

Datenbankzugriffe

***VO Grundlagen & Konzepte der Softwareentwicklung WS
09/10
(B. Resch)***

Gruber Günter, Mayer Stefan

Inhalt

- SQL – Structured Query Language
- Oracle
 - Nearest Neighbor
 - Buffer
 - Within Distance
- JDBC – Datenbankzugriff mit Java

SQL

- standardisierte Sprache zur
 - Definition,
 - Abfrage u.
 - Manipulationvon Daten in relationalen Datenbanken

- aktuell: SQL:2008 ISO 9075:2008

SQL Commands

SELECT

Data retrieval

INSERT

UPDATE

DELETE

MERGE

Data manipulation language (DML)

CREATE

ALTER

DROP

RENAME

TRUNCATE

Data definition language (DDL)

COMMIT

ROLLBACK

SAVEPOINT

Transaction control

GRANT

REVOKE

Data control language (DCL)

- Oracle Database
 - RDBMS und ORDBMS
 - Programmiersprache: PL/SQL
- Oracle Spatial u. Oracle Locator
 - Speicherung von Geodaten (Location, Extent, Shape) durch Datentyp SDO_GEOMETRY
 - entspricht OGC Simple Feature Guidelines
- proprietär, aber für Studienzwecke frei

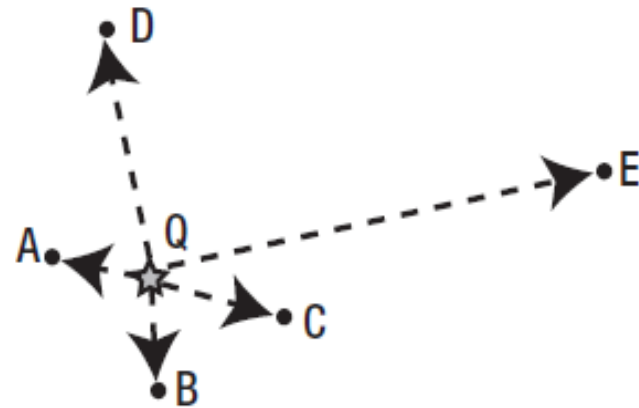
Nearest Neighbor

```
SQL> SELECT poi_name
FROM us_interstates I, us_restaurants P
WHERE I.interstate = 'I795'
      AND SDO_NN(P.location, I.geom) = 'TRUE'
      AND ROWNUM <= 5;
```

POI_NAME

PIZZA BOLI'S
BLAIR MANSION INN DINNER THEATER
KFC
CHINA HUT
PIZZA HUT

5 rows selected.

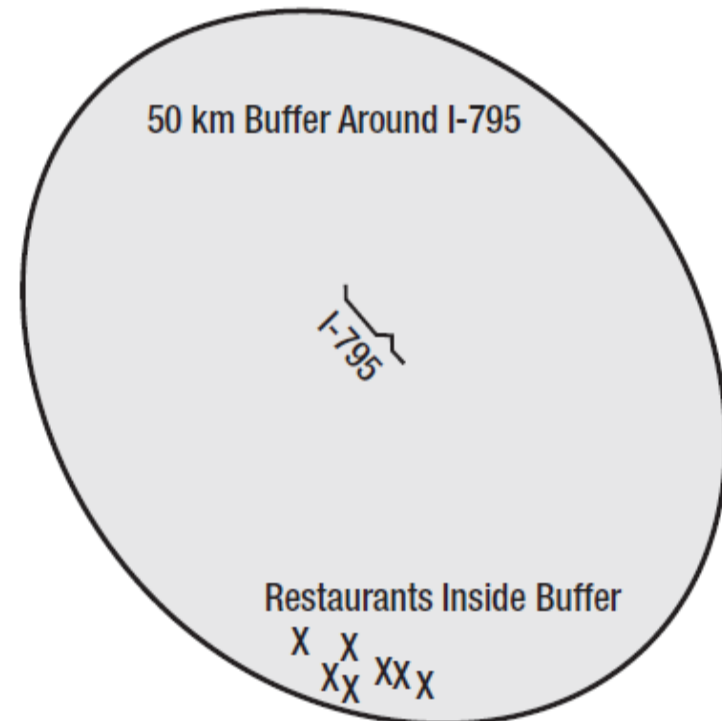


Buffer

```
SQL> SELECT POI_NAME
FROM us_interstates I, us_restaurants P
WHERE
  SDO_ANYINTERACT
  (
    P.location,
    SDO_GEOM.SDO_BUFFER(I.geom, 50, 0.5, 'UNIT=KM')
  ) = 'TRUE'
  AND I.interstate='I795' ;
POI_NAME
```

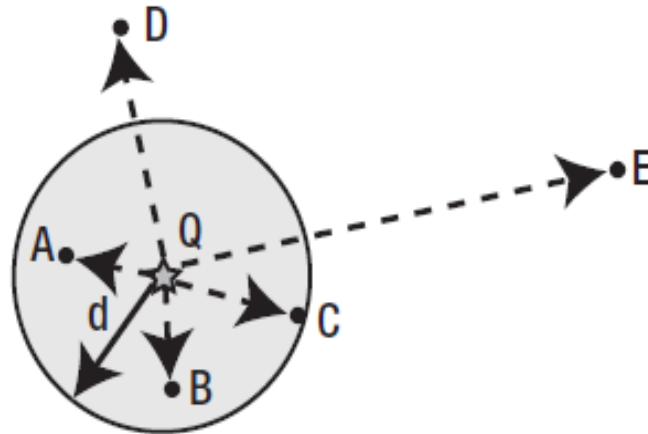
```
-----
SPICY DELIGHT
PHILLY'S STEAK EXPRESS
EL TAMARINDO
MCDONALD'S
PIZZA HUT
CHINA HUT
KFC
BLAIR MANSION INN DINNER THEATER
PIZZA BOLI'S
```

9 rows selected.



Within Distance

```
SQL> SELECT ct.id, ct.name
FROM competitors comp, customers ct
WHERE comp.id=1
  AND SDO_WITHIN_DISTANCE
  (ct.location, comp.location, 'DISTANCE=0.25 UNIT=MILE ' )='TRUE'
ORDER BY ct.id;
```



Datenbankzugriffe mit JAVA

- **Relationales Datenbankmodell**

- Tabellen und ihre Beziehungen untereinander als Grundlage
- Abfrage und Manipulation durch SQL



- **Objektorientiertes Datenbankmodell**

- Objekte und ihre Attribute als Grundlage
- Datenabfrage durch OQL

- **Objekt-relationales Datenbankmodell**

- Bindeglied
- Beziehung von Objektmengen untereinander

PostgreSQL



➔ **JAVA kann auf alle drei Formen zugreifen**

Datenbankzugriffe mit JAVA

Datenbankzugriffe über das Internet (Client-Server-System):

Java Applet

+

Java Applikation (Servlet)

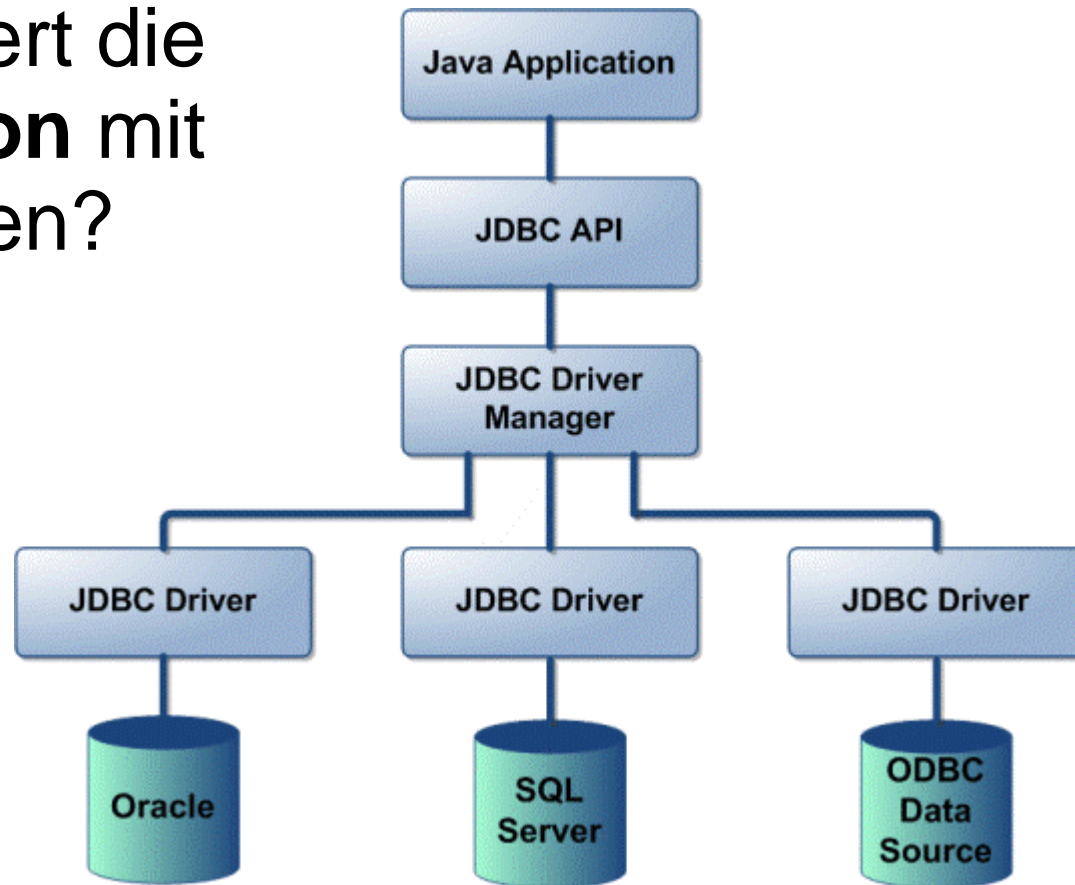


- Läuft im Browserfenster
- Stellt die Benutzerschnittstelle dar
- Sendet Abfragen an den Server und empfängt Information

- Erhält Abfragen vom Client
- Greift auf die Datenbank zu
- Liefert Information an den Client

Datenbankzugriffe mit JAVA

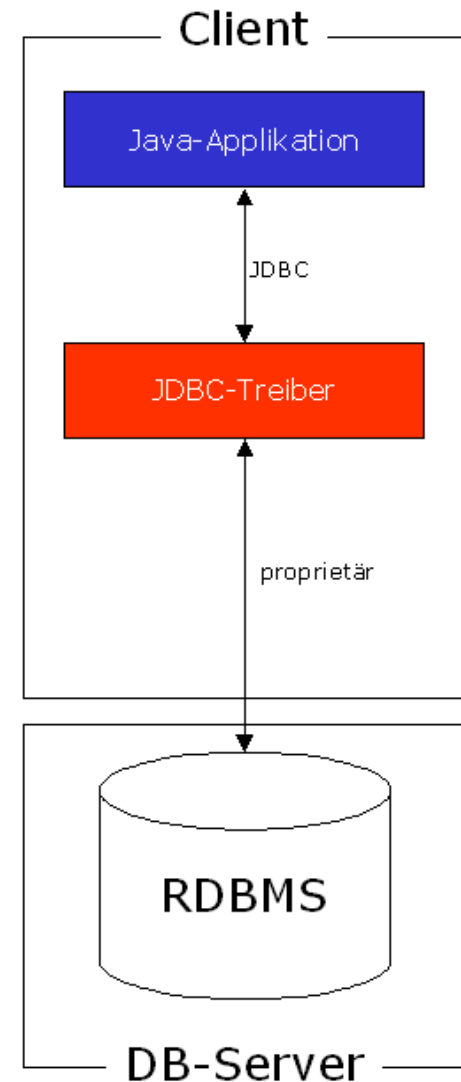
Wie kommuniziert die **Java Applikation** mit den Datenbanken?



Datenbankzugriffe mit JAVA

Was ist **JDBC** (Java Database Connectivity)?

- Eine Schnittstelle, die zwischen der Java-Applikation und dem DBMS vermittelt, d. h. ein Application Programming Interface (API)
- Aufgabe ist, Kommunikationsvorgänge aller Art abzuwickeln
- Normalerweise sind JDBC-Treiber in der Applikation oder der Datenbank implementiert
- Unabhängig vom Betriebssystem und vom Datenbanktyp
- Aber: Probleme bei gleichzeitigem Zugriff auf verschiedenartige Datenbanken



Datenbankzugriffe mit JAVA

```
1  import java.sql.DriverManager;
2  import java.sql.SQLException;
3  import com.mysql.jdbc.Connection;
4
5  public class JDBCConnect {
6      public static void main(String[] args) {
7          try {
8              Class.forName("com.mysql.jdbc.Driver");
9          } catch (ClassNotFoundException e) {
10             System.err.println("Driver class not found");
11             e.printStackTrace();
12         }
13         Connection con = null;
14         try {
15             con =
16                 (Connection) DriverManager.getConnection(
17                     "jdbc:mysql://localhost/jdbctest/",
18                     "mario",
19                     "thePassword");
20         } catch (SQLException e1) {
21             System.err.println("Error establishing database connection");
22             Throwable t = e1;
23             while (t != null) {
24                 System.err.println("Type: " + t.getClass().getName());
25                 System.err.println("Message: " + t.getMessage());
26                 System.err.println("-----");
27                 t = t.getCause();
28             }
29         }
30     }
31 }
32
```

Literatur- & Abbildungsverzeichnis

- GERTZ, M. (2000): Oracle/SQL Tutorial. – <http://www.db.cs.ucdavis.edu>
- KOTHURI, R., A. GODFRIND, E. BEINAT (2004): Pro Oracle Spatial. – www.apress.com
- <http://www.boku.ac.at/javaeinf/jein3.html#datenbank>
- <http://java.sun.com/docs/books/tutorial/jdbc/overview/index.html>
- <http://www.jdbctutorial.com/node/1>
- <http://www.jeckle.de/vorlesung/DB-Anwendungen/>
- <http://www.developersbook.com/jdbc/interview-questions/jdbc-interview-questions-faqs.php>