




Day4; 20220906

📅 날짜	
📁 유형	
🏷️ 태그	


<Spring> properties, annotation(component, autowired, qualifier, resource), DAO, DTO(DO), service, CRUD(명령문 약어)

spring/src at main · u8yes/spring

Contribute to u8yes/spring development by creating an account on GitHub.

 <https://github.com/u8yes/spring/tree/main/src>

u8yes/**spring**



Alt 1 Contributor 0 Issues ☆ 1 Star 0 Forks

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a6ddd14e-a928-4dae-ac01-a9cf27f4e181/04%EC%9E%A5._Dependency_Injection\(%EC%9D%98%EC%A1%B4%EC%84%B1_%EC%A3%BC%EC%9E%85\).pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a6ddd14e-a928-4dae-ac01-a9cf27f4e181/04%EC%9E%A5._Dependency_Injection(%EC%9D%98%EC%A1%B4%EC%84%B1_%EC%A3%BC%EC%9E%85).pdf)

SQLD 자격증 내용

<BOOK> 시험장에 몰래 가져갈 이경오의 SQL+SQLD 비밀노트, 국가 공인 SQL 개발자 자격증

<https://www.hanbit.co.kr/src/10436>

패스트캠퍼스 [직장인 실무교육]

프로그래밍, 영상편집, UX/UI, 마케팅, 데이터 분석, 엑셀강의, The RED, 국
비지원, 기업교육, 서비스 제공.

 <https://fastcampus.co.kr/>




책에서 추천하는 SQL Developer

SQL Developer 22.2 Downloads

SQL Developer 22.2 Downloads


Version 22.2.0.173.2018 - June 28, 2022

- [Release Notes](#)
- [Documentation](#)

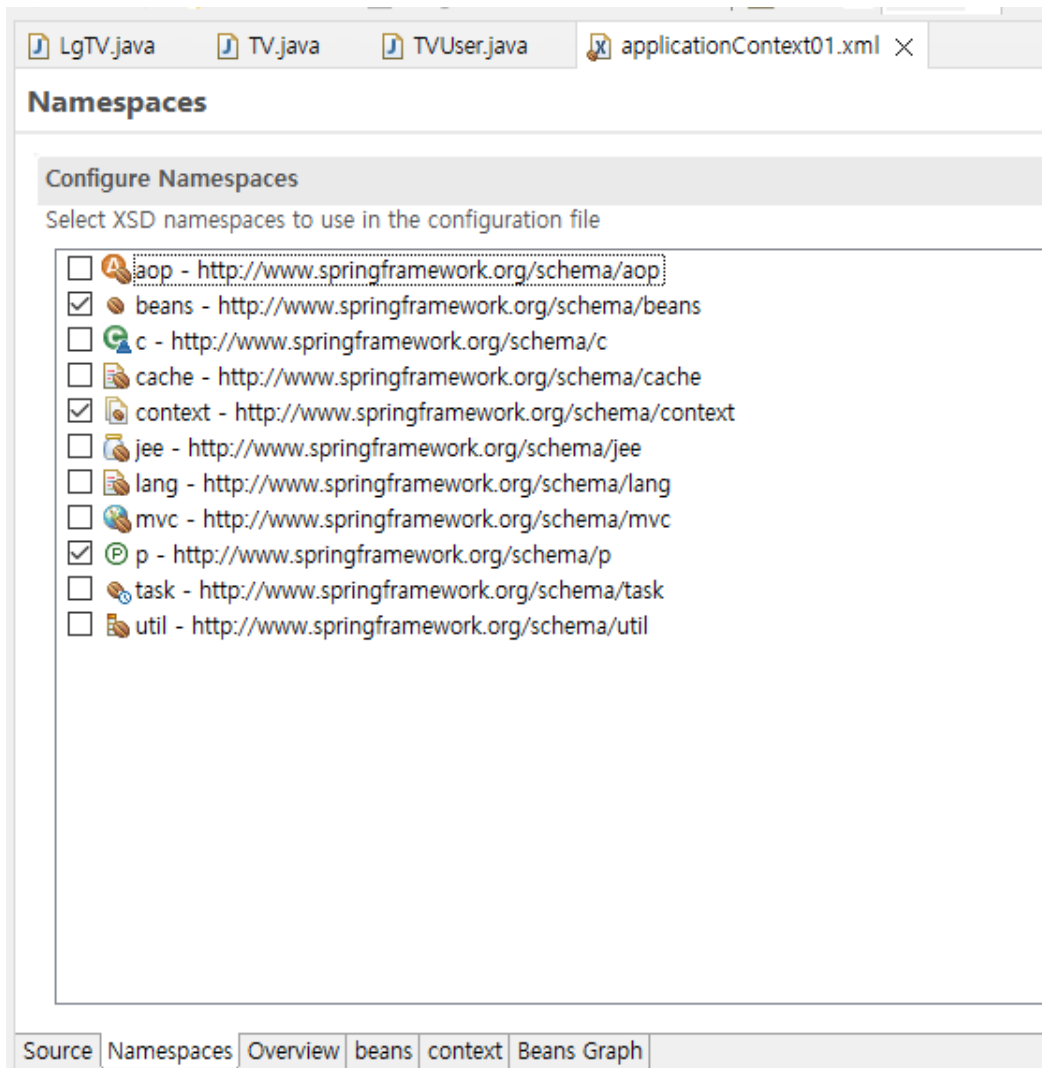
Platform	Download
Windows 64-bit with JDK 11 included	 Download (434 MB)

SQL Developer Downloads

Oracle SQL Developer Downloads

 <https://www.oracle.com/database/sqldeveloper/technologies/download/>

설정파일에서 추가해주는 방법 namespaces TAP



추가됨.

```
LgTV.java TV.java TVUser.java applicationContext01.xml ×
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans h
7         http://www.springframework.org/schema/context http://www.spri
8
9 <bean id="tv" class="ex01.annotation.component.LgTV">
10
11
12 </bean>
13
14
15 </beans>
16
```

< Namespaces Overview beans context Beans Graph

[Annotation]

```
<context:component-scan base-package="ex01.annotation.component"></context:component-scan>
```

```
+
; @Component("tv")
; public class LgTV implements TV{
```

```
TV tv = (TV)factory.getBean("tv");
```

<bean>자체가 사라지게 할 수 있다.

p의 예를 보면 이전에는 property로 작성해줬어야만 했던 것을 속성으로 변경해준다. <property> 사라짐

```
applicationContext05.xml × applicationContext06.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.s
5
6 <bean id="tv" class="ex05.springcontainer.di.setter.LgTV">
7     <property name="speaker" ref="sony" /> <!-- setter 메서드 전용 -->
8     <property name="price" value="500000" /> <!-- setter 메서드 전용 -->
9 </bean>
```

```
<bean id="tv" class="ex06.springcontainer.di.xmlns.p.LgTV"
p:speaker-ref="sony" p:price="300000"/>
```

참조만 ref, 기본은(ref) X

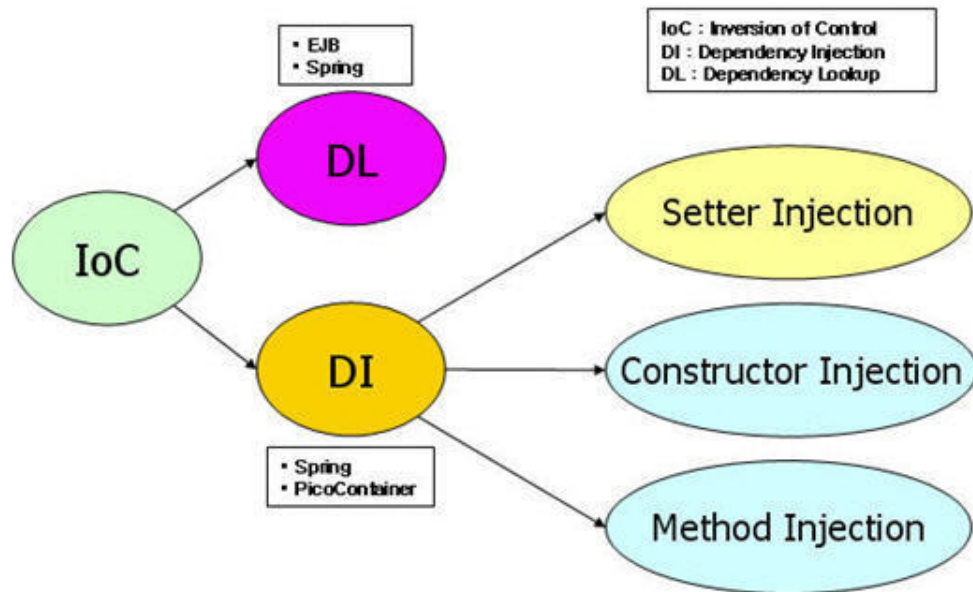
DL(Dependency Lookup)

DL(Dependency Lookup)IoC 컨테이너는 각 컨테이너에서 관리해야 하는 객체들을 관리하기 위한 별도의 저장소를 가지게 되는데,

이 IoC컨테이너가 관리중인 객체 저장소(Pool)에서 객체를 검색하여 참조하는 방법이다.

(컨테이너에서 제공하는 api를 이용하여 찾고자 하는 bean을 lookup하는 방법)

Dependency Injection(의존성 주입)은 자동으로 주입되는 것.



New 객체를 생성해주는 Bean을 또 생성해주는 annotation들

따라서 New 객체와 Bean은 사라지게 됨.

```

5 import org.springframework.stereotype.Component;
6
7 @Component("tv")
8 public class LgTV implements TV{
9
10     //Speaker speaker = new SonySpeaker();
11     @Autowired
12     @Qualifier("apple")
13     Speaker speaker;
14
15     public LgTV() { // 초기화할 목적으로 생성자
16         // speaker = new SonySpeaker();
17     }
18 }
  
```

```

@Component("apple")
public class AppleSpeaker implements Speaker

5 @Component("sony")
6 public class SonySpeaker implements Speaker {

```

<context:component-scan base-package="ex02.annotation.autowired">

자바 코드

```

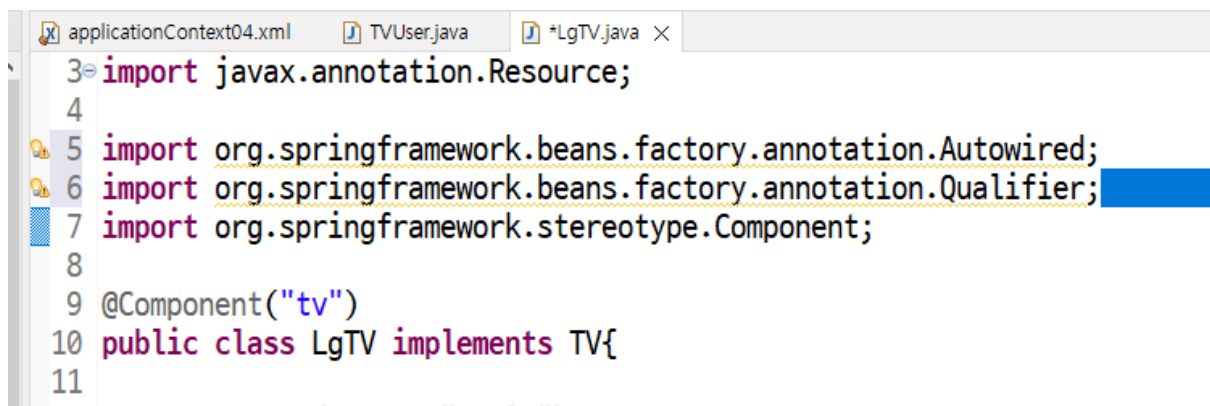
public class WordRegisterServiceUseAutowired {

    @Autowired
    @Qualifier("usedDao")
    private WordDao wordDao;

    public WordRegisterServiceUseAutowired() { }
}

```

Ctrl + Shift + O를 누르면 알아서 import 변경을 해준다.



```

applicationContext04.xml TVUser.java *LgTV.java ×
3 import javax.annotation.Resource;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.beans.factory.annotation.Qualifier;
7 import org.springframework.stereotype.Component;
8
9 @Component("tv")
10 public class LgTV implements TV{
11

```




```
applicationContext04.xml TVUser.java *LgTV.java X
1 package ex04.annotation.resource;
2
3 import javax.annotation.Resource;
4
5 import org.springframework.stereotype.Component;
6
7 @Component("tv")
8 public class LgTV implements TV {
```

import 2개가 사라진 것을 알 수 있음

@Autowired와 @qualifier를 한 줄로 줄여주는 annotation

@Resource

```

> import org.springframework.stereotype
>
> @Component("tv")
> public class LgTV implements TV{
>
>     @Resource(name = "apple")
>     Speaker speaker;
>
>

```

```

AppleSpeaker 객체 생성.
LgTV 객체 생성.
SonySpeaker 객체 생성.
LgTV power On.
AppleSpeaker---소리 올린다.
AppleSpeaker---소리 내린다.
LgTV power Off.

```

@Autowired

```

<bean class="ex05.annotation.xml.annotation.AppleSpeaker"></bean>

```

```

<context:component-scan base-package="ex05.annotation.xml.annotation"></context:component-scan>

```

```

<bean class="ex05.annotation.xml.annotation.AppleSpeaker"></bean>

```

```

5
6 @Component("tv")
7 public class LgTV imp
8
9 @Autowired
0 Speaker speaker;
1

```

```

LgTV 객체 생성.
AppleSpeaker 객체 생성.
LgTV power On.
AppleSpeaker---소리 올린다.
AppleSpeaker---소리 내린다.
LgTV power Off.

```

스피커 어노테이션 사라짐

```

1 package ex05.annotation.xml.annotation;
2
3 import org.springframework.stereotype.Component;
4
5 //@Component("apple")
6 public class AppleSpeaker implements Speaker {
7
8 public AppleSpeaker() { // 생성자는 클래스와 동
9     // 생성자는 반환할 것이 없다.
0     System.out.println("AppleSpeaker 객체 생성
1 }

```

```

1 package ex05.annotation.xml.annotation;
2
3 import org.springframework.stereotype.Component;
4
5 //@Component("sony")
6 public class SonySpeaker implements Speaker {
7
8 public SonySpeaker() { // 생성자는 클래스와 동일한 0
9     // 생성자는 반환할 것이 없다.
0     System.out.println("SonySpeaker 객체 생성.");
1 }

```

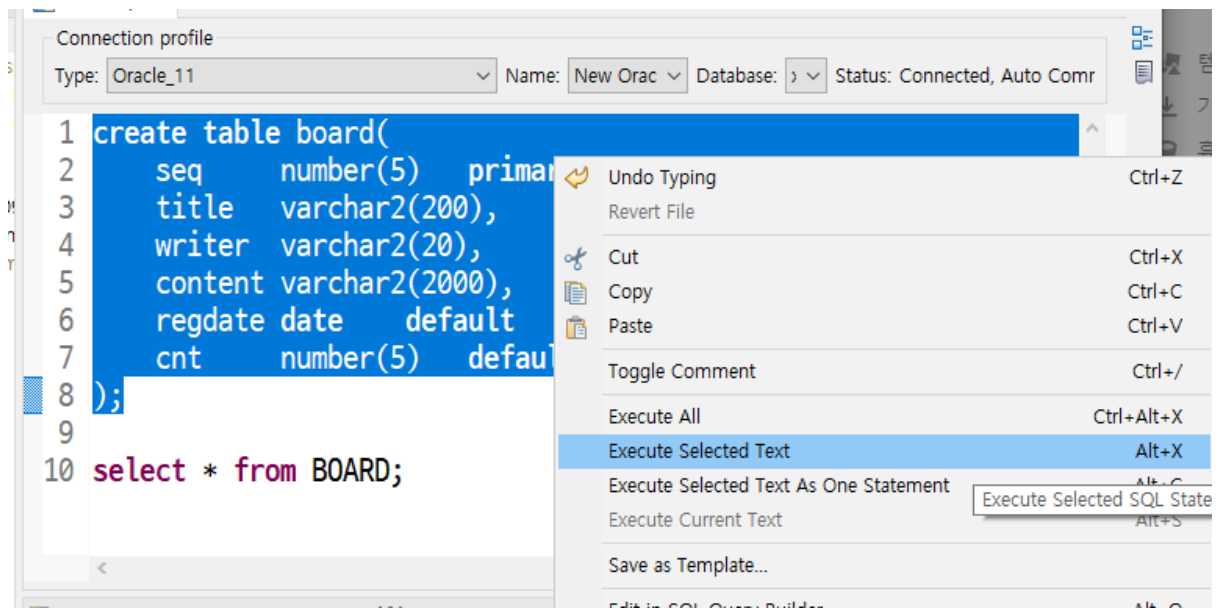
Database → 서버로 화면에 구현할 때,
모든 프로젝트를 진행할 때는 '계획서'가 있다.

네이버 게시판을 server에서 DB로 보내는데 서버로 보내는데 그때 JAVABEAN이라고 함.

1. 그것이 VO(DTO).
JDBC가 이것을 가능하게 해줌.
 2. DB에 CRUD(create read update delete)해주는 것은 DAO작업.
DAO를 가능하게 해주는 것을 Service라고 부름.
-

varchar2를 배리어블 캐릭터라고 말함.

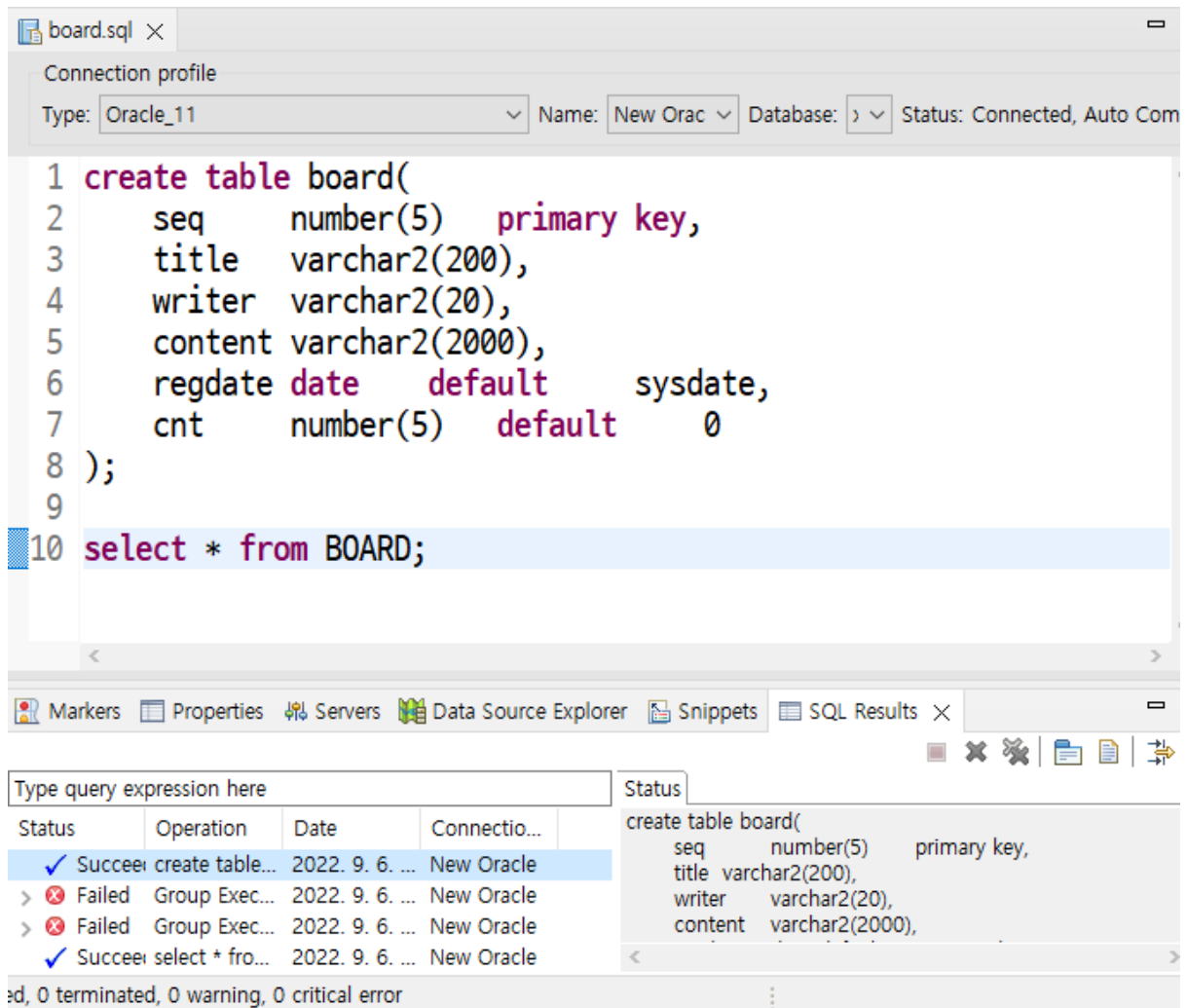
```
create table board(  
  seq    number(5) primary key,  
  title  varchar2(200),  
  writer varchar2(20),  
  content varchar2(2000),  
  regdate date default sysdate,  
  cnt    number(5) default 0  
);
```



Alt + X 누르면 실행됨.

Execute Selected Text	Alt+X
-----------------------	-------

```
select * from BOARD;
```



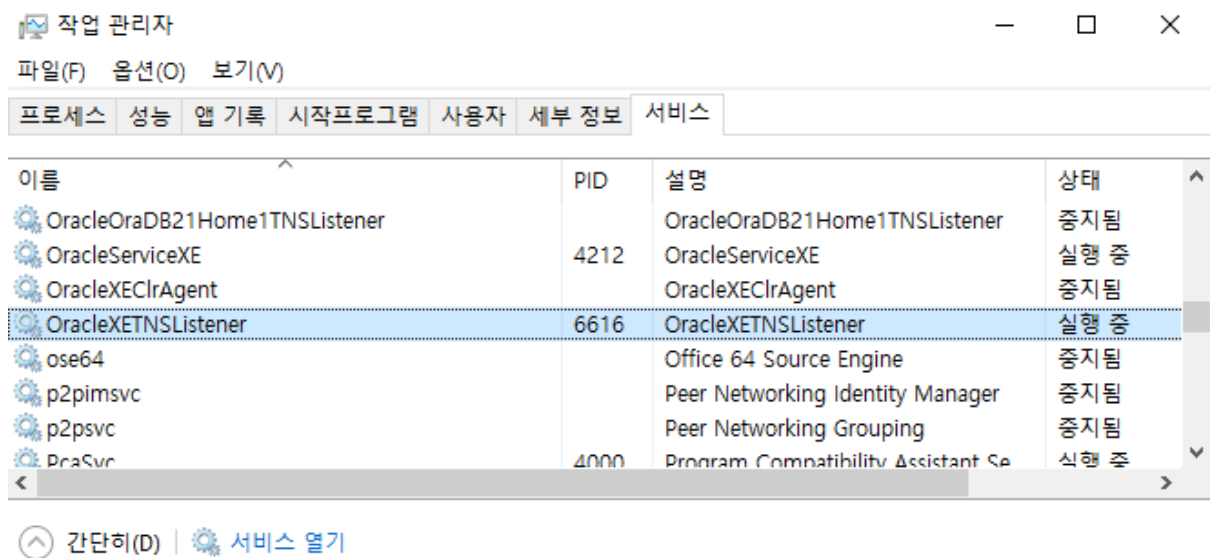
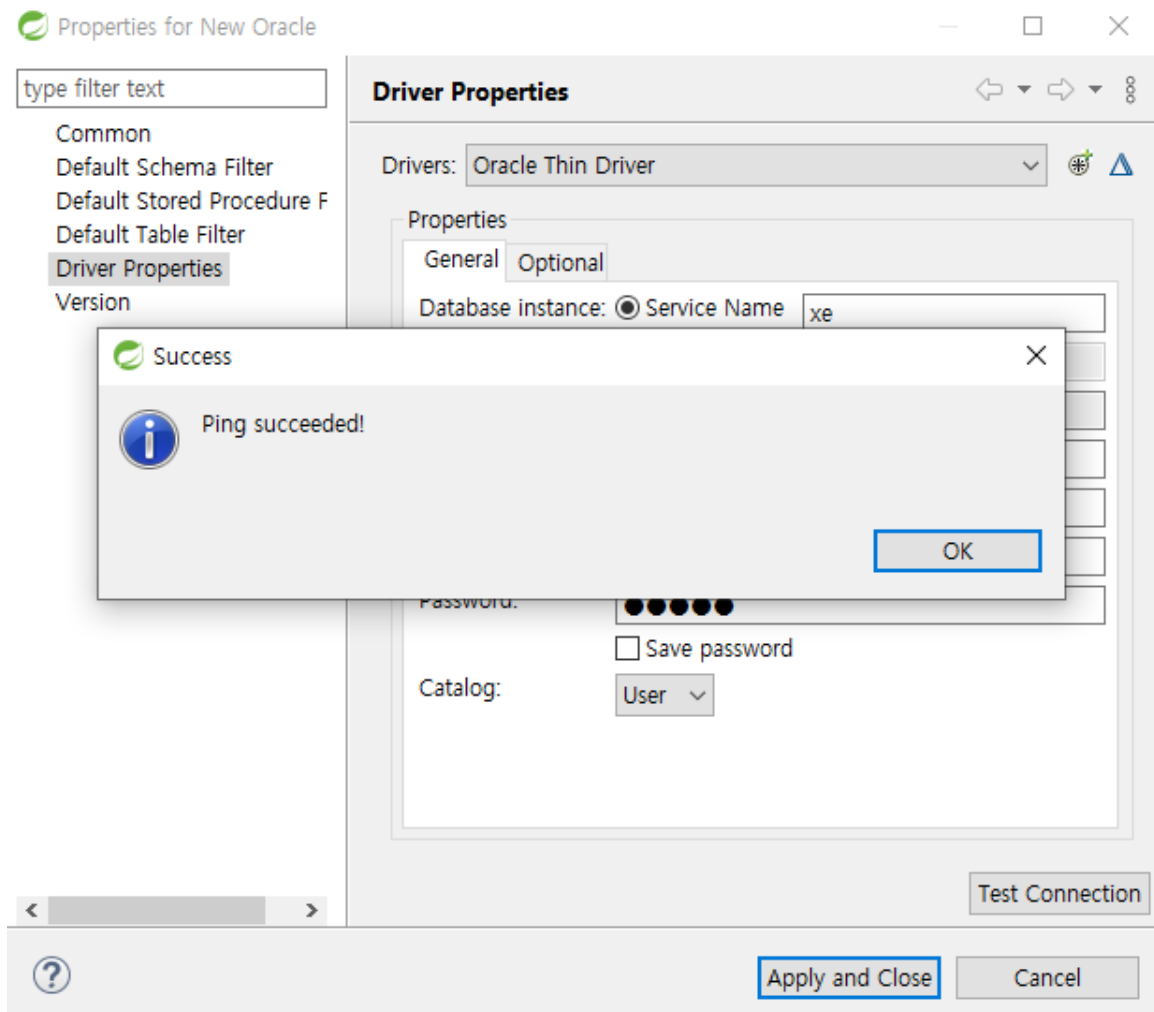
result1으로 직접 볼 수 있다.

sql result 창을 아래쪽으로 끌어와줘야 함.

성공하면 success 보여줌.

아래 오른쪽 창에 결과가 나옴.

Ping failed 했을 경우.



‘패턴’으로 보면 ‘모델’이라고 부르지만

‘일반 개념’에서는 DB와의 연동 작업을 ‘Business Logic’이라고 말한다.

DAO - Access

DTO - Transfer

main → Service → DAO, Business Layer