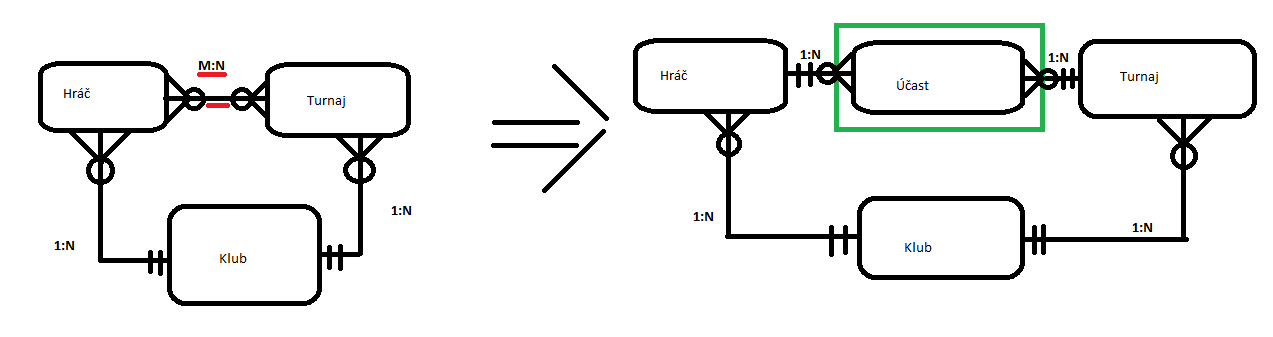
**2. Modelování relační databáze - konceptuální (logické) schéma, relační schéma**

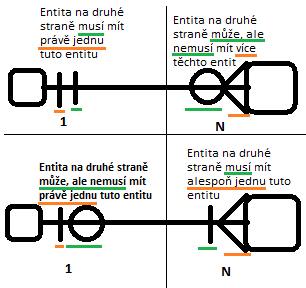
**Pojmy**

1. **Logické schéma**
   * Koncept databáze ze zadání
   * Obsahuje entity a vazby
   * Vazby jsou popsané schématickými značkami pro kardinalitu a parcialitu
   * Slouží pro konzultaci s klientem
   * V logickém schématu se také využívá tzv. **Dekompozice:**
     1. Převedení M:N vazeb na dvě vazby 1:N s novou prostřední, vloženou entitou.

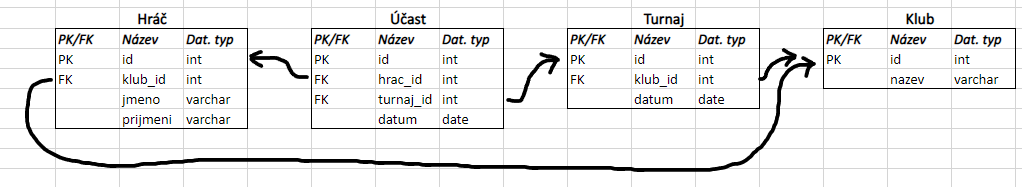
**Logické schéma a Dekompozice**



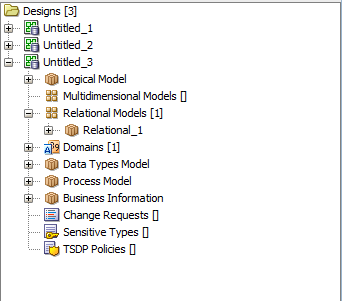
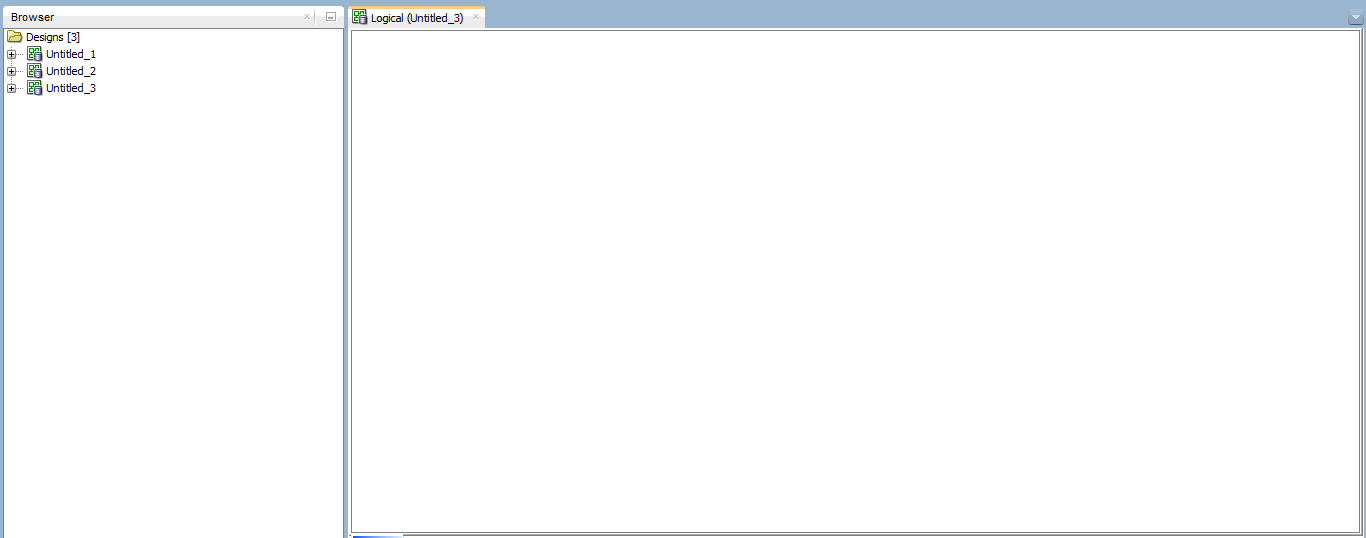
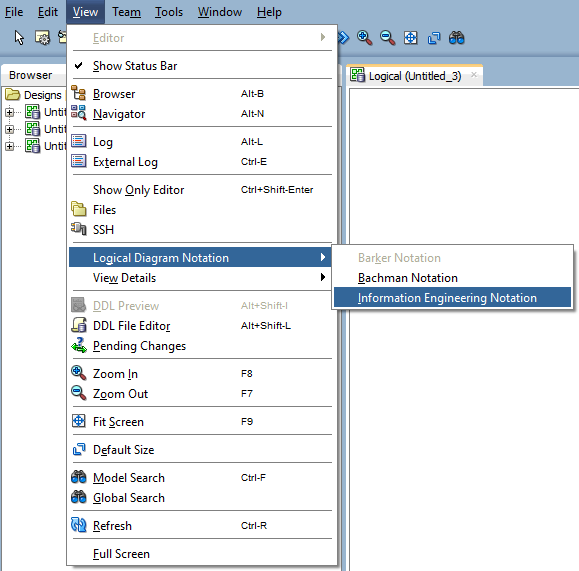
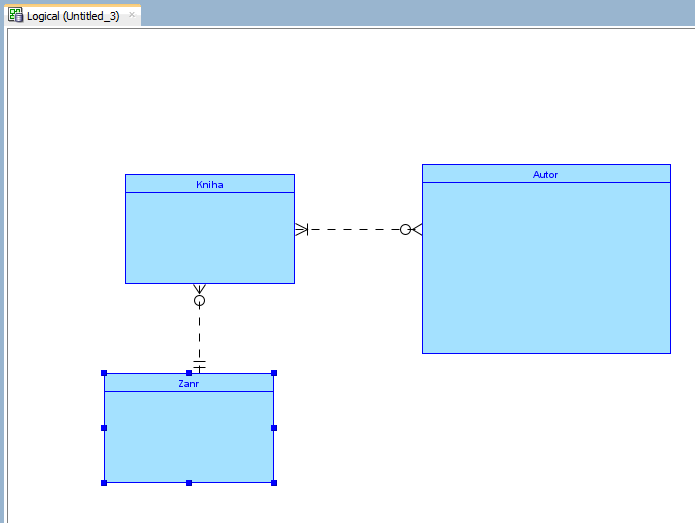
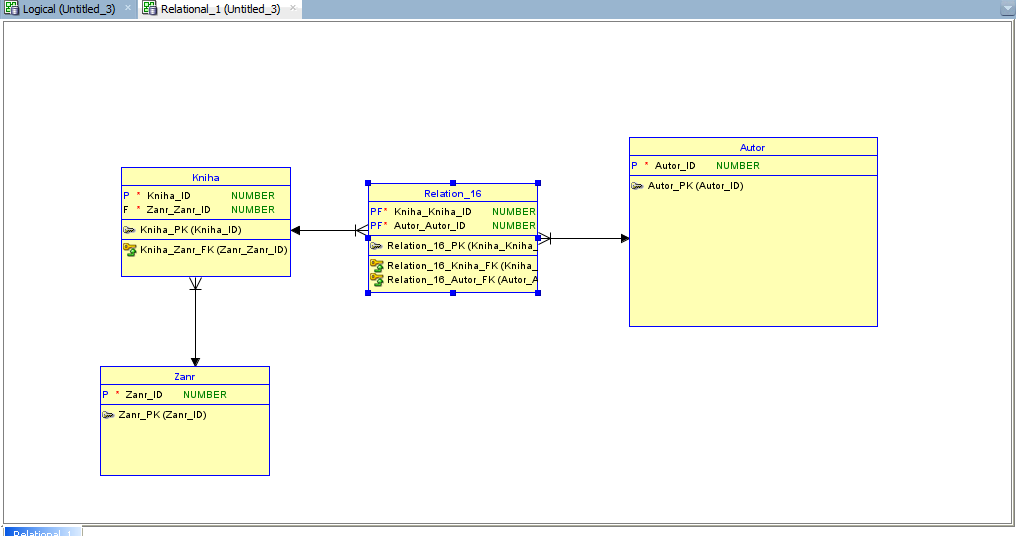
Vysvětlení kardinality a parciality ve schématických značkách



1. **Relační schéma**
   * Vazby jsou převedeny na PK *(Primary Key)* a FK *(Foreign key)* místo schématických značek
   * Obsahuje jak atributy, tak datové typy entit
     1. **Primární klíč *(Primary key****)*
        + Atribut
        + Měla by ho mít každá entita
        + Každá instance má unikátní PK
        + Slouží pro vyhledání a rozlišení dané instance
        + Nesmí být NULL a musí být právě jeden
        + Může být:
          1. **Umělý** – primární klíč nějak uměle vytvoříme
          2. **Přirozený** – atribut, který víme že je unikátní a každá instance ho bude mít, můžeme udělat primárním klíčem
        + A také:
          1. **Jednoduchý** – skládá se z jednoho atributu
          2. **Složený** – skládá se z více atributů
        + Všechny klíče, které by mohli být primární v tabulce se nazývají **Kandidátní**
        + Všechny klíče, které se nestali primárními klíči, ale mohli by jimi být se nazývají **Alternativní**
     2. **Cizí klíč *(Foreign key)***
        + Atribut, který v entitě odkazuje na PK instance stejné či jiné tabulky.
        + Hodnota i datový typ musí tedy odpovídat PK instance na kterou odkazujeme
        + Může být null a nemusí být unikátní

**Relační schéma**

**Modelování v Data Modeláři [PŘÍKLAD]**

1. Vytvoříme nový design kliknutím pravým tlačítkem
   * **“Designs” -> New Design**
2. Zobrazíme logické schéma kliknutím pravým tlačítkem na **“Logical Model” -> Show**
3. Nastavíme Engineering notaci kliknutím na **“View” -> Logical Diagram Notation -> Information Engineering Notation**
4. Vytvoříme entity a nastavíme vztahy z horní lišty 
5. Kliknutím na dvojí modrou šipku v horní liště Engineerneme logické schéma na relační
6. Samozřejmě můžeme nejdříve začit relačním schématem a poté ho “Reverse Engineernout” do Logického, to je ale zbytečně zdlouhavější a složitější na nastavení.
7. Kliknutí v horní liště na ikonku databáze “Generate DDL” získáme celý sql soubor s DDL příkazy k vytvoření tohoto schématu do reálné databáze.