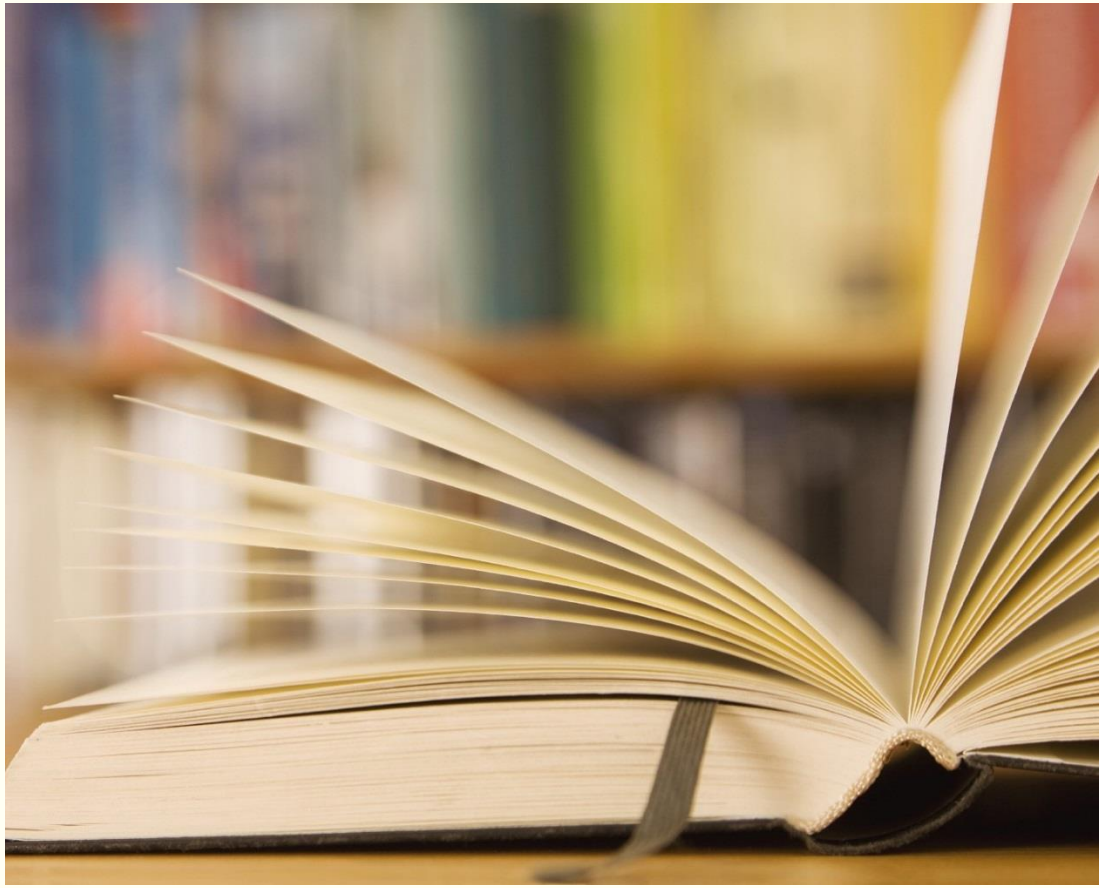
ENTERPRISE ARCHITECTURE

Najeeb Najeeb, PhD



NAMED QUERIES

Named Queries

- What is good about this?
- Software engineering ++
- What is still not happening?

```
@NamedQuery(name="Student.findStudents",
query="SELECT s FROM Student s WHERE
s.gpa = 3.0")

@NamedQuery(name="Student.findStudents
  ByGpa", query="SELECT s FROM Student s
  WHERE s.gpa = :gpa")

@NamedQueries({@NamedQuery(...),
  @NamedQuery(...),...})

TypedQuery<Student> query =
  em.createNamedQuery("Student.findStuden
  ts", Student.class);
```

NATIVE QUERIES

Native Dynamic Queries

- Why use native queries?
- What are the benefits?
- Compare performance
  - To Dynamic Queries
  - To Named Queries
- What is still not happening?
- How can we improve performance?

```
String queryStr = "SELECT s.* FROM
StudentTable AS s WHERE s.Grade = 3.0";

TypedQuery<Student> query
= em.createNativeQuery(queryStr, Student.cla
ss);

List<Student>  result = query.getResultList();

Stream<Student>
   students = result.getResultSttream();
```

Named Native Queries

- What is still not happening?

```
@NamedNativeQuery(name="findStudents",
query="SELECT s FROM StudentTable AS s
WHERE s.Grade = 3.0",
resultClass=Student.class)

@NamedNativeQuery(name="findStudentsBy
  Gpa", query="SELECT s FROM StudentTable
  AS s WHERE s.Grade = :gpa",
  resultClass=Student.class)

@NamedNativeQuery({@NamedQuery(...),
  @NamedQuery(...),...})

TypedQuery<Student> query =
  em.createNamedQuery("Student.findStuden
  ts", Student.class);
```

Main Point

- JPQL is slow compared to SQL, since they must be converted to SQL and sent to the DB. Named Queries improve JPQL by performing the conversion once at application startup. As a side benefit this also results in early syntax checking.
- Native Queries are a good way to migrate an application from using SQL to JPQL. While native queries still use SQL, they return obejcts and this reduces much of the mapping work that is done in legacy systems.