

JDBC

SOFTVERSKI ALATI BAZA PODATAKA

Komunikacija sa bazom podataka

- Podsystemi za komunikaciju sa RDBMS (*Relational database management system*) serverima zasnovanim na SQL-u su:
 - ODBC (Open Database Connectivity) je standardni API za povezivanje sa RDBMS
 - JDBC (Java Database Connectivity) je Java API za povezivanje sa RDBMS
- JDBC (Java Database Connectivity):
 - Pomoću klasa iz ovog paketa se mogu izvršavati SQL naredbe i manipulirati sa rezultatima dobijenim na osnovu ovih naredbi.
 - Celokupan podsystem je definisan u standardnom paketu `java.sql`
 - JDBC API sadrži niz apstraktnih Java interface-a koji dozvoljavaju programeru da ostvari konekciju sa određenom bazom podataka, izvrši SQL naredbe i obradi dobijene rezultate.

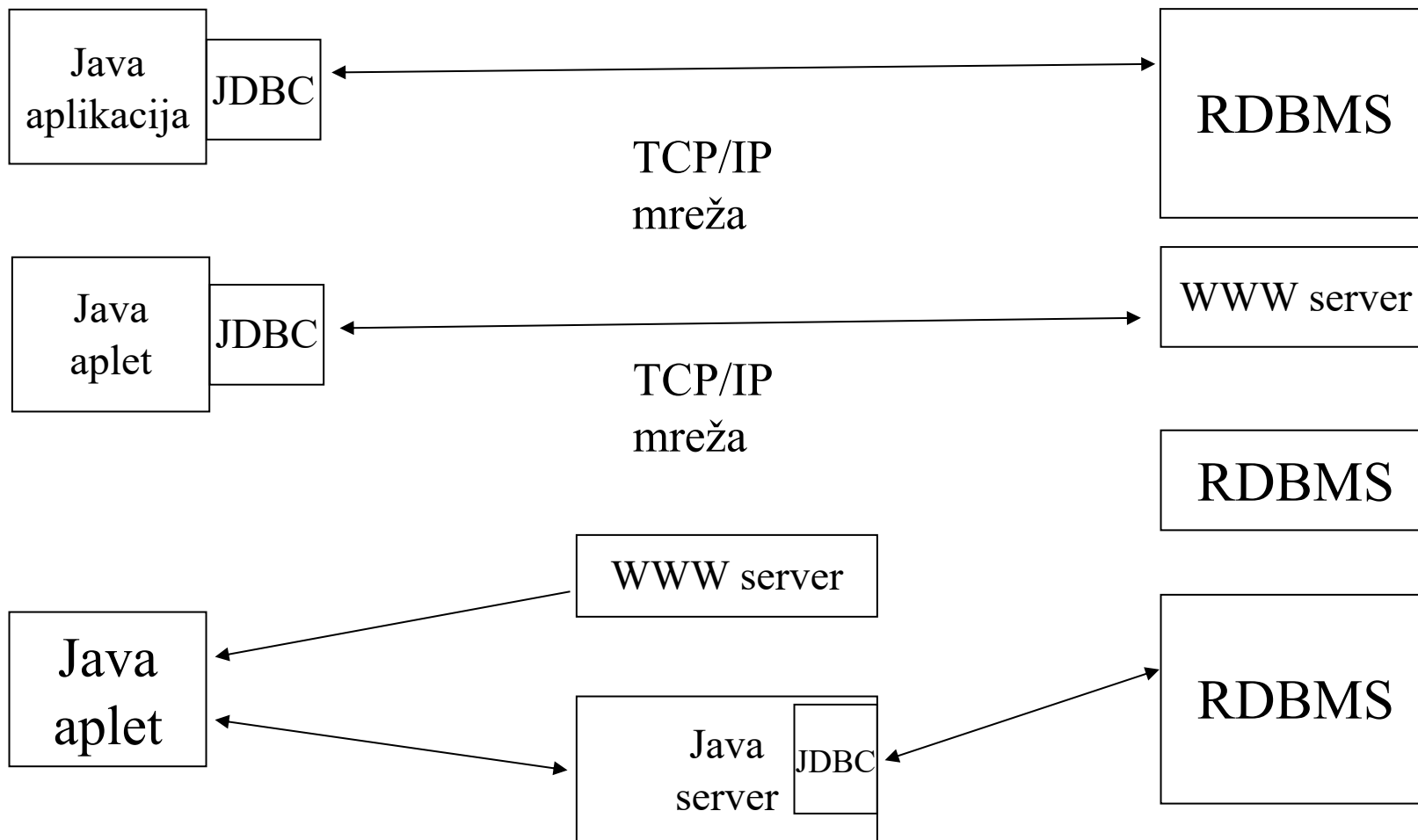
JDBC drajveri

- skup klasa koje implementiraju interfejsse iz paketa `java.sql`
- obezbeđuje ih svaki proizvođač RDBMS servera za svoje sisteme
- svi drajveri se na isti način koriste
- instalacija drajvera = uključivanje u CLASSPATH
- isti Java kod se može koristiti za rad sa serverima različitih proizvođača, uz korišćenje odgovarajućeg JDBC drajvera

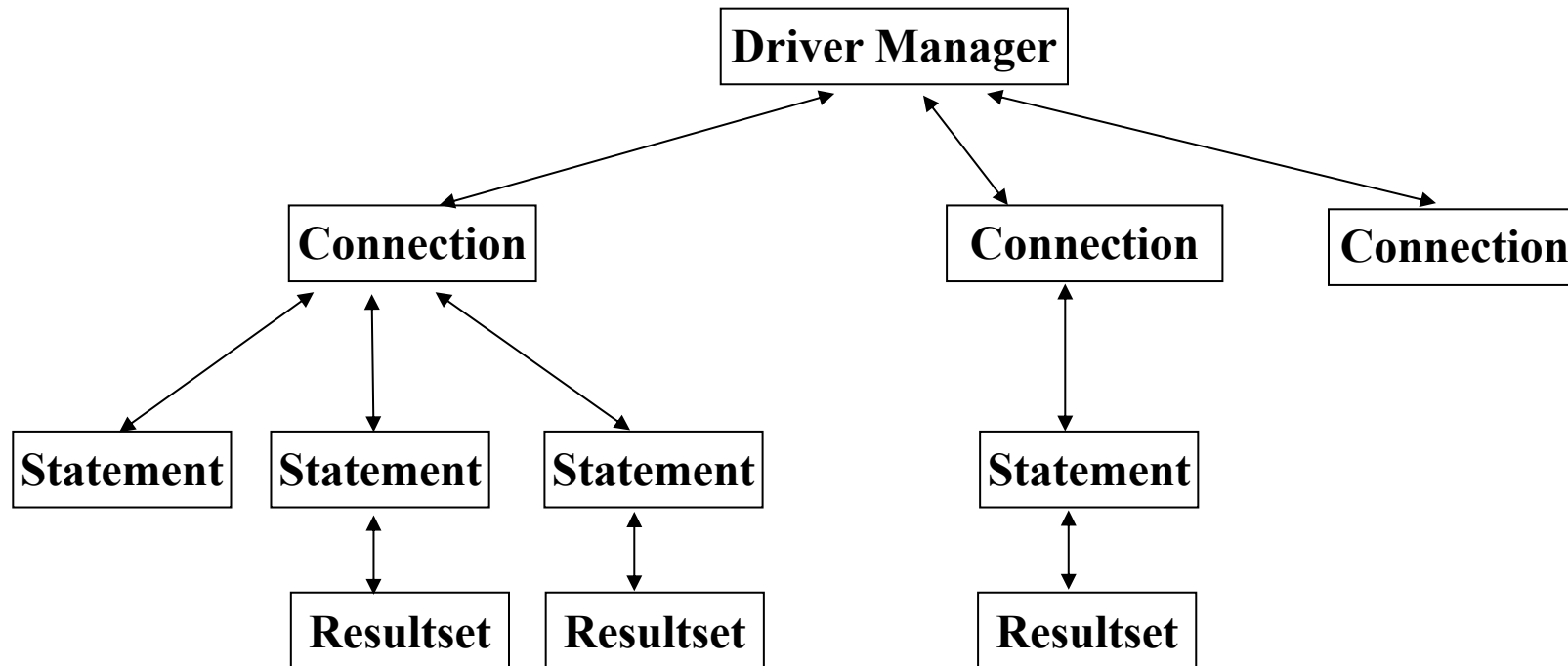
JDBC i klijent/server model

- pristup bazi podataka iz Java aplikacije
- dvoslojni (two-tier) model
- troslojni (three-tier) model
- CGI skriptovi i servleti
- Web servisi

Pristup bazi podataka iz Java aplikacije



Struktura



Osnovne klase

- **java.sql.DriverManager** sprovodi učitavanje driver-a baze podataka i omogućava podršku za kreiranje nove konekcije
- **java.sql.Connection** predstavlja konekciju sa određenom bazom podataka
- **java.sql.Statement** izvršava se u obliku container-a za izvršavanje SQL naredbi u okviru uspostavljene konekcije. Postoje dva podtipa:
 - **java.sql.PreparedStatement** za izvršavanje pre-kompajlirane SQL naredbe
 - **java.sql.CallableStatement** za izvršavanje poziva procedura koje postoje u okviru baze podataka.
- **java.sql.ResultSet** kontroliše pristup rezultatima dobijenim izvršavanjem određene SQL naredbi

Osnovne klase

- **java.sql.DatabaseMetaData** omogućava pristup informacijama o strukturi baze podataka
- **java.sql.ResultSetMetaData** omogućava pristup informacijama o strukturi rezultata dobijenih izvršavanjem određene SQL naredbe
- **java.sql.ParameterMetaData** omogućava pristup informacijama o parametrima kod PreparedStatement-a i CallableStatement-a

Inicijalizacija driver-a

- JDBC management nivo mora da zna koji drajver je raspoloživ i koji drajver se koristi.
- Primer inicijalizacije drajvera baze podataka:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
Class.forName("org.gjt.mm.mysql.Driver");
```

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

Uspostavljanje komunikacije

- Konekcija se uspostavlja pozivanjem metode:
`DriverManager.getConnection(url, [username, password]);`
- URL ima sledeću strukturu: `jdbc:<subprotocol>:<name>;`
- *subprotocol* je ime određene vrste mehanizma pristupa bazi podataka koji može biti podržan od jednog ili više drajvera. Sadržaj i sintaksa dela *name* zavisi od subprotocola.

- Primer Usuostavljanje konekcije:

```
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/STUDENT ",  
"USERNAME", "PASSWORD");
```

```
Connection conn =  
DriverManager.getConnection("jdbc:sqlserver://localhost:1433;database=STUDENT",  
"USERNAME", "PASSWORD");
```

```
Connection conn = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;  
database=STUDENT;integratedSecurity=true");
```

Završetak komunikacije

- Po završetak komunikacije s bazom podataka potrebno je zatvoriti konekciju pozivom metode:

```
Conn.close();
```

Izuzeci

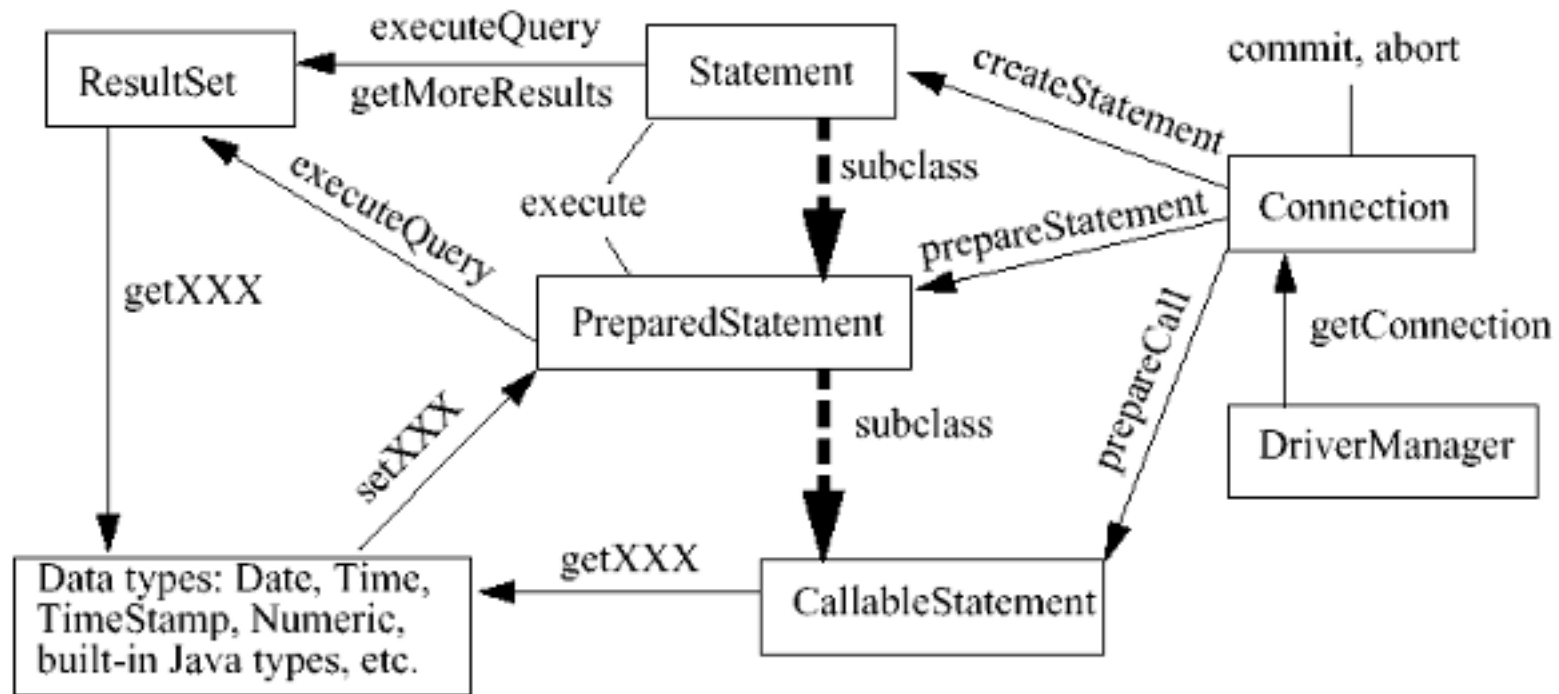
- Izuzeci koji mogu da nastanu su:
 - **ClassNotFoundException** - prilikom učitavanje JDBC drajvera
 - **SQLException** - prilikom uspostavljanja veze i kod svih ostalih metoda

Upravljanje transakcijama

- `conn.commit();`
- `conn.rollback();`
- `conn.setAutoCommit(boolValue);`

- `conn.setSavepoint(savepointName)`
- `conn.releaseSavepoint(savepointName)`
- `conn.rollback(savepointName)`

Dijagram izvršavanja



Statement – Najvažnije metode

- `Statement stmt = conn.createStatement()`
- `PreparedStatement stmt = conn.prepareStatement(query)`
- `CallableStatement stmt = conn.prepareCall(query)`
- `stmt.close()`

- `stmt.execute()`
- `stmt.executeQuery(query)`
- `stmt.executeUpdate(query)`

- `stmt.getUpdateCount()`

Statement – Najvažnije metode

- `stmt.setInt(paramPos, int_value)`
- `stmt.setString(paramPos, string_value)`
- `stmt.getGeneratedKey()`
 - Da bi automatski generisane ključeve bilo moguće dohvatiti, prilikom kreiranja Statement-a ili izvršavanja upita kao parametar `autoGeneratedKeys` prosleđuje se konstanta `Statement.RETURN_GENERATED_KEYS`
- `stmt.registerOutParameter(paramPos, Types.INTEGER);`
- `stmt.getInt(paramPos);`

ResultSet – Najvažnije metode

- `ResultSet rs = stmt.executeQuery(query)`
- `ResultSet rs = stmt.getResultSet()`
- `ResultSet rs = stmt.getMoreResults()`

- `rs.next()`
- `rs.first()`
- `rs.getString(colPos)`
- `rs.getString(colName)`

ResultSet – Najvažnije metode

- `rs.moveToInsertRow()`
- `rs.insertRow()`
- `rs.updateRow()`
- `rs.deleteRow()`

- `rs.updateString(colPos, value)`
- `rs.updateString(colName, value)`

- Da bi izmena podataka bila moguća u okviru ResultSet-a, prilikom kreiranja Statement-a kao parametar `resultSetConcurrency` prosleđuje se konstanta `ResultSet.CONCUR_UPDATABLE`

DatabaseMetaData – Najvažnije metode

- `DatabaseMetaData dbmd = conn.getMetaData()`

- `dbmd.getDriverVersion()`
- `dbmd.getDriverName()`
- `dbmd.getDatabaseProductVersion()`
- `dbmd.getDatabaseProductName()`

- `dbmd.getTables()`
- `dbmd.getColumns()`
- `dbmd.getProcedures()`

ResultSetMetaData – Najvažnije metode

- `ResultSetMetaData rsmd = rs.getMetaData()`
- `rsmd.getTableName()`
- `rsmd.getColumnCount()`
- `rsmd洗getColumnName(colPos)`
- `rsmd.getColumnType(colPos)`
- `rsmd.isReadOnly(colPos)`
- `rsmd.isAutoIncrement(colPos)`

ParameterMetaData – Najvažnije metode

- `ParameterMetaData pmd = rs.getParameterMetaData()`
- `pmd.getParameterCount()`
- `pmd.getParameterMode()`
- `pmd.getParameterType()`
- `pmd.isNullable()`

Izvršavanje Select upita - Primer

```
String query = "SELECT ime, prezime FROM studenti";
```

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

```
ResultSet rs = stmt.executeQuery(query);
```

```
while (rs.next()) {
```

```
    System.out.println(
```

```
        rs.getString(1) + " " + rs.getString(2));
```

```
}
```

```
rs.close();
```

```
stmt.close();
```

Izvršavanje DML upita - Primer

```
String query = "INSERT INTO studenti (ime, prezime) "  
              + " VALUES ('Nenad', 'Nenadovic)";  
Statement stmt = conn.createStatement();  
int num_of_row_changed = stmt.executeUpdate(query);  
stmt.close();
```

Prepared Statement - Primer

```
PreparedStatement stmt = conn.prepareStatement(
    "INSERT INTO studenti (student_id, ime, prezime) values (?, ?, ?)");
stmt.setInt(1, 1);
stmt.setString(2, new String("Marko"));
stmt.setString(3, new String("Markovic"));
stmt.executeUpdate();

stmt.setInt(1, 2);
stmt.setString(2, new String("Nenad"));
stmt.setString(3, new String("Nenadovic"));
stmt.executeUpdate();
stmt.close();
```


Callable Statement - Primer

```
CallableStatement stmt = conn.prepareCall("{? = call uradi (?, ?)}");
```

```
stmt.setString(2, new String("Sima"));
```

```
stmt.setString(3, new String("Simic"));
```

```
stmt.registerOutParameter(1, Types.INTEGER);
```

```
stmt.executeQuery();
```

```
System.out.println("Status: " + stmt.getInt(1));
```

```
stmt.close();
```

ResultSet - Primer

```
Statement stmt = conn.createStatement(ResultSet.TYPE_FORWARD_ONLY,  
    ResultSet.CONCUR_UPDATABLE);
```

```
ResultSet rs = stmt.executeQuery("SELECT ime, prezime FROM studenti");
```

```
if (rs.next()) {  
    rs.updateString(1, "Tamara");  
    rs.updateRow();  
}  
rs.close();  
stmt.close();
```

Povezivanje Jave sa SQL Serverom

- Sa zvaničnog sajta je potrebno skinuti Driver i raspakovati ga. U okviru projekta u kojem zelite da radite sa Microsoft SQL bazom podataka, potrebno je importovati odgovarajući jar koji se nalazi u raspakovanom fajlu.
- Radi autentifikacije korišćenjem windows naloga iz raspakovanog folder potrebno je kopirati fajl sqljdbc_auth.dll na lokaciju C:\Windows\System32.
 - U SQL Server Configuration Manager potrebno je da bude podešeno:
 - U tabu SQL Server Services, prikazuje se da je SQL Server pokrenut.
 - U tabu SQL Server Network Configuration>Protocols for MSSQLSERVER, svi protokoli treba da budu uključeni.
 - Za TCP/IP protokol vrednost TCP port-a treba da bude ista kao ona koja se koristi prilikom konekcije (port 1433).
- Link:
<https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>