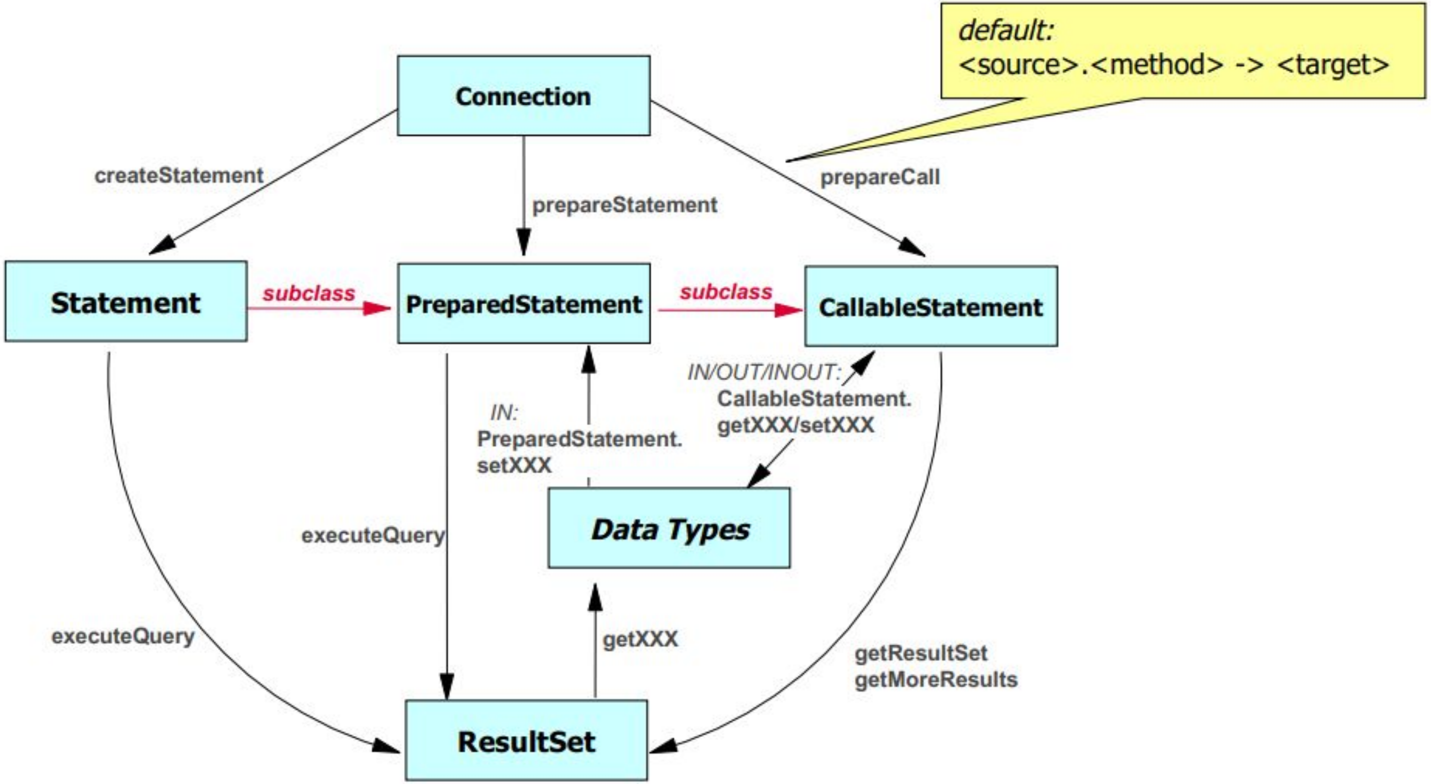
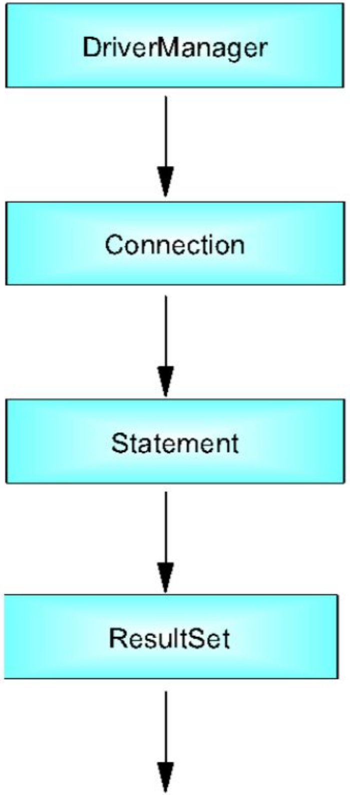
JDBC

ResultSet
PreparedStatement
Connection
JDBC
DataSource
Statement

JDBC - Core interfaces



JDBC Example



```
String url = "jdbc:db2:mydatabase";
...
Connection con = DriverManager.getConnection(url, "dessloch", "pass");

String sqlstr = "SELECT * FROM Employees WHERE dept = 1234";
Statement stmt = con.createStatement( );

ResultSet rs = stmt.executeQuery(sqlstr);

while (rs.next() ) {
      String a = rs.getString(1);
      String str = rs.getString(2);
      System.out.print(" empno= " + a);
      System.out.print(" firstname= " + str);
      System.out.print("\n");
}
```

JDBC Advantage

- Simple and Easy Database Access
- Standardized API and SQL allows for talking to DBs of different vendors
- Vendors provide native JDBC drivers

JDBC Disadvantage

- Error prone
  - Potential for Memory Leak
  - Unsafe code
- Verbose code
- Poor Exception handling
- Complicated Conversion from Result to Object
- Coupling DB model to Code

ResultSet

Connection

JDBC

DataSource

Statement

PreparedStatement