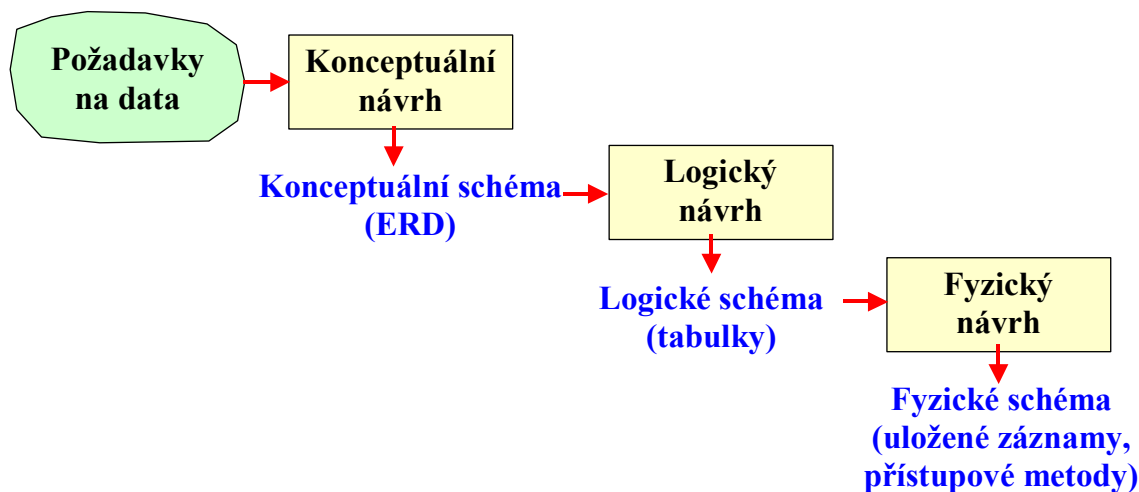

2 Konceptuální modelování a návrh databáze

2.1. Úloha konceptuálního modelování v procesu návrhu databáze ...	2
2.2. E - R modely	6
2.3. Doporučení pro modelování a tvorbu ER diagramu	22
2.4. Transformace ER diagramu na tabulky relační databáze	32
2.5. Transformace objektového modelu (diagramu tříd).....	40
Literatura.....	43

2.1. Úloha konceptuálního modelování v procesu návrhu databáze

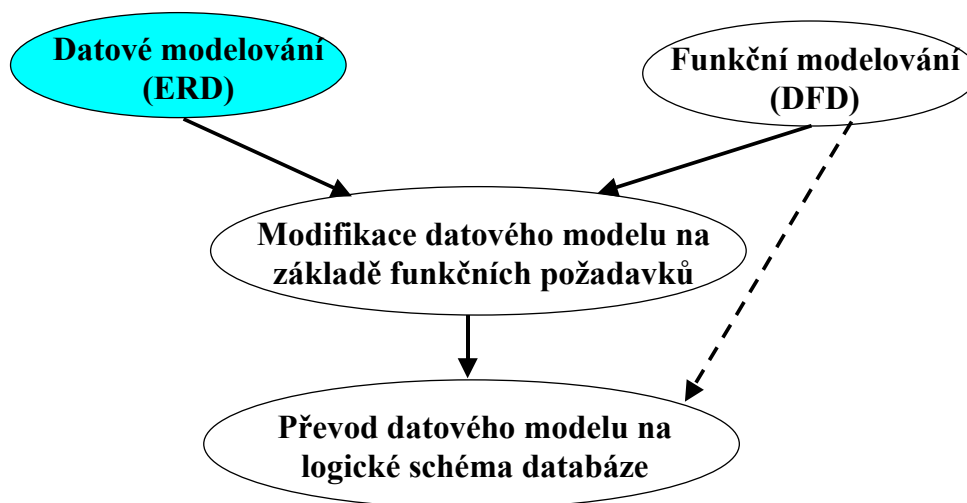
Konceptuální modelování - fáze datové, případně objektové analýzy využívající modelů založených na objektech aplikační domény.

- Fáze návrhu databáze

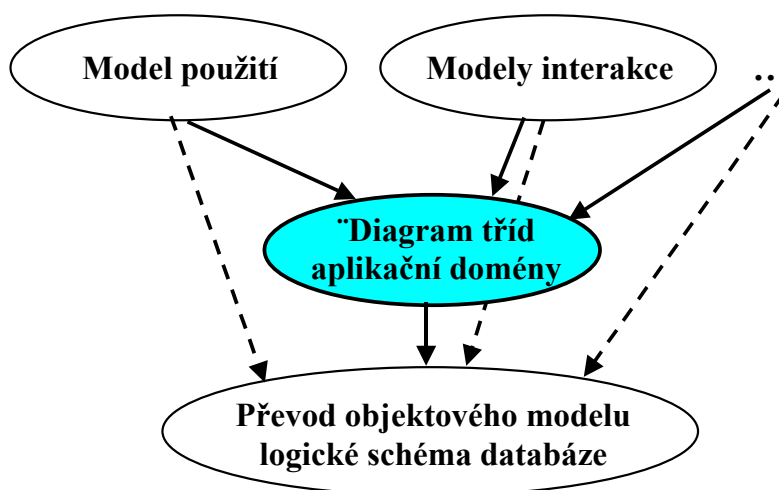


Přístup:

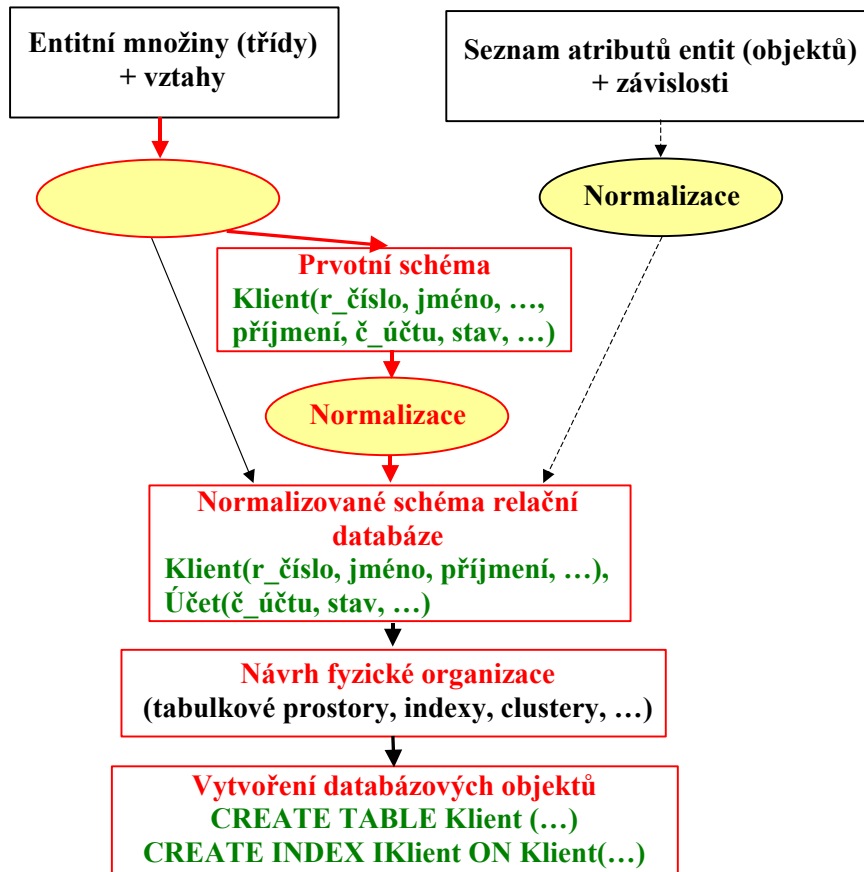
- **strukturovaný** (klasický): východiskem pro návrh databáze je **ER model**



- **objektově-orientovaný**: východiskem pro návrh databáze je **diagram tříd**

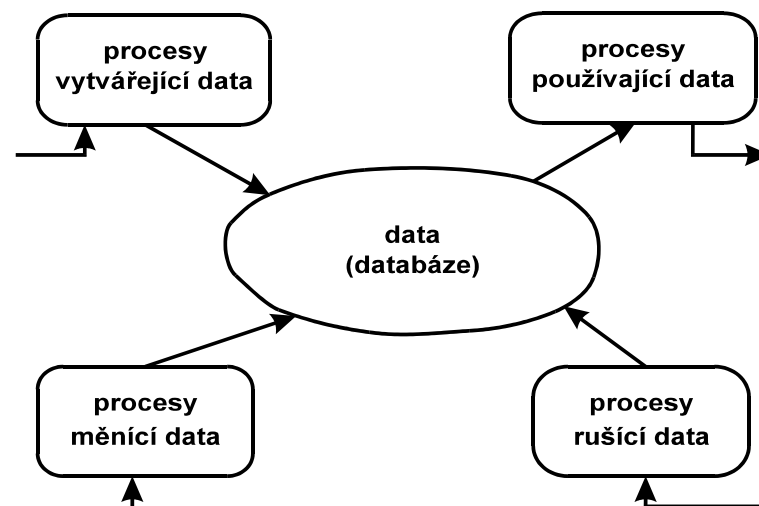


• Přístup k návrhu databáze:

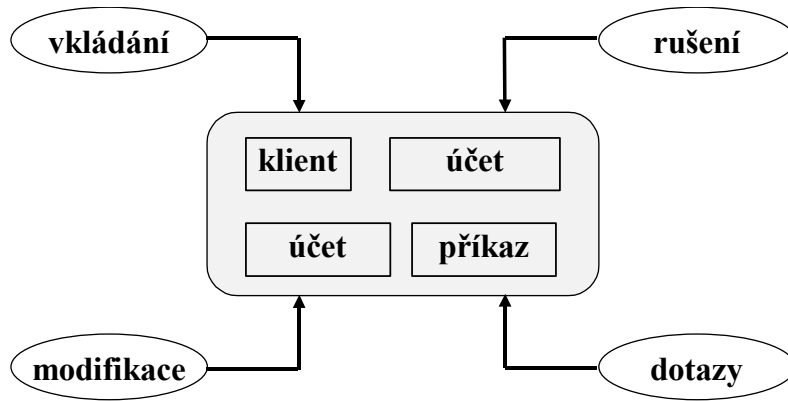


2.2. E - R modely

ER model je založen na chápání světa jako množiny základních objektů - **entit** (*Entity*) a **vztahů** (*Relationship*) mezi nimi. Popisuje data "v klidu", neukazuje, jaké operace s daty budou probíhat. Někdy se označuje také jako ERA – třetím základním prvkem modelu jsou **atributy** (*Attributes*).



Př)



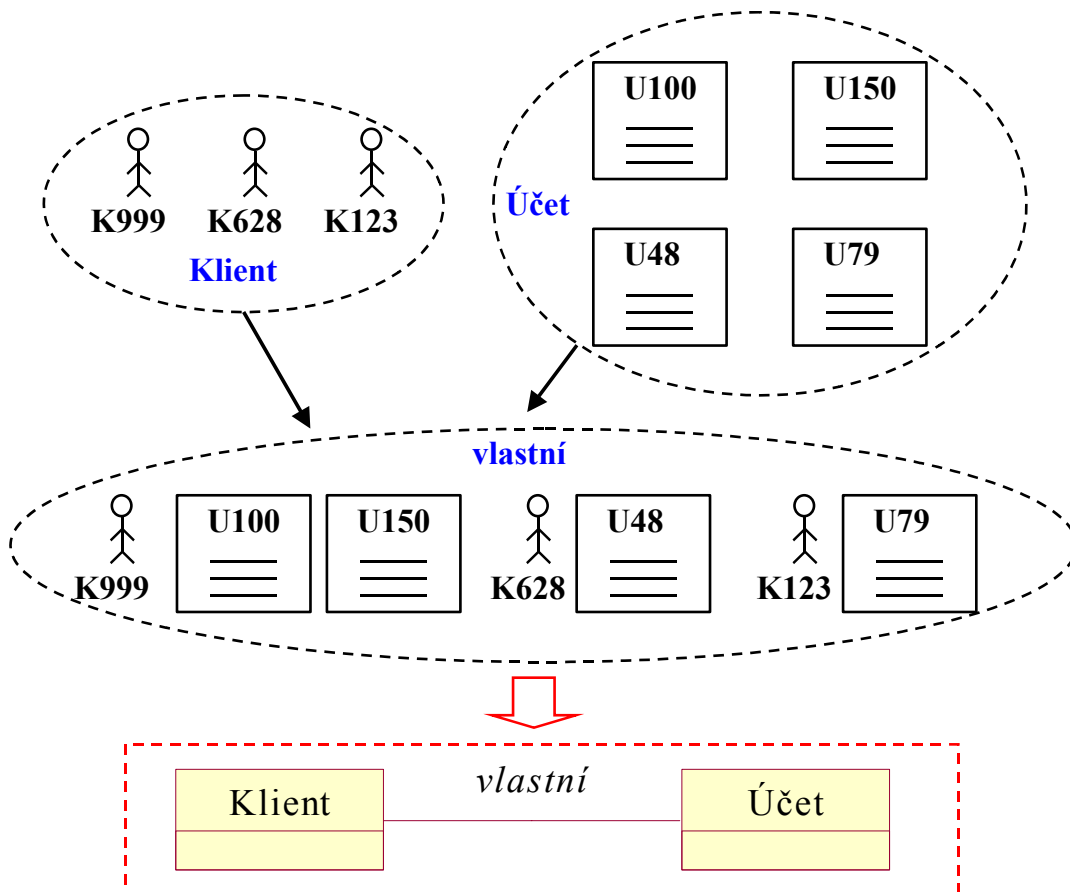
• Základní pojmy

Entita – „věc“ nebo objekt reálného světa rozlišitelný od jiných objektů, o níž chceme mít informace v DB.

Př) Klient banky s identifikačním číslem 999, účet s číslem účtu 100.

Entitní množina - množina entit téhož typu, které sdílí tytéž vlastnosti neboli *atributy*.

Př) Klient, Účet



Atribut - vlastnost entity, která nás v kontextu daného problému zajímá a jejíž hodnotu chceme mít v DB uloženu.

Př) Klient: čísloKlienta, jméno, příjmení, adresa, ...

Doména atributu - obor hodnot atributu.

Vztah – asociace mezi několika entitami.

Př) Klient s číslem klienta K999 vlastní účet s číslem účtu U100.

Vztahová množina - množina vztahů téhož typu, které sdílí tytéž vlastnosti.

Př) Klient vlastní Účet – pro vztah mezi entitami typu Klient a Účet

Identifikátor (primární klíč) entitní nebo vztahové množiny – atribut, jehož hodnota je v rámci dané množiny jednoznačná a neredukovatelná (v případě složeného atributu – viz dále).

Poznámka.: Často se používá označení entita i ve významu entitní množiny, entita se potom zpravidla označuje jako instance entity. Analogicky pro vztahové množiny a vztahy.

• Typy atributů

➤ **Jednoduché** (simple) a **složené** (composite) atributy

Př)

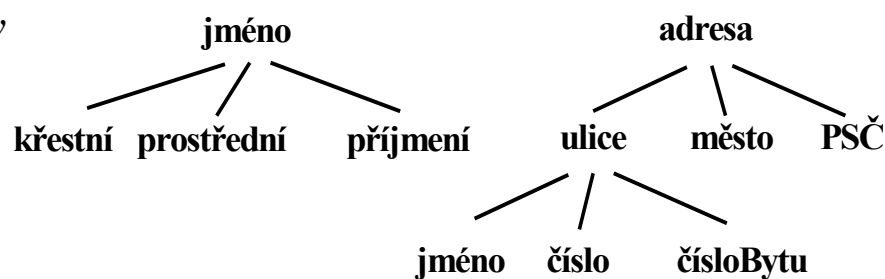
Entitní množina

Klient

Složené atributy

Složky

Složky



- **Jednohodnotové** (single-valued) a **vícehodnotové** (multiple-valued) (někdy také *opakující se atributy*)

Př)

Entitní množina

Klient

Vícehodnotový atribut

telefon

Hodnoty atributu

číslo1, číslo2, číslo3, ...

- **Povolující prázdnou hodnotu**

Mohou nabývat speciální hodnoty **NULL**.

Význam hodnoty NULL:

- hodnota chybí (missing) - existuje, ale my ji neznáme,
Př) **datum narození**
- hodnotu neznáme (unknown) – nevíme, jestli vůbec existuje
Př) **čísloBytu**

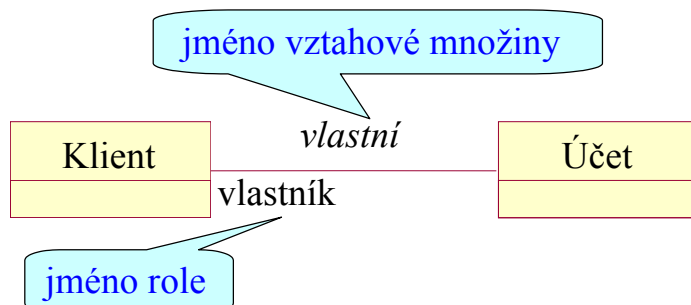
- **Odvozené**

Hodnotu lze odvodit od jiných atributů nebo entit.

Př) **věk, početDispOsob**

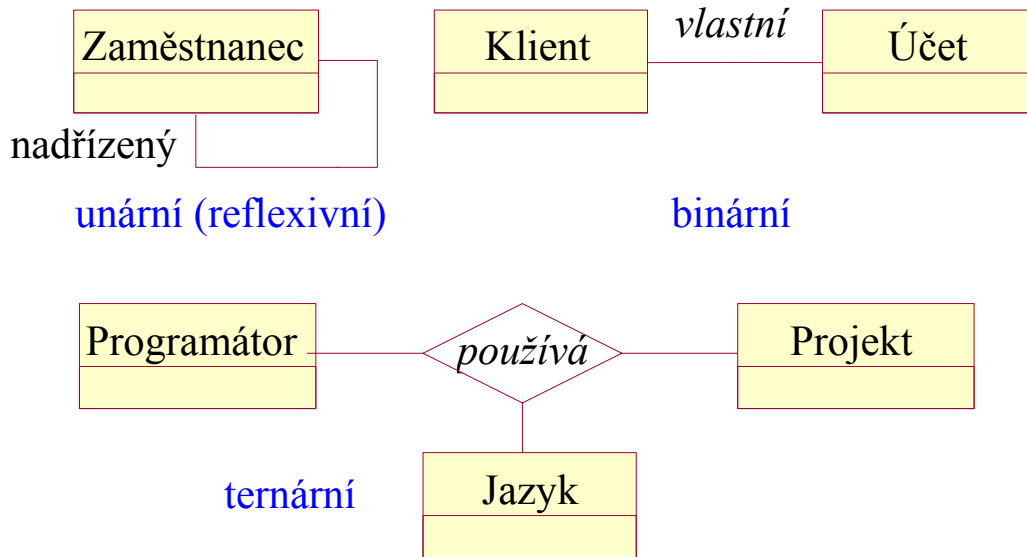
- **Identifikátor (primární klíč) entitní množiny**
 - entity a vztahy musí být identifikovatelné
 - hodnota identifikátoru musí být unikátní (a minimální)
 - identifikátorem je jednoduchý nebo složený atribut
- **Parametry vztahů**
 - **Jméno vztahové množiny, jméno role**
Vyjadřuje význam vztahu.

Př)



➤ **Stupeň**

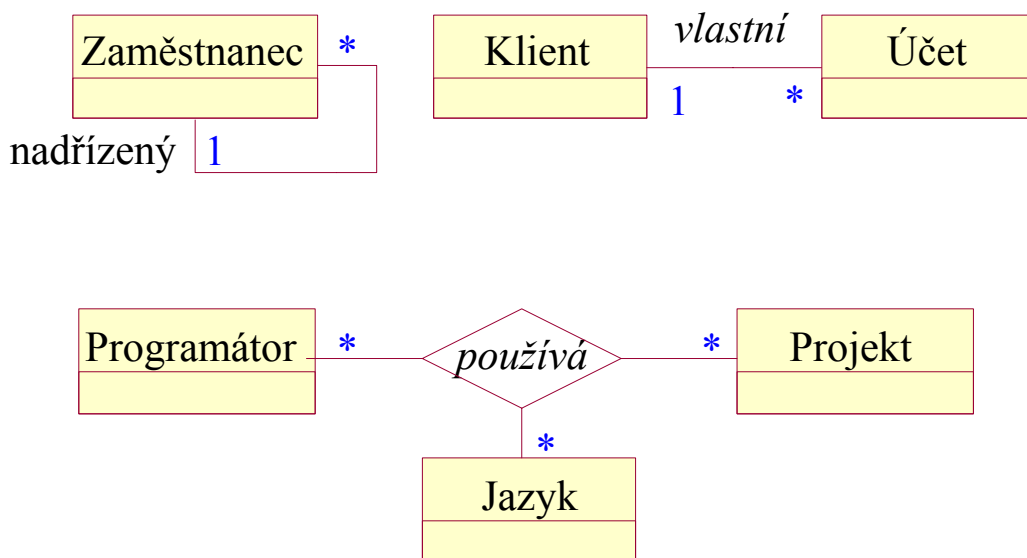
Př)



➤ **Kardinalita** (cardinality) (maximální kardinalita),

Maximální počet vztahů daného typu (vztahové množiny), ve kterých může participovat jedna entita (1,M, případně přesněji).

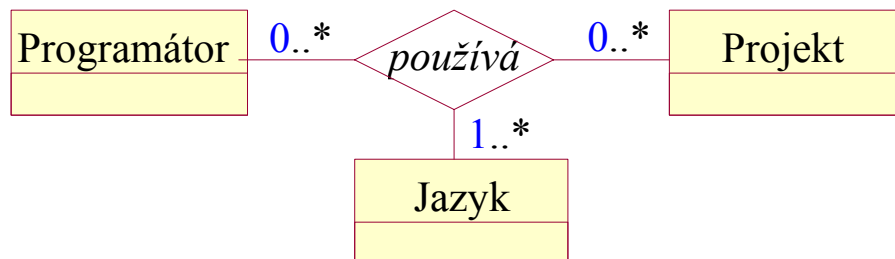
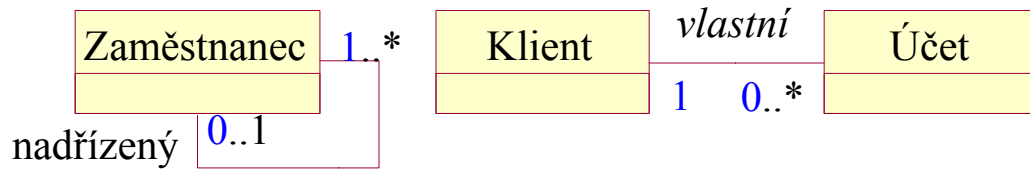
Př)



- **Členství** (membership) (také účast (participation), minimální kardinalita)

Vyjadřuje minimální počet vztahů daného typu (vztahové množiny), ve kterých musí participovat jedna entita (0 – **volitelné**/1 – **povinné**), resp. účast entitní množiny ve vztahu (**částečná** (partial)/**úplná** (total)).

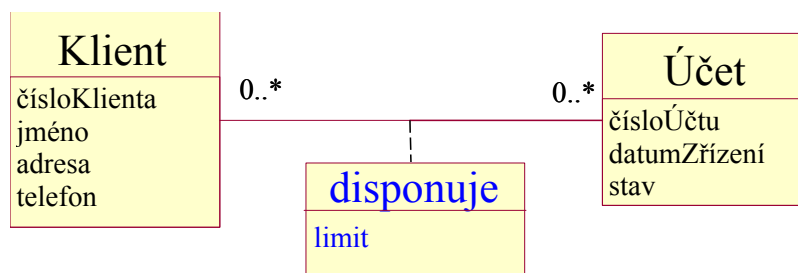
Př)



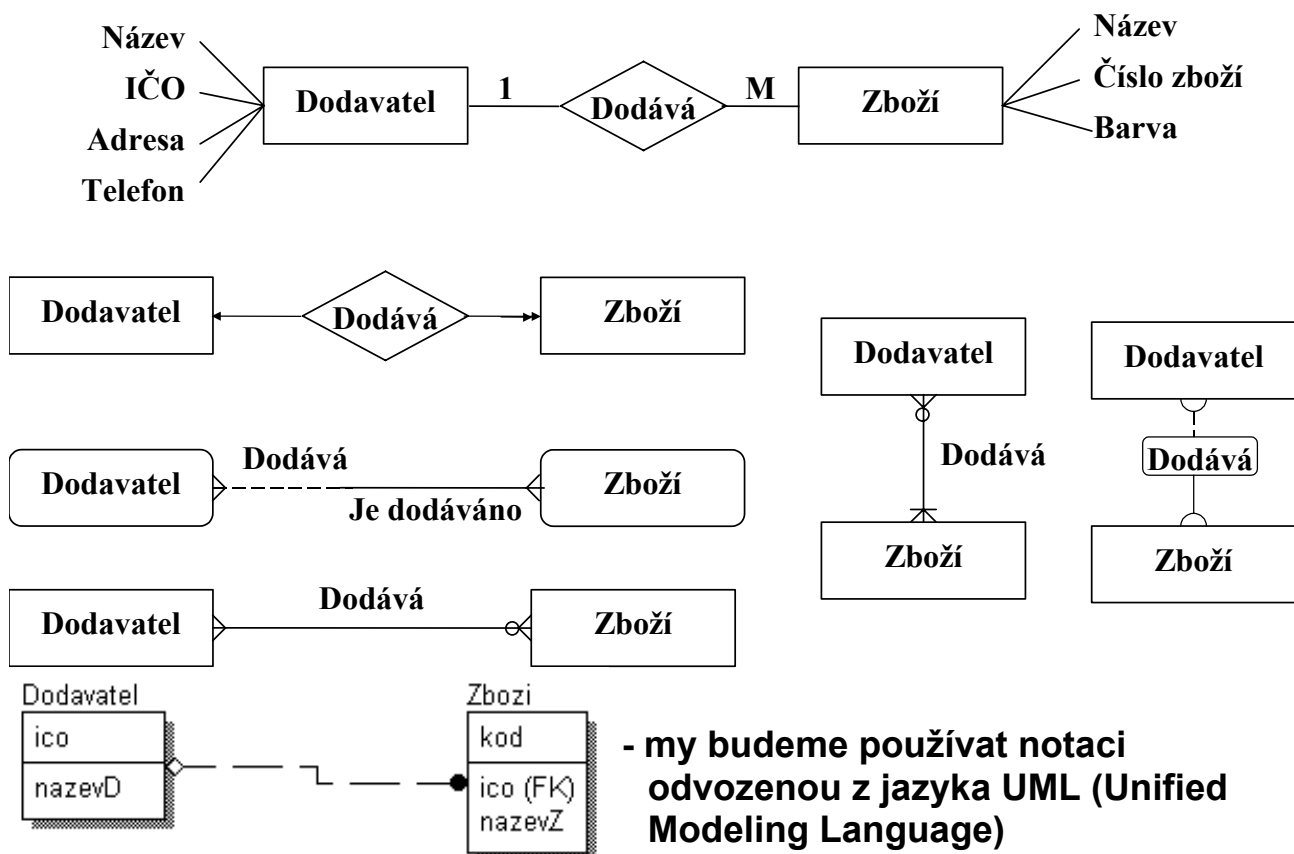
Kardinalita i členství představují omezení (constraint)
Jiným důležitým typem omezení je existenční závislost.

Př) Klient (vlastník účtu) a Účet

- **Atributy** vztahu



• Používané notace pro kreslení ER diagramu

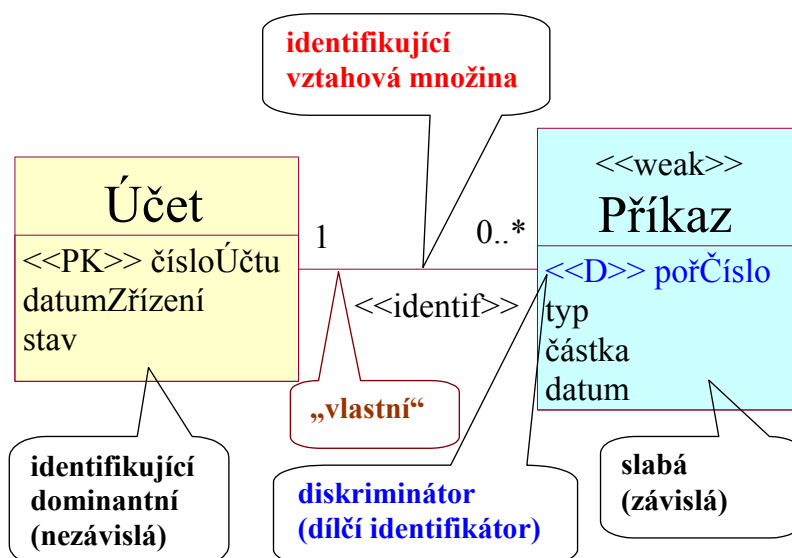


• Rozšíření klasického ER modelu

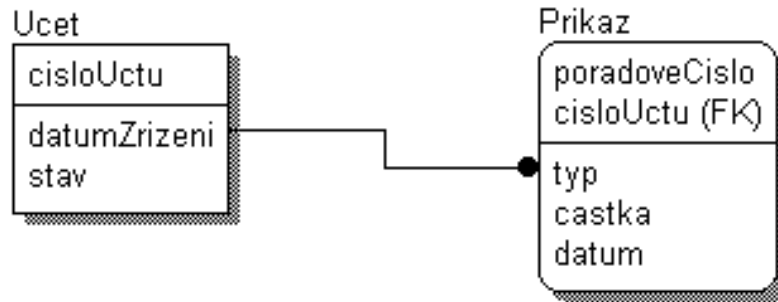
➤ **Slabé (weak) entitní množiny**

Silná (strong) entitní množina – má identifikátor tvořený vlastními atributy.

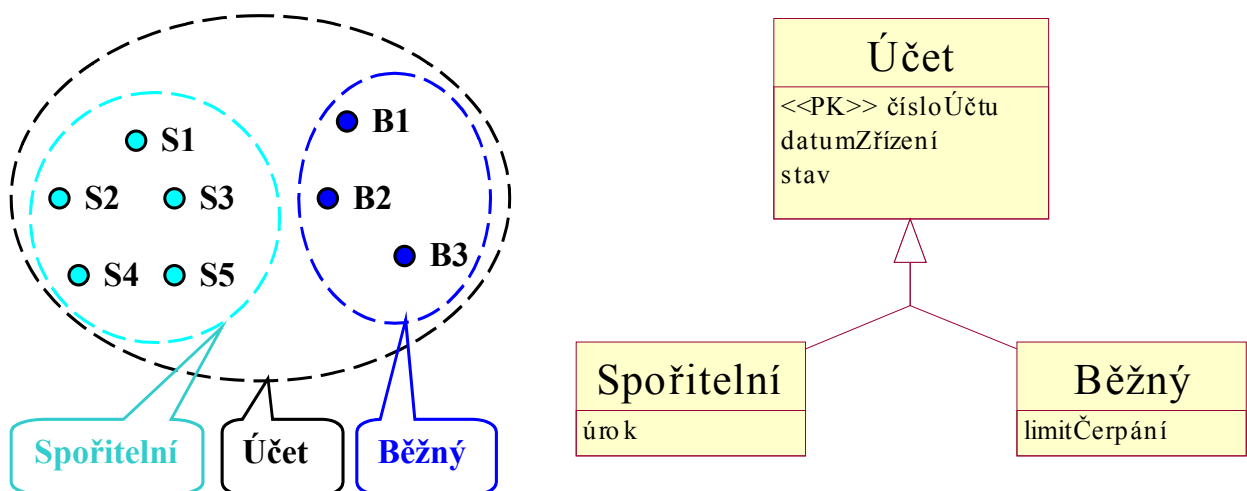
Slabá entitní množina – nemá identifikátor tvořený z vlastních atributů, nýbrž obsahuje identifikátor jiné entitní množiny (pouze jedné), na níž závisí tzv. dominantní).



- identifikátor = identifikátor_dominantní + diskriminátor
- existenční závislost slabé množiny na identifikující



➤ Generalizace/specializace (ISA vztah)

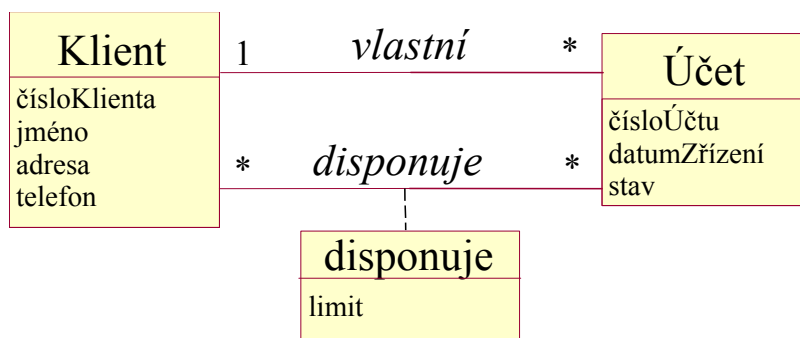


- vyjádření rozdílů (atributy, vztahy)- specializace
- pojmy entitní množina vyšší/nížší úrovně (nadtrída/podtrída)
- dědičnost atributů a účasti ve vztahových množinách, hierarchie generalizace (viz OO přístup)
- identifikátor entitních množin nižší úrovně je stejný jako vyšší

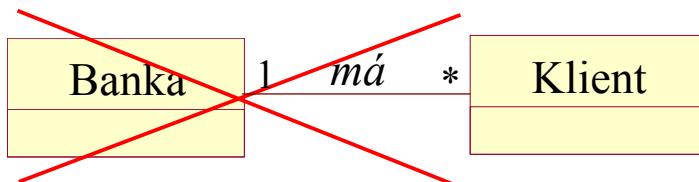
- Omezení generalizace/specializace
 - příslušnost – disjunktní/překrývající se
 - úplnost – totální/částečná

2.3. Doporučení pro modelování a tvorbu ER diagramu

- Jména
 - srozumitelná, musí vyjadřovat význam entitních a vztahových množin
 - entitní množiny: podstatná jména
 - vztahové množiny: slovesa, předložky
 - je-li jméno vztahové množiny jasné ze jmen entitních množin, není nutné uvádět
- Několik různých vztahových množin mezi stejnými entitními
 - nutno použít jméno vztahové množiny nebo jméno role.



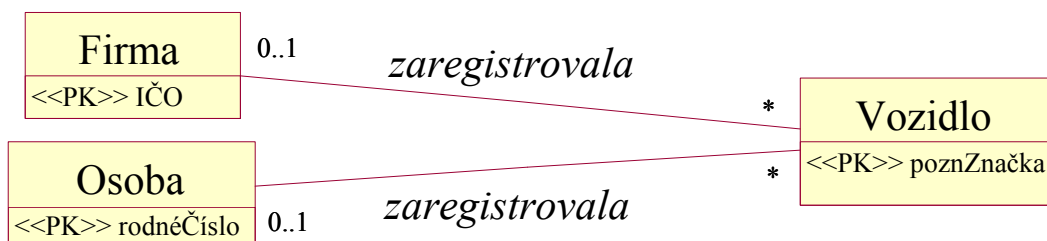
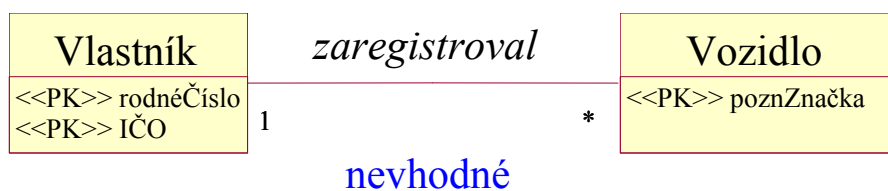
- Celkový systém by neměl být zahrnut do ERD



- Identifikátor entitní množiny

- identifikátorem je jednoduchý nebo složený atribut
- situace, kdy používáme složené identifikátory:
 - přirozená identifikace spojením několika atributů
 - vazební entitní množiny nahrazující vztahové s kardinalitou M:M
 - u slabých entitních množin
 - při modelování časových změn
- unikátnost hodnoty jen v rámci vyvíjeného systému (ne celého světa)

- Nevytvářet entitní množinu s různými identifikátory entit na základě jejich výskytu ve stejné vztahové množině.



vhodné
další možnost - generalizace

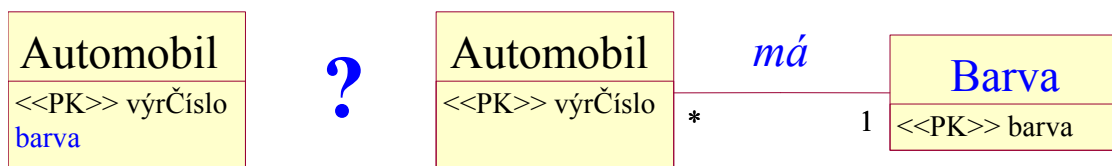
- **Identifikátor vztahové množiny**

- zpravidla složený z identifikátorů participujících entitních množin, případně ještě v kombinaci s některým z atributů vztahu, pokud kombinace identifikátorů nepostačuje.

Př) disponuje ... (čísloKlienta, čísloÚčtu)

disponuje ... (čísloKlienta, čísloÚčtu, platnostOd), za předpokladu, že vztah má např. atribut limit, jehož výši lze měnit a je potřeba uchovávat historii změn

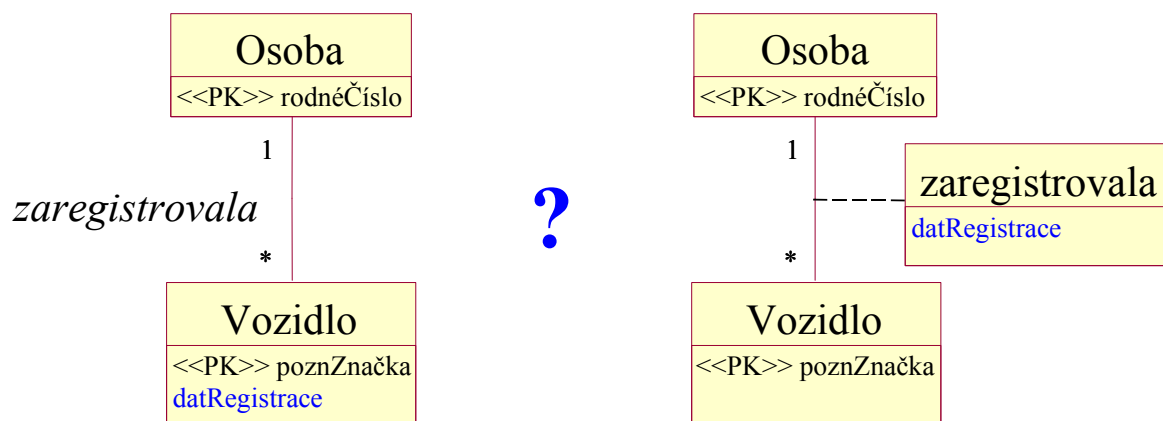
- **Entitní množina nebo atribut?**



Pravidlo: Je-li hodnota atributu důležitá, i když neexistuje žádná entita s touto hodnotou jako vlastností, pak bychom ji měli modelovat jako entitu.

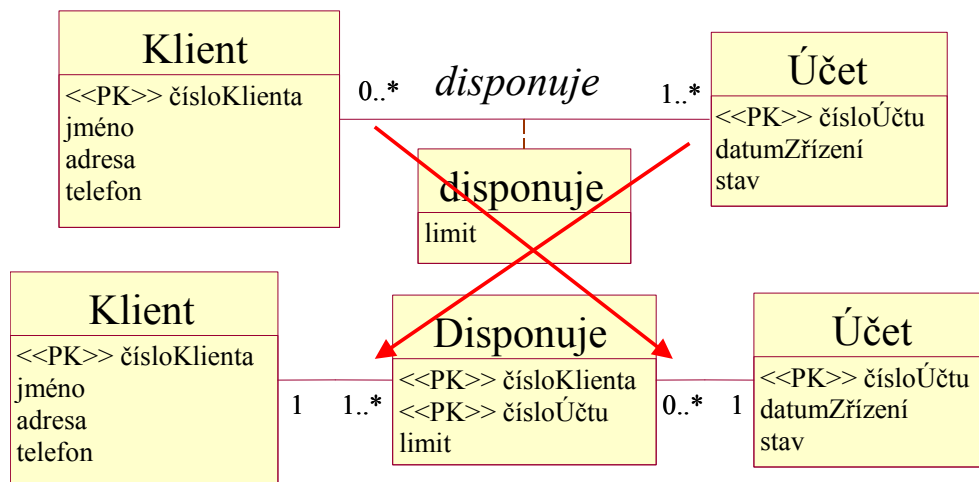
- Z modelování entitní množinou pak vyplývá omezení na hodnotu odpovídajícího atributu.

- **Atributy a vztahy 1:M**



Pravidlo: Měli bychom dávat přednost modelování jako atributu vztahu. Hrozí-li nebezpečí, že by se mohla časem kardinalita vztahové množiny změnit na M:M, pak modelujeme jako atribut vztahu v každém případě.

- **Náhrada vztahů M:M vazební entitní množinou**

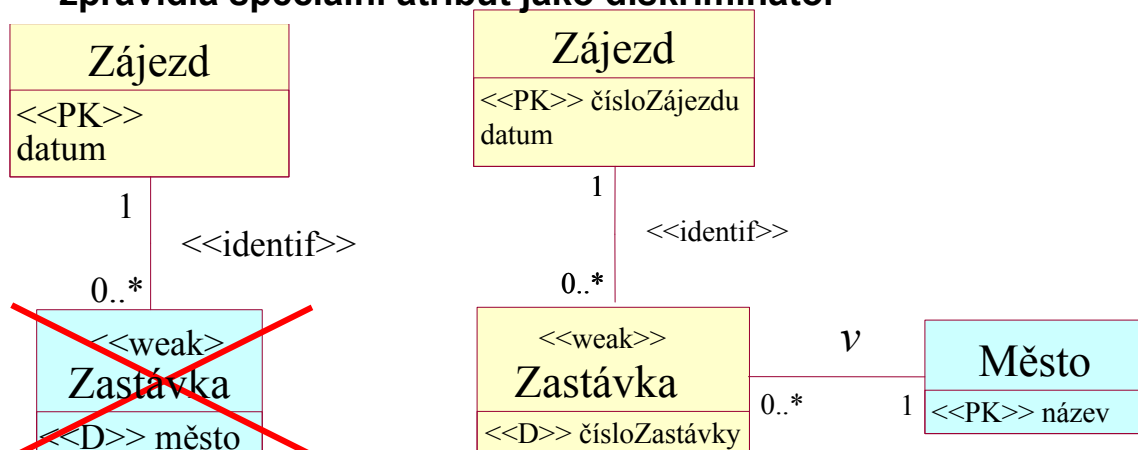


- zpravidla před transformací na schéma relační databáze

- **Doporučení pro použití slabých entitních množin**

- **Volba identifikátoru slabé entitní množiny**

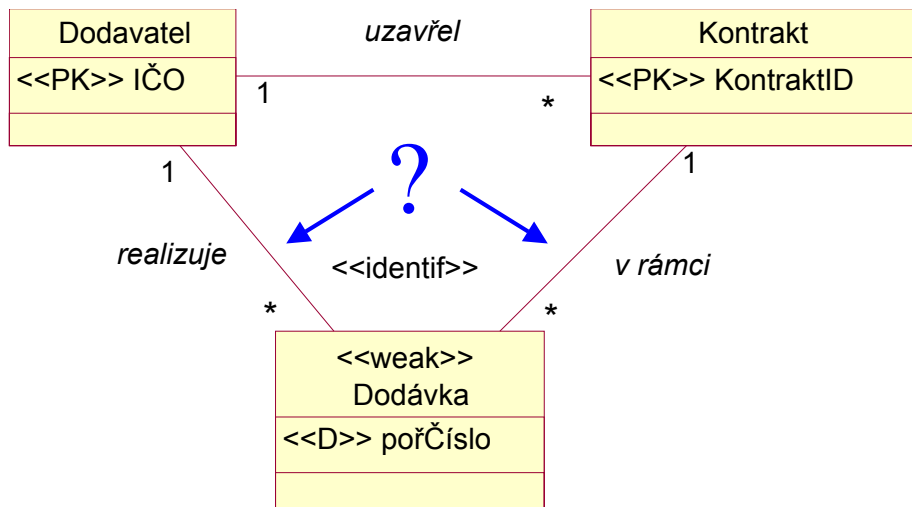
- zpravidla speciální atribut jako diskriminátor



➤ Volba správné identifikující entitní množiny

- vycházet z přirozené závislosti

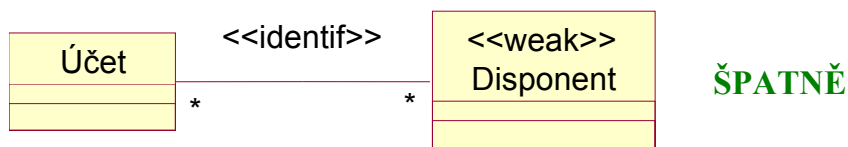
Př) Dodavatel realizuje kontrakt a v jeho rámci dodávky



➤ Vyvarování se redundance

- slabá entitní množina má jen jednu identifikující množinu, se kterou je ve vztahu 1:M → nelze takto modelovat vztahy M:M

Př) Klient – disponující – Účet



➤ Slabá nebo silná entitní množina?

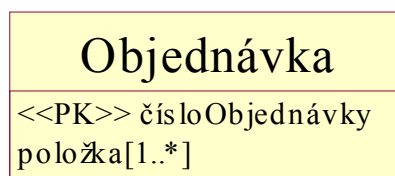
Pravidlo1: Jako slabou modelovat tehdy, kdy entita kompletně zmizí při odstranění odpovídající identifikující entity.

Př) Objednávka – PoložkaObjednávky

Pravidlo2: Cokoliv s atributem, který je jednoznačný, by nemělo být modelováno jako slabá entitní množina.

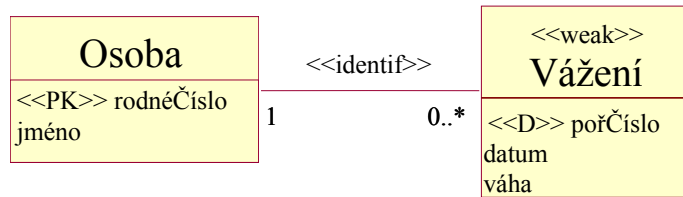
Pravidlo3: Jsme-li na pochybách, modelujeme jako silnou entitní množinu.

➤ Možnost modelování jako vícehodnotového atributu

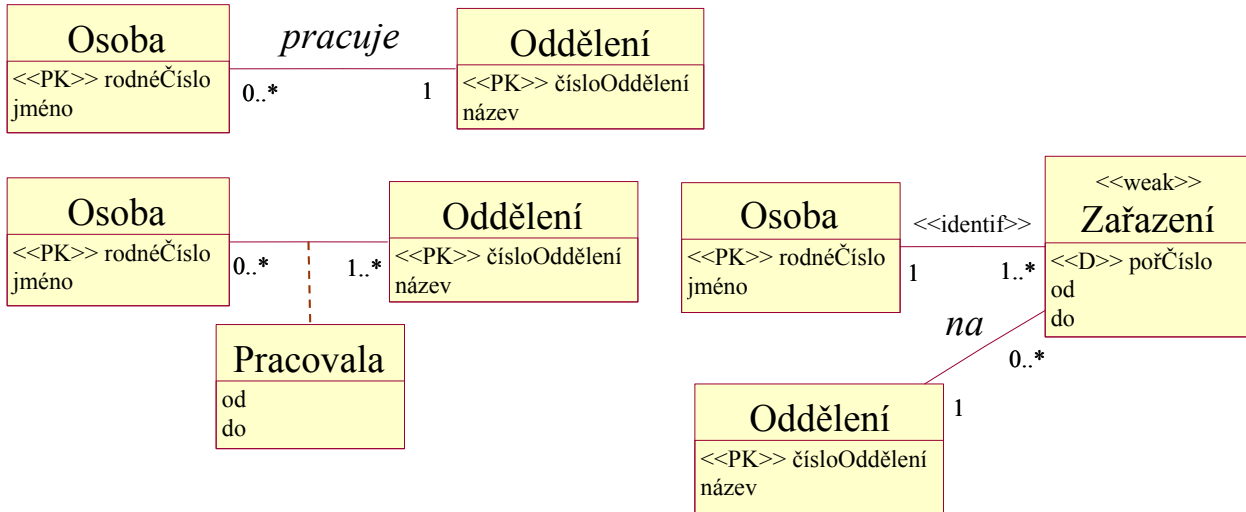


➤ Použití pro modelování časových změn

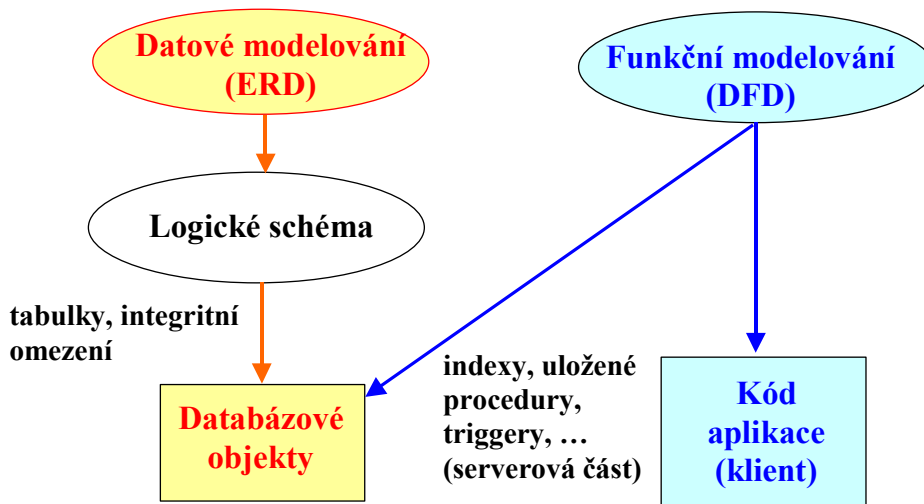
▪ Změny atributů



▪ Změny vztahů



2.4. Transformace ER diagramu na tabulky relační databáze



• Hlavní problémy špatného návrhu

- opakující se informace (redundance)
- nemožnost reprezentovat určitou informaci
- složitá kontrola integritních omezení

} formálně
BCNF, resp.
3NF

Př)r_číslo – rodné číslo disponující osoby

č_úctu	r_číslo	stav	pobočka	jmění
100	600528/0275	100000	Jánská	10000000
100	581015/9327	100000	Jánská	10000000
130	600528/0275	50000	Palackého	5000000
150	450205/3419	150000	Palackého	5000000

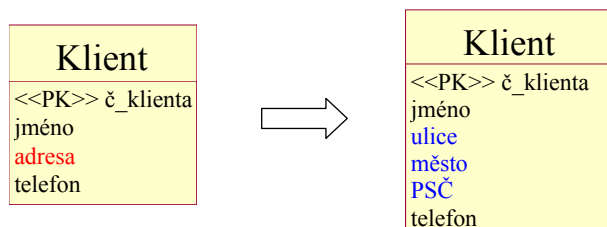
- Cíle návrhu

- vyvarování se problémů špatného návrhu
- splnění dalších kritérií, především výkonostních (nevytvářet zbytečné tabulky!)

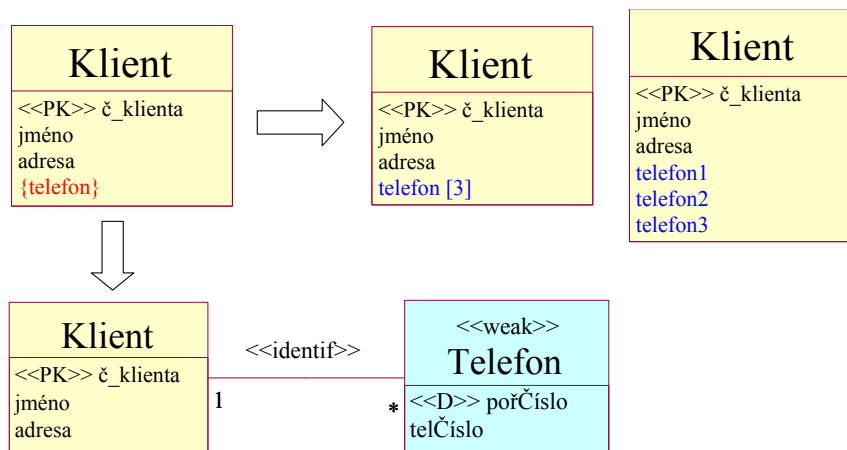
- Pravidla transformace

- Odstranění složených a vícehodnotových atributů (převod do 1NF)

- Složený atribut → několik jednoduchých (složky)

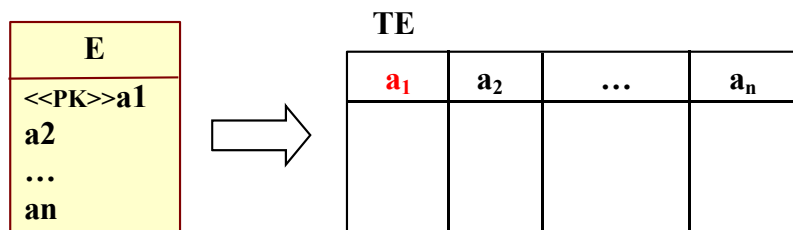


- Vícehodnotový atribut → slabá entitní množina nebo náhrada pevným počtem opakování

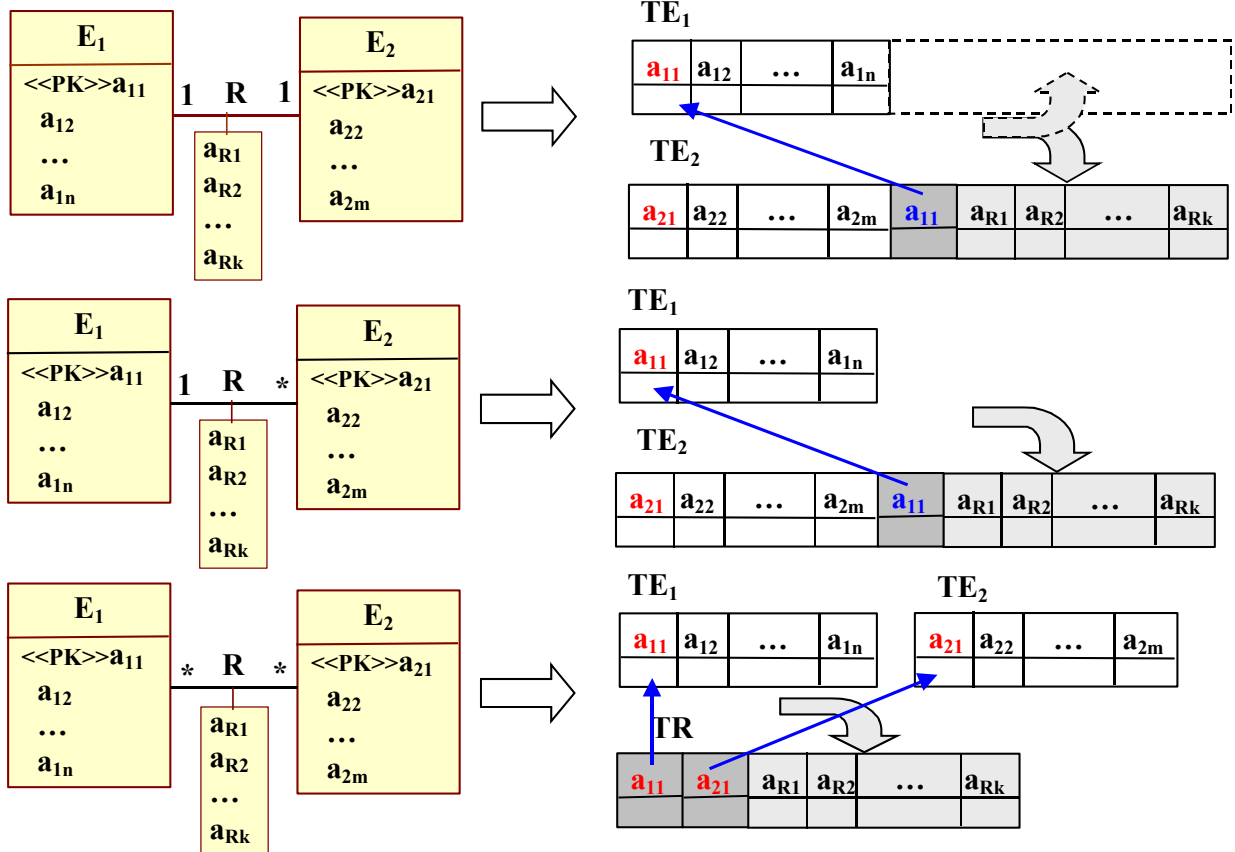


- případně lze provést transformaci a potom normalizovat

- Reprezentace silné entitní množiny

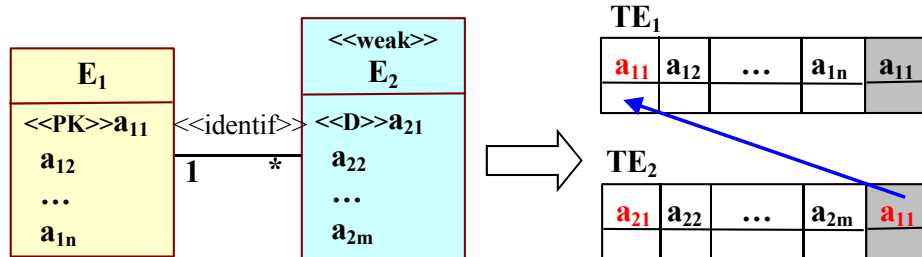


➤ Reprezentace vztahů

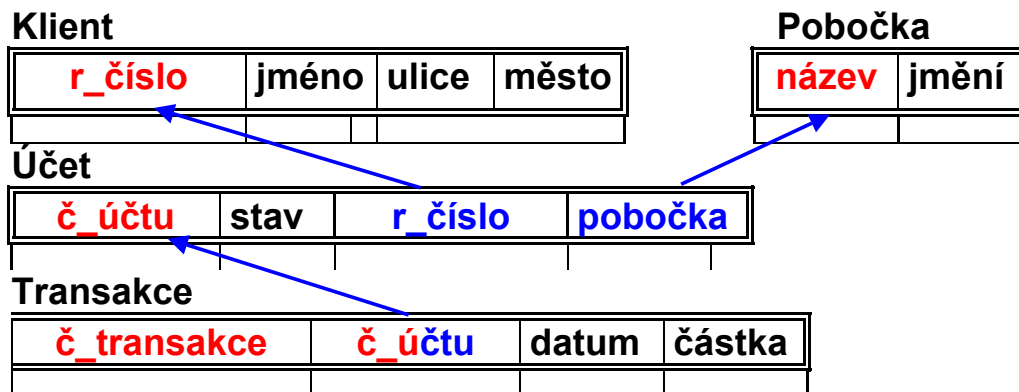
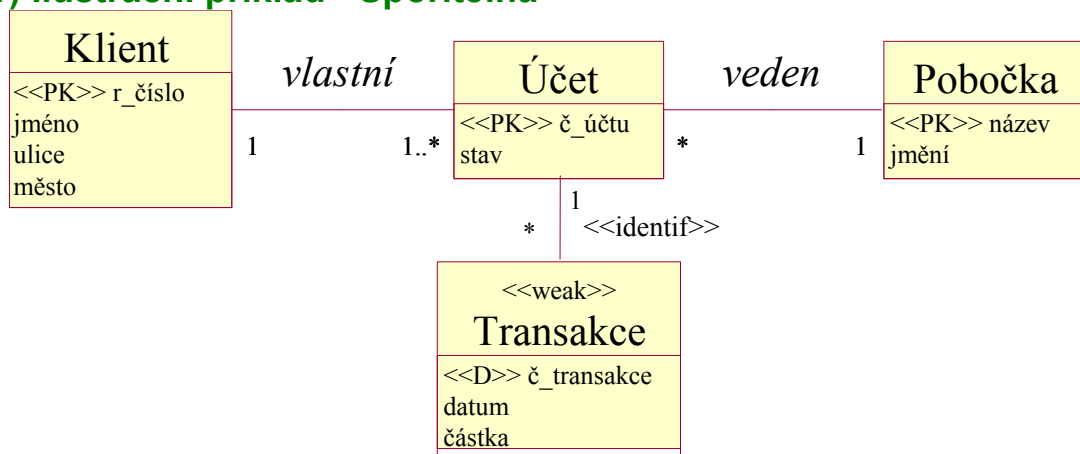


➤ Reprezentace slabé entitní množiny

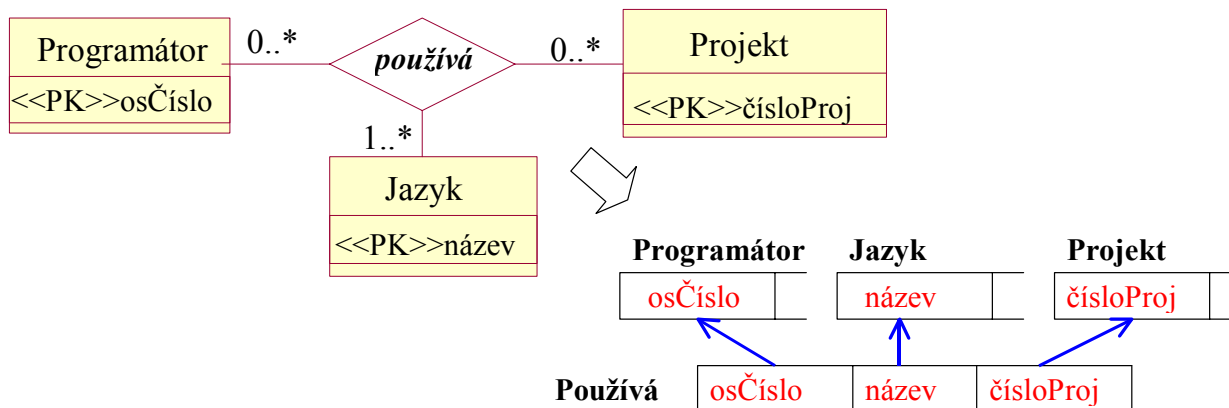
- jako silné + vztah 1:M



Př) Ilustrační příklad - Spořitelna



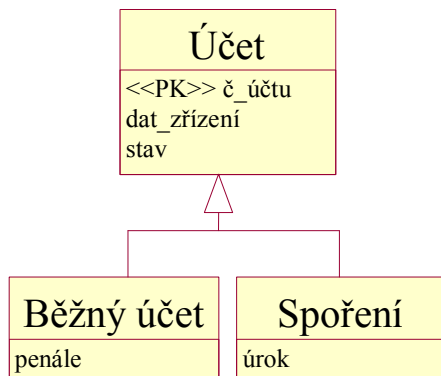
➤ Repräsentace ternárních vztahů



➤ Generalizace/specializace

- Možnosti:
 - tabulka pro nadtyp + pro podtypy s primárním klíčem nadtypu
 - pouze tabulky pro podtypy i s atributy nadtypu
 - všechno v jedné tabulce
 - rozlišení podle prázdné hodnoty nebo tzv. *diskriminátoru*.

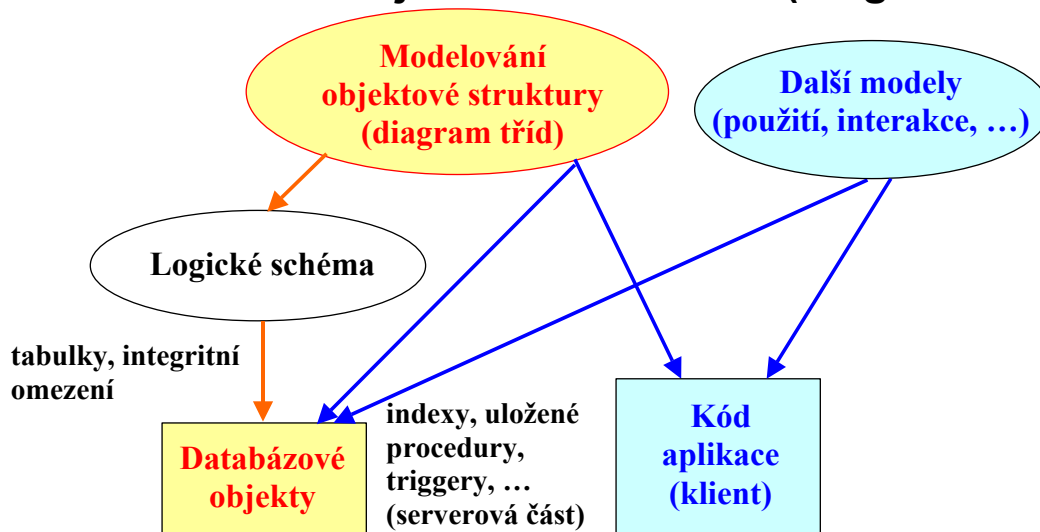
Př)



- 1) Účet(**č_úctu**, dat_zřízení, stav),
Běžný_účet(**č_úctu**, penále),
Spoření(**č_úctu**, úrok)
- 2) Běžný_účet(**č_úctu**, dat_zřízení, stav, penále),
Spoření(**č_úctu**, dat_zřízení, stav, úrok)
- 3) Účet(**č_úctu**, dat_zřízení, stav, úrok, penále) resp.
Účet(**č_úctu**, dat_zřízení, stav, **typ**, úrok, penále).

- Nutno přihlížet zejména k tomu:
 - zda jsou specializace disjunktní,
 - zda je specializace totální,
 - operace s jakými daty (jen specializace nebo i generalizace) budou prováděny.

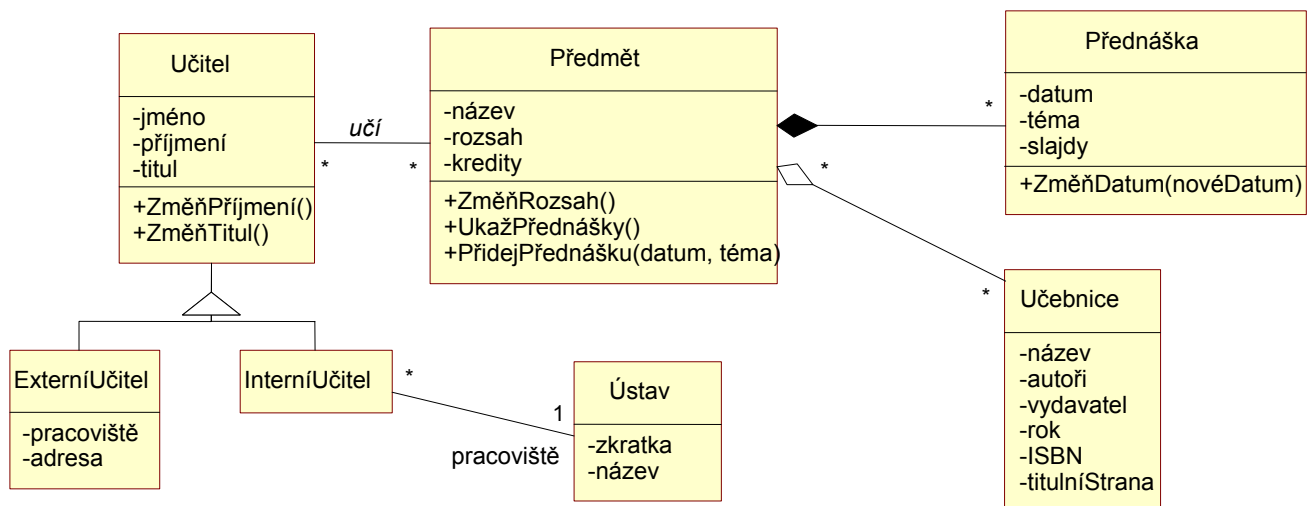
2.5. Transformace objektového modelu (diagramu tříd)



- **Problémy:**

- **operace** - při návrhu tabulek neuvažujeme (bereme v úvahu při případné optimalizaci, návrhu uložených procedur apod.)
- **identifikace pomocí OID** - neexistuje-li atribut s vlastnostmi identifikátoru → přidat
- **složené, složité, vícehodnotové atributy** - viz normalizace u ER modelu, podpora typů proměnné délky v moderních relačních systémech (VARCHAR, BIT VARYING (BLOB)),
- **generalizace/specializace**
- **agregace** – „část“ jako silná nebo slabá entitní množina
- **kompozice(zanořené objekty)** - viz složené a vícehodnotové atributy

Př)



Literatura

- 1. Silberschatz, A., Korth H.F, Sudarshan, S.: Database System Concepts. Fourth Edition. McGRAW-HILL. 2001, str. 27 – 77.**
- 2. T. Hawryszkiewicz, I.T.: Relational Database Design. An Introduction. Prentice Hall Inc. 1990. str. 85 – 152.**
- 3. Batini, C., Ceri, S., Navathe, S., B.: Conceptual Database Design. Benjamin/ Cummings. 1992. s. 460.**
- 4. Pokorný, J.: Databazová abeceda. Science, Veletiny, 1998, str. 73 – 76, 191 – 196.**