

## 3- Form Gönderme

Gönder Düğmesi

&lt;input type = "submit" value = "Uygula"&gt; &lt;/input&gt;

- 101 -

## Sayfa 102

## SQL

## 21. SQL biliyor musunuz?

- Evet, SQL Sorguları ve DDL ve DML komutları yazma konusunda çok rahatım.
- Şu anda AMAZON CLOUD SERVER içinde çalışan Oracle veritabanıyla çalışıyor.
- DDL (Veri tanımlama dili): CREATE, ALTER, DROP, TRUNCATE
- DML (Veri işleme dili): SEÇ, SİL, EKLE, GÜNCELLEME

## 22. SQL?

- Yapılandırılmış Sorgu Dili. Veri tabanındaki verileri yönetmek ve işlemek için kullanılır.
- Çeşitli görevler için ifadeler sağlayın
  - Veri sorgulama
  - Tabloya satır ekleme, güncelleme, silme
  - Nesneleri oluşturma, değiştirme, değiştirme ve bırakma
  - Veritabanına ve nesnelere erişimi kontrol etme
  - Veritabanı tutarlılığı ve bütünlüğü

## 23. SQL ifadelerinin kategorileri nelerdir?

## ben. DML (Veri İşleme Dili)

- DML ifadeleri bir tablodaki kayıtları etkiler. Bunlar temel birkaç kayıt seçme gibi veriler üzerinde gerçekleştirdiğimiz işlemler bir tablodan, yeni kayıtlar eklemek, gereksiz silmek kayıtlar ve mevcut kayıtları güncelleme / değiştirme.

## ii. DDL (Veri Tanımlama Dili)

- DDL ifadeleri bir veritabanını veya tabloyu değiştirmek / değiştirmek için kullanılır yapı ve şema. Bu ifadeler tasarımı ele alır ve veritabanı nesnelerinin depolanması.

## iii. DCL (Veri Kontrol Dili)

- DCL ifadeleri, kullanıcıların erişim düzeyini kontrol eder veritabanı nesnelere sahip.

## iv. TCL (İşlem Kontrol Dili)

- TCL beyanları, kontrol etmenize ve yönetmenize olanak tanır SQL içindeki verilerin bütünlüğünü korumak için işlemler ifadeler.

## 24. Bana TCL'den bahseder misin?

- SQL dili, dört tür birincil dil ifadesine bölünmüştür: DML, DDL, DCL ve TCL.
- Bu ifadeleri kullanarak, veritabanı nesneleri oluşturarak ve değiştirerek bir veritabanının yapısını tanımlayabiliriz ve güncellemeler veya silmelerle bir tablodaki verileri işleyin.
- Tek bir iş birimi oluşturmak için hangi kullanıcının verileri okuyabileceğini / yazabileceğini veya işlemleri yönetebileceğini de kontrol edebiliriz.

## 25. Sürümler

- Java 8 → 2014 mevcut Java 7 → 2011 - 2014 Java 6 → 2006 - 2011  
Selenium 3.5.3

**26. Veritabanı Şeması?**

- Tüm tabloları ve sütun adlarını, veri türlerini ve PK, FK ve tabloların birbiriyle nasıl ilişkili olduğunu içeren bir şema gibidir.

**27. SQL cümlesi?**

- SEÇ ve Kimden

- 102 -

---

**Sayfa 103****28. Ne tür bir Veritabanı testi yapıyorsunuz?**

- Çoğunlukla Veritabanı doğrulamaları yapıyorum.
- Ön uçta değişiklikler yapıyorum veya veri ekliyorum (ödünc oluştuyorum) ve veritabanında doğruluyorum. Ön uçtaki veriler, DB
- Ayrıca RESTapi kullanarak değişiklikler yapıyorum ve Veritabanında da değişikliklerin başarılı olduğunu doğruluyorum.
- Ayrıca DB geçiş sürecini de destekliyorum. Kodum, JDBC'yi kullanarak Sybase'e (eski veritabanı) bağlanıyor ve ardından Oracle'a Bağlanıyor (YENİ DB) daha sonra verilerin taşındığından emin olmak için kayıtları karşılaştırırım

**29. RDBMS**

- İlişkisel Veritabanı Yönetim Sistemi
- Veriler, birbiriyle ilişkili tablolar halinde düzenlenir
  - Nasıl ilişkilidirler?
    - Birincil Anahtar (benzersiz ve NULL değil) ve Yabancı Anahtar (yinelenebilir ve NULL)
  - Ne tür bir veritabanı sistemiyle ilgili uzmanlığınız var?
    - SQL ve Oracle gibi RDBMS

**30. Kısıtlamalar nelerdir?**

- Tablo sütununun uyması gereken özellikler.
- Sütunların, verilerin nasıl depolanacağını tanımlayan kısıtlamaları vardır.
  - Birincil Anahtar: benzersiz ve BOŞ DEĞİL
  - Yabancı Anahtar: yinelenebilir ve NULL olan ve PK'de olmayan verileri ekleyemez
  - Benzersiz Anahtar: yalnızca benzersiz değer
  - Null: boş olabilir
  - null değil: null olamaz

**31. SQL'de veri türleri?**

- Numara
- Tamsayılar
- karakter → char (20): 20 yıllık boşluklar bellekten alınır
- varchar → varchar (30): varchar2 belleğinden 5 boşluk
- boole
- tarih
- para birimi

**32. SQL select deyimleri için yetenekler**

- Projeksiyon → Tablodaki sorgu tarafından döndürülen sütunları seçin
- Seçim → Bir sorgu tarafından döndürülen tablodaki satırları seçer
- Katıl → Farklı tablolarda depolanan verileri aralarındaki bağlantıyı belirterek bir araya getirir

- 103 -

---

**Sayfa 104****33. DML (Veri İşleme Dili) - DDL (Veri Tanımlama Dili)**

DML komut eylemleri geri yüklenebilir.

DDL komutu eylemleri **olamaz** / çözülmek restore edilecek.

**Komutlar :**

- Tablo adından **SEÇİN** ; ( **okuyun** )
- tablo adı değerlerine **INSERT (...)**; ( **ekle** )
- **GÜNCELLEME** tablename SET değeri NEREDE konum;
- tablename WHERE konumdan **SİL** ; ( **satırlar** )
- **BİRLEŞTİR**

**Komutlar :**

- **CREATE** tablo tablename (sütun1, sütun2...);
- **ALTER** tablo tablo adı değiştirme değeri;
- **TRUNCATE** tablo tablo adı; ( **tüm tablo verilerini silin** )
- **DROP TABLE** ; ( **yapısıyla birlikte tüm tabloyu silin** )
- **YENİDEN ADLANDIR**
- **COMMENT**

**34. Oracle'da PL / SQL işlevleri ve Tetikleyiciler?**

- Bir DML ifadesi yürütüldüğünde tetikleyici otomatik olarak tetiklenir.
- 2 tür işlev vardır:
  - Prosedür (Java'daki bir void yöntemi gibi)
  - İşlev (Java'da void olmayan (dönüş) bir yöntem gibi)

**35. SQL JOIN BİRLEŞİMLERİ****JOIN (INNER) JOIN**

birden çok tablodan veri alırken kullanılır  
ve yalnızca eşleşen verileri döndürür

**SOL DIŞ KATILMA**

birden çok tablodan veri alırken kullanılır  
ve soldaki tablo ile eşleşen sağdaki tüm tabloları döndürür  
tablo kayıtları.

**SAĞ (DIŞ) KATIL**

birden çok tablodan veri alırken kullanılır  
ve sağdaki tabloyu ve eşleşen sol tabloyu döndürür  
tablo kayıtları.

**TAM (DIŞ) KATIL**

birden çok tablodan veri alırken kullanılır  
ve hem tablo kayıtlarını hem de eşleşen ve  
eşleşmeyen

**36. BİRLİK**

- Union, iki sorgunun sonuç kümelerini birleştirir

table\_name'den column\_names'i seçin {UNION} table\_name'den column\_name'i seçin)

**37. En yüksek maaşlı ilk 3 çalışan nasıl bulunur?**

- mySQL'de                      \*\* oracle'da ROWNUM var

Çalışanlardan maaş, ad, soyadı SEÇİN  
MAAŞ İLE SİPARİŞ AÇIKLAMA SINIRI 3;

- 104 -

**Sayfa 105****38. Çalışanlarda yinelenen isimleri bulun**

Çalışanlardan ad, COUNT (ad\_adı) SEÇİN  
GROUP BY first\_name  
HAVING (COUNT (ilk\_ad> 1);

**39. Maaşları ortalamanın altında olan çalışanlar nasıl bulunur?**

Çalışanlardan birinci isim, maaş SEÇİN  
NEREDE maaş <= (Çalışanlardan ORTALAMA (maaş) SEÇİN);

**40. Her departmanda maksimum maaş nasıl bulunur?**

İlk\_adı SEÇİN, MAX (maaş)  
Bölümden d DIŞ ÜZERİNDEN ÇALIŞAN ÜZERİNDEN SOL (d. Bölüm\_kimliği = e. Bölüm\_kimliği)  
GROUP BY depart\_id;

**41. En düşük maaş nasıl bulunur?**

Çalışanlardan ad, soyad, maaş, iş kimliği SEÇİN  
NEREDE maaş = (MIN (maaş) Çalışanlardan);

**42. Çalışanların ikinci en yüksek maaşı nasıl bulunur?**

Çalışanlardan MAKS (maaş) SEÇİN  
Maaş NEREDE DEĞİLDİR (Çalışanlardan MAKS (maaş) SEÇİN);

#### 43. SQL Geliştirici

- Geliştirme ortamı (sorguları kullanarak veritabanının manuel olarak test edilmesi)
  - Sürüm 2.1 -2009 → 3.0 - 2011
  - Sürüm 4.0 - 2013 (en son)
- Migration sürümüne sahiptir (1.2) → kullanıcılara üçüncü taraf DB'deki verilere göz atmak ve bunlardan geçiş yapmak için tek bir nokta sağlar DB'den Oracle'a
- Window, Linux ve Mac OS x'i destekler

#### 44. SQL İfadeleri Yazmak

- Anahtar kelimeler büyük harfle yazılırken, sütunlar ve tablo adları küçük harflidir
- İfadeler büyük / küçük harfe duyarlı değildir
- Maddeler genellikle ayrı satırlara yerleştirilir
- Anahtar kelimeler kısaltılamaz veya satırlara bölünemez

#### 45. Aritmetik İfadeler

- Operatörleri herhangi bir maddede kullanırsınız (From maddesi hariç)
- Tarih ve Zaman Damgasıyla - yalnızca toplama ve çıkarma kullanılabilir
- Topla (+), Çıkar (-), Çarp (\*), Böl (/)

#### 46. Tarihlerle Çalışma

- Varsayılan tarih görüntüleme biçimi DD-MON-RR şeklindedir

Sysdate işlevi  
İade tarih ve saati  
Seç SysDate itubaren çift ;

#### 47. SQL tecrübeniz var mı?

- Evet, ilişkisel veritabanlarıyla çalıştım ve DDL ve DML komutları konusunda çok rahatım

- 105 -

## Sayfa 106

#### 48. Arka uç test çerçevesi

- Serileştirme ve serileştirme sürecinin bu şekilde gerçekleşmesi için çerçevemde RESTASSURED'i kullanmak  
Json yanıtını bir java toplama veri yapısında depolayın ve verileri beklenen değerle (ayrıca java'da saklanır) onaylayın  
veri yapısı)
- ÇERÇEVİMİ DAVETMEDEN ÖNCE manuel test için postacı kullanırım

#### 49. Arka Uç Testi

#### 50. Herhangi bir arka uç / veritabanı testi yaptınız mı?

- Evet, veritabanları ile çalışma konusunda çok deneyimim var.
- Ve SQL sorguları yazma konusunda çok rahatım.
- Oracle, MySQL, SQL Server gibi İlişkisel Veritabanları üzerinde çalışma deneyimim var
- İlişkisel olmayan veritabanlarıyla çalıştınız mı?
- Uygulamalı deneyime sahip değilim, ancak JSON formatı gibi olduğunu biliyorum
- Veritabanı ve JSON dosyalarıyla çalışma konusunda iyi deneyime sahibim.
- Ve hızlı öğrenen biriyim

#### 51. Veritabanı test çerçevesi

- Manuel test için SQL sorguları üretmek için SQL geliştiricisini kullanıyorum
- OTOMASYON İÇİN; oracle veritabanından bir BAĞLANTI alıp sonra oluşturarak java'yı entegre etmek için JDBC kitaplığını kullanıyorum  
SQL sorgularını kullanan ve ardından verileri RESULTSET nesnesine depolayan STATEMENTS.

**52. Üst veri nedir?**

- Verilerle İlgili Meta Veri Verileri

```
ResultSetMetaData rsmd = rs.getMetaData ();
int columNum = rsmd.getColumnCount ();
```

**53. Veri Yapıları ve Neden İhtiyacımız Var?**

- Veri yapıları, verileri verimli bir şekilde düzenlemek için
- manipülasyon: Veri ekleme, arama, okuma, silme.
- Verileri okumak ve uygulamamızdan, veritabanımızdan veya API'den veri depolamak için her zaman java veri yapılarını kullanırım.

- 106 -

**Sayfa 107****54. db'yi nasıl bağlayabiliriz?**

```
Bağlantı bağlantısı = DriverManager.getConnection (URL, kullanıcı, parola);
İfade ifadesi = connection.createStatement ();
ResultSet resultSet = deyim.executeQuery ("sorgu");

resultSet.close ();
ifade.close ();
connection.close ();

// Bağlantıdan sonra;
DatabaseMetaData db = connection.getMetaData ();

// resultSet'ten sonra;
ResultSetMetaData rs = resultSet.getMetaData ();
```

- Bağlanamama bir istisna oluşturacaksa:
  - SQLException (hatalı URL veya kimlik bilgileri)
  - ClassNotFoundException (JDB sürücüsü sınıf yolunda değil)

**55. JDBC için Bağımlılık**

```
<bağımlılık>
  <groupId> oracle </groupId>
  <artifactId> ojdbc6 </artifactId>
  <version> 11.2.0.3 </version>
</dependency>
```

**56. Prosedür nedir?**

- Depolanan yordam, veritabanında oluşturulan ve depolanan bir grup SQL deyimidir.
- Depolanan bir prosedür, giriş parametrelerini kabul eder, böylece tek bir prosedür ağ üzerinde birkaç kişi tarafından kullanılabilir. farklı giriş verilerini kullanan istemciler.
- Depolanan bir prosedür, ağ trafiğini azaltacak ve performansı artıracaktır. Depolanan bir yordamı değiştirirsek, istemciler güncellenmiş saklı yordamı alacaktır. Depolanan yordam oluşturma örneği

```
PROSEDÜR OLUŞTUR test_display AS
SEC ad, soyadı DAN tb_test ;
EXEC test_display ;
```

107

**Sayfa 108**

## 57. SQL Kısıtlamaları nedir?

- SQL Kısıtlamaları, bir tabloya girebilecek veri türlerini sınırlamak, doğruluğu ve bütünlüğü korumak için kullanılan kurallardır  
Tablo içindeki verilerin
- Kısıtlamalar aşağıdaki iki türe ayrılabilir:
  - o **Sütun düzeyi kısıtlamaları** : Yalnızca sütun verilerini sınırlar.
  - o **Tablo düzeyinde kısıtlamalar** : Tüm tablo verilerini sınırlar.
- Kısıtlamalar, veri bütünlüğünün veritabanında muhafaza edilmesini sağlamak için kullanılır. Aşağıdakiler en çok kullanılanlardır  
bir tabloya uygulanabilecek kısıtlamalar.

- **NOT NULL** → kısıtlama bir sütunun **NULL** değerine sahip olmasını kısıtlar

```
CREATE TABLE Student (s_id int NOT NULL, Name varchar (60), Age int);
```

- **UNIQUE** → kısıtlama, bir alanın veya sütunun yalnızca **benzersiz değerlere** sahip olmasını sağlar .  
o Tablo oluştururken kısıtlama (Tablo seviyesi)

```
TABLO OLUŞTUR Öğrenci (s_id int NOT NULL UNIQUE, Name varchar (60), Age int);
```

- o Bir tablo oluşturduktan sonra kısıtlama (Sütun seviyesi)

```
DEĞİŞİKLİK TABLOSU Öğrenci ADD UNIQUE (s_id);
```

- **BİRİNCİL ANAHTAR** → kısıtlama, bir veritabanındaki her kaydı benzersiz şekilde tanımlar. Benzersiz ve boş değil  
o Tablo seviyesinde Kısıtlama

```
TABLO OLUŞTUR Öğrenci (s_id int PRIMARY KEY, Name varchar (60) NOT NULL, Age int);
```

- o Sütun seviyesinde Kısıtlama

```
DEĞİŞTİRME TABLOSU Öğrenci EKLE BİRİNCİL ANAHTAR (s_id) ;
```

- **YABANCI ANAHTAR** → iki tabloyu ilişkilendirmek için kullanılır. **FOREIGN KEY** kısıtlaması, aynı zamanda tablolar arasındaki bağlantıları yok edin. Aşağıdaki tabloların yardımıyla kullanımını görelim:

## Müşteri\_Detay Tablosu

**c\_id**      **Müşteri\_adı**      **adres**

101	Adam	Noida
102	Alex	Delhi
103	Stuart	Rohtak

## Sipariş\_Detay Tablosu

**Sipariş Kimliği**      **Order\_Name**      **c\_id**

10	Sıra1	101
11	Sıra2	103
12	Sıra3	102

*Customer\_Detail tablosunda c\_id , Order\_Detail tablosunda yabancı anahtar olarak ayarlanan birincil anahtardır.*

*Herhangi bir yanlış veri eklemeye çalışırsanız, DBMS hata verir ve verileri eklemenize izin vermez .*

- o Tablo Düzeyinde FOREIGN KEY kısıtlamasını kullanma

```
CREATE table Order_Detail (
    order_id int BİRİNCİL ANAHTAR,
    order_name varchar (60) NOT NULL,
    c_id int YABANCI ANAHTAR REFERANSLAR Customer_Detail (c_id)
);
```

*Bu sorguda, Order\_Detail tablosundaki c\_id, yabancı anahtar olarak yapılır, bu da c\_id sütununun referansıdır. Customer\_Detail tablosu.*

- o Sütun Düzeyinde FOREIGN KEY kısıtlamasını kullanma

```
ALTER tablosu Order_Detail ADD YABANCI ANAHTAR (c_id) REFERANSLAR Müşteri_Ayrıntıları (c_id);
```

## Sayfa 109

- **CHECK** → kısıtlama, bir aralık arasındaki bir sütunun değerini kısıtlamak için kullanılır. Değerleri kontrol eder, bunları veritabanına kaydetmeden önce. Verileri bir sütuna kaydetmeden önce durum kontrolü gibidir.

- o Tablo Düzeyinde CHECK kısıtlamasını kullanma

```
Tablo Oluşturma Öğrenci (
    s_id int NULL DEĞİL KONTROL (s_id> 0),
    İsim varchar (60) NOT NULL,
    Yaş int
);
```

*Yukarıdaki sorgu, s\_id değerini sıfırdan büyük olacak şekilde kısıtlayacaktır.*

- o Sütun Düzeyinde CHECK kısıtlamasını kullanma

```
ALTER tablosu Öğrenci ADD CHECK (s_id> 0);
```

- **VARSAYILAN**

---

**Sayfa 110****Git ve GitHub****1. GitHub nedir?**

- Sürüm kontrol sistemi
- Dokümanların yeni / eski versiyonunu takip eder
- Dosya kümesini yönetir / depolar

**2. Depo nedir?**

- Dosyaların kaydedildiği klasör ve
- Tek, dosya koleksiyonları veya tek projeler içerebilir.

**3. Uzak ve Yerel Depo nedir?**

- *Uzak Depo*: Sunucuda ana bilgisayar (GITHUB) Değişikliklerimiz yerelden uzak depoya gider
- *Yerel Depo*: Genellikle bilgisayarınızda -Değişikliklerimiz burada Çalışma Dizini, dizin ve HEAD'den oluşur.

**4. Git komutları nelerdir?**

- **Ekle** : hazırlık alanına ekleyin
- **Kaydetme** : çalışma dizininden ve yerel depodan ekleyin
- **Gönder** : uzak depoya ekle
- **Pull** : değişiklikleri uzaktan çalışma dizinine alın
- url ile klonla: url'yi dizine kopyalar
- Git sürümü: size git sürümünü verir
- Git durumu: hangi daldaki olduğunuzu ve izlenmeyen tüm değiştirilmiş dosyaları gösterir
  - Menşei: uzaktan kumanda adı
  - Master: şubenin adı
- Git ekle:
  - Evreleme alanına ekleme
  - Özyinelemeli ekleme
  - Her şeyi ekler
- **git commit -m** : "mesaj tüm dosyalar için geçerli olacak"
- **git push** : origin nameOfBranch
- **git ignore** :
  - Notepad.gitignore → Not defterinde, hazırlama alanına eklemek istemediğiniz dosyaları ekleyin
  - YOK ETMEK İSTEDİĞİNİZ DOSYALARIN SİPARİŞ EDİLMESİ İÇİN .GITIGNORE DOSYASINI BASMALISINIZ
  - GIT'TE YOK EDİLDİ
  - Bazı dosyalar önemli değildir ve git'e gönderilmemelidir