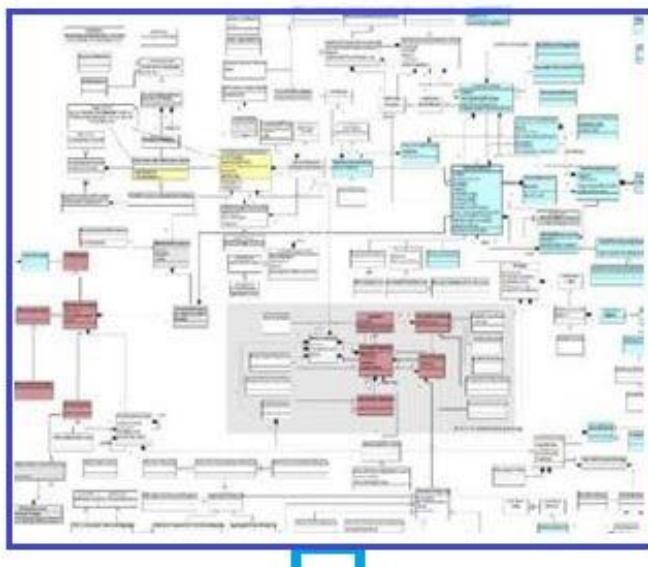
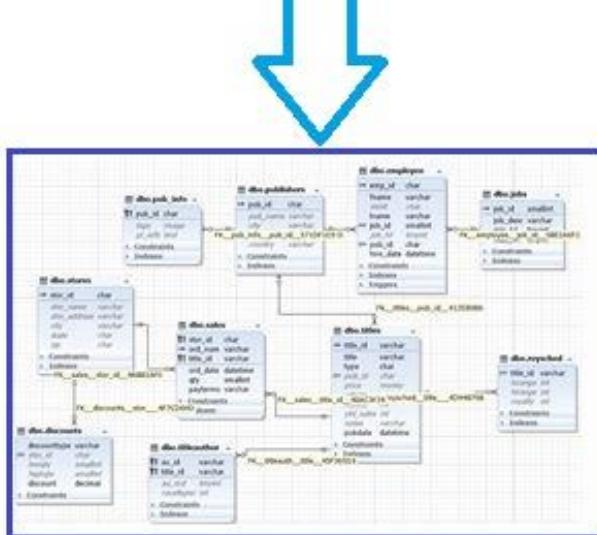


# Objektovo – relačné mapovanie

(Ilustratívny obrázok)



## Diagram tried



## Model objektov uloženia (fyzický model)

# Terminológia

- **Vyšší stupeň abstrakcie**

**Názov:** konceptuálny model, logický model (údajov), ...

Reprezentovaný ako Entitno-relačný model (ERM) alebo diagram tried (DT) v UML

**Vlastnosti:** je nezávislý na implementácii

- **Nižší stupeň abstrakcie**

**Názov:** fyzický model, model uloženia

**Vlastnosti:** berie do úvahy implementačné požiadavky

**Príklad reprezentácie:** ERM po normalizácii

# Nutné poznatky z databáz

Na stránke cvičení je študijný materiál „Úvod do návrhu databáz“ (3. položka):

## Študijné materiály

- Úvod do návrhu databáz [PDF]

## Relačný model databáz a vztahy medzi tabuľkami (poznatky z 2. ročníka)

Násobnosť	Spôsob spojenia tabuľiek
Jedna k jednej	Spoja sa primárne kľúče z obidvoch tabuľiek.
Jedna – mnoho	Do tabuľky s násobnosťou „mnoho“ sa pridá FK, čo je primárny kľúč zo susednej tabuľky
Mnoho – jedna	Ako v predchádzajúcom prípade, ale zrkadlovo otočené.
Mnoho - mnoho	Medzi pôvodné tabuľky sa pridá nová <u>medzitabuľka</u> a do nej sa pridajú FK1 a FK2.

# Objektovo-relačné mapovanie (ORM)

**ORM je proces návrhu modelu údajov z diagramu tried v UML do relačnej databázy.**

## Cieľ a účel:

- Diagram tried je implementačne nezávislý.
- Ak zákazník požaduje uchovávanie údajov prostredníctvom relačnej databázy, je potrebné transformovať triedy na tabuľky, atď. (pozri ďalej)
- Cieľ (1): Vytvorenie fyzického modelu údajov v relačnej databázovej reprezentácii.

# Objektovo-relačné mapovanie (ORM)

**ORM je proces návrhu modelu údajov z diagramu tried v UML do relačnej databázy.**

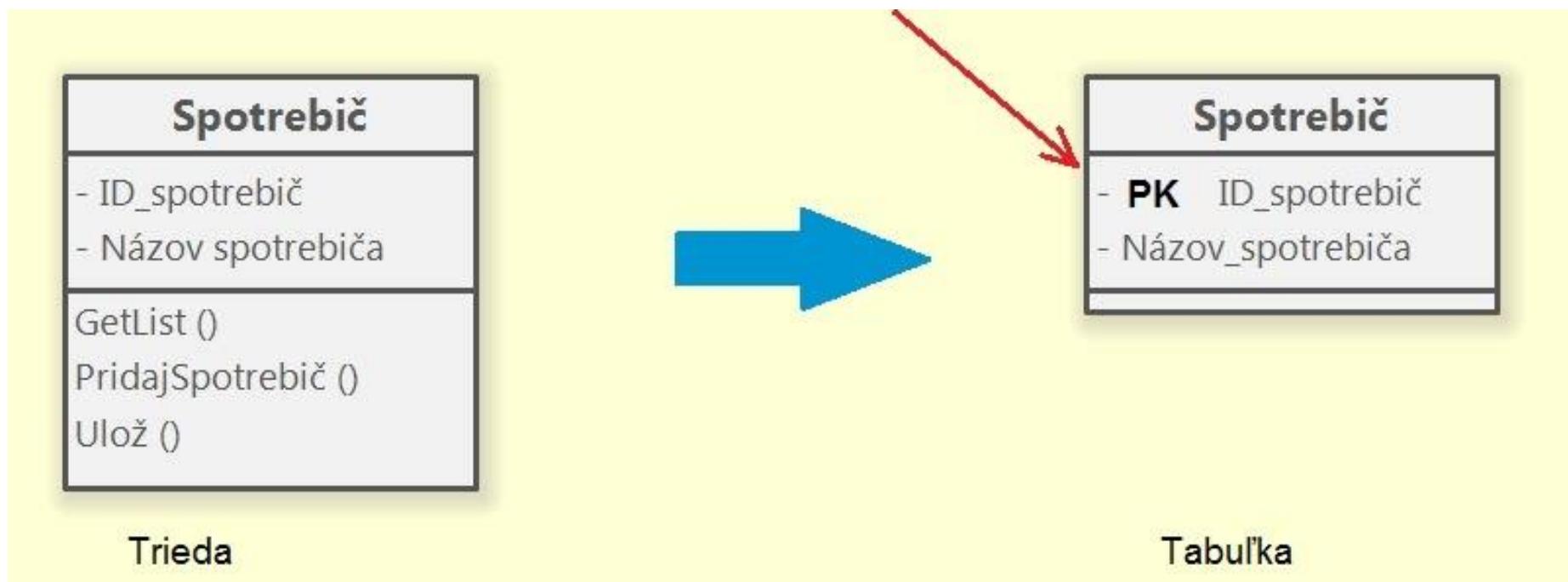
Ciel' (2): Verifikuje správnosť navrhnutého modelu údajov.

**Mapuje:**

- Triedy → tabuľky databázy
- (Konektory) asociácie, kompozície, agregácie, generalizácie → vzťahy v ERD
- Atribúty → atribúty (stĺpce) databázy; v ERD sa používajú kľúče (primárne a cudzie, t.j. **PK**, **FK**)

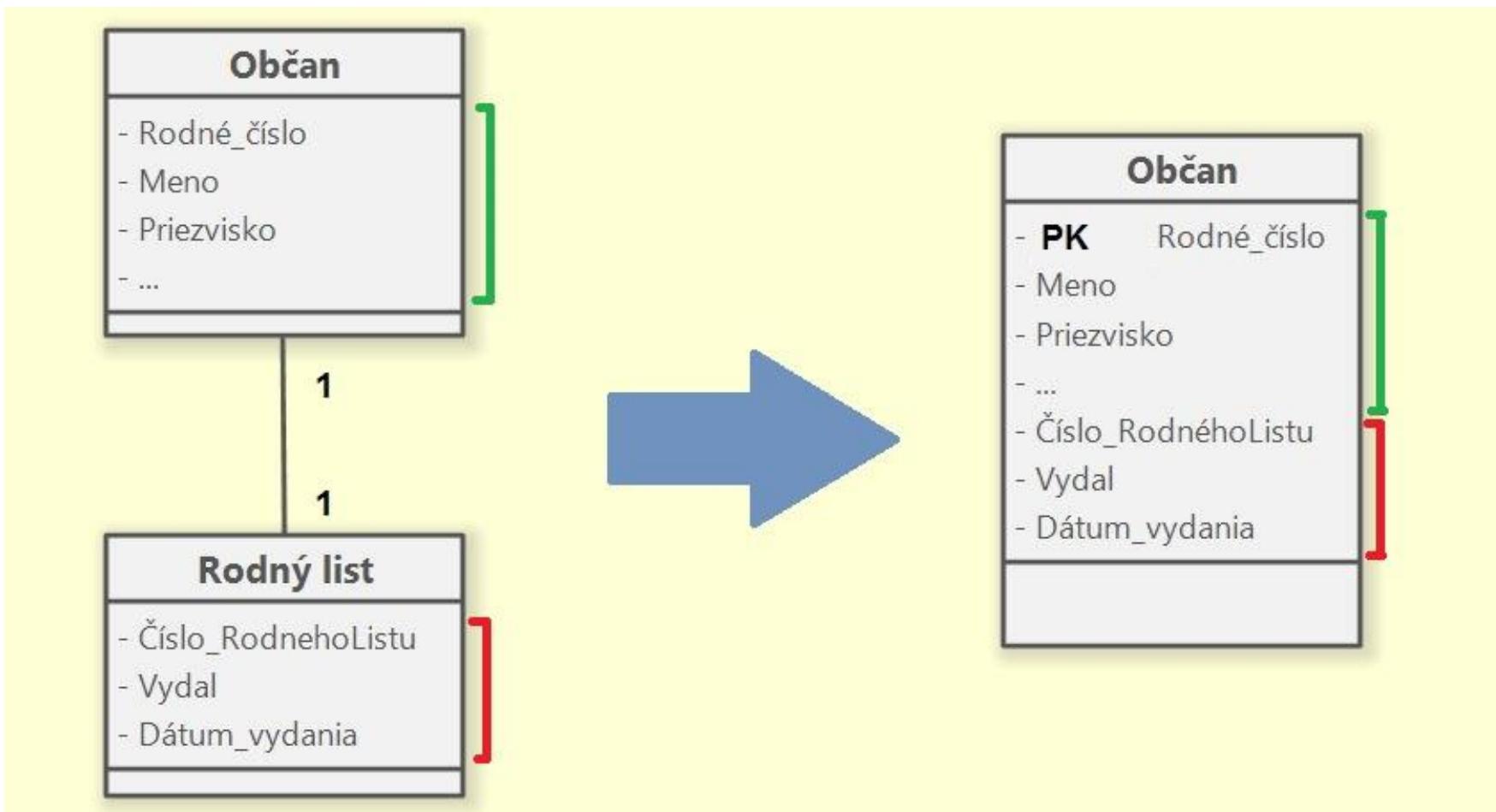
# Mapovanie tried

- Prebieha priamočiaro: trieda sa mapuje na tabuľku
- Metódy pôvodnej triedy sa vynechávajú
- Každá tabuľka musí mať **PRIMÁRNY KĽÚČ (PK)**



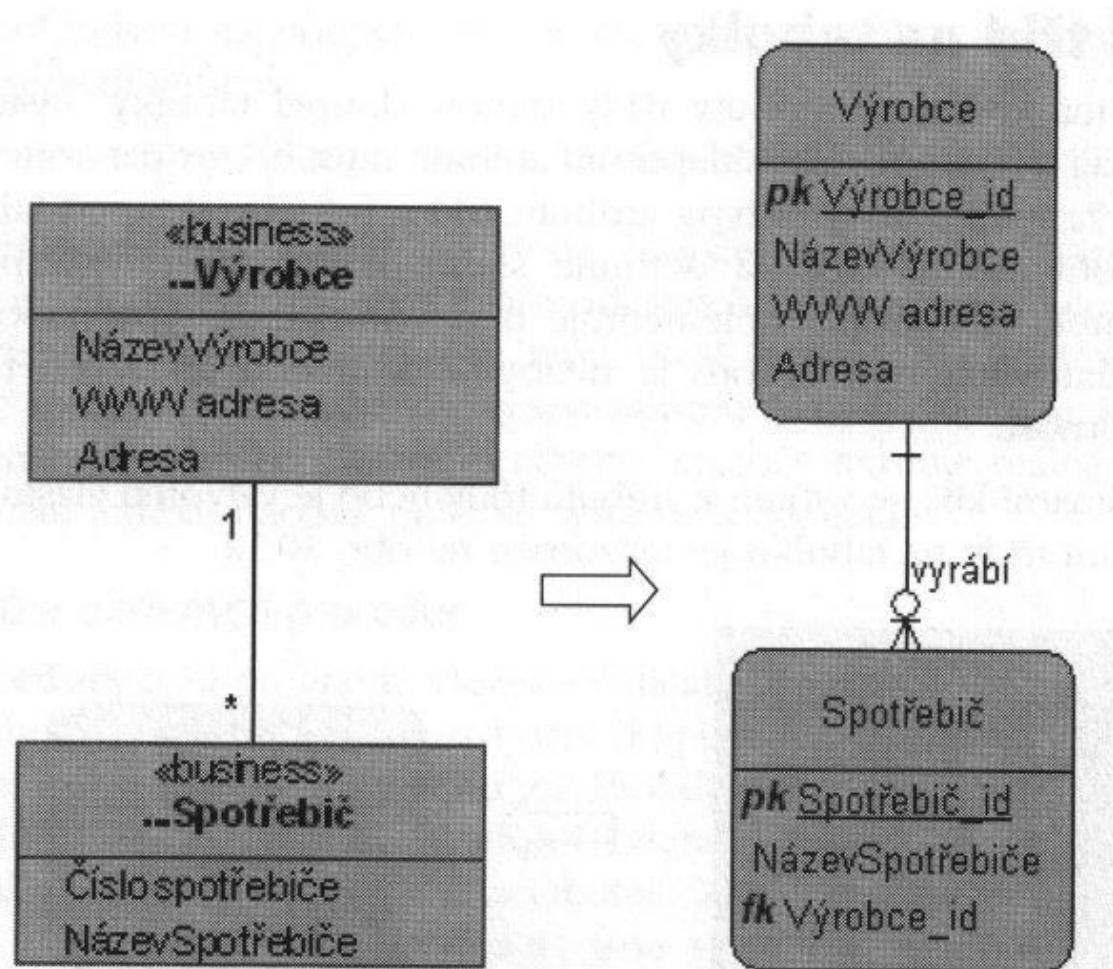
# Mapovanie asociácií 1:1

- Z dvoch tried sa vytvorí jediná tabuľka!
- Atribúty sa zlúčia do výslednej tabuľky



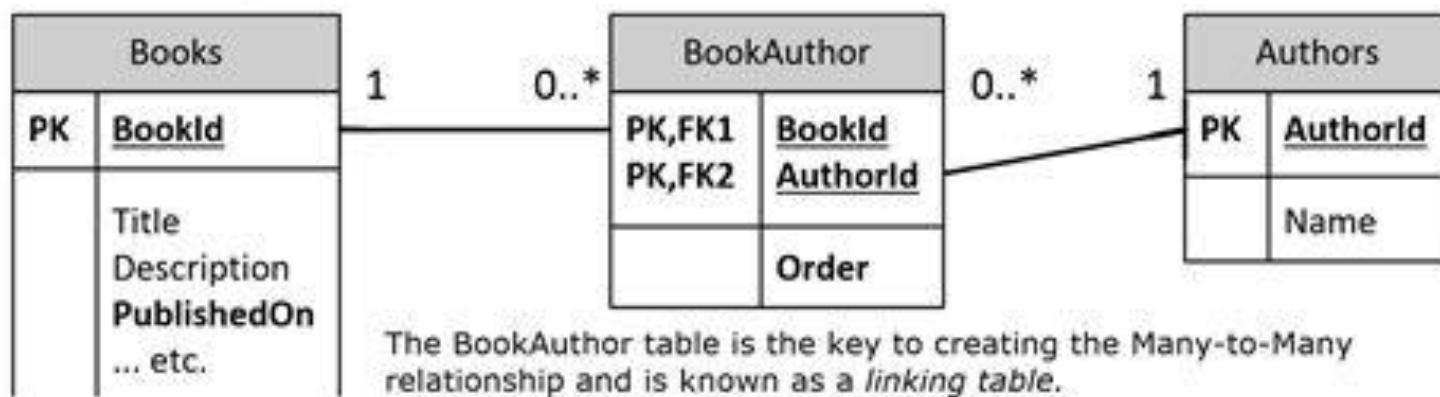
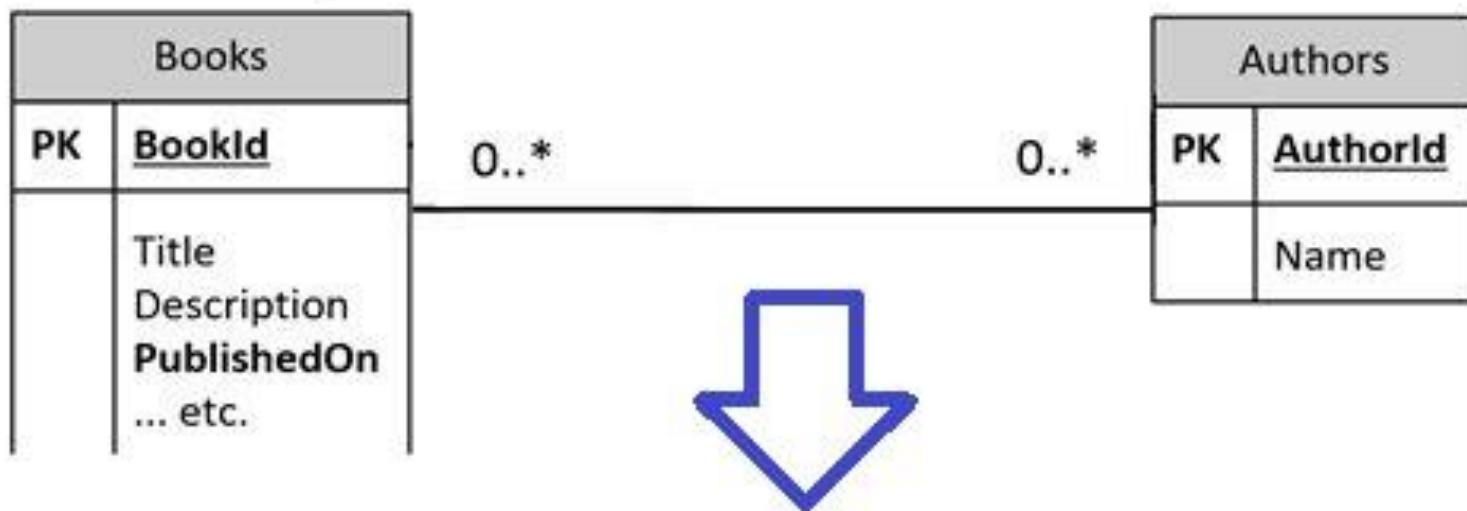
# Mapovanie asociácií 1:mnoho

- Vzťahy sú bezo zmeny
- Označia a pridajú sa kľúče (PK a FK)



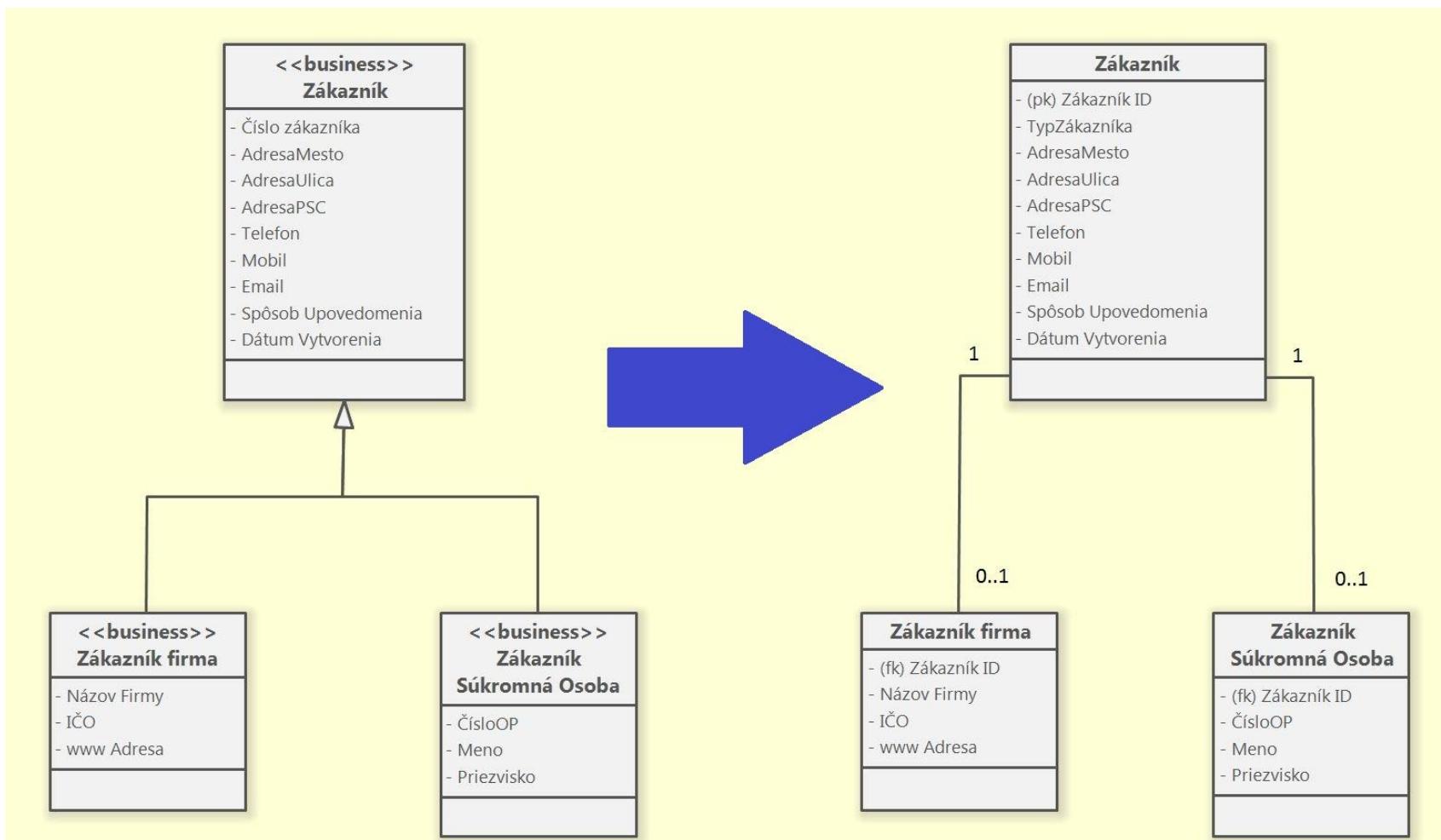
# Mapovanie asociácií mnoho:mnoho

Medzi pôvodnými triedami sa vytvorí nová „prepájacia“ tabuľka.



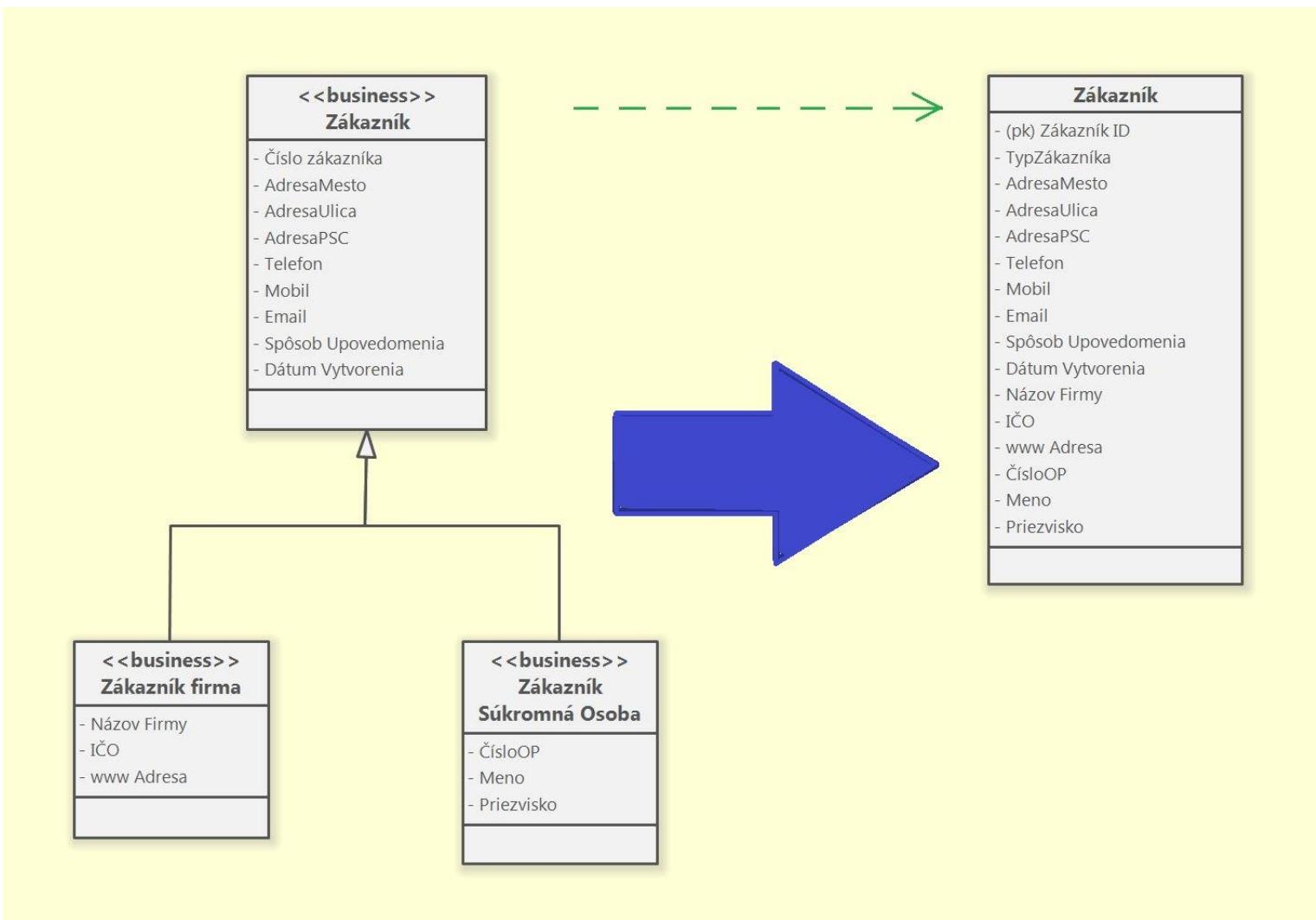
# Mapovanie generalizácie zachovaním tabuliek

- Generalizácia sa nahradí asociáciou, počet tabuliek sa nemení
- Zriedkavé**, lebo inštancie dolných tried sú rozložené v 2 tabuľkách!



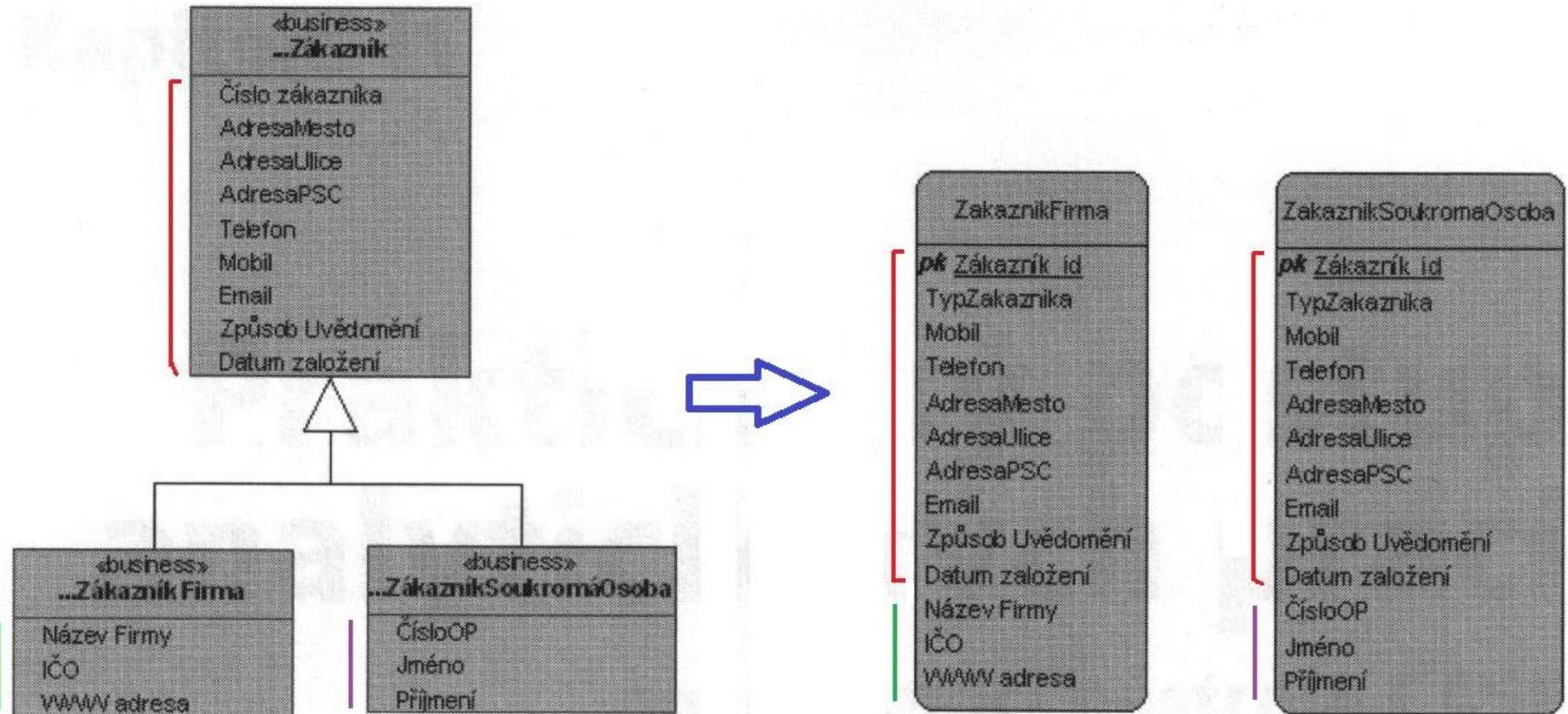
# Mapovanie generalizácie do 1 tabuľky (t.j. do nadriedy)

Ak je málo atribútov v podriedach, tak sa zlúčia všetky atribúty spolu.



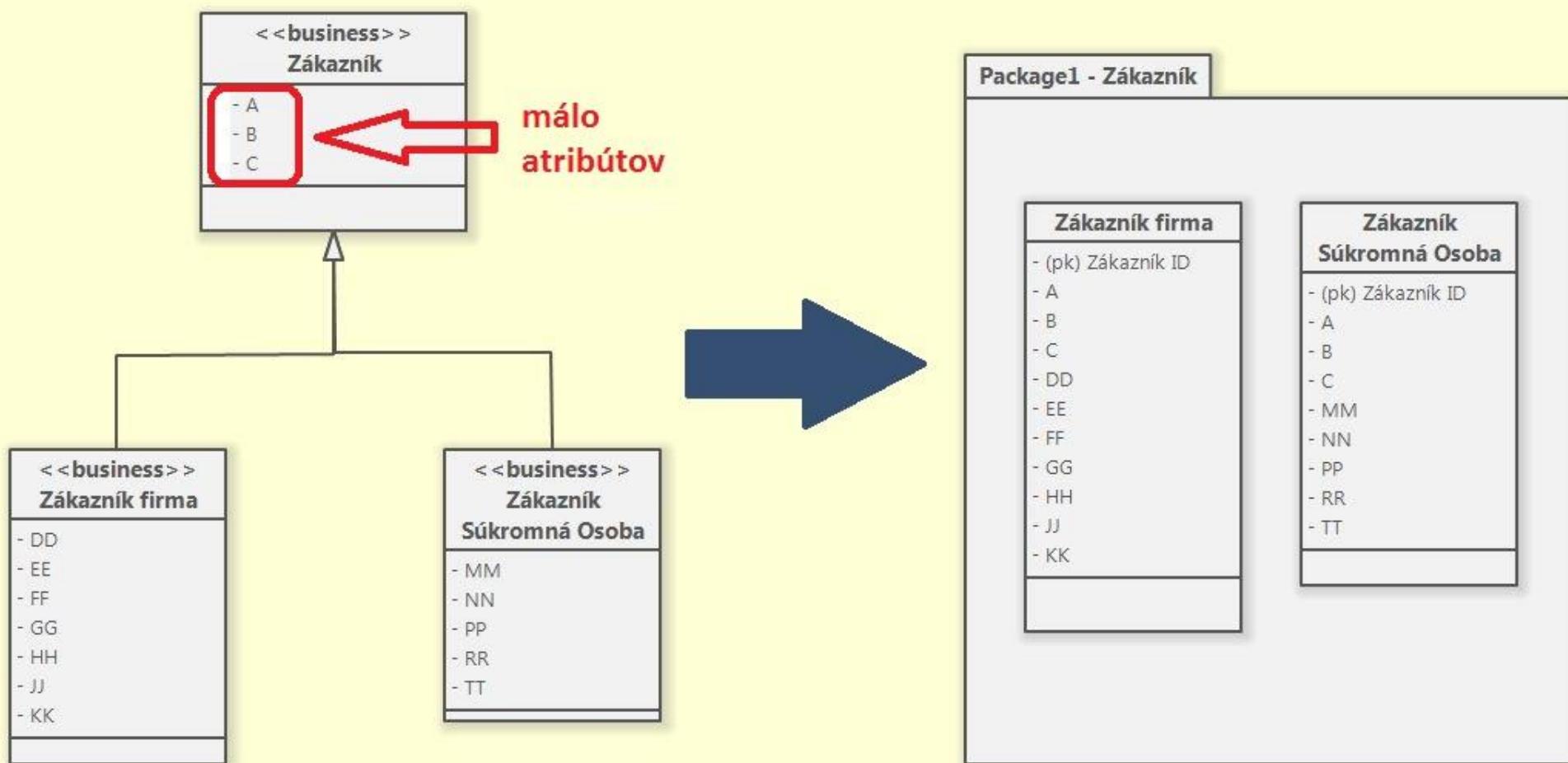
# Mapovanie generalizácie do viac tabuľiek (t.j. do podried)

Ak je **vel'a atribútov** v podriedach, tak sa atribúty nadriedy pridajú do všetkých tabuľiek. (Pozri aj ďalší obrázok.)



# Mapovanie generalizácie do viac tabuľiek (t.j. do podtried, symbolicky)

Ak je veľa atribútov v podtriedach, tak sa atribúty nadtriedy pridajú do všetkých tab. (Atribúty sú nazvané symbolicky; package je nepovinný.)



# Mapovanie ostatných vzťahov

- Kompozície a agregácie sa mapujú rovnako ako asociácie, lebo sú špeciálnymi prípadmi asociácií.
- Násobnosti (1:1, 1:mnoho, ...) sa mapujú rovnako ako pri asociáciách.