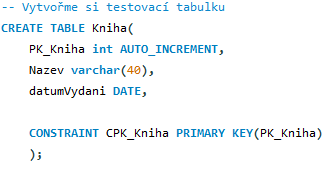
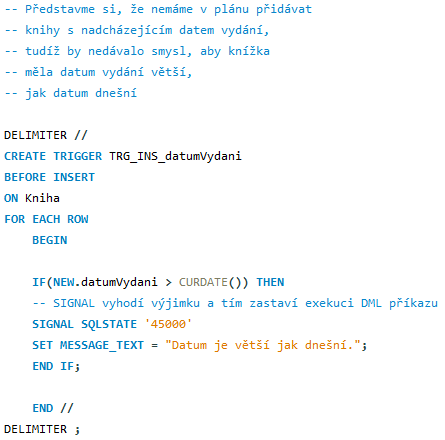
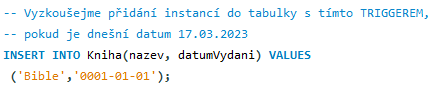
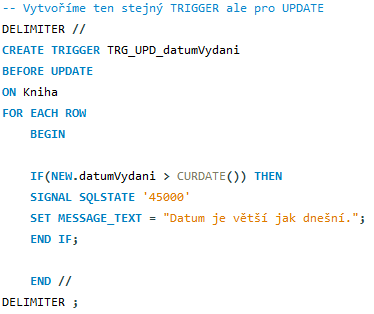
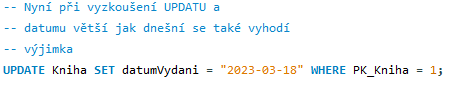
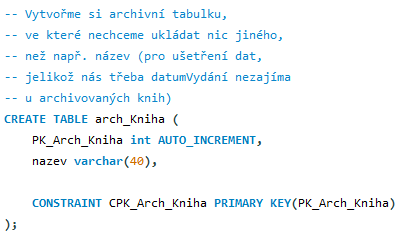
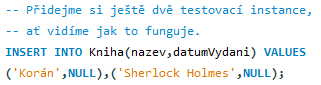
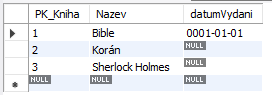
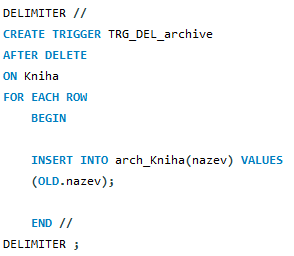
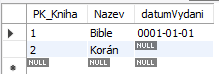
**18. Triggery**

**Úvod**

* Někdy si přejeme, aby se stala nějaká akce po specifických DML operacích
* Např.
  + Chceme zkontrolovat vstupní hodnoty před INSERTEM a UPDATEM, které nejdou zkontrolovat CHECKem *(Např. Kontrola s funkcemi (CURDATE(),DATEDIFF(),…))*
  + Nechceme data vyloženě smazat při DELETE, ale jen je posunout do jiné, archivní tabulky, či dokonce databáze
* Proto existují Triggery
  + Rutiny (bloky kódu), které se spouští buďto před či po dané DML operaci

**DML operace a spouštění TRIGGERŮ**

* **Můžeme přidat Trigger k operacím:**
  + INSERT
    - Trigger má přístup k tabulce **NEW**, která obsahuje všechny instance které budou INSERTOVÁNY
  + UPDATE
    - Trigger má přístup k tabulce **OLD**, která obsahuje staré instance před UPDATEM a tabulce **NEW**, která obsahuje instance které přepíšou instance staré
  + DELETE
    - Trigger má přístup k tabulce **OLD**, která obsahuje instance ke smazání
* **Trigger se může spouštět:**
  + BEFORE
    - Před samotnou operací *(Hodí se např. při kontrolování vstupních hodnot = pokud je něco špatně, DML operaci nespustíme vyhozením exceptionu pomocí příkazu SIGNAL)*
  + AFTER
    - Po samotné operaci *(Např. K archivaci dat)*
* 
* **BEFORE INSERT**
  + 
  +   
    
  +   
      
    
* **BEFORE UPDATE**
  + 
  +   
      
    
* **AFTER DELETE**
  + 
  +   
    
  + 
  +   
      
      
    *Sherlock Holmes nebyl doopravdy smazán, ale vpodstatě přesunut do tabulky, ve které se nemusí ukládat ”datumVydani” - tudíž jsme ušetřili místo, a rychlost prohledávání tabulky reálné.*