

SQL

(Structured Query Language)

YAPISAL SORGULAMA DİLİ

BARIŞ ARIBURNU

barisariburnu.com

barisariburnu@gmail.com

- SQL, insanların veritabanı sistemleri ile konuşmasını sağlayan popüler bir dildir.
- SQL, tüm veritabanı programlarında kullanılabilir.
- SQL bir dildir, ancak programlama dili değildir.
- Program geliştirme aşamasında SQL'den faydalanılır, tek başına bu için yeterli değildir.

- 1983'lü yıllarda IBM laboratuvarlarında yapılan çalışmalarda SQL standartları tanımlanmıştır.
- 1987'de ISO, ardında da ANSI tarafından bir standart olarak kabul edilmiştir.
- Bu standartlar çerçevesinde birçok VTYS geliştirilmiştir. (Oracle, Sybase, MS SQL, MySQL)

SQL NEDİR?

- “Structured Query Language” yani “Yapısal Sorgulama Dili” olarak açılımını yapabiliriz.
- Kullanıcı SQL kullanırken Fonksiyon ve Prosedür yazamaz.
- SQL’de şartlı ifadeler, döngüler ve karşılaştırmalar (if, case, next, do) bulunmaz.
- Bu eksikliği gidermek için Oracle PL/SQL, Microsoft SQL Server, Sybase’de T-SQL dilini geliştirmiştir.

- Oracle tarafından kullanılan PL/SQL komutları ile T-SQL birçok noktada hemen hemen aynıdır.
- SQL'de kullanılmayan prosedür, fonksiyon, şartlı ifadeler , döngüler ve karşılaştırmalar kullanılabilir.
- PL/SQL ve T-SQL if, case, for..next gibi programlama için gereken işlemleri kullanmayı olanağını sunar.

- PL/SQL ve T-SQL sadece içerisinde SQL komutları kullanabilen bir dildir.
- Yani SQL'in yapısını deęiřtirmemiřlerdir.
- Veritabanı programları tüm uygulamalarda SQL kullanmayı esas almıřtır.
- Veritabanı programlarını kullanarak yaptığımız tüm işlemlerin arkasında SQL komutları çalıştırılır.

SQL'in veritabanı işlemleri için kullanılan komutları 5 kategoride toplanabilir.

- Veri sorgulama komutları
- Tabloya veri ekleme, değiştirme ve silme komutları
- Veritabanı nesnelere oluşturma, değiştirme ve silme komutları
- Veritabanına ve veritabanı nesnelere erişimi kontrol etme komutları
- Veritabanının tutarlılığını ve bütünlüğünü koruma komutları

- SQL ile birçok yerde kastedilen şey, ANSI'nin 1992 yılında yayınladığı standarttır.
- Ancak SQL bir programlama dili değildir.
- Bir kullanıcı ara yüzü tanımlayamaz ya da bir dosya yönetimi yapamaz.

SQL temelde 3 alt ifade grubundan oluşur.

- 1- Veri Tanımlama Dili
 - Data Defination Language - DDL
- 2-Veri İşleme Dili
 - Data Manuplation Language - DML
- 3-Veri Kontrol Dili
 - Data Control Language - DCL

Veri Tanımlama Dili

- Tablo, trigger, view gibi veritabanı nesnelere tanımlanır.
- 3 temel ifade içerir: CREATE, ALTER, DROP
- Create; nesne tanımlar.
- Alter; nesne üzerinde değişiklik yapar.
- Drop; nesne siler.

Veri İşleme Dili

- Veri üzerinde düzenlemeler yapar.
- INSERT; tabloya veri ekler.
- UPDATE; tablodaki verileri günceller.
- DELETE; tablodaki verileri siler.
- *SELECT; tablodaki veri seçer ve raporlar.

*Özel olarak bahsedilecektir.

Veri Kontrol Dili

- VTYS'de tanımlı roller ve kullanıcılar için ifade ve nesne kullanma izni tanımlar.
- Temel 3 ifadeden oluşur: GRANT, REVOKE, DENY
- Grant; erişim izni verir.
- Revoke; erişim iznini kaldırır.
- Deny; erişim engeller.



CREATE

TEMEL SQL KOMUTLARI

- Database, tablo, index, prosedür vb. veritabanı nesnelərini oluřturmak için kullanılan komuttur.

CREATE DATABASE database_name

[ON {DEFAULT | database_device} [=size]

[, database_device [=size]]...]

[LOG ON database_device [=size]

[, database_device [=size]]...] [FOR LOAD]

ON

- Yaratılacak olan veritabanının hangi sürücü üzerinde yer alacağını belirten parametredir.
- Aynı zamanda bu sürücü üzerinde **size (alan, ölçü)** parametresi ile veritabanının ne kadar yer kaplayacağı belirtilmiş olur.
- Tanımlanmazsa **5MB**'lık veritabanı yaratır.
- Bu parametre içerisinde birkaç sürücü ismi kullanılarak veritabanının birkaç device üzerinde yer alması sağlanabilir.

LOG ON

- Yaratılacak olan veri tabanının çalışma anındaki bilgilerini tutan log dosyalarının hangi sürücü üzerinde yer alacağını belirten parametredir.
- Aynı zamanda bu sürücü üzerinde **size (alan, ölçü)** parametresi ile veritabanı log'unun ne kadar yer kaplayacağı belirtilmiş olur.
- Tanımlanmazsa default sürücü üzerinde bir log dosyası açar.
- Bu parametre içerisinde birkaç sürücü ismi kullanılarak veri tabanı log'unun birkaç device üzerinde yer alması sağlanabilir.

CREATE DATABASE firat_oracle

- firat_oracle isimli boş bir veritabanı yaratılmış oldu.
- Bu isim disk üzerinde fiziksel bir isimdir ve yaratılacak tablolar bu isimli veritabanının altına yaratılacaktır.

CREATE TABLE tablo_adi

(sütun_adi_1) veri_tipi,
(sütun_adi_2) veri_tipi,
.....);

- Tabloda tanımlanan her bir alan bilgisinden sonra alanları birbirinden ayırt edebilmek için virgül (,) işareti konulmalıdır.
- Sorgunun bittiği ve çalıştırılması gerektiğini belirtmek için ise noktalı virgül (;) işareti konulmalıdır.

```
CREATE TABLE uyeler (  
    OkulNo        number(8)    NOT NULL,  
    TamAdi        char(30),  
    DogumTarihi   DATE  
);
```

- NOT NULL ifadesi; tabloya bilgi yazılırken bu alana muhakkak bilgi yazılması gerektiği durumlarda kullanılır ve alanın boş bırakılmayacağını bildirir.

ALTER TABLE

TEMEL SQL KOMUTLARI

- Önceden oluşturulmuş tablonun yapısını değiştirmek için ALTER TABLE komutu kullanılır.
- Tablolara yeni bir alan ve kısıtlama eklenebilir, var olan alan ve kısıtlamaların durumu değiştirilebilir veya tablodan kısıtlamalar silinebilir.

ALTER TABLE uyeler

```
MODIFY TamAdi char(25);
```

- Uyeler tablosunun TamAdi alanının uzunluğu 30'dan 25'e düşürülmüştür.

ALTER TABLE uyeler

```
ADD Bolum char(15);
```

- Uyeler tablosuna 15 karakter uzunluğunda Bolum alanı eklenmiştir.

DROP TABLE

TEMEL SQL KOMUTLARI

DROP TABLE table_name [CASCADE CONSTRAINTS]

- Tabloyu silmek için DROP deyimi kullanılır.
- Cascade Constraints ifadesiyle; ilişkili tablolardaki master tablo silinince detay tablolarında otomatik olarak silinmesi sağlanır.
- Bu ifade kullanılmazsa diğer tablolarla ilişkisi bulunan bir tablo silinemez. Ancak ilişkiler kaldırıldıktan sonra silinebilir.



DESCRIBE

TEMEL SQL KOMUTLARI

- Tablonun yapısını görmek için DESC[İRİBE] komutu kullanılır.
- Bu komut ile tanımlanan tabloların alan bilgileri alınır.
- Tablo alanları, alanların veri tipleri unutulduğunda veya kontrol edilmek istenildiğinde sıkça kullanılan bir komuttur.

desc uyeler;

- Bize uyeler tablosundaki aşağıdaki bilgileri döndürür.
 - ALAN ADI
 - NULL ?
 - TİP

INSERT

TEMEL SQL KOMUTLARI

- Tabloya bilgi eklemek için kullanılan komuttur.

INSERT INTO table_name **VALUES** (deger_1, deger_2)

- Sütunlara yazılan alanların değerleri sayılardan oluşuyorsa, sayı direk yazılmalı, karakterlerden oluşuyorsa ifade tek tırnak işareti arasına alınarak yazılmalıdır.

INSERT INTO table_name (alan_adi_1, alan_adi_2)
VALUES (deger_1, deger_2)

INSERT INTO uyeler (11541423,'Barış Arıburnu',
24/12/1991)

INSERT INTO uyeler (OkulNo, TamAdi, DogumTarihi)
VALUES (11541423,'Barış Arıburnu',24/12/1991)

- Yukarıda yazmış olduğumuz iki INSERT komutu da aynı işlemi yapmaktadır.

*Detaylı olarak açıklanacaktır.

SELECT

TEMEL SQL KOMUTLARI

- Veritabanındaki tablo veya tablolardan istenilen özellikteki verileri seçip listeleme için kullanılan komuttur.

```
SELECT [ DISTINCT | ALL ] <sütun(lar)>  
FROM <tablo adı (lar)>  
[ WHERE <şart (lar)> ]  
[ GROUP BY <sütunlar> ]  
[ HAVING <grup kısıtlaması> ]  
[ ORDER BY <sütun (lar) [ASC | DESC]> ]
```

* Sunumda detaylara girilecektir.

ORDER BY

TEMEL SQL KOMUTLARI

- Bu komut ile belirtilen sütuna göre artan veya azalan bir sıralama ile sorgulama yapılabilir.
- ASC; kullanarak küçükten büyüğe doğru artan sıralama yapılabilir.
- DESC; kullanılarak büyükten küçüğe doğru azalan sıralama yapılabilir.
- Default sıralama tipi ASC olduğu için sıralama yaparken ASC kullanılmak zorunda değildir.

SELECT * FROM uyeler ORDER BY OkulNo DESC

- Aynı anda birden fazla alanda sıralama yapılabilir.

SELECT * FROM uyeler ORDER BY OkulNo, TamAdi

DISTINCT

TEMEL SQL KOMUTLARI

- SQL'de tablo içinde bazı alanlarda birbirinin aynı olan kayıtlar bulunabilir.
- Aynı satırların listeleme esnasında bir kez yazılması için DISTINCT kullanılır.

SELECT DISTINCT TamAdi FROM uyeler



UPDATE

TEMEL SQL KOMUTLARI

- Tabloda bir kayıt veya kayıtlar değiştirilmek istenirse UPDATE komutu kullanılır.

UPDATE tablo_adi

SET sutun_adi_1 = deger_1, sutun_adi_2 = deger_2

WHERE kosul

DİKKAT !

UPDATE komutu kullanılırken WHERE şartının mutlaka yazılması gerektiğini unutmayın. Aksi takdirde tüm kayıtlarınızı aynı anda güncellemiş olursunuz.

Yedek bilginiz yoksa bilgilere yeniden ulaşmak söz konusu değildir.

DELETE

TEMEL SQL KOMUTLARI

- Tablodaki bir kayıt veya kayıtlar silinmek istenirse DELETE komutu kullanılır.

DELETE FROM tablo_adi **WHERE** kosul

DİKKAT !

DELETE komutu kullanılırken WHERE şartının mutlaka yazılması gerektiğini unutmayın. Aksi takdirde tüm kayıtlarınızı silmiş olursunuz.

Yedek bilginiz yoksa bilgilere yeniden ulaşmak söz konusu değildir.

SQL FONKSIYONLARI

- SUM (Topla) Fonksiyonu
 - Belirtilen sütundaki bütün değerlerin toplamını alır.
 - **Select SUM(sutun_adi) FROM tablo_adi**
- AVG (Ortalama) Fonksiyonu
 - Belirtilen sütundaki bütün değerlerin ortalamasını alır.
 - **Select AVG(sutun_adi) FROM tablo_adi**
- MAX (En Büyük) Fonksiyonu
 - Belirtilen sütundaki en büyük değeri verir.
 - **Select MAX(sutun_adi) FROM tablo_adi**

- MIN (En Küçük) Fonksiyonu

- Belirtilen sütundaki en küçük değeri verir.
- **SELECT MIN(sutun_adi) FROM tablo_adi**

- COUNT (Say) Fonksiyonu

- Tablo içinde, her hangi bir sayma işlemi gerçekleştirmek için kullanılır. COUNT fonksiyonu sorgu sonucu dönen kayıt sayısını verir. Eğer parametre (*) olarak girilirse tablodaki tüm kayıt sayısını verir.
- **SELECT COUNT(sutun_adi) FROM tablo_adi**

GROUP BY

SQL'DE GRUPLANDIRMA

Bir alana göre kayıtları gruplandırmak için kullanılır.

```
SELECT [ DISTINCT | ALL ] <sütun(lar)>  
FROM <tablo adı (lar)>  
[ WHERE <şart(lar)> ]  
[ GROUP BY <sütunlar> ]
```

HAVING

SQL'DE GRUPLANDIRMA

Gruplandırarak kümeleme fonksiyonu kullanılırken, sorguda bir koşulunda verilmesi gerekiyorsa kullanılır.

Bu durumda grup üzerindeki hesaplamalarla ilgili koşul belirtirken HAVING sözcüğü kullanılır.

HAVING yardımcı sözcüğü “GROUP BY” yardımcı sözcüğü ile gruplanan kayıtlar üzerinde kısıtlama yapma işine yarar.

```
SELECT [ DISTINCT | ALL ] <sütun(lar)>  
FROM <tablo adı (lar)>  
[ WHERE <şart(lar)> ]  
[ GROUP BY <sütunlar> ]  
[ HAVING <grup kısıtlaması> ]
```

Örneğin; İşçilerin maaşlarını aylık bazda gruplayalım ve maaşı 1000 üzerindeki getirelim.

SQL'DE JOIN

- Veri tabanı kayıtları oluşturulurken bütün bilgiler bir tabloda değil de, birkaç tablo üzerinde tutulur.
- Bu verilerin daha düzenli olmasını, gereksiz veri tekrarlarının engellenmesini ve veri yönetimini kolaylaştırır.

```
SELECT * FROM tablo_adi_1, tablo_adi_2  
WHERE tablo_adi_1.birincil = tablo_adi_2.foreign
```

VIEWS (TABLO GÖRÜNÜMÜ)

SQL'DE YÖNETİMSSEL FONKSİYONLAR

- Tablo görünümleri veri tabanında tanımı olan tablolardan sorgulama sonucunda elde edilir.
- Tabloların tersine fiziksel bir yer tutmazlar.
- Uygulamalarda pek çok pratiklik sağlarlar.
- Görüntü oluşturabilmek için Select cümlesi kullanılması gerekir.

- Bir görüntü bir yada daha fazla tablodan oluşturulabileceği gibi bir başka görüntüden de oluşturulabilir.

CREATE VIEW <görüntü adı> **AS** <select sorgusu>

CREATE USER

SQL'DE YÖNETİMSEL FONKSİYONLAR

- Kullanıcı veritabanı nesnelерinin sahibidir.
- Kullanıcılar, nesneleri oluşturur, kullanır ve silerler.
- Oracle veritabanı ilk kurulduğunda standart olarak üç kullanıcı tanımlanır.

- Bunlardan ilki SYS'dir.
 - Tüm nesnelere oluşturma hakkına sahiptir ve diğer bütün kullanıcıların nesnelere erişebilir.
 - SYS kullanıcısının ilk şifresi “change_on_install” dur.
- İkinci kullanıcı SYSTEM'dir.
 - Veri sözlüğünü kullanma ve diğer kullanıcıların nesnelere erişme hakkına sahiptir.
 - İlk şifresi “manager” dır.
- Üçüncüsü de SCOTT kullanıcısıdır..
 - Veritabanına başlangıçta yüklenen demo tabloların sahibidir.

CREATE ROLE

SQL'DE YÖNETİMSSEL FONKSİYONLAR

- Rol veritabanındaki hakların toplanmış haline denir.
- Veritabanı yöneticisi rolleri kullanarak sistemin güvenliğini daha kolay sağlayabilir.
- Roller Oracle tarafından önceden tanımlanmış roller ve kullanıcı tanımlı roller olarak iki şekilde düşünülebilir.
- Oracle tarafından önceden tanımlanmış roller 5 tanedir. (CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE, IMP_FULL_DATABASE)

CREATE ROLE tablo_incele

GRANT SELECT ON personel **TO** tablo_incele

- tablo_incele isimli bir rol oluşturulmuştur.
- Bu role personel tablosu üzerinde listeleme işlemi yapma hakkı verilmiştir.
- Böylece rolün atandığı kullanıcı personel tablosu üzerinde yalnızca “SELECT” komutunu çalıştırabilir.

CREATE SEQUENCE

SQL'DE YÖNETİMSSEL FONKSİYONLAR

- Sıra, sıralı olarak artan alanlar için veritabanında tutulan nesnedir.
- Sıra tekrarlamayan sayısal bilgi elde etmek için kullanılır. Örneğin; sicil numarası, öğrenci numarası
- Örneğin; 1'den başlayan ve birer birer artan bir sıra oluşturmak için;

```
CREATE SEQUENCE sicilno START WITH 1  
INCREMENT BY 1
```

DİKKAT !

Oracle gibi bazı veritabanı programlarında her bir SQL cümleciğinin sonuna noktalı virgül konulması gerekir.

Microsoft tabanlı SQL-SERVER gibi bazı veritabanı sistemlerinde bu türden bir zorunluluk bulunmamaktadır.

TEŞEKKÜRLER..

Barış Arıburnu

barisariburnu.com

barisariburnu@gmail.com

Kaynak: Yrd. Doç. Dr. Zehra ALAKOÇ BURMA (Veritabanı
Yönetim Sistemleri ve SQL / PL-SQL / T-SQL)