JDBC

Datenbanktechnologien

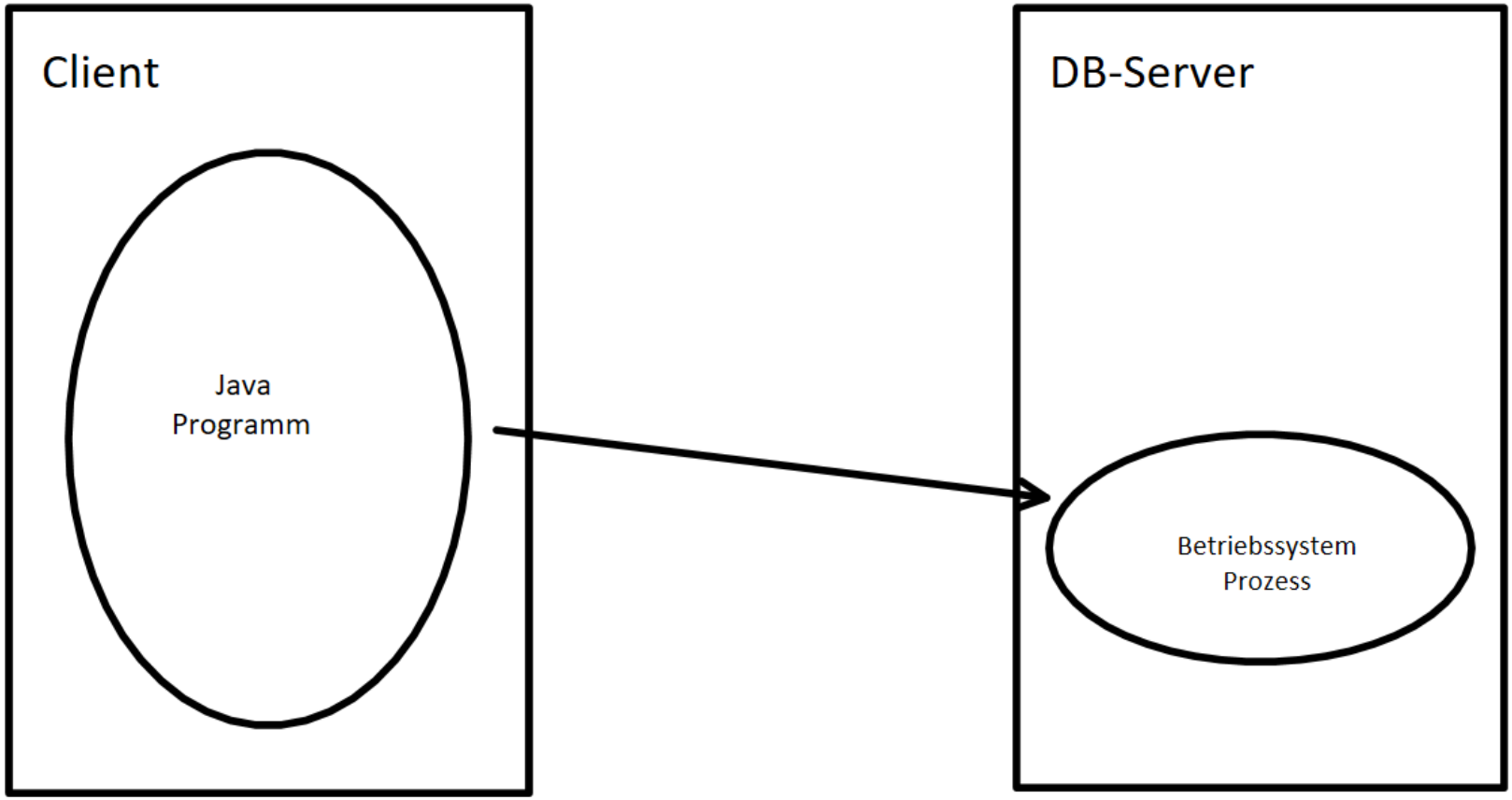
Prof. Dr. Ingo Claßen

Hochschule für Technik und Wirtschaft Berlin

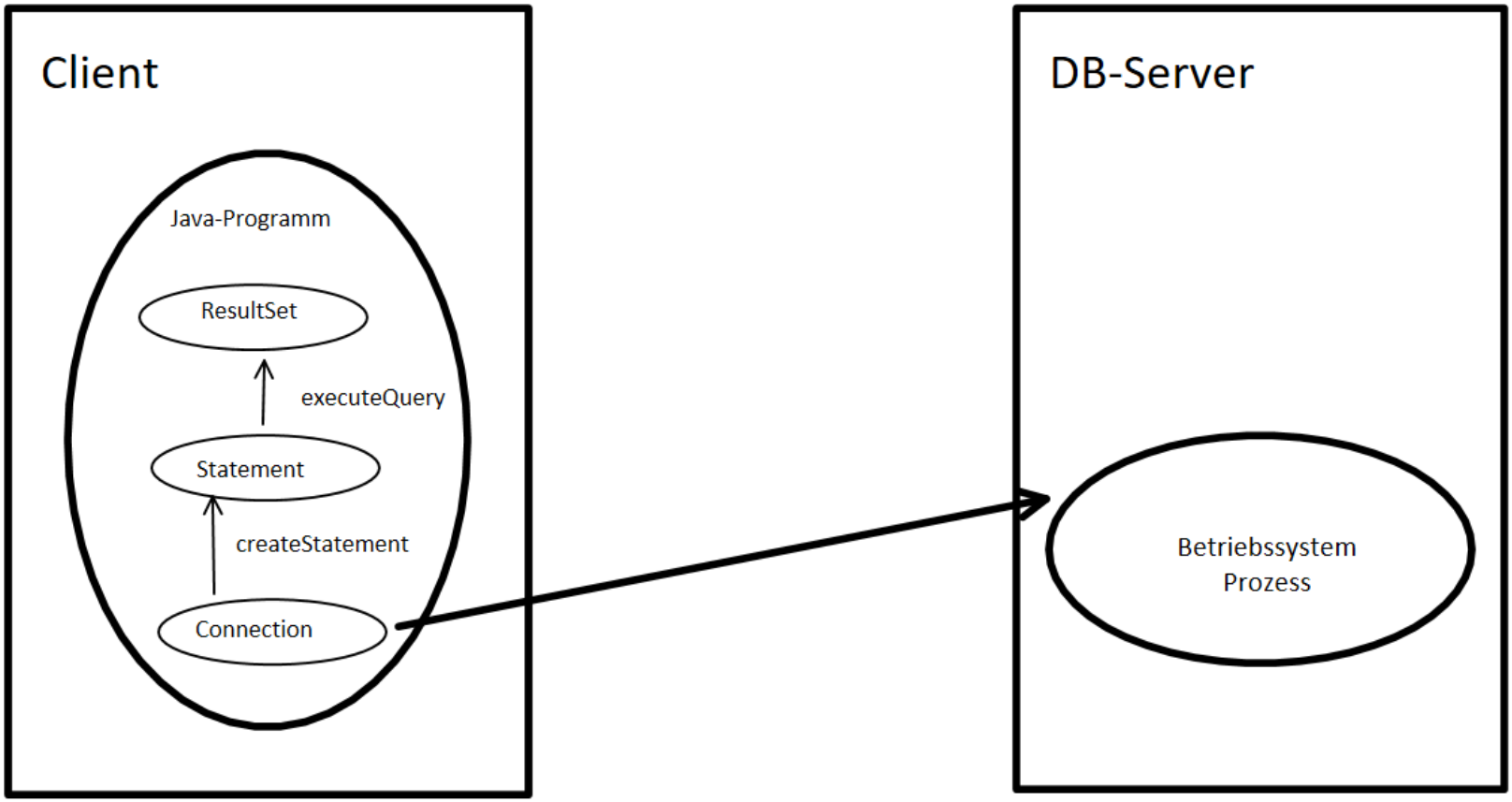
JDBC-Konzepte

JDBC-Programmierung

Verbindung Java Programm – DB-Server



JDBC-Verarbeitungsablauf

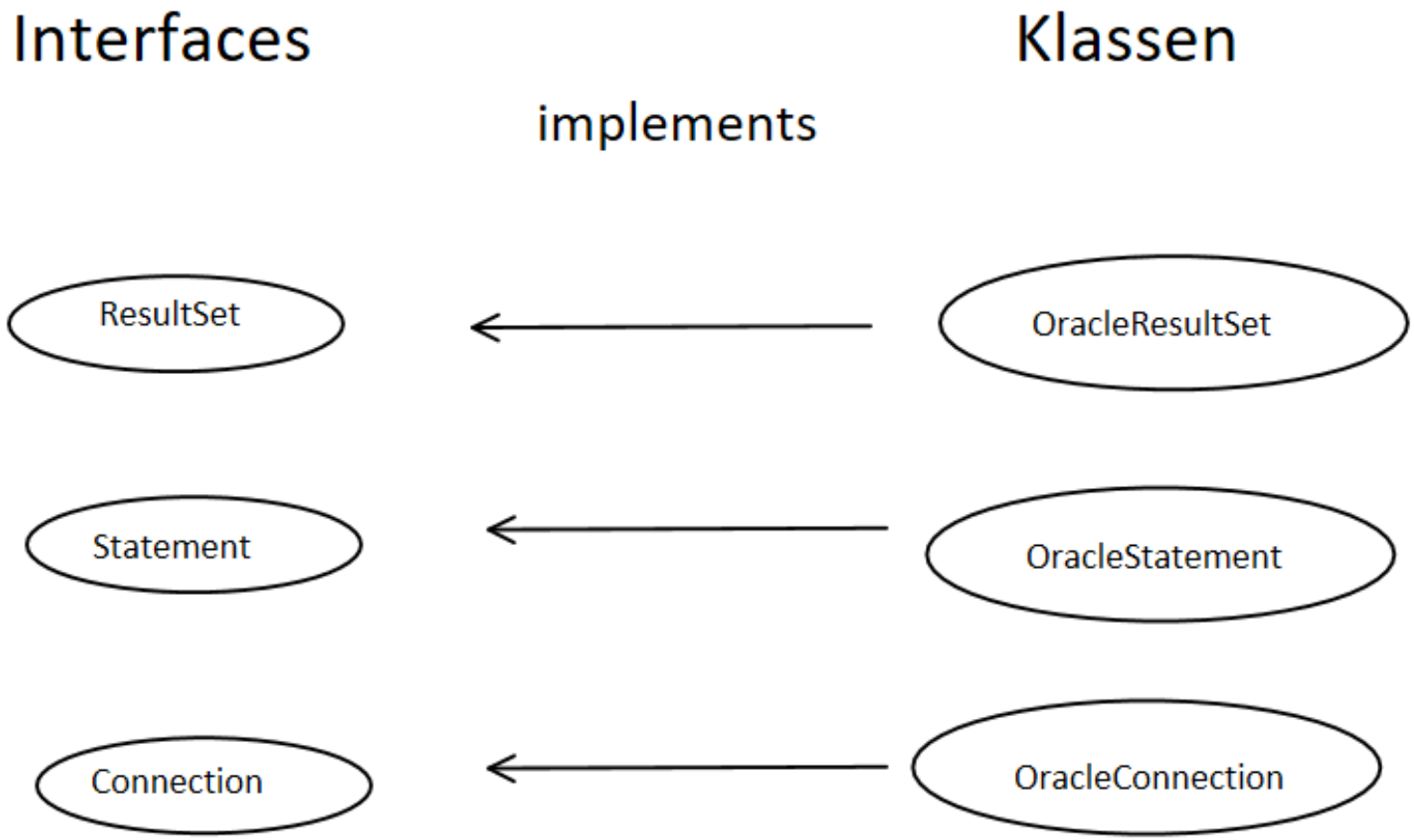


Datenbank-URL

Protokoll	JDBC
Treiber	oracle:thin
Host	host.htw-berlin.de
Port	1521
Instanz	sid

jdbc:oracle:thin:@host.htw-berlin.de:1521:sid

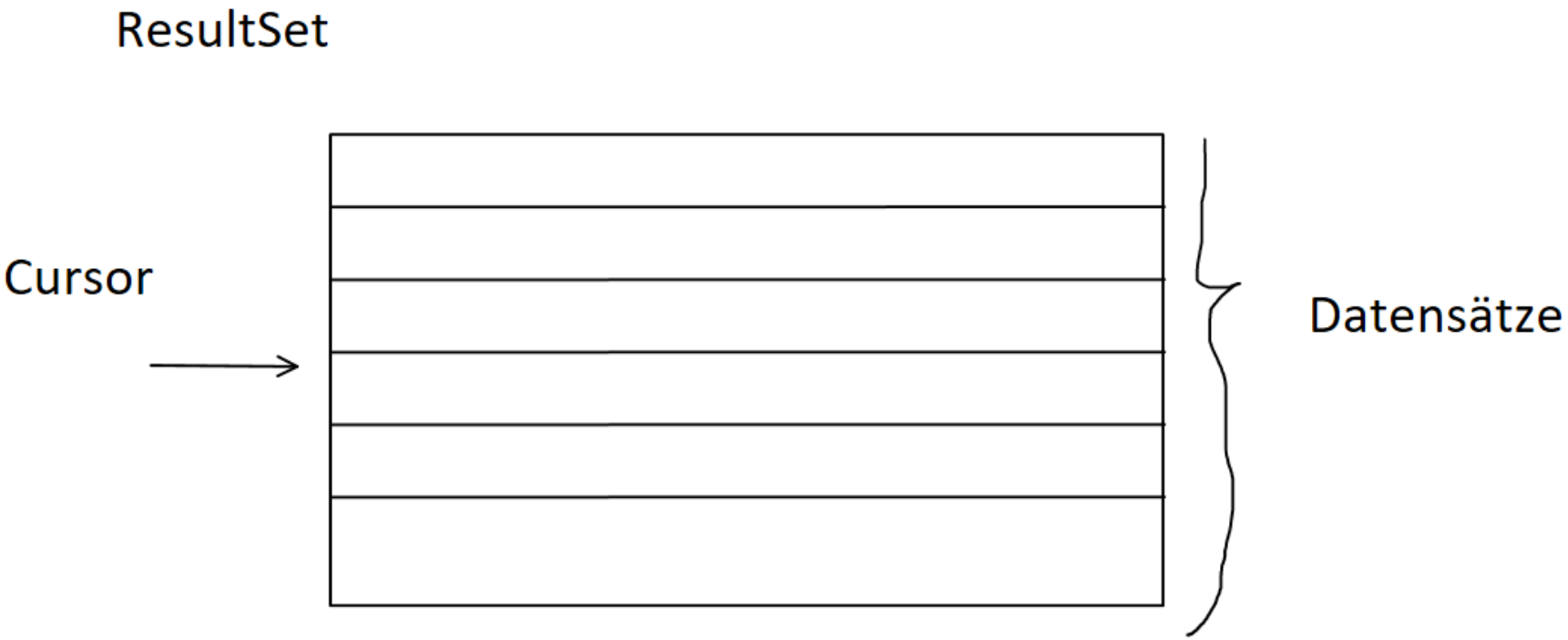
JDBC-Interfaces



Achtung: Die Bezeichnung der Oracle-Klassen erfolgt hier nur beispielhaft.
Die echten Klassen haben andere Namen.

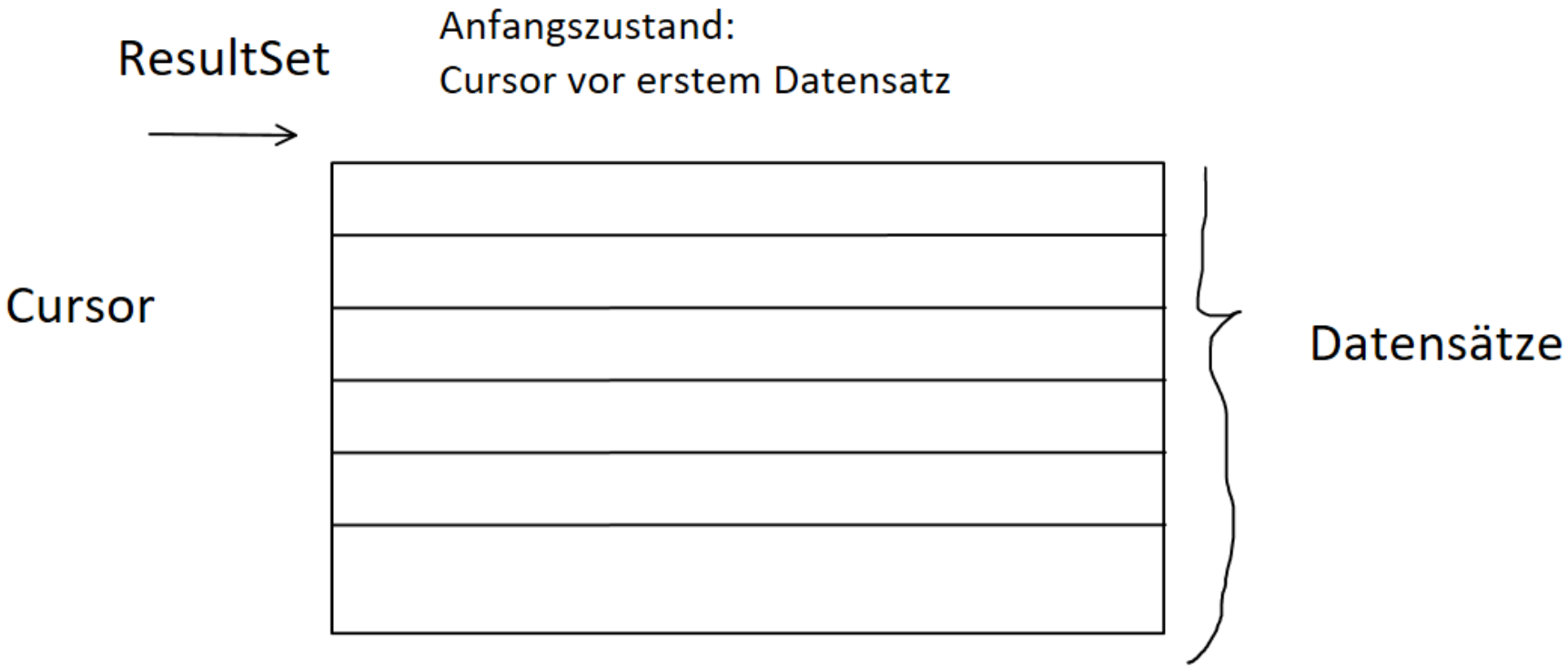
Ergebnismenge

```
statement.executeQuery("select * from Kunde");
```



Ergebnismenge – Anfangszustand

```
statement.executeQuery("select * from Kunde");
```



Damit können auch leere Ergebnismengen verarbeitet werden

Zugangsdaten

```
public interface DbCred {
    final String driverClass = "oracle.jdbc.driver.OracleDriver";
    final String url = "jdbc:oracle:thin:@host.f4.htw-berlin.de:1521:sid";
    final String user = "userid";
    final String password = "password";
    final String schema = "schema";
}
```

Die Zugangsdaten auf dieser Folie sind nur Dummies. Die echten Zugangsdaten erhalten Sie im Abschnitt *Konfiguration* auf folgender Seite `https://bit.ly/32M6okZ`.

Herstellen einer Verbindung

- DriverManager – Fabrik zur Erzeugung von Verbindungen
- Treiberauswahl über DB-URL
- Auto-Closing der Verbindung, d.h. automatisches Schließen der Verbindung beim Verlassen des Try-Catch-Blocks

```
try (Connection c =
        DriverManager.getConnection(
          DbCred.url, DbCred.user, DbCred.password)
    )
{
  L.info("Verbindungsaufbau erfolgreich");
} catch (SQLException e) {
  L.error("Verbindungsaufbau gescheitert", e);
}
```

Lesen von Datensätzen

- Variable `c` enthält eine geöffnete Verbindung
- `c.createStatement()` erzeugt ein `Statement`-Objekt
- `stmt.executeQuery(sql)` führt Befehl aus

```
String sql = "select RID, RaumNr from Raum";
try (Statement stmt = c.createStatement()) {
  try (ResultSet rs = stmt.executeQuery(sql)) {
    while (rs.next()) {
      Integer rid = rs.getInt("RID");
      String raumNr = rs.getString("RaumNr");
      System.out.println(String.format("|%3d|%6s|", rid, raumNr));
    }
  }
} catch (SQLException e) {
  L.error("", e);
  throw new DataException(e);
}
```

```
|  1|  A027|
|  2|  A028|
|  3|  A029|
```

- `ResultSet rs` enthält Ergebnismenge
- Weiterschaltung mit `rs.next()` und Prüfung, ob (noch) ein Datensatz vorhanden ist

Abfragen mit Parameter

- `c.prepareStatement(sql)` erzeugt ein `Statement`-Objekt, das mit dem SQL-Befehl initialisiert wurde
- Abfrage eithält einen Parameter (?)
- `stmt.setInt(1, rid)` setzt Parameter der Abfrage
- `stmt.executeQuery()` führt vorbereiteten Befehl aus

```
String sql = String.join(" ",
  "select RID, RaumNr",
  "from Raum",
  "where rid=?");
try (PreparedStatement stmt = c.prepareStatement(sql)) {
  stmt.setInt(1, rid);
  try (ResultSet rs = stmt.executeQuery()) {
    if (rs.next()) {
      String raumNr = rs.getString("RaumNr");
      System.out.println(String.format("|%3d|%6s|", rid, raumNr));
    } else {
      throw new RaumException("rid existiert nicht in db: " + rid);
    }
  }
} catch (SQLException e) {...}
```

```
|  1|  A027|
```

Einfügen / Löschen eines Datensatzes

- `stmt.executeUpdate(sql)` statt `stmt.executeQuery(sql)`
- `stmt.executeUpdate(sql)` wird für `insert`, `delete` und `update` auf SQL-Ebene verwendet
- Kein ResultSet

```
String sql = String.join(" ",
  "insert into raum(rid, raumnr, anzahlsitze)",
  "values (7, 'A030', 10)");
try (Statement stmt = c.createStatement()) {
  stmt.executeUpdate(sql);
} catch (SQLException e) {...}
```

```
String sql = "delete from raum where RID = 7";
try (Statement stmt = c.createStatement()) {
  stmt.executeUpdate(sql);
} catch (SQLException e) {...}
```

Umwandlung ResultSet in Liste

```
List<String> l = new LinkedList<String>();
String sql = "select * from raum";
try (Statement stmt = c.createStatement()) {
  try (ResultSet rs = stmt.executeQuery(sql)) {
    while (rs.next()) {
      l.add(rs.getString("RaumNr"));
    }
  }
} catch (SQLException e) {...}
```

```
[A027, A028, A029]
```

Umwandlung ResultSet in Objekt

```
String sql = String.join(" ",
  "select RID, RaumNr, AnzahlSitze",
  "from Raum ",
  "where rid=?");
Raum r = null;
try (PreparedStatement ps = c.prepareStatement(sql)) {
  ps.setInt(1, rid);
  try (ResultSet rs = ps.executeQuery()) {
    if (rs.next()) {
      String raumNr = rs.getString("RaumNr");
      int anzahlSitze = rs.getInt("AnzahlSitze");
      r = new Raum(rid, raumNr, anzahlSitze);
    } else {
      throw new RaumException("rid existiert nicht in db: " + rid);
    }
  }
} catch (SQLException e) {...}
```

```
Raum [rid=1, raumNr=A027, anzahlSitze=60]
```