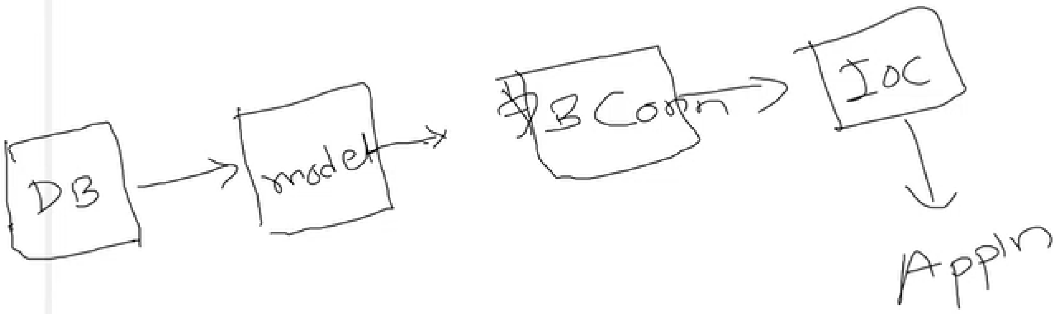
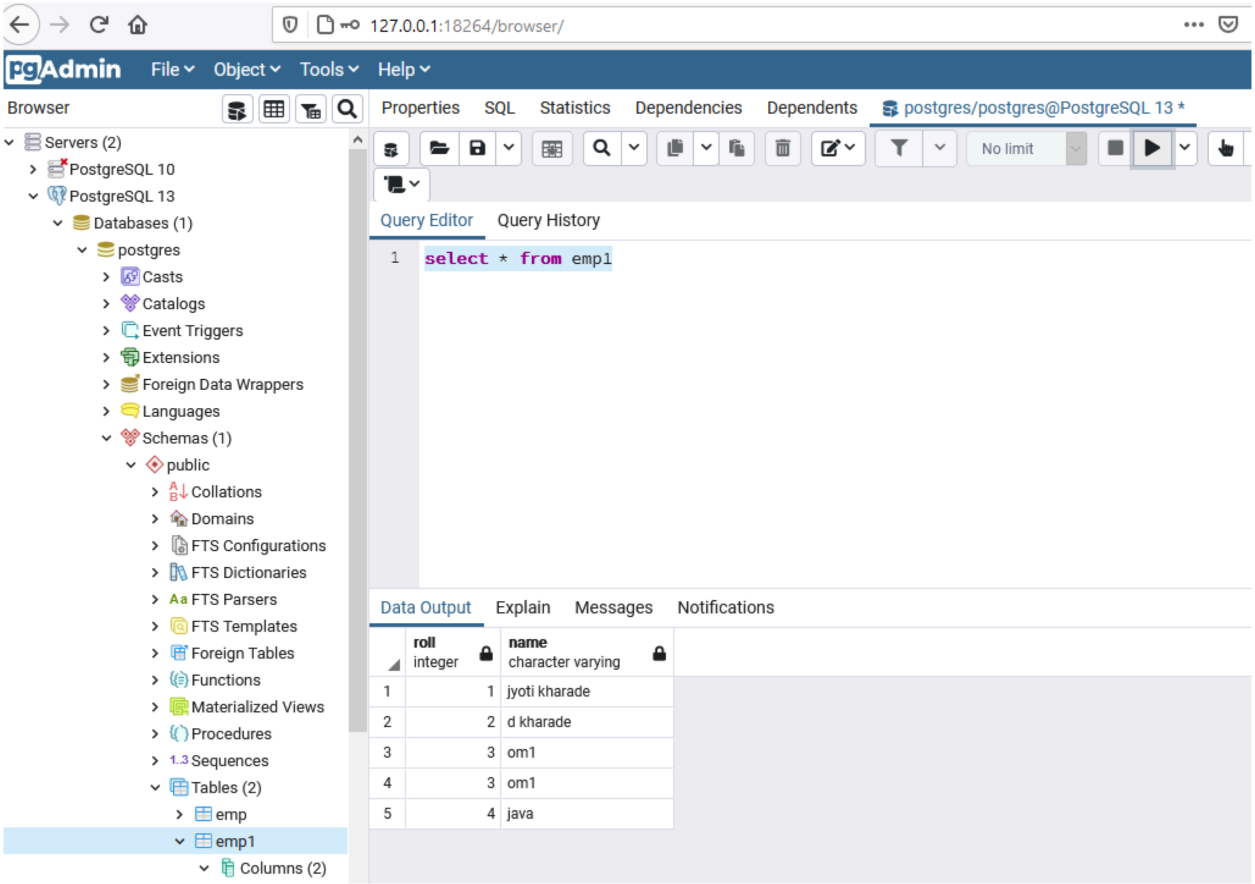


Model which will map with the database(Table) .
This Model and database connection Parameters will be passed to the Spring IoC

Spring and JDBC : Spring legacy Project

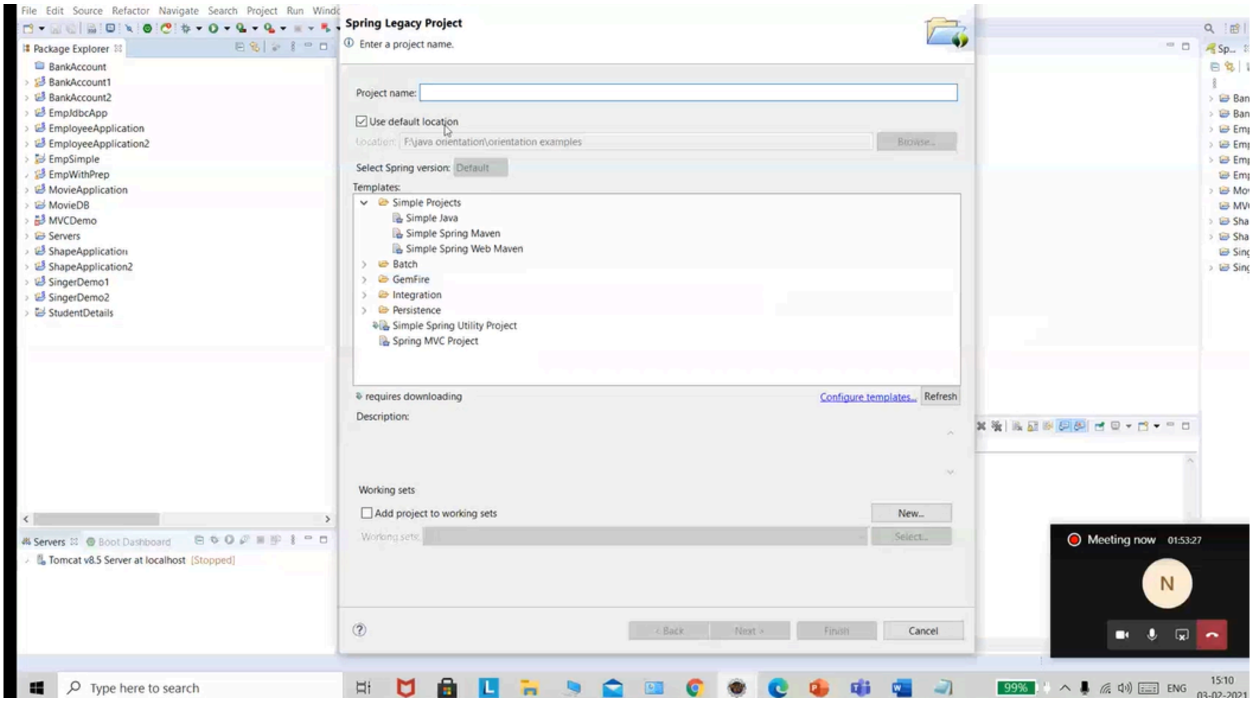
1. Create table emp1in postgreSQL



```
select * from emp1
```

	roll integer	name character varying
1	1	jyoti kharade
2	2	d kharade
3	3	om1
4	3	om1
5	4	java

2. Select New-> other->spring legacy project->simple java



Spring Legacy Project

Enter a project name.

Project name:

☑ Use default location

Location: F:\java orientation\orientation examples

Select Spring version: Default

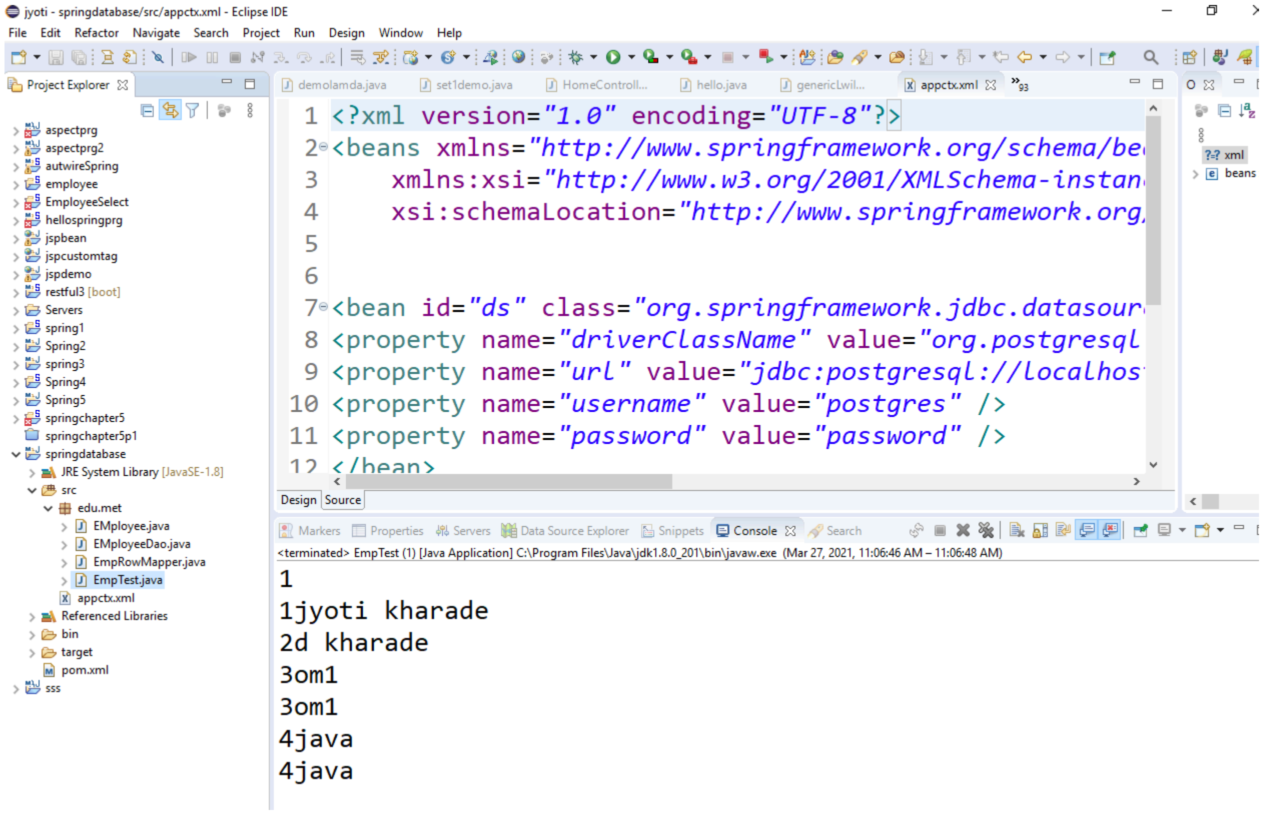
Templates:
- Simple Projects
  - Simple Java
  - Simple Spring Maven
  - Simple Spring Web Maven
- Batch
- GemFire
- Integration
- Persistence
- Simple Spring Utility Project
- Spring MVC Project

requires downloading

Working sets

□ Add project to working sets

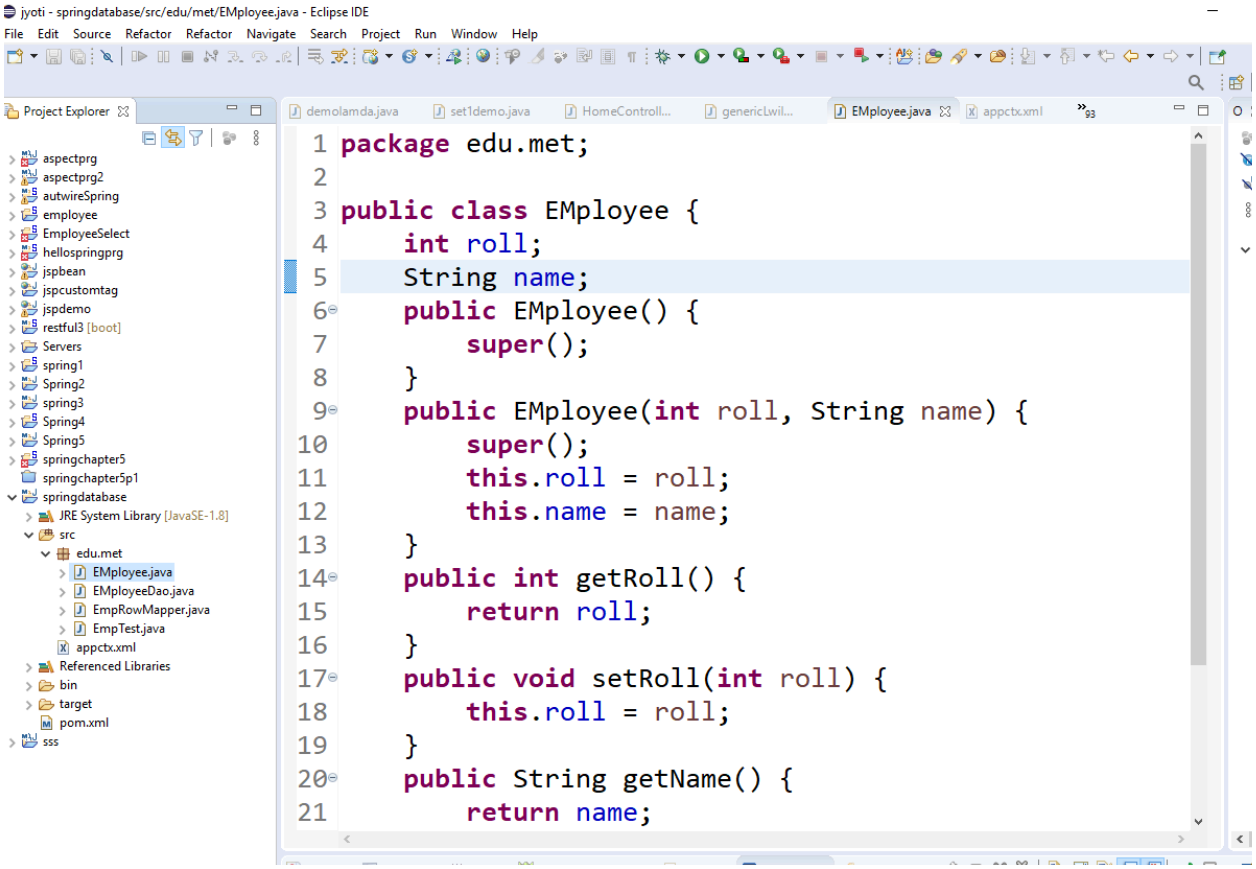
Give name to the project as springdatabase



jyoti - springdatabase/src/appctx.xml - Eclipse IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/be
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instan
    xsi:schemaLocation="http://www.springframework.org,


<bean id="ds" class="org.springframework.jdbc.datasour
<property name="driverClassName" value="org.postgresql
<property name="url" value="jdbc:postgresql://localhos
<property name="username" value="postgres" />
<property name="password" value="password" />
</bean>
```

```
<terminated> EmpTest (1) [Java Application] C:\Program Files\Java\jdk1.8.0_201\bin\javaw.exe (Mar 27, 2021, 11:06:46 AM – 11:06:48 AM)
1
1jyoti kharade
2d kharade
3om1
3om1
4java
4java
```



```java
package edu.met;

public class EMployee {
    int roll;
    String name;
    public EMployee() {
        super();
    }
    public EMployee(int roll, String name) {
        super();
        this.roll = roll;
        this.name = name;
    }
    public int getRoll() {
        return roll;
    }
    public void setRoll(int roll) {
        this.roll = roll;
    }
    public String getName() {
        return name;
```

Create Model Employee with attributes same as in the table emp1 with constructors and getter setter methods.

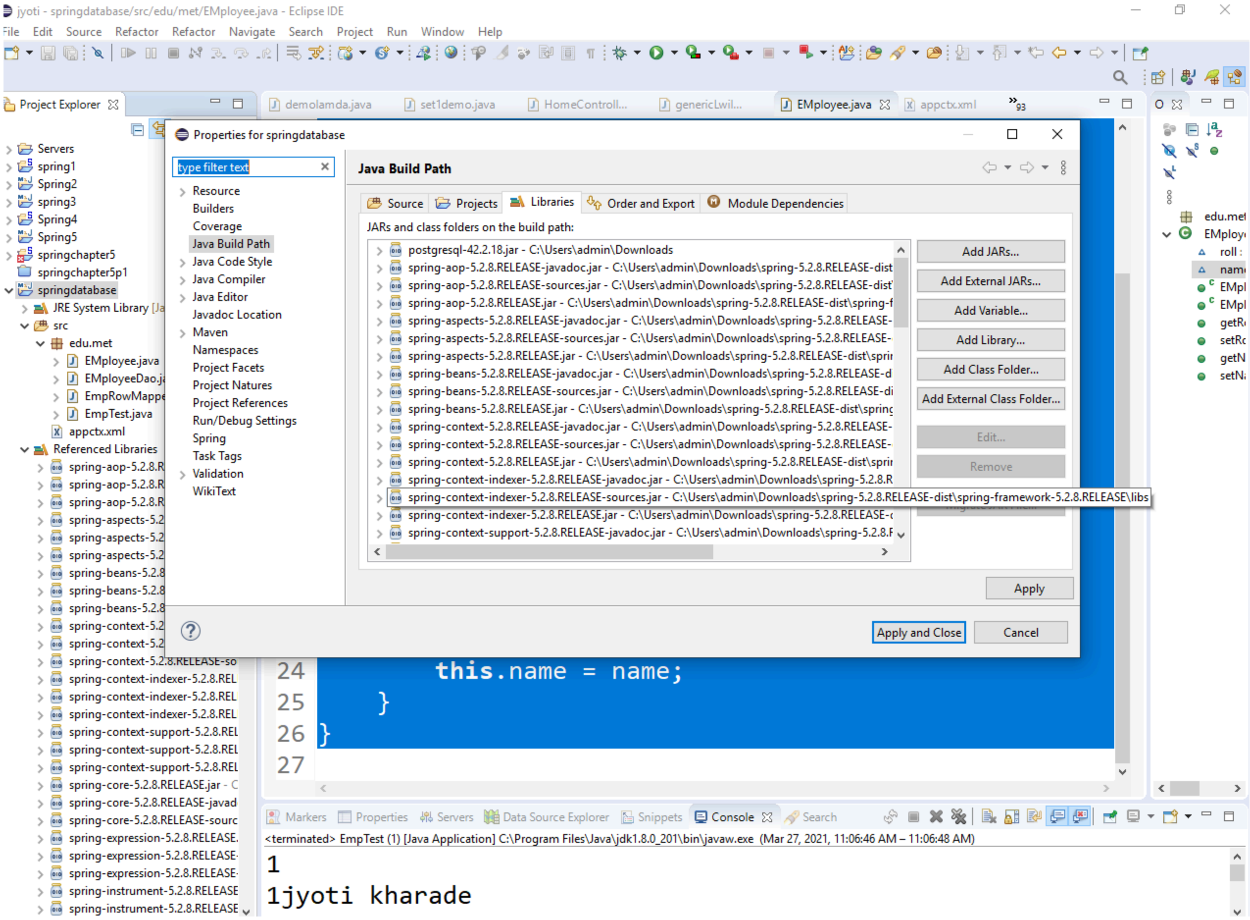
Right click on src -> new package as met.edu
File : EMployee.java
package met.edu;

```java
public class EMployee {
    int roll;
    String name;
    public EMployee() {
        super();
    }
    public EMployee(int roll, String name)
{
        super();
        this.roll = roll;
        this.name = name;
    }
    public int getRoll() {
        return roll;
    }
    public void setRoll(int roll) {
        this.roll = roll;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

Add spring library and jdbc jar files

Right click on springdatabase-> buildpath->configurebuildpath

Select Libraries and add external jar files and click apply



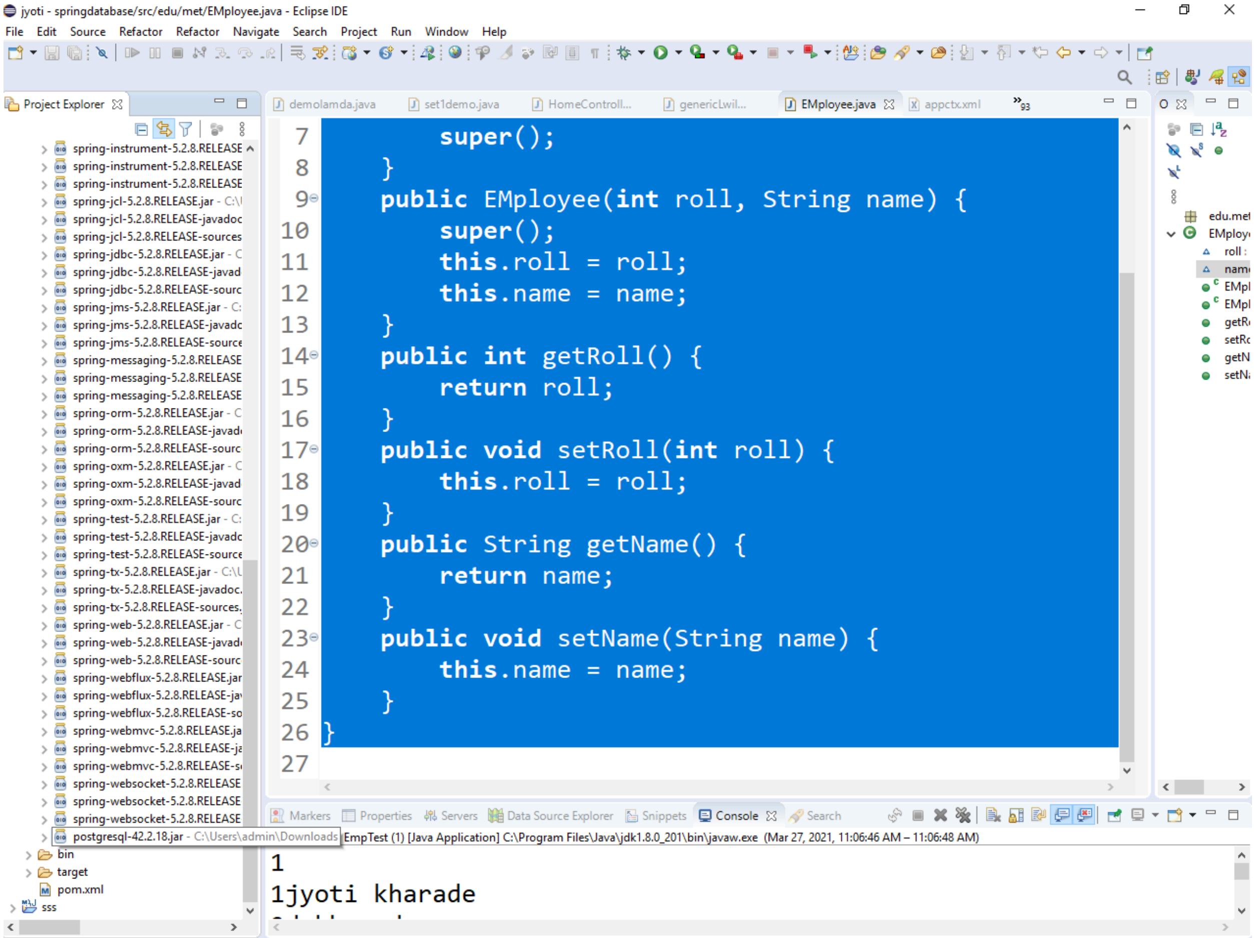
Java Build Path

JARs and class folders on the build path:

- postgresql-42.2.18.jar - C:\Users\admin\Downloads
- spring-aop-5.2.8.RELEASE-javadoc.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist
- spring-aop-5.2.8.RELEASE-sources.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist
- spring-aop-5.2.8.RELEASE.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist\spring-f
- spring-aspects-5.2.8.RELEASE-javadoc.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-
- spring-aspects-5.2.8.RELEASE-sources.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-
- spring-aspects-5.2.8.RELEASE.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist\sprir
- spring-beans-5.2.8.RELEASE-javadoc.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-d
- spring-beans-5.2.8.RELEASE-sources.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-di
- spring-beans-5.2.8.RELEASE.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist\spring
- spring-context-5.2.8.RELEASE-javadoc.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-
- spring-context-5.2.8.RELEASE-sources.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-
- spring-context-5.2.8.RELEASE.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist\sprir
- spring-context-indexer-5.2.8.RELEASE-javadoc.jar - C:\Users\admin\Downloads\spring-5.2.8.R
- spring-context-indexer-5.2.8.RELEASE-sources.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-dist\spring-framework-5.2.8.RELEASE\libs
- spring-context-indexer-5.2.8.RELEASE.jar - C:\Users\admin\Downloads\spring-5.2.8.RELEASE-
- spring-context-support-5.2.8.RELEASE-javadoc.jar - C:\Users\admin\Downloads\spring-5.2.8.F

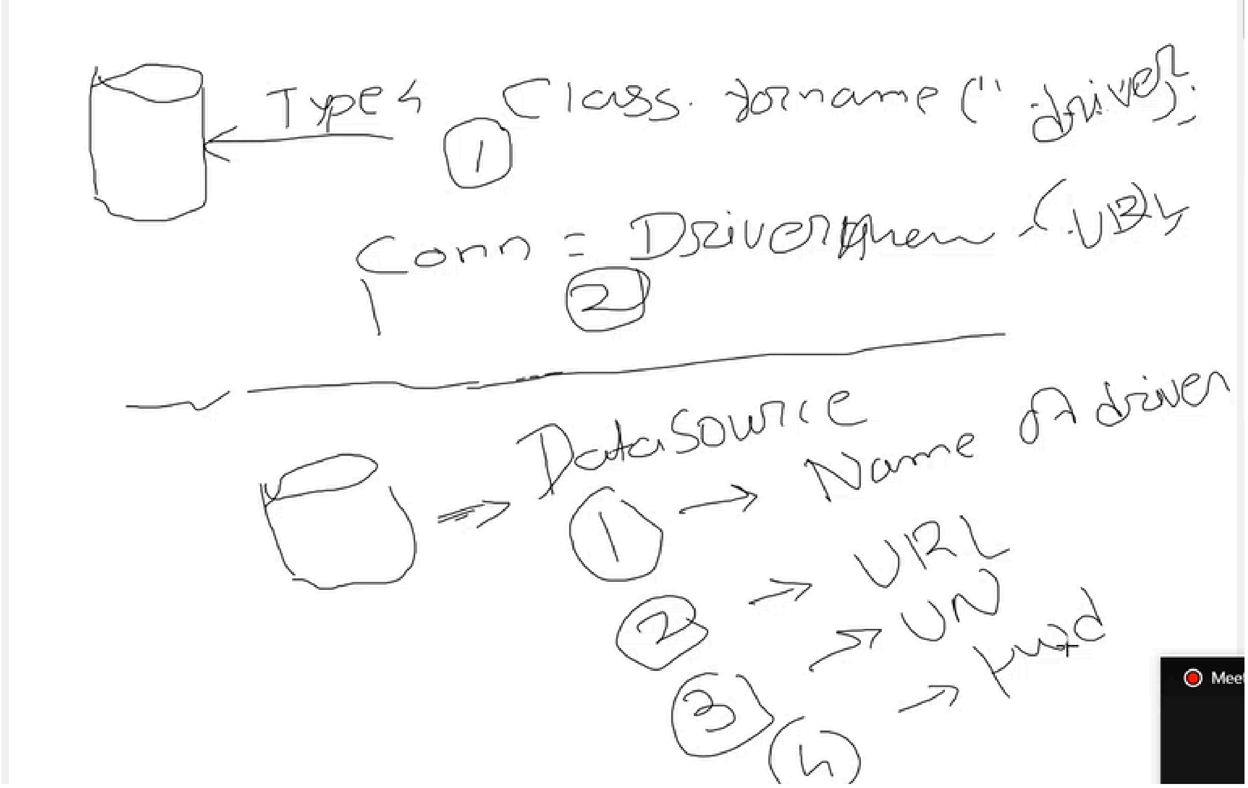
```
            this.name = name;
        }
}
```

```
<terminated> EmpTest (1) [Java Application] C:\Program Files\Java\jdk1.8.0_201\bin\javaw.exe (Mar 27, 2021, 11:06:46 AM – 11:06:48 AM)
1
1jyoti kharade
```


jyoti - springdatabase/src/edu/met/EMployee.java - Eclipse IDE
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Project Explorer
spring-instrument-5.2.8.RELEASE
spring-instrument-5.2.8.RELEASE
spring-instrument-5.2.8.RELEASE
spring-jcl-5.2.8.RELEASE.jar - C:\
spring-jcl-5.2.8.RELEASE-javadoc
spring-jcl-5.2.8.RELEASE-sources
spring-jdbc-5.2.8.RELEASE.jar - C
spring-jdbc-5.2.8.RELEASE-javad
spring-jdbc-5.2.8.RELEASE-sourc
spring-jms-5.2.8.RELEASE.jar - C:
spring-jms-5.2.8.RELEASE-javadc
spring-jms-5.2.8.RELEASE-source
spring-messaging-5.2.8.RELEASE
spring-messaging-5.2.8.RELEASE
spring-messaging-5.2.8.RELEASE
spring-orm-5.2.8.RELEASE.jar - C
spring-orm-5.2.8.RELEASE-javad
spring-orm-5.2.8.RELEASE-sourc
spring-oxm-5.2.8.RELEASE.jar - C
spring-oxm-5.2.8.RELEASE-javad
spring-oxm-5.2.8.RELEASE-sourc
spring-test-5.2.8.RELEASE.jar - C:
spring-test-5.2.8.RELEASE-javadc
spring-test-5.2.8.RELEASE-source
spring-tx-5.2.8.RELEASE.jar - C:\L
spring-tx-5.2.8.RELEASE-javadoc.
spring-tx-5.2.8.RELEASE-sources.
spring-web-5.2.8.RELEASE.jar - C
spring-web-5.2.8.RELEASE-javad
spring-web-5.2.8.RELEASE-sourc
spring-webflux-5.2.8.RELEASE.jar
spring-webflux-5.2.8.RELEASE-ja
spring-webflux-5.2.8.RELEASE-so
spring-webmvc-5.2.8.RELEASE.ja
spring-webmvc-5.2.8.RELEASE-ja
spring-webmvc-5.2.8.RELEASE-s
spring-websocket-5.2.8.RELEASE
spring-websocket-5.2.8.RELEASE
spring-websocket-5.2.8.RELEASE
postgresql-42.2.18.jar - C:\Users\admin\Downloads
bin
target
pom.xml
sss
demolamda.java
set1demo.java
HomeControll...
genericLwil...
EMployee.java
appctx.xml
»93
edu.met
EMploy
roll :
nam
EMpl
EMpl
getR
setR
getN
setN
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Search

```java
        super();
    }
    public EMployee(int roll, String name) {
        super();
        this.roll = roll;
        this.name = name;
    }
    public int getRoll() {
        return roll;
    }
    public void setRoll(int roll) {
        this.roll = roll;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

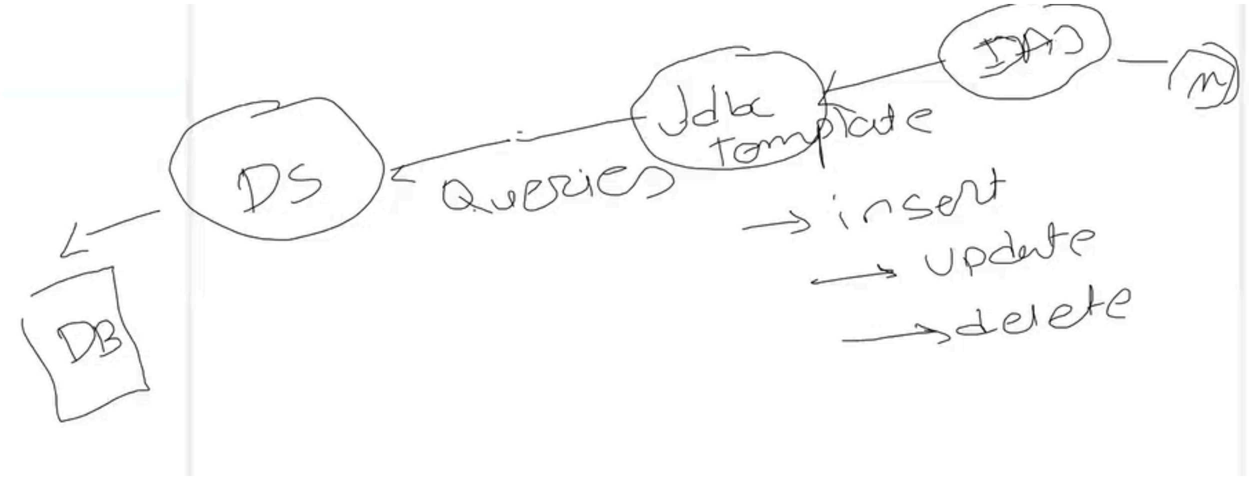
```
<terminated> EmpTest (1) [Java Application] C:\Program Files\Java\jdk1.8.0_201\bin\javaw.exe  (Mar 27, 2021, 11:06:46 AM – 11:06:48 AM)
1
1jyoti kharade
```



Right click on src -> new appctx.xml
File:appctx.xml

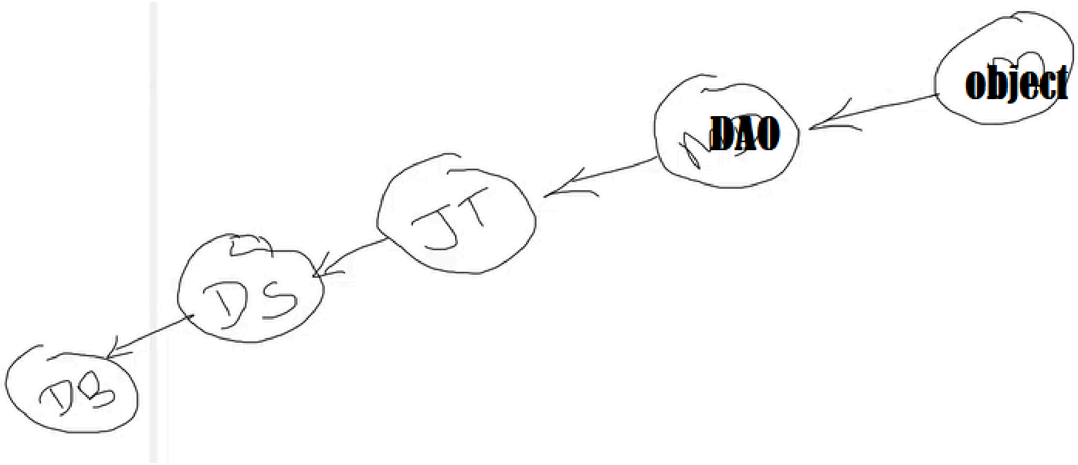
```
<?xml version="1.0" encoding="UTF-8"?>
<beans
xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```
<bean id="ds"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName"
value="org.postgresql.Driver" />
<property name="url"
value="jdbc:postgresql://localhost:5432/postgres" />
<property name="username" value="postgres" />
<property name="password" value="password" />
</bean>

<bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds">
</property>
</bean>
<bean id="Emp1" class="edu.met.EMployeeDao">
<property name="jdbcTemplate"
ref="jdbcTemplate">
</property>
</bean>
</beans>
```



File: EMployeeDao.java

```
import java.util.*;

public class EMployeeDao {
    JdbcTemplate jdbcTemplate;
    public void
setJdbcTemplate(JdbcTemplate jdbcTemplate)
{
        this.jdbcTemplate = jdbcTemplate;
    }
    public int saveEmp(EMployee e){
        String query="insert into emp1
values("+e.getRoll()+",'"+e.getName()+"')"
;
        return jdbcTemplate.update(query);
    }
    public List<EMployee> findAll() {

        String sql = "SELECT * FROM emp1";
        List<EMployee> customers =
jdbcTemplate.query(sql,new
EmpRowMapper());

        return customers;
}
}
```



File: EmpTest.java

```
package edu.met;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import java.util.*;
public class EmpTest
{
private static ApplicationContext appCon;
public static void main(String[] args)
{
appCon = new ClassPathXmlApplicationContext("appctx.xml");
EMployeeDao fac=(EMployeeDao)appCon.getBean("Emp1"); // link
EMployee e1=new EMployee(4,"java jdbc");
System.out.println(fac.saveEmp(e1));
List<EMployee> lstemp=fac.findAll();
for(EMployee e2:lstemp)
{
        System.out.print(e2.getRoll());
        System.out.println(e2.getName());
}
}
}
```

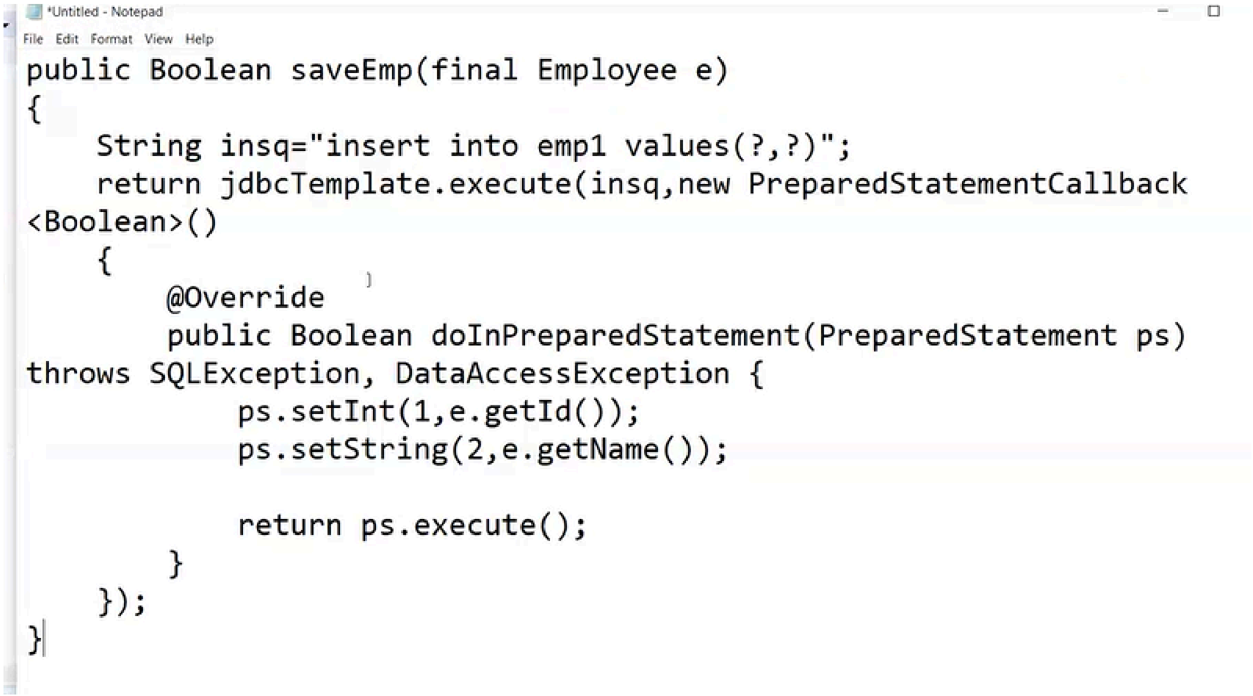
File: EmpRowMapper.java

```
package edu.met;
import org.springframework.jdbc.core.RowMapper;
import java.sql.ResultSet;
import java.sql.SQLException;
public class EmpRowMapper implements RowMapper<EMployee> {

        @Override
        public EMployee mapRow(ResultSet arg0, int arg1) throws SQLException
        {
                EMployee e1=new EMployee();
                e1.setRoll(arg0.getInt(1));
                e1.setName(arg0.getString(2));
                return e1;
        }

}
```

Right click on springdatabase and click on run java application

Output

```
1
1jyoti kharade
2d kharade
3om1
3om1
4java
4java
4java jdbc
5java jdbcprg
```



```java
public Boolean saveEmp(final Employee e)
{
    String insq="insert into emp1 values(?,?)";
    return jdbcTemplate.execute(insq,new PreparedStatementCallback
<Boolean>()
    {
        @Override
        public Boolean doInPreparedStatement(PreparedStatement ps)
throws SQLException, DataAccessException {
            ps.setInt(1,e.getId());
            ps.setString(2,e.getName());

            return ps.execute();
        }
    });
}
```