Nástroje archivace pro otrlé

Workshop by NA

Pardubice 17. 5. 2023

Martin Rechtorik, NAČR

Národní archiv

Agenda

1. Databáze
2. Základy SQL, XML
3. Zpracování objektů
4. Rizika archivace databází
5. Archivace databáze

Databáze

Databáze

- Jsou všude kolem nás – bankovnictví, doprava, obchody,…
- Databáze = soustava propojených kartoték
- Základní typy: hierarchické, síťové, relační
- Dle způsobu uložení: centralizované, distribuované
- První databáze v USA v roce 1935 – sociální pojištění 26 milionům obyvatel, počítač UNIVAC I od firmy IBM

Druhy databází

1. **Objektové databáze** - kombinují databázové schopnosti se schopnostmi objektově orientovaného programovacího jazyka. Výsledky se ukládají jako objekty, lze je replikovat nebo upravovat za účelem vytvoření nových objektů
2. **Relační databáze –** jsou založeny na relačním modelu dat,. Informace jsou rozloženy do tabulek propojených pomocí klíčů (unikátnost, neexistence duplicity, atomicita, jednoduchost, klíče), jazyk SQL vynalezl Tedd Codd 1970

Druhy databází

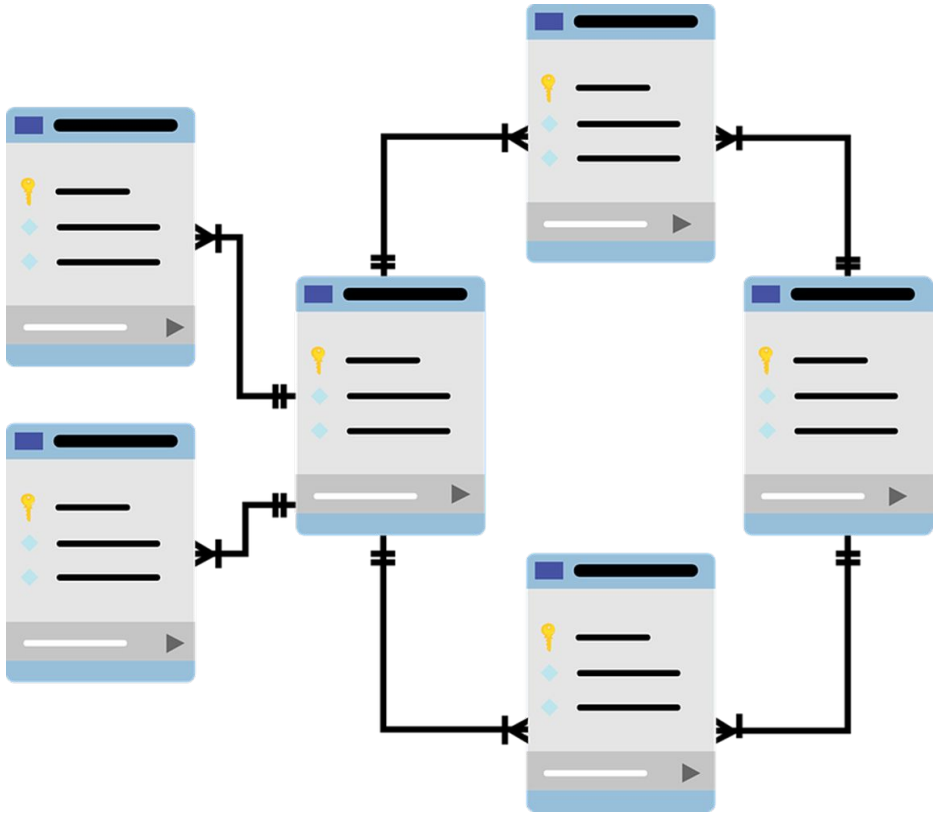
3. **NoSQL databáze** – nerelační databáze, protože informace o datech nejsou v tabulkách, přesnější jméno je „nejen pouze SQL“, existují od 60. let 20. stol., vhodné např. pro „Big Data“, lépe se horizontálně škálují
4. **NewSQL databáze** – kombinace SQL s NoSQL, systémy pro správu relačních databází, které se snaží poskytovat škálovatelnost systémů NoSQL pro pracovní zátěž online zpracování transakcí
5. **Sharepoints + Frameworks** – sdílená prostředí pro práci s dokumenty

Relační databáze

- organizuje data do jedné nebo více tabulek (nebo "relací") v podobě sloupců a řádků, přičemž každý řádek identifikuje jedinečný klíč
- Řádky se také nazývají záznamy nebo n-tice, sloupce se také nazývají atributy.
- platí, že každá tabulka/relace představuje jeden „typ entity“ (jako je zákazník nebo produkt)
- Řádky představují instance daného typu entity
- sloupce představují hodnoty přiřazené k této instanci (jako je adresa nebo cena)
- Např. každý řádek tabulky třídy odpovídá třídě a tabulka třída odpovídá více studentům, takže vztah mezi tabulkou tříd a tabulkou studentů je **„jedna k mnoha“**

Relační databáze

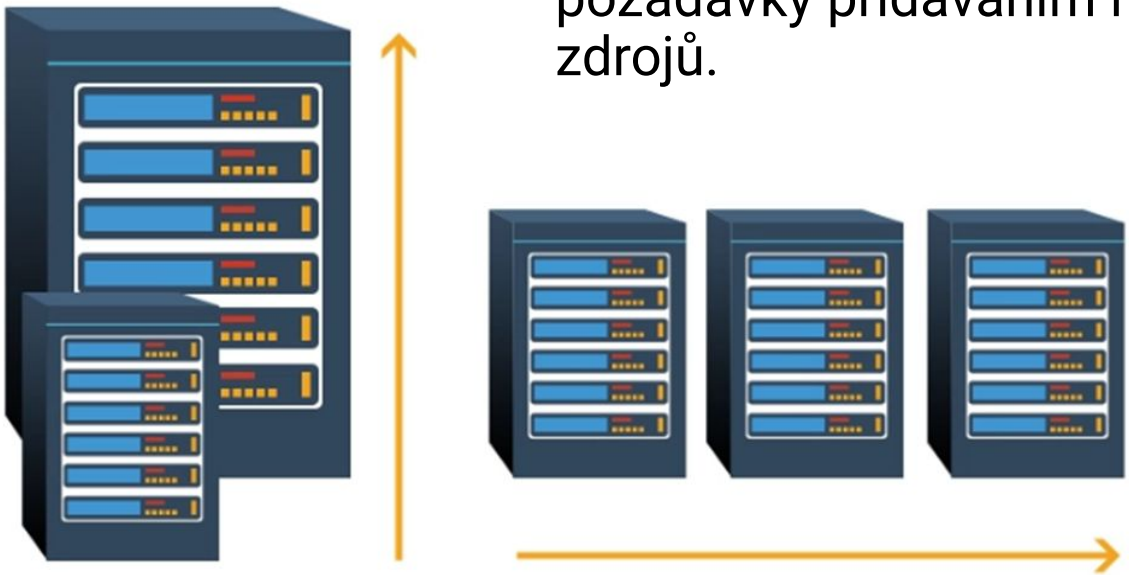
- Relace
- Klíče
- Datové tabulky
- Číselníky
- Funkce
- Rutiny
- Indexy



Relační databáze

- Jazyk SQL – structured query language, SQL je spojen s relačními databázemi
- “zvláštní typ programování jehož výsledkem je INFORMACE”
- Výhody a nevýhody:
  - Jednoduché uložení informací/dat
  - Odolnost vůči změnám
  - Nesnadné něco přidávat (když jsou tabulky plné)
  - Nevhodné pro horizontální škálování

Škálování databází



Schopnost databáze zvládat měnící se požadavky přidáváním nebo odebíráním zdrojů.

Základy SQL, XML

Základy SQL, XML

SQL

- dotazovací jazyk
- relační databáze
- 1970, Tedd Codd
- atomicita
- jeden k mnoha
- jednoduchost

XML

- popisovací jazyk
- konec 90.let
- rozvoj internetu
- výměna informací
- strom
- alternativou je JSON

S

SE

SE

SE

OF

GF

ulky)

```
        SELECT
CASE WHEN archive.archive_code::text ~~ '100000010'::text THEN 'NA'::text
    WHEN archive.archive_code::text ~~ '100000020'::text THEN 'ABS'::text
    WHEN archive.archive_code::text ~~ '211%'::text THEN 'SOA_Pha'::text
    WHEN archive.archive_code::text ~~ '221%'::text THEN 'SOA_Pha'::text
    WHEN archive.archive_code::text ~~ '212%'::text THEN 'SOA_Třeb'::text
    WHEN archive.archive_code::text ~~ '222%'::text THEN 'SOA_Třeb'::text
    WHEN archive.archive_code::text ~~ '213%'::text THEN 'SOA_Plz'::text
    WHEN archive.archive_code::text ~~ '223%'::text THEN 'SOA_Plz'::text
    WHEN archive.archive_code::text ~~ '214%'::text THEN 'SOA_Ltm'::text
    WHEN archive.archive_code::text ~~ '224%'::text THEN 'SOA_Ltm'::text
    WHEN archive.archive_code::text ~~ '215%'::text THEN 'SOA_HK'::text
    WHEN archive.archive_code::text ~~ '225%'::text THEN 'SOA_HK'::text
    WHEN archive.archive_code::text ~~ '216%'::text THEN 'MZA'::text
    WHEN archive.archive_code::text ~~ '226%'::text THEN 'MZA'::text
    WHEN archive.archive_code::text ~~ '217%'::text THEN 'ZAO'::text
    WHEN archive.archive_code::text ~~ '227%'::text THEN 'ZAO'::text
    WHEN archive.archive_code::text ~~ '3%'::text THEN 'Archivy_městské'::text
    WHEN archive.archive_code::text ~~ '4%'::text THEN 'Archivy_správní'::text
    WHEN archive.archive_code::text ~~ '5%'::text THEN 'Archivy_soukr'::text
    WHEN archive.archive_code::text ~~ '5%'::text THEN 'Archivy_spec'::text
    WHEN archive.archive_code::text ~~ '6%'::text THEN 'Univerzity, muzea'::text
    WHEN archive.archive_code::text ~~ '7%'::text THEN 'Bezp_složky'::text
    WHEN archive.archive_code::text ~~ '000000010'::text THEN 'ASMV'::text
    WHEN archive.archive_code::text ~~ '009999010'::text THEN 'Zahraniční inst'::text
    WHEN archive.archive_code::text ~~ '8%'::text THEN 'AUMA'::text
    WHEN archive.archive_code::text ~~ '9%'::text THEN 'Jiné, knihovny'::text
        ELSE NULL::text
        END AS "Archiv",
        rizeni.typ as "Druh řízení",
    COUNT(DISTINCT CASE
      WHEN rizeni.stav = 'UKONCENO'
      then rizeni.rizeni_id end) as "Počet ukočených řízení",
     COUNT(DISTINCT CASE
      WHEN rizeni.stav = 'ZRUSENO'
      then rizeni.rizeni_id end) as "Počet zrušených řízení",
    COUNT(DISTINCT CASE
      WHEN rizeni.stav != 'ZRUSENO' and rizeni.stav != 'UKONCENO'
      THEN  rizeni.rizeni_id end) as "Počet běžících řízení"
    FROM portal_produkce_statistiky.rizeni
    JOIN portal_produkce_statistiky.archive on archive.archive_code = rizeni.archiv_kod
    and rizeni.typ = 'SIP'
```

Zpracování objektů

Druhy dat

Strukturovaná:

Mají standardizovaný formát pro efektivní přístup softwaru i lidí.

Obvykle se jedná o tabulky s řádky a sloupci, které jasně definují atributy dat.

Počítače mohou strukturovaná data díky jejich kvantitativní povaze efektivně zpracovávat a získávat tak informace.

Nestrukturovaná:

Informace, které buď nemají předem definovaný datový model, nebo nejsou předem definovaným způsobem uspořádány.

Obvykle jsou textové, ale mohou obsahovat i data, jako jsou data, čísla a fakta. To má za následek nepravidelnosti a nejednoznačnosti, které ztěžují jejich pochopení pomocí tradičních programů.

XML + / -

XML

+ není ploché jako CSV

+ reflektuje model jeden k mnoha

- velikost(upovídanost)

XML, JSON, HTML

JSON

```
{"archivari":[

  { "krestniJmeno":"Josef",
"prijmeni":"Boček" },

  { "prijmeni":"Sáša",
"lastName":"Dušková" }

]}
```

XML

```
<archivari>

<archivar>
<krestniJmeno>Josef</krestniJmeno>
<prijmeni_>Boček</prijmeni_>
</archivar>

<archivar>
<firstName>Sáša</firstName>
<prijmeni>Dušková</prijmeni>
</archivar>

 </archivari>
```

Práce s XML 1

Notepad ++

- jafa.xml

Práce s XML 2

Pomocí jazyka Python:

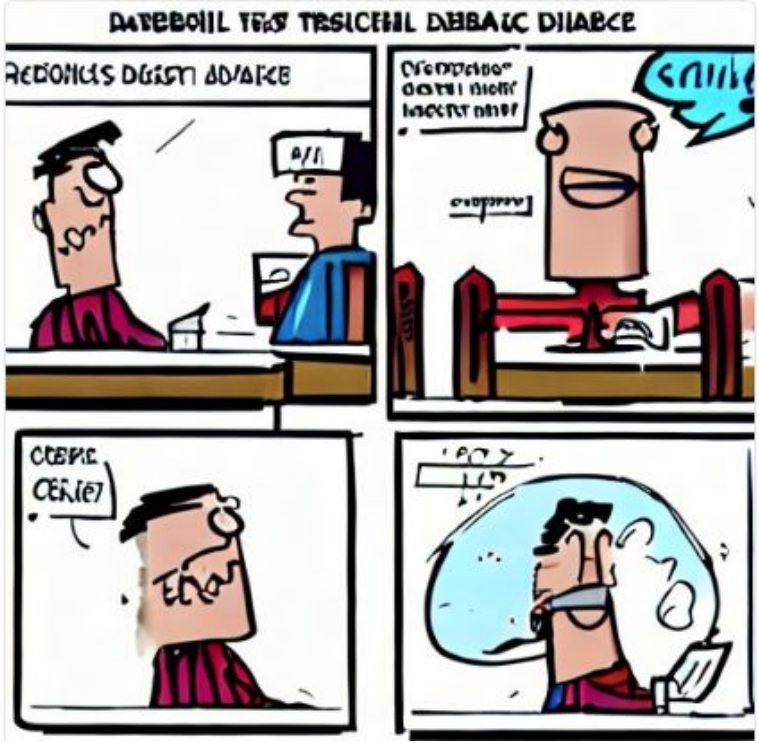
1. Převod souboru XML do CSV
2. Převod souborů XML do CSV

Rizika archivace databází

Rizika archivace DB

Technické

- velikost databáze
- množství objektů
- podpora, zastaralost
- finance, vendor lock
- chybná komunikace
- nereálné cíle



Rizika archivace DB

Právní

- citlivé údaje
- smluvní vztahy
- chráněné spec. EU právem (autorské)
- nezájem původců

Archivace databáze

Volba strategie

- průzkum (co bylo účelem systému, k čemu byl využíván, typ spravovaných informací)
- dokumentace (zajištění manuálů, návodů, příruček, datové modely, screenshoty z originální aplikace)
- metadata (získání logiky v podobě SQL dotazů, skripty atp.

Archivace databáze

Co má informační hodnotu dle §4 a §5 AZ?

- celá databáze
- strukturovaná data (XML, JSON)
- Balíčky SIP dle NSESSS
- Jiné vhodné řešení

SIARD

Vhodný pro akvizici databáze jako celku

- vždycky je levnější a jednodušší vzít vše než dělat výběr
- lze prohlížet obsah, ale …
- má vlastní metadata, ale …
- důvěryhodnost dat

Archivace databáze

Archivář trvající na eSSL je z pohledu světa IT něco jako bezdomovec hrabající se v odpadkových kontejnerech.

Nástroje archivace pro otrlé

Díky za pozornost!

Martin Rechtorik, NAČR

Národní archiv