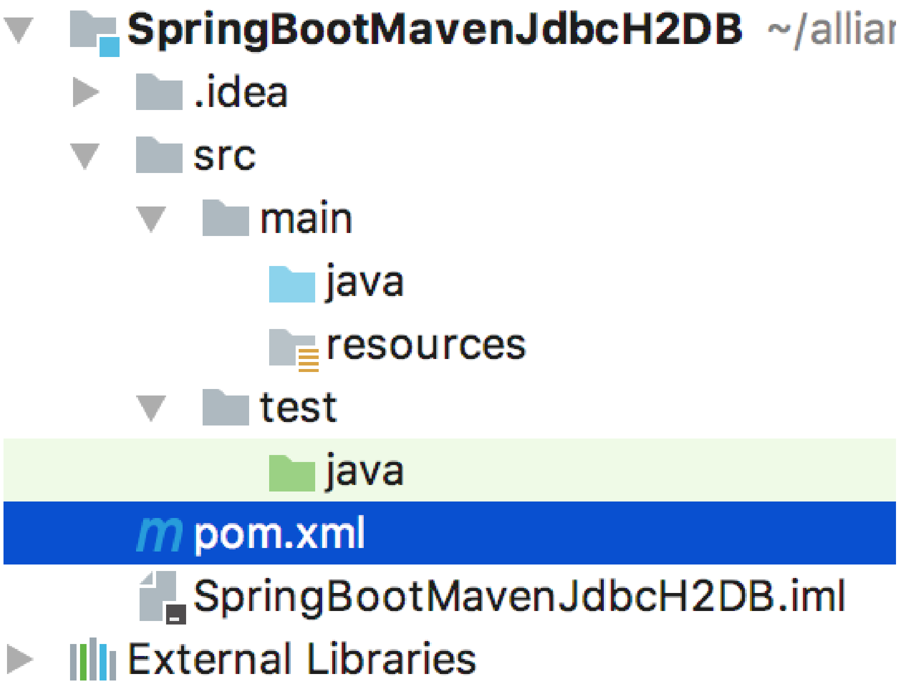
tratto da https://spring.io/guides/gs/relational-data-access/

Utilizzando IntelliJ-IDEA Community Edition,

File > New Project > Maven Project > Skip Archetype

JDK: 1.8
Group Id: com.example.bytecode
Artifact Id: SpringBootMavenJdbcH2DB

Otteniamo la seguente applicazione:



Clicchiamo col destro sulla cartella Java e facciamo un nuovo Package chiamandolo hello.

Da qui ci muoviamo inserendo il POM con le dipendenze di Maven:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example.bytecode</groupId>
    <artifactId>SpringBootMavenJdbcH2DB</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.5.RELEASE</version>
    </parent>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-jdbc</artifactId>
        </dependency>
        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

Poi creiamo due classi Java nel package hello, ovvero Customer.java e Application.java

hello/Customer.java

```
package hello;

public class Customer {

    private long id;
    private String firstName, lastName;

    public Customer(long id, String firstName, String lastName) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    @Override
    public String toString() {
        return String.format(
                "Customer[id=%d, firstName='%s', lastName='%s']",
                id, firstName, lastName);
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

hello/Application.java

```
package hello;
```

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.jdbc.core.JdbcTemplate;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

@SpringBootApplication
public class Application implements CommandLineRunner {

    private static final Logger log = LoggerFactory.getLogger(Application.class);

    public static void main(String args[]) {
        SpringApplication.run(Application.class, args);
    }

    @Autowired
    JdbcTemplate jdbcTemplate;

    @Override
    public void run(String... strings) throws Exception {

        log.info("Creating tables");

        jdbcTemplate.execute("DROP TABLE customers IF EXISTS");
        jdbcTemplate.execute("CREATE TABLE customers(" +
                "id SERIAL, first_name VARCHAR(255), last_name VARCHAR(255))");

        // Split up the array of whole names into an array of first/last names
        List<Object[]> splitUpNames = Arrays.asList("John Woo", "Jeff Dean", "Josh Bloch", "Josh Long").stream()
                .map(name -> name.split(" "))
                .collect(Collectors.toList());

        // Use a Java 8 stream to print out each tuple of the list
        splitUpNames.forEach(name -> log.info(String.format("Inserting customer record for %s %s", name[0], name[1])));

        // Uses JdbcTemplate's batchUpdate operation to bulk load data
        jdbcTemplate.batchUpdate("INSERT INTO customers(first_name, last_name) VALUES (?,?)", splitUpNames);

        log.info("Querying for customer records where first_name = 'Josh':");
        jdbcTemplate.query(
                "SELECT id, first_name, last_name FROM customers WHERE first_name = ?", new Object[] { "Josh" },
                (rs, rowNum) -> new Customer(rs.getLong("id"), rs.getString("first_name"), rs.getString("last_name"))
        ).forEach(customer -> log.info(customer.toString()));
    }
}
```

Procediamo a verificare il build di maven che fa anche il run dell'applicazione con il server incorporato.
Andiamo nel Terminal in basso a sinistra nell'IDE di IntelliJ e scriviamo:

```
mvn spring-boot:run
```

il terminal avvierà Maven che farà la build dell'applicazione e il run di Tomcat integrato e dell'applicazione, ottenendo il seguente output:

```
2017-08-02 17:00:17.700 INFO 11755 --- [ main] hello.Application : Starting Application on Alessandros-MacBook-
Pro.local with PID 11755 (/Users/alessandroargentieri/allianz/SpringBootMavenJdbcH2DB/target/classes started by
alessandroargentieri in /Users/alessandroargentieri/allianz/SpringBootMavenJdbcH2DB)
2017-08-02 17:00:17.702 INFO 11755 --- [ main] hello.Application : No active profile set, falling back to default
profiles: default
2017-08-02 17:00:17.754 INFO 11755 --- [ main] s.c.a.AnnotationConfigApplicationContext : Refreshing
org.springframework.context.annotation.AnnotationConfigApplicationContext@5e9bb084: startup date [Wed Aug 02 17:00:17
CEST 2017]; root of context hierarchy
2017-08-02 17:00:18.451 INFO 11755 --- [ main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure
on startup
```

```
2017-08-02 17:00:18.460 INFO 11755 --- [ main] hello.Application : Creating tables
2017-08-02 17:00:18.763 INFO 11755 --- [ main] hello.Application : Inserting customer record for John Woo
2017-08-02 17:00:18.763 INFO 11755 --- [ main] hello.Application : Inserting customer record for Jeff Dean
2017-08-02 17:00:18.763 INFO 11755 --- [ main] hello.Application : Inserting customer record for Josh Bloch
2017-08-02 17:00:18.763 INFO 11755 --- [ main] hello.Application : Inserting customer record for Josh Long
2017-08-02 17:00:18.795 INFO 11755 --- [ main] hello.Application : Querying for customer records where first_name =
'Josh':
2017-08-02 17:00:18.800 INFO 11755 --- [ main] hello.Application : Customer[id=3, firstName='Josh', lastName='Bloch']
2017-08-02 17:00:18.800 INFO 11755 --- [ main] hello.Application : Customer[id=4, firstName='Josh', lastName='Long']
2017-08-02 17:00:18.802 INFO 11755 --- [ main] hello.Application : Started Application in 1.551 seconds (JVM running
for 4.353)
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.331 s
[INFO] Finished at: 2017-08-02T17:00:18+02:00
[INFO] Final Memory: 31M/399M
[INFO] ------------------------------------------------------------------------
```