

JDBC , JDBC TEMPLATE ve HIBERNATE KARŞILAŞTIRMASI

JDBC

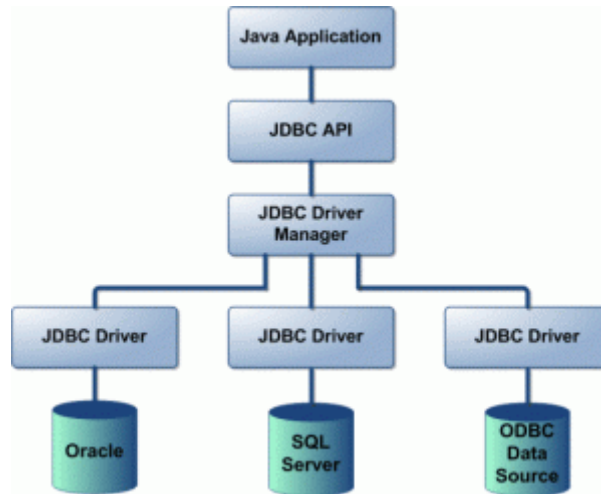
Verilerimizi veritabanında sakladığımızda kod tarafında birtakım işlemler yapmamız gerekir. Bu verileri yönetmek için CRUD adını verdiğimiz Create, Update, Delete işlemlerine ihtiyacımız vardır. JDBC tam da bu noktada bizlere yardımcı olur.

JDBC ilişkisel veritabanına erişmek ve verisel işlemleri gerçekleştirmek için yazılımımız ve veritabanımız arasında bir köprü görevi görür.

Herhangi bir Java uygulamasında (Masaüstü uygulaması, FX, JSP, JSF) fark etmeksizin bize katkı sağlamaya devam eder.

JDBC Driver Manager ile istediğimiz veritabanına bağlanarak sorgularımızı çalıştırabilir ve bizlere dönen sonuç üzerinde yazılımımıza kaldığımız yerden devam edebiliriz.

En büyük handikaplarından biri her işlem gerçekleştirdiğimizde bir Connection açıp işlemimiz bittiğinde Connectionlarımızı kapatmamızdır. Projemizin ölçeği biraz büyüdüğünde bu durumların kontrolü zorlaşmakta ve bazen hatalara sebebiyet vermektedir.



JDBC TEMPLATE

JDBC ‘den ve bize çıkardığı bazı problemlerden yukarıda bahsetmiştik. JDBC Template bizlere JDBC’nin her işlem gerçekleştiren Connection açıp, işlemimiz bittiğinde Connection’u kapatma durumundan kurtarıyor.

Bu durumdan kaynaklanan hataları ortadan kaldırarak. CRUD işlemlerini daha pürüzsüz hale getiriyor. Bizler için önemli olan odak noktasını kaybetmeden Entitylerimiz üzerinde işlem yapmamıza fayda sağlıyor.

Veritabanımıza query() metodunu kullanarak ulaştığımızda verilerimizi alabilmek için org.springframework.jdbc.core.RowMapper<Parse_edilecek_sınıf_türü> böyle bir sınıf oluşturulmuş. rowMapper sınıfımızı verdiğimizde query içinde dönen sonucu otomatik olarak ilgili sınıfa mapler ve bizi çok büyük bir dertten kurtarır.

HIBERNATE

Java geliştiriciler için bir ORM aracı olan Hibernate, yazılımcının veritabanı ile yaptığı işlemlerde bir çok kolaylık sağlar ve kurulan yapıyı sağlamlaştırır.

Nesne tabanlı alan modelini geleneksel bir ilişkisel veritabanına eşlemek için bir çerçeve sağlar. Ayrıca, nesne ilişkisel empedans uyumsuzluğu gibi sorunlarını da çözmektedir.

Eğer karşılaştırma yapmak gerekirse JDBC veritabanına nasıl erişebileceğimizi tanımlayan API’dir. SQL kullanarak sorgular oluşturur ve veritabanı ile ilişkisel bir bağlantı kurmaya olanak sağlar. Hibernate ise nesne tabanlı modellerimizi veritabanına eşlemek için kullanılan ORM kütüphanesidir. Veritabanından bağımsızdır. Aynı kod üzerinde birkaç ufak değişiklik yapılarak farklı veritabanları ile de çalışabilmektedir. JDBC ile ilişkisel tablolar kurmak oldukça zordur fakat Hibernate ile One to one, one to many, many to many gibi ilişkiler Anotasyonlar sayesinde kolayca kurulabilir.

