

GÖRÜNÜM SORGULARI ve UNION İFADELERİ

Öğretim Görevlisi A. Berika VAROL MALKOÇOĞLU

İçindekiler

- Görünüm Sorguları
- Union ifadeleri

Görünümün Sorguları (View)

- View ifadesi yeni bir görünüm sorgusu oluşturur.
- Kendilerine ait herhangi bir veriyi depolamayan ancak diğer tablolarda depolanan verileri görüntüleyen sanal tablolardır.
- Farklı veya aynı tablolardaki sütunları bir araya getirmeye yarar.
- Görünümün adı veritabanındaki başka bir isimle çakışmamalı.
- Görünümler, yetkili kullanıcılara yalnızca amaçlanan verileri göstererek veritabanının güvenliğini artırır.
- Hassas verileri gizlerler.

Görnümün Sorguları (View)

- View sorgusunun syntaxı

```
CREATE VIEW view_name AS  
    sql_statement;
```

Görnümün Sorguları (View) Neden Kullanılır?

- Tabloları kullanmak yerine uygulamanızda orijinal tablo yapısının VIEWS'lerini kullanmanız önerilir.
- Bu, DB'nizi yeniden düzenlediğinizde, eski kodunuzun uygulamayı bozmadan görünüm aracılığıyla orijinal şemayı görmesini sağlar.
- GÖRÜNÜMLER yeniden kullanılabilirliği artırır.
- Tekrar tekrar birleştirme içeren karmaşık sorgular oluşturmanız gerekmez.
- Tüm karmaşıklık tek satırlık sorguya dönüştürülür.

Görnümün Sorguları (View) Neden Kullanılır?

- Yazım hatası olasılığını ortadan kaldıracak ve kodunuz daha okunaklı olacaktır.
- VIEWS veri güvenliğinde yardım eder.
- Ör; kullanıcılara yalnızca yetkili bilgileri göstermek ve kredi kartı numaraları gibi hassas verileri gizlemek için görünümleri kullanabilirsiniz.

Görnümün Sorguları (View)

- Sipariş başına toplam satışları temsil eden bir görünüm oluşturmak için;

orderdetails
* orderNumber
* productCode
quantityOrdered
priceEach
orderLineNumber

```
CREATE VIEW salePerOrder AS  
SELECT orderNumber,  
SUM(quantityOrdered * priceEach) total  
FROM orderDetails  
GROUP BY orderNumber ORDER BY total DESC;
```

Görnümün Sorguları (View)

```
1 • use eczane;  
2  
3 • CREATE VIEW musteribilgi AS  
4   SELECT ad,soyad from musteris;  
5  
6 • select * from musteribilgi;
```

Result Grid

	ad	soyad
▶	Aslı	Can
	Ayşe	Al
	Fatma	Saygın
	Emine	Yılmaz
	NULL	NULL

Görnümün Sorguları (View)

```
8 • CREATE VIEW muster_iurun_view AS
9 SELECT muster_i.ad, muster_i.soyad, muster_i.urun_adet, urun.ad as 'ürün adı'
10 from muster_i inner join urun on muster_i.urun=urun.urun_no;
11
12 • select * from muster_iurun_view;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	ad	soyad	urun_adet	ürün adı
▶	Aslı	Can	2	Aspirin
	Ayşe	Al	1	Katarin
	Fatma	Saygın	1	Bepanthen
	NULL	NULL	1	Bepanthen
	Emine	Yılmaz	1	aferin

SCHEMAS

Filter objects

- ▶ beykoz
- ▶ e_ticaret
- ▶ e_ticaret_2
- ▶ e_ticaret_3
- ▼ eczane
 - ▶ Tables
 - ▼ Views
 - ▶ muster_iurun_view
 - ▶ musteribilgi
 - ▶ Stored Procedures
 - ▶ Functions
- ▶ kitaplik
- ▶ market
- ▶ myo
- ▶ normalizasyon
- ▶ sakila
- ▶ sakli_yordamlar
- ▶ sys
- ▶ world

Görnümün Sorguları (View)

```
use kitaplik;
```

```
CREATE VIEW kitap_yazar AS
```

```
SELECT kitap.kitap_ad, yazar.yazar_ad,yazar.yazar_soyad, kategori.kategori_ad
```

```
FROM kitap
```

```
INNER JOIN yazar ON kitap.yazar_id = yazar.yazar_id
```




```
INNER JOIN kategori ON kitap.kategori_id = kategori.kategori_id;
```

Görnümün Sorguları (View)

9

10 • `select * from kitap_yazar;`

<

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	kitap_ad	yazar_ad	yazar_soyad	kategori_ad
▶	kürk mantolu madonna	sabahattin	ali	türk klasikleri
	kuyucaklı yusuf	sabahattin	ali	türk klasikleri
	sineklı bakkal	halide edip	adivar	türk klasikleri
	ateşten gömlek	halide edip	adivar	türk klasikleri
	handan	halide edip	adivar	türk klasikleri
	mor salkımlı ev	halide edip	adivar	türk klasikleri
	kalp ağrısı	halide edip	adivar	türk klasikleri
	aziz istanbul	yahya kemal	beyatlı	türk klasikleri
	ince memed	yaşar	kemal	türk klasikleri
	kuşlar da gitti	yaşar	kemal	türk klasikleri
	ölmez otu	yaşar	kemal	türk klasikleri
	sarı sıcak	yaşar	kemal	türk klasikleri
	ihtiyar balıkçı	ernest	hemingway	dünya klasik...
	silahlara veda	ernest	hemingway	dünya klasik...
	çanlar kimin için çalışıyor	ernest	hemingway	dünya klasik...
	XXXX KİTABI	orhan	pamuk	fantastik

SCHEMAS

Filter objects

▶	beykoz
▶	e_ticaret
▶	e_ticaret_2
▶	e_ticaret_3
▶	eczane
▼	kitaplik
▶	Tables
▼	Views
▶	kitap_yazar
▶	Stored Procedures
▶	Functions
▶	market
▶	myo
▶	normalizasyon
▶	sakila
▶	sakli_yordamlar
▶	sys
▶	world

Görnümün Sorguları (View)

- Kitaplik veritabanındaki oluşturulan tüm tabloları görüntülemek için;

```
11
12 • show tables;
```

Result Grid		Filter Rows:
	Tables_in_kitaplik	
▶	kategori	
	kitap	
	kitap_yazar	
	kutuphane	
	yazar	

```
12 • show full tables;
```

Result Grid		Filter Rows:
	Tables_in_kitaplik	Table_type
▶	kategori	BASE TABLE
	kitap	BASE TABLE
	kitap_yazar	VIEW
	kutuphane	BASE TABLE
	yazar	BASE TABLE

Görünümün Sorguları (View)




- Mevcut bir görünümü güncellemek için OR REPLACE komutu kullanılır.
- Eğer bir görünüm daha önceden oluşturulmamışsa ve OR REPLACE komutu kullanılıyorsa bunun bir etkisi olmaz.

```
CREATE OR REPLACE VIEW view_name AS  
    sql_statement;
```

Görnümün Sorguları (View)

```
9 • CREATE OR REPLACE VIEW kitap_yazar AS
10 SELECT kitap.kitap_ad, kategori.kategori_ad
11 FROM kitap
12 INNER JOIN kategori ON kitap.kategori_id = kategori.kategori_id;
13
14
15 • select * from kitap_yazar;
```

<

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	kitap_ad	kategori_ad
▶	kürk mantolu madonna	türk klasikleri
	kuyucaklı yusuf	türk klasikleri
	ince memed	türk klasikleri
	sinekli bakkal	türk klasikleri
	kuşlar da gitti	türk klasikleri
	aziz istanbul	türk klasikleri
	ateşten gömlek	türk klasikleri
	handan	türk klasikleri

Görnümün Sorguları (View)

- Bir view ifadesini silmek için;

DROP VIEW view_name;

Union ve Union ALL

- UNION, birleştirme tiplerinin dışında, birden fazla tabloda sorgu gerçekleştirmek amacıyla kullanılan yöntemlerden biridir.
- UNION işleminde birleştirilmek istenen veri tablolarının belirlenen alanlar (kolonlar) aynı sayıda ve aynı veri tipinde olmalıdır.
- Alanların belirlenmesi işlemi SELECT ile sağlanır.
- Aynı satırdan birden fazla olsa dahi sonuç tablosunda oluşabilecek satırları göstermek için UNION ALL kullanılır.

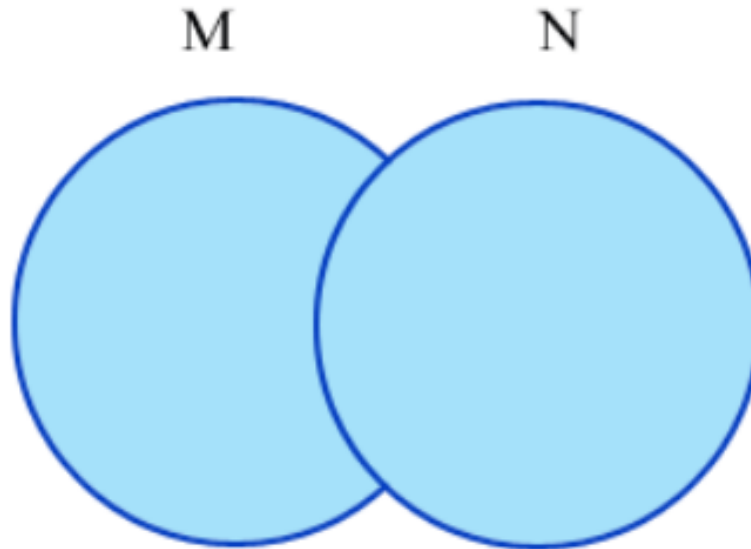
Union ve Union ALL

- İki veya daha fazla sorgunun sonuç kümesini, UNION ve UNION ALL operatörlerini kullanarak tek bir sonuç kümesinde birleştirir.

```
SELECT column_name(s) FROM table1  
UNION / UNION ALL  
SELECT column_name(s) FROM table2;
```

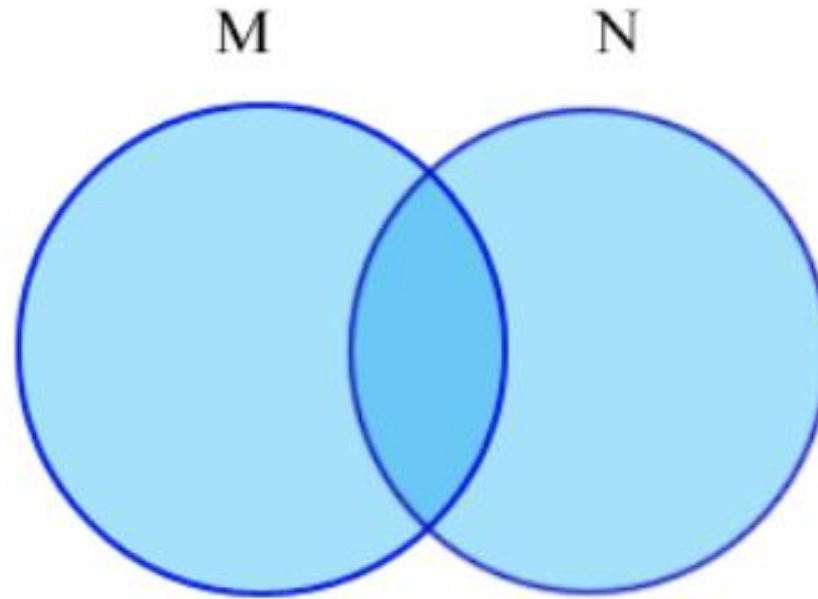
Union

- UNION operatörü, yinelenen satırları eledikten sonra her iki sorgudan da sonucu döndürür.



Union All

- UNION ALL operatörü, her iki sorudan tüm satırları döndürür ve hiçbir yinleme eliminasyonu olmaz.



Örnek Senaryo

```
CREATE TABLE Ogrenci(  
  ID INT IDENTITY,  
  Ad NVARCHAR(20),  
  Soyad NVARCHAR(50),  
  GirisTarihi DATE)
```

```
INSERT Ogrenci VALUES  
( 'Hakan', 'Gürbaşlar', '2017-01-01' ),  
( 'Nurullah', 'Çakır', '2017-01-01' ),  
( 'Faruk', 'Erdem', '2017-01-01' )
```

Örnek Senaryo

```
CREATE TABLE OgrenciMezun  
(ID INT IDENTITY,  
Ad NVARCHAR(20),  
Soyad NVARCHAR(50),  
GirisTarihi DATE)
```

```
INSERT OgrenciMezun VALUES  
( 'Hakan', 'Gürbaşlar', '2015-01-01' ),  
( 'Nurullah', 'Çakır', '2016-01-01' ),  
( 'Faruk', 'Erdem', '2017-01-01' )
```

Örnek Senaryo

```
SELECT Ad,Soyad FROM Ogrenci  
UNION ALL  
SELECT Ad,Soyad FROM OgrenciMezun;
```

	Ad	Soyad
1	Hakan	Gürbaşlar
2	Nurullah	Çakır
3	Faruk	Erdem
4	Hakan	Gürbaşlar
5	Nurullah	Çakır
6	Faruk	Erdem

Union Örneği

Employees tablosu

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
100	Steven	King	SKING	515.123.4567	17-Jun-87	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-89	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-93	AD_VP
103	Alexander	Hunold	AHUNOLD	590.423.4567	3-Jan-90	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	21-May-91	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	25-Jun-97	IT_PROG

Job History tablosu

EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
102	13-Jan-93	24-Jul-98	IT_PROG	60
101	21-Sep-89	27-Oct-93	AC_ACCOUNT	110
101	28-Oct-93	15-Mar-97	AC_MGR	110
201	17-Feb-96	19-Dec-99	MK_REP	20
114	24-Mar-98	31-Dec-99	ST_CLERK	50
122	1-Jan-99	31-Dec-99	ST_CLERK	50

Union

1	Select employee_id, job_id
2	FROM employees
3	UNION
4	Select employee_id, job_id
5	FROM job_history;

employee_id	job_id
100	AD_PRES
101	AD_VP
102	AD_VP
103	IT_PROG
104	IT_PROG
105	IT_PROG
106	IT_PROG
107	IT_PROG
108	FI_MGR
109	FI_ACCOUNT
110	FI_ACCOUNT
111	FI_ACCOUNT
112	FI_ACCOUNT
113	FI_ACCOUNT
114	PU_MAN
115	PU_CLERK
116	PU_CLERK
117	PU_CLERK

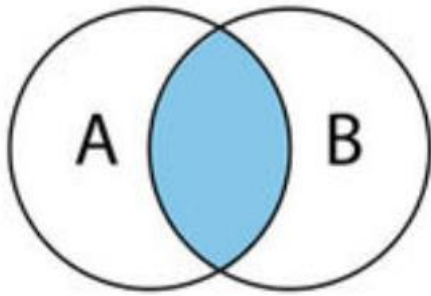
Union All

1	Select employee_id, job_id, department_id
2	FROM employees
3	UNION ALL
4	Select employee_id, job_id, department_id
5	FROM job_history;

employee_id	job_id	department_id
100	AD_PRES	90
101	AD_VP	90
102	AD_VP	90
103	IT_PROG	60
104	IT_PROG	60
105	IT_PROG	60
106	IT_PROG	60
107	IT_PROG	60
108	FI_MGR	100
109	FI_ACCOUNT	100
110	FI_ACCOUNT	100
111	FI_ACCOUNT	100
112	FI_ACCOUNT	100
113	FI_ACCOUNT	100
114	PU_MAN	30
115	PU_CLERK	30
116	PU_CLERK	30
117	PU_CLERK	30
118	PU_CLERK	30
119	PU_CLERK	30
120	ST_MAN	50
121	ST MAN	50

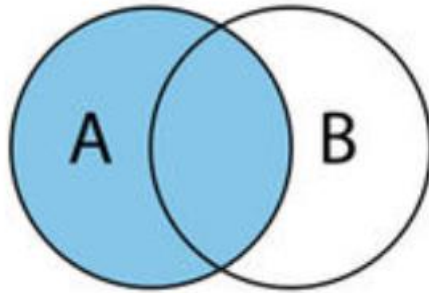
Union ve Join Özet

INNER JOIN



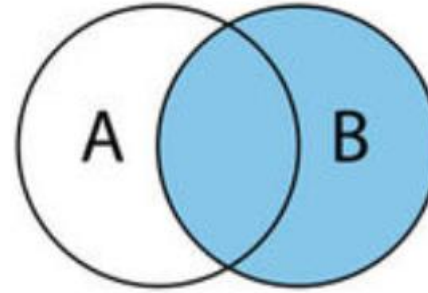
```
SELECT <select_list>  
FROM Table1 A  
INNER JOIN Table2 B  
ON A.Key=B.Key
```

LEFT JOIN



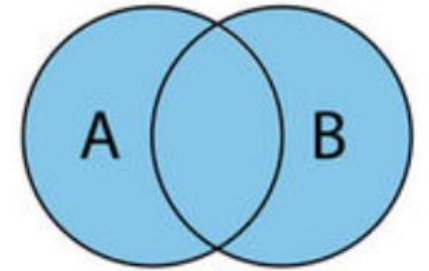
```
SELECT <select_list>  
FROM Table1 A  
LEFT JOIN Table2 B  
ON A.Key=B.Key
```

RIGHT JOIN



```
SELECT <select_list>  
FROM Table1 A  
RIGHT JOIN Table2 B  
ON A.Key=B.Key
```

UNION



```
SELECT <select_list>  
FROM Table1 A  
LEFT JOIN Table2 B  
ON join_condition  
UNION  
SELECT <select_list>  
FROM Table1 A  
RIGHT JOIN Table2 B  
ON join_condition
```