



Tehnici avansate de programare

Curs 8

Departamentul de Informatică

popirlan@gmail.com



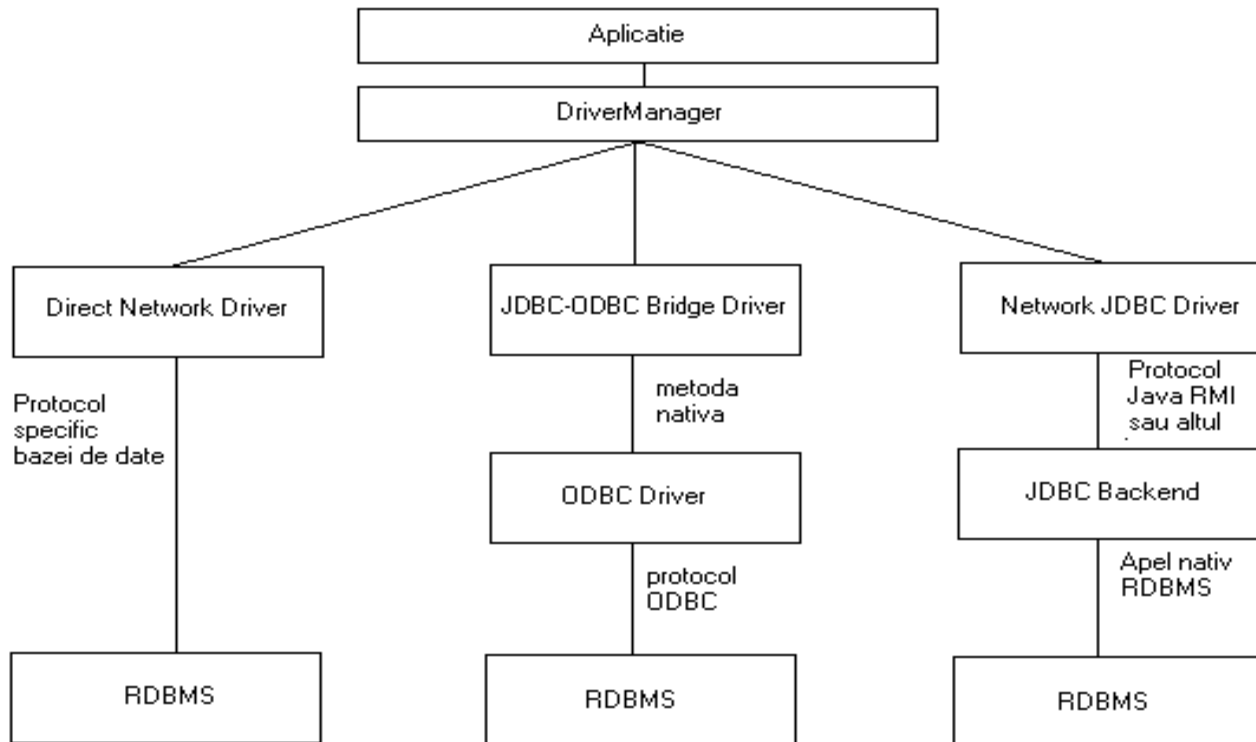
Java Database Connectivity

- Introducere
- Drvere JDBC
- Accesarea unei baze de date folosind JDBC
 - Inregistrarea driver-ului JDBC
 - Stabilirea conexiunii catre baza de date
 - Executia unei instructiuni SQL
 - Procesarea rezultatelor
 - Inchiderea conexiunii catre baza de date
- Relatia dintre tipuri SQL si tipuri Java
- Exemplu de utilizare

Introducere

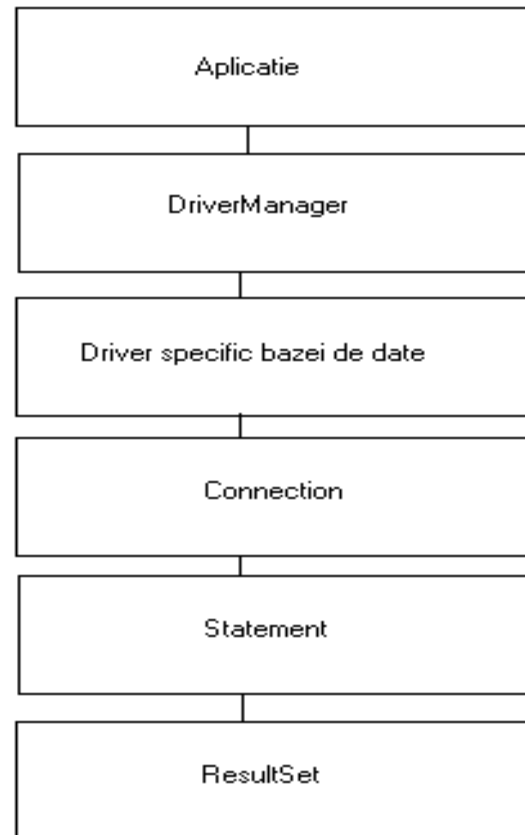
- JDBC (Java DataBase Connectivity) este o interfata standard SQL de acces a bazelor de date. Acesta ne furnizeaza un acces uniform in baze de date relationale. JDBC este constituit dintr-un set de clase si interfete scrise în Java, furnizând un API standard pentru proiectantii de aplicatii baze de date.
- Folosind JDBC este usor sa transmitem secvente SQL catre baze de date relationale.
- În linii mari JDBC face trei lucruri:
 - stabileste o conexiune cu o baza de date;
 - trimite secvente SQL;
 - prelucreaza rezultatele.

Driveri JDBC





Accesarea unei baze de date folosind JDBC





Accesarea unei baze de date folosind JDBC

- **Inregistrarea driver-ului JDBC**

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

- **Stabilirea conexiunii catre baza de date**

```
jdbc<subprotocol>:<nume>
```

```
Connection con=DriverManager.getConnection ("jdbc:odbc:myDB", user,  
password);
```

- **Executia unei instructiuni SQL**

```
ResultSet rs= instructiune.executeQuery ("select * from myDB");
```

```
String sql="insert into myMessages values ("Popa", "Ion", "Deva")";
```

```
int raspuns=instructiune.executeUpdate(sql);
```

- **Procesarea rezultatelor**

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

- **Inchiderea conexiunii catre baza de date**

```
close()
```



Relatia dintre tipuri SQL si tipuri Java

Tip SQL	Tip JAVA	Metoda
CHAR	String	getString()
VARCHAR	String	getString()
LONGVARCHAR	String	getString()
NUMERIC	java.math.BigDecimal	getBigDecimal()
DECIMAL	java.math.BigDecimal	getBigDecimal()
BIT	boolean	getBoolean()
TINYINT	byte	getByte()
SMALLINT	short	getShort()
INTEGER	int	getInt()
BIGINT	long	getLong()
REAL	float	getFloat()
DOUBLE	double	getDouble()
BINARY	byte[]	getBytes()
VARBINARY	byte[]	getBytes()
LONGVARBINARY	byte[]	getBytes()
DATE	java.sql.Date	getDate()
TIME	java.sql.Time	getTime()
TIMESTAMP	java.sql.Timestamp	getTimestamp()



Exemplu de utilizare

- Vom crea o bază de date **Access/SQLServer** numită **StudentiDB**.
- Sursa de date corespunzătoare acestei baze de date se va numi tot **StudentiDB**.
- Se vor prezenta câteva programe Java foarte simple care:
 - creează un tabel **studenti** pentru baza noastră de date,
 - listează informațiile conținute în acest tabel,
 - inserează o înregistrare în tabel.