

AdminService

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

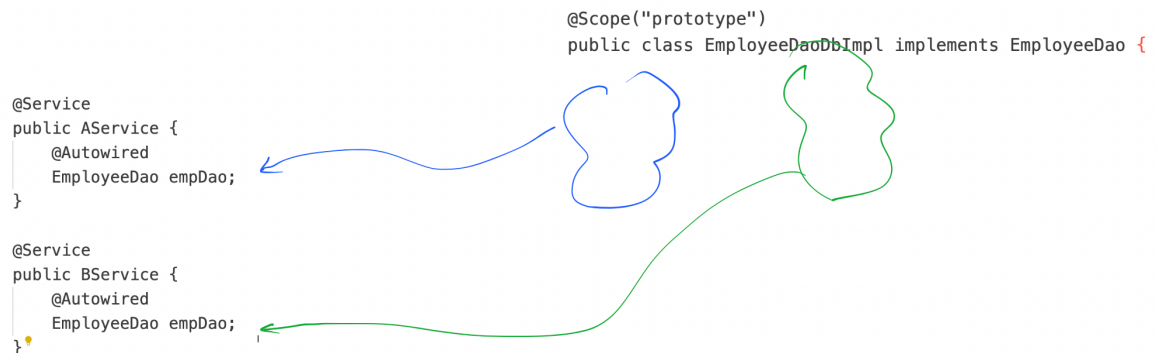
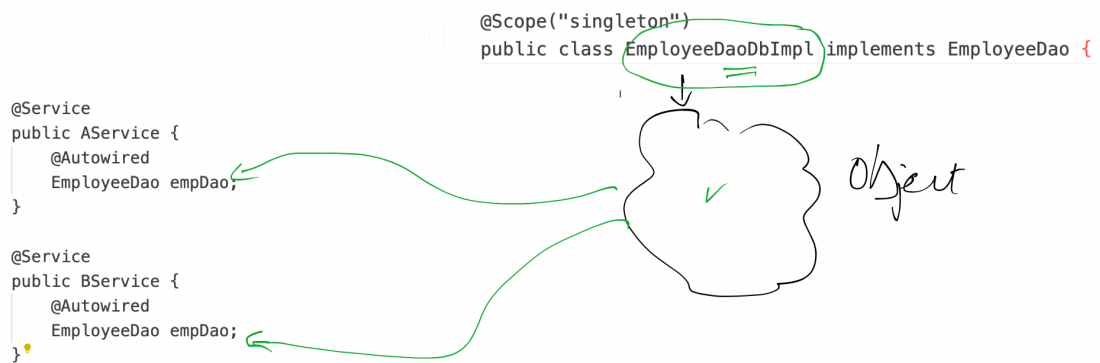
```
@Repository
public class EmployeeDaoMongoImpl implements EmployeeDao{
```

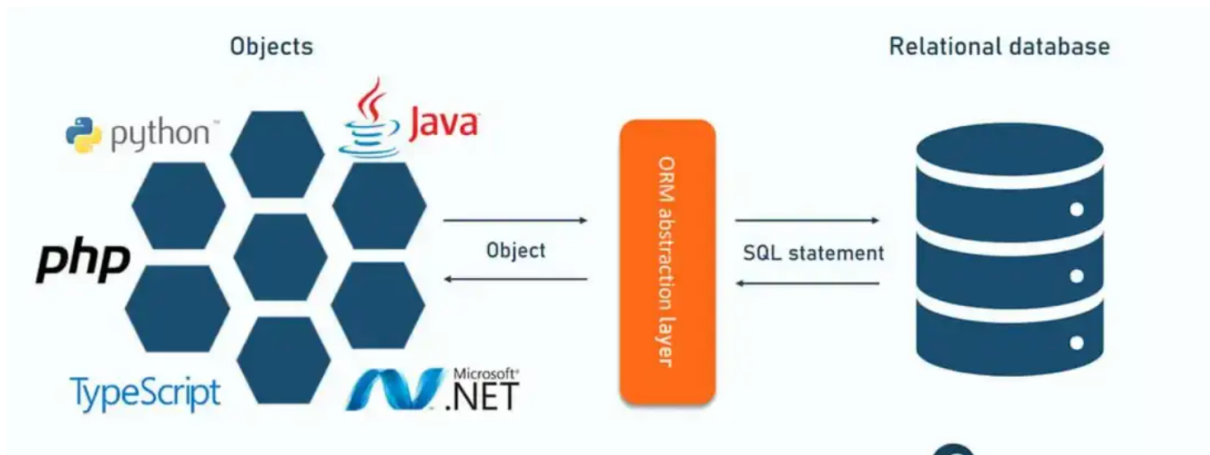
```
@Repository
public class EmployeeDaoDbImpl implements EmployeeDao{
```

CustomerService

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

Singleton Scope:





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(0, ?, ?)"; // 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);
}
```

