

AdminService

```
@Autowired  
@Qualifier("employeeDaoMongoImpl")  
private EmployeeDao employeeDao;
```

```
@Repository  
public class EmployeeDaoMongoImpl implements EmployeeDao {
```

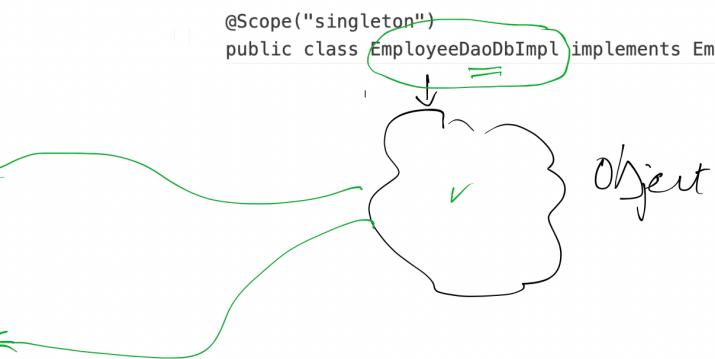
```
@Repository  
public class EmployeeDaoDbImpl implements EmployeeDao {
```

CustomerService

```
@Autowired  
@Qualifier("employeeDaoDbImpl")  
private EmployeeDao employeeDao;
```

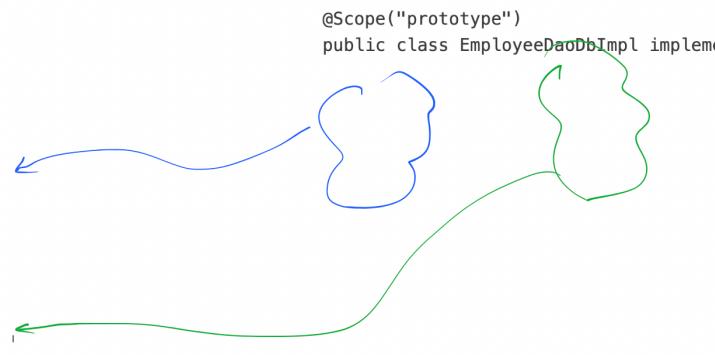
Singleton Scope:

```
@Service  
public AService {  
    @Autowired  
    EmployeeDao empDao; }  
  
@Service  
public BService {  
    @Autowired  
    EmployeeDao empDao; }
```

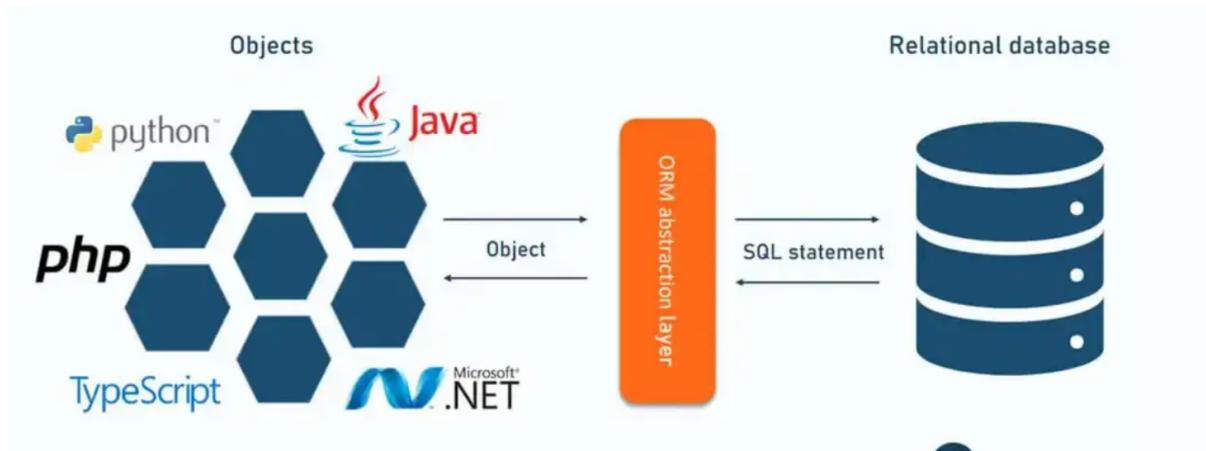


```
@Scope("singleton")  
public class EmployeeDaoDbImpl implements EmployeeDao {
```

```
@Service  
public AService {  
    @Autowired  
    EmployeeDao empDao; }  
  
@Service  
public BService {  
    @Autowired  
    EmployeeDao empDao; }
```



```
@Scope("prototype")  
public class EmployeeDaoDbImpl implements EmployeeDao {
```



email	FNAME	LNAME

```

@NoArgsConstructor
@AllArgsConstructor
@Data
@Entity
@Table("products")
public class Product {
    @Id
    private int id;
    private String name;
    private double price;
    @Column(name="qty")
    private int quantity;
}

```

Products

id	name	price	qty

SQL → ORM

```

@NoArgsConstructor
@AllArgsConstructor
@Data
@Entity
@Table("customers")
public class Customer {
    @Id
    private String email;
    @Column(name="FNAME")
    private String firstName;
    @Column(name="LNAME")
    private String lastName;
}

```

Customers

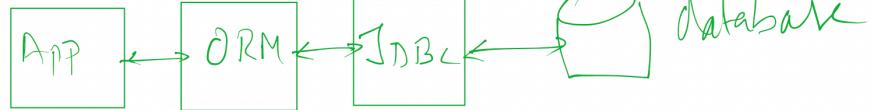
email	FNAME	LNAME

JDBC

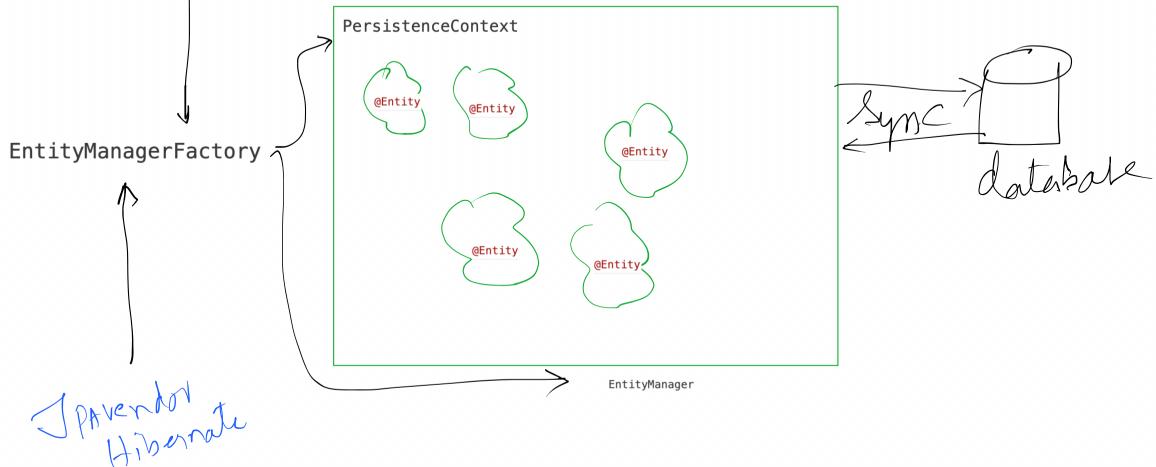
```
@Override  
public void addProduct(Product p) {  
    String SQL = "INSERT INTO products (id, name, price) VALUES(?, ?, ?)" // IN parameter  
    Connection con = null;  
  
    try {  
        con = DriverManager.getConnection(URL, USER, PWD);  
        PreparedStatement ps = con.prepareStatement(SQL);  
        ps.setString(1, p.getName());  
        ps.setDouble(2, p.getPrice());  
        ps.executeUpdate(); // INSERT, UPDATE and DELETE  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    } finally {  
        if( con!= null) {  
            try {  
                con.close();  
            } catch (SQLException e) {  
                throw new RuntimeException(e);  
            }  
        }  
    }  
}
```

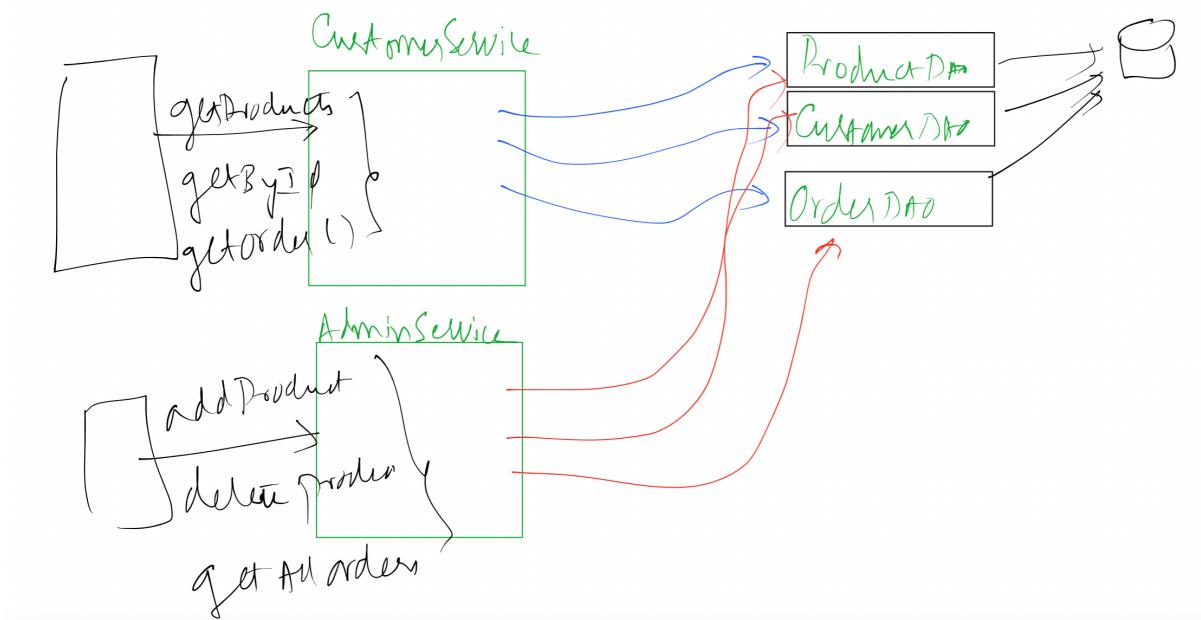
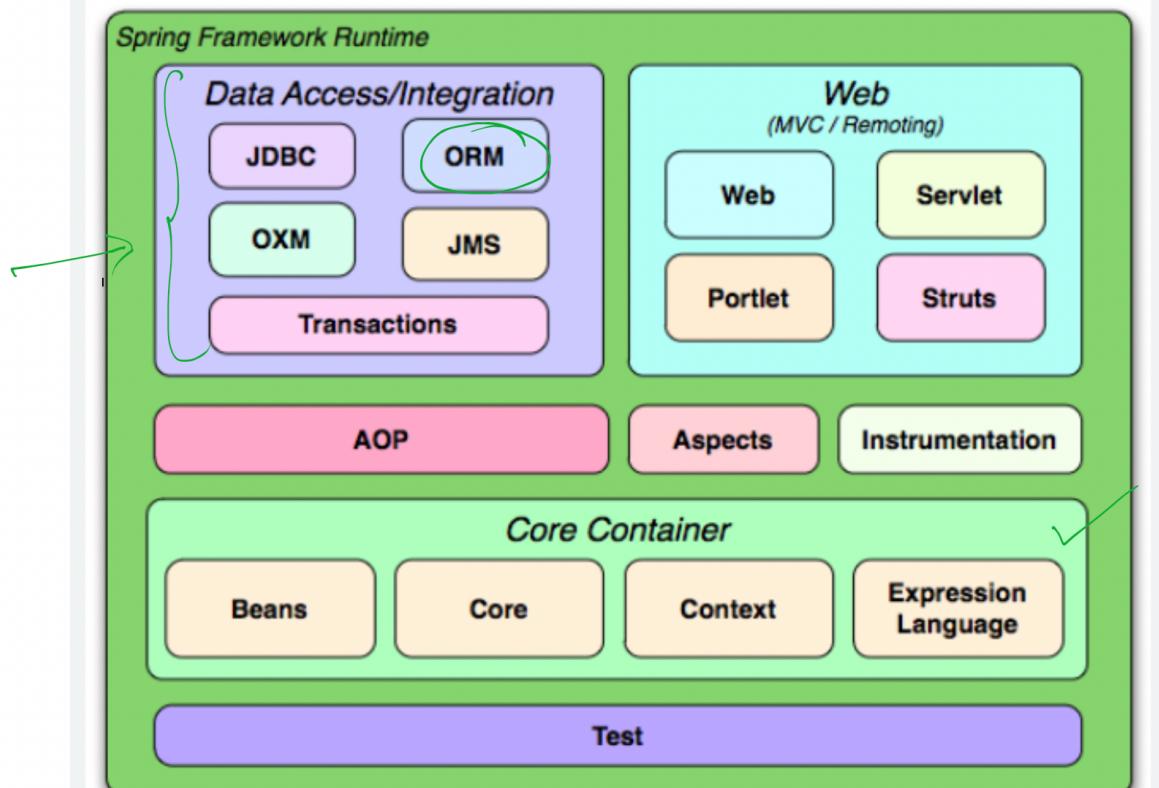
ORM

```
@Override  
public void addProduct(Product p) {  
    em.persist(p);  
}
```



00000000 Datasource





SQL

```
mysql> select * from customers;
+-----+-----+-----+
| email | fname | lname |
+-----+-----+-----+
| angelina@cisco.com | Angelina | Jolie |
| anne@cisco.com | Anne | Hathaway |
| roger@cisco.com | Roger | Smith |
+-----+-----+-----+
```

```
select * from customers;
```

```
select * from customers where fname = 'Roger'
```

```
select email, fname from customers
```

JP-QL

```
@Entity
@Table(name="customers")
public class Customer {
    @Id
    private String email;

    @Column(name="FNAME", length = 100, nullable = false)
    private String firstName;

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

```
from Customer
```

```
from Customer where firstName = 'Roger'
```

```
select email, firstName from Customer
```

} Entity

← Scalar Values
Object []

```
mysql> select * from customers;
+-----+-----+-----+
| email | fname | lname |
+-----+-----+-----+
| angelina@cisco.com | Angelina | Jolie |
| anne@cisco.com | Anne | Hathaway |
| roger@cisco.com | Roger | Smith |
+-----+-----+-----+
```

orders

oid	order_date	customer_fk	total
890	20-4-2025 4:30:00	anne@cisco.com ✓	<<Rs.####>>
891	20-4-2025 5:14:12	angelina@cisco.com	<<Rs.####>>
901	21-4-2025 10:10:11	anne@cisco.com ✓	<<Rs.####>>

```
mysql> select * from products;
```

```
+---+-----+-----+-----+
| id | name | price | qty |
+---+-----+-----+-----+
| 1 | Logitech Mouse | 679 | 100 |
| 2 | Tata Play | 8900.5 | 100 |
| 3 | LG AC | 45000 | 99 |
| 4 | Term 1 | 8990 | 100 |
+---+-----+-----+-----+
```

line-items

item_id	Order_fk	Product_fk	Qty	amount
98	890 ✓	3 ✓	1	42000.00
99	890 ✓	1 ✓	2	1000.00
100	891	3 ✓	2	90000.00

