**VERİ TABANI - 1**

Bahar Dönemi

1. **GELİŞMİŞ VERİ TABANI NESNELERİ**
   1. **GÖRÜNÜM (VIEW)**

**Görünüm / Görüntü / View;** veri tabanında tanımlı olan bir veya daha fazla verinin gösterildiği sanal tablolardır. Görünümler, veri tabanında saklanan SQL sorgularıdır.

Görünümler veri içermez, verileri gösterirler. Görünümler aracılığı ile verilerde işlemler yapabiliriz.

SELECT cümlesi veri tabanının veri sözlüğünde saklanır. Görüntüye ait bir depolama alanı yoktur.

**16.1.2. Kullanım Amaçları:**

* Görünüm, veriye erişimi sınırlandırır. Görünüm sorgulandığında sadece tablodan seçilen sütunların ve kayıtların görüntülenmesi sağlanır.
* Sık kullanılan karmaşık sorguları saha basit yazmamızı sağlar.
* Kullanıcıların belirlenen kriterlere göre veriye erişmesini sağlar.
* Her kullanıcının tablo ve alanlara erişimlerini daha kontrollü ve veri güvenliğini sağlayarak yapabiliriz.
* Sık sorgulanan verilere daha hızlı ulaşmamızı sağlar.

|  |  |  |
| --- | --- | --- |
| **Özellik** | **Basit Görünümler** | **Karmaşık Görünümler** |
| Tablo sayısı | Bir | Bir veya daha çok |
| Fonksiyon içerme | Hayır | Evet |
| Gruplanmış veri içerme | Hayır | Evet |
| Görüntü aracılığı ile DML işlemleri | Evet | Her zaman değil |

**16.1.3. Görüntülerin Oluşturulması**

Görünüm oluştururken **CREATE VIEW** komut grubu kullanılır.

CREATE VIEW gorunum\_adi

AS SELECT alan\_ad(lar)i FROM tablo\_adi

WHERE koşul;

**16.1.4. Görüntülerin Çalıştırılması**

Oluşturulan görünüm SQL sorgusu kullanılarak çağırılır.

SELECT alan\_ad(lar)i FROM gorunum\_adi;

**16.1.5. Görünümlerin Düzenlenmesi**

Oluşturulan görünüm ALTER VIEW komut grubu ile düzenlenir.

ALTER VIEW gorunum\_adi

AS sql\_sorgusu;

**16.1.6. Kısıtlara Göre Görünüm Oluşturmak**

CREATE VIEW goruntu\_adi

AS alt\_sorgu

WITH CHECK OPTION;

**16.1.7. Görünüm Silmek**

Herhangi bir görünümü silmek için DROP VIEW komutu kullanılır.

DROP VIEW gorunum\_adi;

* 1. **INDEKS (INDEX)**

Sıralı olmayan, belirli bir düzene uymayan veya karmaşık olan veriler arasında arama yapmak oldukça zordur. Tablolardaki kayıtları belirli bir mantıkla sıralarsak, aradığımız veriye daha kolay ulaşabiliriz. Yapılan bu işleme, **dizinleme** veya **indeksleme** denilir.

İndeksler, veri tabanında bulunan tablo ve görünüm/view yapılarına benzer. İndeksler, tabloların bir veya birden fazla alanı üzerinde tanımlanabilir.

İndeks, belirli anahtar alanlara göre tabloda aramayı, sıralamayı hızlandıran yapılardır. İndeksler, kayıtlara işaretçiler oluşturur. Bir tablonun belirli bir alanı sürekli sorgulanıyorsa o alana indeks tanımlanması uygun olur.

Gerekmediği sürece indeks tanımlanmamalıdır.

Birincil anahtar tanımlı olan bir yerde otomatik olarak indeks oluşturulur.

* + 1. **İndeks Oluşturma**

İndeksler, CREATE INDEX ifadesiyle oluşturulur.

CREATE [UNIQUE] INDEX indeks\_adi

ON tablo\_adi (alan\_adi1 [ASC | DESC] [, alan\_adi2 [ASC | DESC], …)

[WITH { PRIMARY | DISALLOW NULL | IGNORE NULL }]

**UNIQUE (Tekil):** İndeks tanımlanan alanlarda veriler benzersiz, birbirinden farklı olacaksa Unique olarak tanımlanır.

**ASC (Artan):** Tanımlandığı alanda verilerin artan şekilde sıralanmasını sağlayan parametredir.

**DESC (Azalan):** Tanımlandığı alanda verilerin azalan şekilde sıralanmasını sağlayan parametredir.

**Disallow Null:** Tanımlandığı alana verilerin girişinin zorunlu olmasını sağlar. Alanın NULL olmasını engeller.

**Ignore Null:** Tanımlandığı alanda verilerin girişinin zorunlu olmadığını belirten parametredir.

**Primary:** Alana birincil anahtar eklenmesini sağlar.

* + 1. **Birden Fazla Alan için İndeks Oluşturma**

CREATE INDEX indeks\_adi

ON tablo\_adi(alan\_adi1, alan\_adi2,…)

* + 1. **İndeks Silme**

Oluşturulan indeks DROP INDEX komutu ile silinir.

DROP INDEX indeks\_adi;

* 1. **SAKLI YORDAMLAR (STORED PROCEDURE)**

Veri tabanında işlem yapmayı sağlayan bir diğer yapı saklı yordamlardır. Daha hızlı sonuç alınmasını sağlar.

Stored Procedure kodları, sunucu üzerinde çalışırlar. Kodlar sunucuda olduğundan sunucu-istemci arasında gidiş geliş azalmış olur. Bu da işlemi hızlandırır. Sunucuda çalışan kodun sonucu, istemciye gönderilir.

Mantıksal ifadeler, SQL sorguları içerirler. Parametre alabilirler.

SP, bilgisayar ağ trafiğini azaltırlar. SQL ifadelerini basitleştirirler. Sistemi daha kolay yönetilebilir ve kontrol edilebilir hale getirirler.

* + 1. **Stored Procedure Oluşturma**

1. CREATE PROCEDURE | PROC prosedür\_adi

[@degisken\_ismi degisken\_veri\_tipi]

AS

BEGIN

SQL ifadesi

END

1. CREATE PROCEDURE | PROC prosedür\_adi

[@degisken\_ismi degisken\_veri\_tipi]

AS

SQL ifadesi

GO

* + 1. **Stored Procedure Çalıştırma**

Oluşturulan SP, EXEC komutu ile çağrılarak çalıştırılır.

EXEC sp\_adi parametre;

* + 1. **Stored Procedure Oluşturma**

1. ALTER PROCEDURE | PROC

AS

BEGIN

Sql\_ifadesi

END

1. ALTER PROCEDURE | PROC

AS

Sql\_ifadesi

GO

* + 1. **Stored Procedure Silme**

DROP PROCEDURE | PROC prosedür\_adi;

* 1. **DEĞİŞKENLER**

DECLARE @degisken\_adi veri\_tipi [(boyut)]

DECLARE @isim varchar(50)

DECLARE @ad varchar(20), @soyad varchar(30), numara int

* 1. **IF / ELSE Yapısı**

IF (şart)

BEGIN

Kodlar

END

ELSE IF (şart)

BEGIN

Kodlar

END

ELSE

BEGIN

Kodlar

END

* 1. **EKRANA MESAJ YAZDIRMA**

Ekrana yazı yazdırma PRINT komutu ile yapılır.

print(@degisken);

print ‘değişken’;

* 1. **FONKSİYONLAR**

Veri tabanı için kullanıcının oluşturduğu program parçalarıdır. View ve Stored Procedure’den daha esnek tasarlanabilirler.

1. Scalar Fonksiyonlar
2. Tablo Sonuçlu Fonksiyonlar
3. Sistem Fonksiyonları

Oluşturulan fonksiyon SELECT sorgusuyla çalıştırılır.

* + 1. **SCALAR FONKSİYONLAR (Scalar Valued Functions)**

Resim, yazı gibi veri türleri hariç, numerik ya da sütunların listelenmesinde görüntüsel farklılıklar oluşturulmasını sağlayan fonksiyonlardır.

Değer döndürmek için RETURNS deyimi kullanılır. BEGIN – END komutları arasında blok oluşturularak kodlanır.

CREATE FUNCTION fonksiyon\_adi (@degiskenadi degisken\_veri\_tipi,

@degiskenadi2 degisken\_veri\_tipi2,…)

RETURNS çıktı\_veri\_tipi

BEGIN

RETURN çıktı

END

* + 1. **TABLO SONUÇLU FONKSİYONLAR (Table Valued Functions)**

Sonucu bir tablo olarak dönen fonksiyonlardır. BEGIN-END bloğu ile oluşturulmadan yazılır.

CREATE FUNCTION fonksiyon\_adi (@degiskenadi degisken\_veri\_tipi,

@degiskenadi2 degisken\_veri\_tipi2,…)

RETURNS TABLE

AS

RETURN sql\_ifadesi

Çok satırlı – multi table- tablo sonuçlu fonksiyonlar BEGIN –END komutları arasına yazılabilir.

* + 1. **SİSTEM FONKSİYONLARI**

Matematiksel, tarih-saat vb. işlemlere yönelik veri tabanı sistemi için oluşturulup, veri tabanının kütüphanesinde saklanan fonksiyonlardır.

Abs(), power(), cos(),…

**Kaynakça:**

* Özdemir S., “C# .Net 2008 ve ASP.NET”, Nirvana Yayınları, Ankara, 2008.
* Obalı M., “Oracle 10g”, Pusula Yayıncılık, İstanbul, 2007.
* Kaya Y. ve Tekin R., “Veritabanı ve Uygulamaları”, Papatya Yayıncılık, İstanbul, 2007.

Öğr. Gör. Kevser ONUR