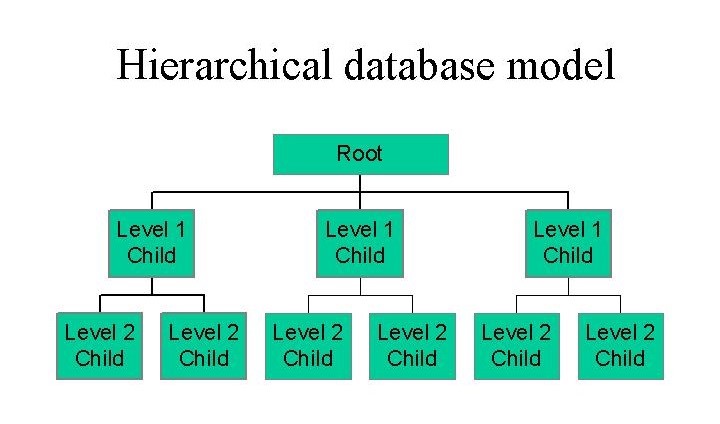
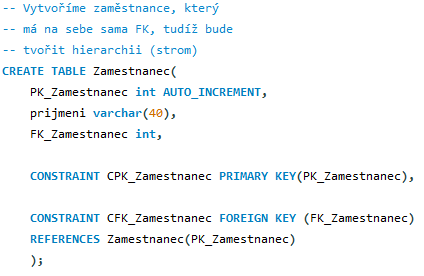
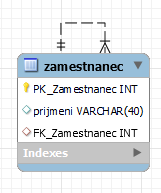
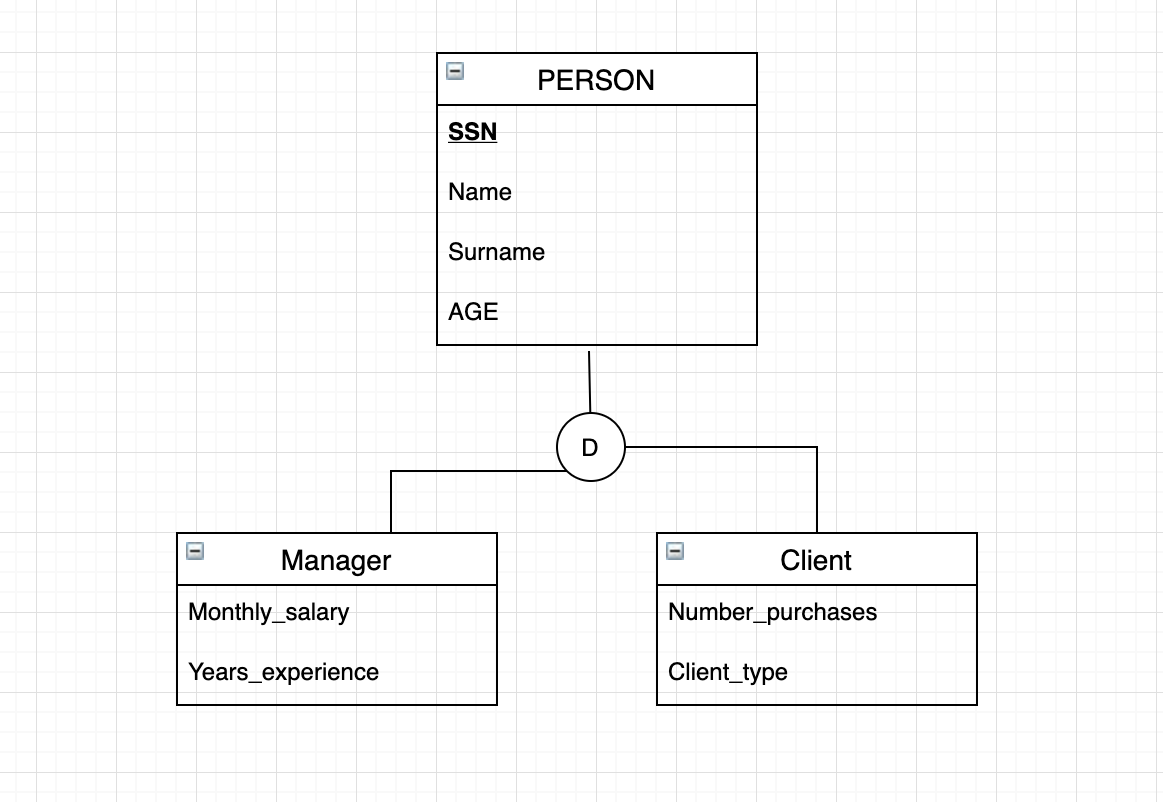
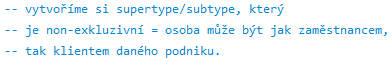
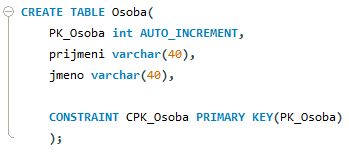
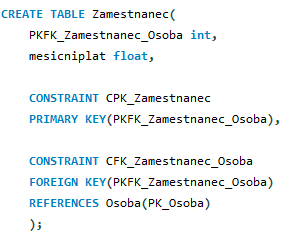
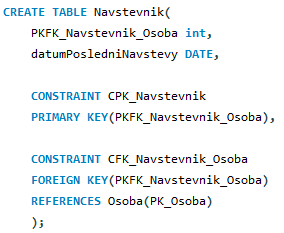
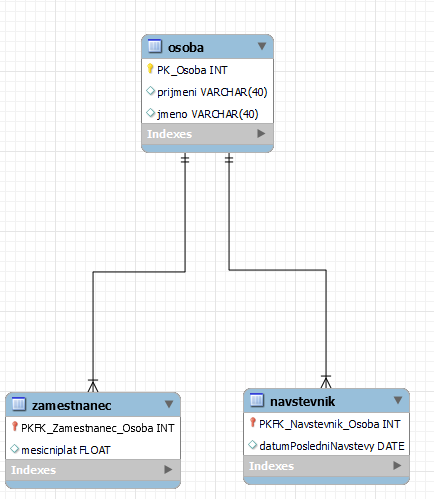
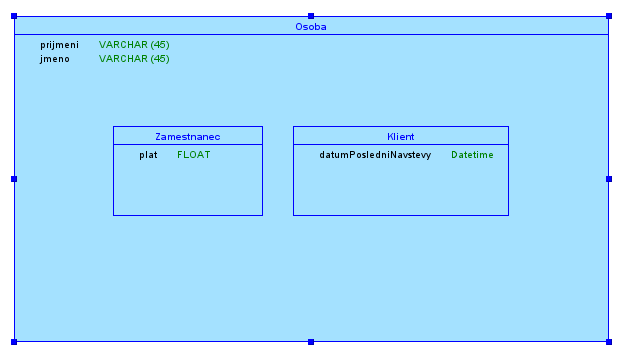
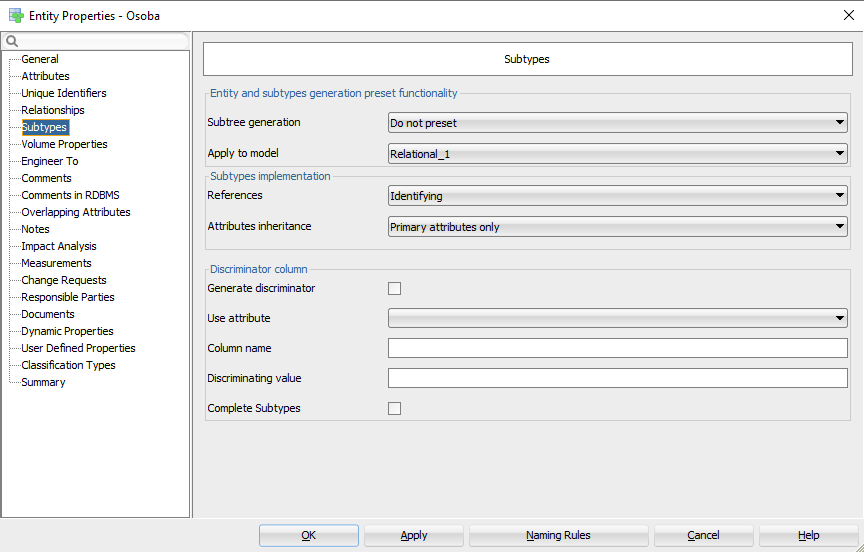
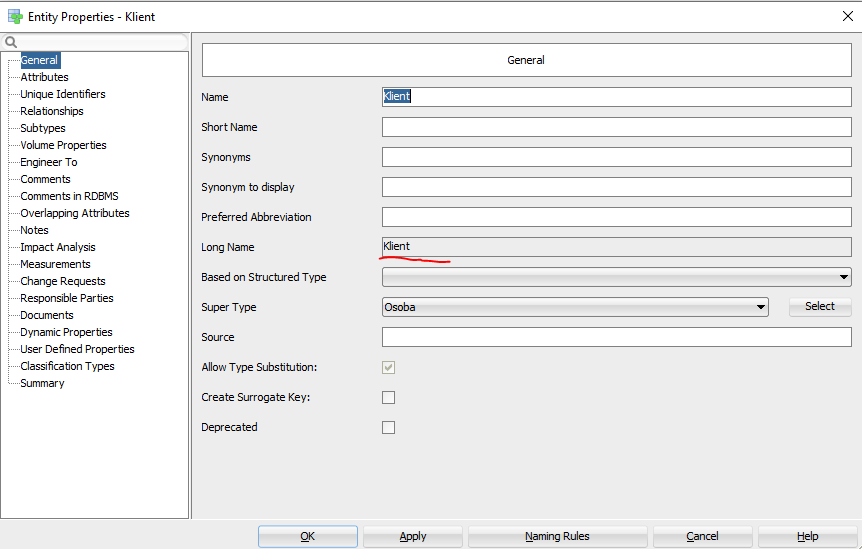
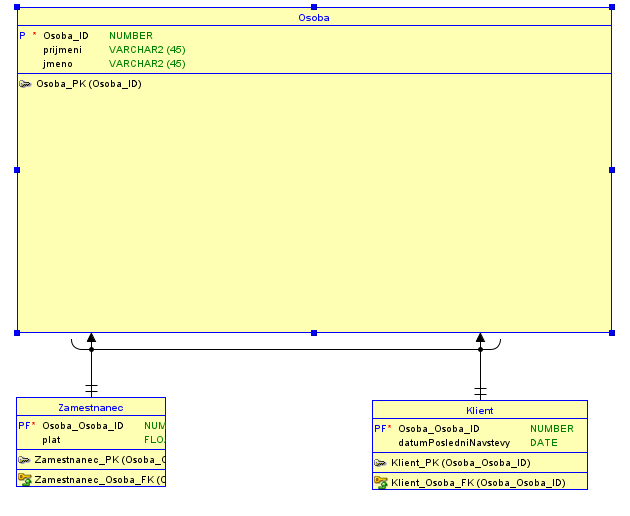
**Maturitní otázka**  
**B**áleš Antonín  
C4a  
17. 03. 2023

**Databázové systémy**

**21. Relační modely hierarchických struktur, Supertype a subtype, příklady využití**

**Základní pojmy**

* Existuje mnoho struktur ve vytváření tabulek v databázi
* Jedny z těchto struktur jsou například právě
  + Hierarchical
  + Supertype a Subtype
* **Hierarchická struktura:**
  + Z valné většiny řešena self-referencí
    - Tvořena z jedné entity, mající FK na tu stejnou entitu
    - FK nesmí být odkázán na tu samou instanci!
  + Tvoří vizuálně tzv. “strom”
  + Používá se např. U organizací
  + Pokud je třeba zaznamenat nadřízenost a několik podřízeností, používá se hierarchická struktura
  + Pravidla hierarchických struktur:
    - Vždy musí být jeden element úplně nahoře, tzv. **ROOT**
      * **ROOT** jako jediný nad sebou nemá žádný element
    - Element může mít pod sebou elementů nekonečné i žádné množství, ale nad sebou jen a pouze jeden!
  + 
  + 
  + 
* **Supertype a Subtype:**
  + Můžeme si ji představit jakožto “dědičnost”
  + Jestliže mají dva předměty několik stejných atributů/vztahů, supertype a subtype je na místě
  + Řeší se takto:
    - Vytvoří se SUPERTYPE entita
      * *Např. Produkt, který má atributy jméno a cena*
    - Vytvoří se SUBTYPE entity
      * Které odkazují na SUPERTYPE entitu, a právě tento samotný FK odkaz je i jejich PK
      * *Např. Elektro, které má vše co PRODUKT a ještě elektrickou spotřebu*
      * *Např. Potraviny, které má vše co PRODUKT a ještě datum spotřeby*
  + Pravidla Supertype a Subtype struktury:
    - Můžeme sami nastavit, zda chceme, aby SUPERTYPE instance mohl být instancí více SUBTYPŮ, či nikoliv
      * *Řešení, pokud chceme aby SUPERTYPE mohl mít jen a pouze jednu SUBTYPE instanci* ***(Exkluzivní)****:* 
        + *V SUPERTYPU využijeme tzv. DISCRIMINATOR sloupec, který udává, jaký ze subtypů má mít jeho instanci*
        + *Kontrola, zda se opravdu nachází instance SUPERTYPU pouze v jednom SUBTYPU, se dělá mnoha různými způsoby, záležíc na modelu*
      * *Řešení, pokud chceme aby SUPERTYPE mohl mít tolik SUBTYPE instancí kolik jen chce* ***(Non-exkluzivní)****:*
        + *Nic extra neděláme - takto funguje normální vztah SUPERTYPE/SUBTYPE*
    - SUBTYPE získá všechny atributy a vztahy, jako má SUPERTYPE
    - SUPERTYPE ze SUBTYPŮ nic nezískává
    - SUBTYPE klidně může mít i více SUPERTYPŮ
    - SUPERTYPE instance může existovat sama bez jakéhokoliv SUBTYPU
    - SUBTYPE může být zároveň i SUPERTYPEM, pokud má další svoje SUBTYPY
  + 
  + 
  + 
  + 
  + 
  + 
  + *Také lze poměrně dobře vytvořit příklad v data modeláři:*
  + *Vytvoříme si stejné tabulky v data modeláři*  
    
  + *Na supertype entitě budeme mít toto nastavení (Všimněme si, že máme možnost zaškrtnutí generování discriminátoru - ten nám zajistí odkaz na jednu z tabulek!)*  
    
  + *V subtype entitách musíme odkázat na supertype entitu*  
    
  + *Při vygenerování do relational diagramu získáme tzv. Arch (I přes to, že to JE supertype/subtype !)*  
    
  + *Poté můžeme vygenerovat v MENU nahoře SQL kód a popsat, co se v něm přesně děje*