

Add a database called books, enter:  
mysql> **CREATE DATABASE books;**

Now, database is created. Use a database with use command, type:  
mysql> **USE books;**

Next, create a table called authors with name, email and id as fields:  
mysql> **CREATE TABLE authors (id INT, name VARCHAR(20), email VARCHAR(20));**

To display your tables in books database, enter:  
mysql> **SHOW TABLES;**  
Sample outputs:

+-----------------+

| Tables\_in\_books |

+-----------------+

| authors |

+-----------------+

1 row in set (0.00 sec)

Finally, add a data i.e. row to table books using INSERT statement, run:  
mysql> **INSERT INTO authors (id,name,email) VALUES(1,"Vivek","xuz@abc.com");**  
Sample outputs:

Query OK, 1 row affected (0.00 sec)

Try to add few more rows to your table:  
mysql> **INSERT INTO authors (id,name,email) VALUES(2,"Priya","p@gmail.com");**  
mysql> **INSERT INTO authors (id,name,email) VALUES(3,"Tom","tom@yahoo.com");**

To display all rows i.e. data stored in authors table, enter:  
mysql> **SELECT \* FROM authors;**  
Sample outputs:

+------+-------+---------------+

| id | name | email |

+------+-------+---------------+

| 1 | Vivek | xuz@abc.com |

| 2 | Priya | p@gmail.com |

| 3 | Tom | tom@yahoo.com |

+------+-------+---------------+

3 rows in set (0.00 sec)

STS Code:

Pom.xml



**package com.app.springcore.jdbc\_lessons;**

**public class Author {**

**private String aut\_id;**

**private String aut\_name;**

**private String aut\_email;**

**public String getAut\_id() {**

**return aut\_id;**

**}**

**public void setAut\_id(String aut\_id) {**

**this.aut\_id = aut\_id;**

**}**

**public String getAut\_name() {**

**return aut\_name;**

**}**

**public void setAut\_name(String aut\_name) {**

**this.aut\_name = aut\_name;**

**}**

**public String getAut\_email() {**

**return aut\_email;**

**}**

**public void setAut\_email(String aut\_email) {**

**this.aut\_email = aut\_email;**

**}**

**}**

**<?xml version="1.0" encoding="UTF-8"?>**

**<beans xmlns="http://www.springframework.org/schema/beans"**

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

**xmlns:context="http://www.springframework.org/schema/context"**

**xmlns:p="http://www.springframework.org/schema/p"**

**xmlns:util="http://www.springframework.org/schema/util"**

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

**http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd**

**http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd">**

**<context:annotation-config />**

**<context:component-scan base-package="com.baeldung.applicationcontext" />**

**<bean name="jdc"**

**class="org.springframework.jdbc.datasource.DriverManagerDataSource">**

**<property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />**

**<property name="url"**

**value="jdbc:mysql://localhost:3306/springdb?useSSL=false" />**

**<property name="username" value="root" />**

**<property name="password" value="root" />**

**</bean>**

**<bean name="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">**

**<property name="dataSource">**

**<ref bean="jdc" />**

**</property>**

**</bean>**

**</beans>**

**package com.app.springcore.jdbc\_lessons;**

**import org.springframework.context.support.AbstractApplicationContext;**

**import org.springframework.context.support.ClassPathXmlApplicationContext;**

**import org.springframework.jdbc.core.JdbcTemplate;**

**public class TestJDBC\_Connection {**

**public static void main(String[] args) {**

**AbstractApplicationContext context = new ClassPathXmlApplicationContext("jdbc\_config.xml");**

**JdbcTemplate jt = context.getBean("jdbcTemplate", JdbcTemplate.class);**

**System.*out*.println("JDBC Bean Created !!!");**

**// create an insert query variable**

**String insertQuery = "INSERT INTO authors (id,name,email) VALUES(?,?,?)";**

**// execute or fire the query**

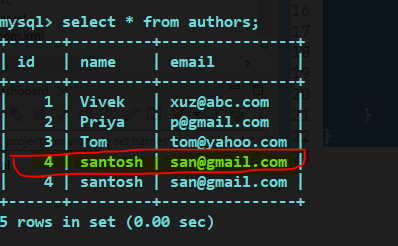
**int no\_of\_records = jt.update(insertQuery, 4, "santosh", "san@gmail.com");**

**System.*out*.println("No of records inserted: " + no\_of\_records);**

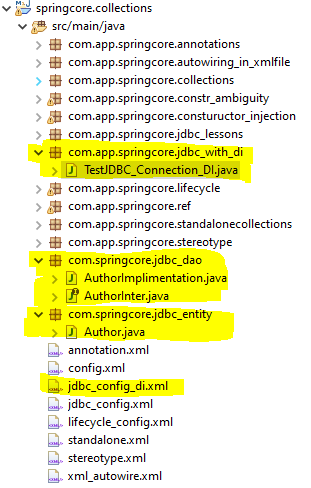
**context.close();**

**}**

**}**



In the above example ‘Dependency Injection’ is not maintained. So following example will do that:



Bean or Entity class:

**package com.springcore.jdbc\_entity;**

**public class Author {**

**private String aut\_id;**

**private String aut\_name;**

**private String aut\_email;**

**public String getAut\_id() {**

**return aut\_id;**

**}**

**public void setAut\_id(String aut\_id) {**

**this.aut\_id = aut\_id;**

**}**

**public String getAut\_name() {**

**return aut\_name;**

**}**

**public void setAut\_name(String aut\_name) {**

**this.aut\_name = aut\_name;**

**}**

**public String getAut\_email() {**

**return aut\_email;**

**}**

**public void setAut\_email(String aut\_email) {**

**this.aut\_email = aut\_email;**

**}**

**}**

Interface for DI:

**package com.springcore.jdbc\_dao;**

**import com.springcore.jdbc\_entity.Author;**

**public interface AuthorInter {**

**public int insert(Author author);**

**}**

DAO class:

**package com.springcore.jdbc\_dao;**

**import org.springframework.jdbc.core.JdbcTemplate;**

**import com.springcore.jdbc\_entity.Author;**

**public class AuthorImplimentation implements AuthorInter {**

**private JdbcTemplate jdbcTemplate;**

**public int insert(Author author) {**

**// create an insert query variable**

**String insertQuery = "INSERT INTO authors (id,name,email) VALUES(?,?,?)";**

**int tot\_records = this.jdbcTemplate.update(insertQuery, author.getAut\_id(), author.getAut\_name(),**

**author.getAut\_email());**

**return tot\_records;**

**}**

**public JdbcTemplate getJdbcTemplate() {**

**return jdbcTemplate;**

**}**

**public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {**

**this.jdbcTemplate = jdbcTemplate;**

**}**

**}**

Config.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"**

**xmlns:context="http://www.springframework.org/schema/context"**

**xmlns:p="http://www.springframework.org/schema/p"**

**xmlns:util="http://www.springframework.org/schema/util"**

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

**http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd**

**http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd">**

**<context:annotation-config />**

**<context:component-scan base-package="com.baeldung.applicationcontext" />**

**<bean name="jdc"**

**class="org.springframework.jdbc.datasource.DriverManagerDataSource">**

**<property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />**

**<property name="url"**

**value="jdbc:mysql://localhost:3306/springdb?useSSL=false" />**

**<property name="username" value="root" />**

**<property name="password" value="root" />**

**</bean>**

**<bean name="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">**

**<property name="dataSource">**

**<ref bean="jdc" />**

**</property>**

**</bean>**

**<bean name="auth\_impl\_dao"**

**class="com.springcore.jdbc\_dao.AuthorImplimentation">**

**<property name="jdbcTemplate" ref="jdbcTemplate" />**

**</bean>**

**</beans>**

Controller Class:

**package com.app.springcore.jdbc\_with\_di;**

**import org.springframework.context.support.AbstractApplicationContext;**

**import org.springframework.context.support.ClassPathXmlApplicationContext;**

**import com.springcore.jdbc\_dao.AuthorImplimentation;**

**import com.springcore.jdbc\_entity.Author;**

**public class TestJDBC\_Connection\_DI {**

**public static void main(String[] args) {**

**AbstractApplicationContext context = new ClassPathXmlApplicationContext("jdbc\_config\_di.xml");**

**AuthorImplimentation ai = context.getBean("auth\_impl\_dao", AuthorImplimentation.class);**

**System.*out*.println("JDBC Bean Created !!!");**

**Author a = new Author();**

**a.setAut\_id("1234");**

**a.setAut\_name("Madhava");**

**a.setAut\_email("madhav@gmail.com");**

**int result = ai.insert(a);**

**System.*out*.println("No of records inserted: " + result);**

**context.close();**

**}**

**}**

