

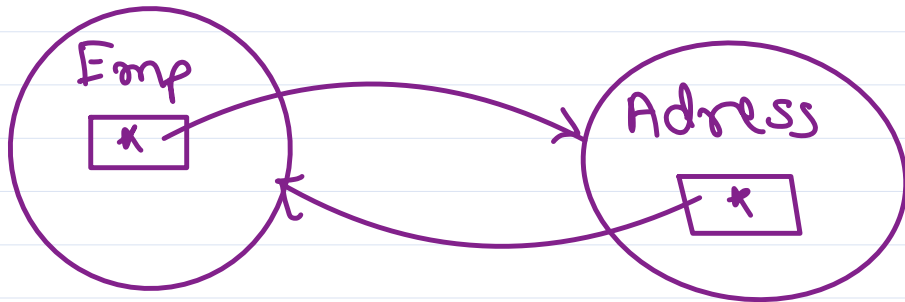
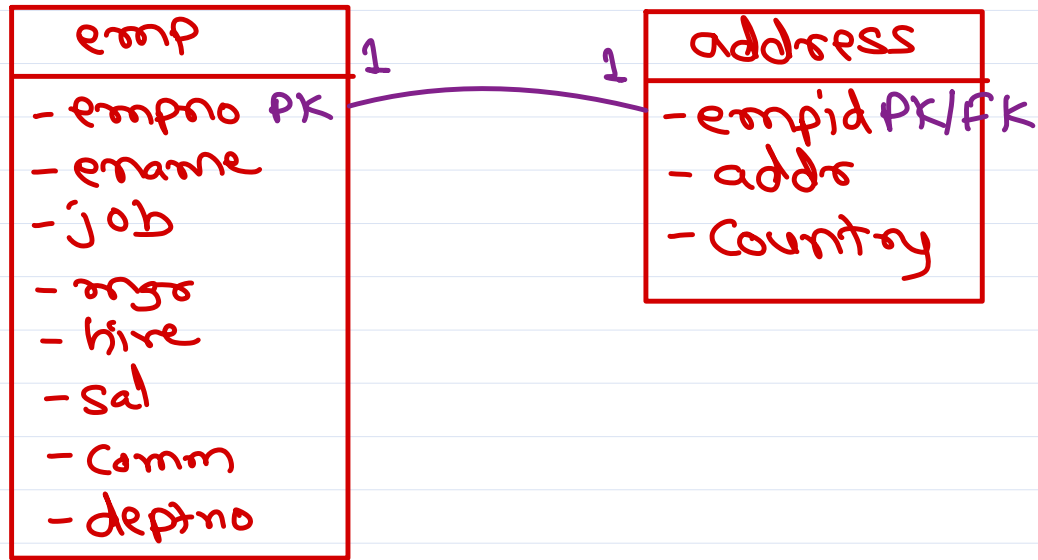


Advanced Java

Trainer: Nilesh Ghule



OneToOne

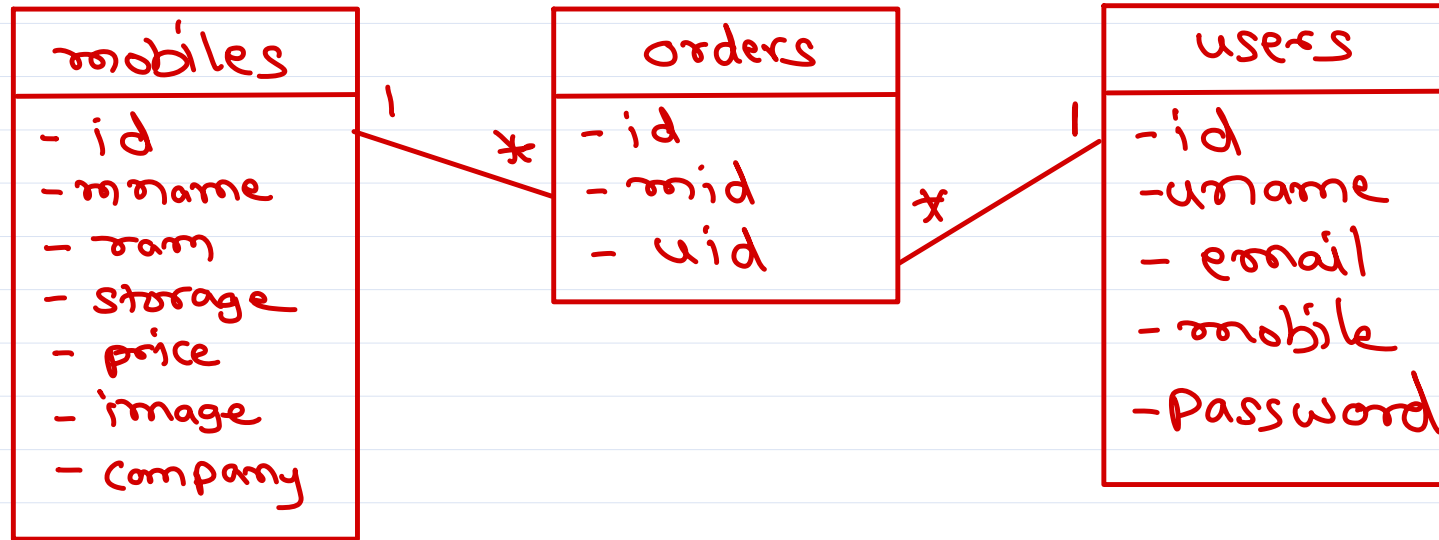


```
class Address {  
    @Id  
    int empid;  
    String addr;  
    String country;  
    @OneToOne  
    @PrimaryKeyJoinColumn  
    Emp emp;  
}
```

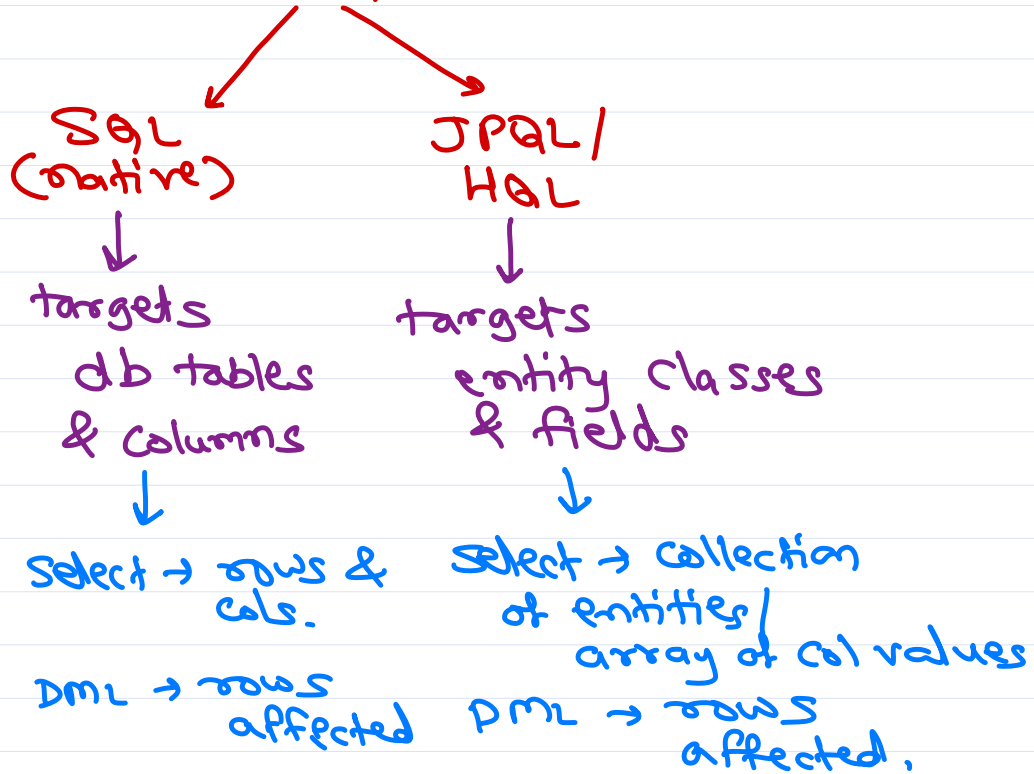
when FK column is PK of the table.

```
class Emp {  
    @Id  
    int empno;  
    ~  
    ~  
    @OneToOne(mappedBy = "emp")  
    Address addr;  
}
```

Mobiles Application



Custom queries



In Spring Data JPQL query written as `@Query("—")` & SQL query as `@Query(value="—", nativeQuery=true)`

delete from Mobile m where m.company = ?

delete from Mobile m where m.ram < ?
and m.storage < ?

update Mobile m set m.price = m.price + 1000
where m.company = ?

Select m from Mobile m where m.company = ?

select m from Mobile m where m.price
between ? and ?

select m.company, sum(m.price) as total
from Mobile m group by m.company

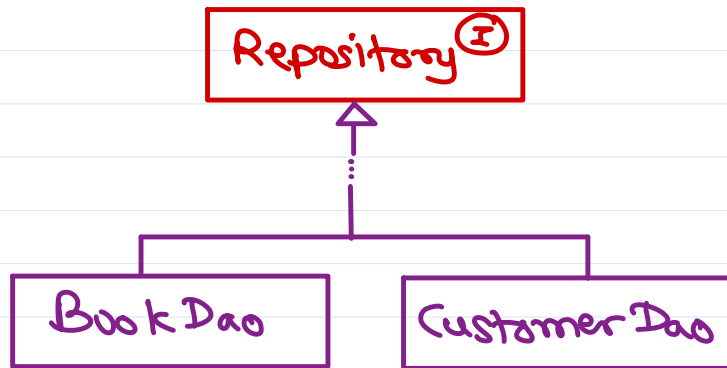
select m from Mobile m where m.ram > ?
and m.storage > ? order by m.price desc.

select m.name, m.company, m.price from Mobile m.

→ DML queries should have `@Modifying` & called in `@Transactional` context.



Spring Data Internals

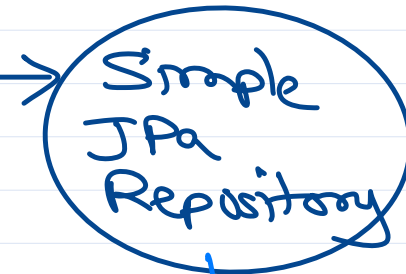


```
@Autowired
BookDao dao;
```

⋮

```
list = dao.findBySubject(m);
```

dao



SimpleJpaRepository

@Repository
@Transactional

Dao/Repository impl class given by spring data framework.
Uses proxy concept to see method calls & generate Java code to execute ORM ops/DB ops.

Checks method name & produces appropriate SQL query & collect result in required return type.





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

