
Napredno softversko inžinjerstvo

JOOQ & SPRING

MILICA VIDOJKOVIĆ
BROJ INDEKSA: 1849

Problemi u radu sa bazom u Java aplikaciji

- Ručno pisanje SQL stringova
- Runtime greške i teško održavanje
- Ograničenja ORM rešenja kod složenih upita

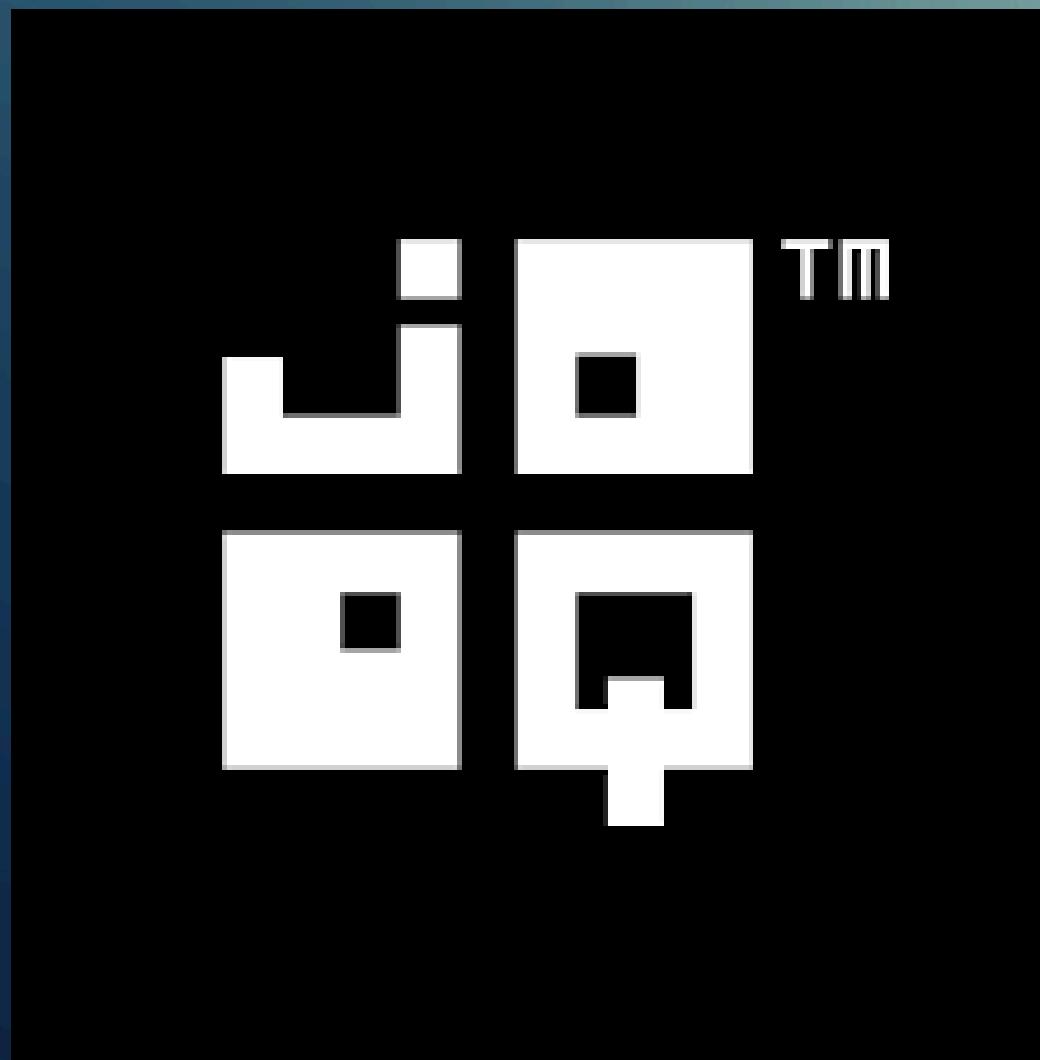
```
SELECT author.first_name, author.last_name  
FROM author  
WHERE author.last_name = 'Shakespeare'  
ORDER BY author.first_name, author.last_name  
LIMIT 10
```

```
ctx.select(AUTHOR.FIRST_NAME, AUTHOR.LAST_NAME)  
    .from(AUTHOR)  
    .where(AUTHOR.LAST_NAME.eq("Shakespeare"))  
    .orderBy(AUTHOR.FIRST_NAME, AUTHOR.LAST_NAME)  
    .limit(10)  
    .fetch();
```

Spring Boot – osnova aplikacije

- Brz razvoj Java aplikacija
- Auto-configuration
- Integracija sa bazama i bibliotekama
- REST API podrška





jOOQ - type-safe SQL

- SQL kroz Java DSL
- Compile-time provera
- Blizak “pravom” SQL-u
- Bez ručnih SQL stringova

Generisanje jOOQ klase

```
generated-sources/jooq
└── com/example/demo/jooq
    ├── DefaultCatalog.java
    ├── Library.java
    ├── Tables.java
    ├── Indexes.java
    ├── Sequences.java
    ├── Keys.java
    └── tables
        ├── Author.java
        ├── Book.java
        └── Loan.java
    └── tables/records
        ├── AuthorRecord.java
        ├── BookRecord.java
        └── LoanRecord.java
    └── tables/pojos (opciono)
```

- Analiza šeme baze
- Automatsko generisanje Java klasa
- Tabele, kolone, ključevi, indeksi

DSLContext klasa

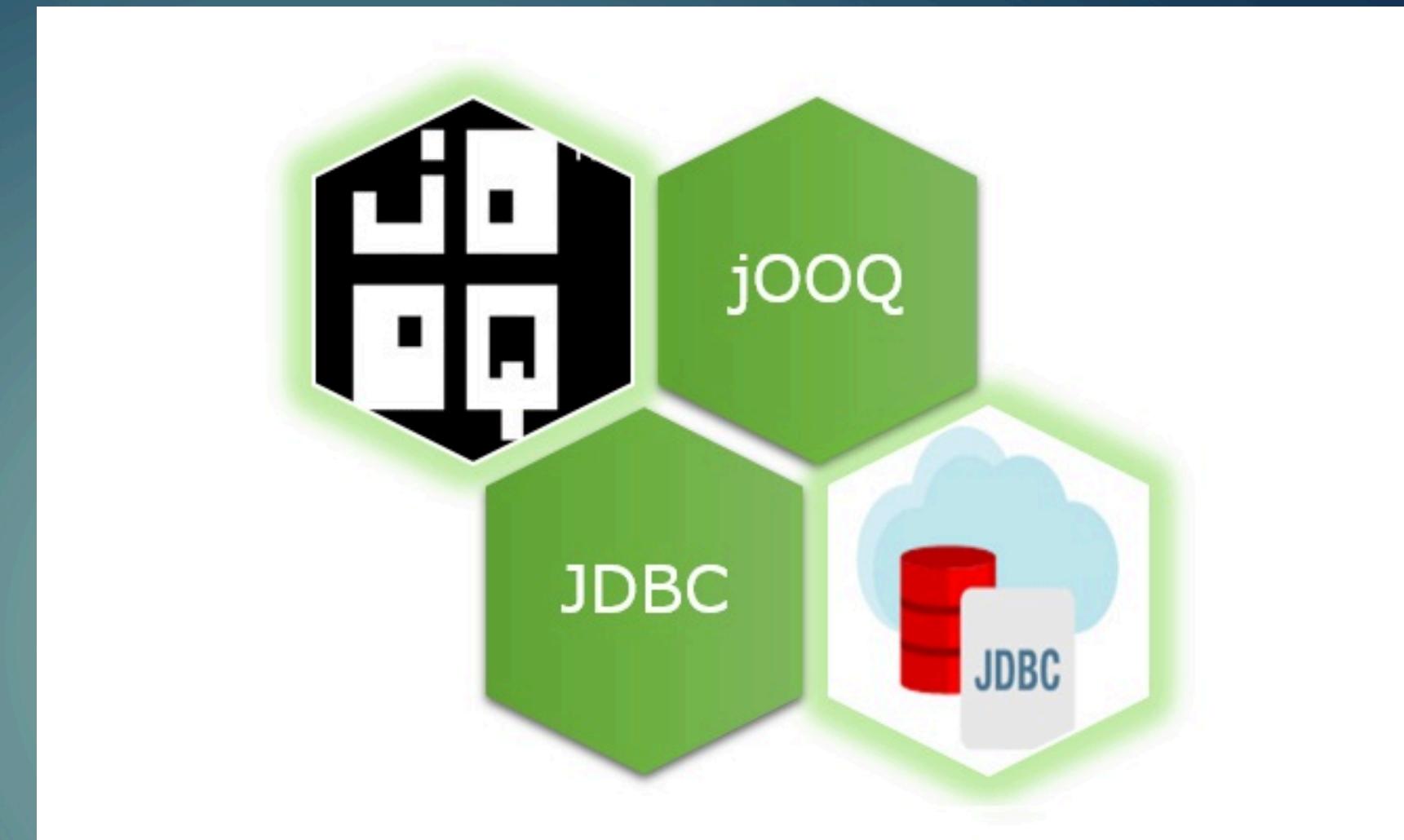
- Centralna ulazna tačka
- Kreiranje i izvršavanje upita
- JDBC + SQL dijalekat
- Integracija sa Spring transakcijama

```
// Create it from a pre-existing configuration
DSLContext create = DSL.using(configuration);

// Create it from ad-hoc arguments
DSLContext create = DSL.using(connection, dialect);
```

Flow: Controller → Database

- Controller
- Service
- jOOQ DSLContext
- JDBC
- PostgreSQL baza



jOOQ upiti

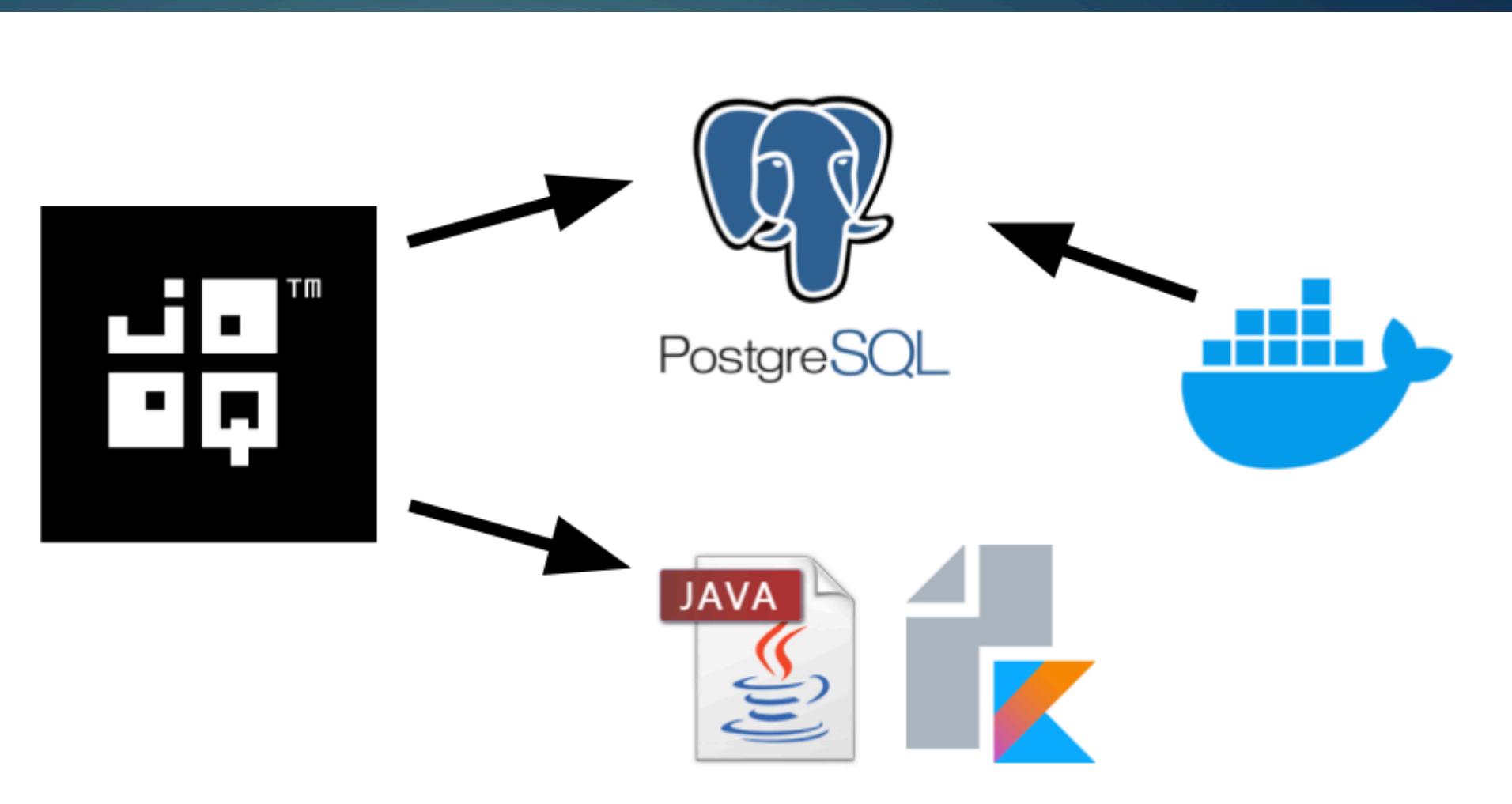
jOOQ omogućava gotovo kompletan SQL standard kroz Java API, uključujući:

- select, insert, update, delete
- join (inner, left, right)
- podupite (subqueries)
- agregacione funkcije (count, sum, avg)
- sortiranje i paginaciju
- transakcije
- returning klauzule (podržane u PostgreSQL)

Demo projekat

- Spring Boot REST aplikacija
- PostgreSQL + Docker
- jOOQ code generation
- SELECT, INSERT, JOIN primeri

Demo projekat



Zaključak

- Kontrola nad SQL-om
- Veća pouzdanost
- Lakše održavanje
- Pogodno za kompleksne upite

HVALA NA PAŽNJI