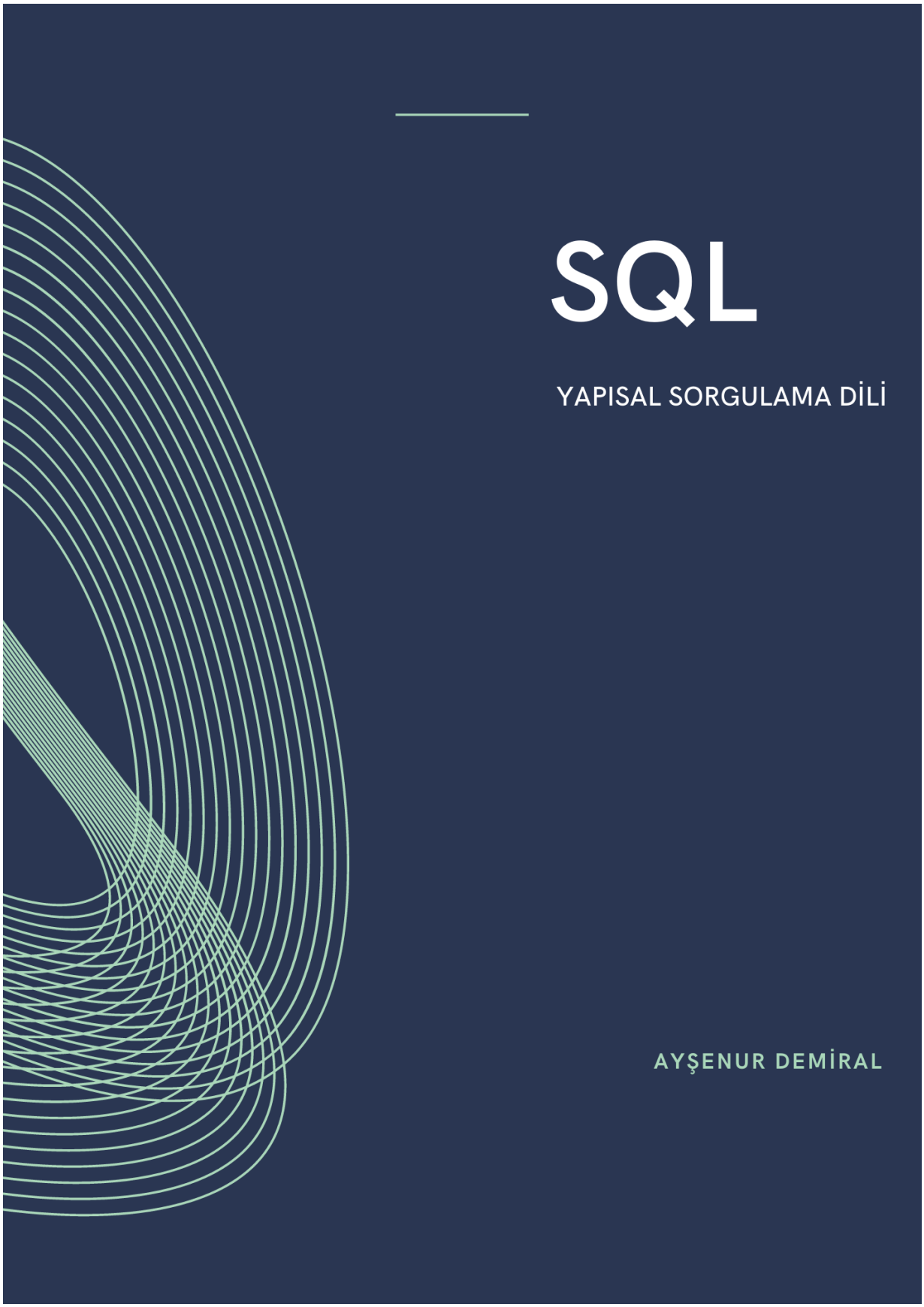


SQL

YAPISAL SORGULAMA DİLİ

AYŞENUR DEMİRAL

İÇİNDEKİLER

SQL - YAPISAL SORGULAMA DİLİ

1.VERİ TABANI NEDİR?

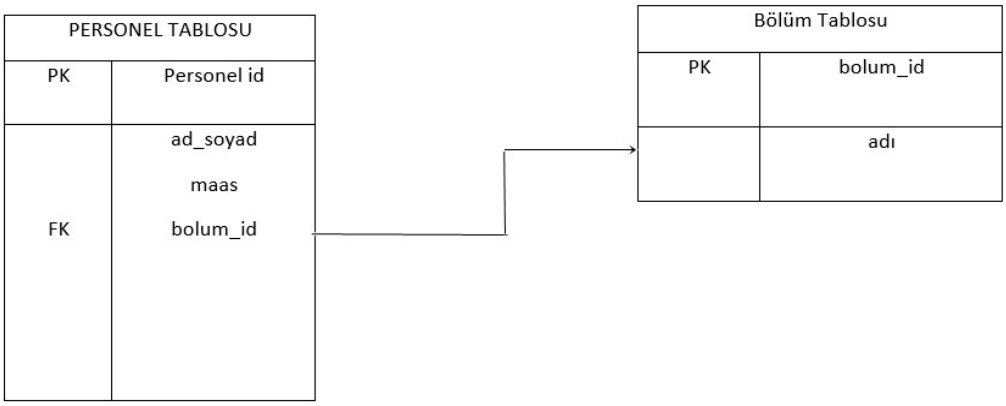
Veri tabanı kavramı günümüzde oldukça yaygın kullanılan bir kavramdır. Veri tabanı, içerisinde birbirleriyle ilişkili verileri barındıran düzenli bilgiler topluluğudur. Üniversitede bulunan öğrenci işleri bilgi sistemi, bir hastanenin otomasyon verileri veya ticari bir şirkette müşterilerin bilgilerine, ödemelerine kadar bilgileri tutmak için kullanılan bir sistemdir. Veri tabanı oluşmasını sağlayan kavramlar vardır. Tablo, satır, sütun, anahtar gibi. Anahtar kavramı özellikle tablolar arası ilişkilerde oldukça önem taşımaktadır.

2.ANAHTAR NEDİR?

Anahtar bir veya birden fazla alanın bir satır için niteleyici olarak girilmesini zorunlu kılan bir çeşit zorlayıcıdır. İki çeşit anahtar türü vardır.

1-Birincil Anahtar (Primary Key); eğer ki bir niteliğin değeri her bir varlık için farklıysa yani aynı değeri birden fazla içermiyorsa bu nitelik birincil anahtar yani anahtar niteliktir. Yani o sütunun ayırt edici özelliğidir. Örneğin bir tabloda aynı isme ait kişiler olabilir bu ayırt edici bir özellik değildir. Ancak personel numarası kişiye özeldir. Anahtarımız tabloya göre değişiklik gösterebilir ben kafanızda bir şey oluşabilmesi için böyle bir örnek verdim. Birincil anahtar NULL değer içeremez.

2-Yabancı Anahtar (Foreign Key); bir tabloya girilecek kayıtları başka bir tablonun belli alanındaki verilerle sınırlandırmayı ve ilişkilendirmeyi sağlayan bir anahtardır.



3.VERİ TABANI YÖNETİM SİSTEMİ NEDİR?

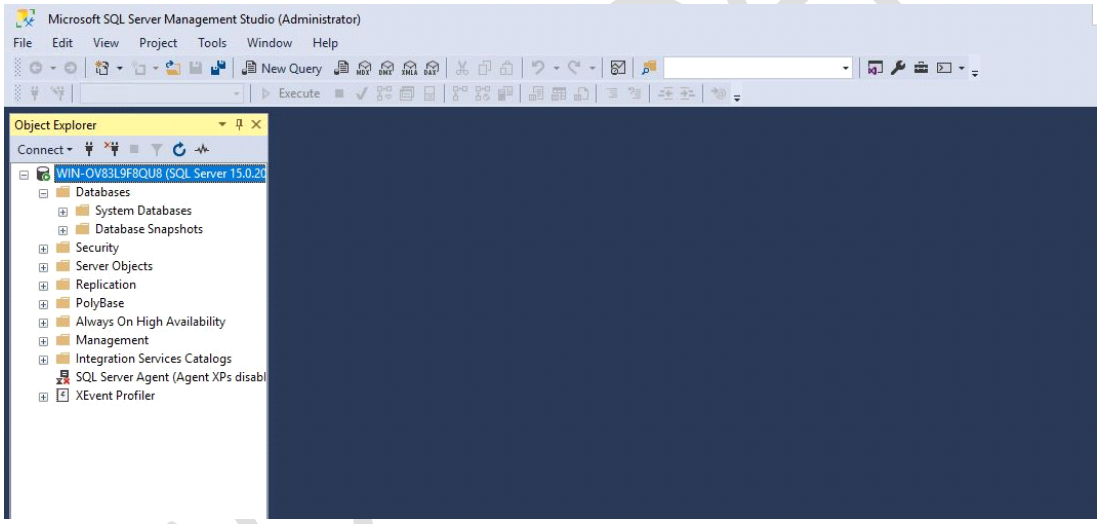
Yeni bir veri tabanı oluşturmak, veri tabanını düzenlemek, kullanmak, geliştirmek veya bakımını yapmak gibi çeşitli karmaşık işlemlerin gerçekleştirildiği bir yazılım sistemidir. Veri tabanı yönetim sistemi veri modeline göre;

Hiyerarşik, Ağ, İlişkisel, Nesneye yönelik olmak üzere 4 grupta sınıflandırılır.

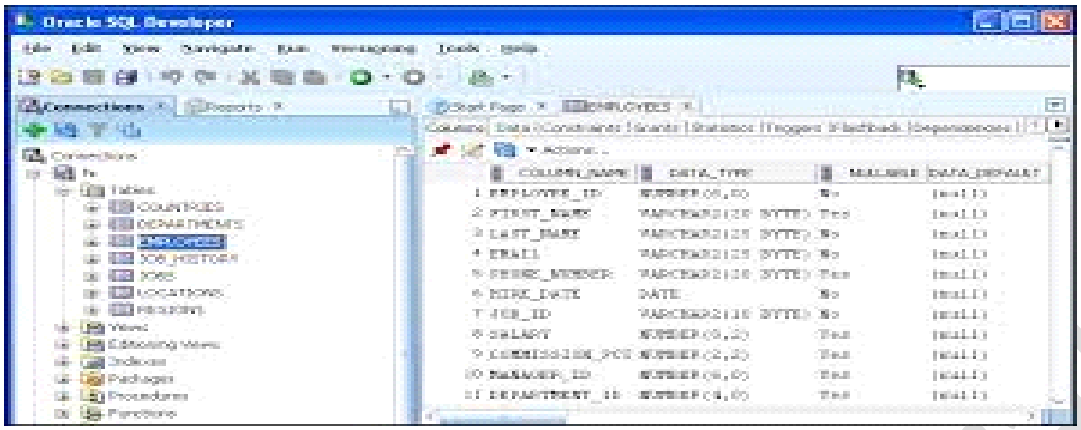
3.1.VERİ TABANI YÖNETİM SİSTEMİ YAZILIMLARI

SQL sistemiyle sorgulama yapan yazılımlardan bazıları şunlardır:

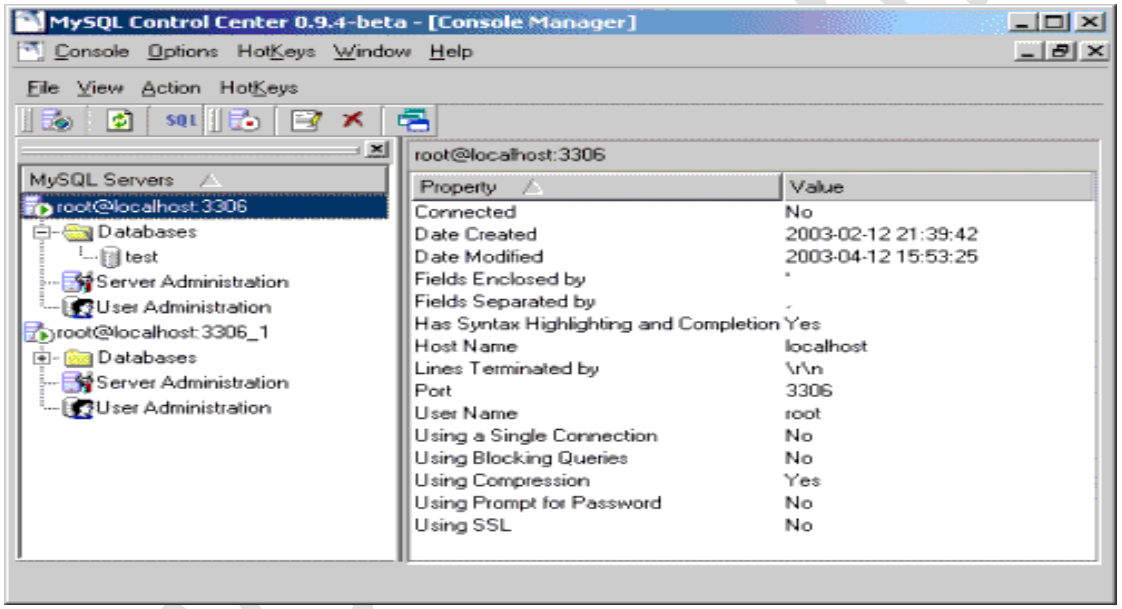
3.1.1.Microsoft SQL Server, Microsoft tarafından yazılmıştır. Veri depolama, gruplama, analiz etme işlevlerine sahiptir. Orta ve büyük ölçekli işlemler için kullanılır ve sadece Windows işletim sistemlerinde çalışır. Aşağıda bu programın ekran görüntüsünü görebilirsiniz.



3.1.2.Oracle, Oracle firması tarafından yazılmıştır. Gelişmiş bir yazılımdır bu yüzden genellikle kurumsal alanlarda kullanılmaktadır. Verileri çok kullanıcılı ortamlarda depolamayı ve güvenli şekilde erişimi yönetir. Birçok işletim sistemi tarafından desteklenmektedir.



3.1.3.MySQL, açık kaynak kodlu bir yazılımdır. Windows, Unix işletim sistemlerini destekler. Çoğunlukla web sunucularında kullanılmaktadır.



Ben sizlere MYSQL veri tabanı yönetim sistemine göre SQL kullanımından bahsedeceğim ama öncesinde MYSQL de tanımlı olan veri tiplerini öğrenmemiz gerekir.

4.VERİ TİPLERİ

Tinyint: çok küçük tam sayı değerleri için kullanılmaktadır. -128 ile 127 arasında tanımlı değerler almaktadır.

Smallint: küçük tam sayı değerleri içindir. -32768 ile 32767 arasında tanımlı değerler almaktadır.

Mediumint: orta büyüklükteki tam sayı değerleri içindir. -8388608 ile 8388607 arasında değerler almaktadır.

Bigint: büyük tam sayı değerli için kullanılmaktadır.

Float: sayırları kesirleri ile birlikte tutmaktadır.

Datetime: yıl+ay+gün+saat+dakika+saniye biçimindeki zaman bilgisini tutar.

(YYYY-MM-DD HH:MM:SS)

Timestamp: 1 ocak 1970'den 18 ocak 2038'e kadar olan ve yıl+ay+gün+saat+dakika+saniye biçimindeki zaman bilgisini tutar. (YYYYMMDDHHMMSS)

Date: 1000-01-01'den 9999-12-31'e kadar değişebilen tarih bilgisini tutar.

Char (n): belirtilen n sayısı kadar numerik ifadeler için kullanılmaktadır.

Text: en fazla 65535 karakter alabilen metin alanıdır.

Vharchar (n): belirtilen n (0-255) sayısı kadar numerik ifadeler için kullanılmaktadır.

Bool: 0 veya 1 değerini alan veri türüdür.

5.SQL NEDİR?

SQL, Structed Query Language kelimelerinin kısaltmasından oluşan bir dildir.

SQL sayesinde kullanıcılar veri tabanı sistemleriyle iletişim kurmaktadır. Veri tabanı sistemlerinin çoğu bu dili kullandığından dolayı standart haline gelmiştir. 1983'lü yıllarda IBM tarafından SQL standartları tanımlanmış ve 1987'de ISO ardından da ANSI tarafından bir standart olarak kabul edilmiştir.

SQL bir programlama dili değildir. SQL komutları kullanılarak; veri tabanına kayıt ekleme, kayıt silme, kayıt değiştirme, tablo oluşturma ve kayıt listeleme gibi birçok işlem geçekleştirilir. Günümüzde kullanılan programlama dillerinin neredeyse tamamı SQL komutlarını desteklemektedir.

SQL kullanmanın birçok faydası vardır. Örneğin, bilgiler kolaylıkla kategorize edilebilmektedir. Bu sayede hem veri dağınıklığı ortadan kalkar hem de zamandan tasarruf edilir. Aranılan bilgiye kolaylıkla erişilebilir. Bunların yanı sıra SQL sayesinde gereksiz tekrarların oluşması da engellenir.

6.SQL KOMUTLARI

SQL ifadeleri yapısal olarak üç gruba ayrılmaktadır:

1- DDL (Data Defination Language/Veri Tanımlama Dili)

2- DML (Data Manipulation Language/Veri İşleme Dili)

3- DCL (Data Control Language/Veri Kontrol Dili)

6.1.VERİ TANIMLAMA DİLİ (DDL)

Veri tanımlama dili tutulan verinin ne olduğundan çok tutulan verinin tipiyle ilgilenir. Bu kısımda bazı temel ifadeler bulunur.

Create: nesne oluşturmak için kullanılır.

Alter: nesneler üzerinde değişiklik yapmak için kullanılır.

Drop: nesneleri silmek için kullanılır.

Create İfadesi

6.1.1.Veri tabanı Oluşturmak (Create Database)

create database komutu yeni bir veritabanı oluşturulmasını sağlar. Aşağıdaki gibi kullanılır.

CREATE DATABASE veritabanı_ismi

6.1.2.Tablo Oluşturmak (Create Table)

Seçili olan veri tabanı üzerinde yeni bir tablo oluşturmak için kullanılır. Tablo oluştururken, sütun isimleri, tutulacak veri tipleri, anahtar değeri gibi özellikler de belirlenebilir.

CREATE TABLE tablo_ismi

(sütun_ismi1 veri_tipi,

Sütun_ismi2 veri_tipi, …)

- Burada anahtar kullanarak bir öğrenci tablosu oluşturalım.

CREATE TABLE öğrenciler(

Ogr_no int NOT NULL PRIMARY KEY,

Ogr_adi varchar(25))

6.1.3.Alter İfadesi

Alter ifadesi veri tabanında değişiklik yapmak için kullanılır. Ekleme, silme, güncelleme vs.

ALTER TABLE tablo ADD sütun_adı özellikler (tabloya yeni bir sütun eklenir)

ALTER TABLE tablo DROP COLUMN sütun_adı (belirtilen sütun silinir)

ALTER TABLE tablo ALTER COLUMN sütun_adı özellikler (belirtilen sütunun özellikleri değiştirilir)

6.1.4.Drop İfadesi

DROP TABLE tablo_adı (belirlenen tablo silinir)

DROP DATABASE veritabanı_adı (belirlenen veritabanı silinir)

6.2.VERİ İŞLEME DİLİ (DML)

Veri işleme dili, veri tabanında tutulan veriler üzerinde işlem yapılmasını sağlar.

Select: veri tabanından kayıtları seçmek için kullanılır.

Insert: veri tabanına yeni kayıt eklemek için kullanılır.

Update: veri tabanında bulunan kayıtlar üzerinde değişiklik yapmak için kullanılır.

Delete: veri tabanından kayıt silmek için kullanılır.

6.2.1.Select İfadesi

Bir tablodan kayıt çekmek için kullanılır. Select temel ve önemli bir komuttur. Kullanımı aynı olsa da select kendisine yardımcı komutlarla da kullanılabilmektedir. Şimdi select komutunun kullanımına bakalım.

AKAREGAME.person...- dbo.tblpersonel × SQLQuery6.sql - AK...ME\AkareSoft (56))*

personelID	ADI	SOYADI	TELEFON	TCKIMLIKNO	MAAS	departmanID
3	Oğuzhan	TAŞ	02127777777	12345678912	3500,00	1
4	Sinan	Engin	02127778899	12345678913	4500,00	1
5	Erhan	Can	02128889900	12345678914	2500,00	2
6	Furkan	Namlı	03129998877	12345678915	3500,00	2
7	Ceydan	Şen	02128889988	12345678916	4500,00	3
8	Sinan	Kurban	0212333444	12345678917	5500,00	1
9	Faruk	Tınaz	02128886677	12345678918	3000,00	4
10	Hatice	Can	04241112233	12345678919	3500,00	4
11	Pakize	Şener	02128882200	12345678921	6000,00	4
12	Şaheser	Soydan	02128883322	12345678922	2500,00	4

SELECT * FROM tablo_adı (“*” ifadesi size o tablodaki tüm kayıtları getirir)

SELECT sütunlar FROM tablo_adı (eğer ki özellikle bir veri çekmek istiyorsanız “*” yerine sütun ismi yazılır)

- Örneğin bir personel tablomuz olsun. Öğrencilerin tüm bilgilerini seçmek istiyorsak;

  SELECT * FROM personel

- Şimdi ise bu personellerin sadece adını seçelim;

  SELECT ad FROM personel

- Select ifadesini yanında başka komutlarla da kullanabileceğimizi söylemiştim. Şimdi bu kullanımlara örnekler verelim.

SELECT * FROM WHERE öğrenciler adı=”sinan” (bu örnekte where ile tanıştık. Where cümleye koşul eklemek için kullanılır. Yani bu komutla adı sinan olan öğrencileri seçtik)

ÇIKTI

Adı Soyadı

Sinan Engin

Sinan Kurban

SELECT * FROM personel WHERE adı=”sinan” AND soyadı=”**engin”** (bu örnekte birden fazla

şartımız olduğu için and kullandık. Yani bu komutla hem adı sinan hem soyadı engin olma şartını sağlayan öğrencileri seçtik)

ÇIKTI

Adı Soyadı

Sinan Engin

SELECT * FROM öğrenciler WHERE adı="sinan" OR soyadı="can" (bu örnekte herhangi bir şartı sağlaması yeterli olduğu için or kullandık. Yani bu komutla adı sinan olanlarda soyadı can olanlarda listelenir. İki şarttan birini taşıyor olması yeterli)

ÇIKTI

Adı Soyadı

Sinan Engin

Sinan Kurban

Erhan Can

Hatice Can

6.2.2.Insert İfadesi

Veri tabanı sistemlerinde herhangi bir tabloya yeni bir veri girişi yapılmak istenirse "insert into" ifadesi kullanılır.

INSERT INTO tablo_adı VALUES (deger1,deger2…)

- Örnekteki tablomuzda yeni bir kayıt ekleyerek bu ifadeyi kullanalım;

  INSERT INTO personel VALUES (1,'Ayşenur','Demiral',5213212343,12345678876,5000,1)

6.2.3.Update İfadesi

Veri tabanı sistemlerinde var olan kayıtları değiştirmek için kullanılırlar.

UPDATE tablo_adı SET sütun1=değer, sütun2=değer… WHERE koşul

Örnekteki tablomuzda adı Erhan olan personelin adını Ali olarak değiştirelim;

UPDATE personel SET adı='Ali' WHERE adı='Erhan'

6.2.4.Delete İfadesi

Tablo içerisinde istenilen kayıtları veya tamamını silmek için kullanılır.

DELETE FROM tablo_adı WHERE koşul (istenilen kaydı siler)

DELETE FROM tablo_adı (tabloyu siler)

6.3.VERİ KONTROL DİLİ (DCL)

Veri kontrol dili, veri tabanı içerisindeki verilere erişimleri düzenlemek için kullanılır.

Grant: yetki tanımlamak için kullanılır.

Deny: erişimi engellemek için kullanılır.

Revoke: engellenen veya verilen yetkilerin geri alınması için kullanılır.

6.3.1.Grant İfadesi

Kullanım şekli;

GRANT (all | izinler)

ON (izin alanı)

TO (kullanıcılar)

- Ali adlı kullanıcıya tablo oluşturma izni verelim;

  GRANT create table TO Ali

- Tüm kullanıcılara personel tablosundaki select yetkisi verelim;

  GRANT SELECT ON personel TO PUBLIC

6.3.2.Deny İfadesi

Kullanım şekli;

DENY (all|izinler)

ON (nesneler)

TO (public | kullanıcılar)

- Ali adlı kullanıcının tablo oluşturma yetkisini sınırlandıralım;

DENY create table TO Ali

6.3.3.Revoke İfadesi

Kullanım şekli;

REVOKE (all | izinler)

ON (nesneler)

TO | FROM public | kullanıcılar

- Ali adlı kullanıcıya daha önce vermiş olduğumuz tablo oluşturma yetkisini iptal edelim;

REVOKE create table ON personel FROM Ali

Bu kısma kadar bölümlere yapısal olarak üç gruba ayırdığımız SQL'in temel komutlarını inceledik. Şimdi SQL de kullanılan diğer komutlara bakalım.

6.4.Distinct İfadesi

Tekrar eden kayıtlar varsa bunların sadece bir kez görünmesini sağlar. Distinct ifadesini select ile kullanırız. Kullanım şekli;

SELECT DISTINCT alan_adı FROM tablo_adı

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

- Yukarıdaki müşteri tablomuzda normalde select ile şehirleri listelesek aynı şehir ismi tekrar tekrar

listelenirdi. Şimdi distinct ile yazalım;

SELECT DISTINCT city FROM müşteri

ÇIKTI

City

Germany

Mexico

UK

Sweden

6.5.Order By İfadesi

Yapılan bir sorguda sıralama yapmak için kullanılan ifadedir. Sıralama yaparken büyükten küçüğe veya küçükten büyüğe sıralanabilir. Bir önceki örnekteki müşteri tablosunu ele alarak sırasıyla order by kullanımına bakalım;

SELECT * FROM müşteri ORDER BY country (listeyi küçükten büyüğe sıralar eğer ASC kullansaydık o da bu komutla aynı işlevi görürdü)

SELECT * FROM müşteri ORDER BY country DESC (listeyi büyükten küçüğe doğru sıralar)

SELECT * FROM müşteri ORDER BY city DESC country (bu kullanımda city'i azalan şekilde country'i ise artan şekilde sıralar)

6.6.AS İfadesi

Sorgulama yaparken bazı zamanlar birden fazla tablo kullanabiliriz veya tablonun adını değiştirmek isteyebiliriz. AS ifadesi bu durumlarda kullanılır.

```
SELECT sütun_Adı1 AS takma_isim1,
sütun_adı2 AS takma_isim2,
…..
FROM tablo_adı
```

6.7.Between İfadesi

İki değer aralığındaki verileri sorgulamak için kullanılır.

SELECT * FROM tablo_Adı

WHERE referans_sütun BETWEEN ilk_değer AND son_değer

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	1	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	1	2	36 boxes	21.35

- Şimdi bu tabloyu baz alarak bir between ifadesi yazalım.

SELECT CustomerID FROM müşteri WHERE price BETWEEN 15 and 20

ÇIKTI

CustomerID

1

2

6.8.Like İfadesi

Like ifadesi joker karakterlerle birlikte kullanılır.

SELECT * FROM müşteri WHERE ProductName LIKE '%A%' (bu ifade içinde "A" geçen ProductName'leri listeler. Hepsinin içinde A harfi olduğu için çıktımızda hepsi olur)

SELECT * FROM müşteri WHERE ProductName LIKE 'A%' (bu ifade 'A' ile başlayanları seçer)

SELECT * FROM müşteri WHERE ProductName LIKE '_ _ A' (bu ifade 3 harften oluşan ve son harfi "A" olanları seçer)

7.GRUPLANDIRARAK SORGULAMA İŞLEMLERİ

Gruplama işlemi yaparken o tablo içerisinde ortak özellikleri bulunanları bir arada toplarız. Buradaki ifademiz **"group by"** olacak.

7.1.Group by İfadesi

Kullanım şekli;

SELECT sütun_adları FROM tablo_adı WHERE koşul

GROUP BY gruplama_sütun_adi

HAVING koşul ORDER BY sütun_adları

Gruplandırma İşleminde Kullanılan Fonksiyonlar

Avg() : Değerlerin ortalamasını verir.

Count() : Satır sayısını verir.

Max() : Değerler içindeki en yüksek olanı verir.

Min() : Değerler içindeki en düşük olanı verir.

Sum() : Değerlerin toplamını verir.

- Şimdi müşteri tablomuzu baz alarak örnek verelim;

SELECT COUNT(CustomerID), Country

FROM Customers

GROUP BY Country;

ÇIKTI

Count (CustomerID)	Country
1	Germany
2	Mexico
1	Sweden

1 UK

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

- Burada da bir OrderDetails tablomuz var. İlk olarak toplamı bulmak için sum ifadesini kullanalım;

SELECT SUM(Quantity)

FROM OrderDetails;

ÇIKTI

Sum (quantity)

76

SELECT AVG(Quantity)

FROM OrderDetails;

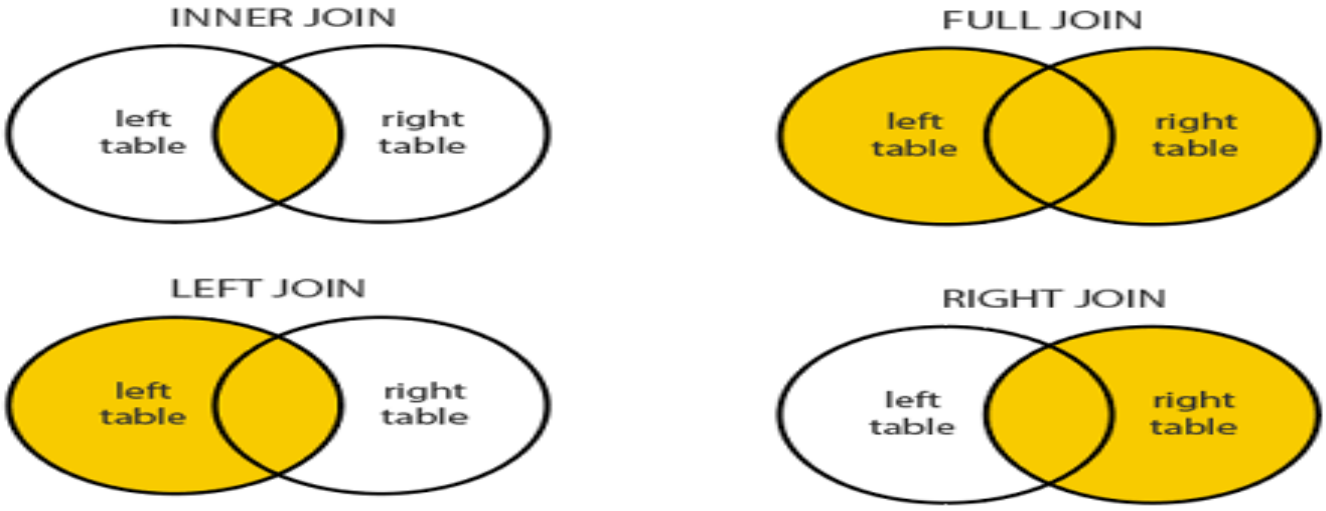
ÇIKTI

Avg (quantity)

15,2

8.TABLOLARIN BİRLEŞTİRİLMESİ (JOIN)

Aynı tür değerler için iki tablonun birleştirilerek tek bir tablo elde edilmesine birleştirme işlemi denir.



8.1.Where İle Birleştime İşlemi

Birleştirme işlemi yapılırken ortak sütunlar baz alınır. Önceki sayfalarda anahtar kavramından bahsetmiştik. Aşağıdaki tabloyu inceleyelim. OrderID ilk tablonun, CustomerID ise ikinci tablonun birincil anahtarı. İki tablodada ortak olan CustomerID var o halde birleştime yapmak için buna ihtiyacımız olacak.

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

SELECT CustomerID, CustomerName FROM Orders, Customers

WHERE Orders.CustomerID = Customers.CustomerID

8.2.Join İle Birleştirme (Inner Join)

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate

FROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID

Bu birleşmeler sonucunda şu şekilde bir tablo olacaktır;

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

8.3.Left Outer Join

Bu tür bir birleştirme işleminde ifadenin solundaki ifade belirleyici olan tablodur. Bundan dolayı diğer tabloyla ilişkisi bulunsun veya bulunmasın tüm satırlar listelenir. Sağdaki tablodan ise sadece ilişkili satırlar listelenecektir.

SELECT Customers.CustomerName, Orders.OrderID

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

ORDER BY Customers.CustomerName;

ÇIKTI

CustomerName	**CustomerID**
Alfreds Futterkiste	null
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365

8.4.Right Outer Join

Bu tür bir birleştirme işleminde ifadenin sağındaki ifade belirleyici olan tablodur. Bundan dolayı diğer tabloyla ilişkisi bulunsun veya bulunmasın tüm satırlar listelenir. Soldaki tablodan ise sadece ilişkili satırlar listelenecektir.

SELECT tablo1.sütun_adı, tablo2.sütun_adı
FROM tablo1
RIGHT JOIN tablo2 ON tablo1.ortak_sütun karşılaştırma operatörü tablo2.ortak_sütun

8.5.Full Outer Join

Bu birleştirme türünde ilişkisi bulunsun veya bulumasın hepsi listelenir.

SELECT tablo1.sütun_adı, tablo2.sütun_adı
FROM tablo1
FULL OUTER JOIN tablo2 ON tablo1.ortak_sütun karşılaştırma operatörü tablo2.ortak_sütun

9.GÖRÜNÜMLER (VİEWS)

View'ler gerçekte var olmayan sanal tablolardır. Tabloların daraltılmış veya filtrelenmiş haline ulaşmamızı sağlar. Görüntü tanımlamaya; bazı sorguları daha kolay hale getirmek, hali hazırda bulunan tablolarda var olan verilerin başka bir tablo formatında sunulması gerektiğinde ihtiyaç duyarız.

SELECT * FROM view_adı

Örneğin bazı bilgiler içeren bir birey tablomuz olsun ama biz sadece adı, soyadı ve doğum tarihi bilgilerini içeren bir vwBirey isimli bir görüntü oluşturalım;

```
Create vıew vwBirey
AS
    SELECT
        Adi,
        Soyadi,
        dTarihi
FROM Birey
GO
```

10.TETİKLEYİCİ (TRIGGER)

Bir tablo üzerinde belirleyici bir olaya bağlı olarak tetiklenip çalışan SQL kodlarına denillir. Tablo üzerindeki trigger'ları tetikleyen olaylar insert, update, delete olaylarıdır. Şimdi bir kitap kaydı ekleme örneği yapalım;

```
CREATE TRIGGER deneme
ON kitap2
AFTER insert
AS
BEGIN
    Select 'Yeni bir kitap kaydı yapıldı'
END
```

insert into kitap2 values ('…','…','…')

11.SAKLI YORDAM (STORED PROCEDURE)

Saklı yordamlar derlenmiş SQL cümlecikleridir. Birer veritabanı nesnesi oldukları için doğrudan veritabanı yöneticisi olan programda yer alırlar. Saklı yordamlar doğrudan bir tabloya bağlı olmaksızın veritabanı içinde tanımlanan belirli bir işi yapmaya yönelik kodlardır.

Create procedure procedure_adı

AS

Sql kodları

EXECUTE procedure_adı

KAYNAKÇA

- **Turgut ÖZSEVEN, VERİ TABANI YÖNETİM SİSTEMLERİ-1, Murathan Yayınevi**
- **https://www.w3schools.com/**