**Maturitní otázka**  
**B**áleš Antonín  
C4a  
14. 03. 2023

**Databázové systémy**

**7. Zálohování a archivace dat**

**Pojmy**

* Zálohování i archivace dat je klíčové pro správný chod databáze.
* *Proč zálohovat?*
  + Představte si, že nechtěně smažete důležité data? Zaútočí na vaší databázi útočník a smaže většinu dat, či dokonce tabulek? Proto zálohujeme, aby se při jakékoliv, byť nepatrné chybě, mohla databáze vrátit do původního funkčního stavu.
* *Proč archivovat?*
  + Představte si, že máte tabulku o několika milion záznamů, a polovina se nijak nepoužívá. Rychlost i kapacita databáze je zneužívána záznamy, které nejsou důležité. Přesně tyto nepoužívané data, které jsou ale třeba zaznamenat pro případné budoucí požadavky či legislativu, archivujeme někam jinam, mimo často používanou databázi.
* Archivace dat většinou nelze použít jako zálohu, jelikož neobsahuje žádné systémové soubory potřebné k obnovení systému, ale pouze data. *(Zálohování -> i se strukturou při poškození db, Archivace -> Jen a pouze data, a nic jiného než data!)*
* Archivační soubory se pak také dále kompresují, aby zabírali méně dat.

**Transakční LOG**

* Koncovka .log
* Kompletní příkazy + transakce provedených v DB
* Defaultní nastavení se liší podle DB
* Při chybě -> **ROLLBACK** (zpět do konzistentního stavu)
* Log se může mazat po každém **COMMITU**
* Většinou není potřeba celý LOG (Velikost velice rychle narůstá)
* Může být potřeba při testování
* Lze ho i zálohovat

**Datový soubor**

* Koncovka .mdf
* Slouží k uchování dat a metadat
* Struktury, účty, objekty, data
* V případě změny se provede změna v tomto souboru

**Strategie zálohování**

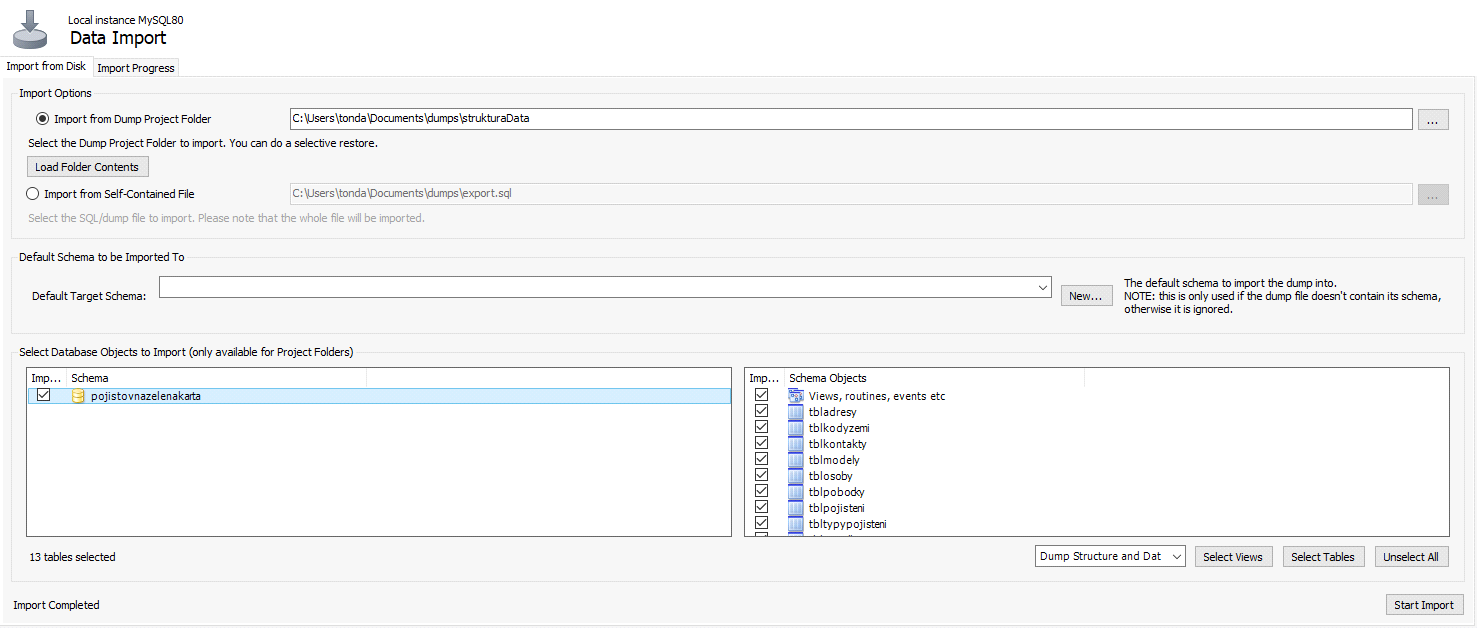
* Typ souboru záloh
* Frekvence provedení záloh
* Potřebný HW

**Zabezpečení**

* Testování záloh
  + Zda jsou čitelné, přístupné
* Přístupová práva
  + Kdo je zodpovědný za provádění, držení, zabezpečení
* HW alokace -> zabezpečený HW
  + Zálohy na jednom serveru
  + Ideálně na jednom místě (lokace)

**MySQL zálohování/archivace**

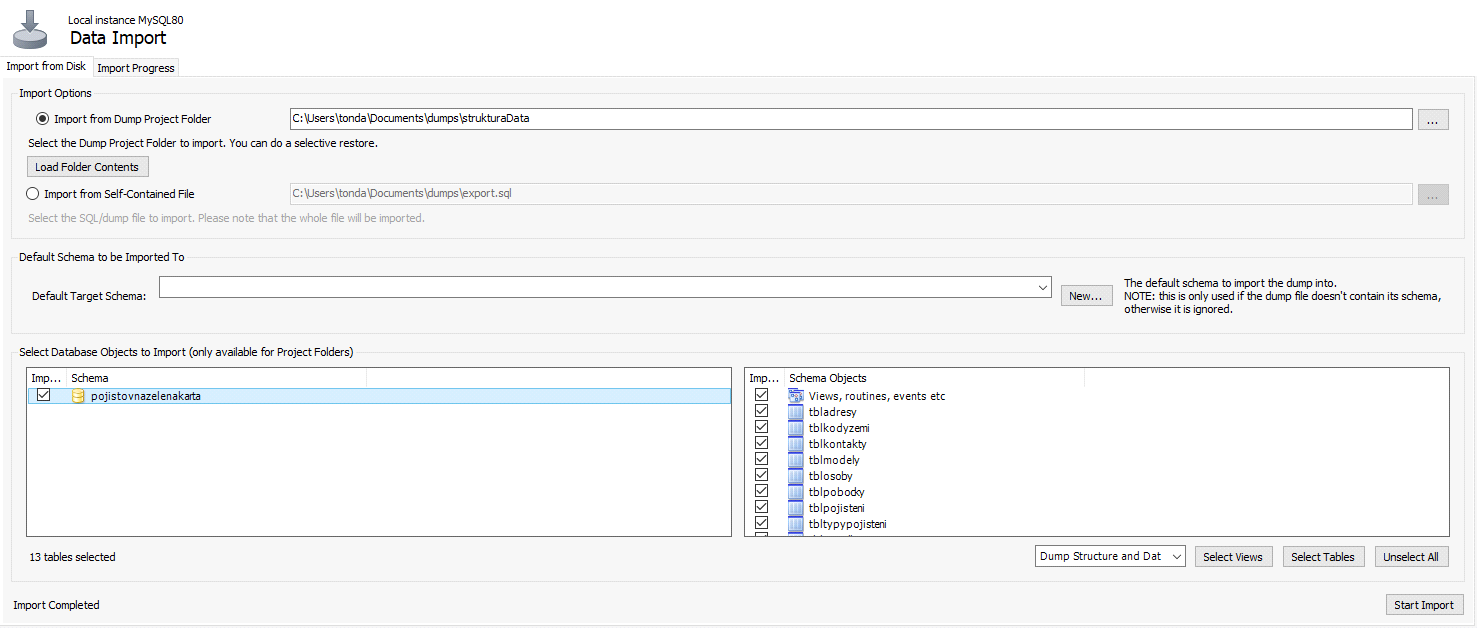
* Nejjednoduší zálohování je buďto přes Workbench, či přes příkazovou řádku.
* Výhodou u MySQL je poměrně velká škála možností daného exportu, každopádně velikou nevýhodou je nemožnost zálohování automatizovat.  
  (Tento problém se dá obejít externí automatizací příkazu pro zálohování databáze)
  + Export se nazývá tzv. “dump”
  + Existují dva typy uložení exportu:
    1. Projektová složka  
       (Každý databázový objekt se vyexportuje do svého vlastního .sql souboru do dané složky)
    2. Jediný soubor  
       (Celá databáze se vyexportuje do jediného .sql souboru)
  + Typy co SQL export může obsahovat:
    1. Pouze data
    2. Pouze strukturu
    3. Strukturu i data
  + Při exportu si můžeme vybrat zda do něho chceme zahrnout:
    1. Tabulky (a jaké)
    2. Pohledy (a jaké)
    3. Procedury a funkce
    4. Eventy
    5. Triggery
  + Ostatní možnosti v SQL exportu:
    1. Zahrnutí vytvoření databáze jako takové
    2. Vnoření celého exportu do jedné transakce (při případném importu)
  + Postup SQL exportu databáze v MySQL:
    1. V horní kolonce klikneme na **Server**
    2. Vybereme **Data Export**
    3. Vybereme požadovanou databázi / databáze a jak je chceme exportovat
    4. Klikneme na **Start export**
    5. *Hotovo!* Dump se bude nacházet ve vámi předtím vybrané složce

Příklad GUI při SQL importu databáze v MySQL

**MySQL načtení zálohy**

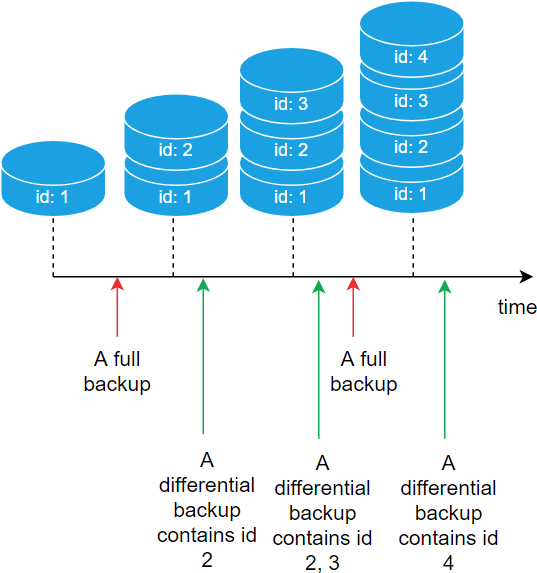
* Pro import musí existovat již nějaký dump, z kteréhu budeme importovat
* Import může obsahovat pouze to, co se v dumpu nachází (tudíž nemůžeme importovat data, když byl export pouze struktury apod.)
* Při importu si můžeme vybrat, zda do něho chceme zahrnout:
  + Tabulky (a jaké)
  + Pohledy + Eventy + Triggery apod.
* Typy naimportování:
  + Pouze data
  + Pouze strukturu
  + Strukturu i data
* Další možnosti:
  + Vybrání cíle databáze, kam se má dump naimportovat  
    (Jestliže existuje vytvoření databáze v dumpu, cíl se bude ignorovat)
  + Vybrání dumpové složky / souboru
* Postup exportu databáze v MySQL:
  + V horní kolonce klikneme na **Server**
  + Vybereme **Data Import**
  + Vybereme požadovanou složku / soubor
  + Vybereme požadovanou databázi / databáze a jejich objekty, co chceme naimportovat
  + Vybereme cílové schéma
  + Nastavíme dále import dle našeho uvážení
  + Klikneme na **Start import**
  + *Hotovo!* Databáze je zpátky naimportovaná ze zálohy

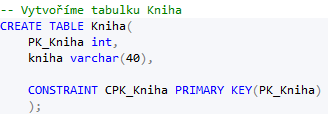
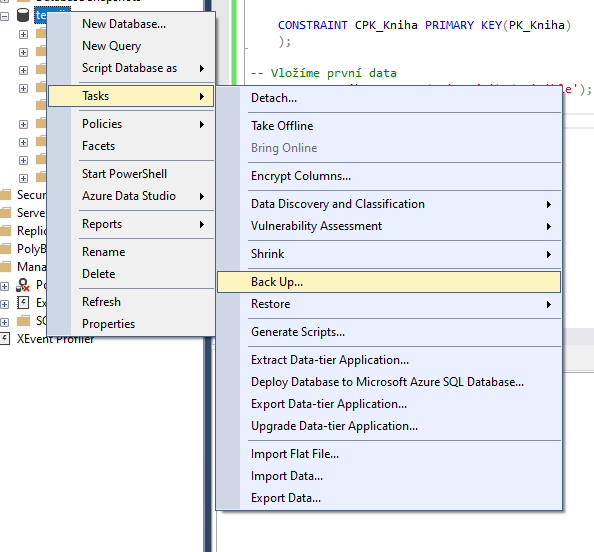
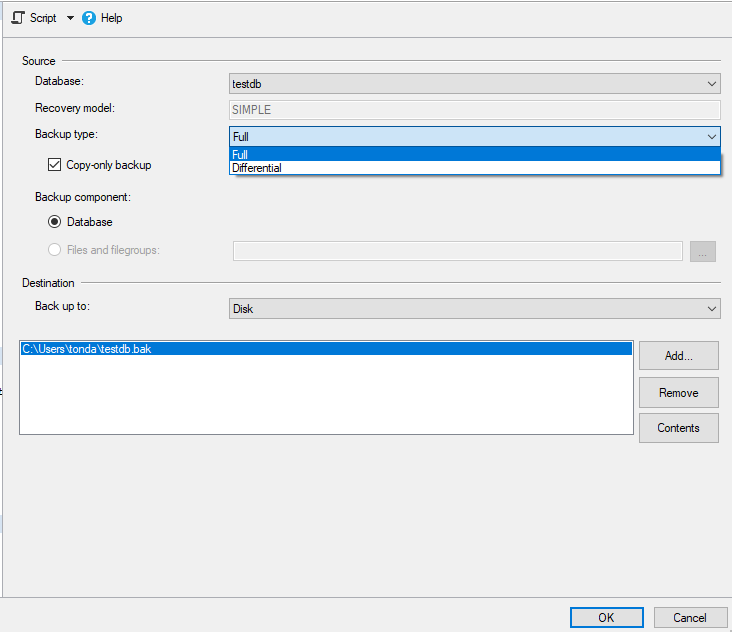
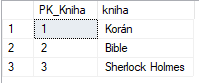
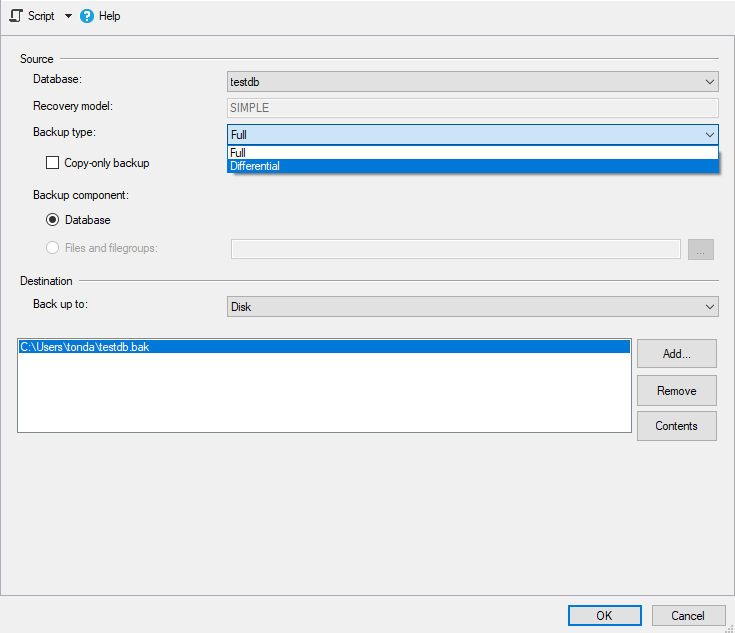
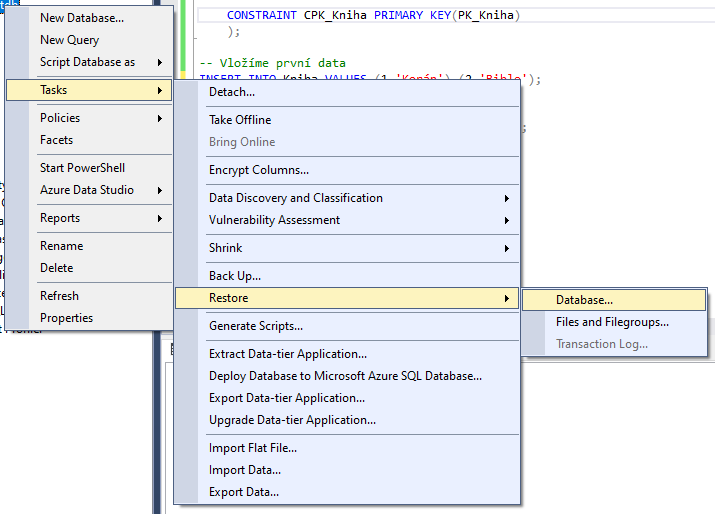
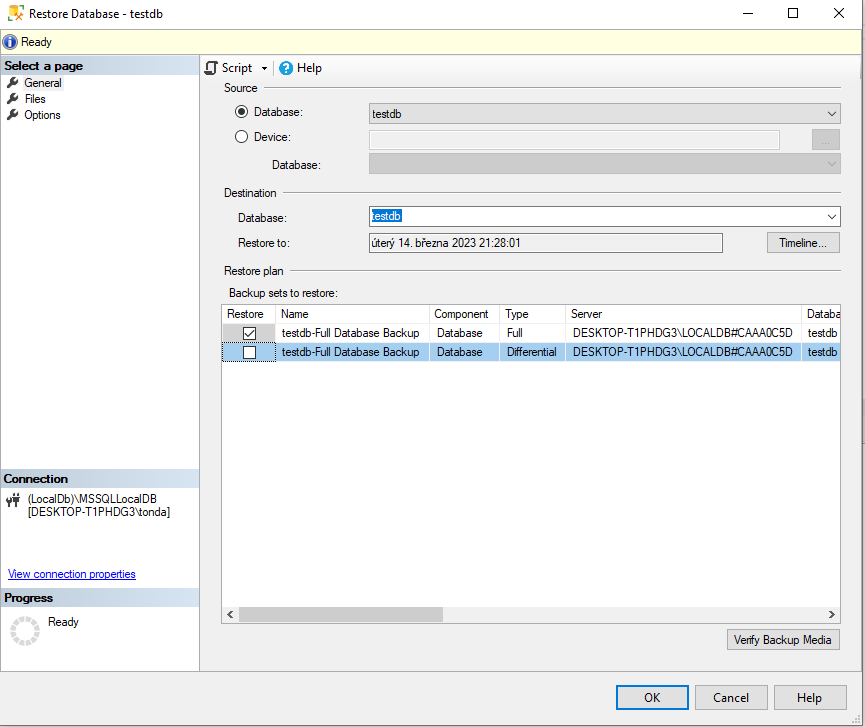
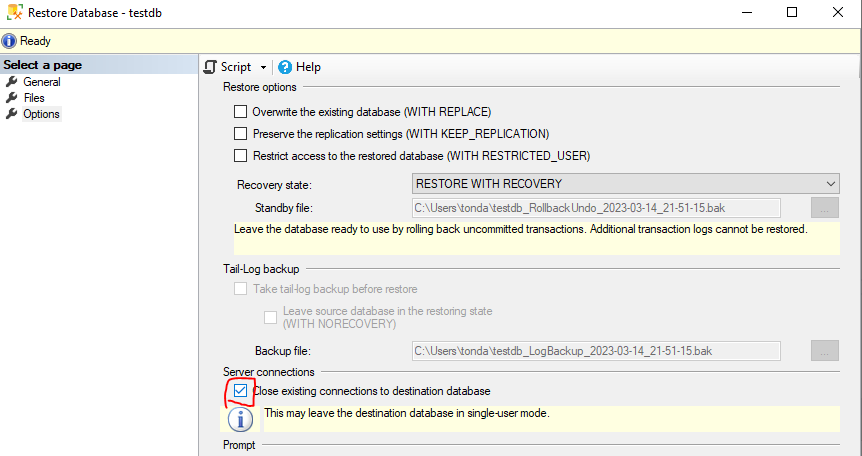
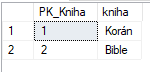
Příklad GUI při SQL importu databáze v MySQL



**MSSQL zálohování/archivace**

* Výhodou zálohování/archivace v MSSQL je možnost rozdělit jimi zvaný “BACKUP” na dvě části:
  + Full backup
    - Backup celé databáze
    - *Zabírá hodně místa*
  + Differential backup
    - Backup změn, které nastaly po Full backupu
    - Zabírá méně místa = neukládá celou databázi



* **Jak zálohovat v MSSQL a jak fungují Full/Differential backupy?**
  + 
  + 
  + 
  + Pravým tlačítkem klikneme na databázi
  + Uděláme FULL Backup
  + 
  + 
  + Provedeme DIFFERENTIAL backup  
    
  + 
  + **Nyní otestujeme, jaké data mají backupy:**
  + Klikneme pravým tlačítkem na databázi
  + 
  + Nejdříve vybereme pouze FULL
  + 
  + Možná bude třeba zaškrnout aby se veškeré připojení k DB ukončily - jinak recovery nebude možný  
    
  + Takto vypadá databáze po recovery FULL Backupu:  
    
  + 
  + Nyní vybereme jako recovery FULL i DIFFERENTIAL
  + Získáme zpět databázi FULL recovery i poté její rozdíl (rozdělenou červenou čarou)