BATCH	:	BATCH
LESSON	:	MENTORING
DATE	:	00.00.2023
SUBJECT	:	QA-Team

techproeducation

techproeducation

techproeducation

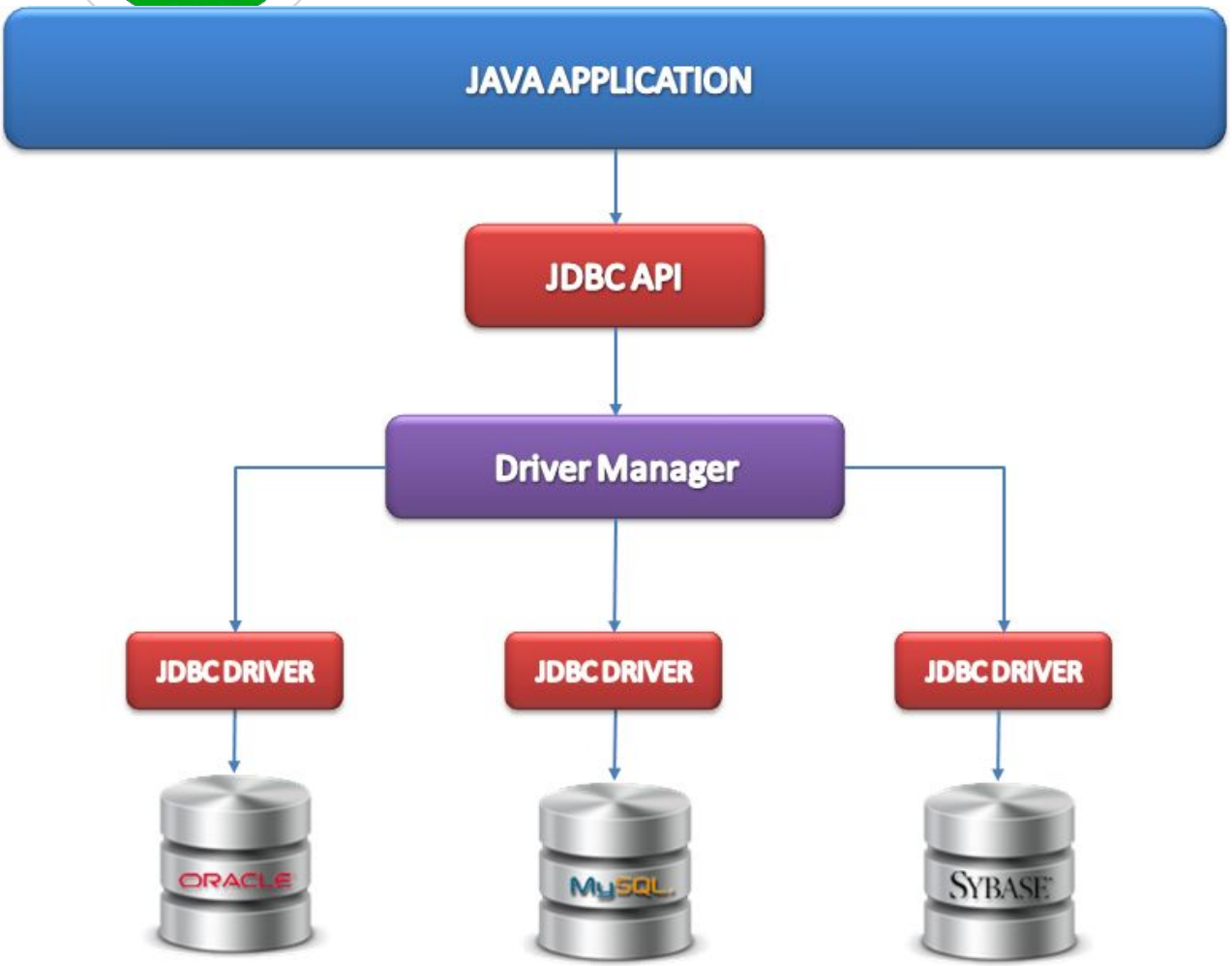
techproeducation

techproedu

Hosgeldiniz…

- Java Database Connectivity (JDBC), Java programlama dilinde yazılmış uygulamaları (application) veritabanı (database) ile etkileşime girmesini sağlayan (bağlayan) bir uygulama programlama ara yüzüdür (API'dir).
- JDBC ile hemen hemen tüm ilişkisel veri tabanı yönetim sistemlerine SQL sorgusu gönderilebilmektedir

JDBC Mimarisi 2 katmana ayrılır;



1. **JDBC API:** Uygulama ve veritabanı arasında bağlantı sağlar.

2. **JDBC Driver API:** Uygulama ve kullanılan veritabanı sürücü bağlantısını destekler. Aşağıdaki diyagram bu yapının yerini göstermektedir.

Neden JDBC kullanılır?

- Direk veri tabanına (database) bağlanıp veri (data) almak için kullanılır. Bunun için;

- 1. Database'e bağlanılır
- 2. Query gönderip data alınır.
- 3. Bu dataları test caselerde kullanıp assert yapılır.

Database Testing Nasıl yapılır?

UI'daki (Front End) ve Database (Back End) deki değerlerin aynı olup olmadığı database testing ile yapılır. Örneğin UI dan girilen dataların okunması ve bu dataların doğrulanması (Assertion).

Database testi, UI (User Interface) testinden daha hızlı. Çünkü sayfa yükleme, veri getirme gibi beklemeleri yapmıyor.

Mülakatta gelirse;

Eklediğim verilerin (data) veri tabanına (database) eklenip eklenmediğini, UI'da yaptığım işlemlerin veri tabanında (database) gerçekleşip gerçekleşmediğini test etmek için direk veri tabanına bağlanarak JDBC kullandım.

Örnek-1; kayıt formuna girilen kullanıcı adı ve şifresi database de oluşmuş mu?

Örnek-2; database den girilen bir datanın UI da bulunup bulunmadığını test için,

Örnek-3; UI da yapılan değişikliğin database de olup olmadığı (yazıların değişmesi gibi).

JDBC Bileşenleri

1. DriverManager:
2. Driver
3. Connection
4. Statement
5. resultSet
6. SQLException

JDBC kullanarak bir SQL Statement'ı ile işlem yapabilmek için aşağıdaki adımlar izlenmelidir:

- 1- Bir Connection elde edilmelidir,
- 2- Bir Statement yaratılmalıdır,
- 3- Sorgu çalıştırılmalıdır,
- 4- ResultSet objesi işlenmelidir,
- 5- Connection kapatılmalıdır.

JDBC kullanarak bir SQL Statement'ı ile işlem adımları:

1- Bir Connection elde edilmelidir :

Öncelikle üzerinde işlem yapılacak kaynağa (DB, File System vs) erişim sağlanmalı ve bir Connection objesi elde edilmelidir. Bir önceki yazıda Connection'ın nasıl sağlanacağı anlatılmıştır.

JDBC kullanarak bir SQL Statement'ı ile işlem adımları:

- **2- Bir Statement yaratılmalıdır :** Statement, SQL Statement'ı temsil eden bir Interface'tir. Statement objelerini çalıştırdığımızda, veritabanı sonuç kümesini temsil eden bir data tablosu olan ResultSet objelerini elde ederiz. Bir Statement objesi elde edebilmek için bir Connection objesine ihtiyaç duyarız.

JDBC kullanarak bir SQL Statement'ı ile işlem adımları:

- Üç çeşit Statement bulunur :

1. Statement : Parametresi olmayan basit SQL Statement'larını implemente etmek için kullanılır.

2. PreparedStatement : (Statement sınıfından extend eder) input parametrelerine sahip olan (sahip olma ihtimali olan) SQL Statement'larını implemente etmek için kullanılır.

3. CallableStatement : (PreparedStatement sınıfından extend eder) Hem input hem output parametrelerine sahip olma ihtimali olan stored procedure'leri çalıştırmak için kullanılır.

JDBC kullanarak bir SQL Statement'ı ile işlem adımları:

- **3- Sorgu çalıştırılmalıdır :** Bir sorgu çalıştırmak için Statement'ta bulunan execute() metodlarından biri çağrılmalıdır. Bu metodlar aşağıdaki gibidir:
- execute() : Query'nin döndüğü ilk obje bir ResultSet objesi ise eğer, true döner. Eğer sorgu bir veya daha fazla ResultSet dönecekse bu metod kullanılmalıdır. Dönen ResultSet objelerini tekrarlamalı bir şekilde almak gerekir.
- executeQuery() : Tek bir ResultSet objesi döner.
- executeUpdate() : SQL Statement'ın etkilemiş olduğu satır sayısını integer cinsinden döner. INSERT, DELETE veya UPDATE SQL Statement'ları kullanılacaksa bu metod kullanılmalıdır.

JDBC kullanarak bir SQL Statement'ı ile işlem adımları:

- **4- ResultSet objesi işlenmelidir :** ResultSet içindeki dataya bir *cursor* yardımıyla erişmek gerekir. Bu cursor bir database cursor değildir. Bu cursor, ResultSet içindeki satırları işaret eden bir pointer'dır. Başlangıçta cursor, ResultSet objesindeki ilk satırın öncesini işaret eder durumdadır. ResultSet'in ilgili metodları çağrılarak cursor hareket ettirilir.

JDBC kullanarak bir SQL Statement'ı ile işlem adımları:

- **5- Connection kapatılmalıdır :** Statement ile işimiz bittiğinde Statement.close() metodunu çağırarak hemen connection'ı kapatmamız gerekmektedir. Böylece Statement'ın tüketmiş olduğu kaynakları serbest bırakmış oluruz. Bu metodu çağırdığımızda ResultSet objelerini de kapatmış oluruz.

- JDBC kurulumunu DevOps yapar, bizimde yapmamız gerekebilir. Veri tabanına bağlanıp, verileri okuyacağız