



JDBC Grundgerüst

- Basisschnittstelle für alle SQL-Anweisungen:

`java.sql.Statement`

- Statement Object erzeugen:

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

- SQL-Anweisungen ausführen

```
ResultSet rs = stmt.executeQuery("SELECT * FROM CUSTOMER");
```

- SQL-Anweisung enthält kein abschließendes Semikolon

- unterschiedlich von DBMS zu DBMS
 - Treiber ergänzt dies bei Bedarf



JDBC Grundgerüst

- ④ Ergebnis der SQL Abfrage (Tabelle): ResultSet
 - ④ Ergebniscursor (=Position in Ergebnismenge)
 - ④ Zeilenweise vorwärts mittels .next()
 - ④ Anfang: .first(), Ende: .last()
 - ④ Spaltenzugriff über getxxx() Methoden
 - ④ Spaltenindex bzw. Spaltenname als Parameter



JDBC Grundgerüst

- ④ Entsprechend den Java-Typen gibt es passende getter-Methoden in der Klasse `ResultSet`:
 - ④ `byte getByte(...)`
 - ④ `short getShort(...)`
 - ④ `int getInt(...)`
 - ④ ...



JDBC Grundgerüst

① Beispiel 1:

```
ResultSet rs = stmt.executeQuery( "SELECT * FROM CUSTOMER" );  
while ( rs.next() )  
    System.out.printf( "%s, %s %s, %s %s \n",  
        rs.getString(1),  
        rs.getString(2),  
        rs.getString(3),  
        rs.getString(4),  
        rs.getString(5)  
    );
```



JDBC Grundgerüst

④ Beispiel 2:

```
rs = stmt.executeQuery( "SELECT LASTNAME,  
                           CITY FROM Customer" );  
  
while ( rs.next() )  
    System.out.printf( "%s, %s\n",  
                      rs.getString(1),  
                      rs.getString(2)  
    );
```



JDBC Grundgerüst

④ Beispiel 3:

```
rs = stmt.executeQuery( "SELECT LASTNAME,  
                         CITY FROM Customer" );  
  
while ( rs.next() )  
    System.out.printf( "%s \n",  
                       rs.getString( "LASTNAME" ) );
```



JDBC Grundgerüst

- ④ Freigabe der DB Ressourcen
 - ④ RecordSet
 - ④ Statement
 - ④ Connection
- ④ mit .close() schließen!!



JDBC Grundgerüst

④ Datenbankänderungen

- ④ Änderungen der DB-Struktur (z.B. CREATE TABLE)
- ④ Änderungsbefehle (INSERT, UPDATE, DELETE)

```
int executeUpdate(String sql)  
throws SQLException
```

④ Rückgabewert bei INSERT, UPDATE, DELETE

- ④ Anzahl der geänderten Datensätze



JDBC Datentypen

- SQL arbeitet mit anderen Datentypen als Java
- => eigene JDBC-Datentypen zur Typkonvertierung definiert
- zu finden in: `java.sql.Types`



JDBC Datentypen

JDBC Typ	Java Typ	Java Object-Typ
TINYINT	byte	java.lang.Byte
SMALLINT	short	java.lang.Short
INTEGER	int	java.lang.Integer
BIGINT	long	java.lang.Long
REAL	double	java.lang.Double
FLOAT	double	java.lang.Double
DOUBLE	double	java.lang.Double



JDBC Datentypen

JDBC Typ	Java Typ	Java Object-Typ
DECIMAL	java.math.BigDecimal	java.math.BigDecimal
NUMERIC	java.math.BigDecimal	java.math.BigDecimal
DATE	java.sql.Date	java.sql.Date
TIME	java.sql.Time	java.sql.Time
TIMESTAMP	java.sql.Timestamp	java.sql.Timestamp



JDBC Datentypen

JDBC Typ	Java Typ	Java Object-Typ
CHAR	java.lang.String	java.lang.String
VARCHAR	java.lang.String	java.lang.String
LONGVARCHAR	java.lang.String	java.lang.String
BIT	boolean	java.lang.Boolean
BINARY	byte[]	byte[]
VARBINARY	byte[]	byte[]
LONGVARBINARY	byte[]	byte[]



ÜBUNG

- ④ Übung 1 – JDBC Datentypen
- ④ Übung 2 – JDBC Grundgerüst