

# Transactional Information Systems

## Practical Exercise 1 (deadline Nov 7. – 18:00)

“Knock, knock.”

“Who’s there?”

very long pause....

“Java.”

# Topics

## JDBC Programming

- Connect to H2 database
- Access the databases meta information
- Run SQL queries with JDBC

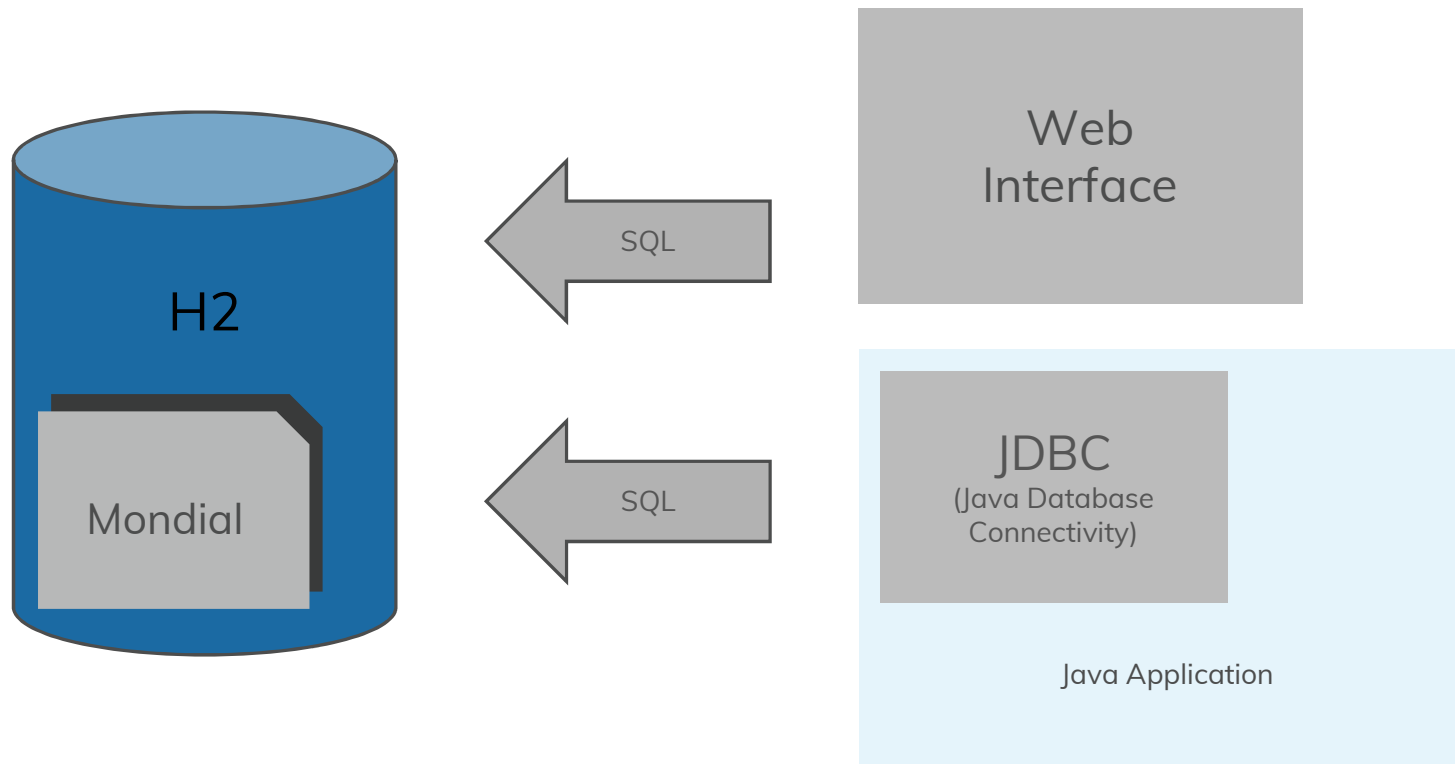
## SQL

- Create / alter tables
- Insert, update, select statements

## H2 database

- Create triggers

# Overview



# What do we need?

## *H2 + templates*

- Download TIS-Prac1.zip from TIS exercise web page

## *Java and IDE (integrated development environment)*

- Java should be installed for most operating systems, otherwise: Please install Java (no java 9).
- Programming environment:
  - Without IDE: just use editor and javac (please use **java 8** or earlier!)
  - With IDE: I recommend Eclipse, available for all operating systems

# Programming Java without IDE



Dresden Database  
Systems Group

```
> cd sXXXXXXX
```

```
> ls
```

Main.java

NewPopulationTrigger.java

```
> javac NewPopulationTrigger.java -cp ../lib/h2.jar
```

```
> javac Main.java -cp ../lib/h2.jar
```

```
> ls
```

Main.class

Main.java

NewPopulationTrigger.class

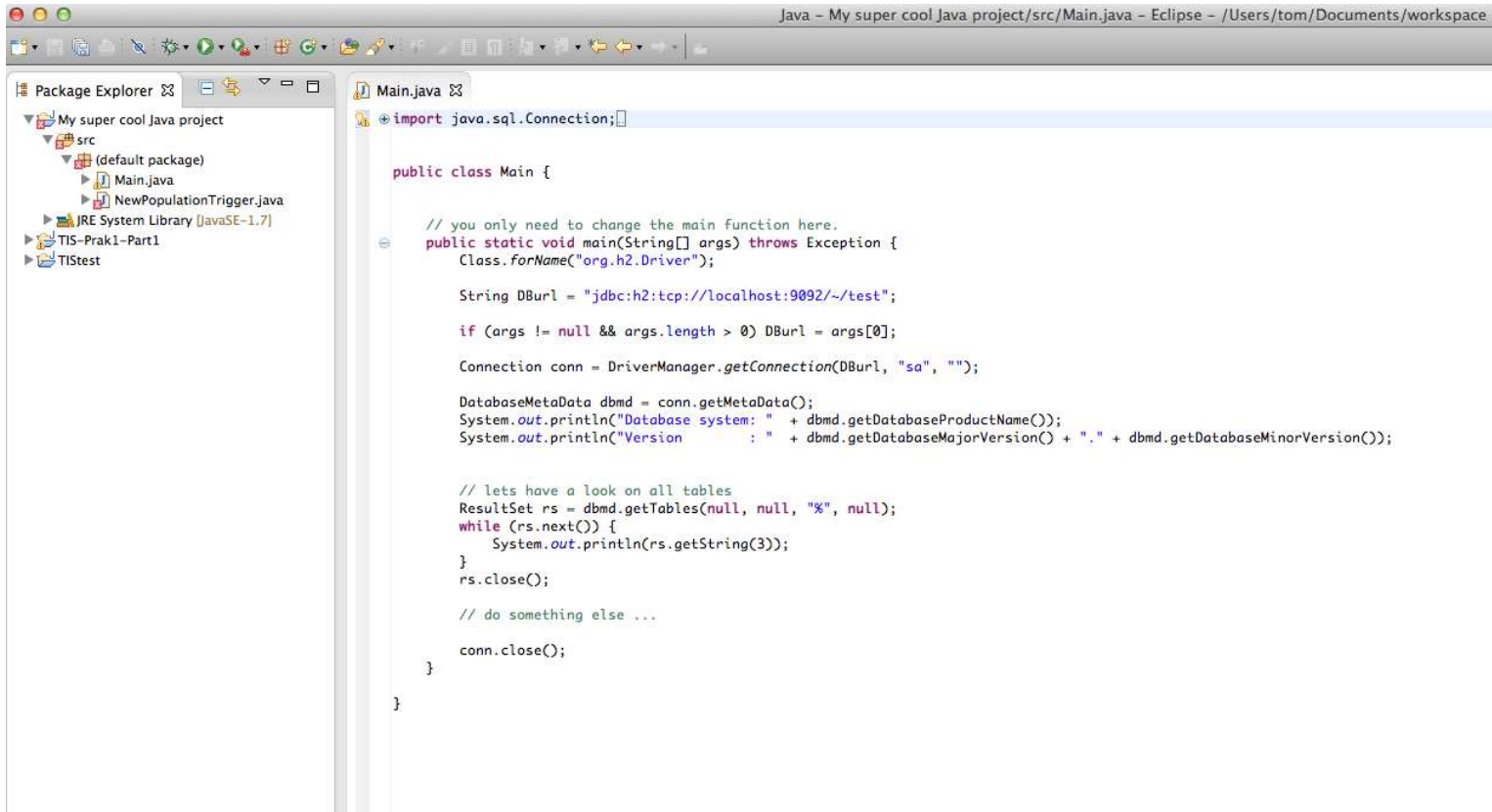
NewPopulationTrigger.java

```
> java -cp ../lib/h2.jar Main "jdbc:h2:tcp://localhost:9092/~test"
```

-cp means class path  
(where are the Java  
classes)

That means 2 class paths: . and ../lib/h2.jar  
(Separator in Linux/Max ":", on Windows ";")

# Programming Java with IDE: Eclipse



```
Java - My super cool Java project/src/Main.java - Eclipse - /Users/tom/Documents/workspace

Package Explorer
My super cool Java project
  src
    (default package)
      Main.java
      NewPopulationTrigger.java
  JRE System Library [JavaSE-1.7]
  TIS-Prak1-Part1
  TISTest

Main.java
import java.sql.Connection;

public class Main {

    // you only need to change the main function here.
    public static void main(String[] args) throws Exception {
        Class.forName("org.h2.Driver");

        String DBUrl = "jdbc:h2:tcp://localhost:9092/~test";

        if (args != null && args.length > 0) DBUrl = args[0];

        Connection conn = DriverManager.getConnection(DBUrl, "sa", "");

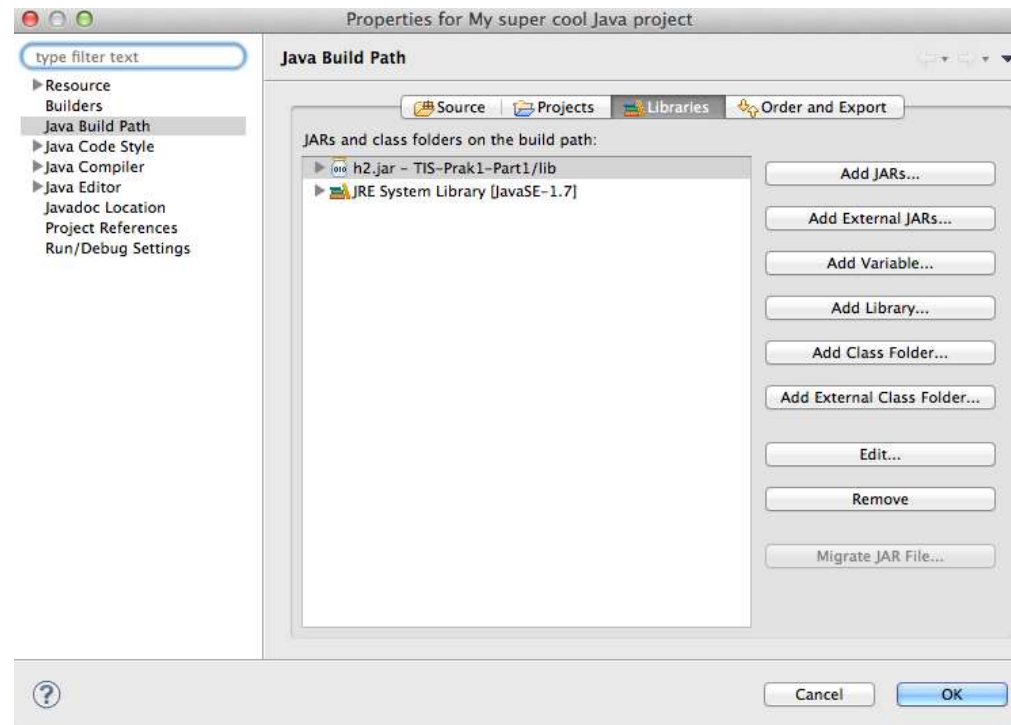
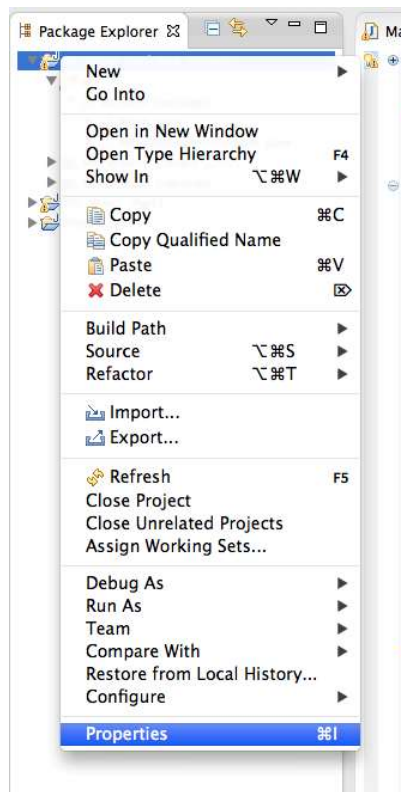
        DatabaseMetaData dbmd = conn.getMetaData();
        System.out.println("Database system: " + dbmd.getDatabaseProductName());
        System.out.println("Version      : " + dbmd.getDatabaseMajorVersion() + "." + dbmd.getDatabaseMinorVersion());

        // lets have a look on all tables
        ResultSet rs = dbmd.getTables(null, null, "%", null);
        while (rs.next()) {
            System.out.println(rs.getString(3));
        }
        rs.close();

        // do something else ...

        conn.close();
    }
}
```

# Programming Java with IDE: Eclipse



## Part I - OK, let's get started

*Start H2:* `java -cp h2.jar org.h2.tools.Server`

- You can login to the web interface with “sa” and “”
- You can even change the language (that's luxury, isn't it)

*Load Mondial:*

- `java -cp h2.jar -jar preFill.jar "jdbc:h2:tcp://localhost:9092/~ /test"`
- This time, the data is not loaded automatically, why is this:
  - The database saves you data now (usually in your home directory)
  - Delete all test.\* files to reset your database

Sometimes this  
URL does not  
work: try  
127.0.0.1 or your  
real IP



## Part II - Write a JDBC program

*Use main.java as template, adjust „Main“ function*

1. Create Table NEWPOPULATION
  - First column: COUNTRY\_CODE (varchar(4))
  - Second column: AMOUNT (int)
2. Add column CURRENT\_POPULATION (int) to country
3. Insert 100 tuples into NEWPOPULATION
  - COUNTRY\_CODE = 'NZ', AMOUNT = 5;
4. Update COUNTRY.CURRENT\_POPULATION with the new values
  - For each country:  $CURRENT\_POPULATION = POPULATION + SUM(NEWPOPULATION)$

## Part II - Write a JDBC program

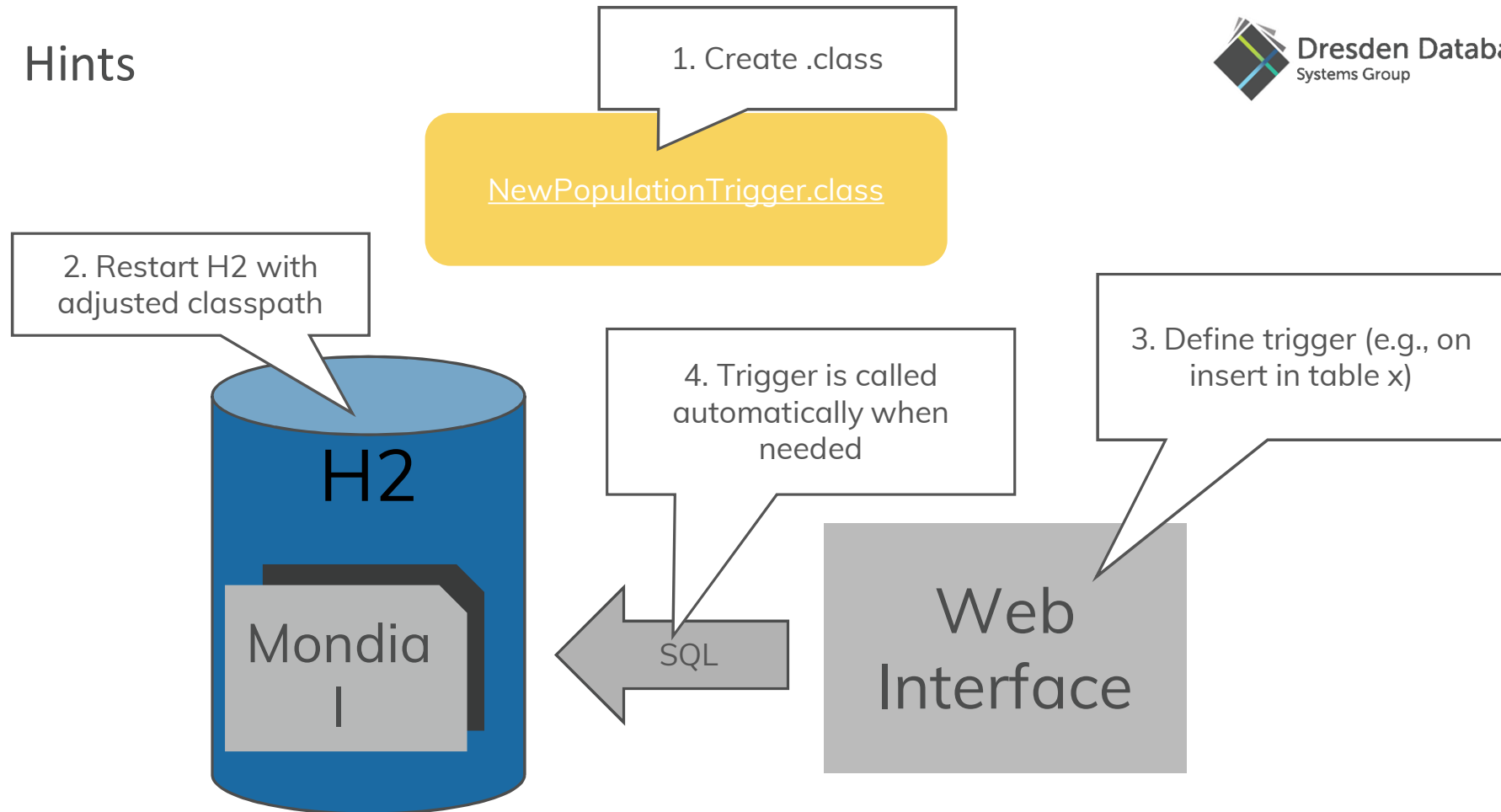
### HINTS:

- You can test everything in the web interface, before you write the JDBC program.
- In the java program, always check if the table or column exists before creating or altering.
- Look in the lecture script, the H2 website, or online tutorials for SQL and JDBC for help.
- H2 has builtin functions like IFNULL(a,b), see H2 website.

## Part III - Write a H2 Trigger

- Use *NewPopulationTrigger.java*, adjust the fire function
- Implement a trigger that updates *CURRENT\_POPULATION* whenever something is inserted in *NEWPOPULATION*.
  - Put your code in the fire function
  - Compile .java to .class
  - Restart Server with classpath adjustment:  
java -cp ../libs/h2.jar:bin/ org.h2.tools.Server
  - Register trigger in H2
    - CREATE TRIGGER MYTRIGGER AFTER INSERT ON NEWPOPULATION FOR EACH ROW CALL "NewPopulationTrigger"
  - Test your trigger with INSERT queries from the web interface

## Hints



# What/when to Submit

- Please submit your solution until Nov 7th. – 18:00 !
- Only Submit
  - sXXXXXXXX
    - Main.java
    - NewPopulationTrigger.java
- Change sXXXXXXXX to your login number and zip this folder
- Send the zipped folder to: [mikhail.zarubin@tu-dresden.de](mailto:mikhail.zarubin@tu-dresden.de)
- Don't change any file names, class names, or function names.