**SQL Sorguları**

İnsert – data ekleme

1. Yöntem

INSERT into

users([name], [surname], [email], [phone], [password], [savedate], [status])

VALUES(

'Erkan',

'Bilmem',

'erkan@mail.com',

5436667788,

'12345',

'2025-03-08 09:27:12',

1

)

1. Yöntem

INSERT into

users

VALUES(

'Ali',

'Bil',

'ali@mail.com',

5436667755,

'12345',

'2025-03-08 09:34:12',

1

)

1. Yöntem

INSERT into

users([surname], [name], [email], [password], [phone])

VALUES(

'Bilirler',

'Mehmet',

'mehmet@mail.com',

'12345',

5647778866

)

Select – data çekme

1. Tüm users bilgilerini getiren sorgu

SELECT \* FROM users

1. ilk eklenen 100 datayı getir

SELECT TOP(100) \* FROM users

1. belirli sütun değerlerini getirme

SELECT TOP(100) [uid], [name], [surname] FROM users

1. gelen değerlerin sütun adlarını değiştirme

SELECT TOP(100) [uid], [name] as username, [surname] as usersurname FROM users

1. (=) uid değeri 50 olan kişiyi getir

SELECT \* FROM users WHERE uid = 50

1. (!=) status değeri 1 olmayanları getir

SELECT \* FROM users WHERE [status] != 1

1. ( > ), ( >= ) uid değeri 100 den büyük olan kullanıcıları getir

SELECT Top(50) uid, name, surname FROM users WHERE uid > 100

1. (<), (<=) uid değeri 100’den küçük olan kullanıcıları getir

SELECT Top(50) uid, name, surname FROM users WHERE uid < 100

1. (between) 2024 yılı ile 2025 arasında kayıt yapan kullanıcıları listele

SELECT \* FROM users WHERE savedate BETWEEN '2024-07-15' and GETDATE()

1. ( like ) name değeri ‘ali’ olan kullanıcıları getir

SELECT \* FROM users WHERE [name] LIKE 'ali'

1. (like ‘%ali%’) name değeri içinde ‘ali’ geçen kullanıcıları getir.

SELECT \* FROM users WHERE [name] LIKE '%ali%'

1. (like ‘%ali%’ or ) name değeri yada surname değeri içinde ‘ali’ geçen kullanıcıları getir.

SELECT \* FROM users WHERE [name] LIKE '%ali%' or [surname] LIKE '%ali%'

1. (like ‘%al%’ and) name değeri ve surname değeri içinde ‘al’ geçen kullanıcıları getir.

SELECT \* FROM users WHERE [name] LIKE '%al%' and [surname] LIKE '%al%'

1. (and) email ‘ali@mail.com’ ve şifresi ‘12345’ olan kullanıcıyı getir.

SELECT \* FROM users WHERE [email] = 'espiaggia0@g.co' and [password] = '12345' and [status] = 1

1. (in) uid değeri (5,6,15,28) olan kullanıcı bilgileri getir.

SELECT \* FROM users WHERE email IN (5,6,15,28)

SELECT \* FROM users WHERE email IN ('ali@mail.com', 'aholmea@google.pl', 'mrepp29@salon.com')

SELECT \* FROM users WHERE uid

IN

( SELECT uid FROM users WHERE name LIKE '%eb%' or surname LIKE '%eb%' )

1. (order by asc, desc) user tablosu içindeki listede ilk eklenenden son eklenene doğru sırala

SELECT \* FROM users ORDER BY uid ASC

SELECT \* FROM users ORDER BY uid DESC

SELECT \* FROM users ORDER BY savedate DESC

SELECT \* FROM users WHERE [name]

LIKE '%el%' or [surname] LIKE '%el%'

ORDER by savedate DESC

1. (top) users tablosu içinde belirlediğim sayı kadar satır getir.

SELECT top 5 \* FROM users

1. (sayfalama) users tablosu içindeki dataları 10’arlı satırlar halinde sayfala

SELECT \* FROM users

ORDER BY uid ASC

OFFSET 0 ROWS

FETCH NEXT 10 ROWS ONLY

1. (Count) users tablosu içindeki satır sayısını verir

SELECT COUNT(\*) as userCount FROM users

1. (Month) users tablosu içindeki verilerin şubat ayında eklenenlerini getir.

SELECT \* FROM users WHERE MONTH(savedate) = 2 and DAY(savedate) = 3

1. (max) users tablosunda en yüksek yaşa sahip değeri getir.

SELECT MAX(age) as maxage FROM users

1. (min) users tablosunda en düşük yaşa sahip değeri getir.

SELECT MIN(age) as maxage FROM users

1. (AVG) users tablosundaki kişilerin yaş ortalamasını getir.

SELECT AVG(age) as avgAge FROM users

1. (len) users tablousu içindeki string değerlerin karakter sayısını verir.

SELECT uid, [name], [surname], LEN([surname]) FROM users

1. (group by) users tablosu içindeki kişileri status değerlerine göre grupla.

SELECT AVG(age) avgAge, COUNT(\*) count, [status] FROM users GROUP BY [status]

1. (DateAddd)

SELECT \* from users WHERE savedate BETWEEN DATEADD(MONTH, -1, GETDATE()) and GETDATE()

Join

1. (inner join) bire bir eşleme

SELECT \* FROM product

INNER JOIN category ON product.cid = category.cid

SELECT \* FROM product pr

INNER JOIN category ct ON pr.cid = ct.cid

SELECT

pr.pid, pr.cid, pr.title, pr.detail, pr.price, ct.name as catName

FROM product pr

INNER JOIN category ct ON pr.cid = ct.cid

SELECT

pr.pid, pr.cid, pr.title, pr.detail, pr.price, ct.name as catName

FROM product pr

INNER JOIN category ct ON pr.cid = ct.cid WHERE ct.cid = 1

SELECT

pr.pid, pr.cid, pr.title, pr.detail, pr.price, ct.name as catName

FROM product pr

INNER JOIN category ct ON pr.cid = ct.cid WHERE ct.cid = 1

ORDER BY pr.pid ASC

OFFSET 0 ROWS

FETCH NEXT 10 ROWS ONLY

Normalize tablosu sorgusu (join)

SELECT

pro.pId, pro.title, pro.detail, pro.price,

ca.cId, ca.name

FROM product as pro

INNER JOIN product\_category as procat

ON pro.pId = procat.pid

INNER JOIN category as ca

ON ca.cId = procat.cid

1. Left Join – sol tarafta(ilk çağrılan tablo) ile sağ tarafta son(çağrılan tablo) id’leri uyuşmasa bile bilgileri getir.

SELECT TOP(10) \* FROM product as pro

LEFT JOIN category as cat

ON pro.cid = cat.cId

metin, sayı, numara, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Right Join – sağ taraftaki tablo ile sol tarafta çağrılan tablonun uyuşmasa bile en az 1 satırla tamamlanması olayı.

SELECT TOP(100) \* FROM product as pro

RIGHT JOIN category as cat

ON pro.cid = cat.cId

metin, sayı, numara, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Full Join \* sol taraf yada sağ taraf id olarak uyuşmasa bile tüm dataları getirir.

SELECT TOP(1500) \* FROM product as pro

FULL JOIN category as cat

ON pro.cid = cat.cId

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Update – data düzenle

1. Tüm users tablosu içinde age değerini 25 yap

UPDATE users SET [age] = 25

UPDATE users SET [age] = FLOOR( RAND() \* 42) + 18

WHERE [uid] = 1

DECLARE @i int = 0;

WHILE @i<1004

BEGIN

SET @i = @i + 1;

UPDATE users SET [age] = FLOOR( RAND() \* 42) + 18

WHERE [uid] = @i;

END

Delete – data silme

1. Uid değeri 1 olan kullanıcıyı sil

delete from users WHERE uid = 1

1. Belirli zaman aralığındaki dataları sil

delete FROM users WHERE savedate BETWEEN '2025-02-20 17:00:00' and GETDATE()

View – Hazır SQL

1. Uzun sql sorgularında sql içinde daha önceden planlanmış sql sorgularıdır.
2. View Oluşturma – select sırasında gelen veriler aynı sütün isimlerine sahip olduklarında uygulama hata verecektir, aynı isme denk gelen column isimlerinin değişmesi gerekmektedir.

CREATE VIEW pro\_cat\_join

AS

SELECT

pro.pId, pro.cid, pro.title, pro.detail, pro.price, cat.name, cat.[status]

FROM product as pro

INNER JOIN category as cat

ON pro.cid = cat.cid

1. View Silme

DROP VIEW pro\_cat\_join

Functions – Method oluşturma

1. Hazır sql gövdeleri kurarak tekrarlı bir şekilde lazım olduğu yerde tetikleyici ile tetiklenmesi ve kullanılmasına olanak tanır.

Fonksiyon oluşturma

**CREATE** **FUNCTION** dbo.Fnc\_Sum(**@num1** int, **@num2** int)

**RETURNS** **INT**

**AS**

**BEGIN**

**Declare** **@sum** int;

**SET** **@sum** = **@num1** + **@num2**;

**return** **@sum**;

**END**

**USE** project;

**CREATE** **FUNCTION** dbo.Fnc\_NameConcat(**@uid** int)

**RETURNS** **Varchar**(150)

**AS**

**BEGIN**

**Declare** **@NameConcat** **Varchar**(150);

**SET** **@NameConcat** = 'Ali Bilmem';

**return** **@NameConcat**;

**END**

;

**USE** master;

**CREATE** **FUNCTION** dbo.Fnc\_NameConcat(**@uid** int)

**RETURNS** **Varchar**(150)

**AS**

**BEGIN**

**Declare** **@NameConcat** **Varchar**(150);

**SET** **@NameConcat** = (**select** **concat**('Sn. ',name, ' ' ,surname) **from** users **where** uid = **@uid**);

**return** **@NameConcat**;

**END**

Procedure

1. Dışarıdan parametre alabilen ve almış olduğu parametreler ile sorgular yapıp geriye tablo döndürme yeteneğine olan çok işlevsel bir özelliktir.

**CREATE** **PROCEDURE** dbo.Pro\_ProCat(**@page** int)

**AS**

**SELECT** \* **FROM** product *p*

**Inner** **JOIN** product\_category *pc*

**ON** *p*.pId = *pc*.pid

**INNER** **JOIN** category *c*

**ON** *c*.cId = *pc*.cid

**Order** **by** *p*.pId **asc**

OFFSET **@page** **ROWS**

**FETCH** **NEXT** 10 **ROWS** **ONLY**;

**EXEC** dbo.Pro\_ProCat 0

**CREATE** **PROCEDURE** dbo.Pro\_ProCat(**@page** int)

**AS**

**declare** **@count** int;

**SET** **@count** = 10;

**SET** **@page** = **@page** \* **@count**;

**SELECT** \* **FROM** product *p*

**Inner** **JOIN** product\_category *pc*

**ON** *p*.pId = *pc*.pid

**INNER** **JOIN** category *c*

**ON** *c*.cId = *pc*.cid

**Order** **by** *p*.pId **asc**

OFFSET **@page** **ROWS**

**FETCH** **NEXT** **@count** **ROWS** **ONLY**;

Trigger - Tetikleyici

1. Bir tabloya ekleme, düzenleme yada silme işlemleri olduğunda bunların öncesinde yada sonrasında çalışmasını istediğimiz sql kodlarını tasarladığımız sql özellikleridir.

**CREATE** **OR** **ALTER** **TRIGGER** AddStock **ON** project.dbo.product

**AFTER** **INSERT**

**AS**

**BEGIN**

**declare** **@pid** int

**SET** **@pid** = (**select** **top** 1 pid **from** product **order** **by** pid **desc**)

**INSERT** **INTO** stock **values**(**@pid**, 10)

**END**

**CREATE** **TRIGGER** UpdateStock **ON** project.dbo.product

**AFTER** **UPDATE**

**AS**

**BEGIN**

**DECLARE** **@pid** **INT**;

**SELECT** **@pid** = pId **FROM** inserted;

**UPDATE** stock **SET** **count** = 10 **WHERE** pId = **@pid**;

**END**;