Swing ve JDBC™ ile Database Erişimi

- JDBC API, tablolanmış herhangi bir tür veriye, özellikle İlişkisel Veritabanı, erişim sağlayan bir Java API'sidir.
- JDBC, aşağıda verilen üç etkinliğin gerçekleştirilebileceği java uygulamaları oluşturulabilmesine olanak tanır:
  - Veritabanı gibi bir veri kaynağına bağlanmak
  - Veritabanına sorgu ve güncelleme ifadeleri göndermek
  - Sorgu sonucu veritabanından alınan sonuçları işlemek

- Veri tabanı bir tür organize edilmiş veri koleksiyonudur. Veriler çeşitli şekillerde organize edilebilir ve kullanılabilir.

- Günümüzdeki en yaygın uygulama ilişkisel veri tabanları denilen sistemlerdir bu sistemler neredeyse tamamen Sequential Querry Language-Ardışık sorgulama dili denilen bir dil üzerinden programlanır ve kullanılırlar.

- Kısaca SQL diye anılan bu dil çok basit sorgulama deyimleriyle bir veri dizininden istenilen alt dizilere ulaşma olasılığı sağlar.

- Piyasada çeşitli şirketlerin geliştirdiği ve herbiri birbirinden biraz farklı SQL dilleri mevcuttur.

- Bunlar arasında en popülerleri olarak Microsoft SQL server, Microsoft Office Access, Oracle, Sybase, IBM DB2, MySQL SQLite, Java DB sayılabilir.

Bu işlemi nasıl gerçekleştireceğimiz gösteren bir kod parçacığı verilmiştir:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ConnectingDatabase {

   public ConnectingDatabase() throws SQLException {

      Connection conTest = DriverManager.getConnection(
                  "jdbc:myDriver:wombat", "myLogin", "my-Password");

      Statement stTestQuery = conTest.createStatement();

      ResultSet rsTestResult = stTestQuery.executeQuery(
                     "SELECT a, b, c FROM Table1");

      while (rsTestResult.next()) {

         int iCol1Val = rsTestResult.getInt("a");
         String strCol2Val = rsTestResult.getString("b");
         float fCol3Val = rsTestResult.getFloat("c");

      }

   }

}
```

JDBC Bileşenleri

JDBC aşağıda verilen dört bileşeni içerir:

JDBC API

JDBC™ API Java™ programlama dili kullanılarak ilişkili veriye erişim sağlar. JDBC API kullanarak, SQL ifadelerini gerçekleştirebilen, sonuçları alabilen ve değişiklikleri yeniden veri kaynağına geri gönderebilen Java uygulamaları geliştirilebilir. JDBC API aynı zamanda farklı ya da ayrı çevrelerdeki veri kaynaklarına tepki verebilir.

JDBC API, the Java™ Standard Edition (Java™ SE) ve the Java™ Enterprise Edition (Java™ EE)'i içeren Java Platformunun bir parçasıdır. JDBC 4.0 API iki pakete bölünmüştür: `java.sql ve javax.sql`. Her iki pakette Java SE ve Java EE'de içerilir.

JDBC Driver Manager

JDBC DriverManager sınıfı bir Java uygulamasını bir JDBC sürücüsüne bağlayan nesneler tanımlar. DriverManager geleneksel olarak JDBC'nin omurgasını oluşturur.

Standard Extension paketleri `javax.namig` ve `javax.sql`, bir veri kaynağla bağlantı kurmak için Java Naming and Directory Interface™ (JNDI) adlandırma servisine kaydedilmiş bir `DataSource` nesnesi sağlar. Her iki bağlantı biçimi de kullanılabilir.

JDBC Test Suite

JDBC deneme takımı, JDBC sürücülerinin uygulamanızda çalışır olup olmadığının belirlenmesini sağlar. Bu denemeler kapsamlı ya da ayrıntılı değildir; ancak JDBC API içerisindeki birçok önemli özelliği deneme olanağı tanır.

JDBC-ODBC Bridge

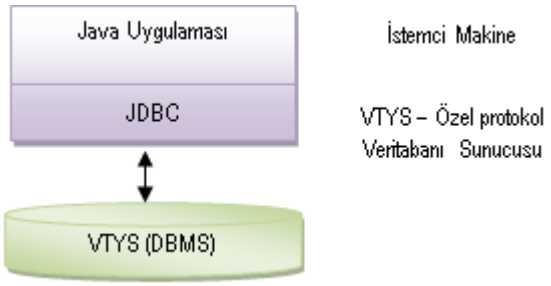
Java Yazılımı köprüsü ODBC sürücüleri aracılığıyla JDBC erişimi sağlar. Ancak ODBC sürücülerinin, bu sürücüyü kullanan her bir istemci makinede çağrılması gerektiği unutulmamalıdır. Bu yüzden ODBC sürücüsü en çok, istemci kurulumlarının çok fazla sorun olmadığı küçük şirket ağlarında uygundur.

JDBC Mimarisi

İki ve Üç Aşamalı İşlem Modeli

JDBC API veritabanı erişimi için, hem iki hem de üç aşamalı işlem modelini destekler.

Veri Erişimi İçin İki aşamalı Mimari



- Veri tabanı uygulaması için NetBeans + JDK ve MySQL kurulu olması gerekmektedir.

- Veritabanı uygulamaları son kullanıcı için bir görsellik ve kullanması için bir yol göstermediği için veritabanı uygulamaları sadece veritabanı olarak kalamaz.

- Bu nedenle son kullanıcının anlayabileceği bir şekle ve görselliğe dönüştürmesi gerekmektedir. Eğer bu yapılmazsa veri bütünlüğünü sağlanması zorlaşır ve son kullanıcıya zorluk çıkabilir.

Veri tabanı uygulaması 3n katman mimarisi gerçekleştirilmektedir. Bunlar:

Sunum Katmanı : Bu son kullanıcıya açık olan katmandır. Bilgileri düzenli bir şekilde göstermeye ve düzgün, kurallara uygun olarak veri girilmesini bu katman sayesinde yapılmaktadır. Kullanıcıya yol gösterir ve bazı sınırlamalar getirir. Örneğin; Forum sitesine kullanıcı sadece oy ve yorum yazabilir fakat yeni bir başlık ekleyemez bunu site adminin yapması gerekir.

İş Katmanı : Başlıca amacı sunum katmanı ile Veri katmanı arasındaki bağlantıyı sağlamaktır. Yapılan iş ile ilgili şeyler bu katmanda yapılır. Örneğin; forum sitesine bir konu eklenebilir. Bu eklemeyi denetleyen ve kurallara uygunsa veri tabanına yazan bu katmandır. Yapılacak işler, kontroller, izinler bu katman aracılığı ile yapılır.

Veri Katmanı : Veritabanının kendisidir. Uygulama için gerekli veriler burada tutulur ve istendiği taktirde iş katmanı aracılığı ile sunum katmanına aktarılır.