíč je bezpochyby tvořen z nezávislých dat, kdy produkt není vázán na určitou pobočku. Z toho důvodu je porušena čtvrtá normální forma a pro její splnění je zapotřebí provést dekompozici tabulky, která by vypadala následovně:

|  |  |
| --- | --- |
| **Tabulka Prodejce-Produkt** | |
| **Prodejce** | **Produkt** |
| Prodejce 01 | Lyže |
| Prodejce 01 | Lyžařské brýle |
| Prodejce 02 | Lyže |
| Prodejce 02 | Lyžařské brýle |
| Prodejce 03 | Lyže |

|  |  |
| --- | --- |
| **Tabulka Prodejce-Pobočka** | |
| **Prodejce** | **Pobočka** |
| Prodejce 01 | Praha |
| Prodejce 01 | Liberec |
| Prodejce 01 | Pardubice |
| Prodejce 02 | Praha |
| Prodejce 02 | Liberec |
| Prodejce 02 | Pardubice |
| Prodejce 03 | Pardubice |

Po dekompozici tabulky již máme splnění čtvrté normální formy zajištěné.

**Pátá normální forma (5NF)**

**Pátá normální forma** tkví ve splnění následujících podmínek:

·        čtvrté normální formy (4NF)

·        tabulku není možné dále bezeztrátově rozdělovat.

Ve všech předchozích normálních formách nedocházelo dekompozicí tabulek ke ztrátě dat[[11]](https://cs.wikipedia.org/wiki/Normalizace_datab%C3%A1ze#cite_note-11), ovšem upravíme-li příklad a situaci ze čtvrté normální formy, kdy prodejci již neposkytují všechny své produkty na každé své pobočce, ale pouze některé z nich, dojde při rozkladu ke ztrátě informací.

**Příklad 5**

Mějme tedy následující upravenou tabulku z minulého příkladu:

|  |  |  |
| --- | --- | --- |
| **Tabulka Prodejce-Produkt-Pobočka** | | |
| **Prodejce** | **Produkt** | **Pobočka** |
| Prodejce 01 | Lyže | Praha |
| Prodejce 01 | Lyžařské brýle | Praha |
| Prodejce 01 | Rukavice | Pardubice |
| Prodejce 02 | Rukavice | Praha |

Když se pokusíme o dekompozici tabulky, stejně jako při uspokojování čtvrté normální formy, dostaneme následující:

|  |  |
| --- | --- |
| **Tabulka Prodejce-Produkt** | |
| **Prodejce** | **Produkt** |
| Prodejce 01 | Lyže |
| Prodejce 01 | Lyžařské brýle |
| Prodejce 01 | Rukavice |
| Prodejce 02 | Rukavice |

|  |  |
| --- | --- |
| **Tabulka Prodejce-Pobočka** | |
| **Prodejce** | **Pobočka** |
| Prodejce 01 | Praha |
| Prodejce 01 | Pardubice |
| Prodejce 02 | Praha |

Zajisté jsme dekompozicí pomohli snížit redundanci dat v naší databázi, která by byla zjevná především u rozsáhlých databází s velkým počtem záznamů. Ovšem ve výsledku jsme ztratili důležité informace o závislosti prodejců – produktů – poboček. Při aplikaci některého z možných spojení tabulek, např. [přirozeného spojování](https://cs.wikipedia.org/wiki/JOIN#P%C5%99irozen%C3%A9_spojov%C3%A1n%C3%AD) (NATURAL JOIN), dostáváme následující výsledky:

|  |  |  |
| --- | --- | --- |
| **Tabulka Prodejce-Produkt-Pobočka** | | |
| **Prodejce** | **Produkt** | **Pobočka** |
| Prodejce 01 | Lyže | Praha |
| Prodejce 01 | Lyžařské brýle | Praha |
| **Prodejce 01** | **Rukavice** | **Praha** |
| Prodejce 01 | Rukavice | Pardubice |
| **Prodejce 01** | **Lyžařské brýle** | **Pardubice** |
| **Prodejce 01** | **Lyže** | **Pardubice** |
| Prodejce 02 | Rukavice | Praha |

Všimněme si, že dle obdržených výsledků vrácených spojením tabulek *Prodejce-Produkt* a *Prodejce-Pobočka* došlo k vytvoření tří nových záznamů, které tvrdí, že např. Prodejce 01 prodává lyžařské brýle v Pardubicích, což ovšem v porovnání s původní tabulkou víme, že není pravda. Uvažujme tedy o dekompozici původní tabulky na tři tabulky oproti dvěma původním. Tedy kromě tabulek Prodejce-Produkt a Prodejce-Pobočka přidáme ještě tabulku Pobočka-Produkt evidující vztah mezi produkty a jejich dostupností na jednotlivých pobočkách. Mějme tedy navíc následující tabulku:

|  |  |
| --- | --- |
| **Tabulka Pobočka-Produkt** | |
| **Pobočka** | **Produkt** |
| Praha | Lyže |
| Praha | Lyžařské brýle |
| Praha | Rukavice |
| Pardubice | Rukavice |

Problém však nastává ve chvíli, kdy se pokusíme tyto tři tabulky spojit a dostat tak původní data. Zjistíme, že se nám takové spojení realizovat nepodaří a že není způsob, jak bychom dekompozici původní tabulky mohli provést bezeztrátově. Tedy lze konstatovat, že tabulka Prodejce-Produkt-Pobočka již nelze dále bezeztrátově rozložit (je tedy v 5NF), a proto se musíme spokojit s redundancí dat původní tabulky pro zajištění, že nepřijdeme o žádná potenciálně důležitá data.

**Příklad 6**

Víme už tedy o jaké tabulce můžeme konstatovat, že je v páté normální formě. Ukažme si ještě tabulku, která pátou normální formu porušuje. Upravíme-li záznamy původní tabulky a určíme-li, že se v Praze prodávají všechny produkty (lyže a lyžařské brýle), ale v Pardubicích pouze lyžařské brýle, bude nová tabulka vypadat následovně:

|  |  |  |
| --- | --- | --- |
| **Tabulka Prodejce-Produkt-Pobočka** | | |
| **Prodejce** | **Produkt** | **Pobočka** |
| Prodejce 01 | Lyže | Praha |
| Prodejce 01 | Lyžařské brýle | Praha |
| Prodejce 01 | Lyžařské brýle | Pardubice |
| Prodejce 02 | Lyže | Praha |
| Prodejce 02 | Lyžařské brýle | Praha |

Budeme-li postupovat stejně jako při pokusu o dekompozici bezeztrátově nerozdělitelné tabulky z minulého příkladu, dostaneme následující tři tabulky:

|  |  |
| --- | --- |
| **Tabulka Prodejce-Produkt** | |
| **Prodejce** | **Produkt** |
| Prodejce 01 | Lyže |
| Prodejce 01 | Lyžařské brýle |
| Prodejce 02 | Lyže |
| Prodejce 02 | Lyžařské brýle |

|  |  |
| --- | --- |
| **Tabulka Prodejce-Pobočka** | |
| **Prodejce** | **Pobočka** |
| Prodejce 01 | Praha |
| Prodejce 01 | Pardubice |
| Prodejce 02 | Praha |

|  |  |
| --- | --- |
| **Tabulka Pobočka-Produkt** | |
| **Pobočka** | **Produkt** |
| Praha | Lyže |
| Praha | Lyžařské brýle |
| Pardubice | Lyžařské brýle |

Při pokusu o jejich opětovné spojení, zjistíme, že se nám v tomto případě podařila bezeztrátová dekompozice a výsledkem naší selekce je totožná tabulka s tabulkou před rozdělením. Po spojení těchto tabulek tedy získáváme:

|  |  |  |
| --- | --- | --- |
| **Tabulka Prodejce-Produkt-Pobočka** | | |
| **Prodejce** | **Produkt** | **Pobočka** |
| Prodejce 01 | Lyže | Praha |
| Prodejce 01 | Lyžařské brýle | Praha |
| Prodejce 01 | Lyžařské brýle | Pardubice |
| Prodejce 02 | Lyže | Praha |
| Prodejce 02 | Lyžařské brýle | Praha |

Jelikož se nám podařilo opětovně spojit tři rozdělené tabulky a získat tak nezměněnou původní tabulku, lze konstatovat, že tabulka nesplňuje pravidla stanovená pátou normální formou. Srovnáme-li oba příklady této normální formy, zjistíme, že hlavním rozdílem bylo uvedení rozšiřujících informací, které nejsou ihned zjevné při pohledu na samotnou tabulku. Identifikování tabulek, které porušují pátou normální formu bývá v praxi velmi složité, a navíc reálný přínos její aplikace bývá často nezřetelný. Pátá normální forma patří mezi formy založené spíše na teoretickém zázemí, postrádajíc reálně dosažitelné výsledky při aplikaci v praxi.