MarkLogic

NDBI040 - Moderní databázové systémy

Jáchym Bártík

19. 5. 2022

Úvod

- NoSQL multi-modelová databáze zaměřená na dokumenty.
- Podporované typy jsou JSON, XML, text, RDF a binární data.
- Přímo v databázi lze spustit skripty v jazycích JavaScript a XQuery.
- Poskytuje tři API - Node.js, Java, REST.

Enterprise

- Není opensource, ale nabízí zdarma licence pro vývojáře.
- Důraz na bezpečnost, spolehlivost a dostupnost.
  - Pokročilé možnosti nastavení přístupových oprávnění či šifrování
  - Používají ji především velké společnosti pracující s kritickými, citlivými a zároveň opravdu rozsáhlými daty - banky, technologické firmy a státní agentury.
- Je distribuovaná a velmi dobře škálovatelná pomocí shardingu.
- Na rozdíl od běžných NoSQL zaručuje ACID vlastnosti transakcí.
- Podle https://db-engines.com/en/ranking je v současnosti kolem 40. - 50. místa.

Instalace

- Podpora pro Windows, Linux (Red Hat, CentOS a Amazon) a macOS.
  - Ale jde to i na Ubuntu 20 na WSL pomocí programu `alien`.
- Aplikace defaultně poslouchá na třech portech:
  - `8000` - konzole pro spouštění příkazů a prohlížení výsledků,
  - `8001` - nastavení, vytváření databází, práva, ...,
  - `8002` - monitorování požadavků a paměťové a výpočetní zátěže.
- Na dalších portech je možné spřístupnit REST API či přímo klasické webové stránky.
- Pro import či export spousty dokumentů najednou slouží samostatný program MarkLogic Content Pump.

Insert

- Vytvoření dokumentu a přiřazení do kolekce `orders`:

```
declareUpdate();
xdmp.documentInsert(
    '/order_57590.json',
    {
        label: 'Clean Code: 2st Edition',
        type: 'book',
        created: '2022-04-16',
        state: 'paid',
        price: { value: 45, currency: 'USD' }
    },
    {
        collections: [ 'orders' ]
    }
)
```

Select

- Zobrazení názvu, stavu a URI všech objednávek:

```
const results = [];

for (const result of fn.collection('orders'))
    results.push({
        label: result.root.label,
        state: result.root.state,
        uri: fn.documentUri(result)
    });

results;
```

Where

- Potřebujeme najít `finished` objednávky vytvořené před 25. 3. 2022.

```
cts.search(cts.andQuery([
    cts.collectionQuery('orders'),
    cts.jsonPropertyWordQuery('state', 'finished'),
    cts.jsonPropertyRangeQuery(
        'created',
        '<=',
        xs.date('2022-03-25')
    )
]));
```

- Pro range dotazy je ale nutné nejdříve vytvořit příslušné indexy.

Update

- Vyhledané dokumenty (`node`) jsou immutable.
  - Je ale možné převést je na JS objekty a ty upravit a uložit.

```
const ordersToSend = cts.search( ... );

declareUpdate();

for (const node of ordersToSend) {
    const object = node.toObject();
    object.state = 'sent';
    object.sent = fn.currentDate();
    xdmp.nodeReplace(node, object);
}
```

Delete

- Smazání podle URI:

```
declareUpdate();

xdmp.documentDelete('/order_57590.json');
```

- Smazání vyhledané node:

```
declareUpdate();

const node = ...;
xdmp.nodeDelete(node);
```

Co dál?

- JavaScript je jen jednou z několika možností.
- XQuery umí to samé, ale je mnohem kompaktnější.
- SPARQL zase slouží pro RDF data.
- Dokonce lze použít i SQL.
  - Pouze ale pro vyhledávání (`SELECT`, `EXPLAIN`) v předem definovaných pohledech.
- Mimo to existují velmi podrobná nastavení pro všechno od vyhledávání přes ukládání dat po uživatelská oprávnění.

Závěr

- Není opensource.
- Je nutné opustit známé SQL koncepty a naučit se něco nového.
  - JavaScript či XQuery jsou ale dobré alternativy.
- Velmi dobrá škálovatelnost, dostupnost a bezpečnost.
  - Lze použít na prakticky libovolně velkou aplikaci.
- **Vše funguje**, snadná instalace a snadné použití.