Veri Tabanı Programlamaya Giriş

Kitap özeti

Veri Tabanı Programlamaya Giriş

SQL insanların veritabanı sistemleri ile konuşmasını sağlayan popüler bir dildir. Bu dil sayesinde, bir veritabanından kayıtları alabilir, değiştirebilir ya da yeni kayıtlar ekleyebiliriz. SQL bir dildir; ancak bir programlama dili değildir. Program geliştirme aşamasında SQL'den faydalanılır, ancak tek başına bu iş için yeterli değildir.

Verilerin belli özelliklerine göre gruplanıp diske kaydedilmesi işine veritabanı yönetimi denir. Veritabanlarından en popüler olanı, ilişkisel veritabanıdır.

Veritabanı Yönetim Sistemleri (VTYS)

Veri Tabanı Yönetim Sistemleri, fiziksel hafızada bilgileri çeşitli özelliklerine göre gruplandırıp şekillendirdikten sonra saklayan programlardır. Kısaca VTYS diye adlandıracağız. VTYS, saklanan bu veriyi, SQL komutları ile insanların istekleri çerçevesinde işler, yeniden şekillendirirler.

Veritabanı Nerelerde Kullanılır?

Veritabanı programlama ile bir çok proje geliştirilebilir. Bir İngilizce-Türkçe sözlük bu yolla kolayca yazılabilir. Bir kütüphane takip otomasyonu, bir hastane otomasyonu, muhasebe programları ve daha bir çok otomasyon programı temelde veritabanı projesidir.

Tablolar:

Bir tablo yani günlük hayattaki 'liste' kavramı, satırlardan ve sütunlardan oluşur.Örneğin elimizde bir kitap tablosu olduğunu düşünelim.Her satırda bu kitap ile ilgili özellikler yer alacaktır.

Satır (Row): Bir tabloda yer alan her bir kayıt bir satıra karşılık gelir.

Sütun (Column): Tablolar dikey sütunların yan yana gelmesiyle meydana gelmiştir.

Kayıt (Record): Yapılandırılmış verilerden her birine kayıt denir.

Veri Tipi (Data Type): Bilgisayar, kayıtları yapısal olarak tutarken, onların yapıları hakkında fikir sahibi olabilmek için bazı özelliklerinin önceden tanımlanması gerekir.

Zorlayıcı (Constraint): Herhangi bir alan için girilebilecek verileri kısıtlayıcı kurallara zorlayıcılar denir.

Değişken İsimlendirme Kuralları:

1. Değişken isimleri, harf ile başlamak zorundadır.
2. Değişken isimleri, harf, rakamlar ve '_' dan oluşmak zorundadır.
3. Değişken isimlerinde Türkçe'de ki noktalı harfler (İ,ı,Ğ,ğ,Ü,ü,Ş,ş,Ç,ç,Ö,ö,) yer alamaz.
4. Ayrılmış kelimeler değişken adı olamazlar (select, like, not, or, delete, update vs.)
5. SQL büyük-küçük harf duyarlı değildir.
6. Değişken isimlerinde boşluk yer alamaz.

SQL Veri Tanımlama Dili:

Bir veritabanı şu şekilde açılır :

CREATE DATABASE database_name à CREATE DATABASE dbKutuphane;

Bir veritabanı şu şekilde silinir :

DROP DATABASE database_name à DROP DATABASE dbKutuphane

Tablo Oluşturma

CREATE TABLE tablo_adi à CREATE TABLE Kitap

Örnek:

```
CREATE TABLE Kitap (
kitapNo int NOT NULL ,
kitapAdi varchar63),
ISBNNo varchar (15),
sayfaSayisi int ,
kitapOzeti varchar (255)
)
```

Bir sütunu silmek için,

ALTER TABLE Kitap DROP COLUMN kitapBedeli;

İndeksler:

Veritabanlarında indeks oluşturarak, verileri veritabanındaki kayıtlı oldukları sıradan başka bir sırada gösterebiliriz.

-CREATE UNIQUE INDEX indKitapNo ON Kitap(kitapNo)

Dedikten sonra, iki farklı kitaba aynı kitap no'sunu vermek mümkün olamayacaktır.

-DROP INDEX Kitap.indKitapNo

Oluşturduğumuz indexi siler.

View'ler

Bazen, tabloları olduklarından farklı gösterecek filtrelere ihtiyaç duyarız. Aslında tablo gibi kullanılsa da böyle bir tablo halihazırda yoktur.

-CREATE VIEW vwBTKitaplari AS SELECT * FROM kitap WHERE kitapNo > 5

Bu örneği çalıştırdığımızda veri tabanındaki sadece kitapNo'su 5'den büyük olan veriler ekrana gelecektir.

SQL Veri İşleme Dili

Veri İşleme Dili, verinin şablonu üstünde değişiklik yapmaz. Sadece var olan tablolardaki bilgileri uygun şekilde raporlamak (SELECT), yeni kayıtlar eklemek (INSERT), kayıtlar üstünde güncelleme yapmak(UPDATE) ve kayıtları silmek (DELETE) için kullanılır.

1.RESULTSET(RECORDSET, DATASET)

VTYS'de bir sorgu çalıştırıldığında, tablo mantığında bir sonuç üretir. Bu sonuca ResultSet denir. Bir ResultSet, birden fazla tablodan kayıt içerebilir.

Kitaplar tablosundaki tüm kayıtları seçelim:

-SELECT * FROM Kitap

Koşula bağlı SELECT ve WHERE Yapısı:

Bazı koşullara uyan bilgileri almamız gerekebilir. Bu durumda WHERE cümleciğini takip eden kısımda, bu şartı belirtebiliriz.

-SELECT alan1[,alan2, alan3,.....] * FROM tablo_adi WHERE şart1[AND şart2[OR şart3[NOT şart4]]];

Örnek:

Sayfa sayısı 200'den fazla olan ve kitap numarası da 12'den büyük olan kitapların listesi:

-SELECT * FROM Kitap WHERE kitapNo > 12 AND sayfaSayisi > 200;

Mantıksal İşaretler

AND İŞARETİ:Şartlardan her ikisini de sağlayan kayıtları seçmek için kullanılır..

OR İŞARETİ: Şartlardan en az birinin sağlanması halinde, kayıtlar seçilir.

NOT İŞARETİ: Şartı sağlamayan kayıtları bulmak için kullanılır.

Matematiksel Karşılaştırma İşaretleri:

Koşullarda, verilerin durum Matematiksel Karşılaştırma İşaretleri kullanılarak ifade edilir. Bu işaretler Şunlardır:

=: Eşittir

>: Büyüktür

<: Küçüktür

>=: Büyüktür veya Eşittir(Büyük-Eşit)

<=: Küçüktür Veya Eşittir(Küçük-Eşit)

<>: Eşit Değildir

!=: Eşit Değildir

Kitap No 12'den büyük olan veya, sayfa sayısı 200'den büyük olan veya adı 'Visual Basic.NET' olan ama ISBN NO '975-316-622-2' olmayan kitapların listesi:

SELECT * FROM Kitap WHERE (kitapNo > 12 AND sayfaSayisi > 200 OR KitapAdi='Visual Basic.NET') AND ISBNNO <> '975-316-622-2';

1,5 ve 6 nolu kitapların ödünç hareketlerini görmek için

-SELECT * FROM odunc WHERE kitapNo=1 OR kitapNo=5 OR kitapNo=6;

Yerine

-SELECT * FROM odunc WHERE kitapNo IN(1,5,6);Kullanımı daha kolaydır.

3,5 ve 11 nolu kitapların herhangi birinden kalın olan ve kitap no da 11'den büyük olan kitapların listesini bulalım:

SELECT kitapNo,kitapAdi,sayfaSayisi FROM Kitap WHERE sayfaSayisi > ANY(SELECT sayfaSayisi FROM Kitap WHERE kitapNo IN(3,5,11))

AND kitapNo>11

UNION (BİRLEŞTİRME)

UNION komutu, iki SELECT sorgusunun sonucunu veya iki tabloyu tek bir sonuç halinde alabilmek için kullanılır.

Genel kullanımı şu şekildedir:

1.SELECT İFADESİ....

UNION

2.SELECT İFADESİ....

EXISTS, NOT EXISTS

EXISTS kullanıldığında, dışarıdaki sorguda, bir veya daha fazla kayıt dönerse, dışarıdaki sorgu çalıştırılır. Hiç kayıt dönmezse, dışarıdaki sorgu çalıştırılmaz. NOT EXISTS ise içerideki sorgunun sonucunda sıfır kayıt dönüyorsa, dışarıdaki sorgunun çalıştırılması için kullanılır.

5 no'lu kitap ödünç verildi ise kitap no'sunun ve kitap bilgilerini seçelim:

SELECT kitapNo,kitapAdi FROM Kitap WHERE EXISTS(SELECT * FROM odunc WHERE kitapNo=5)

AND kitapNo=5;

EXCEPT (FARK BULMA)

İki SELECT sorgusu sonucu veya iki tablo arasındaki farkı bulmaya denir. Bu işlem için de bir çok yöntem kullanılabilir. NOT IN, NOT EXISTS bunlardan ikisidir.

Hangi kitaplar ödünç verilmemiştir?

SELECT * FROM Kitap WHERE kitapNo NOT IN (SELECT kitapNo FROM odunc WHERE geldiMi=0);

BETWEN.... AND....

Bir Aralık içersinde sorgulama yapmak için BETWEEN altSinir AND ustSinir şeklindeki yapı kullanılır.

Sayfa Sayısı, 300 ile 500 arasındaki Kitapların listesini almak istediğimizde,

SELECT * FROM Kitap WHERE sayfaSayisi>300 AND sayfaSayisi < 500;

Yerine

SELECT * FROM Kitap WHERE sayfaSayisi BETWEEN 300 AND 500; de diyebiliriz.

Joker Karakterler:

Seçimi biraz genişletelim ve adında 'yol' geçen Kitapları listelemek istesek?Bu durumda joker karakterleri kullanmamız gerekecektir. '%' ve '_' veya '?' karakterlerine joker karakterler denir.

Adında yol geçen kitapların bir listesini alalım:

SELECT * FROM Kitap WHERE kitapAdi LIKE '%Yol%';

Adının ilk harfi 'Y' olan üyelerin listesi.

SELECT * FROM uye WHERE Adi LIKE 'Y%';

Joker Karakterler

%: Birden fazla harf veya rakamın yerini tutar.

*: Bazı sistemlerde, birden fazla harf ve rakam yerini tutar.

_: Bir tek harf veya rakam yerini tutar (Bir çok sistemde)

?: Bir tek harf veya rakam yerini tutar (Bazı diğer sistemlerde)

[ABC]: Herhangi bir harf yerine gelebilecek harfleri belirtir.(SQLServer,Sybase)

[^ABC]: Herhangi bir harf yerine gelemeyecek harfleri belirtir.(SQL Server, Sybase)

Null Karşılaştırma

Bazen bir alana değer girilmiş olup olmadığın karşılaştırmak zorunda kalabiliriz.

ISBN numarası olmayan kitapların listesi:

SELECT *FROM Kitap WHERE ISBNNo IS NULL;

3.ORDER BY

SELECT işlemi ile elde edilen sonucun bir alana göre sıralanmış olarak listelenmesini isteyebiliriz. Bu durumda ORDER BY deyimi kullanılır.

Genel Yapısı şu şekildedir:

SELECT alanlar

FROM tablo_adi

WHERE şartlar

ORDER BY sutun1 [DESC|ASC][,sutun2 [DESC|ASC],.....];

ASC bir alana göre ARTAN sıralatmak için kullanılır. Aslında artan-azalan şartı verilmediğinde de normalde artan

sıralanır. Ancak kayıtların azalan sırada sıralanması isteniyorsa, mutlaka DESC kullanılmak zorundadır.

COUNT(sutun_adi): Referans sütuna göre toplam kaç kayıt yer aldığını bulur.

Yazarlar tablosunda, tüm yazarların doğum tarihi henüz girilmemiştir. Bu durumda iken, yazarların sayısını COUNT(*) ve bir de COUNT(dogum) olmak üzere iki farklı şekilde seçelim:

SELECT COUNT(*) FROM yazar;

>> 20

SELECT COUNT(dogum) FROM yazar;

>> 0

6.INSERT

Bir tabloya SQL ile kayıt eklemek için INSERT deyimi kullanılır.

INSERT INTO tablo_adi(alan1[,alan2,....])

VALUES(deger1[,deger2,....])

Yeni bir kitap ekleyelim.

INSERT INTO Kitap(KitapAdi,SayfaSayisi, kitapOzeti)

VALUES('Photoshop Efektleri',330,'Photoshop efektlerini yakından tanıyın');

7.UPDATE

Bir kayıt üstünde değişiklik yapmak gerektiğinde, UPDATE deyimi kullanılır. Yapısı şu şekildedir:

UPDATE tablo_adi

SET alan1=yeniDeger1[, alan2=yeniDeger2, ...]

[WHERE sart]

Örnek:

UPDATE kitap

SET kitapBedeli = kitapBedeli*1,18;

8.DELETE

Tabloda yer alan kayıtların tamamını veya bir kısmını silmek için DELETE deyimi kullanılır.

Genel kullanımı şu şekildedir:

DELETE

FROM tablo_adi

[WHERE sart];

Örnek:

Soyadı 'OZAN' olan üyeleri silmek istersek:

DELRTE FROM uye WHERE soyad LIKE '%OZAN%'

GENEL SQL FONKSİYONLARININ KULLANIMI

Tarih-Zaman Fonksiyonları:

GETDATE() : Microsoft ve Sybase temelli sistemlerde sitem tarihini bulur.

DATE() olarak da geçebilmektedir.

SYSDATE() : Oracle'de aynı işlevi yerine getirir.

NOW() : MS temelli sistemlerde, anlık detaylı zamanı(gün-ay-yıl-saat-dk-sn) verir.

Örnek:

Üstünde çalıştığımız VTYS'de şu anki vakti almak için;

SELECT GETDATE()

>>6/22/2003 11:39:19 PM

Karakter İşleme Fonksiyonları:

CHR(): Bir ASCI kodunun karakter karşılığını döndürür.

CONCAT(metin1,metin2): İki veya daha fazla karakter ifadeyi uç uca eklemek için kullanılır.

Kitap Adları ile yazarlarının adlarını bir tek alan olarak seçtirmek istersek:

SELECT kitapAdi + yazarAdi FROM Kitap;

Oracle'de

SELECT kitapAdi || yazarAdi FROM Kitap;

Veya

SELECT CONCAT(kitapAdi,yazarAdi) FROM Kitap;

INITCAP(karakter): İlk harfi büyük yapar.(Oracle)

REPLACE(karakter,ifade,yerinegelecekIfade): İfadeleri bulup değiştirir.

LOWER(metin), UPPER(metin) : Tamamı Küçük veya tamamı büyük harf yapmak için kullanılır.(Oracle)

LCASE(metin) ve UCASE(metin):Tamamı Küçük ve tamamı büyük harf yapmak için kullanılır. (MS)

LTRIM(metin), RTRIM(metin), TRIM(metin): Soldaki, sağdaki veya her iki taraftaki boşlukları atmak için kullanılır.

LENGHT(metin): Bir metnin uzunluğunu verir.(Oracle)

LEN(metin): Bir metnin uzunluğunu verir.(MS temelli Sistemler)

ARİTMETİK İŞARETLER

Her yerde olan aritmetik işlemleri SQL'de de geçerlidir.

+ :Toplama

- :Çıkartma

* :Çarpma

/ :Bölme

% :Mod(Bazı Sistemlerde MOD da kullanılır)

Ruşan Sakarya