

```

@Service
public class AppService {
    @Autowired
    private BookRepo bookRepo;
}

@Service
public class AdminService {
    @Autowired
    private BookRepo bookRepo;
}

```

```

@Repository
@Scope("prototype") ✓
//@Profile("prod")
public class BookRepoDbImpl implements BookRepo{
    ...
}

```

```

@Service
public class AppService {
    @Autowired
    private BookRepo bookRepo;
}

@Service
public class AdminService {
    @Autowired
    private BookRepo bookRepo;
}

```

```

@Repository
@Scope("prototype") ✓
//@Profile("prod")
public class BookRepoDbImpl implements BookRepo{
    ...
}

```

1 Request

```

@Repository
@RequestScope ✓
public class BookRepoDbImpl implements BookRepo{
    ...
}

```

2 Request

```

@Repository
@RequestScope ✓
public class BookRepoDbImpl implements BookRepo{
    ...
}

```

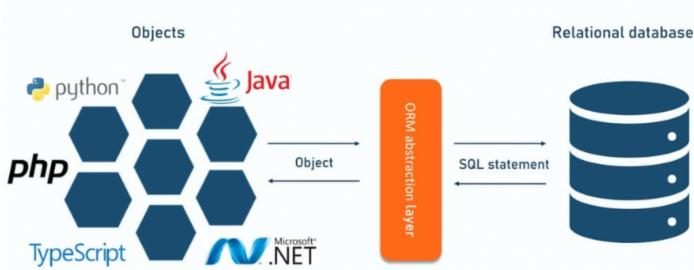
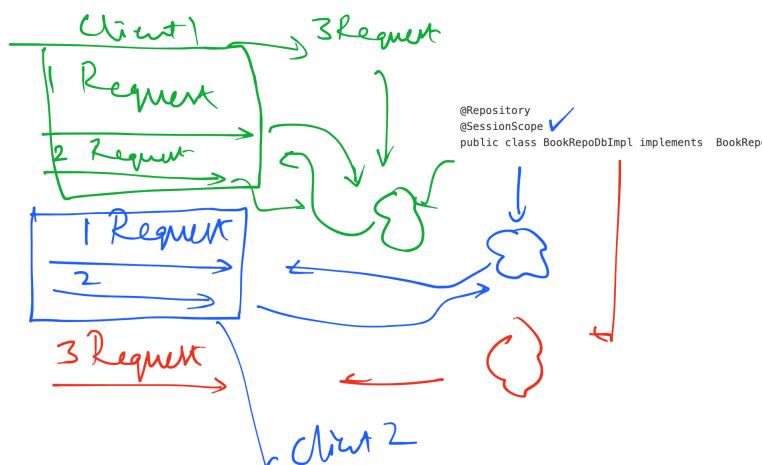
3 Request

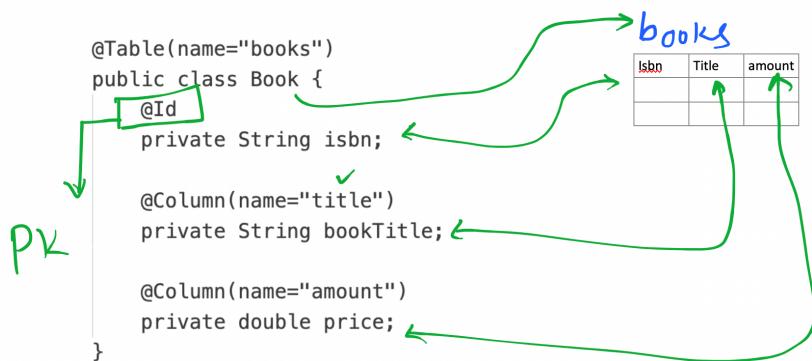
```

@Repository
@RequestScope ✓
public class BookRepoDbImpl implements BookRepo{
    ...
}

```

Session Scope:



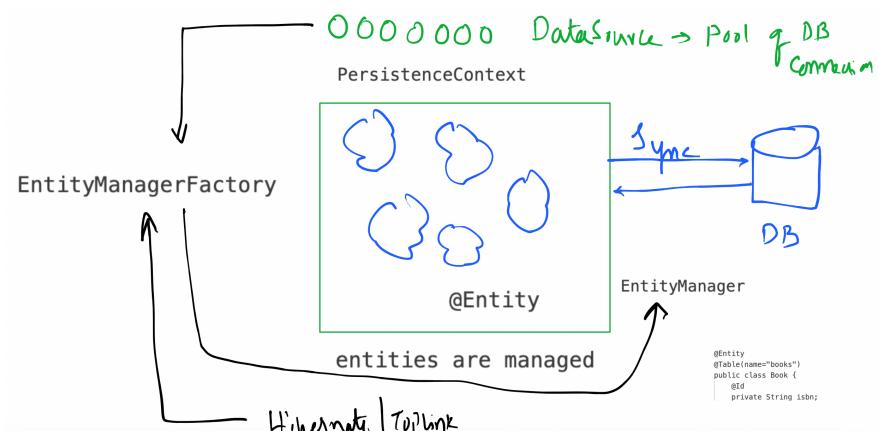
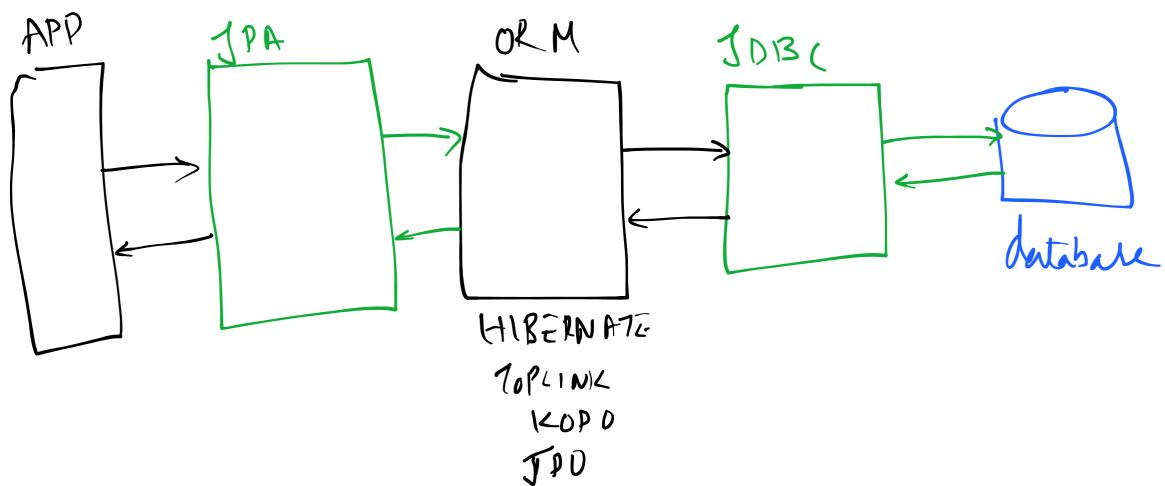


JDBC

```
    @Override
    public void addProduct(Product product) throws DaoException {
        Connection con = null;
        try {
            String SQL = "INSERT INTO products (id, name, price, category) VALUES (?, ?, ?, ?)";
            PreparedStatement ps = con.prepareStatement(SQL);
            ps.setString(1, product.getName());
            ps.setDouble(2, product.getPrice());
            ps.setInt(3, product.getQuantity());
            ps.executeUpdate();
        } catch (SQLException e) {
            if(ex.getErrorCode() == 1062) {
                throw new DaoException("Product with given id: " + product.getId() + " already exists!");
            } else if(ex.getErrorCode() == 1054) {
                throw new DaoException("Unable to add Product!!!");
            }
        } finally {
            DBUtil.closeConnection(con);
        }
    }
}
```

BRM

```
public void addProduct(Product product)
    em.persist(product);
}
```



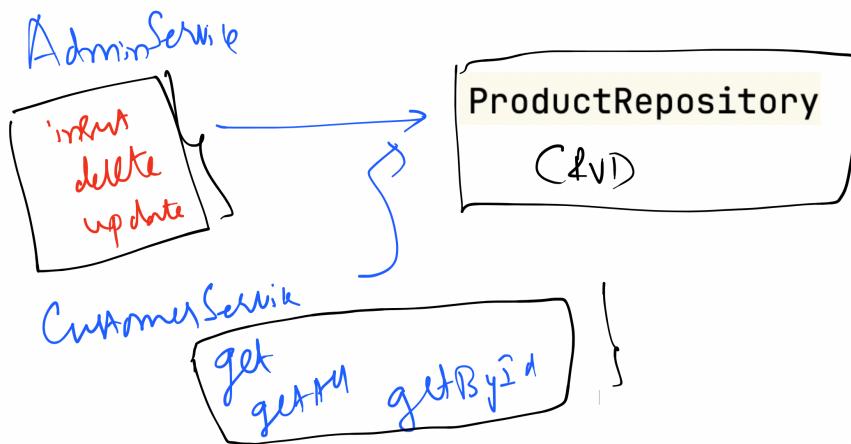
```
mysql> desc customers;
+-----+-----+-----+
| Field | Type   | Null | Key |
+-----+-----+-----+
| email | varchar(255) | NO   | PRI |
| fname | varchar(255) | YES  |       |
| lname | varchar(255) | YES  |       |
+-----+-----+-----+
```

```
@Entity
@Table(name="customers")
@NoArgsConstructor
//@RequiredArgsConstructor
@AllArgsConstructor
@Data
@Builder
public class Customer {
    @Id
    private String email;

    @Column(name="FNAME")
    private String firstName;

    @Column(name = "LNAME")
    private String lastName;
}
```

SQL [table and column names]	JP-QL [class and field names, case sensitive]
select * from customers	from Customer
select * from customers where lname='Rathod'	from Customer where lastName="Rathod"
Select email, fname from customers	select email, firstName from Customer



OrderService

```
@Service  
public class OrderService {  
    @Autowired  
    private ProductRepository productRepository; // create  
  
    public List<Product> getProducts() { return productRepository.findAll(); }  
  
    public Product getProductById(int id) {  
        Optional<Product> opt = productRepository.findById(id);  
        if(opt.isPresent()) {  
            return opt.get();  
        }  
        return productRepository.findById(id).get();  
    }  
  
    public Product addProduct(Product p) {  
        return productRepository.save(p);  
    }  
}
```

```
public interface ProductRepository extends JpaRepository<Product, Integer> {}
```

↓ class & Repository

findAll

save

findById