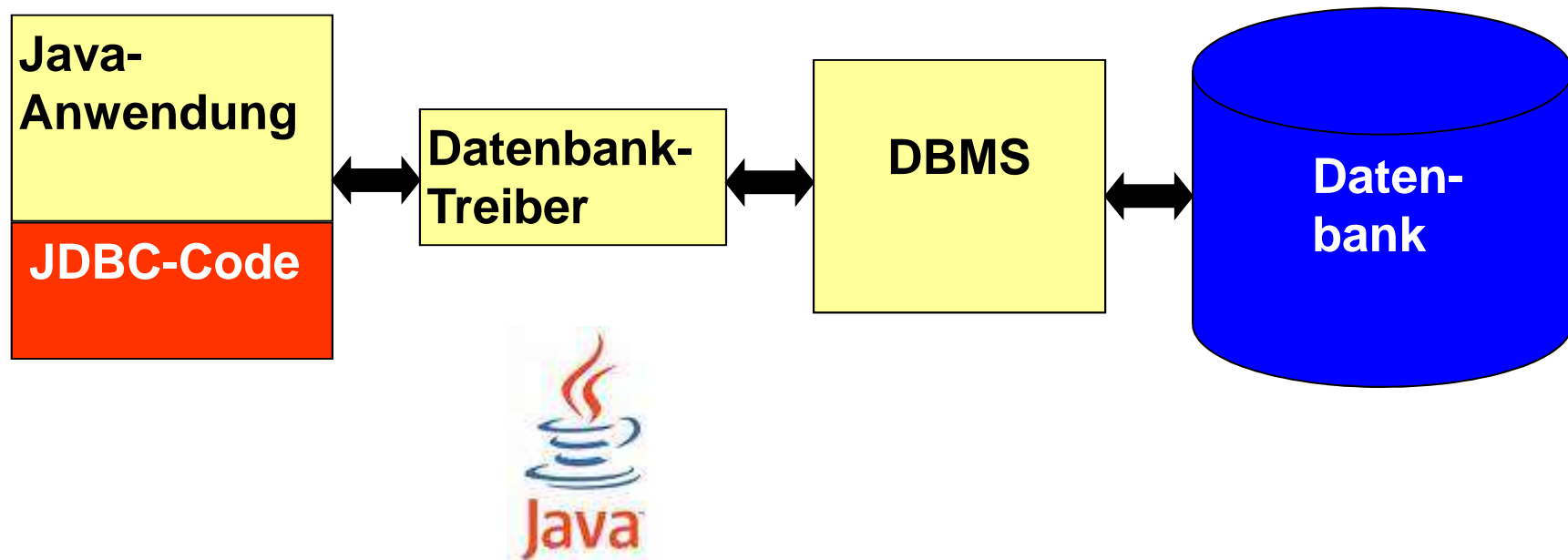


# JDBC



Programmierschnittstelle für portable Java-Datenbankanwendungen.

- java.sql: generische Typen und abstrakte Methodendeklarationen
- Treiberpaket: Implementierung des Frameworks



# Struktur einer JDBC-Anwendung (1)



## Treiber registrieren:

```
DriverManager.registerDriver(new driver)
```

oder

```
Class.forName(driver)
```

## Verbindung zur DB herstellen:

```
Connection conn = DriverManager.getConnection(...)
```

## Statement erzeugen:

```
Statement stmt = conn.createStatement()
```

## Statement ausführen:

```
ResultSet rset = stmt.executeQuery(statement)
```

oder

```
int count = stmt.executeUpdate(statement)
```

# Struktur einer JDBC-Anwendung (2)



## Metadaten abfragen:

```
ResultSetMetaData rsmd=rset.getMetaData();
```

## Iteration durch Ergebnismenge:

```
while (rset.next() )  
{ ...  
    String val1 = rset.getString(colname);  
    int val2 = rset.getInt(colindex)  
    ... }
```

## ResultSet und Statement schließen:

```
rset.close();  
stmt.close();  
rsmd.close();
```

## Verbindung abbauen:

```
conn.close();
```

# Der JDBC-Treiber (Beispiel Access)



```
import java.sql.*;
class jdbcctest
{ public static void main(String args[ ]) throws SQLException, ClassNotFoundException
{ //connection logic
  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
  Connection con =
      DriverManager.getConnection("jdbc:odbc:DbPerson","admin","admin");
  Statement stmt = con.createStatement( );
  ResultSet rset =stmt.executeQuery("Select * from Person");
  ResultSetMetaData rsetmd = rset.getMetaData( );
  int numberOfColumns = rsetmd.getColumnCount( );
  for (int i=1; i<=numberOfColumns; i++)
  { System.out.print(" "+rsetmd.getColumnName(i));}
    while (rset.next())
    { System.out.println(" ");
      for (int i=1;i<=numberOfColumns;i++) {System.out.print(" "+rset.getString(i));}
    }
    System.out.println(" ");
  } }
```

# Metadaten abfragen



Das Objekt `ResultSetMetaData` liefert Informationen über das `ResultSet`.

```
ResultSetMetaData rsmeta = rset.getMetaData ( );
```

## Einige Methoden von ResultSetMetaData

- `int getColumnCount ( )` throws `SQLException`;
- `String getTableName (int col)` throws `SQLException`;
- `String getColumnName (int col)` throws `SQLException`;
- `int getColumnType (int col)` throws `SQLException`;
- `String getColumnName (int col)` throws `SQLException`;

Das Objekt `DatabaseMetaData` liefert Infos über die DB und das DBMS.

```
DatabaseMetaData dbmeta = con.getMetaData ( );
```

```
String [ ] tbl_types = { "TABLE", "VIEW" }; ResultSet rset1, rset2;  
rset1 = dbmeta.getTables (con.getCatalog ( ), schema, "%", tbl_types);  
rset2 = dbmeta.getColumns(con.getCatalog ( ), schema, tab_name, "%);
```

# Abbildung von SQL-Datentypen in Java



Lesen von Attributwerten aus einem ResultSet mit `getXXX`

| JDBC-Typ | Java-Typ             | Methode           |
|----------|----------------------|-------------------|
| CHAR     | String               | getString ( )     |
| VARCHAR  | String               | getString ( )     |
| DATE     | Java.sql.Date        | getDate ( )       |
| INTEGER  | int                  | getInt ( )        |
| REAL     | float                | getFloat ( )      |
| NUMERIC  | Java.math.BigDecimal | getBigDecimal ( ) |
| ...      | ...                  | ...               |

Hinweis: Es können *alle* Datentypen mit `getString ( )` und `getObject ( )` gelesen werden.

# JDBC-Treiber von Oracle



**Für Oracle 12c im AI-Labor:**

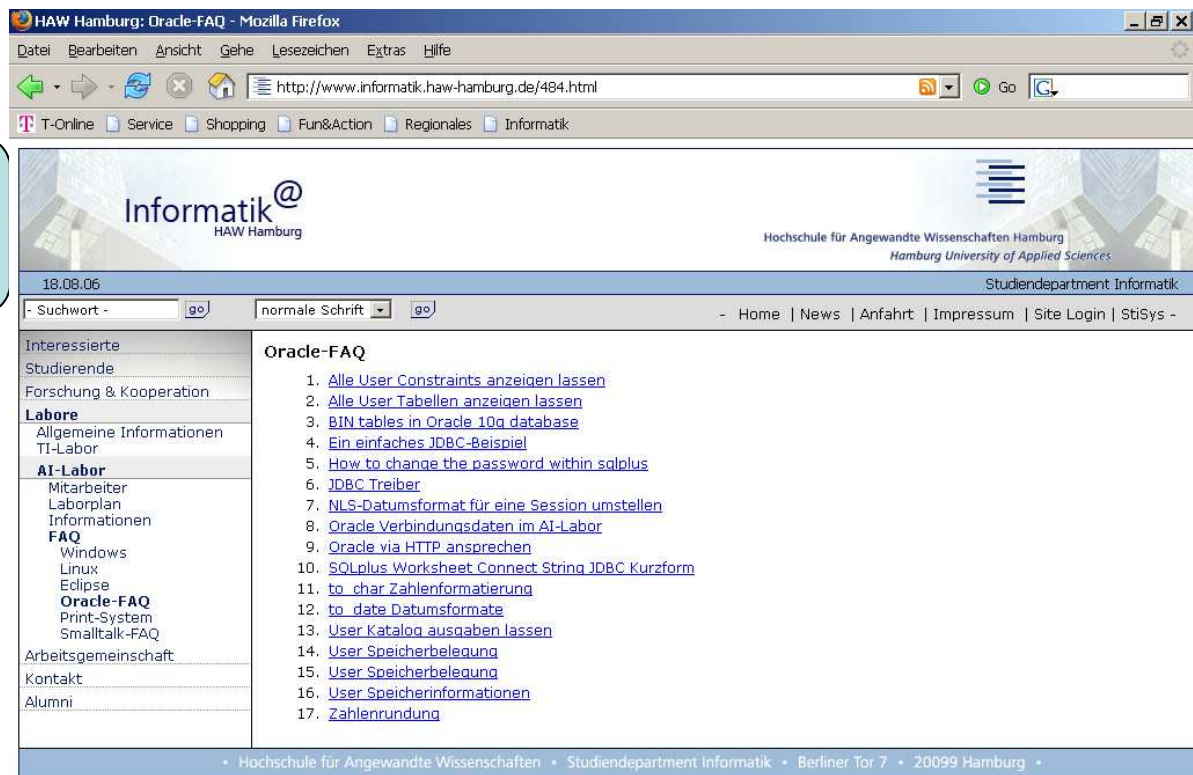
classpath: O:\client\ojdbc6.jar

DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());

Connection conn = DriverManager.getConnection

("jdbc:oracle:thin:@ora14.informatik.haw-hamburg.de:1521:inf14",  
"user", "pwd");

Eingeben oder aus  
Property-Datei



04.11.2015

