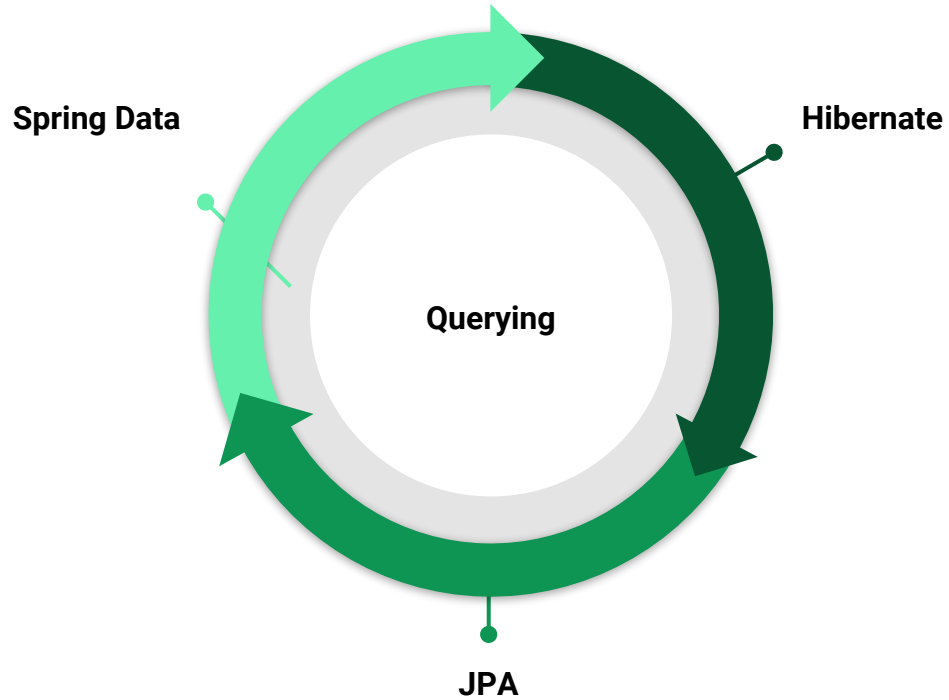


Querying

Persistence Layer



Overview

- HQL
- JPQL
- Spring Data
- Objectives
 - Return entities
 - Discuss different query capabilities
- HQL > JPQL

API

- Hibernate - `org.hibernate.Query`
- `javax.persistence.Query`
- `org.springframework.data.jpa.repository.Query`

Expressions and Operators

- from Player p where p.age < 28
- from Player p where (p.age between 28 and 30) and firstName like '%ik%'
order by p.firstName
- [Hibernate Expressions List](#)
- [Java Persistence Expressions](#)

Parameters

- Positional

- hibernate zero based

```
from Player p where (p.age between ? and ?) and firstName like '%ik%' order by p.firstName
```

- JPA 1 based

```
from Player p where (p.age between ?1 and ?2) and firstName like '%ik%' order by p.firstName
```

- Named

```
from Player p where (p.age between :ageMin and :ageMax) and firstName like '%ik%' order by p.firstName
```

- Spring Data

```
@Query("select p from Player p where (p.age between :minAge and :maxAge) and p.firstName like '%ik%' order by p.firstName")
```

```
List<Player> findPlayersParameters(@Param("minAge") Integer minAge, @Param("maxAge") Integer maxAge);
```

Joins

- implicit join

`select distinct p.team from Player p group by p.team.name having count(*) > 2`

- explicit join - preferred

`select distinct p.team from Player p join p.team group by p.team.name having count(*) > 2`

Functions

- Both HQL and JPQL allow for using functions

```
select distinct(p.team) from Player p join p.team where lower(p.team.name) = 'Crvena Zvezda'
```

- [Hibernate Functions](#)
- [JPA Functions](#)

Named Queries

```
@Entity
@Table(name = "s_team")
@NamedQueries(
    @NamedQuery(name = "Team.withMoreThanOnePlayer", query = "select distinct p.team from Player p join p.team group
by p.team.name having count(*) > 2")
)
```

Native Queries

- Native Queries

```
NativeQuery query = getSession().createNativeQuery("select * from s_player")  
.addEntity(Player.class);
```

- Stored Procedures

```
StoredProcedureQuery query = entityManager  
.createStoredProcedureQuery("count_players")  
.registerStoredProcedureParameter(  
    "playerId", Long.class, ParameterMode.IN)  
.registerStoredProcedureParameter(  
    "playerCount", Long.class, ParameterMode.OUT)  
.setParameter("playerId", playerId);
```

```
query.execute();
```

```
Long playerCount = (Long) query  
.getOutputParameterValue("playerCount");
```