Prof. DI Dr. Erich Gams

Datenbankzugriffe mit Java

JDBC

informationssysteme htl-wels

Übersicht ☛ Was lernen wir?

- Was ist JDBC?
- JDBC Architekturen
- Grundgerüst eines Datenbankzugriffs
- Metadaten auslesen
- SQL Abfragen
- Beispiele

Codezeilenquizz

- Age of Empries: 1,1Mio Codezeilen
- Hubble Space Teleskop: 2 Mio
- Boeing 787: 13 Mio
- Windows 7: 39 Mio
- Photoshop (Version C.S.6): 4 Mio
- Modernes Auto (GPS, Sensoren, Steuerungszentrale): 100 Mio
- Facebook (inkl Backend): 62 Mio
- Google Dienste (YouTube, bis zu Android): 2 Mrd.
- Menschliches Gehirn: 3300 Mrd.


I THINK WE SHOULD BUILD AN SQL DATABASE.
UH-OH
S. Adams
E-mail: SCOTTADAMS@AOL.COM
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
© 1995 United Feature Syndicate, Inc.(NYC)
WHAT COLOR DO YOU WANT THAT DATABASE?
I THINK MAUVE HAS THE MOST RAM.

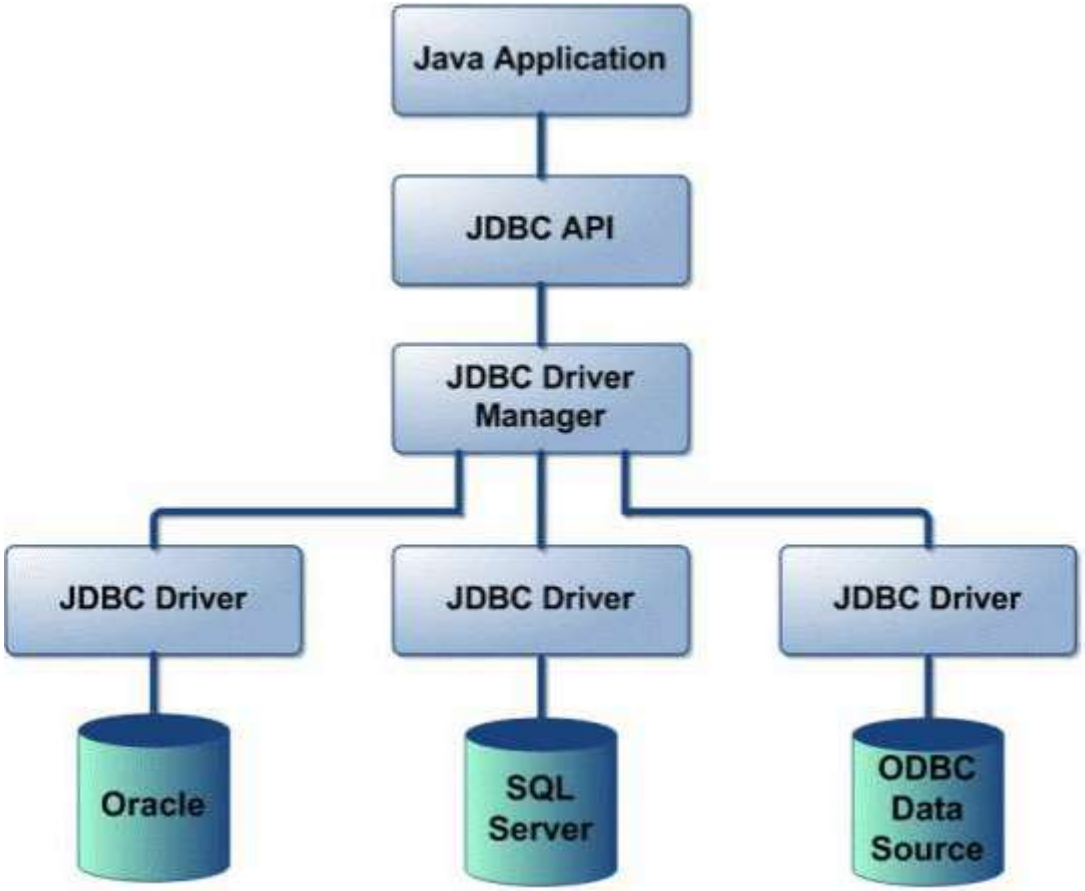
Was ist JDBC?

- Mit JDBC hat Sun Microsystems einen Standard definiert, um aus Java-Programmen heraus auf relationale Datenbanksysteme zugreifen zu können.
- JDBC steht für Java Database Connectivity.
- JDBC stellt ein API (Application Programming Interface) zur Verfügung

Was ist JDBC?

- Der Datenbankzugriff erfolgt unabhängig vom verwendeten DBMS.
- Der Datenbank-Client braucht (und sollte) nicht mit einem bestimmten DBMS "im Hinterkopf" entwickelt werden.
- Alle DBMS-spezifischen Details übernimmt ein JDBC-Treiber. Er ist ebenfalls in Java geschrieben und wird von den Datenbankherstellern oder von Dritten angeboten.
- Ein Treiber wird gebraucht.

JDBC Architecture



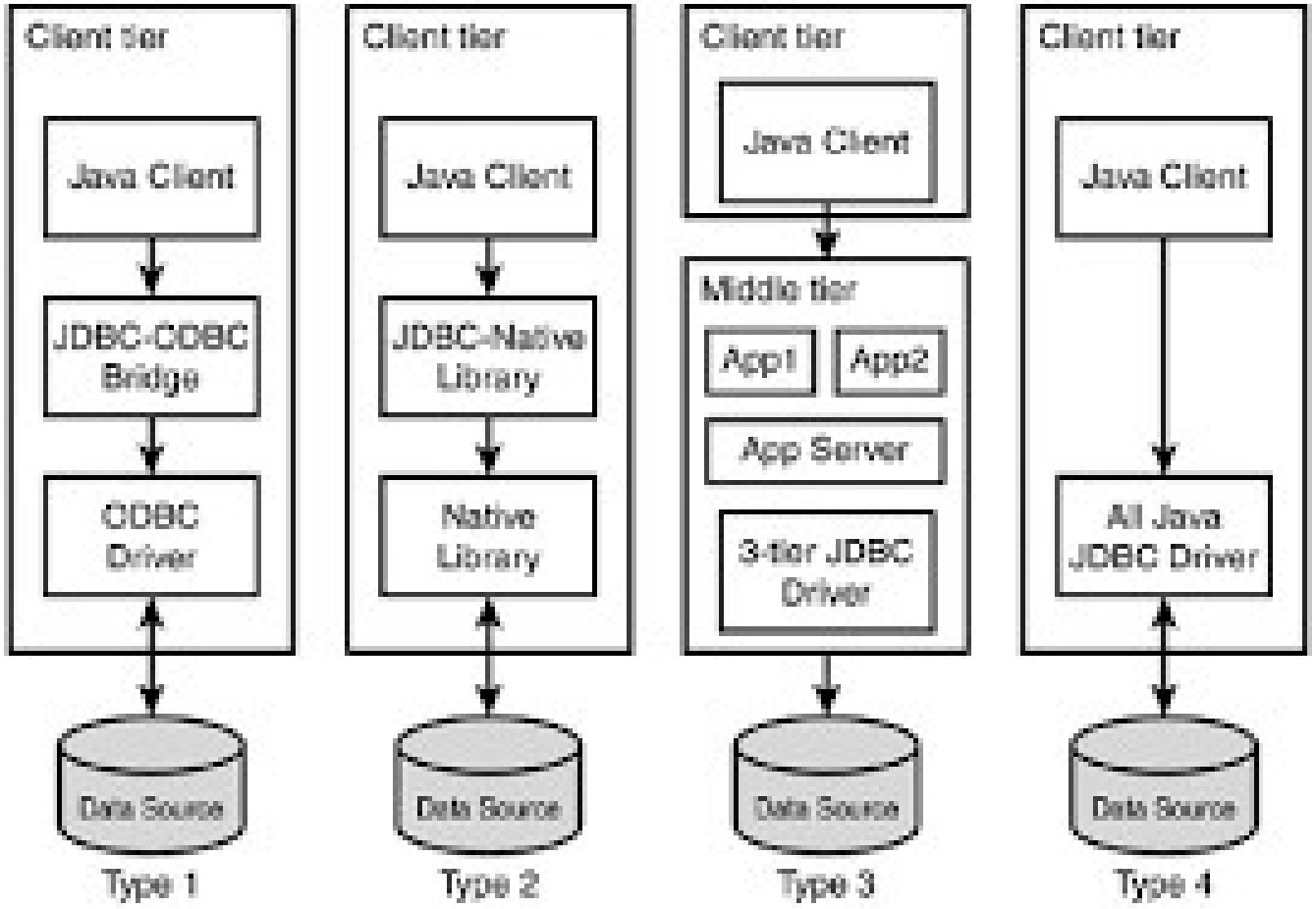
JDBC driver types

- JDBC type 1 driver
  - Is a JDBC-ODBC Bridge which converts JDBC methods into ODBC function calls.
  - ODBC must be installed on the computer and the database must support ODBC driver.
  - a JDBC bridge is used to access ODBC drivers installed on each client machine
- JDBC Native-API Partly-Java driver
  - The JDBC type 2 driver uses the libraries of the database which have to be available at the client side.
  - The driver converts the JDBC method calls into native calls of the database.

JDBC driver types

- JDBC-Net Pure-Java Driver
  - The JDBC-Net Pure-Java driver consists of client and server portions.
  - The client portion contains pure Java functions and the server portion contains Java and native methods
- Native-Protocol Pure-Java Driver
  - It is a Java driver that interacts with the database directly using a vendor-specific network protocol.
  - As opposed to the other JDBC driver you do not require to install any vendor-specific libraries to use the Type 4 driver.

JDBC driver types



Steps to connect?

- Download and install the JDBC Driver
- Define the connection URL.
- Established the connection.
- Create the Statement object.
- Execute a query.
- Process the results.
- Close the connection

Steps to connect?

- Define the connection url :

Class.forName();

- For MySQL driver:

Class.forName("com.mysql.jdbc.Driver");

- You don't need that, if you place the driver in CLASSPATH

Steps to connect?

- Established the connection:

```
Connection con =DriverManager.getConnection("url","user_name","pass");
```

- Create the Statement object:

```
Statement stmt=con.createStatement();
```

Execute a query

- For the SELECT query:

```
String sql="SELECT * FROM EMP";
stmt.executeQuery(sql);
```

- For the INSERT/UPDATE query:

```
String sql="INSERT INTO EMP VALUES(47,'TEDDY')";
stmt.executeUpdate(sql);
```

Execute a query

- Process the result:

```
ResultSet rs=stmt.executeQuery(sql);
while(rs.next()){
  System.out.println(rs.getInt(id));
  System.out.print(rs.getString(name));
}
```

Overview Methods

- **execute**
  - Returns true if the first object that the query returns is a ResultSet object. Use this method if the query could return one or more ResultSet objects. Retrieve the ResultSet objects returned from the query by repeatedly calling Statement.getResultSet.
- **executeQuery**
  - Returns one ResultSet object. (SELECT query)
- **executeUpdate**
  - Returns an integer representing the number of rows affected by the SQL statement. Use this method if you are using INSERT, DELETE, or UPDATE SQL statements.

Overview Methods

executeQuery()	executeUpdate()	execute()
This method is used to execute the SQL statements which retrieve some data from the database.	This method is used to execute the SQL statements which update or modify the database.	This method can be used for any kind of SQL statements.
This method returns a ResultSet object which contains the results returned by the query.	This method returns an int value which represents the number of rows affected by the query. This value will be the 0 for the statements which return nothing.	This method returns a boolean value. TRUE indicates that query returned a ResultSet object and FALSE indicates that query returned an int value or returned nothing.
This method is used to execute only select queries.	This method is used to execute only non-select queries.	This method can be used for both select and non-select queries.
Ex : SELECT	Ex : DML → INSERT, UPDATE and DELETE DDL → CREATE, ALTER	This method can be used for any type of SQL statements.

Steps to connect?

- Close the connection release all the resources that the connection is holding.

```
rs.close();
stmt.close();
con.close();
```

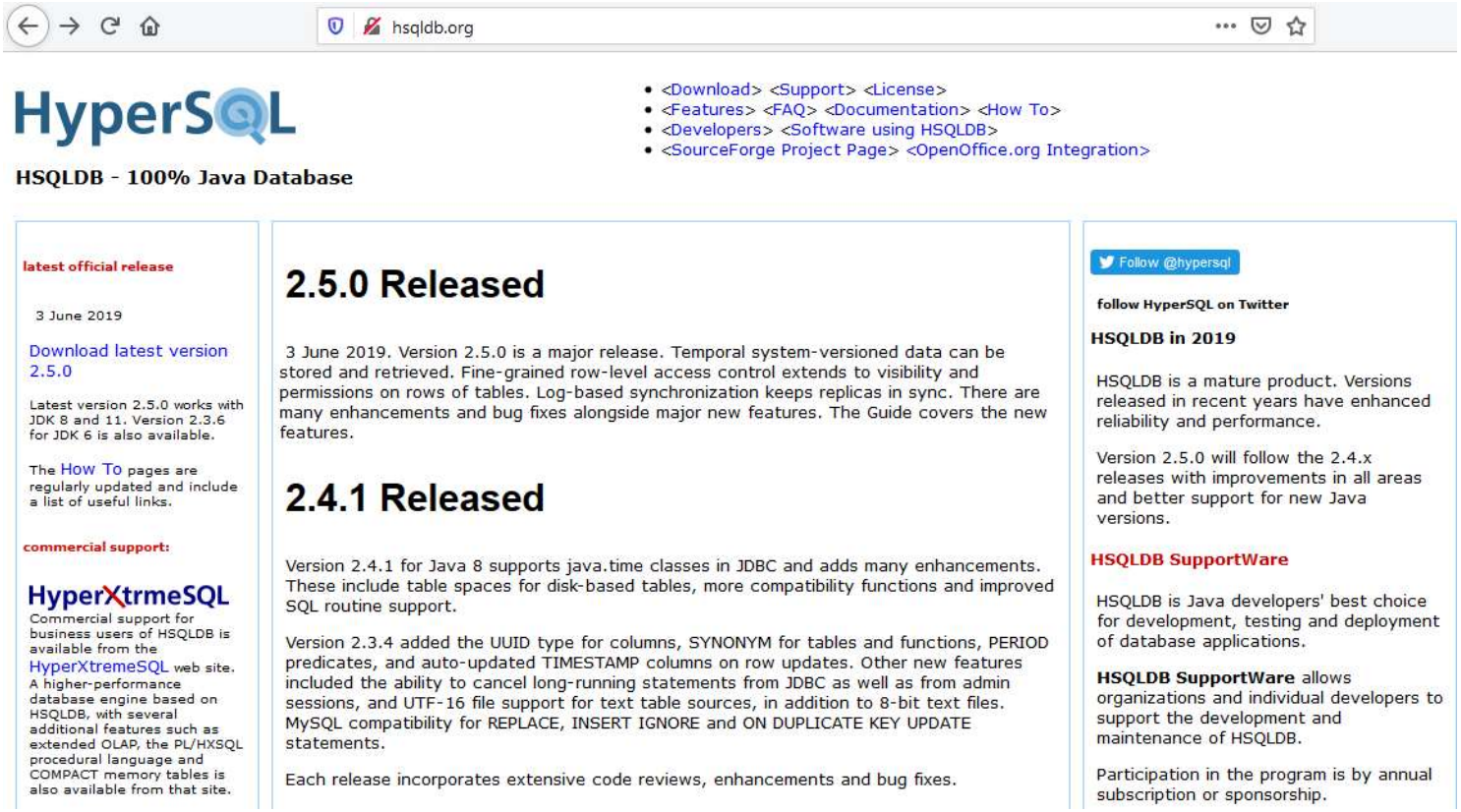
Outsource your connection data!

```
try (
      FileInputStream in = new FileInputStream("dbconnect.properties");
)
{
      Properties prop = new Properties();

      // Properties laden
      prop.load(in);

      String driver = prop.getProperty("driver");
      String url = prop.getProperty("url");
      String user = prop.getProperty("user");
      String pwd = prop.getProperty("pwd");
}
catch (IOException e) {
      e.printStackTrace();
}
```

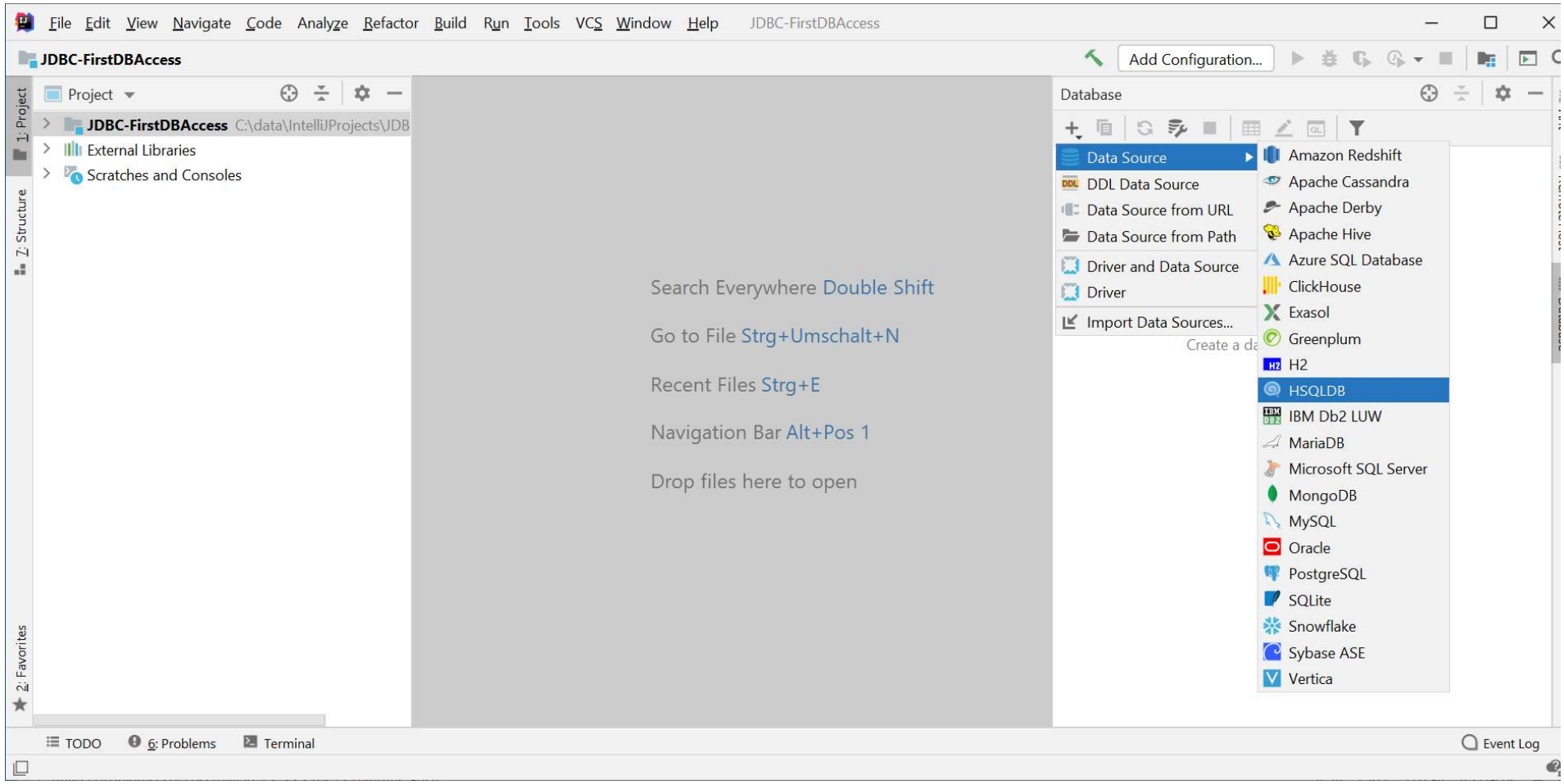
HSQLDB

- Java-RDBMS
- Eingebettetes DBS (Dateiorientiert)
- IntelliJ und HSQLDB
  - https://www.jetbrains.com/help/idea/configuring-database-connections.html#add-a-user-driver-to-an-existing

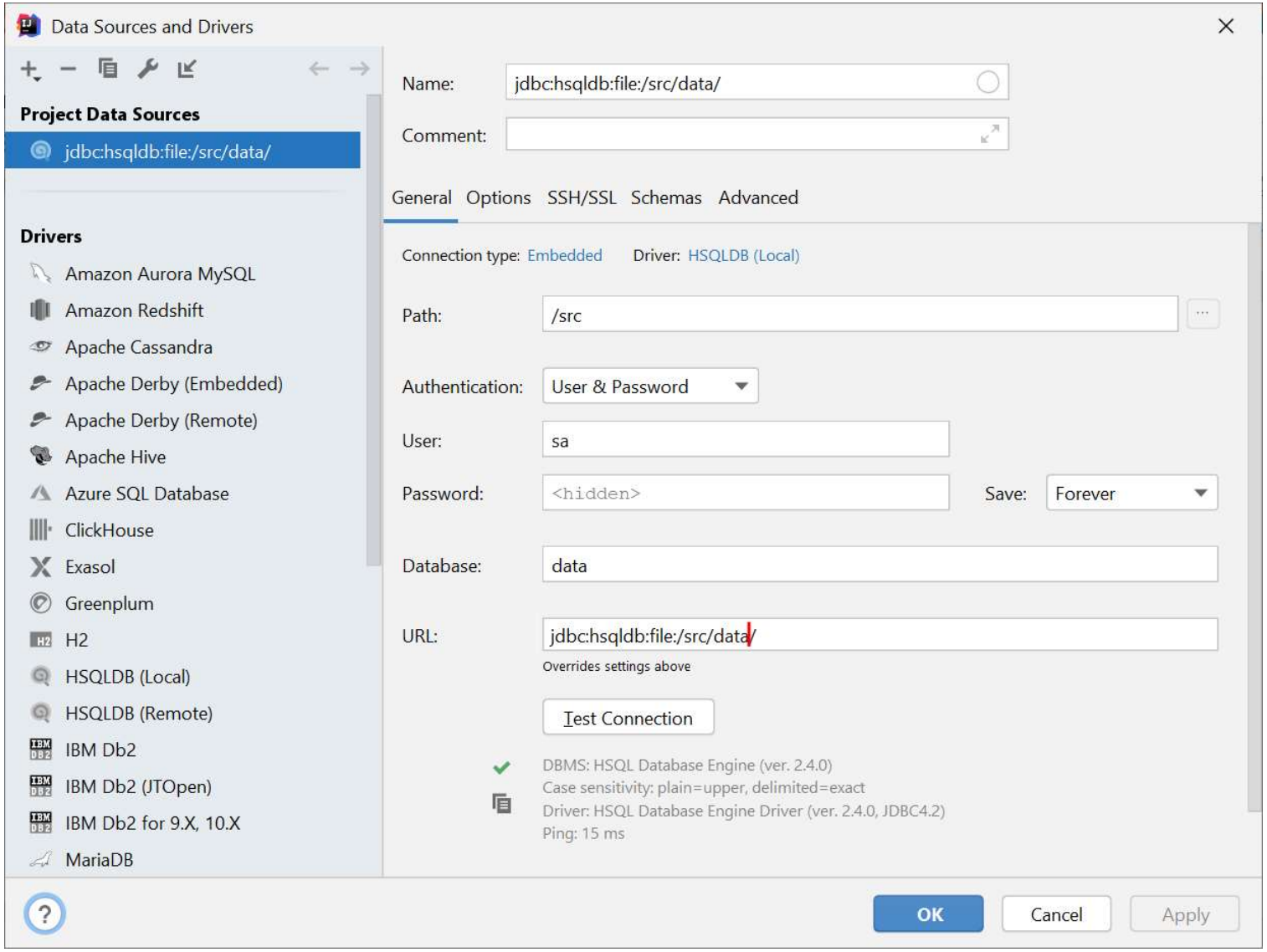
HSQLDB


hsqldb.org
HyperSQL
HSQLDB - 100% Java Database
<Download> <Support> <License>
<Features> <FAQ> <Documentation> <How To>
<Developers> <Software using HSQLDB>
<SourceForge Project Page> <OpenOffice.org Integration>
latest official release
3 June 2019
Download latest version 2.5.0
Latest version 2.5.0 works with JDK 8 and 11. Version 2.3.6 for JDK 6 is also available.
The How To pages are regularly updated and include a list of useful links.
commercial support:
HyperXtremeSQL
Commercial support for business users of HSQLDB is available from the HyperXtremeSQL web site. A higher-performance database engine based on HSQLDB, with several additional features such as extended OLAP, the PL/HXSQL procedural language and COMPACT memory tables is also available from that site.
2.5.0 Released
3 June 2019. Version 2.5.0 is a major release. Temporal system-versioned data can be stored and retrieved. Fine-grained row-level access control extends to visibility and permissions on rows of tables. Log-based synchronization keeps replicas in sync. There are many enhancements and bug fixes alongside major new features. The Guide covers the new features.
2.4.1 Released
Version 2.4.1 for Java 8 supports java.time classes in JDBC and adds many enhancements. These include table spaces for disk-based tables, more compatibility functions and improved SQL routine support.
Version 2.3.4 added the UUID type for columns, SYNONYM for tables and functions, PERIOD predicates, and auto-updated TIMESTAMP columns on row updates. Other new features included the ability to cancel long-running statements from JDBC as well as from admin sessions, and UTF-16 file support for text table sources, in addition to 8-bit text files. MySQL compatibility for REPLACE, INSERT IGNORE and ON DUPLICATE KEY UPDATE statements.
Each release incorporates extensive code reviews, enhancements and bug fixes.
Follow @hypersql
follow HyperSQL on Twitter
HSQLDB in 2019
HSQLDB is a mature product. Versions released in recent years have enhanced reliability and performance.
Version 2.5.0 will follow the 2.4.x releases with improvements in all areas and better support for new Java versions.
HSQLDB SupportWare
HSQLDB is Java developers' best choice for development, testing and deployment of database applications.
HSQLDB SupportWare allows organizations and individual developers to support the development and maintenance of HSQLDB.
Participation in the program is by annual subscription or sponsorship.

IntelliJ and Databases

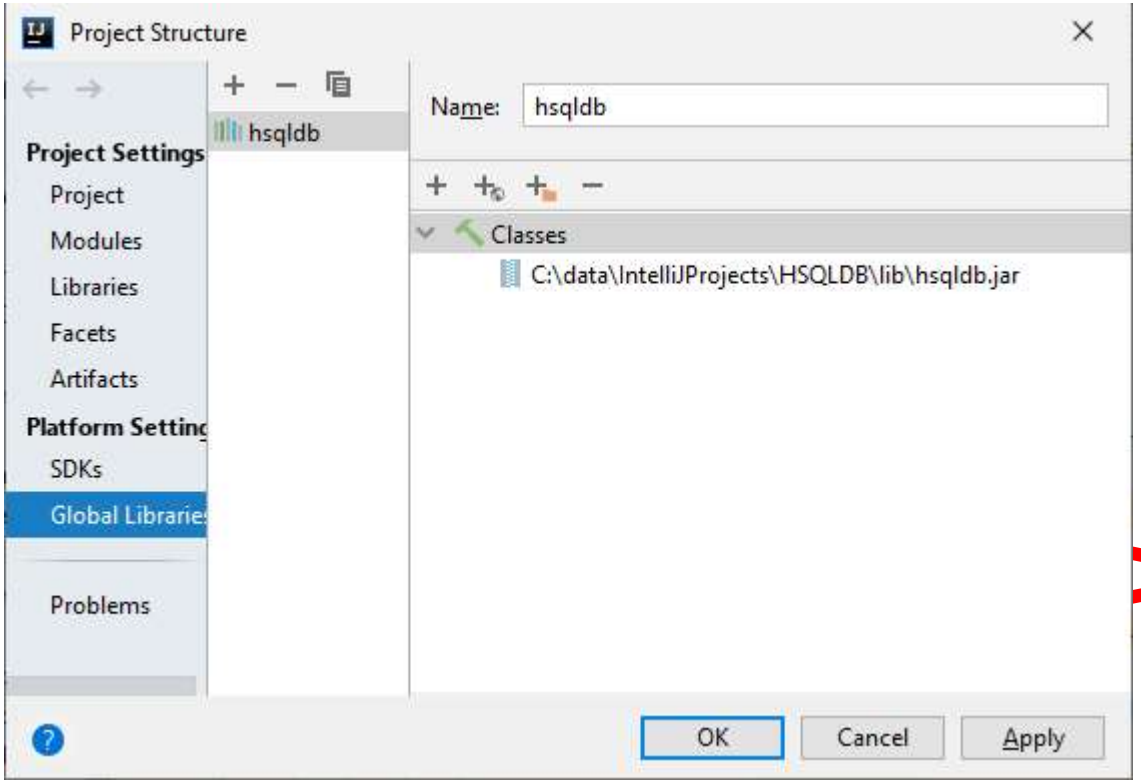


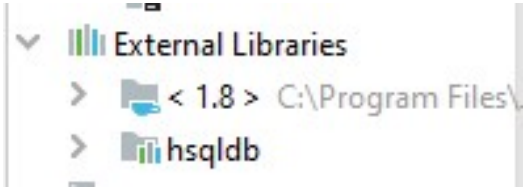
IntelliJ und HSQLDB



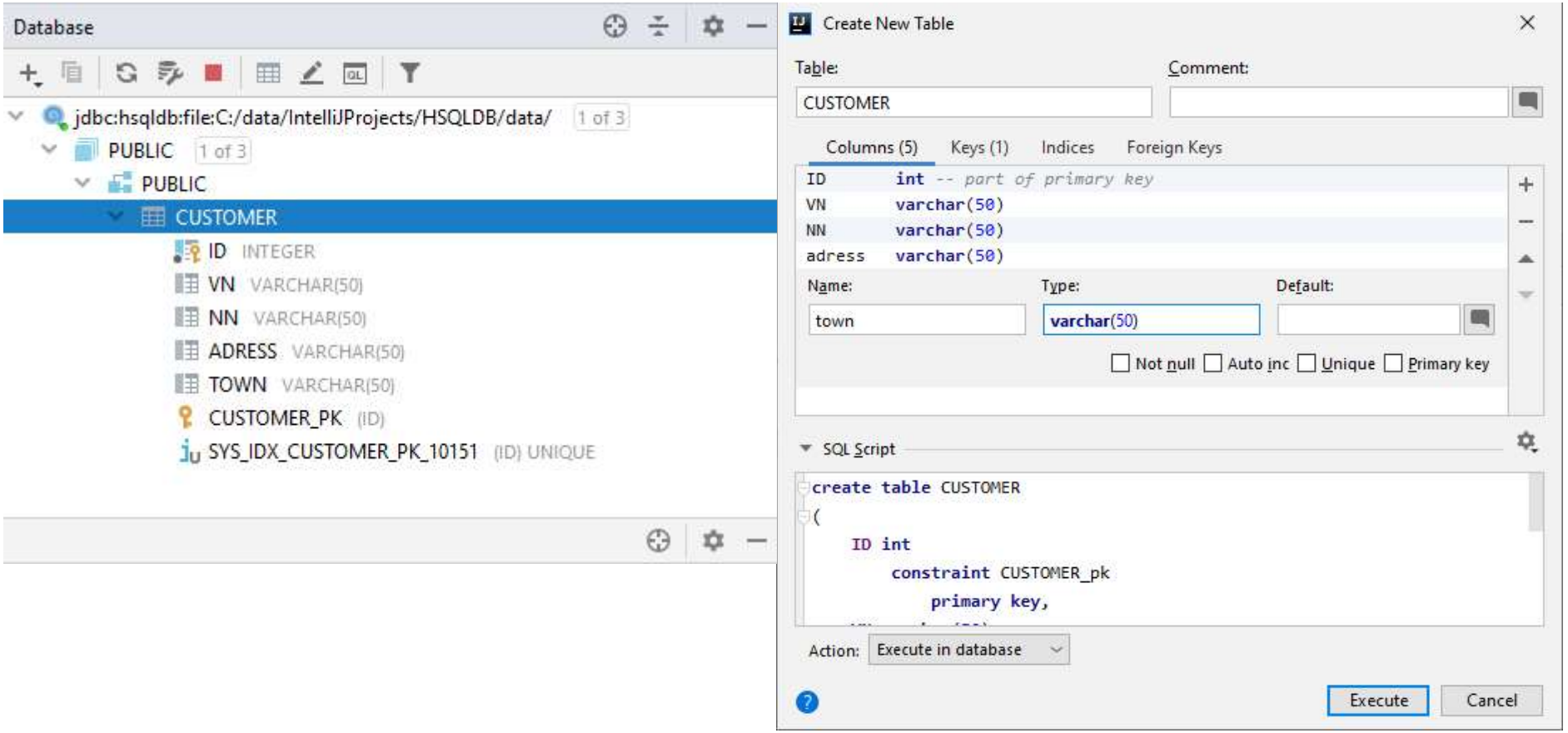
IntelliJ und HSQLDB

- Treiber hinzufügen!





View -> Tool Windows -> Database



Client für HSQLDB

- Beispiel FirstSqlAccess

Struktur mit „try with ressources“

```
try (Connection con=DriverManager.getConnection("jdbc:hsqldb:file:\\src\\data\\;shutdown=true
hsqldb.lock_file=false","sa","");
Statement stmt=con.createStatement();) {

   ResultSet rs=stmt.executeQuery("SELECT * FROM \"Persons\";");
   while (rs.next()) {
      System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
   }
   rs.close();
} catch (SQLException throwables) {
   throwables.printStackTrace();
}
```

Java und SQL Datentypen

Java-Methode	SQL-Typ
getInt(...)	INTEGER
getLong(...)	BIG INT
getFloat(...)	REAL
getDouble(...)	FLOAT
getBignum(...)	DECIMAL
getBigDecimal(...)	NUMBER
getBoolean(...)	BIT
getString(...)	VARCHAR
getString(...)	CHAR
getAsciiStream(...)	LONGVARCHAR
getDate(...)	DATE
getTime(...)	TIME
getTimestamp(...)	TIME STAMP
getObject(...)	jeder Typ

Prepared Statement

- Represents a precompiled SQL statement which allows improved performance.
- It allows to execute the query multiple times and we can set the values according to our needs.

Prepared Statement

- Class: java.sql.PreparedStatement
- Method: prepareStatement() of Connection
- As arguments the method takes over the SQL string, whereby concrete, changing values are replaced by the placeholder ?.

Prepared Statement

Select * from kunden where name=“Maier“ and vorname=“Hugo“

Connection conn=….

PreparedStatement ps=conn.prepareStatement(“select * from kunden where name = ? and vorname =?“);

ps.setString(1,“Meier“);

ps.setString(2,“Hugo“);

ResultSet rs = ps.executeQuery();

Was ist SQL-Injection?

SQL Injection

- Die „harmlose“ Variante:

```
stmt.executeUpdate("INSERT INTO PUBLIC.CUSTOMER VALUES
(5,'Lässig','Laura','Musterstraße 10','Musterstadt');Delete from
PUBLIC.CUSTOMER");
```

SQL Injection

- **SQL-Injection** (dt. *SQL-Einschleusung*) is the exploitation of a SQL database vulnerability caused by a lack of validation of metacharacters in user input
- SQL injections are possible when data such as user input gets into the SQL interpreter.
- Solution: Prepared Statements

SQL Injection

Die Hacker erbeuteten laut US-Experten 1,2 Milliarden Zugangsdaten. Und der Angriff ist noch nicht vorbei.

06.08.2014 | 18:29 | (Die Presse)

Washington/Wien. Sie sitzen angeblich in einer Kleinstadt im Süden Zentralrusslands, sind alle keine 30 Jahre alt und einem Bericht der „New York Times“ zufolge verantwortlich für den größten Datenklau in der Geschichte des Internets: 1,2 Milliarden Benutzernamen inklusive Passwörtern sollen dem kriminellen Ring in die Hände gefallen sein. 542 Millionen E-Mail-Adressen seien betroffen. Aufgedeckt wurde der Datendiebstahl von Hold Security, einer kleinen US-Firma, die bereits mehrere große Sicherheitslücken aufzeigte, wie etwa bei Adobe.

Wer diesmal betroffen ist, verraten die Experten nicht. „Denn die meisten Seiten sind noch immer angreifbar“, wird Firmengründer Alex Holden zitiert, der Datendiebstahl sei weiter im Gang und nur ein kleiner Teil der betroffenen Unternehmen bereits alarmiert. Nur so viel: Unter den angegriffenen Websites

März 2008: Hacker (darunter übrigens ein wahres Mastermind) ergattern 134 Millionen Kreditkartendaten beim amerikanischen Konzern Heartland Payment Systems. Mitte 2016: Mutmaßlich russische Hacker verschaffen sich Zugriff auf die Datenbank registrierter Wähler des Illinois State Board of Elections. Ähnliches passiert in Arizona. Februar 2017: Dem amerikanischen Waffenverkäufer Airsoft GI werden Daten von 65.000 User-Accounts gestohlen. März 2017: Mutmaßlich chinesische Hacker gelangen an die persönlichen Daten von 4.000 Kunden einer koreanischen App und verschicken teils obszöne Textnachrichten an die Opfer.

Date, Time and Timestamp

- java.sql.Date: represents a date (yyyy-mm-dd)
- java.sql.Time: represents a time (hh:mm:ss)
- java.sql.Timestamp: combines date and time
- Conversion java.util.Date
  - java.sql.Date sqlDate=new java.sql.Date(java.util.Date.getTime());

Conversion LocalDate/LocalDateTime

- java.sql.Date date = java.sql.Date.valueOf(LocalDate.now());
- java.sql.Date date = java.sql. Date.valueOf(LocalDate.of(2020, 03, 12));
- java.sql.Timestamp date = java.sql.Timestamp.valueOf(LocalDateTime.now());
- java.sql.Timestamp date = java.sql.Timestamp.valueOf(LocalDateTime.of(2020, 03, 12, 13, 55, 36, 123));

Null-Value

String s=rs.getString(column);

If (rs.**wasNull()**) sout („SQL-NULL“);

Row counter (Work Around)

```
rs.last();

Int rows=rs.getRow();

rs.beforeFirst();
```

SQLException

- Bündel von Ausnahmen

```
try {…}
catch(SQLException e)
{
  for (;e!=null;e=e.getNextException())
  {
        System.err.println(“Message: “+e.getMessage());
        System.err.println(“SQL State: “+e.getSQLState());
        System.err.println(“Error Code: “+e.getErrorCode());
  }
}
```

Metadata

- ResultSetMetaData
- Example: Infos about ResultSet

```
ResultSetMetaData meta=rs.getMetaData();
meta.getColumnCount();
meta.getColumnLabel();
meta.getColumnTypeName();
.....
```

Metadata

- DatabaseMetaData
- Examples:

```
DatabaseMetaData meta=con.getMetadata();
meta.getDatabaseProductName();
meta.getDatabaseProductVersion();
```