



# JDBC QUESTION BANK

2023

# Önsöz

## JDBC Nedir?

Java Veritabanı Bağlantısı olarak bilinen JDBC'nin dünyasına hoş geldiniz. Günümüz dijital çağında, verinin hemen hemen her yaşam alanında kritik bir rol oynadığı bir ortamda, veritabanlarıyla etkileşimde bulunmak ve bilgileri verimli bir şekilde yönetmek önemlidir. Yazılım geliştirici, veri analisti veya IT profesyoneli olmanız fark etmeksizin, veritabanlarıyla veri bağlantısı kurma, veri çekme, manipülasyon yapma ve depolama gibi becerilere sahip olmak önemlidir.

JDBC, Java programlama dilinin ve çeşitli veritabanı yönetim sistemleri arasında köprü görevi görerek sorunsuz iletişim ve etkileşim sağlar. Bu kitapta, JDBC'nin temellerini, mimarisini ve dahil olan temel bileşenleri keşfedeceğiz. JDBC'nin sezgisel ve güçlü API'sini kullanarak bağlantı kurmayı, sorguları çalıştırmayı ve veritabanlarından veri çekmeyi öğreneceksiniz.

JDBC'nin dünyasına doğru bir yolculuğa başlayalım!

## JDBC'yi öğrenmek neden önemlidir?

JDBC'yi öğrenmek, Java uygulamaları ile veritabanları arasında sorunsuz etkileşim sağlayarak verilerin etkin bir şekilde depolanmasını, alınmasını ve manipüle edilmesini mümkün kılar. JDBC ile geliştiriciler, veritabanlarının gücünden yararlanarak sağlam ve dinamik uygulamalar oluşturabilir, bu da günümüz veri odaklı dünyasında temel bir beceri haline gelir.

## Kitap hakkında

Bir JDBC soru bankası, JDBC kavramları ve işlevselliği hakkındaki bilginizi test etmek ve değerlendirmek için özel olarak tasarlanmış soruların bir koleksiyonunu sunar. Bu JDBC soru bankası, mülakatlara, sınavlara hazırlanan veya JDBC konusundaki anlayışını derinleştirmek isteyen bireyler için değerli bir kaynak olarak hizmet eder. JDBC soru bankasıyla pratik yaparak, geliştiriciler JDBC'deki yeteneklerini geliştirebilir ve veritabanlarıyla etkin bir şekilde çalışma konusunda güven kazanabilirler. Sadece başlamakta olanlar veya bilgilerini bir üst seviyeye taşımak isteyenler için ideal bir kaynaktır.



**TechPro Education LLC.**

**JDBC SORU BANKSAI**

Copyright © TechPro Education LLC, 2022

All publication rights of this work belong to TechPro Education LLC.

All right reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system without permission in writing from TeachPro Education.

## İÇİNDEKİLER

JDBC	1
JDBC & ODBC	4
JDBC Driver (Sürücüsü)	5
Connection (Bağlantı)	6
Transaction (İşlem)	9
Execute()	10
Create Table	12
Execute Query()	15
CRUD	18
LIKE (Benzeri Arama)	27
Prepare Statements	28
Mülakat Sorusu	30
Cevab Anahtari	33

## YAZAR HAKKINDA

Mohammad Frotan, çok yönlü bir uzmandır ve bilgisayar bilimi alanında uzmanlaşmıştır. Tam yığın ve mobil geliştirme konularında uzmanlığa sahiptir. Lise düzeyinde üç yıl ve backend eğitmeni olarak iki yıl tecrübesi bulunmaktadır. Kariyeri boyunca üç proje başarıyla tamamlamış ve bu süreçte becerilerini geliştirmiştir.

## 1. JDBC kısaltması neyi ifade eder?

- A) Java Veritabanı Bağlantısı
- B) Java Veritabanı Kontrolü
- C) Java Veri Bağlama ve Dönüştürme
- D) Java Veritabanı Bağlantısı

## 2. Aşağıdakilerden hangisi JDBC hakkında doğrudur?

- A) Bir programlama dili olarak kullanılır.
- B) Bir veritabanı yönetim sistemi olarak kullanılır.
- C) Java uygulamalarını veritabanlarına bağlamak için kullanılan bir API'dir.
- D) Verileri internet üzerinden iletmek için bir protokoldür.

## 3. DBC'nin hangi bileşeni, bir veritabanına bağlantı kurmayı sağlamaktan sorumludur?

- A) Connection
- B) Statement
- C) ResultSet
- D) DriverManager

## 4. JDBC'deki ResultSet arabiriminin rolü nedir?

- A) Bir veritabanıyla bağlantıyı temsil eder.
- B) Bir SQL ifadesini temsil eder.
- C) Bir veritabanı sorgusunun sonuçlarını temsil eder.
- D) Bir veritabanı meta verisini temsil eder.

## 5. JDBC kullanarak bir veritabanına bağlantı kurmak için hangi yöntem kullanılır?

- A) getConnection()
- B) executeQuery()
- C) prepareStatement()
- D) close()

**6. JDBC'de SQL sorgularını yürütmek ve sonuçları döndürmekten sorumlu olan arayüz hangisidir?**

- A) Connection
- B) Statement
- C) ResultSet
- D) PreparedStatement

**7. Which method is used to execute a parameterized SQL statement in JDBC?**

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) preparedStatement()

**8. JDBC'de bir ResultSet'ten veri almak için hangi yöntem kullanılır?**

- A) getData()
- B) getResult()
- C) next()
- D) fetch()

**9. JDBC'de depolanan prosedürlerle çalışmak için hangi arayüz kullanılır?**

- A) Statement
- B) ResultSet
- C) CallableStatement
- D) Connection

**10. JDBC'deki close() yönteminin amacı nedir?**

- A) To close the database connection.
- B) To close the ResultSet object.
- C) To release the resources used by JDBC objects.
- D) All of the above.

**11. Java uygulamalarında neden JDBC kullanmalıyız?**

- A) Uygulamaya grafik kullanıcı arayüzü sağlamak için.
- B) Veritabanlarıyla iletişim ve etkileşim sağlamak için.
- C) Uygulamanın performansını artırmak için.
- D) Verilerden raporlar ve analizler oluşturmak için.

**12. JDBC kullanmanın temel avantajı nedir?**

- A) Uygulama geliştirme sürecinde veritabanlarına olan ihtiyacı ortadan kaldırır.
- B) Çeşitli veritabanı yönetim sistemleriyle çalışmak için standart bir yol sağlar.
- C) Veritabanı işlemleri için gereken kod miktarını azaltır.
- D) Uygulama verilerinin güvenliğini artırır.

**13. ODBC kısaltması neyi ifade eder?**

- A) Open Database Connectivity
- B) Object Database Connector
- C) Oracle Database Control
- D) Online Data Backup and Conversion

**14. Aşağıdakilerden hangisi ODBC API hakkında doğrudur?**

- A) Bir programlama dili.
- B) Bir veritabanı yönetim sistemi.
- C) Verilerin internet üzerinden iletilmesi için bir protokol.
- D) Uygulamaları veritabanlarına bağlamak için bir API.

**15. ODBC API'nin hangi bileşeni veritabanına bağlantı kurma işlemini gerçekleştirir?**

- A) ODBC Manager
- B) ODBC Driver
- C) ODBC Data Source
- D) ODBC Cursor



**16. ODBC Sürücüsünün rolünü en iyi şekilde açıklayan ifade aşağıdakilerden hangisidir?**

- A) ODBC API çağrılarını DBMS'e özgü bir ağ protokolüne çevirir.
- B) Veritabanı işlemleri için grafiksel kullanıcı arayüzü sağlar.
- C) Veritabanında verilerin depolanmasını ve geri alınmasını yönetir.
- D) SQL sorgularını DBMS bağımsız bir ağ protokolüne dönüştürür.

**17. Java uygulamalarında JDBC kullanmanın nedenleri nelerdir?**

- A) Uygulama geliştirmede veritabanlarına ihtiyacı ortadan kaldırmak için.
- B) Çeşitli veritabanı yönetim sistemleriyle çalışmak için standart bir yol sağlamak için.
- C) Veritabanı işlemleri için gereken kod miktarını azaltmak için.
- D) Uygulama verilerinin güvenliğini artırmak için.

**18. JDBC ve ODBC arasındaki farkı doğru bir şekilde açıklayan ifade hangisidir?**

- A) JDBC, çeşitli programlama dillerini veritabanlarına bağlamak için kullanılırken, ODBC, Java uygulamalarını veritabanlarına bağlamak için kullanılır.
- B) JDBC, istemci makinede yerel veritabanı istemci kitaplıklarının yüklü olmasını gerektirirken, ODBC bu gerekliliği taşımaz.
- C) JDBC, DBMS bağımsız bir ağ protokolü sağlarken, ODBC API çağrılarını DBMS'e özgü bir ağ protokolüne çevirir.
- D) JDBC, Oracle tarafından geliştirilen açık bir standartken, ODBC Microsoft tarafından geliştirilen tescilli bir API'dir.

**19. JDBC ve ODBC sürücüleri arasındaki temel fark nedir?**

- A) JDBC sürücüleri platform bağımsızdır, ODBC sürücüleri ise platforma özgüdür.
- B) JDBC sürücüleri farklı veritabanı yönetim sistemleriyle çalışmak için standart bir yol sağlarken, ODBC sürücüleri API çağrılarını DBMS'e özgü bir ağ protokolüne çevirir.
- C) JDBC sürücüleri genellikle Windows ortamlarında yaygın olarak kullanılırken, ODBC sürücüleri öncelikle Linux ortamlarında kullanılır.
- D) JDBC sürücüleri ek kurulum ve yapılandırmaya ihtiyaç duyarken, ODBC sürücüleri herhangi bir ek adım olmadan kurulabilir.

## 20. JDBC sürücüsü nedir?

- A) JDBC API kullanarak Java uygulamalarının veritabanlarıyla iletişim kurmasını sağlayan bir yazılım bileşeni.
- B) Java uygulamalarında veritabanlarını yönetmek için kullanılan grafiksel kullanıcı arayüzü.
- C) Özel olarak Java uygulamaları için tasarlanmış bir veritabanı yönetim sistemi.
- D) Java uygulamalarında veritabanı sorgularını yazmak için kullanılan bir programlama dili.

## 21. JDBC sürücüsü, Java uygulamalarında hangi rolü oynar?

- A) Kullanıcıların veritabanıyla etkileşimde bulunabileceği bir görsel arayüz sağlar.
- B) JDBC API çağrılarını veritabanı yönetim sistemi tarafından anlaşılabilir bir formata çevirir.
- C) Veritabanı içindeki verilerin depolanması ve alınması işlemlerini yönetir.
- D) Veritabanında depolanan verilere dayalı raporlar ve analitikler oluşturur.

## 22. Java uygulamalarında JDBC sürücüsü neden gereklidir?

- A) Uygulamadaki veritabanı işlemlerinin performansını artırır.
- B) Uygulama ile veritabanı arasında güvenli bir bağlantı sağlar.
- C) Uygulamanın SQL sorgularını göndermesine, veri almasına ve veritabanı işlemleri yapmasına olanak tanır.
- D) Java uygulamalarında SQL sorgularına ihtiyacı ortadan kaldırır.

## 23. Java uygulamasıyla JDBC bağlantısı kurarken ilk adım nedir?

- A) Gerekli Java sınıflarını içe aktarmak
- B) JDBC sürücüsünü yüklemek
- C) Bir Connection nesnesi oluşturmak
- D) SQL sorgularını çalıştırmak

## 24. Java uygulamasında JDBC sürücüsünü yüklemekten sorumlu olan sınıf hangisidir?

- A) java.sql.DriverManager
- B) java.sql.Connection
- C) java.sql.ResultSet
- D) java.sql.Statement

**25. JDBC URL'nin amacı nedir?**

- A) Veritabanı sunucusunun konumunu belirtir.
- B) Çalıştırılacak SQL ifadelerini tanımlar.
- C) Veritabanı için kullanıcı adı ve şifreyi depolar.
- D) Kullanılacak belirli veritabanı sürücüsünü tanımlar.

**26. Java'da veritabanına bağlantı kurmak için hangi yöntem kullanılır?**

- A) DriverManager.openConnection()
- B) Connection.connect()
- C) DriverManager.getConnection()
- D) Connection.open()

**27. Bir JDBC bağlantısı kurmak için ne gereklidir?**

- A) Sadece JDBC sürücüsünün yüklenmesi yeterlidir.
- B) Sadece veritabanı adının sağlanması yeterlidir.
- C) JDBC URL'si, kullanıcı adı ve şifrenin doğru bir şekilde belirtilmesi gereklidir.
- D) Doğru veritabanı şemasının seçildiğinden emin olunması gereklidir.

**28. SQL sorgularını çalıştırmak için bir Statement nesnesi oluşturmak için hangi yöntem kullanılır?**

- A) Connection.createStatement()
- B) Statement.executeQuery()
- C) Statement.create()
- D) Connection.createStatement()

**29. DBC'de Statement nesnesinin amacı nedir?**

- A) Veritabanına bağlantı kurar.
- B) Çalıştırılan bir SQL sorgusunun sonucunu temsil eder.
- C) Veritabanında SQL sorgularını ve güncellemeleri çalıştırır.
- D) Veritabanı nesnelerinin meta verilerini alır.

**30. JDBC bağlantı kurulumu sırasında oluşabilecek istisnaları nasıl yönetirsiniz?**

- A) try-catch bloklarını kullanarak
- B) İstisnaları görmezden gelerek
- C) Uygulamayı sonlandırmak için System.exit() kullanarak
- D) JDBC sürücüsü tarafından otomatik hata yönetimine güvenerek

**31. JDBC bağlantısını kapatmak için önerilen yaklaşım nedir?**

- A) Statement nesnesini kapatmak
- B) Connection nesnesini null olarak atamak
- C) Connection.close() yöntemini çağırmak
- D) Ağı kesmek (disconnect)

**32. Class.forName() yönteminin Java'daki amacı nedir?**

- A) Bir sınıfın yeni bir örneğini oluşturur.
- B) JDBC sürücü sınıfını yükler ve kaydeder.
- C) Bir sınıfın tam nitelikli adını alır.
- D) Bir sınıf üzerinde yansıma işlemleri yapar.

**33. Aşağıdaki ifadelerden hangisi parametresiz bir statik SQL ifadesini çalıştırmak için kullanılır?**

- A) PreparedStatement
- B) CallableStatement
- C) Statement
- D) ResultSet

**34. DBC'de giriş parametreleri olan bir SQL ifadesini çalıştırmak için hangi ifade kullanılır?**

- A) Statement
- B) PreparedStatement
- C) CallableStatement
- D) ResultSet

**35. Aşağıdaki ifadelerden PreparedStatement ile ilgili hangisi doğrudur?**

- A) Static SQL ifadelerini çalıştırmak için kullanılır.
- B) Giriş parametrelerini kabul edemez.
- C) SQL enjeksiyon saldırılarına karşı koruma sağlar.
- D) Sadece SELECT sorgularını çalıştırabilir.

**36. PreparedStatement, JDBC'de Statement'tan nasıl farklıdır?**

- A) PreparedStatement bir seferde birden fazla sorguyu çalıştırabilirken, Statement yalnızca bir sorguyu bir seferde çalıştırabilir.
- B) PreparedStatement, Statement'a göre daha hızlı performansa sahiptir.
- C) PreparedStatement, giriş parametrelerinin kullanılmasına izin verirken, Statement izin vermez.
- D) PreparedStatement, saklanmış prosedürleri çalıştırmak için uygundur, Statement ise değildir.

**37. Bir PreparedStatement içerisinde giriş parametrelerini ayarlamak için hangi yöntem kullanılır?**

- A) setInputParameter()
- B) setParam()
- C) setParameter()
- D) setXxx()

**38. Performans açısından, PreparedStatement'ın Statement'e göre kullanımının faydası nedir?**

- A) PreparedStatement, Statement'e göre performans açısından daha yavaştır.
- B) PreparedStatement ve Statement benzer performans özelliklerine sahiptir.
- C) PreparedStatement önceden derlenebilir ve yeniden kullanılabilir, bu da performansta iyileşme sağlar.
- D) Performans, PreparedStatement ve Statement seçimi tarafından etkilenmez.

**39. JDBC bağlamında bir işlem (transaction) nedir?**

- A) Veritabanına bağlantı kurma sürecini ifade eder.
- B) Veritabanında gerçekleştirilen tek bir SQL ifadesini temsil eder.
- C) Birleşik bir mantıksal işlem olarak tek seferde gerçekleştirilen birden fazla SQL ifadesinden oluşan bir iş birimidir.
- D) Belirli bir SQL işlemi tarafından etkilenen satır sayısını gösterir.

**40. DBC'de, commit ve rollback gibi işlemle ilgili yöntemleri sağlayan hangi JDBC arayüzü bulunur?**

- A) Connection
- B) Statement
- C) PreparedStatement
- D) ResultSet

**41. DBC'de commit() yönteminin amacı nedir?**

- A) Bir SQL ifadesini çalıştırır ve sonuç kümesini döndürür.
- B) Bir işlem içinde gerçekleştirilen işlemleri iptal eder ve geri alır.
- C) Bir işlem içinde yapılan değişiklikleri kalıcı olarak kaydeder.
- D) Veritabanına bağlantı kurar.

**42. DBC'de rollback() yöntemi ne zaman kullanılmalıdır?**

- A) Bir SQL ifadesi çalıştırılmadan önce.
- B) Her SQL ifadesi çalıştırıldıktan sonra.
- C) Bir işlem başarıyla işlendikten sonra.
- D) Bir işlem içinde hata oluştuğunda.

**43. JDBC'de execute() yöntemi kullanılarak hangi SQL sorguları çalıştırılabilir?**

- A) SELECT \* FROM Customers;
- B) UPDATE Employees SET salary = 50000 WHERE department = 'IT';
- C) INSERT INTO Orders (customer\_id, product\_id) VALUES (1, 100);
- D) Hepsi yukarıdakiler

**44. execute() yöntemi bir SELECT sorgusu yürütüldüğünde dönen değer nasıl yorumlanır?**

- A) Sorgu tarafından etkilenen satır sayısını döner.
- B) Sorgudan elde edilen sonuç kümesini döner.
- C) Sorgu yürütmesinin başarılı olup olmadığını gösteren bir boolean değeri döner.
- D) SELECT sorguları için uygulanmaz.

**45. execute() yönteminin döndürdüğü değer, INSERT veya UPDATE gibi SELECT olmayan sorgular için neyi ifade eder?**

- A) Sorgu tarafından etkilenen satır sayısını döner.
- B) Sorgudan elde edilen sonuç kümesini döner.
- C) Sorgu yürütmesinin başarılı olup olmadığını gösteren bir boolean değeri döner.
- D) SELECT olmayan sorgular için uygulanmaz.

**46. JDBC'de execute() yöntemi çağrıldığında hangi SQL sorgusu yürütülmesi durumunda true değeri döner?**

- A) CREATE TABLE Employees (id INT, name VARCHAR(50), age INT);
- B) SELECT COUNT(\*) FROM Customers;
- C) DROP TABLE Orders;
- D) INSERT INTO Products (name, price) VALUES ('Phone', 999.99);



**47. JDBC'de execute() yöntemini CREATE TABLE, DROP TABLE veya ALTER TABLE gibi DDL (Veri Tanımlama Dili) ifadeleriyle kullanırken dönen değer ne olacaktır?**

- A) true
- B) false
- C) Bu veritabanı yönetim sistemi tarafından belirlenir.
- D) Bu belirli DDL ifadesine bağlıdır.

**48. JDBC'de execute() yöntemini SELECT gibi DQL (Veri Sorgulama Dili) ifadeleriyle kullanırken dönen değer ne olacaktır?**

- A) true
- B) false
- C) Bu veritabanı yönetim sistemi tarafından belirlenir.
- D) Bu belirli DQL ifadesine bağlıdır.

**49. JDBC'deki execute() yöntemi true değer döndürdüğünde ne ifade eder?**

- A) Sorgu yürütmesi başarılı oldu.
- B) Sorgu yürütmesi başarısız oldu.
- C) Sorgudan bir sonuç kümesi elde edildi.
- D) Sorgu belirli bir satır sayısını etkiledi.

**50. Aşağıdaki ifadelerden hangisi DDL (Veri Tanımlama Dili) örneğidir?**

- A) SELECT \* FROM customers;
- B) CREATE TABLE employees (id INT, name VARCHAR(255), age INT);
- C) UPDATE orders SET status = 'Shipped' WHERE id = 123;
- D) INSERT INTO products (name, price) VALUES ('Phone', 999);

**51. Aşağıdaki ifadelerden hangisi DQL (Veri Sorgulama Dili) örneğidir?**

- A) DELETE FROM customers WHERE id = 456;
- B) ALTER TABLE products ADD COLUMN weight DECIMAL(10,2);
- C) SELECT COUNT() FROM orders;
- D) INSERT INTO employees (name, salary) VALUES ('John Doe', 5000);



**52. Belirtilen sütun adlarıyla birlikte "employee" adında bir tablo oluşturan doğru SQL sorgusu hangisidir?**

A)

Sql code

```
CREATE TABLE employee (  
    employee_id INT,  
    employee_name VARCHAR(255),  
    salary DECIMAL  
);
```

B)

Sql code

```
CREATE TABLE employee (  
    employee_id INTEGER,  
    employee_name TEXT,  
    salary NUMERIC(10,2)  
);
```

C)

Sql code

```
CREATE TABLE IF NOT EXISTS employee (  
    employee_id INT PRIMARY KEY,  
    employee_name VARCHAR(255),  
    salary DECIMAL(10,2)  
);
```

D)

Sql code

```
CREATE TABLE emp (  
    emp_id INT,  
    emp_name VARCHAR(255),  
    emp_salary DECIMAL(10,2)  
);
```

Açıklama: Doğru SQL sorgusu (c), belirtilen sütun adlarıyla birlikte "employee" adında bir tablo oluşturur. "CREATE TABLE" ifadesini kullanır, "employee\_id", "employee\_name" ve "salary" sütun adlarını ve bunlara ait veri tiplerini ve boyutlarını içerir. "IF NOT EXISTS" ifadesi, tablonun zaten var olmadığı durumda sadece tabloyu oluşturur. Ek olarak, "employee\_id" sütunu her bir kaydı benzersiz şekilde tanımlamak için birincil anahtar olarak ayarlanmıştır.

**53.** employee tablosuna "city" adında VARCHAR(20) bir sütun ekleyen SQL sorgusu şöyledir:

A)

Sql code

```
ALTER TABLE employee  
ADD COLUMN city VARCHAR(20);
```

B)

Sql code

```
ALTER TABLE employee  
ADD city VARCHAR(20);
```

C)

Sql code

```
ALTER TABLE employee  
ADD COLUMN city VARCHAR;
```

D)

Sql code

```
ALTER TABLE employee  
ADD city VARCHAR;
```

Açıklama: Doğru SQL sorgusu (a), mevcut employee tablosuna "city" adında yeni bir sütun eklemek için "ALTER TABLE" ifadesini ve "ADD COLUMN" ifadesini kullanır. Sütun, VARCHAR(20) veri tipiyle belirtilir ve en fazla 20 karakter depolayabileceğini gösterir.

**54.** Belirtilen sütun adları ve veri tipleriyle "students" adında bir tablo oluşturmak için aşağıdaki SQL sorgusunu kullanabilirsiniz:

A)

Sql code

```
CREATE TABLE students (  
  id INTEGER PRIMARY KEY,  
  name VARCHAR(255),  
  age INT,  
  grade DECIMAL(3,2)  
);
```

B)

Sql code

```
INSERT INTO students (id, name, age, grade)  
VALUES (1, 'John', 18, 85.5);
```

C)

Sql code

```
UPDATE students SET age = 19 WHERE id = 1;
```

D)

Sql code

```
SELECT * FROM students;
```

Açıklama: Doğru SQL sorgusu (a), belirtilen sütun adları ve veri tipleriyle "students" adında bir tablo oluşturmak için "CREATE TABLE" ifadesini kullanır. "id" sütunu için PRIMARY KEY kısıtı olan INTEGER tipi, "name" için VARCHAR(255) tipi, "age" için INT tipi ve "grade" için DECIMAL(3,2) tipi olan sütunlar içerir.

**55. JDBC'deki executeQuery() yönteminin amacı nedir?**

- A) Bir sorgu olmayan SQL ifadesini yürütür.
- B) Bir sorgu SQL ifadesini yürütür ve sonuç kümesini döndürür.
- C) Bir dizi SQL ifadesini yürütür.
- D) Bir saklı procedure'ı yürütür.

**56. JDBC'deki executeQuery() yöntemi hangi veri türünü döndürür?**

- A) int
- B) String
- C) ResultSet
- D) Boolean

**57. JDBC'de executeQuery() yöntemi kullanılırken genellikle hangi türde SQL ifadesi yürütülür?**

- A) UPDATE
- B) DELETE
- C) INSERT
- D) SELECT

**58. Eğer executeQuery() yöntemi, bir sorgu olmayan SQL ifadesini yürütmek için kullanılırsa ne olur?**

- A) Bir istisna (exception) fırlatılır.
- B) Yöntem null değer döner.
- C) Yöntem boş bir ResultSet döner.
- D) İfade başarılı bir şekilde yürütülür, ancak sonuç kümesi döndürülmez.

**59.** Aşağıdakilerden hangisi `executeQuery()` yöntemini doğru bir şekilde kullanan bir örnektir?

- A) `Statement stmt = conn.createStatement("SELECT * FROM customers");  
ResultSet rs = stmt.executeQuery();`
- B) `PreparedStatement pstmt = conn.prepareStatement("INSERT INTO employees VALUES (?, ?)");  
pstmt.executeQuery();`
- C) `CallableStatement cstmt = conn.prepareCall("{CALL getEmployeeDetails(?)}");  
cstmt.executeQuery();`
- D) `Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery("UPDATE products SET price = 10.99");`

**60.** Kod çalıştırıldığında ve veritabanında "employee" tablosu zaten varsa, davranış spesifik olarak yürütülen koda bağlı olacaktır.

- A) `execute()` yöntemi bir istisna (exception) fırlatacaktır.
- B) Mevcut tablo, yeni tablo tanımıyla üzerine yazılacaktır.
- C) `execute()` yöntemi herhangi bir değişiklik yapmadan false değeri döndürecektir.
- D) `execute()` yöntemi, başarılı bir yürütme işlemini gösteren true değerini döndürecektir.

Açıklama: Eğer "employee" tablosu veritabanında zaten mevcutsa, `CREATE TABLE` ifadesini çalıştırmak bir istisna fırlatacaktır. `execute()` yöntemi mevcut bir tabloyu üzerine yazmayacak, bunun yerine tablonun zaten var olduğunu belirten bir istisna fırlatacaktır.

**61.** Kodu çalıştırdıktan sonra "employee" tablosundaki tüm kayıtları almak için hangi SQL ifadesi uygun olur?

- A) `SELECT * FROM employee;`
- B) `GET employee;`
- C) `RETRIEVE FROM employee;`
- D) `FETCH ALL FROM employee`

**62.** JDBC'de bir SELECT sorgusunu yürütmek ve elde edilen verileri almak için hangi yöntem kullanılır?

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeBatch()

**63.** JDBC'deki executeQuery() yönteminin dönüş tipi:

- A) int
- B) boolean
- C) ResultSet
- D) Statement

**64.** JDBC'den elde edilen bir SELECT sorgusunun sonuç kümesi üzerinde döngü kullanarak işlem yapmanın farklı yolları vardır:

- A) for döngüsünü kullanarak.
- B) while döngüsünü kullanarak.
- C) do-while döngüsünü kullanarak.
- D) forEach döngüsünü kullanarak.

**65.** SELECT sorgusunda sonuç kümesini filtrelemek için hangi SQL ifadesi kullanılır?

- A) FILTER BY
- B) HAVING
- C) WHERE
- D) GROUP BY

**66.** ResultSetMetaData kullanarak bir ResultSet'teki toplam sütun sayısını almak için hangi yöntem kullanılır?

- A) getColumnCount()
- B) getColumnLabel()
- C) getColumnName()
- D) getColumnType()

**67.** Veritabanı işlemlerinde aşağıdakilerden hangisi CRUD'un bir parçası değildir?

- A) Create
- B) Read
- C) Update
- D) Delete
- E) All of the above operations are part of CRUD.

**68.** JDBC'de bir INSERT sorgusunu yürütmek için hangi yöntem kullanılır?

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeBatch()

**69.** What is the return type of the executeUpdate() method in JDBC?

- A) int
- B) boolean
- C) ResultSet
- D) Statement

**70.** JDBC'de bir UPDATE sorgusunu nasıl yürütürsünüz?

- A) executeQuery() yöntemini kullanarak.
- B) executeUpdate() yöntemini kullanarak.
- C) execute() yöntemini kullanarak.
- D) executeBatch() yöntemini kullanarak.

**71.** JDBC'de bir DELETE sorgusunu nasıl yürütürsünüz?

- A) executeQuery() yöntemini kullanarak.
- B) executeUpdate() yöntemini kullanarak.
- C) execute() yöntemini kullanarak.
- D) executeBatch() yöntemini kullanarak.

**72.** Prepared bir ifade için parametre değerlerini ayarlamak için hangi yöntem kullanılır INSERT, UPDATE veya DELETE işlemleri için?

- A) setString()
- B) setInt()
- C) setObject()
- D) All of the above

**73.** JDBC'de setAutoCommit() yöntemi ne işe yarar?

- A) Veritabanı işlemleri için otomatik işlem modunu etkinleştirir.
- B) Veritabanı işlemleri için otomatik işlem modunu devre dışı bırakır.
- C) Mevcut işlemi işler.
- D) Mevcut işlemi geri alır.

**74.** JDBC'de birden fazla tablodan ortak bir sütuna dayalı olarak veri almak için hangi SQL ifadesi kullanılır?

- A) SELECT
- B) INSERT
- C) UPDATE
- D) JOIN

**75.** JDBC'de istisnalar nasıl ele alınır?

- A) try-catch blokları kullanarak.
- B) throws ifadesi kullanarak.
- C) hem try-catch blokları hem de throws ifadesi kullanarak.
- D) JDBC tarafından istisnalar otomatik olarak ele alınır.

**76.** JDBC'de bir Statement nesnesini nasıl kapatırsınız?

- A) Statement nesnesi üzerinde close() yöntemini çağırarak.
- B) Connection nesnesi üzerinde close() yöntemini çağırarak.
- C) Statement, Connection kapatıldığında otomatik olarak kapanır.
- D) Statement, sonuç kümesi kapatıldığında otomatik olarak kapanır.



**77. JDBC'de PreparedStatement arayüzünün amacı nedir?**

- A) SQL sorgularını yürütmek ve sonuç kümesi almak için.
- B) Toplu işlemleri verimli bir şekilde gerçekleştirmek için.
- C) Parametrelili SQL ifadelerini yürütmek için.
- D) Veritabanı işlemlerini yönetmek için.

**78. DBC'de bir ResultSet nesnesindeki toplam satır sayısını nasıl alırsınız?**

- A) getRowCount() yöntemini kullanarak.
- B) getTotalRows() yöntemini kullanarak.
- C) getRow() yöntemini kullanarak.
- D) Doğrudan toplam satır sayısını almak mümkün değildir.

**79. "customers" tablosunu id, name ve email sütunlarıyla oluşturmak için hangi SQL sorgusu kullanılır?**

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET name = 'John' WHERE id = 1;
- D) DELETE FROM customers WHERE id = 1;

**80. "customers" tablosuna yeni bir satır eklemek için hangi SQL sorgusu kullanılır?**

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET name = 'John' WHERE id = 1;
- D) DELETE FROM customers WHERE id = 1;

**81.** "customers" tablosundaki id'si 1 olan satırın "name" sütununu güncellemek için hangi SQL sorgusu kullanılır?

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET name = 'John' WHERE id = 1;
- D) DELETE FROM customers WHERE id = 1;

**82.** "customers" tablosundan id'si 1 olan bir satırı silmek için hangi SQL sorgusu kullanılır?

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET name = 'John' WHERE id = 1;
- D) DELETE FROM customers WHERE id = 1;

**83.** "customers" tablosundan tüm satırları seçmek için hangi SQL sorgusu kullanılır?

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET name = 'John' WHERE id = 1;
- D) SELECT \* FROM customers;

**84.** "customers" tablosundan ismi 'John' olan satırları seçmek için hangi SQL sorgusu kullanılır?

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET name = 'John' WHERE id = 1;
- D) SELECT \* FROM customers WHERE name = 'John';

**85.** "customers" tablosundaki tüm satırların "email" sütununu güncellemek için hangi SQL sorgusu kullanılır?

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET email = 'newemail@example.com';
- D) DELETE FROM customers;

**86. "customers" tablosundaki tüm satırları silmek için hangi SQL sorgusu kullanılır?**

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET email = 'newemail@example.com';
- D) DELETE FROM customers;

**87. "customers" tablosundaki "name" sütununun benzersiz değerlerini seçmek için hangi SQL sorgusu kullanılır?**

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET email = 'newemail@example.com';
- D) SELECT DISTINCT name FROM customers;

**88. "customers" tablosundaki satır sayısını saymak için hangi SQL sorgusu kullanılır?**

- A) CREATE TABLE customers (id INT, name VARCHAR(50), email VARCHAR(50));
- B) INSERT INTO customers (id, name, email) VALUES (1, 'John', 'john@example.com');
- C) UPDATE customers SET email = 'newemail@example.com';
- D) SELECT COUNT(\*) FROM customers;

**89. "employees" tablosundan maaşı 5000'den büyük olan satırları seçmek için hangi SQL sorgusu kullanılır?**

- A) SELECT \* FROM employees WHERE salary > 5000;
- B) SELECT \* FROM employees WHERE salary = 5000;
- C) SELECT \* FROM employees WHERE salary < 5000;
- D) SELECT \* FROM employees WHERE salary >= 5000;

**90. "products" tablosundan kategorisi 'Electronics' olan ve fiyatı 1000'den küçük olan satırları seçmek için hangi SQL sorgusu kullanılır?**

- A) SELECT \* FROM products WHERE category = 'Electronics' AND price < 1000;
- B) SELECT \* FROM products WHERE category = 'Electronics' OR price < 1000;
- C) SELECT \* FROM products WHERE category = 'Electronics' AND price > 1000;
- D) SELECT \* FROM products WHERE category = 'Electronics' OR price > 1000;

**91. SQL sorgusundaki OFFSET anahtar kelimesi ne işe yarar?**

- A) Sonuç kümesine başlamadan önce atlanacak satır sayısını belirtir.
- B) Sonuç kümesinden alınacak maksimum satır sayısını belirtir.
- C) Sonuç kümesini geçme notuna göre artan sırayla sıralar.
- D) Sonuç kümesini belirli bir değerden büyük geçme notuna sahip satırlarla filtreler.

**92. "customers" tablosundaki belirli bir kaydı müşterinin ID'sine göre silmek için hangi SQL sorgusu kullanılır?**

- A) DELETE FROM customers WHERE id = 1;
- B) DELETE FROM customers WHERE name = 'John';
- C) DELETE FROM customers WHERE email = 'john@example.com';
- D) DELETE FROM customers WHERE city = 'New York';

**93. Belirli bir kaydı silmek için SQL sorgusunu yürütmek için hangi JDBC yöntemi kullanılır?**

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeDelete()

**94. Bir DELETE sorgusunun WHERE ifadesine koşul eklemenin amacı nedir?**

- A) Silme işlemi için tablo adını belirtmek için.
- B) Tablodan silinecek sütunları belirtmek için.
- C) Silinecek kayıtları seçmek için koşulu belirtmek için.
- D) Tabloya güncellenecek yeni değerleri belirtmek için.

**95. "orders" tablosundan sipariş durumu 'İptal Edildi' olan ve sipariş tarihi belirli bir tarihten eski olan kayıtları silmek için hangi SQL sorgusu kullanılır?**

- A) DELETE FROM orders WHERE order\_status = 'Cancelled' AND order\_date > '2022-01-01';
- B) DELETE FROM orders WHERE order\_status = 'Completed' AND order\_date < '2022-01-01';
- C) DELETE FROM orders WHERE order\_status = 'Shipped' AND order\_date > '2022-01-01';
- D) DELETE FROM orders WHERE order\_status = 'Pending' AND order\_date > '2022-01-01';

**96.** "employees" tablosuna "name" ve "age" sütunları için belirtilen değerlerle yeni bir kayıt eklemek için hangi SQL sorgusu kullanılır?

- A) INSERT INTO employees (id, name, age) VALUES (1, 'John', 30);
- B) INSERT INTO employees (name, age) VALUES ('John', 30);
- C) INSERT INTO employees (name, age) SELECT name, age FROM employees WHERE id = 1;
- D) INSERT INTO employees VALUES ('John', 30);

**97.** Bir tabloya veri eklemek için SQL sorgusunu yürütmek için hangi JDBC yöntemi kullanılır?

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeInsert()

**98.** "products" tablosundaki tüm kayıtları almak için hangi SQL sorgusu kullanılır?

- A) SELECT \* FROM products;
- B) SELECT COUNT(\*) FROM products;
- C) SELECT DISTINCT category FROM products;
- D) SELECT name, price FROM products;

**99.** SQL sorgusunu yürütmek ve sonuç kümesini almak için hangi JDBC yöntemi kullanılır?

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeSelect()

**100.** "id", "name" ve "salary" sütunlarına sahip yeni bir "employees" tablosu oluşturmak için hangi SQL sorgusu kullanılır?

- A) CREATE TABLE employees (id INT, name VARCHAR(50), salary DECIMAL(10,2));
- B) INSERT INTO employees (id, name, salary) VALUES (1, 'John', 5000);
- C) UPDATE employees SET salary = 6000 WHERE id = 1;
- D) DELETE FROM employees WHERE id = 1;

**101.** "customers" tablosundaki tüm kayıtları silmek için hangi SQL sorgusu kullanılır?

- A) DELETE \* FROM customers;
- B) DELETE FROM customers;
- C) TRUNCATE TABLE customers;
- D) DROP TABLE customers;

**102.** "products" tablosunun "quantity" sütununu tüm kayıtlar için 10 olarak güncellemek için hangi SQL sorgusu kullanılır?

- A) UPDATE products SET quantity = 10;
- B) UPDATE products SET quantity = 10 WHERE quantity < 10;
- C) UPDATE products SET quantity = quantity + 10;
- D) UPDATE products SET quantity = 10 WHERE id = 1;

**103.** "employees" tablosundaki tüm çalışanların ortalama maaşını almak için hangi SQL sorgusu kullanılır?

- A) SELECT AVG(salary) FROM employees;
- B) SELECT COUNT(\*) FROM employees;
- C) SELECT DISTINCT name FROM employees;
- D) SELECT MAX(salary) FROM employees;

**104.** JDBC'de CREATE, DELETE veya UPDATE sorgusu çalıştırmak için kullanılan JDBC yöntemi hangisidir?

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeInsert()

**105.** JDBC'de bir SELECT sorgusu çalıştırıldıktan sonra sonuç kümesini almak için hangi JDBC yöntemi kullanılır?

- A) executeQuery()
- B) executeUpdate()
- C) execute()
- D) executeSelect()

**106.** "orders" tablosuna birden fazla sütun için yeni bir kayıt eklemek için hangi SQL sorgusu kullanılır?

- A) INSERT INTO orders VALUES (1, 'John', '2022-01-01');
- B) INSERT INTO orders (id, customer\_name, order\_date) VALUES (1, 'John', '2022-01-01');
- C) INSERT INTO orders (customer\_name) VALUES ('John');
- D) INSERT INTO orders SET customer\_name = 'John', order\_date = '2022-01-01';

**107.** "products" tablosundaki kayıt sayısını almak için hangi SQL sorgusu kullanılır?

- A) SELECT COUNT(\*) FROM products;
- B) SELECT MAX(price) FROM products;
- C) SELECT DISTINCT category FROM products;
- D) SELECT AVG(quantity) FROM products;

**108.** Çalışanın kimlik numarasına göre "employees" tablosundan belirli bir kaydı silmek için hangi SQL sorgusu kullanılır?

- A) DELETE \* FROM employees WHERE id = 1;
- B) DELETE FROM employees WHERE id = 1;
- C) DELETE FROM employees WHERE name = 'John';
- D) DELETE FROM employees WHERE salary > 5000;



**109.** "employees" tablosundaki çalışanların adı "J" ile başlayan tüm kayıtları almak için hangi SQL sorgusu kullanılır?

- A) `SELECT * FROM employees WHERE name = 'J%';`
- B) `SELECT * FROM employees WHERE name LIKE 'J%';`
- C) `SELECT * FROM employees WHERE name = '%J';`
- D) `SELECT * FROM employees WHERE name LIKE '%J';`

**110.** SQL sorgusunda "LIKE" anahtar kelimesiyle herhangi bir karakter dizisine eşleşmek için hangi joker karakter kullanılır?

- A) \_
- B) %
- C) \*
- D) ?

**111.** Ürün adı herhangi bir yerinde "shirt" kelimesini içeren tüm kayıtları almak için hangi SQL sorgusu kullanılır?

- A) `SELECT * FROM products WHERE name = '%shirt%';`
- B) `SELECT * FROM products WHERE name LIKE 'shirt%';`
- C) `SELECT * FROM products WHERE name LIKE '%shirt%';`
- D) `SELECT * FROM products WHERE name = 'shirt';`

**112.** "@gmail.com" ile biten müşteri e-posta adresine sahip tüm kayıtları almak için hangi SQL sorgusu kullanılır?

- A) `SELECT * FROM customers WHERE email = '@gmail.com';`
- B) `SELECT * FROM customers WHERE email LIKE '@gmail.com';`
- C) `SELECT * FROM customers WHERE email LIKE '@gmail.com%';`
- D) `SELECT * FROM customers WHERE email = '@gmail.com';`



**113.** Ürün adı tam olarak üç karakterden oluşan ve ardından "hat" kelimesini içeren tüm kayıtları almak için hangi SQL sorgusu kullanılır?

- A) `SELECT * FROM products WHERE name LIKE '___hat';`
- B) `SELECT * FROM products WHERE name LIKE '%hat';`
- C) `SELECT * FROM products WHERE name LIKE '___hat';`
- D) `SELECT * FROM products WHERE name = '%___hat%';`

**114.** JDBC'de PreparedStatement nesnesi oluşturmak için hangi yöntem kullanılır?

- A) `executeQuery()`
- B) `executeUpdate()`
- C) `prepareStatement()`
- D) `getConnection()`

**115.** Bir PreparedStatement nesnesinde bir SQL sorgusu için bir parametre değeri nasıl ayarlanır?

- A) `setParam()` yöntemini kullanarak
- B) `setString()` yöntemini kullanarak
- C) `setParameter()` yöntemini kullanarak
- D) `setValue()` yöntemini kullanarak

**116.** Bir PreparedStatement'in SQL sorgu dizisinde bir parametreyi temsil etmek için hangi yer tutucu kullanılır?

- A) `:param`
- B) `?`
- C) `#param`
- D) `$param`

**117.** Bir PreparedStatement nesnesi kullanarak bir INSERT sorgusunu nasıl yürütürsünüz?

- A) `executeQuery()` yöntemini kullanarak
- B) `executeUpdate()` yöntemini kullanarak
- C) `execute()` yöntemini kullanarak
- D) `executeInsert()` yöntemini kullanarak

**118.** Bir PreparedStatement nesnesi kullanarak bir SELECT sorgusunu nasıl yürütür ve sonuç kümesini alırsınız?

- A) executeQuery() yöntemini kullanarak
- B) executeUpdate() yöntemini kullanarak
- C) execute() yöntemini kullanarak
- D) executeSelect() yöntemini kullanarak

**119.** Bir PreparedStatement nesnesi kullanarak bir tablodan veri nasıl silinir?

- A) executeQuery() yöntemini kullanarak
- B) executeUpdate() yöntemini kullanarak
- C) execute() yöntemini kullanarak
- D) executeDelete() yöntemini kullanarak

**120.** JDBC'de hazırlanmış ifadeler kullanarak bir kaydı silmenin doğru yolu aşağıdakilerden hangisidir?

- A) DELETE FROM table\_name WHERE column\_name = ?
- B) UPDATE table\_name SET column\_name = NULL WHERE column\_name = ?
- C) INSERT INTO table\_name (column\_name) VALUES (?)
- D) SELECT \* FROM table\_name WHERE column\_name = ?

## 1. JDBC nedir?

Cevap: JDBC (Java Veritabanı Bağlantısı), Java programlarının standart SQL sorgularını kullanarak veritabanlarıyla etkileşimde bulunmasını sağlayan bir Java API'sidir.

## 2. JDBC mimarisinin temel bileşenleri nelerdir?

Cevap: JDBC mimarisinin temel bileşenleri şunlardır:

JDBC API: Veritabanlarıyla etkileşim için arayüzler ve sınıflar sağlar.

JDBC Sürücü Yöneticisi: Veritabanı bağlantıları için sürücülerini yönetir.

JDBC Sürücüler: Belirli veritabanları için JDBC API'sini uygular.

## 3. What are the different types of JDBC drivers?

Cevap : The four types of JDBC drivers are:

Type 1: JDBC-ODBC Bridge driver

Type 2: Native-API driver

Type 3: Network-Protocol driver

Type 4: Native-Protocol driver

## 4. JDBC kullanarak bir veritabanına nasıl bağlantı kurarsınız?

Cevap: JDBC kullanarak bir veritabanına bağlantı kurmak için uygun sürücüyü yüklemeniz, veritabanı URL'sini, kullanıcı adını ve şifreyi sağlamanız ve DriverManager sınıfının getConnection() yöntemini kullanmanız gerekmektedir.

## 5. JDBC'de bir SQL sorgusunu çalıştırmak için hangi adımlar izlenir?

Cevap: JDBC'de bir SQL sorgusunu çalıştırmak için izlenen adımlar şunlardır:

Sürücü sınıfını yükleyin.

Veritabanına bağlantı kurun.

Bir ifade veya hazırlanmış ifade oluşturun.

İfadeyi kullanarak sorguyu çalıştırın.

Sonuç kümesini işleyin (uygulanabilirse).

Bağlantıyı kapatın ve kaynakları serbest bırakın.

## 6. JDBC'de işlemleri nasıl yönetirsiniz?

Cevap: JDBC, işlemleri kontrol etmek için `setAutoCommit()` yöntemini sağlar. Varsayılan olarak, otomatik işlem etkin durumdadır. Bunun devre dışı bırakabilir ve işlemleri açıkça yönetmek için `commit()` ve `rollback()` yöntemlerini kullanabilirsiniz.

## 7. JDBC'de ResultSet nedir?

Cevap: Bir `ResultSet`, bir SQL sorgusunun sonucunu temsil eder. Sorgu sonucundan veri almak ve gezinmek için yöntemler sağlar.

## 8. execute(), executeQuery() ve executeUpdate() yöntemleri arasındaki farkı açıklayın.

Cevap: `execute()` herhangi bir SQL ifadesini çalıştırmak için kullanılır, `executeQuery()` `SELECT` ifadelerini çalıştırmak için kullanılır ve `executeUpdate()` veritabanını değiştiren ifadeleri (`INSERT`, `UPDATE`, `DELETE`) çalıştırmak için kullanılır.

## 9. JDBC'de istisnalar nasıl yönetilir?

Cevap: JDBC, veritabanıyla ilgili istisnalar için `SQLException` fırlatır. İstisnaları try-catch blokları kullanarak yönetmek ve kaynakları finally bloğunda düzgün bir şekilde kapatmak önemlidir.

## 10. JDBC'de bağlantı havuzlama nedir?

Cevap: Bağlantı havuzlama, yeniden kullanılabilirlik, performans ve ölçeklenebilirlik açısından geliştirme sağlamak için oluşturulan ve yönetilen bir veritabanı bağlantılarının havuzunu içerir. Her veritabanı işlemi için yeni bir bağlantı oluşturma işleminin gerektirdiği yükü azaltır.

## 11. ResultSetMetaData arabiriminin amacı nedir?

Cevap: `ResultSetMetaData` arabirimi, sonuç kümesinin yapısı ve meta verileri hakkında bilgi sağlar. Sütun adları, türleri ve boyutları gibi bilgileri içerir.

## 12. JDBC'de toplu işleme nedir?

Cevap: Toplu işleme, birden fazla SQL ifadesini bir araya getirerek ve bunları tek bir ağ işlemiyle veritabanına göndererek performansı artıran bir yöntemdir. Bu, `addBatch()` ve `executeBatch()` yöntemleri kullanılarak gerçekleştirilebilir.

## 13. JDBC'de bir INSERT ifadesinden sonra otomatik oluşturulan anahtarları nasıl alırsınız?

Cevap: `executeUpdate()` yöntemi etkilenen satır sayısını döndürür. Otomatik oluşturulan anahtarları almak için `Statement` veya `PreparedStatement` nesnelerinin `getGeneratedKeys()` yöntemini kullanabilirsiniz.

## 14.JDBC'deki CallableStatement arabiriminin amacı nedir?

Cevap: CallableStatement arabirimi, veritabanında saklı prosedürleri veya fonksiyonları çalıştırmak için kullanılır. Saklı prosedür parametrelerini ve çıktılarını işlemek için ek yöntemler sağlar.

## 15.JDBC'de büyük nesneleri (LOB'lar) nasıl işlersiniz?

Cevap: JDBC, büyük ikili ve karakter verileri sırasıyla Blob ve Clob arabirimleriyle işlemek için sağlar. LOB'lar, bu arabirimler kullanılarak alınabilir, güncellenebilir ve eklenir.

## 16.JDBC'deki DatabaseMetaData arabiriminin amacı nedir?

Cevap: DatabaseMetaData arabirimi, veritabanı hakkında bilgi sağlar, örneğin sürümü, desteklenen özellikleri, tabloları ve sütunları gibi. Veritabanı içgörüsü için kullanılabilir.

## 17.JDBC'de veri bütünlüğü ve referans bütünlüğü kısıtlamalarını nasıl yönetirsiniz?

Cevap: JDBC doğrudan veri bütünlüğü ve referans bütünlüğü kısıtlamalarını yönetmez. Bu kısıtlamalar veritabanı düzeyinde uygulanır. SQL ifadelerinin veya hazırlanmış ifadelerin tanımlanan kısıtlamalara uyduğundan emin olmanız gerekmektedir.

## 18.JDBC'deki ResultSetConcurrency arabiriminin amacı nedir?

Cevap: ResultSetConcurrency arabirimi, bir ResultSet için eş zamanlılık modunu belirlemenizi sağlar. Bu, sonuç kümesinin güncellenebilir olup olmadığını ve eşzamanlı erişime izin verilip verilmediğini belirler.

## 19.JDBC'de SQL enjeksiyonunu nasıl önlersiniz?

Cevap: SQL enjeksiyonunu önlemek için, Statement yerine PreparedStatement kullanmanız ve parametre değerlerini yer tutucular (?) kullanarak sağlamanız önerilir. Bu, kullanıcı girişinin doğru bir şekilde parametrelendirilmesini ve temizlenmesini sağlar.

İYİ ŞANSLAR!!!!!!

# CEVAB ANHATARI

## JDBC

1- D	4- C	7- D	10- D	13- A
2- C	5- A	8- B	11- B	14- D
3- D	6- B	9- C	12- B	15- C

## JDBC & DBC -DRIVER -CONNECTION

16- A	19- B	22- C	25- C	28- A
17- B	20- A	23- B	26- C	29- C
18- A	21- B	24- A	27- C	30- A

## STATEMENTS-TRANSACTION-EXECUTE

31- C	34- B	37- D	40- A	43- D
32- B	35- C	38- C	41- C	44- B
33- C	36- C	39- C	42- DB	45- A

## CRATE TABLE -EXECUTE QUERY

46- C	49- A	52- C	55- B	58- A
47- B	50- B	53- A	56- C	59- A
48- C	51- C	54- A	57- D	60- A

# CEVAB ANHATARI

## CRUD

61- A	64- B	67- E	70- B	73- B
62- A	65- C	68- B	71- B	74- D
63- C	66- A	69- A	72- D	75- A
76- A	79- A	82- D	85- C	88- D
77- C	80- B	83- D	86- D	89- A
78- D	81- C	84- D	87- D	90- A
91- A	94- C	97- B	100- A	103- A
92- C	95- A	98- A	101- B	104- B
93- B	96- B	99- B	102- A	105- A

## CRUD - LIKE - PREPARE STATEMENTS

106- C	109- B	112- B	115- B	118- A
107- B	110- B	113- A	116- B	119- B
108- B	111- C	114- C	117- B	120- A